

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

S M C C M A N U A L

M A I N T A B L E O F C O N T E N T S

PAGE

BASIC SPECIFICATIONS. 1

START-UP PROCEDURE. 19

PROGRAMMING MANUAL. 49

PARAMETER AND COMMAND DIAGRAMS. 136

PARABOLIC SMCC DESCRIPTION. 145

USE OF ABSOLUTE ENCODERS AND RESOLVERS. 186

MEMORY AND I/O MAP. 198

PROGRAMMING EXAMPLES 222

E-POINT (JUMPER) DESCRIPTIONS 308

CONNECTORS LISTING. 318

CONNECTION DIAGRAMS 331

TIMING DIAGRAMS 343

SMCC SCHEMATICS 349

ACCESSORIES AND OTHER INTERCONNECTION DIAGRAMS. . . . 356

APPLICATION NOTES AND FURTHER EXAMPLES. 362

- - INDEX - -

EACH MAIN SECTION WILL HAVE ITS OWN TABLE OF CONTENTS

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

LIST OF DIAGRAMS AND TABLES

TABLE OF CONTENTS

PAGE

ACCELERATION: GENERAL S-CURVE. 162

ACCELERATION: MAX. NEGATIVE INFLECTION, PARABOLIC . . . 154

ACCELERATION: MAX. POSITIVE INFLECTION, PARABOLIC . . . 155

ACCELERATION: STRAIGHT LINE. 155

ACCESSORY 8H (JMACH), SMCC. 334

ACCESSORY 12/12A LCD DISPLAY 339

ACCESSORY 14 TO OPTO22. 359

ANALOG INPUT POTENTIOMETER CONNECTIONS. 342A

ANALOG SIP SCHEMATICS. 355A

BACKGROUND PLC 143

BLOCK DIAGRAM, SMCC 3

BRANCHES AND SUB-BRANCHES, PLC. 144

BUFFER FULL AND IN-POSITION TIMING IN MDI MODE. 346

BURR-BROWN TM2500 TERMINAL CONNECTIONS. 297

CIRCLE 4 MODE CALCULATIONS

TABLE 1 166
TABLE 2 167
TABLE 3 168

CIRCLES - NO STOP AND TILTED ELLIPSE. 232

CIRCULAR CUT ON THE FLY 261

CIRCULAR INTERPOLATION 120A

COMMUNICATION 345

CONCATENATED MOVE 371

CONNECTOR LOCATIONS AND PIN ASSIGNMENTS, SMCC 335

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DATA FLOW, SMCC 344

DECELERATION (STOP) TIMES FOR SPECIAL CONDITIONS. . . . 96

DEMO 1 - FLOWCHART. 242

DEMO 1 - PATH 243

DETECT MARK AND CUT 290A

DUAL RESOLVER CONNECTION TO SMCC. 193

E-POINTS AND TEST POINTS. 317

ELLIPSE, HORIZONTAL 231

ELLIPSE, TILTED 232

ELLIPSE, VERTICAL 231

ENCODER, TYPICAL CONNECTIONS DIFFERENTIAL DRIVER OR OPEN COLLECTOR OUTPUT, A/B QUADRATURE 342

FEEDRATE AND TIME CONTROL 100

FEEDRATE-BASED MOVE DIAGRAM 369

GENERAL PARABOLIC MOVE SEGMENT. 159

GENERAL S CURVE ACCELERATION. 162

GLUE DISPENSING 306

HANDWHEEL ENCODER CONNECTIONS, DIFFERENTIAL DRIVER OR OPEN COLLECTOR OUTPUT A/B QUADRATURE, SMCC 341

HANDWHEEL PROFILE GENERATOR 289A

i PARAMETERS 137

I/O CONNECTOR WIRING, PARALLEL. 333

I/O CONNECTOR WIRING, RS232 FLAT CABLE. 332A

INTERFACE BUSS TIMING DIAGRAM 347

J1 SCHEMATIC 352

J1, J2, J7, AND J8 SCHEMATIC. 350

J1, J10 SCHEMATIC 353

J2 SCHEMATIC 355

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J2, J12 SCHEMATIC 351

J2, J5, J6, J9, J11 SCHEMATIC 354

M AND C FUNCTION MAP, PARABOLIC SMCC. 139

M AND C FUNCTION MAP, STANDARD SMCC 138

M FUNCTION OUTPUTS BY PLC TIMER CONTROL 268

M FUNCTIONS INPUTS AND OUTPUTS, SMCC. 140

MINI-T OR SMCC TWO AXIS SYNCHRONIZED MOTION 228

MOTION BUFFER COMMANDS. 142

MOTION PROFILE, PARABOLIC 152

MOVE PERFORMED AT FULL SPEED. 386

MOVE PERFORMED AT HALF SPEED. 385

MOVE PERFORMED WITH THUMBWHEEL SET TO 15.00". 391

MOVE PERFORMED WITH THUMBWHEEL SET TO 30.00". 390

NON-BUFFER COMMANDS 141

PLC BRANCHES AND SUB-BRANCHES 144

PLC PROGRAM FOR CONTROLLING i14 AND SENDING MESSAGE TO HOST. 238

POINT-TO-POINT PARABOLIC MOVE 154

POINT-TO-POINT S CURVE MOVE 159

POSITION FEEDBACK DEVICES 197

RS232 I/O CONNECTOR (FLAT CABLE). 332A

RS232 TIMING AND HANDSHAKE. 348

S CURVE ACCELERATION 258

SMCC TO HYDRAULIC CYLINDER. 360

SMCC TO OPTO22 358

SMCC TO THUMBWHEEL. 357

SERPENTINE 266

SERVO AMP SIGNAL, POWER AND MOTOR, TACH AND ENCODER INPUT/OUTPUT CONNECTIONS. 340

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SERVO COMMAND OUTPUT TO STEPPER MOTOR DRIVER CONVERTER, SMCC 332

SERVO FILTER. 74

SINGLE ABSOLUTE BINARY ENCODER CONNECTION - SMCC FOR X AND Y AXIS 189

SINGLE AXIS 3-PHASE STANDALONE POSITIONING SYSTEM . . . 338

SINGLE RESOLVER CONNECTION - SMCC FOR X AND Y AXIS. . . 191

SINGLE PART S CURVE MOVE. 162

SQUARE WITH CIRCULAR CORNERS, SMCC. 230

STRAIGHT LINE ACCELERATION. 155

TERMINAL BLOCK FAN-OUT FOR SMCC CONNECTORS. 361

THUMBWHEEL, TYPICAL CONNECTION TO SMCC PARALLEL PORT. . 234

THUMBWHEELS, TYPICAL USE OF 276

TIME-BASED MOVE 369

TWO-AXIS BRUSH TYPE SERVO MOTOR POSITIONING SYSTEM. . . 337

TWO-AXIS CONCATENATED MOVE DIAGRAM. 371

TWO-AXIS SERVO MOTOR POSITIONING SYSTEM 336

TWO-AXIS SYNCHRONIZED MOTION. 228

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

LIST OF ADDITIONAL LITERATURE

(NOT INCLUDED IN MAIN MANUAL)

SMCC BROCHURE

SMCC PRICING SCHEDULE

SMCC-PC USER'S MANUAL (DOC. #602164-361)

SMCC-PC BROCHURE

SMCC-PC PRICING SCHEDULE

SMCC-VME USER'S MANUAL (DOC. #602163-360)

SMCC-VME BROCHURE

SMCC-VME PRICING SCHEDULE

SMCC-STDbus USER'S MANUAL

SMCC TERMINAL PROGRAM MANUAL

SMCCAD CONVERSION PROGRAM MANUAL

IBM PC BI-DIRECTIONAL PARALLEL PORT MANUAL (DOC. #602045-360)

CONTROL PANEL MANUAL

ACCESSORY 14 (I/O EXPANSION) MANUAL

ACCESSORY 18 (THUMBWHEEL MULTIPLEXER) MANUAL

ACCESSORY 19 (AMPLIFIER) MANUAL

AC INDUCTION MOTOR ARTICLE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC

ONE/TWO-AXIS CONTROLLER

P/N 602090-100

BASIC SPECIFICATIONS

TABLE OF CONTENTS

	PAGE
OVERVIEW	2
Block Diagram	3
PHYSICAL SPECIFICATIONS	4
ELECTRICAL SPECIFICATIONS	4
MEMORY CAPACITY	5
PERFORMANCE SPECIFICATIONS	6
I/O SPECIFICATIONS	8
OPTIONS	11
ACCESSORIES	13

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC
ONE/TWO-AXIS CONTROLLER
P/N 602090-100

BASIC SPECIFICATIONS

****

OVERVIEW

Delta Tau's SMCC (Smart Motion Control Card) is a third-generation state-of-the-art product for high-speed precision closed-loop digital servo control. It can control two coordinated axes if the motor phase commutation is performed within the motor (by brushes) or in the amplifier. Alternately, it can provide full phase commutation for multiphase motors (DC brushless, AC induction, stepper, switched reluctance) on a single axis.

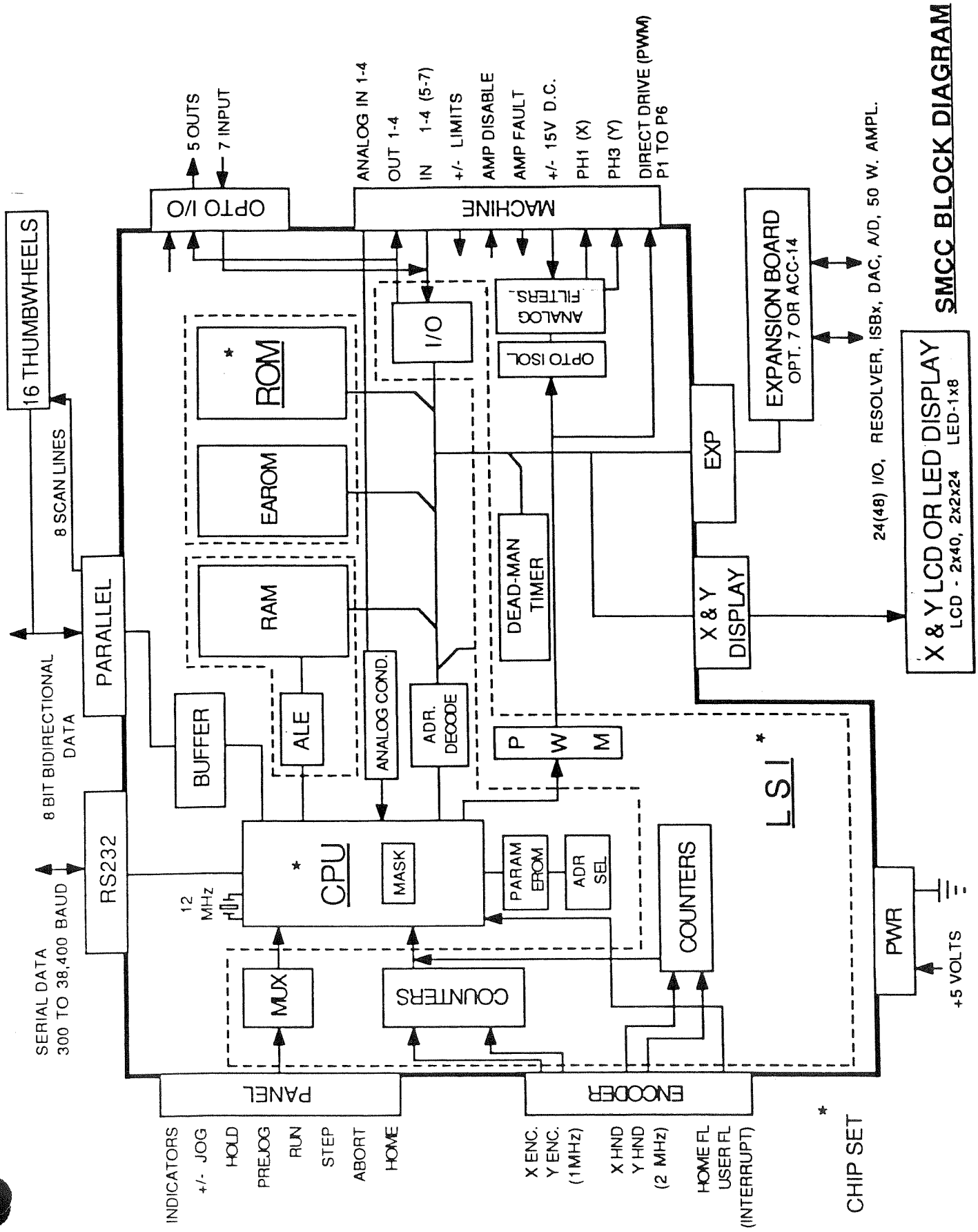
For two-axis control, the SMCC provides a +/-10V range command for self-commutating motor-amplifier systems using either tachometer feedback (velocity command) or current feedback (torque command). For single-axis control with commutation, the SMCC provides two +/-10V range commands to control phases directly; the third (and fourth, if necessary) phase commands are driven by balance loops. The card also provides PWM pulse outputs that can be used directly to switch the amplifier's totem-pole power drivers if the user desires. In both cases, phase commutation is handled directly by the card, which can generate sinusoidal, trapezoidal, or six-step phasing waveforms at up to 1000Hz. Stepper motors may be driven in either step or microstep mode in this manner. Also, an SMCC equipped with Accessory 8V (with a V-to-F converter) may be used to drive two stepper motors through standard stepper amplifiers as if the SMCC were an indexer.

The card is programmable by means of a high level command structure using ASCII alphanumeric characters, (see attached "Command Directory"). Communication with the card is via an RS232 serial channel with handshake or an 8-bit parallel bi-directional I/O port. The parallel port may also be used as a means for sending data to the card via thumbwheel switches.

The card also provides direct operator controlled input switches such as JOG, RUN, STEP, RETURN TO PREJOG POSITION, ABORT, HOME, RESET, and supports a one digit status display unit for error reporting, as well as two LCD or LED displays for position, velocity, and status indication.

Two A/B/C quadrature position encoder channels are provided for the X and Y axes as well as two additional encoder or pulse/direction inputs for electronic handwheel or external synchronization means. The card can control one or two

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

amplifiers/drivers and has dedicated inputs for +/- limits, home flag, amplifier fault, and disable. Also, seven each general purpose inputs and five outputs are provided.

An auxiliary I/O connector is provided for allowing I/O expansion to 48 lines, resolver operation, and general purpose analog signal generation and reception.

The SMCC may be daisy chained to allow synchronized and interpolated motions for up to 16 cards (32 axes).

PHYSICAL SPECIFICATIONS

SIZE: 9.2" x 6.3" x .75"
Designed to fit standard Double Eurocard chassis as well as vertical or horizontal plate mounting for standard industrial servo amplifier chassis, using two mounting screws.

Other configurations can be provided, such as Standard Bus, IBM PC Bus, Multibus, and VME bus.

WEIGHT: 0.3 lbs without mounting plate
1.2 lbs with mounting plate

OPERATING TEMP:
32 degrees F to 140 degrees F (0 degrees C to 60 degrees C)

STORAGE TEMP:
10 degrees F to 180 degrees F (-12 degrees C to 82 degrees C)

ELECTRICAL SPECIFICATIONS

POWER REQUIREMENT:
The SMCC itself requires only +5 Volts +/-5%, 0.8 amps with all options installed, and two encoders (HP7501) connected.

If Accessory 14 (old Option 7) is installed and has A/D, DAC, or R/D converter installed, the SMCC then requires an additional +/-15 volts supply at 100 mA each.

NOTE: The SMCC also requires separate isolated +/-15 volt power supplies for operating the optically coupled amplifier command output. This is normally provided by the amplifier. +15 volts require 50 mA, -15 volts require 15 mA. The voltage range can be +/-12 to 15 volts.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

AUXILIARY FUNCTION I/O:
Seven inputs: 0 to +5 volts, TTL.
Note that three of the seven inputs are used to display amplifier error codes when used with Delta-Tau's or other amplifiers equipped with error code generation.

Five outputs: 0 to +24 volts, DC 60 mA maximum. (A special "Opto" Connector is provided for flat cable connection to commercially available optical isolation devices).

An additional 48 individually programmable I/O lines may be used by adding ACCESSORY 14. Up to three ACC-14s may be used on one SMCC for a total of 12 + (3 x 48) = 156 programmable I/O lines.

RS232: +/-10 volts output.
The RS232 channel may be bussed to service as many as 16 cards in a daisy chain configuration. Daisy chaining provides automatic synchronization and allows bi-directional, handshake communications from 300 to 38400 BAUD.

PARALLEL: An 8 bit, bi-directional I/O port is provided for fast communication; it can optionally be used for inputting data via Thumbwheel switches. Optional DMA communication is available at data rates of three to four microseconds/character. The parallel port may also be daisy chained and provides automatic synchronization for up to 16 cards.

NOTES:
1) RS232 and parallel use special Tri-Stated outputs and require that each SMCC be individually addressed (A0 to Af) before data is sent back to the host.

2) Each SMCC on the bus may be used to control one or two axes, with or without commutation.

MEMORY CAPACITY

EXECUTIVE CONTROL PROGRAM:
8K in CPU MASK (Chip Set).
MC3A is Standard CPU, "PARAB" in Parabolic CPU
32K in external PROM.

NOTE: Chip Set consists of CPU, PROM, and LSI unit with address decoder PAL.

USER PROGRAM BUFFER:
8K RAM for 1,790 Program Step Storage (Standard).
32K RAM for 5,630 Program Steps (Optional, OPTION 5).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PERMANENT MEMORY STORAGE:

1) 512 x 8 EAROM
For permanently storing parameters and program steps. On the Parabolic card, 'i' parameters and 80 program steps can be stored if this is the only permanent memory. On the Standard card, 'i', 'C', 'M', and 'P' parameters plus subroutines and label locations, and seven program lines can be stored if this is the only permanent memory.

2) 8K x 8 EAROM (Optional, OPTION 2)
With the addition of this option, 1790 program steps, plus all 'i', 'C', 'M', and 'P' parameters, and subroutines and label locations can be stored on either Standard or Parabolic cards.

3) 32K x 8 Battery Backed RAM - OPTION 5
With the addition of this option (instead of Option 2), 5630 program steps, plus all 'i', 'C', 'M', and 'P' parameters, and subroutines and label locations can be stored on either Standard or Parabolic cards.

PERFORMANCE SPECIFICATIONS

SERVO CYCLE TIME:

One axis/two axis control servo loop time updates position, calculates needed correction, and outputs signal to amplifier in specified time. All servo times include a 65% duty cycle which means that 35% of the time specified is used for background tasks.

1) Handwheel only (Slave Follower)
150/300 microseconds

2) Normal Position and Velocity Loop
480/960 microseconds

3) Brushless Direct Drive Servo Loop Time
360 microseconds

4) AC Induction, 600 microseconds

5) Data Blocks/Seconds
Execution rate of 60/100 blocks/seconds or approximately 10/16 milliseconds/block when controlling two axis/one axis in linear mode.

VELOCITY RANGE:

.005 to 1,000,000 encoder counts/second Digital Phase locked loop, crystal controlled. Smooth operation to low RPMs, dependent upon encoder resolution used. Enhanced low speed operation is possible using optional 1/T (period) encoder measurement.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

VELOCITY ACCURACY:
(Long Term) .002% absolute accuracy
(Short Term) system dependent, typically 0.5% to 1%
(Multi-Axes) .001%, adjustable to zero following error between axis

VELOCITY REPEATABILITY: .001%

POSITION RANGE: +/-67 million encoder counts

POSITION ACCURACY: +/- one encoder count

POSITION ENCODERS:
Two X and Y independent channels for position control. A/B quadrature with "C" (home) channel. X1, X2, and X4 multipliers. Single ended or differential TTL. Maximum input rate after multiplication, is 1,000,000 counts/second with X4 multiplier. Using X4 multiplier actually reduces rate to approximately 800,000 counts/second due to encoder quadrature imperfection. Pulse and Direction input is also provided. Maximum actual encoder input rate is 500 KHz in X1 and X2 mode and 250 KHz in X4 mode.

HANDWHEEL ENCODER:
Two X and Y independent channels for use as quadrature encoder inputs or pulse and direction inputs (Stepper Motor Replacement). Maximum rate after multiplication is 2,000,000 counts/second with X4 multiplier. Maximum rate is approximately 1,800,000 and depends upon quadrature imperfection. Maximum encoder rate is 500 KHz. The handwheel encoder input can be used in four different modes:

a) Direct position offset moves motor and changes position counter, with selectable transfer ratio.

b) Position translation moves motor shaft but does not change position counter, with selectable transfer ratio.

c) Program execution rate (programming time base) varies as a function of external handwheel input frequency. Input frequency may be selected for 1:1 time base. SMCC's execution rate will vary linearly with input frequency, including zero frequency.

d) Handwheel only mode where the SMCC's normal program execution capability is limited in order to allow for very tight servo following capability of the handwheel encoder.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

I/O SPECIFICATIONS

INPUTS/OUTPUTS TO AMPLIFIER DRIVER:

Error output signal (command signal to amplifier), amplifier disable output signal, and "Amplifier Fault" input are optically isolated. All three of these signals use the amplifier's +/-12 to 15 volts power supplies, which are referenced to the amplifier common in order to achieve total isolation between the SMCC's digital circuits and the power section.

ANALOG OUTPUTS:

These are used as velocity or torque commands to the amplifier input.

1) Brushless three or four phase current demand torque outputs. Selectable, sinusoidal, or trapezoidal or six-step waveforms, +/-10 volt amplitude, 1000 Hz maximum frequency. Two outputs are provided 120 degrees apart for three phase and 90 degrees apart for four phase.

2) Phases one and three, when so designated, are used for two axis standard +/-10 volts DC (velocity) error output with 9 bits of resolution.

DIGITAL OUTPUTS:

Six direct PWM digital outputs are available for driving one or two brush type or one three or four phase brushless or stepper Motor by using the digital outputs to directly control two, three, or four totem pole drivers. The six outputs are pulse width modulated and provide totem pole overlap drive protection to prevent turn on of the top and bottom drives simultaneously.

ANALOG INPUTS:

Four analog inputs -- CUR0 (ANA0), CUR1 (ANA1), ANA2, and ANA3 -- are provided on the card and they serve the following functions:

1) CUR0 and CUR1 are dedicated to measuring the motor/amplifier current levels for the X and Y axes, respectively. These can be used for overcurrent protection if Option 3 (analog signal conditioning) is installed on the card. They should receive a voltage proportional to the absolute value of the current -- with a range of 0 to +10 volts. Two adjustment potentiometers are provided for proper scaling of these inputs. From the inputs, the card performs I-squared-t calculations to prevent the motor from being overdriven. Parameters i32[i52] and i33[i53] control the level settings for this protection. (It is possible to connect these inputs for

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

current protection without Option 3; consult the factory for details.)

2) ANA2 and ANA3 may be used to control motion speed in several different modes. The mode is set by the i14 parameter (refer to the i14 description for details of operation). These inputs may be used with or without the Option 3 analog signal conditioning: with Option 3 the input voltage range is -10V to +10V; without it the range is 0 to +5V. The modes are:

   a) Time Base (Feed Rate) Control mode: In this mode, the voltage at ANA2 is used to control the speed of execution of all moves, programmed or jog, for both axes. This control is identical to that provided by the '%' or 'POT' commands, or by the handwheel input frequency.

   b) Jog speed control mode: In this mode, ANA2 controls the speed of jog moves for the X-axis, and ANA3 for the Y-axis. Move timing and direction are controlled by the standard jog inputs or commands.

   c) Continuous speed control mode: In this mode, both axes are always in a continual jog state, with the speed for each axis controlled by its analog input. There are four speed ranges, selected by the jog inputs for the axis.

Refer to the Analog SIP (Option 3) Schematic for details of the conditioning circuitry; refer to the Analog Input Potentiometer Connections diagram for instructions on wiring the analog inputs with or without Option 3.

OTHER I/O :

The SMCC has several dedicated lines serving the amplifier/machine. They are:

1) +/- limits for X and Y axes.

2) Amplifier disable output (OPTO - Isolated).

3) Amplifier fault input (OPTO - Isolated).

4) Home Flag inputs for locating the encoder "C" channel or reference location input (interrupt driven).

5) User Flag inputs for precision, interrupt driven synchronization to external events.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

6) Five general purpose outputs and rated at 24 volts, 60 mA. Seven general purpose inputs are also provided, TTL compatible, to +5 volts maximum.

MANUAL CONTROLS:

The SMCC supports several dedicated direct inputs allowing manual operation by using momentary toggle switches.

+/- JOG,(j,J): For X or Y individually

HOLD,(H): Stops motion immediately

PREJOG,(=): Return to prejog or pre-handwheel position

START,(RUN): Starts continuous program execution

STEP,(S): Executes one block of data at a time

ABORT,(q): Stops running, halts immediately, resets program pointer to next block

HOME,(h): Starts X and Y axes on a home cycle

RESET (INIT.), ($,$$): Initializes the SMCC.

NOTE: All above functions are supported by ACC-16 Control Panel.

EXPANSION OUTPUTS:

The JEXP connector on the SMCC card supports the ability to make use of existing Delta Tau expansion modules for greater I/O capability and positioning with absolute encoders (8 to 23 bits parallel binary input) or resolvers single or dual speed (geared). (See Accessories 14 and 15)

ERROR AND STATUS DISPLAY:

The SMCC is equipped with a one digit hexadecimal display which is used to indicate operational status to the operator.

.(Left Decimal) = In Position (i01)
.(Right Decimal) = Following error indicator (i10)
0 = O.K. (SMCC is happy)
1 = Hall effect failure (Brushless Motor)
2 = Following Error Failure (i00)
3 = Parabolic only, not enough time to calculate move [i12 or t (fatal error)]
4 = Circle radius too small for specified a
5 = "Go Until" command did not receive

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

a trigger
6 = Not Used
7 = Communications Error (Invalid Character)

8 = AMPlifier failure, (AMPlifier without Fault Data Bits).
*9 = SHUNT-FAULT
*A = GROUND-FAULT
*B = +15 or -15 = LOW VOLTAGE
*C = Floating +15V (Bias Supply) = LOW VOLTAGE
*D = OVER-TEMP
*E = BUSS-OVER-VOLTAGE
*F = DRIVER-FAULT, (OVER-CURRENT, BUSS-UNDER VOLTAGE)

NOTE: Delta Tau's amplifiers provide three fault indication bits which are used to isolate the cause of the problem and display it. Please note that physically the indicator may be mounted to accommodate both the stand alone and the rack-mount configuration for visibility. An extension cable and mounting bracket are needed for the rack-mount version.

DISPLAYS: The SMCC supports two types of displays:

a) LCD - Liquid crystal displays 1 x 24, 2 x 24, 1 x 40 for X and Y individually or a 2 x 40 single display for both axes. Displays indicate position, velocity, following error, and status.

b) LED - Light emitting diode (red) 1 x 8. 3/4 inch high digits separate displays for X and Y axes. Displays position only.

OPTIONS

The SMCC is provided with the following options (OPTIONS 1 through 5):

1) PARALLEL INPUT/OUTPUT:
For 8-bit bi-directional communications with handshake and fast DMA communications. Thumbwheel programming is also available via the parallel port (see ACC-18 for further information on use of Thumbwheel inputs). Eight additional outputs are provided for the purpose of scanning thumbwheels or for general purpose use. The parallel port may be used for:

a) Communications
b) Thumbwheel inputs
c) General purpose 8 inputs and 8 outputs.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2) PERMANENT PROGRAM STORAGE OPTION (8K):
Even though a small 512 x 8 EAROM is standard on the SMCC and provides parameter and a few program lines of storage; subroutines and labels cannot be saved. A large 8K x 8 EAROM is optionally available for permanently storing 1,790 programming lines, subroutines, and other label assignments.

3) ANALOG INPUT CONDITIONING:
Although 0 to +5 volt analog signals can be directly accepted by the SMCC, the conditioning module allows +/-10 volt signals to be accepted and also provides for "Absolute" value inputs (positive or negative) voltages to be accepted for motor current feedback.

4) PARABOLIC PATH GENERATOR:
This option uses a custom masked CPU to provide additional special functions and features such as:

a) Parabolic Path Generation: All points are defined by position, velocity at destination, and time to make the move. Third order algorithms are used to provide smooth, continuous path and interpolated motion control, with zero acceleration at move segment boundaries. Includes special "CIR4" (three point smoother mode) with "S" curve acceleration capability.
b) DMA (Direct Memory Access) Communication: Both in RS232 and Parallel for transmitting and receiving binary data.
c) Data snapshot and storage capability at servo rate of position following error and velocity as well as I/O ports or other registers. Used for analyzing servo and machine performance.
d) 1/T (period) measurement of encoder pulses for greatly improved low velocity operation and stiffer servo due to higher usable gains.
e) "Go Until" Command (^d) which allows new destination on trigger and allows synchronization of motion with external events.
f) A "TORQUE" Command which allows control of torque applied to the load and detection of torque limits.
g) Improved AC induction motor control due to faster servo cycle in masked CPU.
h) Addition of velocity and acceleration feedforward gain terms in the servo algorithm. This provides the ability to greatly reduce following error.
i) The ability to provide altered destination by reception of a special command followed by a new destination.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

5) EXPANDED PROGRAM MEMORY:
Provides a 32K byte RAM for storing up to 5,630 program steps instead of the standard 1,790. The 32K RAM is battery backed and is therefore used as both permanent memory and program buffer. It replaces an 8K byte RAM and the 8K byte EAROM of Option 2. Note that any change made to program or parameters is automatically saved - there are not separate active (volatile) and permanent memories. Also note that you cannot change parameters or programs and still expect the old parameters or programs to come back when you reset the board.

ACCESSORIES

Many accessories are provided with SMCC as a convenience and in order to simplify its application in various systems. These are:

1) +5 VOLTS POWER SUPPLIES - For 1 to 16 card systems.

2) +/-15 VOLTS POWER SUPPLY - For Accessory 14 or SMCC's Option 7 or for powering the optically isolated outputs of SMCC when the amplifier is not so equipped.

3) 10 FOOT RS232 or PARALLEL COMMUNICATION CABLE (specify which type is needed) - For connecting the host computer to as many SMCCs that are on a daisy chain. When using the cable for daisy chaining several SMCCs, specify how many cards will be on the daisy chain; the cable is then made with the required number of SMCC connectors (J2's mates) spaced six inches apart, unless specified otherwise.

4) Additional INSTRUCTION MANUAL

5) MATING CONNECTOR for J1 (JMACH) and J2 (JPAR-RS232) - Connectors (96 pins) each with solder tails on connector to provide access to all pins.

6) HANDWHEEL ENCODER - Hewlett Packard HEDS-7501, with 256 counts per revolution, A/B quadrature. Mounts like a regular potentiometer and has 1/4" shaft with built in friction.

7) IEEE 488 CABLE - To be used with Accessory 13.

8) INTERCONNECTION TERMINAL BLOCK (P/N 602065-103) - Consists of a small printed circuit board with a mating connector for SMCC's J1 (JMACH) connector on one side and terminal strips on the other; for connections to the amplifier, encoder, and various machine inputs and outputs. Three versions of ACC-8 are available.

a) Horizontal version (ACC-8H) - When mounted to SMCC, is perpendicular to the plane of SMCC.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

b) Vertical version (ACC-8V) - When mounted to SMCC, extends vertically in the same plane as SMCC. This version also has three additional features:

1) It provides built in X- and Y- axis V-to-F converters, for using the SMCC's Phase 1 and 3 outputs as frequency generators (0 - 10 or 100 KHz). The output frequency is then used to drive any stepper motor driver (full, half or microstep) in pulse and direction mode. The same output frequency is used to provide position information back to SMCC, thus generating a nearly ideal output profile. If desired, an encoder may be used on the stepper motor to provide actual position feedback.

2) When SMCC is commutating a brushless DC or AC induction motor, it provides a summing circuit for adding SMCC's Phase 1 and 3 outputs to create Phase 2 when driving current amplifiers requiring all three phases for commutation. SMCC normally outputs two phases and expects the amplifier to create the third as a balance loop.

3) It also provides a means to invert or not invert the FAULT input to SMCC from the amplifier since some units provide a low true output and SMCC requires a high true input to detect an amplifier fault.

c) Remote version (ACC-8R) - This version of ACC-8 does not mount on the SMCC itself but is designed to be connected to SMCC via a 64 wire flat cable terminated with connectors at both ends. ACC-8R is mounted separately and is 3.0 inches wide by 10.5 inches long. In addition to the three features contained in ACC-8V, it has an additional one which consists of two small, 30 watt DC amplifiers that may be used for driving either two small DC motors or proprotional hydraulic valve actuators. This amplifier feature requires a +/-15 volt power supply, and is capable of sourcing or sinking two amps.

An additional feature of ACC-8R is that it can be used by the SMCC, the SMCC-PC, SMCC-VME, and SMCC-STDbus.

9) SMCC IBM PC SOFTWARE DISKETTE - Provides a means to write, edit, and save SMCC programs on an IBM PC. The current version is SMCC V3.61 with "Dump" capability.

The "Dump" routine allows sending files larger than the SMCC's buffer can hold on a continuous basis. Mostly used with SMCCAD program.

Version 3.61 also provides an interactive procedure for tuning a servo amp. and motor combination and provides means for direct communication on the PC bus with the SMCC-PC. When performing data gathering and plotting operation, V3.61 provides all the needed facilities for printing or displaying the data.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

10) PARALLEL, BI-DIRECTIONAL 8-BIT I/O PORT FOR IBM PC (P/N 601793-100) - Fits on PC's bus and is a half-sized card. Provides handshake signals to SMCC and supports interrupts on the PC bus. Needed when SMCC's parallel port will be used for communications. The output from this card can be flat-cable connected to the SMCC's parallel port. Using the parallel port provides a 4 to 8 fold increase in data throughput between the PC and the SMCC.

11) RUGGEDIZED CONTROL PANEL (P/N 601810-100) - This unit is a complete, stand-alone control panel with membrane touch-keys and 2 x 24 display unit. A local CPU permits programming the panel to achieve various functions. Communicates with SMCC via RS232, and stores 64 programs. Normally this device is used in dedicated applications and is custom programmed. Such a custom program already exists for use in back-gauging systems for shear and press-brake applications, and in another version, is available as a simple terminal for operating and programming the SMCC.

12) DISPLAYS - Three types of display are provided for SMCC. They all plug directly into the SMCC's connectors provided for this purpose:

ACC-12 - 2 line x 24 digit alphanumeric LCD display. Two units may be used to provide independent X and Y displays where each display has all information per axis, or one unit with X and Y information, where only position and velocity is displayed per axis.

ACC-12A - 2 line x 40 digit Alphanumeric LCD display. One unit displays all information for X and Y, position; velocity following error and status.

ACC-12B - 1 line x 8 digit RED, highly visible, LED display. Displays position only. Two units may be used, one for X and other for Y. Operation of all displays is software selectable by SMCC. Either a single display for XY or two separate displays for X and Y may be used.

13) IEEE488 (P/N 601790-100) - This accessory is a separate card and provides communications with the SMCC via the IEEE488 bus. One ACC-13 may be used to control as many as 16 SMCCs. ACC-13 performs the function of a talker/listener for a host computer equipped with IEEE488 and uses the parallel port to talk to SMCC.

14) I/O EXPANSION MODULE (P/N 602158-100) - A PC board which may be mounted piggyback on the SMCC or standalone with its own mounting plate. It provides 48 individually programmable I/O lines and uses connectors which are flat cable compatible with OPTO-22 (PB8, PB16, and PB24) or other manufacturer's optically isolated modules. It connects to SMCC via a flat

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

cable connector to the JEXP connector. The maximum cable length between SMCC and ACC-14 should not exceed 12 inches due to high speed signals. A six inch cable is provided with each ACC-14.

Additionally, two iSBX connectors are provided for allowing the use of any iSBX bus compatible products to be used with ACC-14 such as: A/D, D/A, R/D converters, etc. iSBX is a registered Intel Corporation bus structure.

Another use for ACC-14 is to allow the use of parallel binary data from 8 bits minimum to 23 bits maximum to be used as position feedback data from an absolute optical encoder. The SMCC has the ability to use this type of parallel input data for positioning. Also, when using the R/D converter, the SMCC is capable of using one R/D essentially as an encoder, but it is also able to handle dual resolvers and R/D converters, a "fine" on the motor shaft and a "coarse" on the output shaft, with a reduction of between 2 and 256 between the motor and the output shaft, thus providing an absolute position detection capability over the total distance traveled.

Up to three ACC-14s may be daisy chained to serve one SMCC. The SMCC to ACC-14 flat cable connector is then specifically made to daisy chain all the ACC-14s.

The ACC-14 Boards are provided in three basic forms:

1) All items installed. 48 I/O and the iSBX connectors.
2) 48 I/O only
3) iSBX connectors only.

15) RESOLVER TO DIGITAL CONVERTER (P/N 602159-100) - This module is designed in the iSBX bus format and can be used in conjunction with ACC-14 to provide two channels of R/D conversion capability to the SMCC so as to enable it to control motors equipped with resolvers. The R/D converter provides 10, 12, 14, or 16 bits of resolution and provides sinusoidal excitation voltage of 8 volts RMS at 5 KHz and requires sine/cosine feedback of 2 VRMS minimum amplitude, to 8 VRMS. The R/D converter also handles geared resolvers for absolute position applications where two resolvers, a "coarse" and a "fine", are used with specified gear ratio (see ACC-14). When a resolver is used on the motor it can also be used by SMCC to provide phase commutation for brushless DC motors without the need to do a "phase finding" search on power turn-on.

16) CONTROL PANEL - This accessory provides a 2 x 40 LCD display, a handwheel encoder, power on/off control, manual control switches such as X and Y jog +/-, reset, hold, quit, etc., and LED status indicators. It can be used either standalone or mounted on the customer's panel as part of the machine controls. It plugs directly into SMCC's JPAN and JDISP connectors via two flat cable connectors. Optionally the

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

control panel provides an A/B quadrature oscillator (0 to 100 KHz) to be used as time base by SMCC.

Power on and off switches are provided and may be used with external relays to provide a latching power control circuit. The SMCC's "error" output can then be optionally used to cause an automatic power turn off.

The control panel is also recommended as a field trouble shooting or startup tool since it provides direct manual control of SMCC functions and permits a complete diagnostic to be run on SMCC which tests the following:

1) All manual controls, home, and limit switches
2) Encoders, position, and handwheel
3) Allows the reloading of default i parameters for operating SMCC with DC brush, DC brushless, and AC induction motors.

17) SMCCAD - This is a software diskette for IBM PC allowing the use of AutoCad, Design Cad, and other Cad systems to be used as a means for generating two dimensional programs for SMCC. SMCCAD uses the HPGL output files from the CAD system to derive SMCC compatible motion commands. The pen UP/DOWN and pen color selections are used as means to specify subroutines for actuating various peripheral devices including "z axis" movement. SMCCAD allows the generation of any type of pattern including alpha and numeric characters. The program also has the ability to maintain constant feed rate, swap and/or reverse axis, provide scale factors, etc.

18) THUMBWHEEL MULTIPLEXER (P/N 602178-100) - This accessory is a printed circuit board which provides needed circuitry for reading 16 thumbwheel switches via the parallel port of the SMCC. Up to 32 thumbwheel multiplexer boards can be daisy chained to permit the reading of 512 thumbwheels or any other TTL level inputs by the SMCC. The thumbwheels may be either plugged into the multiplexer board or they may be wired to it using 16-pin DIP plug-in cables. Also, for simpler requirements, provisions are made for using 8 position DIP switches which can be used to input data to SMCC. The thumbwheel multiplexer board also provides a convenient access to the M function inputs and outputs of SMCC by providing a flat cable connection to the J OPTO connector and a terminal strip for easily connecting external inputs and outputs. Each of the I/O lines are also provided with LED indicators.

Please note that if only two thumbwheels are to be used they may be directly connected to SMCC's parallel port without the use of the thumbwheel multiplexer or isolation diodes. Also, it is possible to connect up to 16 thumbwheels to the SMCC by using isolation diodes only, however, this method is not recommended since the noise immunity of the system is marginal.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The thumbwheel multiplexer is provided as a basic board with four options. The basic board consists of the needed circuits for addressing (selecting) and reading each of the 16 thumbwheel locations on the board and a connector for connecting the board to the SMCC. The interconnect cable is not provided but may be purchased (specify desired length).

Option 1 - Daisy Chain Expansion Connector
This is used to allow daisy chaining to the next thumbwheel multiplexer board for reading more than 16 thumbwheel switches. Each board has an address selector 5-position DIP switch for setting the address of each card, 32 maximum.

Option 2 - DIP Switches, 8 position, P/N 76SB08S or 76SB08 by Grayhill
For using the thumbwheel multiplexer without thumbwheels, but with DIP switches as data entry devices. Each DIP switch serves as a two digit (4 bits each) of BCD or binary data.

Option 3 - J OPTO
Option provides the convenience of connecting the thumbwheel multiplexer board to the SMCC's J OPTO connector and provides a visual indicator (LED) for each of the general purpose inputs and 5 outputs as well as a terminal strip for access to the I/O lines.

Option 4 - Provides thumbwheels and sockets for thumbwheels. Sockets are soldered by customer at the desired locations on the board and thumbwheels are then plugged in.

Thumbwheel by Cherry or equivalent P/N T20-02A
012-0347 Tie Rod
012-0744 Fastener
009-1153 End Caps

Thumbwheel Socket by Edac or equivalent
P/N 307-006-501-104 ALT. 306-006-521-101

19) AMPLIFIER(P/N 602173-100) - A small 50 watt amplifier is provided: 24 volts, 2 amps continuous, 4 amps peak. The amplifier is iSBX mounting compatible and can therefore be used on ACC-14 for driving two axes of small DC motors.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC START-UP PROCEDURE

TABLE OF CONTENTS

PAGE

POWER-UP 20

CONNECTING SMCC TO HOST 21

SMCC OPERATION VERIFICATION 23

CONNECTING STANDALONE SMCC TO SYSTEM 26

CONNECTING HOST, SMCC, AMPLIFIER/MOTOR, MACHINE. 29

SYSTEMS WITH TACHOMETER USING PARABOLIC OPTION 32

SYSTEMS WITHOUT A TACHOMETER USING PARABOLIC OPTION . . . 35

BRUSHLESS DC MOTOR SETUP 36

AC INDUCTION MOTOR SETUP 38

SMCC WITH RESOLVER 39

Parameters For Parallel Encoders/Resolvers. 42

PROGRAMMING SMCC, GETTING STARTED. 43

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC START-UP PROCEDURE

****

POWER-UP

This section concerns itself with first-time-out-of-the-box power-up procedure for the "Smart Motion Controller Card" (hereafter referred to as "SMCC").

Five volt DC power can be applied to the SMCC at J10 Pin 2 = +5V and J10 Pin 1 = GND or at J1 Pin 6 Row A, B, or C = +5V and J1 Pin 7 Row A, B, or C = GND. Only one of these pairs need be used. Current required is one ampere.

The recommended way to get started with the SMCC is to apply five volts DC power to it and observe the local LED status display. The display should read ".0." for about 1/2 second, then read ".0". If this is the case, you can now start to communicate via the RS232C or parallel port. The SMCC was shipped to you running as an A0 CARD communicating at 9600 baud, EAROM SAVE ENABLED, NO SWITCH OR JUMPER changes should be necessary to communicate at 9600 baud.

If the status display shows a ".0." (two dots, one on each side of the 0), it is an indication that SMCC's program is not running and that a hardware problem exists. Check the following:

1) SMCC must be addressed as A0 in order to function correctly as a standalone card and should have E2 jumpered (and E3 not jumpered). Any other address, A1 to Af, requires that E3 jumper be installed on the card and that an A0 card with E2 jumper installed be on the communication daisy chain in order to function properly. (In other words, all card addresses other than A0 require a "Sync" Clock from the A0 card in order to function properly.) Check and be sure your card is an A0 by examining the SW DIP switch on SMCC and insuring that the first four switches are set to "Closed" position. (Refer to drawing 602154, SMCC E-Point and Test Point.)

2) Verify that all chips are plugged in correctly and none of the socketed I.C.s are loose. During power-up cycle, SMCC verifies RAM and several other hardware devices. Any unplugged chips may cause the card to malfunction.

3) Be sure your +5 volt supply is between 4.75 and 5.25 volts. Use a voltmeter to verify voltage, since even if the green "Power On" LED is on, the power supply may be out of tolerance. Also, note that only +5 volts is needed for this test.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If SMCC's status display shows ".0", you are ready to proceed since this means the SMCC is okay ("0") and is in position ".0" inside the band specified by (i01) parameter (in-position band). The right decimal point, "0.", indicates a non-fatal following error (i10). If both decimal points are on (".0.") it is an indication that SMCC is not functioning correctly (see previous section). If just a "0" is showing, it means SMCC is neither in-position nor inside the non-fatal following error band. This condition should not occur during this power-up test since an encoder is not connected to SMCC and it should therefore be within the in-position band.

CONNECTING SMCC TO HOST

Before connecting the SMCC to the amplifiers and motors, it is a good idea to connect the host to the SMCC and run a test to ensure that the host and the SMCC are communicating properly.

The terminal can be an IBM PC or compatible running the "SMCQ" or "SMCC V3.X" IBM PC terminal programs, or it can be any type of computer or "Dumb Terminal" running in the terminal mode at 9600 baud, 1 start bit, 8 data bits, no parity, 1 stop bit, hardware handshake (RTS,CTS), half duplex. The DSR, DTR signals are not used by the SMCC but are tied together at the SMCC. If the terminal cannot run at 9600 baud, select a different baud rate using the SW DIP switch.

Connect the SMCC to the terminal via the supplied RS232 cable (Accessory 3) with +5 volts power applied to the SMCC and the terminal turned on.

Refer to SMCC V3.X Manual for a description of operation with an IBM PC host.

Send a "U" (75H) ASCII character, followed by <CR> (0DH), carriage return, to the SMCC. The local LED display should read ".7" which indicates a bad or unrecognizable command was received since "U" is not a valid command. If this is true, the terminal is communicating with the SMCC. To clear the error, simply send a <CR>(0DH) character. The status display should now display ".0".

If you are having a problem establishing communication via RS232 from the SMCC to the host computer, here are some possible causes:

1) No power on SMCC.

2) RS232 cable not plugged in correctly or improperly wired.

   The minimum number of interconnect wires needed to communicate with the SMCC from the host computer are:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        J2-A12 <---------------------- 2 TXD/
        J2-A11 ----------------------> 3 RXD/
SMCC    J2-A10 <---------------------- 4 RTS   HOST
        J2-A9  ----------------------> 5 CTS
        J2-A7  ---------------------- 7 GND
```

3) SMCC baud rate and/or card address set wrong. Unit is shipped at 9600 baud, addressed at A0, EAROM enabled. (All switches on SW switch closed.)

4) Host baud rate set wrong. Host should be set to 9600 baud, 8 bits, 1 start bit, 1 stop bit, no parity.

5) Signal swap on TXD1, RXD1 and RTS, CTS. Check configuration jumpers E7 and E5. Unit is shipped with data input on "TXD/" and data output on "RXD/", handshake input on "RTS", handshake output on "CTS". Some hosts require the opposite. Use E7 and E5 to swap signals or rewire your cable.

6) Host computer is not setting the "RTS" handshake line high.

7) If communication is erratic, check to see if:

   a) Data cable is too long,
   b) Not shielded,
   c) Too close to high noise environment, such as a power line or amplifier and motor lines.
   d) Cable has too much capacitance for data rate being used. For long data runs use Belden 9863, 3 Pair 124 Ohm Twinax AWM, style 2094, or equivalent cable which has low capacitance and is especially made for long distance data transmissions. In general, RS232 cable length may be four feet for 38,400 bauc and the distance can be doubled for every reduction of baud rate by half. This rule may vary greatly depending upon the electrical noise in the environment and the cable type and quality you are using.
   e) If you are using the SMCC V3.X IBM PC based software, be sure you have selected the proper communications delay setting for your computer. Refer to SMCC V3.X instructions. If you are in doubt, set the delay (clock speed multiplier) to nine. This will ensure sufficient communications delay, although it may make program response sluggish.

Once you've established good RS232 communications, type i031 <CR>. This is the default value and should place the SMCC in a "prompt" mode. The SMCC will echo or prompt with a "line feed" LF (0AH) character when it receives a "carriage return" <CR> (0DH) character which terminates an instruction. In case an error in communications is detected, such as a wrong command or unrecognizable character, the Standard SMCC sends back a "?" character, while the Parabolic SMCC sends the "bell" character, which emits an audible character when received by a terminal or PC. The prompting feature may be disabled by setting i03 to 0.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Also, remember that in order to receive information from the SMCC, you must address it first (once after each power up or reset) even if you are using a single card (A0). Use the "An" (n=0 to f card number) Command followed by a carriage return. You may then send the following error "f", position "p", or status "?" requests one at a time, followed by <CR> to which the SMCC will promptly respond. In this case, both position and following error will return a (0). Also, you may request the SMCC to send its "%" feedrate value by typing % <CR> or its i parameter values by typing inn <CR>, where nn is 00 to 99 for the parameter you want to see. An "ix" requests all parameters. The SMCC will send back the stored value of the parameter(s) you requested. Please refer to "Command Summary" for a list of all commands that may be used to request various information from SMCC. If you are using the SMCC V3.X IBM PC program, refer to its manual for receiving data from the SMCC. And refer to i Parameter Instructions Section and E-Point Description Section or detailed explanations.

If you are using parallel communications, first establish that SMCC is okay by using RS232, then proceed to use the parallel mode. Use the SW switch to change communications mode.

8) Verify software version used. It may be sometimes necessary to establish the operating software version the SMCC is operating with. The version number is recorded in the PROM of the SMCC always at the same location and may be read as follows:

For Parabolic:

Type r 2034, 5 <CR>. This is the SMCC's "Read Memory" Command and causes five bytes of data stored and starting at memory location 2034 to be sent back. Data displayed is in hex ASCII format. For example if the following data is sent: 32, 2e, 35, 39, 50 it may be interpreted as 2,.,5,9,P, which means the software version is VERSION 2.59 Parabolic.

For Standard:

Proceed in the same manner. Type r 202e, 5 <CR>. If the response is: 32, 2e, 33, 39, 20, it is interpreted as 2, ., 3, 9, space which means the software version is VERSION 2.39 Standard.

SMCC OPERATION VERIFICATION

In order for the user to quickly establish SMCC's functionality, a series of simple procedures are described below. The procedures are based on a "Bench Test" requiring only the SMCC, a +5 volt power supply, an RS232 cable, and a host computer or terminal. The test philosophy is based on commanding the SMCC to

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

move and since no position encoder or resolver feedback is present, to cause the SMCC to generate a following error thus indicating that it receives and executes commands correctly and detects a following error when position feedback is not present.

With SMCC connected as discussed above, first be sure to either cycle power or reset the SMCC by grounding J1B - 19 or J7 - 15 (RESET) momentarily and to send an A0 <CR> Address Command to SMCC thus enabling it to send data back when requested.

Send a "p" (70H) ASCII character (position request) followed by a <CR> (0DH) carriage return to the SMCC. The terminal should display (0 0). Please note that if i15 is set to one (X axis only, Y axis disabled), then position returned would be one (0) for the X axis only.

Send a "?" (3FH) ASCII character (status request) followed by a <CR> (0DH) carriage return. The terminal should display (00008888) for non-parabolic, Standard SMCC and should display (0000a0a000000000) for a Parabolic SMCC.

Proceed to disable SMCC's travel limit protection by installing jumpers as follows:

J1-A4 = Y + Limit (YPLM) AND J1-A5 = Y-Limit (YMLM) tied to J1-A7 = GND and J1-C4 = X + Limit (XPLM) and J1-C5 = X-Limit (XMLM) tied to J1-C7 = GND. This will enable both the X and Y axes. Please note that <u>both</u> X <u>and</u> Y limits must be grounded, even if you are using one axis only.

Now send a "J" (4AH) ASCII character followed by a <CR> (0DH) carriage return to the SMCC. This is the "Jog X Axis in Positive Direction" Command.

Within 1/2 second the local display should read " 2." indicating the SMCC tried to jog in the (+) direction and the following error stopped it since there is no encoder feedback.

Reset the SMCC by cycling power or grounding the INIT line (reset) at J1B-19 or J7-15. Then try "j" (6AH) ASCII character. Results should be the same as for the (upper-case J). Reset the SMCC again.

If the Y axis is enabled (i15 = 0), send "JY" (4A 59H) ASCII characters followed by a <CR>(0DH) carriage return to the SMCC. This is the command for jogging the Y axis in the positive direction.

Within 1/2 second the local display should read "2." indicating the SMCC tried to jog in the (+) direction and the following error stopped it.

Reset the SMCC by cycling power or resetting, then try "jY" (6A 59H) ASCII characters. Results should be the same as for the upper-case JY.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

This error "2." is caused by the lack of motor encoder feedback. If it does NOT occur, it means that the SMCC receives your command to jog but is unable to make its move due to some requirement that is not being met or is disregarding the following error. Check the following items in order to ensure that SMCC is enabled to make its move and able to detect following error:

1) Check i00 and be sure that it is not set to "0" since this disables following error detection and SMCC will continue to "move" without stopping and will disregard any following error. The value of i00 should be set at a value of approximately 2000.

2) Check i14 and be sure it is set to "0". If it is set to "1", the SMCC requires an analog 0 to +5 volt input in order to set its feedrate. The input is at J1-C1 (ANA2) and may be set to zero or disconnected. To be sure SMCC can move, set i14 to "0". This makes it use the % Feedrate Command for feedrate control instead of the analog signal input.

3) Check i06, the feedrate value, and be sure it is not set to "0". If it is, set it to a small value of 5000.

4) Check i31[i51], the jog feedrate, and be sure it is not set to "0". If it is, set it to a value of 64 to 128, 25/50% of maximum.

5) Check i16 and be sure it is set to "0" or "1". If it is set to "2" or "3", SMCC requires an external pulse/direction or quadrature clock at its handwheel input to generate its feedrate, and the "%" Feedrate Command is disabled. To be sure SMCC can make its commanded move, set i16 to "0".

6) Check and be sure that the "%" Feedrate Command is not set to "0". The default value is 65498. If % is set to "0", the SMCC will not make any moves.

7) Be sure the X and Y "+" and "-" travel limits (J1-C4,J1-C5,J1-A4, and J1-A5) are ALL grounded.

8) Check again and be sure your terminal is sending correct data to SMCC. If the Jog Commands are not being received, SMCC will not make any moves. At this point you may wish to try direct Jog Commands by using the control panel connector pins (JPAN) to issue a hardware Jog Command to the SMCC. Successively, one at a time, jumper:

JPAN	pin 3	Y-JOG (jY)
JPAN	pin 4	X-JOG (j)
JPAN	pin 5	Y+JOG (JY)
JPAN	pin 6	X+JOG (J)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

to JPAN pin 26 which is ground and activates the jog function. If SMCC then does not detect a following error "2.", then check your RS232 setup again. However, if all of the above are set and executed correctly and you are still not able to cause SMCC to make a following error "2." when you issue the Jog Command, you may have a defective SMCC. Call the factory for assistance.

CONNECTING STANDALONE SMCC TO SYSTEM (PRE-INSTALLATION CHECKS)

Before connecting the SMCC to the servo amplifier/motor it is important to set up the amplifier/motor by itself. A tachometer type system is assumed here (a tachless system is described later). Follow the manufacturer's recommendations and "tune" the amplifier for optimum performance so it controls the motor in a "critically damped" manner. Check speed and direction of motor travel (use a small power supply or battery for the amplifier signal input). The SMCC generates a positive signal for motor travel in the positive direction so be certain that with a positive input signal, the amplifier turns the motor in a rotation that will cause the machine to move in the "+" direction. To change motor direction, swap both the tach +/- leads and the motor +/- leads.

Set up motor speed so that with approximately 8.0 to 9.0 volts of error signal to the amplifier, the motor's RPM is the maximum you expect in your system. This leaves some "head" room for the SMCC to operate. This test is valid only for amplifiers which are being used with tachometer feedback from the motor. Please refer to the end of this section for discussion of differences between tachometer (velocity loop) and non-tachometer (torque loop) operation.

It is recommended that the host, SMCC, and amp/motor be connected to each other for testing and set up prior to connecting the motor to the machine. If you are using linear encoder scales, this is not possible and the tests must be done on the machine.

After connecting the SMCC to the tuned and stabilized amplifier, check to see that if you issue a "J" (4AH) Jog in Positive Direction Command, the position the SMCC is displaying or can send back to you when you send it a "p" (70H) Position Request Command, is also increasing in the positive direction. If it is not doing so and is increasing in the negative direction, it means that you have connected the amplifier backwards (wrong direction of motion).

To correct this, you must change three items:

1) Reverse the motor's +/- leads
2) Reverse the tachometer's +/- leads (only in tachometer systems)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3) Swap the encoder's A and A/ leads for B and B/ and vice versa or, to avoid rewiring the encoder, use the i39[i59] parameter. Change it from a "3" to a "7" or from a "7" to a "3" (see i39[i59] description).

This procedure should correct the direction discrepancy between the SMCC commands and the direction of motor travel.

It is also possible that the motor will simply "run away" at high speed. If this occurs, simply perform step 3 above (swap encoder A and B channels in hardware or software), but only if you have previously made sure the amplifier/motor are properly connected. You should then recheck your direction polarity for correctness again. Refer to the following documents in order to insure proper connection to amplifier/motor:

1) 602103 - Two Axes Servo Motor Positioning System

2) Drawing - SMCC Servo Amplifier Signal, Power & Motor, Tach & Encoder Input/Output Connections

3) Schematic - J1 - (JMACH) for Encoders

4) Schematic - J1/J10 for Amplifier

If for some reason the SMCC and amplifier/motor are not functioning correctly together, check the following:

1) A Jog or Move Command is issued and the motor does <u>not</u> move:

   A. The SMCC displays a following error "2." after command is issued.

      1) Check that SMCC is correctly connected to amplifier and that amplifier is not disabled.

      2) Check that motor is correctly connected to amplifier.

      3) Check that i20[i40] (proportional gain) is not set to zero.

      4) Check that i32[i52] (PWM limit) which controls the magnitude of SMCC's output is not set to zero.

      5) Check that the +/- 12 to 15 volt power supplies needed by SMCC's optically isolated error output (command) signal are present at J1-A29 or J1-C29 for +15V (P15A) and J1-A25 or J1-C25 for -15V (M15A) and that the return (AGND) is applied to JI-A30 or J1-C30.

   B. The SMCC displays ".0" (no error) and is in position: This means that SMCC is not accepting or is not executing the jog request or the command. Please go over the same check list used to check the SMCC alone and

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

insure that communications between the host and SMCC are functioning correctly.

2) A Jog or Move Command is issued and the motor is moving but not at the correct speed. Following error may or may not occur.

A. Check that i06 (machine feedrate) and i31[i51] (jog feedrate) are properly selected. If the encoder has 2000 counts per revolution and i06 is set to 4000 and i31 is set to 128, the motor should turn at one revolution per second.

B. Check that i32[i52] (PWM limit) is set high enough to attain desired speed. 255 is maximum and will output +/- 10V. A low value prevents SMCC from driving the amplifier fast enough.

C. Check that i33[i53] (PWM protective limit) is set high enough (set to 255 to disable its action).

D. Check i20[i40] (proportional gain) and be sure it is set high enough.

E. Make sure that the (P15A) +15 volt and (M15A) -15 volt power supplies for the SMCC's optically coupled command output are at least 12 volts. The SMCC will operate with less voltage but will not be able to attain maximum speed.

F. Be sure that "E1" for X and "E9" for Y jumpers are installed. Without these jumpers, SMCC's output is only +/- 5 volts with respect to AGND. If the amplifier has single-ended input, it will not function properly.

If you are using a differential input on the amplifier, then E1 and E9 should be left out, since most amplifiers only require a +/- 5 volts differential input.

G. Check that (Phase 1/) X axis output and (Phase 3/) Y axis output are not shorted to ground. If not used, these should be left floating. Grounding either one will reduce the output levels at (Phase 1) and (Phase 3) which actually drive the amplifier.

You may also use the "O" (4FH) (upper-case O) Command to set the proper speed on the amplifier/motor. The "O" Command places the SMCC in an open loop servo mode and outputs a DC Velocity Command signal to the amplifier. For example, for the X axis O255 will output +10.0 volts to the amplifier, O25 will output approximately 1.0 volt, O-64 will output -2.5 volts, and so on. For the Y axis use OY255, OY-32, and so on. Using the "O" Command, send 8.00 volts, approximately O200, and using the amplifier's "signal" potentiometer adjust the motor RPM to be the

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

maximum RPM you will require in your system. To exit the "O" open loop mode, send a "q" or "Q" Command to SMCC.

==========

Exercise **CAUTION** and always start with small "O" values and increase gradually.

==========

If possible, now mount the motor on the machine and verify that the amplifier/motor combination you are using provides you with sufficient power to achieve your desired accel and decel times. Looking at the tach signal with an oscilloscope will give you accurate readings, or, if you are using Parabolic SMCC, you may use the "data gathering" feature to evaluate your system and optimize the servo parameters.

Also, SMCC V3.6 or higher software diskette for IBM PC (ACC-9) provides an interactive tuning procedure you may use to optimally tune and adjust the motor/amplifier.

NOTE: If you are not achieving desired performance with the amp/motor, connecting the SMCC will not improve your system performance. If anything, it will slightly degrade it. Total accel and decel times are measured as follows:

$$Ts = (Tm^2 + Tc^2)^{1/2}$$

Where: Ts - is the system response,

Tm - is the machine response with motor and amplifier alone,

Tc - is the SMCC's accel or decel time and is a function of i08 or "t"

For example, if Tm is 100 msec. and Tc is also 100 msec., Ts = 1.4 x 100 = 140 msec.

You are now ready to connect the SMCC and the amplifier/motor to the machine.

CONNECTING HOST, SMCC, AMPLIFIER/MOTOR, MACHINE

==========**CAUTION**==========

Exercise caution when turning on your machine with SMCCs and motors/amplifiers connected to your machine, as wiring errors can cause dangerous "runaway" conditions.

==========

1) Make sure that the encoder wiring is as it should be and that the limit switches are properly connected. If a single

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

axis is used, it must be addressed as A0. If multiple SMCCs are used, the communication daisy chain must be installed and at least one SMCC must be addressed A0. The A0 card must be the only one with a jumper installed at E2. The rest of the SMCCs should have a jumper installed at E3.

2) If the amplifier has a disable input, disable the amplifier and turn the power on. No motion should occur. Motor should be free to move. Move motor so at least 2000 encoder lines are generated by the motion. The SMCC's local display should read "2." indicating a following error has occurred. Reset SMCC and try motion in opposite direction. Same thing should occur. This indicates that the encoder is properly connected and the SMCC is reading it. The following error occurs because the encoder is moving without being told to do so.

   If the local display does not read "2.", the encoder connections are not as they should be. Do not go to step (3) until this step is completed.

   This test verifies that the encoder is properly connected and that the SMCC is detecting following error correctly, since a motion of the motor (or encoder counts received) without a command to do so must result in a following error.

   NOTE: For test purposes it may be desirable to rotate the encoder itself rather than the motor to generate the following error. This test applies to either incremental encoders or absolute position synchro or encoder feedback.

3) Enable amplifier and apply power, being careful to quickly turn it off if rapid motion occurs. If the encoder direction is wrong, the motor will attempt to move quickly (runaway) and the SMCC will generate a following error: status display will read "2.". To correct this, change encoder direction by either swapping the encoder's "A" and "B" and "A/" and "B/" channels or by using i39[i59] parameters on SMCC. See i39[i59] description.

   The system should be stable and motionless on power turn-on. The motor shaft should resist being turned and should hold position.

4) Now try jogging either by using control panel inputs or host commands. The motor will move at a velocity of 1600 encoder counts/second. To go faster or slower use i06 or "%" Commands. It is possible that as you try to jog the motor, the SMCC will generate a following error "2.". This may be because i20[i40] (proportional gain) is set too low and SMCC isn't "pushing" hard enough.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Send an "A0" Command then a "p" (70H) (send position) Command. The SMCC will report its position. On power turn-on or reset, position should be zero. If it isn't, use amplifier balance pot to adjust position to zero or activate i23[i43] (integral gain or auto bias generator) with approximately ten. To remove the position error be sure i09 (integrator mode control) is set to a one which activates the integrator only when the SMCC is at rest. Adjust i20[i40] (proportional gain) up or down from 50 to suit your needs. With proper gain, the motor will get into position without overshoot or dragging. Typical gains are between 50 and 1,000 for a system with tachometer. Then adjust i08 for desired accel and decel time.

NOTES: 1) If you have a non-Parabolic SMCC and need more gain, use i30[i50] to multiply the i20[i40] values by 256. (See i30 Description)

2) Please refer to "Tach" and "Tachless" tuning procedure for Parabolic SMCC as it provides more insight into tuning procedures.

Write a small program like:

```
z50         ;Clear buffer and
            ;assign 50 steps in
            ;buffer for program.
F1024       ;Set feedrate to
            ;100% of i06 value.
            ;Note that if feed-
            ;rate Command is not
            ;issued SMCC will not
            ;move.
X10000      ;Go to position 10000
X0          ;Go to zero
END
```

and step through it using the "S" (53H) Step Command or control panel "Step" switch. Change machine speed by using i06 or "%" Command or changing the "F" feedrate number.

In order to write a new program the buffer must be cleared by using the "z" (7AH) Command. To try the "home" Command you must first set i04 to some number like 25 (10%) and check that by using i26[i46] you have chosen the correct polarities for your "Home Flag Switch" and encoder "C" channel then issue the "h" (68H) Command or activate the control panel "Home" switch. The SMCC will start homing until the machine home flag switch is activated; it then looks for the encoder "C" channel. When the "C" channel is found, the SMCC will reverse forward by 50 encoder counts (set by i25[i45]) and call that position zero or "Home".

YOU ARE NOW READY TO USE YOUR SYSTEM!!

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Some of the finer points of setup, however, require further discussion since applications and requirements vary greatly. The following descriptions are general in nature and the user should apply his own techniques to optimize his application.

Differences between setting up a system with tachometer velocity feedback (closed loop servo amplifier) and one without tachometer (brushless motor or direct driven DC motor) are as follows.

SYSTEMS WITH TACHOMETER USING PARABOLIC OPTION

==========NOTE==========
The Parabolic SMCC allows you to "gather" motion profile data and display it on the screen. Refer to SMCC V3.X IBM PC diskette for details of operation. The diskette also contains an SMCC Tuning Procedure which can be used to set up and optimize your system.
==========

Generally, these types of systems provide a tighter servo loop at low velocities and at rest and are easier to optimize on a standalone basis since the motor is able to move in a stable fashion due to the closed velocity loop. They do, however, exhibit relatively large following errors due to the tachometer feedback.

The main difference between a Standard SMCC and a Parabolic version for servo tuning is the fact that the Parabolic version has two additional terms in the servo equations which are:

1) **i22[i42] Velocity Feedforward** term (Kvff), which is used to counteract the effect of the i21[i41] feedback term and to reduce or eliminate following error at various velocities.

2) **i30[i50] Acceleration Feedforward** term (Kaff), which is used to reduce or eliminate the following error peaks which occur during accel and decel.

Also, the Parabolic version's i21[i41] velocity feedback term (Kd) operates differently that the equivalent term in the Standard SMCC. In the Parabolic version, the higher the i21[i41] number (range is 0 to 255) the larger the feedback term and the more stable the system. In the Standard version, due to a different servo filter implementation, the converse is true. A smaller number provides greater feedback. (See i21 for further explanation.)

First, set i21[i41] (feedback gain), i22[i42] (velocity feedforward), and i30[i50] (acceleration feedforward) to "0" and adjust i20[i40] (proportional gain) so that moves are made as smoothly as possible and without overshoot. Adequate gain values

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

will range on the lower side and will tend to be in the range of 5 to 1600. The system should be as "tight" as possible without becoming unstable, breaking into oscillations or overshooting position or velocity. Note that the amplifier itself must first be tuned correctly before attempting a closed-loop setup.

Then, use i08 to provide accel and decel ramps as required to reduce overshoot as programmed velocities become higher until you reach your system's desired velocity. Set i09 (integrator control) to one and activate i23[i43] (integrator gain) to approximately ten. These steps activate the error integrator at rest only and will provide excellent position stability and repeatability. The integrator will automatically monitor any in-position errors caused by friction, eccentric loads, temperature drifts, etc., and will compensate for these disturbances so that position error becomes zero. If it is necessary to reduce run-time following error, adjust i06 for desired velocity nd issue a Jog Command. Use i22[i42] (velocity feedforward) by slowly increasing it and examining the following error (F.E.) with the "f" (66H) Command. Note that the less the F.E., the snappier the move becomes and there may be overshoot tendency when getting to position.

If the application requires a "zero" F.E. such as in accurate velocity-locked web controls, i23[i43] (integrator) may be activated during the move by setting i09 (integrator control) to "0" instead of "1". It is best to use this method only in systems that are velocity based (will run forever) without many start and stops since it is sometimes difficult to compensate for the integrator's effects during accel and decel. i30[i50] (acceleration feedforward) may be used to compensate the integrator's overshoot tendencies.

NOTE: It is possible to increase i22[i42] (velocity feedforward) to the point where the F.E. actually goes through zero and changes sign, meaning that the system is now being pushed so hard that the motor is actually leading the desired position. This, in general, should be avoided since it may cause undesirable overtravel when arriving at destination.

Assuming that we have a small following error while the system is moving steadily, a more careful examination will reveal that during acceleration and deceleration, the F.E. exhibits peak values; this is normal since the system is changing velocity and F.E. is trying to keep up. To reduce or eliminate the F.E. peaks during accel/decel, use the i30[i50] (acceleration feedforward) term since i30[i50] compensates for rate of change of velocity. This may be used to even eliminate in-position arrival overshoot.

In this setup procedure, note that i21[i41] (differential gain) has been left at "0". This is because in systems with a tachometer the amplifier is provided feedback gain by the tachometer and there should be no need to use i21[i31] to stabilize the system. If there is a need (and this is not

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

recommended since it indicates a problem with the amplifier, motor, mechanical system) to use some amount of feedback (differential) gain, the values should be small, 1 to 5 or 10, and should be compensated by a corresponding increase of the 122[i42] (velocity feedforward) term in order to keep the F.E. the same.

To summarize the procedure for a tachometer system, the following applies:

1) **i20[i40] (Proportional Gain)** - Use as large a value as possible without inducing instability or oscillations. (Typical values - 5 to 1000.) The higher the value the snappier the system and the smaller the F.E.

2) **i21[i41] (Differential Gain)** - Set to zero value. You should not have to use this since the amplifier is using the tachometer as negative feedback. If you do use i21[i41], then increase i22[i42] correspondingly to cancel out the increased F.E.

3) **i22[i42] (Velocity Feedforward)** - Use this term to reduce or eliminate following error during a constant velocity move. (Typical values - 1 to 50)

4) **i30[i50] (Acceleration Feedforward)** - Use this term to reduce or eliminate following error during the accel/decel part of the move only. (Typical values - 1 to 50)

5) **i23[i43] (Integral Gain)** - May be used two different ways:

   A. Active only when in position by setting i09 to 1.
   B. Active both during a move and at rest, by setting i09 to 0.

   For normal operation it is best to use the integrator at rest only, where it monitors position and ensures zero following error by generating a bias in the SMCC's output signal for counteracting the effects of temperature, time, friction, off-balance loads, etc. In velocity control applications, the integrator may be enabled at all times and will strive to automatically maintain zero following error, regardless of changing load and inertia conditions. Unlike the integrator, the i22[i42] (velocity feedforward) term provides a constant correction value for following error and will not correct a changing condition. The disadvantage of having the integrator turned on at all times is that if the system will be start/stopping or changing speeds (accel/decel), then the integrator may generate excessive following error during the accel/decel since it will not be able to keep up with the rapidly changing velocity and may cause undesirable overshoot.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

6) **i08 (Accel/Decel)** - This parameter controls both the acceleration and deceleration time and must be set to a value the amplifier/motor system can handle. Too small a value causes the amplifier to be unable to start or stop the motor in the specified time (amplifier current limit may occur) and this results in overshoots which cannot be corrected by any SMCC servo parameters.

   It is important, therefore, to "size" the system properly for amplifier power level and motor torque and speed to properly handle the friction and inertial loads of the system. (Typical values of i08 are 50 to 1000.)

SYSTEMS WITHOUT A TACHOMETER USING PARABOLIC OPTION

==========NOTE==========
The SMCC V3.X (Version 3.6 or greater) IBM PC terminal software provides special "Servo Tuning" aids which guide the process of tuning the servo system. By providing position, velocity, acceleration, and following error plots, augmented by correlation and ratio calculations, the SMCC V3.X provides a means of optimally tuning the servo quickly and with great precision.
==========

When driving brushless DC motors, AC induction motors, or brush motors with current amplifiers (also known as "transconductance amplifiers"), the SMCC is equipped to provide the necessary compensation factors to stabilize the position and velocity control loops.

Most commercially available DC closed loop servo amplifiers have a special low gain (current mode) which is strap selectable and enables them to be used in tachless applications. Amplifier gain should be set such that an input voltage of 0 to 10 volts will provide from zero to the maximum current the amplifier is capable of.

In general, the main advantage of a tachless operation is that the system following error will be very small and the system is much more responsive since the SMCC has direct control of the motor current. The disadvantage is that the motor at rest (in-position) or at low velocities will exhibit more softness because the encoder does not provide good velocity feedback when it is moving slowly or is at rest. This problem can be improved by using the 1/T feature of the SMCC and by using an encoder with high count resolution. The greater the encoder counts per revolution, the better the system will operate. The limitation in increasing the encoder counts/rev. is that at high RPMs the encoder's output frequency capability may be exceeded.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The main differences between setting up a tachless motor using a current loop amplifier are:

1) It is more difficult to set up the amplifier/motor by themselves since the motor will "run away" even with a small input voltage, since there is no stabilizing feedback.

2) The output from SMCC to the amplifier now represents "torque" or "current" demand and is not a "velocity" Command. This is an advantage, since now the SMCC can be commanded to control torque as well as position, velocity, and acceleration.

3) The SMCC now has to provide the necessary feedback (differential) compensation by using the encoder or resolver on the motor and its internal digital filter.

4) The value of i20[i40] (proportional gain) that must now be used is much larger than with a tachometer system. Increase the gain gradually until desired stiffness is obtained. Also observe that the following errors are now much smaller.

You may refer to the tuning procedure described previously for the tachometer type system since all statements apply equally to both tach and tachless systems, with the exception that the i21[i41] (differential gain) term must be non-zero and set to a value of approximately 5 to 100 in order to provide the necessary stabilization for the system. Use the smallest possible value to achieve stable operation. A good test is to (WITH GREAT CAUTION!!!) manually disturb the motor shaft and observe that it returns to rest "cleanly" and without damped oscillations. Too small a value will cause sustained oscillations upon disturbance. Also, short start/stop moves may be used to observe how well the motor comes to rest. The greater i21[i51] the more stable the system becomes but also, the greater the following error.

i22[i42] (velocity feedforward) should be used, as in tachometer systems, to reduce following error while running at constant velocity and to compensate for the following error caused by i21[i41]. In general, the value of i22[i42] should be very nearly equal to the i21[i41] value. All other procedures for system tuning are identical to those used with the tachometer type setup.

BRUSHLESS DC MOTOR SETUP (COMMUTATED BY SMCC)

When setting up a brushless servo, the first consideration is to ascertain whether the SMCC must provide the phase commutation. IF THE AMPLIFIER REQUIRES ONLY ONE +/- 10 VOLT SIGNAL TO DRIVE THE MOTOR AND IT PROVIDES THE PHASE COMMUTATION REQUIRED BY THE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

MOTOR, THEN THE SMCC DOES NOT CONSIDER IT TO BE A BRUSHLESS SERVO. The servo is considered to be brushless only if the SMCC is required to provide two out-of-phase signals and commutate the motor directly. The amplifier would need to provide all other phase currents by using "balance loops". If the amplifier is not equipped with "balance loops", then the SMCC ACC-8V can be used to provide the additional phase. All other servos should be set up the same as DC brushed motors.

The brushless motor parameter i13 must be set to one and the induction motor parameters must be set to zero before the SMCC will try to phase a brushless motor. The induction motor parameters for parabolic option are i42 and i45 and for non-parabolic the only induction motor parameter that must be set to zero is i51. When first setting up the brushless motor it is best to set i41, i45, and i54 to zero.

The number of counts per electrical revolution must be entered into i40. This can be calculated by dividing the number of counts per revolution by the number of cycles the motor makes per revolution. The number of cycles the motor makes per revolution is one half the number of poles (the number of pole pairs) the motor has. For example, an eight-pole motor is also a four-cycle motor. If the encoder produces 2000 counts per revolution, then the number of counts per electrical revolution will be 2000/4 or 500 counts. This number must be entered into i40. Make sure that the number of counts per revolution includes any multiplication factor on the encoder.

The next parameter that must be entered is the phase type of the motor. If the motor is a 3-phase motor then i44 must be set to one. If it is a 4-phase motor then i44 must be set to zero. This parameter is used to set up the phase shift between the two drive signals. For a 3-phase motor the phase shift will be 120 degrees and for a 4-phase motor the phase shift will be 90 degrees.

The next two parameters that must be entered determine the current and time allowed to phase the motors. The time is entered into i52 in multiples of about 44 msec. If 40 is entered into i52, then each phasing position is allowed about 1.75 seconds to settle. The current output to the motor is entered into i53. This current should be large enough to overcome any friction in the system but not so large that the motor never settles down in the time allowed. The SMCC first drives a current through one of the phases for the time entered into i52 and then moves to the next phase for the same length of time. At the end of this phasing time the SMCC assumes that the motor is at a position with minimum torque. The SMCC initializes its phase position to its present position and produces torque by adding or subtracting 90 electrical degrees to its present phase.

Once the motor has gone through its phasing cycle, the direction of the encoder must be verified. The SMCC should be commanded to output a constant torque using the "O" Command. The number sent

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

by the "O" Command should be large enough to overcome any friction in the system. It should result in an increasing velocity until the friction equals the commanded torque. If the motor locks up and does not start rotating, then most likely the encoder is counting in the wrong direction. There are two ways to fix this condition, depending on which direction is desired to be positive. If the encoder is counting in the right direction, then two of the phases on the motor must be changed. If the encoder is counting in the wrong direction, then all that is required is to change the i39 parameter for the opposite direction. Add or subtract four from its present value and enter this new value into i39. Once these changes are made the motor should be re-initialized by sending a "$" to the SMCC or cycling power. If power is cycled be sure to save the new settings before removing power.

Once the motor turns freely with an "O" Command, the rest of the set up is the same as a brushed DC motor. Some additional tuning can be made to the brushless motor while it is moving freely with an "O" Command. The phase advance parameter can be used to increase the torque at high speeds. If i41 is changed while in the "O" Command, the motor speed should change. Use this parameter to maximize the speed for a constant setting of the "O" Command.

The i15 parameter can be used to select the shape of the drive signal to the motor. With i15 set to one the shape will be trapezoidal and also will minimize the duty cycle required to phase the motor. If a sinusoidal drive is desired, then i15 should be set to zero and the address of the sine table should be entered into i43. The address for a 3-phase sine wave is 130 and for a 4-phase sine wave is 131.

AC INDUCTION MOTOR SETUP

Much of the setup for an AC induction motor is the same as for the brushless DC motor. The number of counts per electrical revolution is the same and must be entered into i40 and the shape for the waveform is selected by i15 and i43. There are two additional parameters that must be entered for an induction Bmotor. The first is the magnetization current and is entered into i52 for non-parabolic and i42 for Parabolic systems. This number should be set to about 25% of the rated current of the motor. By increasing the magnetization current the motor can produce more torque (within limits). However, the motor will produce more heat while standing still.

The magnetization current also effects the KE of the motor. Thus an AC induction motor designed to operate at a base speed of 1800 RPM (4-pole motor on 60 Hz AC line) may be run at many times base speed by adequately controlling the magnetization current. SMCC's PLC capability can be used to change magnetization and any

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

other servo parameter as a function of velocity, position, or following error as circumstances dictate.

In general, an AC induction motor can be operated at constant torque from zero up to its base speed and at constant horsepower above its base speed.

The second parameter that must be entered is the slip gain. For Parabolic a reasonable starting point is about 200 in i45 and for the non-parabolic the number should be about 10 in i51. These numbers are dependent on the type of motor being used and the number of counts on the position feedback device. The best way to start to optimize the slip gain is in the "O" Command mode. Try to adjust the slip gain to produce maximum linear acceleration up to the operating speed of the motor.

Once these parameters have been entered, no phasing sequence is required since there is no fixed phasing for an induction motor. If the motor turns slowly when an "O" Command is sent to the SMCC, then the encoder is most likely counting backwards. The way to correct this is the same as for the brushless motor. Either change two phases to the motor or use i39 to change the direction of the encoder counts. Once the motor turns freely and at high speed using the "O" Command, the setup is the same as a brushed DC motor.

USING SMCC WITH A RESOLVER AND PREVENTING START-UP PHASING JUMP ON BRUSHLESS MOTORS (PARABOLIC SMCC) (COMMUTATED BY SMCC)

The SMCC is designed to function with resolvers or encoders and to perform automatic phase-finding. The phase finding procedure requires that the motor shaft move a maximum of one half electrical motor revolution (90 degrees for a 4-pole motor) in order to precisely locate correct motor phasing. Even though the phase finding time and torque level are both definable, it is sometimes undesirable to have a "phase finding" cycle on a machine especially since full torque cannot be guaranteed during phasing and motion occurs bi-directionally.

An improvement to this is to use hall effect devices as absolute position sensors to locate motor windings which then permit the SMCC to find a phase by making a smaller movement at full torque and in one direction to find a signal edge. Even this small movement may be eliminated by using a resolver feedback, which provides enough resolution to prevent any motion on power turn-on or reset.

The procedure to follow for eliminating the initial phasing search when using a resolver is:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1) Be sure that i64 is set correctly:

   i64 is ffb8 or 65464 for S-type SMCC (also known as Micro SMCC)
   i64 is fd50 or 64848 for Parabolic SMCC (ACC-14 has E2 on, E1 off)

   This parameter assigns the memory address at which the resolver is located.

2) Set i39 to a value of 3 or 7, which sets up the resolver parallel-data-to-quadrature converter to provide a x 4 multiplication either in CW or CCW direction. This causes the quadrature counts to be equal to the resolver counts. If the direction selection is incorrect, the motor will not phase and either the 3 or 7 must be changed or any two of the motor leads must be swapped. Having chosen the correct direction, now set i76 to 1, if i39 is 3, or i76 to 0, if i39 is 7. i76 is used to complement the resolver data and must be set correctly.

3) First set i72 = 0, i54 = 0, and reset the SMCC with "$" or equivalent. SMCC will now perform its normal power-on phasing procedure.

4) When phasing is completed, do a "HOME" with i25 (home offset parameter) set to 0 (zero offset). This causes the resolver to be at zero position.

5) Now read the value of memory location fe26 by issuing an "r" Command for two bytes: rfe26,2.

   The SMCC will send back two hexadecimal (LSByte first) bytes as the answer which represent the present phasing position offset. Note that it is important to have done a home with zero offset before this procedure; otherwise the value of the resolver position has to be subtracted from the value of fe26.

6) Next, convert the value of fe26 from hexadecimal to decimal and enter this value into i54, phase offset.

7) Finally, change the value of i72 from "0" to "3" which tells the SMCC to skip doing a phasing search and to use the value stored in i54 as the phase offset.

   The user may adjust the value of i54 in order to optimize the phase offset. It is also possible to change the phasing angle as a function of velocity by using i41 to generate a phase shift proportional to velocity. Please refer to i Parameter Section of SMCC for explanation.

   Please be aware that the motor lead phasing and the resolver phasing must be set correctly; otherwise the system will neither phase nor function correctly.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

8) It should also be noted that the SMCC with Parabolic option is capable of handling encoder or resolver counts per revolution which are not divisible by three. This capability is controlled by i50. Under normal circumstances with i50 set to zero (the default value), i40 is used to define the number of counts per electrical revolution. This assumes that the number of counts per actual shaft revolution is evenly divisible by the number of electrical cycles and if it is, the system can be used normally. In the particular case of 6-pole motors (three electrical cycles), a division by three of the i40 value is necessary. To use the SMCC's divide-by-three feature proceed as follows:

A. Set i40 to the value of counts per <u>three</u> electrical cycles. In the case of a 6-pole motor, i40 is set to the <u>actual count per motor shaft revolution</u> and <u>NOT</u> to the normally used value of counts per one electrical revolution. A 12-pole motor, for example, would require that i40 be set to 1/2 the value of counts per shaft revolution since a 1/2 revolution of the shaft represents three electrical cycles for the 12-pole motor.

B. Then set i50 to one. This action tells the SMCC to use the i40 number for a 6-pole motor (or 12-pole motor) and perform the divide-by-three function.

You may now proceed to operate SMCC normally. Please note that setting i50 = 1 does increase the servo duty cycle somewhat (approximately 5%) therefore, when doing intensive communication and blocks/second execution rates, do give the SMCC a bit more time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PARAMETERS FOR PARALLEL ENCODERS/RESOLVERS

(Parabolic Version Only)

			Value	First ACC14 (E2 on, E1 off)	Second ACC14 (E2 off, E1 on)
i64[65]	x[y]	Parallel Encoder Base Address	64832 [64835]	J1[J3]	
i64[65]	x[y]	Single Speed Resolver Base Addres	64848 [64864]	J7[J6]	
X Axis Only Using Dual Resolvers:					
i64	x	Dual Speed Resolver Base Address	64848	J7	
i68	x	Second Resolver Base Address	64864	J6	
Y Axis Only Using Dual Resolvers:					
i65	y	Dual Speed Resolver Base Address	64848	J7	
i69	y	Second Resolver Base Address	64864	J6	
Both Axes Using Dual Resolvers:					
i64[65]	x[y]	Dual Speed Resolver Base Address	64848 [64912]	J7	[J7]
i68[69]	x[y]	Second Resolver Base Addres	64864 [64928]	J6	[J6]
i66[67]	x[y]	Second Resolver Speed	0 to 255		
i70[71]	x[y]	Encoder/Resolver Bias	+/- 134217727		
i72[73]	x[y]	Encoder/First Resolver Size	# of bits (See Note 1)		
i74[75]	x[y]	Second Resolver Size	# of bits		
i76[77]	x[y]	Logic Level Control Bits	0 = Strobe 'LOW' Data 'LOW' 1 = Strobe 'LOW' Data 'HIGH' 2 = Strobe 'HIGH' Data 'LOW' 3 = Strobe 'HIGH' DATA 'HIGH'		

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

NOTE 1: If encoder size is less than 8 bits then a resolver is enabled. The size of the resolver is i72(73) + 9. A resolver between 10 and 16 bits can be selected.

NOTE 2: An ACC-14 configured to the lowest possible address (E2 on, E1 on) will not work properly with absolute encoders or resolvers.

The position is initialized to the absolute zero extended encoder reading minus the encoder bias at turn-on. Incremental changes in the encoder are then used for position and velocity information.

PROGRAMMING SMCC, GETTING STARTED

The assumption is made here that for the X and/or Y axis the SMCC is up and running with an amplifier and motor connected and that you have verified the system's stability and have gone through the start-up procedure. The following procedure will require that the motor shaft be free to rotate for extended periods of time. Therefore, if at all possible, attempt to disconnect the motor shaft from the machine so as to be able to follow this procedure without constraints. If this is not possible or too difficult, proceed with great caution in order to avoid injury to yourself or damage to your machine. Verify that the travel limit switches in the system are fully functional before proceeding.

You may use a "dumb terminal" or (highly recommended) an IBM PC with SMCC V3.X Terminal Program (ACC-9) to program the SMCC. You may use serial RS232 or parallel communications. It is also recommended that, if an Accessory 16 (ACC-16) control panel is available, you make use of it since it provides great convenience in operating and monitoring the performance of SMCC. Turn power on and verify that your system is stable and holding position. Also, verify that position is zero; to do that proceed as follows:

```
Type  "A0 <CR>"        ;You must address the SMCC
                       ;at least once after power-on or
                       ;reset  before you can ask it
                       ;to send any information and data back.
                       ;If there are more than
                       ;one SMCC in your system, each
                       ;must be individually addressed before
                       ;information can be sent to or returned from
                       ;the particular SMCC.
                       ;
Then type "p <CR>"     ;This is a position request from
                       ;SMCC.  It will respond with one zero
                       ;for one axis or two zeroes for two axes.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;
Type  "f <CR>"          ;This is a following error request
                        ;and will also send one or two
                        ;zeroes back.
                        ;Please note that if position shows
                        ;a few counts, such as "4"
                        ;on an axis, the corresponding following
                        ;error will be a "-4".
                        ;
Type  "J <CR>"          ;Observe that X axis begins to move in
(Typing a "j<CR>"       ;positive direction. Use "p" and "f"
would move in the       ;to monitor the move's progress. You may
negative direction)     ;request from and send data to SMCC
                        ;even while it is moving.
                        ;
To change the jog       ;SMCC will send the current value
speed,type "i06<CR>";of i06 (machine feedrate).
                        ;
Now if you type         ;Where nn is larger than the current value
 "i06 nn <CR>"          ;of i06, the speed will increase. A smaller
                        ;value will decrease the speed.
                        ;
                        ;You may also change the
                        ;speed by:
Type  "% nn <CR>"       ;A smaller  value will reduce the speed  or
                        ;larger one will increase it;  maximum
                        ;value for % is 65535 and default value is
                        ;65498.
                        ;To stop the jog motion,
Type  "J or Q or        ;
  q <CR>"               ;Motion will stop.
                        ;
                        ;To jog both X and Y together,
Type  "JJY <CR>"        ;Notice that both X and Y axes will
                        ;move  and that "X" does not have
                        ;to be typed in  but "Y" must be used.
Type  "j <CR>"          ;Note that X axis reverses direction.
                        ;
Type  "jY <CR>"         ;Note that Y axis reverses direction.
                        ;
Type  "jjY <CR>"        ;Note that X and Y both stop.  You could
                        ;also use "Q" or "q" to stop motion at any
                        ;time in any direction.
                        ;To return to zero position (where SMCC
                        ;was before the jog was started):
Type  "= <CR>"          ;This is the "Return to Prejog" Command and
                        ;the SMCC will begin moving in
                        ;the opposite direction than before
                        ;until it reaches "0" position  where it
                        ;started.
Type  "v <CR>"          ;While the SMCC is returning to
                        ;zero position  check the velocity
                        ;of the motor. Note that if both
                        ;X and Y were moved, the return
                        ;velocities of X and Y are "Vector Velocities"
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;and will arrive at zero at the same
                        ;time.  Also note that if the "handwheel"
                        ;was used to move X and Y, the "=" Command
                        ;will also return X and Y to their previous
                        ;position.



                        ;Now let's proceed to make some
                        ;position command moves in MDI
                        ;mode.
Type  "i021 <CR>"       ;This places the SMCC in MDI
                        ;mode.
                        ;
Now type "F 1024        ;
  <CR>"                 ;This sets the feedrate of SMCC to be 100% of
                        ;the i06 parameter.
                        ;In MDI mode, a feedrate "F" or Time
                        ;to move "T" must be used once in
                        ;order to provide the SMCC with feedrate.
                        ;
Type  "X 10000          ;
  <CR>"                 ;The X axis will move to 10000
                        ;encoder counts.
Type  "X 10000 Y        ;
  10000 <CR>"           ;Only the Y axis, if there is one, will
                        ;move 10000 counts since X axis is
                        ;already there and this is an absolute command.
Type  " X0 Y0 <CR>"     ;Both X and Y will return to zero
                        ;position.
If your SMCC has        ;
Parabolic option        ;
(OPTION 4),             ;
Type  "t512 <CR>"       ;Sets accel/decel  time
                        ;or parabolic move time to 1/2 second.
Type "X1000:0 <CR>"     ;Tells SMCC to make a parabolic
                        ;velocity profile from 0 to 1000
                        ;counts and have 0 velocity upon
                        ;arrival. The move will take 1/2 second.
                        ;
Type "U 50000 <CR>"     ;SMCC moves to 51000 counts
                        ;but makes a normal trapezoidal move.
                        ;
Type "X0 <CR>"          ;Return to 0 position.  SMCC still uses the 1/2
                        ;second accel/decel time set in MDI.
                        ;
                        ;To set an output (M02)
                        ;high,
Type "SET 02 <CR>"      ;This command by itself  will not
                        ;cause M02 to turn-on.
Now type  "DWE10        ;
  <CR>"                 ;The Dwell Command is a
                        ;Servo Command and will cause
                        ;the SET 02 to execute. M02 turns on.
                        ;
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
Type "RES 02 <CR>"   ;Turn off M02. Nothing happens.
Now type  "X 1000    ;
  <CR>"              ;Notice that M02 turns off as
                     ;soon as X axis begins its
                     ;move to 1000 counts.
Type    "0 <CR>"     ;X axis returns to zero. Note that


                     ;the "X" is  not needed when commanding
                     ;the X axis moves.
                     ;
Type "i02 0 <CR>"    ;This makes SMCC exit the MDI mode.
                     ;We are now ready to write a
                     ;small program into the buffer and
                     ;then execute it.
Type  "z50 <CR>"     ;Clears the program buffer and
[Reserves 50 lines   ;Sets the current position
for the program.]    ;of SMCC to zero. Does
                     ;not get stored in buffer.
Type "Z <CR>"        ;Reserves 50 lines for main program
                     ;sets present position to zero.
Type  "FOR 3 <CR>"   ;Start a loop, do 3 times.
Type  "T2048 <CR>"   ;Specify a move time of 2 seconds.
Type "X 10000 <CR>"  ;Tells X axis to move to 10000.
Type "DWE 1024 <CR>";Wait one second.
Type "X0 <CR>"       ;Returns X to zero.
Type "DWE 1024 <CR>";Wait one second.
Type "NEX"           ;End of Loop.
Type "END"           ;End of Program.
                     ;
Type "R <CR>"        ;The "Run" Command will cause the
                     ;SMCC to execute the program continuously,
                     ;go to position 10000, wait 1 second
                     ;return to 0, wait 1  second and
                     ;start again.  It will do this
                     ;3 times and stop at zero.
                     ;The time for each cycle is:
                     ;2 + .5 + 1 + 2 + .5 + 1 = 7 seconds
                     ;the 0.5 seconds are the decel time,
                     ;set by t512 when in MDI mode,
                     ;see above. Move time (T) does not
                     ;include decel time.
                     ;
Type "S <CR>"        ;The "Step" Command makes
                     ;the X axis move to 10000 in
                     ;(2 seconds + 0.5 seconds) = 2.5 seconds
                     ;and stop.  The program will execute
                     ;one step at a time.
                     ;
Type "S <CR>"        ;No motion occurs in position.
                     ;Turns off for 1/2 second. This is the Dwell.
                     ;
Type "S <CR>"        ;X axis returns to zero.
                     ;
Type "R <CR>"        ;The Run Command starts
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;continuous execution of the
                        ;program  until completed.
                        ;
                        ;Now, let's assume the
                        ;"T" time should be changed to
                        ;1 second instead of 2. To
                        ;edit, proceed as follows:
Type "REP0 <CR>"        ;SMCC does nothing, but is now
                        ;placed  in  Edit Mode.  This is the
                        ;editor Replace Command.
Type "LIS <CR>"         ;SMCC now sends all of the
                        ;stored program, but adds line
                        ;numbers to the date:
                        ;0000: FOR 3
                        ;0001: T2048
                        ;0002: X10000
                        ;0003: DWE1024
                        ;0004: X0
                        ;0005: DWE1024
                        :0006: NEX
                        ;0007: END
                        ;To change T2048 to T1024,
Type "REP01 <CR>"       ;SMCC does nothing. We are telling SMCC
                        ;to replace line 1.
Type "T1024 <CR>"       ;SMCC replaces line 1 with the
                        ;new entry. T2048 is now changed to T1024,
Enter "<CR>"            ;SMCC exits edit mode.
Type "R <CR>"           ;Program now runs with 1 second
                        ;move times, instead of 2 seconds.
                        ;Please note that the editing
                        ;can be done even while the
                        ;program is running.
                        ;When SMCC comes to a stop, let's
                        ;assume we wish to save the program
                        ;permanently. (OPTION - 2).
Type "s <CR>"           ;The "s" save Command will save
                        ;the program, all the
                        ;"i parameters", the "P variables",
                        ;and the C and M assignments
                        ;into permanent memory.  These
                        ;will remain permanently saved until
                        ;new data is saved  with
                        ;another "s" Command. To save, SMCC must be
                        ;stopped and the SW-5 switch (Save Enable)
                        ;must be on.
Type "LIS <CR>"         ;Observe that SMCC sends the
                        ;program back, but line numbers
                        ;are not appended. Note that T is now 1024.
                        ;A more versatile program
                        ;would be:
Type "z 100 <CR>"       ;Note that the program is written by using
Type "FOR (P1) <CR>";variable P parameters.  This allows the
Type "t (P6) <CR>"      ;
Type "F (P2) <CR>"      ;setting of any desired value, either
Type "U (P3) V (P4) ;
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
  <CR>"                ;in the program by using SET Pn = value or
Type "DWE (P5) <CR>";on-line, even while the program is
Type "U (-P3) V      ;
  (-P4) <CR>"        ;running, by typing Pn = value.
Type "DWE (P5) <CR>";Also, the program uses incremental X and
Type "NEX <CR>"      ;Y moves, so both axes will go back
Type "END"           ;and forth as set by the value of P3 and P4.
                     ;P6 sets the accel/decel time, P2 the feedrate,
                     ;P5 the Dwell time, and P1 the Loop cycles.
```

Refer to Programming Examples Section for a detailed description of various programs.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PROGRAMMING MANUAL
SMART MOTOR CONTROLLER CARD

TABLE OF CONTENTS

PAGE

COMMAND FORMAT. 52
CARD ADDRESS SELECT COMMANDS. 53
COMMUNICATION WITH HOST COMPUTER. 54
PARAMETERS, VARIABLES, AND FUNCTIONS (I, P, C, M) 55
INITIALIZATION/SETUP COMMANDS 56
i Parameter Functional Grouping 56
Detailed Parameter Definitions 57
i01 - i02. 58
i03 59
i04 60
i05 - i06. 61
i07 - i08 62
i09 63
i10 - i12 (STD). 64
i12 [PARAB] - i13. 65
i14 66
i15 (i13 = 1). 66A
i15 (i13 = 0) - i16. 67
Jog Mode. 68
Feedrate Mode 68
Time Mode 68
i17 - i18 71
i19 72
i20[i40] 73
SMCC Servo Algorithm. 73
Servo Filter Diagram. 74
i21[i41] 75
i22[i42] - i23[i43] 76
i24[i44] - i26[i46]. 77
i27[i47] - i28[i48] 78
i29[i49] 79
i30[i50] [STD] - i32[i52]. 80
i33[i53] - i35[i55]. 81
i36[i56] - i38[i58]. 82
i39[i59] 83
i40 - i41. 84
i42 [PARAB] - i45 [STD] 85
i45 [PARAB] - I50 [PARAB]. 86
i51 [STD] - i52 87
i53 - i60[i61] 88
i62 - i64[i65] 89
i66[i67] 90
i68[i69] - i72[i73]. 91
i74[i75] - i76[i77]. 93
i78[i79] 94

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PAGE

DIRECT CONTROL COMMMANDS 95
Direct Motion Commands
Q, R, S, H, $ 95
Deceleration (Stop) Times for Special Conditions . . 96
$$, =, Z, h(X/Y), k, q 97
J(X/Y), j(X/Y), O(X/Y)n, %N,n. 98
"%", Feedrate, and Time Control Diagram. 100
Program Control Commands (Non-Buffer)
bnn, PC = v, l 101
zn, rA,v, dA,v, wA,d 102
s, innv, Cmm = A, Mnn = Cmm.b. 103
Mnn = 0/1, Pn = V. 104
Status Query Parameters (Non-Buffer)
inn, innH, innL, innD, ix 104
Control V, f, p, v, ?. 105
%, Bnn, Bx, Cmm. 106
Cx, Mnn, Mx, Pn, Px, PC, PB. 107
Program Editing Commands (Non-Buffer)
LIST n,m, LIST n, DELETE n,m, INSERT n 108
REPLACE n, zn. 108
MAP n, PLC n, PURGE n 109
Control Character Commands
<CNTRL B, C, or D>, <CNTRL R>. 109
<CNTRL V>, <CNTRL M>, <CNTRL H>. 110
BUFFER COMMANDS 111
Explanatory Codes
0 - 4, a - d 111
e, 1ae SET nn, 1ae RESET nn, 1e GOTO+nn 112
GOTO-nn, 1e GOTO nn. 112
1e GOSUB nn 113
IF/IF NOT nn = b 113
IF/IF NOT nn > b 113
IF/IF NOT nn + b 113
IF/IF NOT nn * b 113
IF/IF NOT (CC1), (CC1 and CC2), (CC1 or CC2) 113
Lnn 114
START 114
STOP 114
END 114
RETURN 114
for d 115
NEXT 115
FND 115
CASE mm, nv. 115
DIVIDE d 116
F d, T d, t d, HOME 116
DWELL d, X d, Y d. 117
AX d, AY d 118
U d 118
V d 118
: d 118
WX d 118
WY d 119

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PAGE

BUFFER COMMANDS (cont.)
d ^ d. 119
DLY d. 119
x = d 120
Y = d 120
RX = d 120
RY = d 120
R = D 120
C = d 120
LINEAR. 120
CIRCLE n 120
Circular Interpolation Diagram. 120A
TORQUE. 121
POT d 121
SLEW d. 121
Inn d 121
m a,n 121
mm a,n. 122
n a,n 122
u a,n 122
x a,n 122
SET Pn = d. 122
SET Pn = HX, HY, AX, AY 122
RESET P 122
SET Cnn = a 122
SET Mnn = Cmm.b 122
PLC Instructions (in Buffer) 123
Description of PLC Program Statements
PLC n,. 124
IF/ANDIF/THENIF/ELSEIF/AND/OR 124
THEN/AND 127
ELSE/AND 129
END 129
PURGE n 129
Variable Parameters. 129
Timers 130
ATAN (Arctangent). 131

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1. COMMAND FORMAT

The following special characters are accepted by the Smart Motor Controller (SMCC) with meanings described herein.

1. Unless otherwise stated, <CR> (Hex 0D, Carriage Return) character is interpreted as "End of Line" character. The SMCC processor responds with <LF> (Hex 0A, Line Feed) character when it receives <CR>, if i03 is set to one.

2. The SPACE character (Hex 20) is interpreted as a separation character. It is simply neglected by the processor. The operator may or may not use this character for ease of text legibility.

3. Backspace (Hex 08 or Control-H) can be used to cancel the previous character, while Control-X (Hex 18) can be used to cancel all characters in the current line.

4. The maximum number of characters in a single line of commands depend on the number of cards daisy chained. Each card can hold 52 characters in a command line; therefore, if n is the number of cards, n x 52 + <CR> characters may be sent to the cards.

5. When using SMCC V3.X (ACC-9), the semicolon character (;) (Hex 3B) is used for allowing comments to be written along with the program. The semicolon begins a comment line and a <CR> terminates it. During a download operation, the comments are stripped by the SMCC V3.X PC based program. Programs which are loaded directly by the user without using SMCC V3.X or equivalent, must not contain any comments since the SMCC itself cannot handle comments or semicolons (;).

6. Each SMCC card can only receive one command line at a time which must be terminated by a carriage return. A "command" is a character or a string of characters which cause the SMCC to take a specific action such as: J (jog +), hX (home X axis), X500Y500 (move X and Y 500 counts), etc.

7. The SMCC uses only the first three letters (at most) of a motion buffer command keyword. More than three letters may be sent, but SMCC will ignore the further letters. A listing of the program form the card will show only the first three letters of each keyword.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2. CARD ADDRESS SELECT COMMAND

The Address Select Command is composed of an "A" character followed by a single character address number. Cards 0 to 9 are denoted by numbers 0 to 9 and cards 10 to 15 are denoted by small letters a to f. For example, "A8" selects card #8 and "Ab" selects card #11. Also "AA" selects all cards simultaneously. When "AA" is selected, no slot will be sent back by SMCC to the host computer. To transmit data to the host, each SMCC must be specifically addressed.

Upon reset or power turn-on, the SMCC's CPU selects "AA" (All Card Select) as a default. Desired card must be addressed before data is requested by the host computer otherwise a "?" or <BELL> character is sent back indicating a procedural error.

Each card supports 2 axes ("X" and "Y" axes) when Brushed DC Motors or Self Commutated Brushless Amplifiers are used. Therefore, a maximum of 32 axes can be simultaneously controlled. Each card's address can be set up by setting the DIP switch on the card as follows:

SW1	SW2	SW3	SW4		
0	0	0	0	card address	0
1	0	0	0	"	1
0	1	0	0	"	2
1	1	0	0	"	3
0	0	1	0	"	4
1	0	1	0	"	5
0	1	1	0	"	6
1	1	1	0	"	7
0	0	0	1	"	8
1	0	0	1	"	9
0	1	0	1	"	10 (a)
1	1	0	1	"	11 (b)
0	0	1	1	"	12 (c)
1	0	1	1	"	13 (d)
0	1	1	1	"	14 (e)
1	1	1	1	"	15 (f)

Please note that when a single SMCC is used, it must be set up as an A0 card and when more than one SMCC is used on a daisy chain, one of them must be addressed as A0 in order to provide the SYNC clock needed for interpolated and coordinated motion.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2a. COMMUNICATION WITH HOST COMPUTER

The DIP switch is used to select the communication mode and baud rate as follows:

SW6	SW7	SW8	
0	0	0	9600 Baud
1	0	0	4800 "
0	1	0	2400 "
1	1	0	1200 "
0	0	1	----- Parallel Mode
1	0	1	300 Baud
0	1	1	38400 "
1	1	1	19200 "

SW5 of the DIP switch is used to Enable =0 or Disable =1 saving programs and parameters into EAROM memory. Regardless of the switch position, the contents of the EAROM are always uploaded upon power turn-on or rest.

When the DIP switch settings are changed, it is necessary to reset the SMCC or cycle power in order to activate the new setup.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3. PARAMETERS, VARIABLES, AND FUNCTIONS (I, P, C, M)

The SMCC gives the user several different types of variables for more power and flexibility in operating and programming the card. Below is an explanation of the different types: what they are, and how to use them.

I parameters are for the initialization and setup of the card. They give the card its 'personality' for a particular application or part of an application. They have specific meanings. Some are used only at power-on or reset; others are continually accessed. Standard SMCC has 61 I parameters; Parabolic has 80.

P variables are general-purpose parameters for the user to utilize in programs. They can be used to set positions, times, speeds, and many other things. The program can do complex arithmetic with them (within a limited syntax -- see the legal uses at the top of the motion buffer section in the programming manual). There are 15 P variables, numbered P1 to P15.

C functions are user-assigned locations in the SMCC's memory or I/O space. The user can assign them to input and output ports, to registers that hold intermediate calculations, or to I or P parameters. There are 64 C functions, numbered C00 to C63 (always use two digits). A few C functions are already mapped to input and output ports at the factory (these can be changed, but it is not advised). Each C function holds a byte of data, but some operations can work on up to 4 consecutive locations in memory for multi-byte values. C functions can be used to get at or manipulate data of interest.

M functions are user-assigned bit locations in the SMCC's memory. They are defined in terms of a particular bit of a C function byte. They can be mapped to a particular digital input or output, to a control bit, to an empty location in memory for use as a flag, to the sign bit of a parameter, or many other things. There are 64 M functions, numbered M00 to M63 (always use two digits). They can be used to control program logic and to set outputs. Several of the M functions have been pre-mapped at the factory to general-purpose digital inputs and outputs in a convenient manner (these can be changed, but it is not advised): M01 thru M05 are mapped to Machine Outputs 1 thru 5 (MO1 - M05); and M11 thru M17 are mapped to Machine Inputs 1 thru 7 (MI1 - MI7). Setting the outputs or acting on the inputs can be simply a matter of a single line in the program using M functions.

Variable values (I and P) or locations (C and M) can be stored in the EAROM (EEPROM) permanent memory. I parameter values can be stored on any SMCC. P variable values, and C and M function locations can be stored on any Standard SMCC, but only those Parabolic SMCCs with Option 2 (expanded permanent memory). Upon power-up or reset, the stored information is loaded into active memory. The information in active memory can then be changed as the card is run, but the data in permanent memory will not be changed until a new "s" (Save) command is given.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SUMMARY OF PROGRAMMING USES OF PARAMETERS

	Non-buffered Set	Non-buffered Read/Use	Motion Buffer Set	Motion Buffer Read/Use	PLC Buffer Set (THEN)	PLC Buffer Use (IF)
I	inn v	inn ix	Inn d		Inn=d	
P	Pn=v	Pn Px	SET Pn=d SET Pn= HX**	...IF(X<Pn)** X(Pn)** R=(Pn)**	Pn=d Pn=ATAN (VY/VX)**	Pn>d Pn<d
C	Cnn=a* w a,v	Cnn Cx	SET Cnn=a* m a, v mm a, v	CASEnn,vx ...IFnn=v**	Cnn,v=d Cnn,v= Cnn+d	Cnn=v Cnn,v<d Cnn,v>d
M	Mnn= Cmm.v* Mnn=v	Mnn Mx	SET Mnn= Cmm.v SETnn RESETnn	IF (Mnn=v)	Mnn=v	Mnn=v

* defining address, not setting value
** and other similar commands
v = constant value (of legal range)
n,nn,mm = parameter number
d = data (constant or P expression)
a = address (4-digit hex)

TRANSMITTING INFORMATION BETWEEN PARAMETER TYPES (WHERE IT CAN BE DONE)

To \ From	I	P	C	M	Constant
I	(Thru C and P)	Motion Buffer PLC	(Thru P)		Non-buffered Motion Buffer PLC
P	(Thru C or M)	Motion Buffer PLC	PLC		Non-buffered Motion Buffer PLC
C	(By Definition)	(By Definition) PLC	(Thru P)	(By Definition)	Non-buffered* Motion Buffer* PLC
M	(By Definition)	(By Definition)	(By Definition)	Motion Buffer** PLC**	Non-buffered Motion Buffer PLC

*by writing to address itself
**by using program logic

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

4. INITIALIZATION/SETUP COMMANDS ("i" Command Set Parameters)

Commands (parameters) described here are for X axis both in Brushed and Brushless Mode. Commands inside [ ] are equivalent commands for Y axis, when i13 (Brushless Motor Select) is set to rushed Motor. These commands may also be used in Brushless Motor Setup Commands with different meanings.

Sending a command "inn", where nn can be any number from 00 to 79, followed by <CR> without any parameter value is interpreted as Parameter Value Query. The SMCC prompts with the value of the stored parameter. Also, sending an ix will cause the SMCC to send all parameters at one time. Using an "I" (capital i) in the command line causes the line to be placed in the program buffer and acted upon during program execution. However, i38[i58] (Move and Display Scale Factor), is a two part number and cannot be placed in the buffer.

Whenever an i parameter's function is different for the Parabolic version of SMCC, a notation is made to that effect.

Following Error means the difference between actual position and desired position. Number of encoder counts is used as a unit, regardless of scale factor used.

For reference purposes, there follows a list of the i parameters in functional groupings:

---

i Parameter Functional Grouping

Mode Setup Parameters:

i02, i03, i09, i11, i12, i13, i14, i15, i19, i63

Home Setup Parameters:

i04, i24[i44], i25[i45], i26[i46]

Following Error Setup Parameters:

i00, i01, i10, i34
i60[i61] (Parabolic SMCC only)

Handwheel Control Parameters:

i16, i27[i47], i28[i48], i29[i49]

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

System Feedrate Control Parameters:

i31[i51], i06, i08

Servo Time Scale Parameters:

i05, i07

System Stability and Response Time Parameters:

i20[i40], i21[i41], i22[i42], i23[i43], i09, i30[i50], i39[i59]

PWM Limit Parameters:

i32[i52], i33[i53]

Display Control Parameters:

i17, i18, i38[i58]

Position Limit Setup Parameters:

i35[i55], i36[i56], i37[i57]
i78[i79] (Parabolic SMCC only)

Brushless/Induction Motor Control Setup Parameters:

i13, i40, i41, i43, i44, i45

Induction Motor Control Setup Parameters:

i51, i52 (Standard SMCC)
i42, i45 (Parabolic SMCC)

PLC Control Parameter:

i60 (Standard SMCC)
i62 (Parabolic SMCC)

ACC-14 AND RESOLVER Parameters (Parabolic SMCC):

i64[i65], i66[i67], i68[i69], i70[i71], i72[i73], i74[i75] i76[i77]

Detailed Parameter Definitions

---

i00 Set Following Error Limit (FAULT)
Range (0 +/- 32767), Default (2000) Unit: Encoder Count

If a positive number is used and the servo cannot follow the desired position within the specified limit, the SMCC

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

turns off the amplifier and the GLT (Get Lost Timer) is set thus disabling the SMCC and outputting the FEOD/ output (available on J1-B2 or J10-7). This is a fatal error and the operator has to reset the system in hardware to resume operation. The FEOD/ output may be used to turn system power off.

If a negative number is used in i00, the following error is still detected and the amplifier disabled, but the SMCC remains active and the host can communicate with it, even though it has stopped motion. A soft "$" or hard "$$" reset may then be used to clear the fault and re-enable the SMCC.

i01 Set In-Position Band
Range (0 - 65535), Default (100) Unit: Encoder Count

The SMCC turns on the in-position light and IPOS/ output whenever the actual position is within the specified counts from its current desired position when it is at rest. Once in-position has occurred, the SMCC is ready for the next move regardless of subsequent IPOS status. IPOS will first go true after a move is complete and the SMCC's actual position comes within +/- i01 counts of the desired position. Subsequently, IPOS will remain true as long as the actual position stays within this band and before the next move commences.

i02 Set MDI (Manual Data Input) Mode
Range (0,1), Default (0)

0: Motion Buffer Commands are stored in the Motion Buffer and can then be processed by the "S" (Step) or "R" (Run) Commands or by the "Step" or "Run" front panel switches.

1: Buffer Commands, instead of being stored in the buffer, are immediately processed upon reception of a <CR> character. This mode is used to execute one command at a time without using the Motion Buffer. Data can be ASCII or binary, and all on-line commands may be used.

The processing of Direct Control Commands or Non-Motion Buffer Commands is not affected by this mode. Continuous motion can be achieved in this mode by issuing a second command while the first is being executed. The "Buffer Full" output status can then be used as a ready/busy handshake. To exit the MDI mode i020 may be entered or a "S" (Step) or "R" (Run) Command may be given.

Note that Buffer Commands that need to execute with a "Servo Command" will not execute by themselves in MDI either. For example, a SET nn <CR> sent to SMCC (set M function output to a true value) will not function unless it is immediately

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

succeeded by a command such as DWE 10 <CR> (dwell 10 msec.). Also note that when in MDI mode, a "F" (Feedrate) or "T" (Time) must be issued at least once in order to cause SMCC to make a move when it receives a Motion Command.

Exercise caution because if returning to the program flow, the latest feedrate will be in effect until a new Feedrate Command is executed by the program.

i03 Set Handshake Enable in Host Communication
[STD] Range (0,1), Default(1)
[PARAB] Range (0,3)

0: Handshake is disabled. Processor does not send handshake character. Messages are sent without <LF> (line feed). Data is sent as: Data, <CR>.

1: Handshake is enabled. Processor sends <LF> character upon reception of <CR>. Upon receiving unrecognizable or illegal commands, it also sends <BELL> (for Parabolic SMCC) and <LF>?<CR><LF> (for Standard SMCC). Data is sent as: <LF>, DATA, <CR>, <LF>.

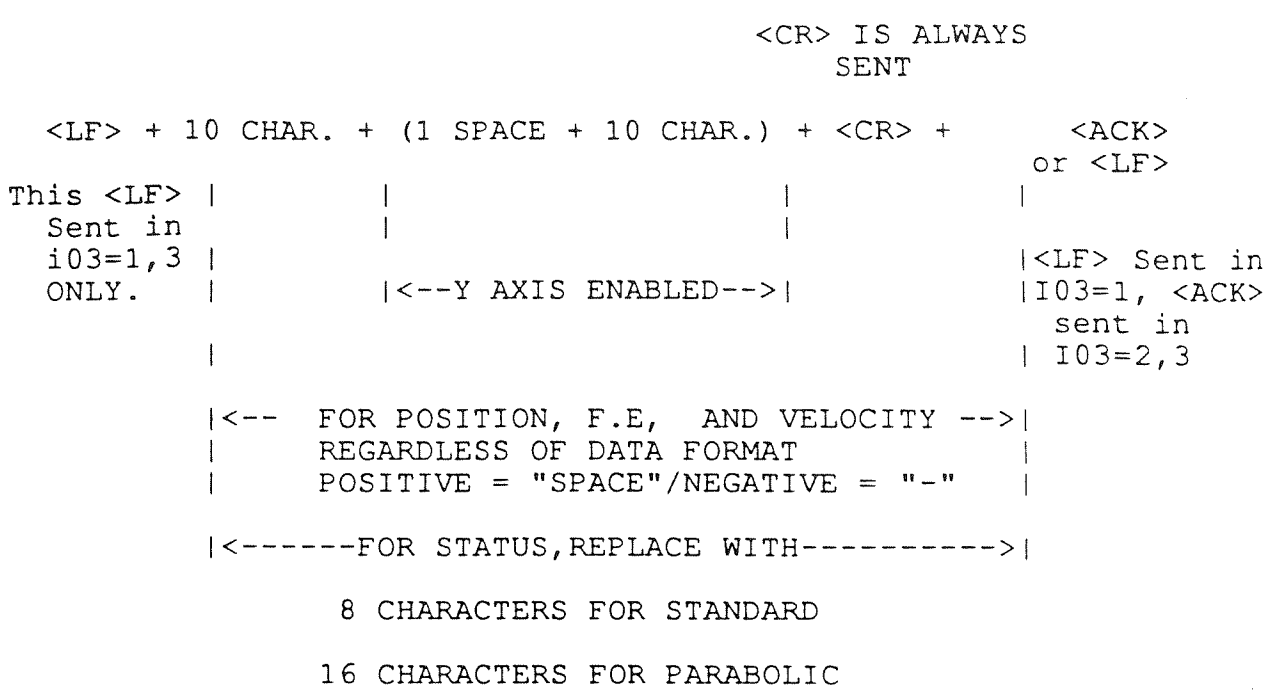
[PARAB]
2: Handshake is enabled; responds with <ACK> for valid commands or <BELL> for illegal commands. Messages are sent without <LF>. Data is sent as: DATA, <CR>, (DATA, <CR>, . . .) <ACK> (with <ACK> sent after last line only).

[PARAB]
3: Handshake is enabled; sends <ACK> or <BELL> as appropriate. Data is preceded by <LF> and followed by <ACK>. Data is sent as: <LF>, DATA, <CR>, (<LF>, DATA, <CR>, . . .) with <ACK> (after last line only).

NOTE: Regardless of the handshake, the controller sends messages if a command specifically asks for it (as in f <send following error>, inn <send stored inn parameter value>, etc.).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Below is a diagram of the format the SMCC uses to return position, velocity, following error, or status:



i04 Set Home Feedrate
Range (0 - 255), Default (0) Unit: None

This parameter sets up the speed of homing operation. Home Feedrate is calculated by the following equation.

$$\text{Home Feedrate} = \frac{i04}{256} \times i06 \text{ (Encoder counts/sec.)}$$

If i04 is set to zero, homing function will not occur.

The homing function may be started by the Control Panel switch labeled "Home", by the on-line command "h", and in the program flow (Buffer) by the "HOM" Command. The "HOM" Command when executed does not reset the program's execution pointer "PC," whereas the other two reset the "PC" causing the program to restart from the beginning after a homing cycle (see i24[i44], i25[i45], and i26[46] for more details on SMCC homing procedure).

Note that even though the X and Y axes may be independently homed, the feedrate for each axis during home is set by i04. Separate parameters for X and Y are not available.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i05 Set Servo Time
Range (0 - 65535), Default (487) Unit: Microsecond/2

This parameter specifies period of servo control servo time includes all calculations and output to amplifier. Servo time is calculated by

Servo Time Period = 2 x (i05 + 1) (microseconds)

The default setting of 487, which defines a servo time of 976 microseconds, provides approximately 30% to 35% of background time for communications, PLC, and other tasks.

Caution should be exercised when changing i05 because i21 (feedback gain) and i23 (integrator) are affected and must be readjusted since they are dependent on servo cycle timing. i20 (proportional gain) may be increased if the servo cycle is shortened, since a more rapid servo update time will improve servo performance. If servo time is reduced by 1/2, performance (gain) may be improved by a factor of 2 to 4. Conversely, if servo cycle time is shortened too much in an attempt to increase performance, the background time may be reduced to the point where SMCC will cease to function because of insufficient time to handle calculations, communications, and PLC.

No value of i05 other than 487 may be stored in permanent memory. If you wish to use a different value, you must change it after every power-up or reset.

NOTE: Do not change i05 except for special requirements.

i06 Set Reference Feedrate
Range (0 - 8388607), Default (2000)
Unit: Encoder Counts/sec.

This parameter specifies the Absolute Feedrate of the machine when F1024 (100%) is used in the program. Commands i31[i51] (Jog Feedrate) and i04 (Home Feedrate) are specified in relation to i06. The actual machine feedrate is a function of Fn (Feedrate) Command as:

$$\text{Machine Feedrate} = \frac{n}{1024} \times i06 \text{ (Encoder count/sec.)}$$

$$\text{Jog Speed} = \frac{i31[i51]}{256} \times i06$$

$$\text{Home Feedrate} = \frac{i04}{256} \times i06$$

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i07 Set Time Scale
Range (0 - 65535), (%) Default (65498) Unit: None

i07 sets the time scaling for SMCC movements. That is, it defines the time incremental (dt) for calculating the desired position for the next servo cycle. The governing equation is:

i07 = 65536 x [2 x (i05 + 1)/dT]
where
dT = Units for T or t Commands (in microseconds)

If we wish to work in our standard time units of 1/1024 seconds (binary milliseconds) and use the default servo time, we calculate:

with dT = 1/1024 sec. and i05 = 487 : i07 = 65498

This is the default value.

i07 serves as the power-on or reset value for the time base. Once a "%" or "POT" Command is executed, the value with that command overrides the i07 value for moves. However i07 always controls the time base for DWELL commands. i07 also serves as the % value scaled to maximum input voltage in i14 = 1 mode.

If i05 is changed in order to speed up or slow down the servo cycle, i07 should be readjusted in order to keep the actual time scale of the SMCC constant. The standard time scale is dT = 1/1024 sec. (one binary millisecond). Also note that if i07 is changed, the on-line (non-buffer) % Feedrate Command (POT in buffer) must also be readjusted in order to keep the velocity factor constant.

NOTE: i07 should normally not be changed except for special requirements.

i08 Set Acceleration/Deceleration Time
Range (0 - 65535), Default (512) Unit: 1/1024 sec.

This parameter governs total accel/decel time whenever speed change occurs. When i08 = 1024, accel/decel time is one second for any speed change.

The accel/decel time is also controlled by the "t" Command in the program buffer.

Accel and decel times are equal and both controlled simultaneously. It is possible to make accel and decel different by breaking up one move into two consecutive moves with an I08 or "t" change between the split blocks.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

In the Standard SMCC, i08 may be changed on the fly and will affect all accels and decels. A "t" Command executed in the program buffer will change the value of the i08 parameter. An "I08" Command in the program buffer or a non-buffered "i08" Command will override the last "t" Command. That is, for the Standard SMCC, the "i08", "I08", and "t" Commands all have exactly the same effect when executed.

In the Parabolic SMCC, i08 is read once upon power turn-on or reset and sets the value of "t" initially. Once a "t" Command is executed, i08 does not affect the calculated (programmed motion) accel/decel, but only the jog accel/decel. It is therefore possible to have different accel/decel values set for jog and programmed moves.

"t" also controls the minimum move time and will override the "T" move time if T<t and (in Parabolic) the i12 (calculation time) if i12<t. "t" should not be set to a value less than 10 for one-axis control and 15 to 20 for two axes. The effect of too small a "t" time is to cause the SMCC to stop executing the program because it has run out of block-move calculation time. SMCC's capability is approximately 60 blocks/second with two axes and 100 blocks/second with one axis. When using intensive PLC programs in the SMCC, a larger i12 time ensures smoother operation because it provides the servo with more calculation time to counteract the effect of PLC's "background" time reduction.

If i11 = 1 (feed hold slew control on), then i08 sets the deceleration time upon hitting a software or a hardware limit (on either a jog or a calculated move), or for after an H (feed hold) Command on a jog move.

Refer to Parabolic Section for further description of the difference in effects of i08 and t between the Standard and Parabolic SMCC.

i09 Set Position Integration Mode
Range (0,1), Default (1)

1: Position integration (i23[i43]) is enabled only when motion is stopped. The integrator is automatically activated at the end of the move when run velocity becomes zero.

0: Position integration is enabled all the time: both during a move and when stopped. Caution must be exercised since this mode may result in consistent overshoot when stopping even though following error will be near zero during the move (see i23[i43] for integrator time constant and explanation of its operation).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i10 Set Following Error Limit (Indicator)
Range (0 - 65535), Default (1000) Unit: Encoder Count

If the following error is greater than this limit, the SMCC turns on the Following Error Light and outputs FILD/signal. However, shutdown does not occur unless the i00 limit is reached.

This parameter is useful in providing a visual indication of following errors occurring when the Following Error output is tied to an LED, such as when connected to an ACC-16 Control Panel.

The FILD/signal may be tied to FEOP/ by installing the E22 jumper on the SMCC. This causes both the i00 and i10 to generate FEOP/.

i11 Set Feed Hold Slew Control
Range (0,1), Default (1)

0: Feed Hold Slew Control is disabled. When feed hold occurs a Travel Limit switch is activated (opened), a software limit is reached, or a "q" Command is given, feedrate change is accomplished as rapidly as possible without controlled deceleration. The motor will actually come to a stop and then return to the point at which Feed Hold occurred (plus the following error distance, if any).

1: Feed Hold Slew Control is enabled. Deceleration rate on feed hold is controlled by i08 (acc/dec time) when jogging or by % slew rate if in a calculated move (default is 64 : 1 second for a full scale to 0 change). Motor comes to a stop at the end of decel. The % slew rate value is deducted every servo cycle from the current setting of the % Command until zero % value (stop) is reached. Deceleration rate is controlled by i08 if a hardware or software limit is hit, or if a "q" Command is given.

i12(STD) Select 24/6 kHz PWM Frequency
Range (0,1), Default (0)

0: 24 kHz PWM Frequency.

1: 6 kHz PWM Frequency.
Normally, this parameter should be left at 24 kHz. However, if the SMCC's direct drive outputs are used for switching "Totem Poles" in an H bridge or multiphase application, 6 kHz may be used for slow or bipolar semiconductor devices.

NOTE: Do not change i12 except for very special requirements.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i12 [PARAB] Move Calculation Time
Range (0-65535), Default (40)
Unit: Binary Milliseconds, Minimum (10)

The Parabolic SMCC uses i12 as a predefined servo calculation time to ensure that in multiple-axis systems all cards will begin motion simultaneously. A certain amount of time is always necessary to do the planning calculations for a move or move segment. Note that it may become necessary to use a higher value of calculation time when moves based on variable parameters are made, especially those involving divisions.

When SMCC is making a normal trapezoidal move, the calculation for the next move begins i12 time before the end of the move, which is the beginning of decel ("t" time is the accel/decel time).

When SMCC makes a parabolic move, i12 time occurs at the very end of the move because "t" time represents the move time and a parabolic move does not have a defined deceleration time. If i12 time becomes longer than "t" time in either trapezoidal or parabolic moves, calculations are then made in "t" time instead of i12 time.

i13 Select Brushless Motor Mode (Commutated by SMCC)
Range (0,1), Default (0)

0: Brush type DC Motor Control - Two-axis control is possible. The two error output signals from the SMCC now function as independent +/-10 volt DC outputs for the X and Y axes. i15 1 may be used to disable the Y axis or i15 0 to enable both X and Y axes. In this mode, i40 through i59 define the Y-axis setup just as i20 through i39 do for the X axis.

1: Brushless DC Motor Control (or AC Induction or Stepper Motor) - Only a single axis is enabled (see Brushless Motor Control Setup Commands, if brushless motor is selected). The two error output signals from the SMCC now generate motor phase commands 90/120 degrees apart for 4/3 phase sinusoidal (i15 0) or trapezoidal (i15 1) outputs where the frequency controls the rotation (velocity) of the motor and the amplitude the amount of torque delivered. The third (and fourth, if necessary) phase is generated by a local balance loop in the amplifier. Maximum frequency is 1000 Hz (60,000 RPM for two-pole motors) and maximum amplitude is +/-10 volts peak to peak. In this mode, i40 through i59 control the setup of the motor commutation.

Note that the meaning of i15 now changes. It is used to select sinusoidal/trapezoidal commutation outputs.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i14 Select Processor A/D Convertor Mode
Range (0,1,2,3), Default (0)

There are four A/D inputs available on SMCC. Two are always used to monitor current levels (see below). The other two (ANA2 and ANA3) have several uses, depending on the setting of i14, as described here:

0: Analog inputs ANA2 and ANA3 are not used.

1: Time Base (Feed Pot) Mode -- In this mode, the X-axis analog input ANA2 is used to control the time base (and so the speed) of both axes. The input voltage controls the time base just as the '%' and 'POT' commands, or the input frequency with i16=2 or 3, would do. This allows a simple potentiometer to control the rate of execution of a program. '%' commands are overridden by the input voltage in this mode. Do not try to control the time base of multiple cards with an analog input; the A/D converters on the different cards cannot be counted on to give identical results, and the cards will lose coordination. Proper shielding of the analog input wiring is very important in this mode, as noise on the line translates very noticeably into movement jitter.

On a card without analog input conditioning (no Option 3), 0 volts in means 0 rate of execution (%0) and 5 volts in means maximum rate of execution (%65535). On a card with input conditioning (Option 3) -10 volts in means zero rate of execution, and +10 volts in means maximum rate of execution. The actual % value set by the voltage can be gotten at any time using the '%' command without an argument. (When leaving 14=1 mode, the % value is left where it was last set by the analog input voltage until something else changes it.) In this mode, the ANA2 input should never be left floating -- results could be unpredictable. Do not have i16=2 or 3 when i14=1, because in that setting, the handwheel frequency input will override the analog input voltage in controlling the time base.

2: In this mode, the analog inputs control the speed of jog moves: ANA2 controls X-axis speed; and ANA3 controls Y-axis speed. Without option 3, 0 volts in means zero speed, and +5 volts in means a jog speed equal to i06; with Option 3, -10 volts in means zero speed, and +10 volts in means a speed of i06. The analog inputs thus replace the function of the jog speed parameters i31 and i51 when the SMCC is in this mode. The starting and stopping (and direction) of the jog moves are still controlled by the mechanical jog inputs (XJPI, XJMI, YJPI, and YJMI on the J1 connector) and/or the jog commands (J, j, JY, jY).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3: In this mode, both axes are always in continuous jog moves, with speed controlled by the analog inputs, and the speed range controlled by the mechanical jog inputs. ANA2 controls the X-axis speed; and ANA3 controls the Y-axis speed. The jog inputs for each axis control the speed range for that axis as follows:

J+ off	J- off	x1 speed range (max of i06/8)
J+ on	J- off	x2 speed range (max of i06/4)
J+ off	J- on	x4 speed range (max of i06/2)
J+ on	J- on	x8 speed range (max of i06)

With a standard control panel, J+ and J- cannot be turned on simultaneously by the jog switch. However, the Home switch turns on both J+ and J- for the X axis, and the Prejog (=) switch turns on both for the Y axis.

Without Option 3, 0 volts in means maximum negative velocity for the specified range, 2.5 volts in means zero velocity, and 5 volts in means maximum positive velocity for the range. With Option 3, -10 volts in means maximum negative velocity, 0 volts in means zero velocity, and +10 volts in means maximum positive velocity. With the x8 speed range and maximum velocity, the jog speed is equal to i06.

The remaining two analog input channels -- CUR0 (ANA0) and CUR1 (ANA1) -- are used to monitor current and provide I-squared-T protection (see i32[i52] and i33[i53] descriptions for setting the desired protective current levels) if Option 3 is installed. These lines input a voltage proportional to the absolute value of the motor current with a range of 0 to 10 volts. The Analog Inputs section of Basic Specifications has more details.

Refer to the Analog SIP (Option 3) Schematic for details of the condtioning circuitry; refer to the Analog Input Potentiometer Connections diagram for instructions on wiring the analog inputs with or without Option 3.

i15 Set Y Axis Disable in Brushed Motor Mode (i13=0)
Range (0,1), Default (1)

0: Y axis is enabled. SMCC is a two-axis controller.

1: Y axis is disabled. If the Y axis is disabled, the Y error output can be used as a programmable voltage generator by using the PLC Commands to write to the appropriate registers - - ff16 contains the magnitude (0 - 255), and ff00 bit 1 is the sign bit (0 = positive, 1 = negative) (see Address Map), or by using the "OYn" Command.

NOTE: Even while using one axis only the limit switches for Y axis MUST be disabled (grounded); otherwise SMCC will not make any moves.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

It is not essential to disable the Y axis if you are only using one axis, but you may want to do so for three reasons: 1) you may use the Y-axis output as a general purpose analog output; 2) you may use the Y-axis encoder counter for data logging purposes; and 3) you ease the calculation load on the SMCC.

i15 Select Look Up Table Sinusoidal or Trapezoidal Mode (i13 = 1)
Range (0,1), Default (0)

0: Current Control Valve (error output) is specified by an Index Table indicated by i43. This mode is used to set sinusoidal or six-step motor current.

1: Trapezoidal Motor Mode - Current Control Value is calculated internally instead of using table.

i16 Select Handwheel Operating Mode
Range (0,1,2), Default(0)

0: Standard Mode - Handwheel input is used to change motor position. Number of pulses represents position and pulse frequency represents velocity. The ratio of handwheel to motor movement or scale factor is governed by i27[i47] (where a value of 4096 for i27[i47] represents a scale factor of one) and the mode of operation by i28[i48].

1: Moves strictly follow handwheel encoder or pulse and direction. No calculated moves are possible in this mode. This is a special mode allowing very fast servo cycle thus permitting "tight" handwheel following. i27 [i47] and i28[i48] also affect this mode.

2: Synchronization Mode - Time base uses Y handwheel in Parabolic and X in Standard. Externally injected pulse and direction or quadrature encoder input can be used to control the SMCC's time base so that multiple cards use an identical and variable time base for synchronous program execution. The SMCC will execute whatever programmed motion it is commanded to execute at a rate of proportional to input frequency.

3: Parabolic Only - Time base uses X handwheel; otherwise same as i16 2.

NOTES: When using i16 2 or 3, it is good practice to disable the X and Y handwheel mode by setting i28[i48] to 3 (or 7 on the Standard SMCC when using external handwheel). When i16 = 2 or 3, do not have i14 = 1, or you may have a conflict as to what is controlling the execution speed.

Please refer to "%" chart for a better understanding of i16 and i14.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

In settings 2 and 3, program execution rate is slaved to the external clock or encoder frequency. The effects of i27[i47] are explained in detail for various modes of operation:

Jog Mode

i31[i51] * i06 < 2 EXP 28 = 268435456

Jog Speed = (i31[i51] * i06)/256

$$\frac{\text{out speed}}{\text{in speed}} = \underbrace{\frac{i31[i51] * i06 * i47}{i27 * 2^{33}}}_{\text{Parabolic}} = \underbrace{\frac{i31[i51] * i06 * i27}{2^{36}}}_{\text{Standard}}$$

Feedrate Mode

$$\frac{\text{out speed}}{\text{in speed}} = \underbrace{\frac{Fn * i06 * i47}{i27 * 2^{35}}}_{\text{Parabolic}} = \underbrace{\frac{Fn * i06 * i27}{2^{38}}}_{\text{Standard}} \quad Fn = \text{Feedrate}$$

In the feedrate mode, move times are calculated to the nearest 1/1024th of a second. Hence, the actual speed can be different than the commanded speed. The error is small for long moves.

Time Mode

$$\frac{\text{out speed}}{\text{in speed}} = \underbrace{\frac{(\text{Delta P}) * i47}{T * i27 * 2^{15}}}_{\text{Parabolic}} = \underbrace{\frac{(\text{Delta P}) * i27}{T * 2^{18}}}_{\text{Standard}}$$

In the three explanations above, a further clarification can be made which better explains the various equations:

$$\frac{\text{Out Speed}}{\text{In Speed}} = (\text{DATA})$$

Out Speed = (DATA) x In Speed

Jog Rate or Feedrate or Time-Based Speed	=	(DATA) x (Handwheel Input Frequency)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

An additional way to represent the effective time and feedrate is:

1) For Parabolic SMCC

100% feedrate = (counts/binary millisecond) = (i27 *32768)/i47

Note that Parabolic uses a ratio of both i27 and i47 for more flexibility and accuracy.

2) For Standard SMCC

100% feedrate = (counts/binary millisecond) = (8 * 32768)/i27

The term feedrate above refers to the i07 value, the % Command (on-line) or the POT Command in the buffer. In i16 2 or 3 mode the external frequency is used by SMCC to calculate the effective "%" it will use when it makes its moves. Calculations are made every servo cycle and "SLEW" control (rate of change of "%") is disabled in order to avoid errors when multiple SMCCs are used. The result of instantaneous changes in "%" may be some velocity granularity. It is therefore recommended that as high a frequency as possible be used for the handwheel time base. This reduces the granularity quite effectively. Once a maximum frequency is chosen, for example, 33.156 KHz (= 32,379 cts/bin. msec.) for Standard SMCC:

$$32.379 = (8 \times 32768)/i27$$
$$i27 = 8096$$

When i27 is set to 8096, a frequency of 32.379 (cts/b. msec.) will be calculated to represent a "%" value of 65,498 which is 100% feedrate. Caution must be exercised, therefore, to limit the maximum input frequency since the maximum value of "%" is 65,535. Therefore a frequency of (32.379 x 65,535)/65,498 = 32.397 cts/b. msec. will cause the % Command to be equal to 65,535 which is its maximum value.

A frequency higher than 32.397 cts/b. msec. will cause the % Command to overflow and the SMCC's speed can drop to 0 or a small value since % 65,536 is the same as % 0.

Yet another way to discuss the control of the SMCC's % Command is to consider that with a % value of 65,498, the scaling of SMCC's "T" (time to move) and "t" (i08 or accel/decel time) is such that a T of 1024 represents 1 second of time. If % is changed to 1/2 of its value, 32749, then a T of 1024 will represent a time of 2 seconds. The % Command therefore controls the time base of SMCC.

A small programming example is described for a machine with a spindle driver and two traverse axes which will wind wire on the spindle. The system's physical data is as follows:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Spindle: Max. RPM = 600 = 10 RPS
Total travel = 100 Revolutions
Encoder Counts/Rev = 2000 x 4 = 8000
This is an independently driven axis.

Traverse: Y axis is expected to vary its pitch every 10th rotation of the spindle. Total travel is 10 inches; 10 rotations of spindle should cause traverse to move 1 inch. X axis moves across at constant pitch. Both X and Y are "slaved" to the spindle axis and the spindle axis encoder is fed to the SMCC's "X Handwheel" input and used to establish the "%" value which controls the SMCC's time base.

```
                    ;FOR PARABOLIC SMCC
A0z100              ;Clear buffer
i283                ;Disable X and Y handwheel mode
i483                ;
                    ;Assume 8000 counts/rev. of spindle
i163                ;X axis handwheel is time base for SMCC
i27100              ;Make 1 bin. msec. = 100 counts
i4732768            ;100% feedrate (%) = Counts/bin. msec.
                    ;=(i27 x 32768)/i47 = 100, uses i27 and i47
                    ;for more flexibility and precision
                    ;1 rev = 80 msec. of time = 80 x 100 = 8000
                    ;counts of the spindle
                    ;100 cts/bin.msec. is %65498 = 100% feedrate
                    ;Actual time will be 1 sec. per 10 revs (600 RPM)
                    ;total move, 10 seconds, 80000 counts
T800                ;Time for 10 revolutions of spindle
U(P1)V(P2)          ;10 revs, P1 and P2 are distances to move
U(P1)V(P2)          ;20 revs, change ratio (Y axis)
U(P1)V(P3)          ;30 revs, change ratio (Y axis)
U(P1)V(P3)          ;40 revs, change ratio (Y axis)
U(P1)V(P4)          ;50 revs, change ratio (Y axis)
U(P1)V(P4)          ;60 revs, change ratio (Y axis)
U(P1)V(P5)          ;70 revs, change ratio (Y axis)
U(P1)V(P5)          ;80 revs, change ratio (Y axis)
U(P1)V(P6)          ;90 revs, stop Y axis
U(P1)V(P6)          ;100 revs
U0V0                ;If you are going to disable the external time
                    ;base so you can move the axis independently, be
                    ;sure axis is stopped before switching from
                    ;i163 to i160. On the fly switching of time
                    ;base is possible; just take care that a big
                    ;velocity change will not occur.
END                 ;End of program
                    ;P1=wire traversed each 10 revs of
                    ;spindle in encoder counts for X axis
                    ;P2, 3, 4, 5, 6 same for Y axis
                    ;Just send new variable parameter
                    ;values to change the program.
                    ;All P variables represent distance to
                    ;move the traverse axis in encoder counts.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i17 Set Velocity Display Time Base
Range (0 - 2 ^31), [STD] Default (15000)
[PARAB] Default (30000)

i17 sets the sample period at which encoder counts are measured and converted to velocity.

$$i17 = \frac{\text{"sample period"}}{2} \text{ in microseconds (Standard SMCC)}$$

i17 ="sample period" in microseconds (Parabolic SMCC)

The displayed data is the distance which SMCC measures, in scaled encoder counts (see i38[i58]), during the "sample period" specified by i17.

Velocity = Delta position during sample period

For Standard SMCC, if i17 = 15000, sample period is 30 milliseconds with a 2000 counts per revolution encoder.

Velocity = counts/30 milliseconds
= 1/2000 Rev Per 30 milliseconds
= Rev/60 seconds
= RPM

Velocity is measured and displayed at the rate of three times per second.

By using i38[i58] X and Y scale factor it is possible to adjust scaling so that on systems that have different encoder counts per motor revolution for X and Y, the displayed velocity can be made identical.

i18 Select Display Type to be Used
Range (0-3), Default (3)

This parameter selects the type of hardware output display that SMCC will address. The display information can include position, velocity, following error, and status byte for each axis. The settings are:

0 is LED display - 2 units x 1 line (8 digits/line)
1st display shows X position.
2nd display shows Y position if i13 = 0;
X velocity if i13 = 1.

1 is LCD display - 2 units x 2 line (24 digits/line)
1st display shows X position and X velocity on 1st line;
X F.E. and X status on 2nd line.
2nd display shows Y position and Y velocity on 1st line;
Y F.E. and Y status on 2nd line.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2 is LCD display - 2 units x 1 line (40 digits/line)
1st display shows X position, X velocity, X F.E., and X status;
2nd display shows Y position, Y velocity, Y F.E., and Y status.

3 is LCD display - 1 unit x 2 lines (40 digits/line)
1st display shows X position, X velocity, X F.E., and X status on 1st line;
Y position, Y velocity, Y F.E., and Y status on 2nd line.
(2nd display can show X F.E. and X status on 1st line;
Y F.E. and Y status on 2nd line;
useful if actual displays are shorter than 40 digits long.)

X position format is governed by i38.
Y position format is governed by i58.
X velocity format is governed by i17 and i38.
Y velocity format is governed by i17 and i58.
X following error format is governed by i38.
Y following error format is governed by i58.
X status shows bits of 3rd status byte
(febb for Standard SMCC; feb7 for Parabolic SMCC)
Y status shows bits of 4th status byte
(febc for Standard SMCC; feb8 for Parabolic SMCC)

Refer to the SMCC Memory Bit Map for definitions of the individual bits.

If you are not using a display setting, i18 to 0 takes the least background time for the least interruption of other background tasks.

i19 Not Used in Standard SMCC.

i19[PARAB] Set Highest Card Address
Range (0 - 15), Default (0)

In Parabolic SMCC, i19 is used to set the highest card address in a system. The value of i19 is used in DMA mode to allow the SMCC to receive and send "Packet" data without interruption for card addressing. It is not necessary to have all card addresses consecutive to each other. However, if there are gaps, the host would have to send extraneous data to non-existing cards. DMA is automatically received by each card in succession and it is also sent automatically in successive order. DMA mode allows only binary data transmission of position and velocity and can be entered by sending characters <CNTRL B, C, or D>. (See section on "SMCC-Parabolic" for detailed description.) Also, parameter i63 controls the data type to be used in DMA mode.

The following parameters, i20-23 [i40-43], are used to control the system response time and stability.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i20[i40] Set Proportional Gain Constant

Range (0 - 65535), Default (50) Unit: 1/Encoder Counts

(Please refer to SMCC Setup Section for a detailed description of i20[i40].) The proportional gain is used to increase, in direct proportion, the amount of error correction signal the SMCC outputs for a given following error. The output is as follows:

(1.25 * i20[i40])/65536 = VOLTS/ENCODER COUNT

The SMCC's servo algorithm is shown here for reference both for the Standard and Parabolic SMCC.

The Standard SMCC uses i30[i50] as a X1 or X256 gain multiplier.

****

SMCC SERVO ALGORITHM

1) SMCC WITH PARABOLIC OPTION

$$Eout = i20*k0*\{EP(n) - [i21*AV(n)] + i22*DV(n) + i30*k1*[DV(n)-DV(n-1)] + i23*k2*IP(n)\}$$

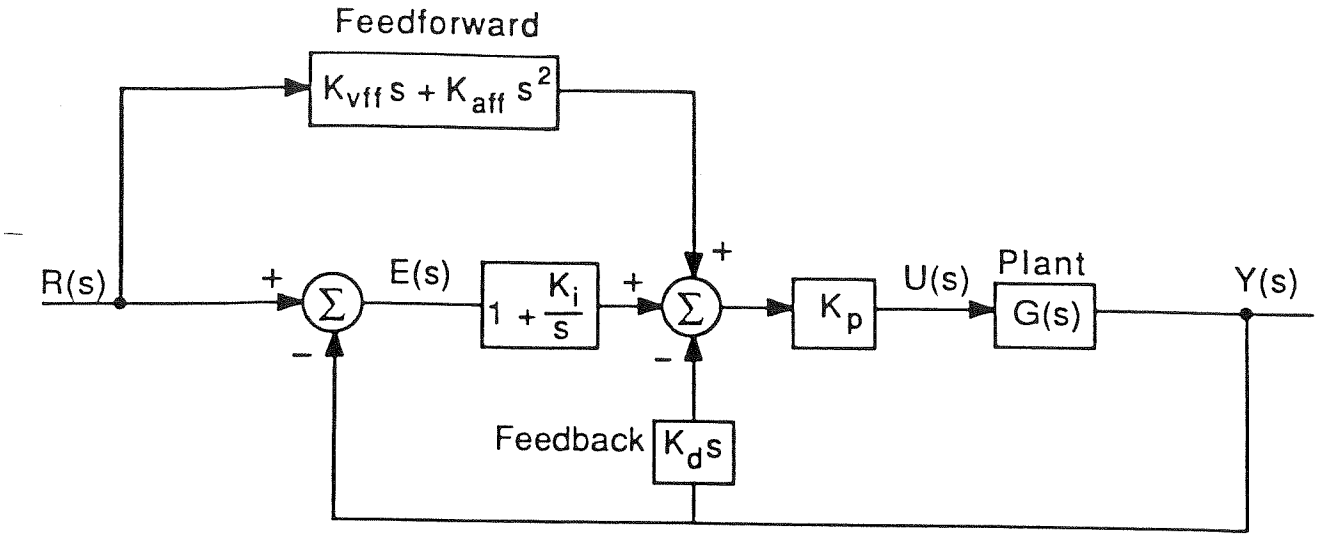
Where:

- i20 = Proportional Gain
- k0 = 1.25 volts/65536 (Used to scale proportional gain)
- EP(n) = Position error = Dpos(n) - Apos(n)
- Dpos(n) = Present desired position
- Apos(n) = Present actual position
- i21 = Differential gain (Velocity feedback)
- AV(n) = Present velocity = Apos(n) - Apos(n-1)
- i22 = Velocity feedforward gain
- DV(n) = Desired velocity = Dpos(n) - Dpos(n-1)
- i30 = Acceleration feedforward gain
- k1 = 4 (Used to scale accel feedforward)
- i23 = Integral gain
- k2 = 1/4096 (Used to scale integral gain)
- $IP(n) = \sum_{j=0}^{n-1} EP(j)$ (integral of position error)

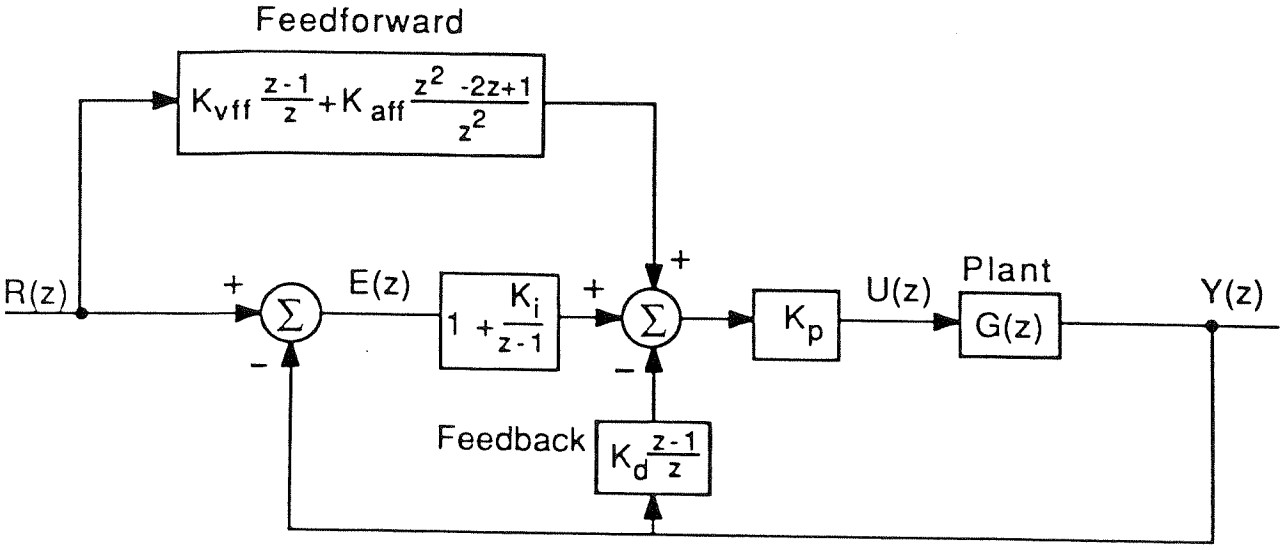
(n-j) is used to represent the value j servo cycles old.

NOTE: The actual velocity is modified when the 1/T mode is used by the ratio of the servo time to the actual measured time between encoder counts.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



ANALOG EQUIVALENT OF SMCC-P SERVO FILTER



SMCC-P DIGITAL SERVO FILTER BLOCK DIAGRAM

R(s) or R(z) : Reference (Computed Trajectory)
E(s) or E(z) : Error in Position (Following Error)
U(s) or U(z) : Command Output
Y(s) or U(z) : Actual Position
G(s) or G(z) : Plant Transfer Function
K_p : Proportional Gain
K_i : Integral Gain
K_d : Derivative Gain
K_{vff} : Velocity Feedforward Gain
K_{aff} : Acceleration Feedforward Gain

Artisan Scientific - Quality Instrumentation. Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2) SMCC STANDARD (NON-PARABOLIC)

$$Eout = i20*k0*\{EP(n) - [i21*k1*EP(n-i22-1)] + i23*k2*IP(n)\}$$

Where:
- i20 = Proportional gain
- k0 = 1.25 volts/65536 for i30 = 0
  = 1.25 volts/256 for i30 = 1
  (Used to scale proportional gain)
- EP(n) = Position error = Dpos(n) - Apos(n)
- Dpos(n) = Present desired position
- Apos(n) = Present actual position
- i21 = Differential gain
- k1 = 1/256 (Used to make i21*k1 < 1)
- i22 = Time base for feedback in servo cycles.
- i23 = Integral gain
- k2 = 1/65536 (Used to scale integral gain)
- $IP(n) = \sum_{j=0}^{n-1} EP(j)$ (integral of position error)

(n-j) is used to represent the value j servo cycles old.

****

i21[i41] Set Differential Gain Constant

Range (0 - 255), [STD] Default (240)
[PARAB] Default (40) Unit: None

The differential gain constant is used to provide SMCC the velocity feedback needed to stabilize the servo loop if a tachometer is not used in the amplifier. If a tachometer is used, i21 should generally be set to zero (although it can be used to add more damping if the tachometer loop is not properly tuned).

The greater the differential gain used, the more stable the system is, but also the greater the following error at velocity. On the Parabolic SMCC, this following error can be counteracted using i22[i42] velocity feedforward.

The effect of i21[i41] is different between the Standard and Parabolic SMCC:

$$i21[STD] = \frac{256 * i21[PARAB]}{(1 + i21[PARAB])}$$

or

$$i21[PARAB] = \frac{i21[STD]}{(256 - i21[STD])}$$

On the Standard SMCC, i21 must usually be well above 200 to have significant effect (the default is 240). On the Parabolic SMCC, i21 is almost always set below 50.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i22[i42] [STD] Set Velocity Filter Time Delay
Range (0 - 15), Default (0) Unit: Servo Cycles

In the Standard SMCC, i22 is used to provide velocity filtering in order to improve low velocity performance. Normally, velocity in a digital position system is computed as Pos(n) - Pos(n-1). At low velocities, this measurement gets rough, because it is incapable of computing fractional velocities. Since the digital servo filter is dependent on this quantity, its action also can get rough at low velocities. The i22 parameter allows the velocity to be computed over a larger number of servo cycles: that is, Vel(n) = Pos(n) - Pos(n-i22-1). A higher value of i22 effectively averages the velocity over a number of cycles (i22 cycles), reducing low velocity jitter, making for smoother operation at low speeds. However, this averaging makes the system slow to respond to speed changes, resulting in a "mushy" feel to the servo. If i22 is set to the default value of zero, of course, velocity is measured in the standard manner.

i22[i42] [PARAB] Set Velocity Feedforward Gain
Range (0 - 255), Default (0) Unit: None

In Parabolic SMCC, i22 is used to generate a term proportional to the desired velocity that is added into the servo command. This term is very useful in reducing following errors caused by tachometer damping (in systems with tachometer-based amplifiers) or differential-gain damping (in systems with no tachometer). The damping is necessary for system stability, but it introduces a following error roughly proportional to the velocity. Velocity feedforward can then be used to eliminate (or greatly reduce) this following error, and since it is not part of the feedback, it does not affect system stability. Also, it is more useful than error integration for reducing following errors on variable-speed moves, because error integration cannot keep up with varying speeds as feedforward can.

In a system without a tachometer, i22 should generally be set equal to i21. In tachometer-based systems, i22 should be set to minimize following error at speed (and its correlation to velocity).

i23[i43] Set Integral Gain Constant
Range (0 - 255), Default (0) Unit: None

The integral gain of the SMCC is used to correct residual position errors due to off-balance loads, friction, temperature, and time drift, etc. It may also be used to dynamically reduce the following error to zero while the SMCC is performing a move. Essentially, the integrator makes a correction every servo cycle and the amount of correction is controlled by the value of i23[i43]. The

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

"Capture Range" of the integrator is 1000 encoder counts. This means that SMCC will correct up to that much error automatically. Greater errors will leave a residual error. (Refer to Servo Algorithm Section in i20 and to Setup Procedure Section for better understanding of the integrator.)

Parameter i09 controls when the integrator will be active. i09 = 1 activates it only when SMCC is at rest; i09 = 0 activates it at all times.

i24[i44] Select Home Direction
Range (0,1), Default (0)

0: Home in negative direction.
1: Home in positive direction.

This parameter sets the direction in which Home will take place. (See i04 for Homing Feedrate Setup and i25[i45] for home offset.) Note that during a homing procedure, the i35 [i55] and i36[i56] software travel limits are disabled.

i25[i45] Set Home Offset
Range (0 - +/-32767), Default (500) Unit: Encoder Counts

During the homing process and after detection of the machine's Home switch (Home Flag) and the encoder's "C" channel, the axis will move by the amount and direction specified by i25[45]. (See i26[i46] for definition of polarities that may be used for the "C" and Home Flag inputs.) Note that the "User Flag" input may be used as well.

i35[i55] and i36[i56] software travel limits are disabled during homing procedure.

i26[i46] Set Home Flag and User Flag Control Bits
Range (0 - 7), Default (0)

This parameter allows the user to specify the manner in which a position-capturing interrupt is generated. There are several uses for this interrupt: to capture the reference position in a homing operation (home flag); to verify that no counts on an incremental encoder have been lost in a revolution (encoder "C" channel - the once around pulse); and to capture the position at a certain event (user flag). These interrupts capture the position only of their own axis. The time uncertainty of the capture is on the order of 20 microseconds: at the maximum 1 MHz encoder rate, the error can be up to 20 counts; at rates below 50 KHz, the error is zero.

i26[i46] controls the polarity and logical combination of the C Channel, Home Flag, and User Flag inputs that generate

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

the interrupt. (A "/" after the signal name means the logical complement, not the differential signal; "AND" denotes the formal Boolean operator.)

i26[i46] Value	Triggering Condition
0	(C/ AND HF/) falling
1	(C AND HF/) falling
2	HF rising
3	UF rising
4	(C/ AND HF) falling
5	(C AND HF) falling
6	HF falling
7	UF falling

The combination of C Channel and Home Flag is useful because it allows the home switch to be located loosely, with the C Channel providing the precise positioning.

(Parabolic SMCC only) i26 also establishes the condition of X-axis inputs that generate the trigger in the "Go Until" move (Xd ^ d). The X-axis interrupt acts on both axes for this situation only. i46 establishes the condition of Y-axis inputs that generate the trigger for WUT (Wait Until Trigger) moves. See Software Update section, page 414, for details.

i27[i47] Set Handwheel Scale Factor
Range (0 - 65535), Default (4096) Unit: None

This determines the ratio of handwheel count vs. motor encoder count. Desired position indicated by handwheel is calculated as follows.

$$\text{Desired Position} = \frac{\text{i27[i47]}}{4096} \times \text{Handwheel Count}$$

The maximum ratio yields 16 times the input count. Exercise caution when using a high ratio.

See i16 2 or 3 for the use of i27[i47] in external time base mode.

When the handwheel counter is used for 1/T velocity decoding by i28[i48] = 2, i27[i47] should be set to 32 x (i05 + 1) for proper velocity scaling. With i05 at the default value of 487, i27[i47] = 15616.

i28[i48] Select Handwheel Mode Control
[STD] Range (0 - 7), Default (6)
[PARAB] Range (0 - 3), Default (3)

On the Standard SMCC, modes 0 - 3 specify the other-axis encoder signal as handwheel input, while Modes 4 - 7 use external handwheel inputs.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

On the Parabolic SMCC, modes 0 - 3 specify the external handwheel input. Modes 4 - 7 are not used. The Parabolic SMCC cannot use the other-axis encoder as handwheel input.

0 or 4: Standard Mode - Machine zero position is not changed or translated. Position follows handwheel during move and position display changes - inactive during a calculated move.

1 or 5: Position Offset Mode - Machine zero position is changed and is translated as the handwheel moves. Position display does not change - active during run or stop.

2: 1/T Mode [PARAB only] - Handwheel is disabled and its counters are used for special velocity decoding mode. Jumper E40 must be removed for the X axis to work in 1/T mode, and E41 must be removed for the Y axis. Also, i27[i47] must be set for proper velocity scaling (15616 for the default servo cycle time). See i27[i47] description. See Parabolic section of manual for details.

3 or 7: Handwheel position control disabled but the specified handwheel signal source (external or other axis) may be used for time synchronization (see i16).

i29[i49] Set Handwheel Encoder Control Bits
Range (0 - 19)[0 - 11], Default (3)

This parameter controls the direction and multiplication factor of the handwheel encoder's A/B quadrature input. Also, it allows a pulse and direction input mode to be used where the A channel is used for pulse input and the B channel for direction.

i29[i49]

0[0]	Pulse and Direction	CW (counts up when B is high)
1[1]	x 1 Quadrature Decoding	CW (counts up when B leads A)
2[2]	x 2 Quadrature Decoding	CW (counts up when B leads A)
3[3]	x 4 Quadrature Decoding	CW (counts up when B leads A)
16[8]	Pulse and Direction	CCW (counts up when B is low)
17[9]	x 1 Quadrature Decoding	CCW (counts up when A leads B)
18[10]	x 2 Quadrature Decoding	CCW (counts up when A leads B)
19[11]	x 4 Quadrature Decoding	CCW (counts up when A leads B)

All other values are not allowed.

When using the handwheel, be sure the handwheel inputs are enabled (see i28[i48]).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i30[i50] [STD] Select Gain Multiplication Scaler Control SMCC
Range (0 - 1), Default (0)

0: Proportional Gain Constant (i20) is directly used internally.

1: Proportional Gain Constant (i20) is multiplied by 256 and then used internally. This control is used to effectively increase the range of Proportional Gain Constant. The value of D/A converter output DAC is calculated by:

$$DAC = i20 \times k0 \times \left[E(n) - \frac{i21}{256} \times E(n - i22 - 1) + \sum_{j=0}^{n-1} \frac{i23 \times E(j)}{65536}\right]$$

where E(n) is error input used in the internal processing, and k0 = 1.25V/65536 for i30 = 0, or 1.25V/256 for i30 = 1.

i30[i50] [PARAB] Acceleration - Feedforward Gain
Range (0,255), Default (0) Scale: None

The Acceleration Feedforward Gain allows the user to reduce the following error generated during accel and decel. i30[i50] generates a term which is proportional to acceleration and adds it to its servo error output.

All other servo gains should be adjusted satisfactorily first and then i30[i50] should be increased gradually until a desirable following error during accel/decel is achieved.

i31[i51] Set Jog Feedrate
Range (0 - 255), Default (200) Unit: None

This parameter sets up the speed of jogging operation. Jog Feedrate is calculated by the following equation:

$$\text{Jog Feedrate} = \frac{i31[i51]}{256} \times i06$$

Note that Jog Feedrate is affected by both i06 and i31[i51]

Jog speed also can be changed by using "%" on-line or "POT" in the Buffer Command. This, however, will also affect the time values used in the program.

i32[i52] Set Normal PWM Limit Control
Range (0 - 255), Default (128) Unit: None

Limits error output (Current or Velocity Command) to

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

amplifier. 0 is no output, 255 is full output or +/-10V.

May be used in tachometer systems to limit velocity, and in systems without a tachometer to limit torque.

i33[i53] Set Protective PWM Limit Control
Range (0 - 255), Default (64) Unit: None

Sampling Rate: 43 msec.
Suggested Setup: i32[i52] > 2 x i33[i53]

This parameter sets up a "fallback" value if a time-integrated maximum allowable PWM limit is exceeded. Once 75% of i32[i52] is reached and maintained for approximately three seconds, SMCC goes into Protective Mode which uses Protective PWM Limit Value (i33[i53]) until current goes down to corresponding value. It then restores the output limit to i32[i52] level. The I-squared-t output and LED indicator are turned on during protective cycle.

i34[i54] Set Backlash Compensation
Range (0 - 65535), Default (0) Units: Encoder Counts

This parameter allows the SMCC to compensate automatically for backlash between the motor and the load. When direction is changed, SMCC will add i34[i54] counts to the length of the move.

For example, if your last move was negative ending at X = 0, and i34 were set to 100 counts, then an X2000 move would actually cause a move of 2100 encoder counts. If the next following command were X1500, there would be a move of -600 counts.

The position as shown on the LED/LCD display, or sent in response to the "p" Command does not reflect the changes caused by i34[i54]. In the example above, it would show rest positions of 0, 2000, and 1500. (A position caught right when a move starts in a new direction can show the backlash. In the above example, you may see -100 and 2100 displayed for an instant.)

i34[i54] does not affect jog moves or handwheel following moves, except to note the direction of movement. If the next programmed move is in the opposite direction from the last jog or handwheel move, i34[i54] will be added to the length of the programmed move.

See Software Update section, page 416, and i80, for information on an enhanced backlash function available on new Parabolic SMCCs.

i35[i55] Set Software Positive Position Limit
Range (-1 to [67,108,863]-1), Default (-1) Unit: Encoder Count

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i36[i56] Set Software Negative Position Limit
Range (-1 to 67,108,863), Default (-1) Unit: Encoder Count

The i35[i55] and i36[i56] parameters are used to prevent motion beyond set limits. A -1 value sets the limits to maximum value (effectively no limit). i36 takes a positive value, but means a negative limit (e.g. i36 20000 sets the negative position limit to -20000). Caution must be exercised, since the SMCC will travel beyond the set limits by an amount equal to the distance needed for deceleration (done in i08 time if i11 = 1) plus a maximum of five milliseconds of delay due to background servo tasks.

Both of these parameters are disabled when a homing function is being performed. On the Standard SMCC, these limits are not operative during handwheel following moves.

i37[i57] Set Position Range
Range (+/- [2*27]-1), Default (0) Unit: Encoder Count

This parameter sets a circular position rollover range.

If a negative number is specified, range is set up for both directions up to the specified absolute value. If a positive number is specified, range is set up for positive direction only (0 - specified range).

Range is used primarily in rotary tables or closed cycle systems where an automatic position rollover is desired.

In the Parabolic SMCC, parameter i78[i79] is used to control the SMCC's behavior in this mode as follows:

i78[i79] = 0 specifies motion within the range using the shortest path to follow.

i78[i79] = 1 specifies direction of motion as specified in the Move Command.

i38[i58] Set Position and Display Scale and Format
(n:m.d), Default (1:1.0)
Range: n (-2^31 - +2^31 - 1)
m (-2^31 - +2^31 - 1)
d (0 - 7)
m/n (0 - 32)

Desired Display and Motion Data = Encoder Counts x $\frac{m}{n}$ x 10^{-d}

The scale factor can be used to set up the display and data entry so as to represent the machine's moves in engineering units rather than encoder counts.

This multiple-part parameter controls the input and output scaling of position data. Parts m and n define the

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

conversion ratio, and d sets the order of magnitude of the conversion, plus the number of digits used to the right of the decimal point.

Example: Motor has 500 lines/Rev. encoder. Lead screw is 5 pitch. It is desired to display position in inches with 2 decimal points.

1) Multiply encoder by 4, using i39[i59] value of 3 or 7.
2) You now have 2000 encoder counts/Rev.
3) The 5 pitch lead screw requires 5 revolutions for 1" of travel. 5x2000 = 10000 encoder counts/inch.
4) Set i38[i58] so n=10000 (convert to inches), m=100 (show hundredths of an inch), and d=2 (decimal point two places in): n can be reduced to 100 and m to 1 because it is their ratio that is important, so the command becomes i38[i58] 100:1.2.
5) The position displayed will now show "1.00" for 1 inch of displacement.

Note that when i38[i58] is active, the input data to the MCC must also be entered with the scaled format. An "X0.50" entry in the above example will move one half inch. Any further digits to the right of the decimal point (more than d digits) will be ignored.

The leading zero must be used for fractional counts. For example X0.25 is okay, X.25 is not.

Important note: This parameter only affects position values as they enter and leave the SMCC (i.e. the scaling is done as part of the transmission process). This means that this scaling factor does not affect P-Variable moves (e.g. X(P1) Y(P2)). Distances and positions specified by P-Variables must always be in encoder counts, regardless of the i38[i58] scaling factors in effect.

i39[i59] Set Position Encoder Mode Control Bits
Range (0-7), Default (3)

0	Pulse and Direction CW	(counts up when B is high)
1	x 1 Quadrature Decoding	(counts up when B leads A)
2	x 2 Quadrature Decoding	(counts up when B leads A)
3	x 4 Quadrature Decoding	(counts up when B leads A)
4	Pulse and Direction CCW	(counts up when B is low)
5	x 1 Quadrature Decoding	(counts up when A leads B)
6	x 2 Quadrature Decoding	(counts up when A leads B)
7	x 4 Quadrature Decoding	(counts up when A leads B)

As shown by the above selections, the position encoder may be selected as A/B quadrature, multiplied by 1, 2, or 4 and direction may also be selected. Note that reversing encoder direction may also be accomplished by swapping the "A Channel" signals for "B Channel". Also, a pulse (A Channel) and direction (B Channel) (Hi = positive direction) may be specified as encoder input.

(See i29[i49] for controlling the operating mode of the handwheel encoder.)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

NOTE: THE FOLLOWING DEFINITIONS OF i40 THROUGH i59 ARE ONLY IN FORCE IF i13=1 (BRUSHLESS MOTOR MODE). IF i13=0, i40 THROUGH i59 ARE THE Y-AXIS EQUIVALENTS OF i20 THROUGH i39.

i40 Set Encoder Counts per Electrical Revolution (per pole pair or cycle, for AC induction and brushless DC motors only) Range (0 - 65535), Default (1000) (4-pole motor, 2000 count/ Revolution Encoder) Units: Encoder Counts

Correct setup of this parameter is critical because the internal phase commutation requires this information. Note that the unit is counts per electrical revolution which is calculated by

$$\text{Counts Per Rev. (i40) Electr.} = \frac{2}{\text{\# of poles}} \times \text{encoder counts per Mech. Rev.}$$

If a 1000 count per revolution encoder is installed in a four-pole motor, i40 should be 500. The default value of 1000 corresponds to a 1000 count/rev. encoder for a two-pole motor or a 2000 count/rev. encoder for a four-pole motor.

Examples:

encoder Counts/Rev.	number of motor poles	calculated i40 value
2000	4	1000
2000	2	2000
1000	4	500
1000	2	1000
500	4	250
500	2	500

NOTE: On the Parabolic SMCC, if i50 is set to one, i40 must be set to the number of counts in three electrical cycles of the motor (see i50 description).

i41 Set Phase Advance Scale Factor
Range (0 - 255), Default (64) Unit: None

Commutation phase advance value is proportional to the product of shaft speed and i41. An initial adjustment is necessary to obtain maximum torque at the desired maximum speed (adjust for minimum current). Once this adjustment at maximum speed is made, the SMCC will then automatically adjust the phase advance as a function of velocity to optimize current consumption.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

This parameter adds a phase advance proportional to the velocity. For a speed of V in revolutions per second, the phase advance will be i41*V*k. The k is different for Standard and Parabolic.

k = 45/16384 for Standard (.082 degrees for 1800 RPM per increment of i41)
k = 45/1024 for Parabolic (1.32 degrees for 1800 RPM per increment of i41)

i42 [PARAB] Induction Motor Magnetization Current
Range (0 - 255), Default (0)

(i42 is not used for AC induction motors on the Standard SMCC. This function is controlled by i52 on the Standard card.)

In Parabolic SMCC, when induction motor control is selected by i13=1 (select brushless motor) and i45 > 0 (slip gain), then i42 is used to set the magnetization current level of the AC induction motor. The current level is dependent on the amplifier used. Normally a 5% to 10% value of maximum current is chosen; therefore, a value of i42 of 5 to 25 can be used. The higher i42 is, the greater the torque and the warmer the motor will get, even at standstill.

i43 Set Look Up Table Address
Range (129 - 131), Default (130) Unit: None

(129): Address for 6-Step Mode

(130): Address for 3-Phase Sinusoidal Mode

(131): Address for 4-Phase Sinusoidal Mode

i44 Select 3/4 Phase
Range (0, 1), Default (1) Unit: None

0: 4-Phase Motor

1: 3-Phase Motor

This parameter sets the phase 1, phase 3 output phase shift to 90 degrees for 4-phase and 120 degrees for 3-phase motors.

i45 [STD] Set Hall Sensor Phase Offset
Range (0 - +/-32767), Default (0) Unit: Encoder Counts

For Parabolic SMCC, parameter i54 controls this function. This parameter is used to correct mechanical misalignment between the hall effect physical position and the actual phase position.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

This is an angle (in encoder counts) that is added to the initial phase position after the auto phasing is complete. It is active in all forms of phasing, including Hall effect, Resolver and Current Null phasing, and in the case of Parabolic SMCC, i54 can be used to set resolver offset.

i45 [PARAB] Set Induction Motor Slip Gain
Range (0 - 65535), Default (0) Unit: None

Refer to i51 for description.

i48 Set Star-Delta Connection
Range (0,1), Default (1) Unit: None

0: Brushless Motor Star (Y) Connected

1: Brushless Motor Delta Connected
This parameter controls the manner in which the SMCC performs its phase finding when used with a brushless motor.

i50 [STD] Encoder/Resolver Selection
Range (0,1), Default (0)

0 = Use Standard Phasing Initialization (assumes encoder)
1 = Initializes Phasing with Resolver

NOTE: On the Parabolic SMCC, this function is controlled by i72.

i50 [PARAB] Enable Triple-Cycle Motor Definition
Range (0, 1), Default (0)

Setting this parameter to one permits the use of encoders and resolvers whose number of counts per revolution is not divisible by three in cases where the brushless or induction motor has 3 or 6 cycles (6 or 12 poles). On the Standard SMCC, or with i50=0, there must be an integer number of counts per electrical cycle of the motor (i40 is set to this value). When i50 is set to one, i40 must be set to the number of counts in three electrical cycles of the motor, so the number of counts in one cycle does not have to be an integer. This is particularly useful for using absolute encoders or resolvers, whose counts per revolution are typically powers of two, with 6-pole or 12-pole brushless motors.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i51 [STD] Set Induction Motor Slip Gain
Range (0 - 255), Default (0) Unit: None

NOTE: This function is controlled by i45 in the Parabolic SMCC (where it has a range of 0 - 65535). i51 is not used in the Parabolic SMCC.

If this value is not zero, then i52 controls the magnetization PWM and induction motor control is enabled (see i52). When i51 is in use (non-zero) it controls the slip frequency fluctuations needed to control the AC induction motor's torque.

This term is used for induction motors only and is used to produce a slip frequency that is proportional to the torque required. The range and scale is different for Standard and Parabolic. The range for Standard is 0 to 255 and for Parabolic is 0 to 65535. The slip frequency is T*i51[i45]*k lines per second where T is the torque current in PWM units (255 = full scale) and k is a constant that depends on the SMCC type. k = 1024/256 for Standard (1024 is the servo update rate) and k = 5859.375/65536 for Parabolic (5859.375 is the phasing update rate).

i52 [STD and PARAB] Set DC Brushless Motor Phase Finding Time
Range (0 - 255), Default (20) Unit: 43 milliseconds

[STD only] Set AC Induction Motor Magnetization Current
Range (0 - 255), Default (0) Unit: None

(On the Parabolic SMCC, this function is controlled by i42.)

For a DC brushless motor (when i13 = 1 and i51 (STD) or i45 [PARAB] is set to zero), i52 controls the length of time allocated to the automatic phase finding that must take place upon power turn-on or reset. Generally, phase finding requires from 1/2 to 2 seconds.

For an AC induction motor (when i51 (STD) or i45 [PARAB] is not set to zero), phase finding is not necessary, and i52 controls the quiescent magnetization current into the motor, i52 0 sets 0 current and i52 255 sets maximum current available from the amplifier. Caution must be used not to set this value too high since it will tend to overheat the motor even while it is a rest. A typical setting is i52 5 to 20.

NOTE: On the Standard SMCC, if i51 (Slip Gain) is greater than zero (this selects an induction motor since brushless DC motors have no slip gain), then i52 = Magn. Current. If i51 is zero, then i52 is phase finding time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

On the Parabolic SMCC,if i45 (Slip Gain) is zero, then i52 is phase finding time. If i45 is greater than zero, i52 is not used.

In effect, i51 and i52 (STD) or i45 and i42 [PARAB] control the AC induction motor's gain and operating characteristics.

i53 Set Phasing Current Value
Range (0 - 255), Default (20) Unit: None

Sets the amount of current delivered to motor (brushless) when the phases are being located upon power turn-on or reset. This motion causes the motor to move approximately +/- 60 degrees of shaft rotation. Note that if hall effects are installed, the phase finding motion is reduced to looking for the first hall effect edge transition and that the direction of motion can then be controlled and occurs at full torque.

i54 [PARAB] Hall Effect Or Resolver Offset
Range (+/-32767), Default (0) Unit: Counts

(See i45 for explanation of phase offset operation for hall effect, phasing, or resolver alignment.)

NOTE: Please refer to Installation and Startup procedure for starting up brushless and AC induction motors. Parameters i15 and i40 to i54 all pertain to these types of motors.

i60 [STD] PLC Enable
Range (0 - 1), Default (0)

NOTE: This function is controlled by i62 in the Parabolic SMCC.

0 = Disabled 1 = Enabled

This parameter is used to activate or deactivate the PLC program stored in SMCC. If i60 = 1, the PLC program will be active at all times, regardless of whether a regular program is executing or not. The only exception to this is during the loading of a PLC program, when SMCC automatically disables the PLC (from the time it receives an 'IF' Command until it receives an 'END' Command).

i60[i61] [PARAB] Deadband
Range (0 - 255), Default (0)

Specifies a +/- position range from desired position where SMCC does not close the servo loop. This is useful in systems that have a lot of backlash between the motor and the load.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

NOTE: Parameters i61 and greater are for Parabolic version only.

i62 PLC Enable
Range (0 - 1), Default (0)

0 = Disabled 1 = Enabled

See i60 for description.

i63 Parabolic Mode Control for DMA Communication
Range (0 - 1), Default (0)

0 = SMCC receives Parabolic data: binary position and velocity data.
1 = SMCC receives Standard data: binary position data only.

i63 determines what data the Parabolic SMCC expects when it is in DMA mode. When i63 = 0, it expects binary position and velocity data for each axis (to define a parabolic move segment). When i63 = 1, it expects only binary position data (to define a linear move segment).

For details on the format of DMA communications, see the DMA section of the Parabolic Manual.

i64[i65] Encoder/Resolver Base Address
Range (0 - 65535), Default (0)

These parameters for the X and [Y] axis are used to specify the address location at which the absolute encoder's data (parallel binary) or the resolver data is to be read by SMCC. An ACC-14 must be used for reading parallel binary (encoder or other) data and an ACC-14 with ACC-15 plugged into it must be used to read resolver data. The following chart provides a list of all the allowable addresses that can be used for these i parameter settings.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

	E2/E1 JUMPER E2 E1	BASE ADDRESS ACC-14	BASE + 0 PARALLEL ENCODER J1	BASE + 3 PARALLEL ENCODER J3	BASE +32 RESOLVER J6	BASE +16 RESOLVER J7
ACC-14 #1	ON ON	fd00h 64768	ACC-14 Located at 64768 may not be used for encoder/resolver feedback.			
ACC-14 #2	ON OFF	fd40h 64832	fd40h 64832	fd43h 64835	fd60h 64864	fd50h 64848
ACC-14 #3	OFF ON	fd80h 64896	fd80h 64896	fd83h 64899	fda0h 64928	fd90h 64912
ACC-14 #4	OFF OFF	fdc0h 64960	fdc0h 64960	fdc3h 64963	fde0h 64992	fdd0h 64976

Hex values are provided for reference only (they are useful for debugging with the "r" Command). i64[i65] values are entered in decimal.

Please note that any combination of encoders and resolvers may be used for the X or Y axis so long as the proper hardware (ACC-14 and ACC-15 combination) and i parameter definitions are used.

i66[i67] Speed of Second Resolver (Ratio of "fine to "coarse")
Range (0 - 255), Default (0)

These parameters are used to specify the gear ratio between two resolvers, "fine" and "coarse", when they are used to provide absolute position data on a machine. The gear ratio may be one to 256 and represents the count ratio of the "fine" and "coarse" resolvers. A ratio of "zero", the default value for i66[i67], tells the SMCC that there is only one resolver attached to that axis. Any value greater than zero causes SMCC to look for two resolvers and combine the "coarse" (geared-down) and "fine" (direct-connected) resolver readings in order to derive an absolute position.

The principle of operation here is that the "fine" resolver is used to provide a once per motor revolution position data while the "coarse", through the gear ratio, keeps track of the number of revolutions the "fine" resolver has made. Therefore i66[i67] specify the "speed" or ratio of the per revolution count of the "fine" resolver and the "coarse" resolver's counts for that revolution. For example, if both resolvers are set to a "size" or resolution of 4096 (12 bits) and the gear ratio is ten, and there are therefore ten counts of "fine" for every one count of "coarse", then i66[i67] is set to ten. If the "fine" resolver's resolution is increased to 16384 (14 bits), then the ratio of "fine" to "coarse" is now 40, even though the gear ratio is still ten, and i66[i67] should be set to 40 thus representing the ratio of actual counts between the "fine" and "coarse" resolvers.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i68[i69] Second Resolver Base Address
Range (0 - 65535), Default (0)

These parameters, like the i64[i65] "fine" resolver parameters, are used to specify the base address at which SMCC will collect the data for the "coarse" resolver. The table of addresses given in i64[i65] is applicable and may be used.

i70[i71] Absolute Encoder, Resolver Bias (Position Offset)
Range (67108863 to -67108864), Default (0) Units: Counts

The i70[i71] parameters are used to provide a "bias" or "offset" to the absolute position of the SMCC so as to allow the user to create an absolute zero position which coincides with the machine's actual zero position. Without this bias capability, the user would have to position the axis at the machine's zero position and then decouple the resolver(s) or Absolute Binary Parallel encoder and rotate them manually until their reading is also zero. The bias parameter can be used instead as follows:

1) Position the axis using the jog or handwheel so that it is physically where you wish to see your machine be at zero position (Machine Zero).

2) Read the position using "p" Command.

3) Enter the negative of the position read as your i70[i71] bias parameter.

   For example, if indicated position is +16,500, use -16,500 for bias.

4) When the bias number has been entered, it will not take effect until an online "$" Command (Soft Reset) is issued to SMCC. Saving the bias number using the "s" Command (Save) will cause the SMCC to always read the absolute position, add the bias number to it, and then set its current position accordingly.

i72[i73] Encoder, Resolver Size (Fine Resolver Resolution)
Range (0 - 23), Default (0)

These are the parameters that are used to configure the SMCC for reading an A/B quadrature encoder, a resolver, or a parallel binary encoder.

1) When i72[i73] is set to zero (Default Value), this tells the SMCC to read its A/B Quadrature encoder channels.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2) When i72[i73] is set from one to seven, it specifies the "size" or "resolution" of the resolver being used. The ACC-15 R/D converter card has software selectable resolutions. This means that when using the same resolver, it's possible to obtain 10, 12, 14, or 16 bits of resolution or counts per revolution of 1024, 4096, 16384, or 65536. The penalty for higher resolution is that the maximum RPM of the motor is reduced from 60,000 at 10 bits to 900 at 16 bits. It is the ACC-15's R/D converter which limits the RPM so that the maximum position counts per second are approximately 1 MHz. Even though the SMCC provides seven possible resolutions for the resolver, only four are used as follows:

i72[i73]	BITS/RESOLUTION	COUNTS/REVOLUTION	MAXIMUM RPM
1	10	1,024	62,400
2		N O T U S E D	
3	12	4,096	15,600
4		N O T U S E D	
5	14	16,384	3,900
6		N O T U S E D	
7	16	65,536	975
0	USE A/B QUADRATURE	INCREMENTAL ENCODER	

NOTE: The i39[i59] parameter, "Position Encoder Mode Control", must be set to three or seven (x 4 multiplication) for the resolver counts to be counted correctly.

Please note that the ACC-15's R/D converter outputs parallel data bits to indicate its position. The SMCC reads this absolute position once upon power turn-on or reset and then subsequently relies on the fact that ACC-15 converts the parallel data from the resolver into A/B quadrature output pulses which must be connected to the SMCC's position encoder channel. This is why the i39[i59] Position Encoder Mode Control parameter must be set to a "x 4" state in order to count the resolver counts properly. Setting i39[i59] to "x 2" or "x 1" will divide the counts per resolver revolution down by two or four.

3) Setting i72[i73] to a number greater than eight sets the SMCC up to read a parallel binary encoder as follows:

i72[i73] 8 to 23 specifies a parallel binary input of 8 bits to 23 bits (256 counts to 8,388,608 counts). 23 bits is the maximum allowed.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Also note that even though the J1 and J3 connectors on the ACC-14, where the parallel data input is connected, can handle 24 input lines, only 23 bits are allowed because the 24th line is used as data strobe by the SMCC in order to "freeze" the data sent to it by the Absolute Parallel Binary encoder. The encoder should be capable of freezing the data as long as the strobe line is present (the strobe polarity is controllable by SMCC).

Similar to the resolver operation, the SMCC reads all the parallel data bits upon power turn-on or reset and subsequently reads only the 16 least significant bits of the data and derives incremental position data from these. The data read process involves special data filtering in order to prevent erroneous data from being read.

4) Setting i72[i73] to a number greater than 23 is not allowed even though the SMCC will accept values up to 31. A summary of all the values for i72[i73] is as follows:

i72[i73]

0	Use A/B Quadrature Encoder
1 - 7	Resolver Size
8 - 23	Parallel Encoder
24 - 31	Not Allowed

Do not change i72[i73] while moving. A reset or power-on cycle is needed to make a new setting register properly.

i74[i75] Second Resolver Size (Coarse Resolver Resolution)
Range (0 - 7), Default (0)

The settings of these parameters are the same as i72[i73] except that they apply only to the "coarse" or geared resolver and values greater than seven (which would specify a parallel encoder) are not allowed. The SMCC does not handle two Absolute Parallel Binary encoders in a "geared" system where a "fine" and "coarse" encoder can be used.

i76[i77] Encoder "Data and Strobe" Logic Level
Range (0 - 3), Default (0)

This parameter is used to specify the polarity of the Absolute Parallel Binary encoder data bits and the polarity of the "strobe" or "freeze" output that the SMCC sends to the encoder.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i76[i77]

0 (Default)	Data Lines and Strobe Non-Inverted
1	Data Lines Inverted, Strobe Non-Inverted
2	Data Lines Non-Inverted, Strobe Inverted
3	Data Lines and Strobe Inverted

Please note that if the data lines are inverted, this in effect reverses the direction of the encoder count.

i78[i79] Range Mode Control (Shortest Distance or Absolute)
Range (0 - 1), Default (0)

See parameter i37[i57] which controls the SMCC's position range (rollover).

i78[i79] = 0 selects the shortest distance to travel from one position to the next.

i78[i79] = 1 selects a direction controlled move within the range.

i80 Backlash Take-up (PROM Versions V2.78P and newer only)
Range (0 - 15), Default (0)

When i80 = 0, the backlash function is as on older versions, when it is only active on programmed moves, not jog or handwheel moves.

When i80 > 0, backlash compensation is also active on jog or handhweel moves, and the value of i80 determines the number of counts per background cycle (approximately 4 msec.) at which the backlash is taken up on direction reversal. See Software Updates section, page 416, for details.

i81 Circle Error Tolerance (PROM Versions V2.78P and newer only)
Range (0 - 15), Default (0), Units (Encoder Counts)

This parameter allows a circular move of distance slightly greater than twice the radius to be done without error (this situation usually occurs due to roundoff errors when specifiying a half-circle). If the specified move distance is greater than twice the radius by a number of encoder counts less than i81, the radius is increased slightly so that a half-circle move is made. If the error is a greater than i81, the program will stop and an error will be generated. See Software Updates section, page 417, for details.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

5. DIRECT CONTROL COMMANDS

These commands are not stored in the buffer but executed immediately upon reception.

Direct Motion Control Commands

NOTE: * Indicates that these commands are duplicated by the manual control panel.

Q Quit
Quit running the program at the end of the current block. Also used to stop Jogging or Terminating "O" (Open Loop Output) Command.

* **R** Run
Start Executing Motion Buffer Instructions continuously without stopping between steps. (Can also be issued via Control Panel switch.)

* **S** Step
Execute Current Buffer Instruction, stop and advance buffer pointer. (Can also be issued via Control Panel switch.) After stop, buffer pointer advances one block so that another Step Command executes next instruction.

* **H** Feed Hold (Stop/Abort)
Stop immediately and hold current position. To resume motion and complete current block, use "R" (Run) or "S" (Step) Command. (Can also be issued via control panel switch.)

$ Soft Reset
Initiates phase finding operation and preserves all internal data. Usable to correct motor phasing, motor type, and position feedback device type only.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DECELERATION (STOP) TIMES FOR SPECIAL CONDITIONS (if i11 = 1)

Stop Condition \ Type of Move	Programmed (Calculated) Move	Jog (Non-Calculated) Move
Feed Hold ("H" or Hold Button)	$t_{stop} = \frac{\%\ POT\ \#}{SLEW\ \#}$ (bin. msec.)	t_{stop} = i08
Abort Move ("q" or Abort Button)	t_{stop} = i08	t_{stop} = i08
Software Limit Exceeded	t_{stop} = i08	t_{stop} = i08
Hardware Limit Tripped	t_{stop} = i08	t_{stop} = i08

NOTES:

1) These times control the deceleration of the commanded positions. If the stop time is short, the motor may not be able to follow the stop without overshoot.

2) These times are valid for both Standard SMCC and Parabolic SMCC. **However, only on the Standard SMCC does a "t" command change the i08 value.**

3) There may be several (up to 5) milliseconds of delay after the condition occurs before the deceleration begins, due to processing limitations.

4) If i11 = 0 (feed hold slew control disabled), the commanded position is stopped immediately. The motor will almost certainly overshoot and bounce back.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

* **$$** Hard Reset
(Can also be issued via control panel switch.) Essentially causes a complete reset to occur. Hard reset from host computer does not change baud rate and card address.

* **=** Return to Pre-Jog Position
Return to the position from which jog or handwheel movement was started. This command is allowed only when the motion is stopped. (Can also be issued via Control Panel switch.) "=" may also be used after "a" to complete the move that was aborted (but not proceed further in the program).

Z Zero Set
This command sets current position to be zero position. It is allowed only when the motion is stopped. The equivalent buffer commands are X = N and Y = N where position can be set to any number from 0 to 64 million.

* **h(X/Y)** Home
Go to home position as specified by the X and Y, and i04, i24, and i25 [i45]. The equivalent buffer command is "HOME (X/Y)". (Can also be issued via Control Panel switch but only for both X and Y simultaneously.)

The exact format of the "h" Command is as follows:

hX causes the X axis only to home
h causes the X and the Y axes to home
hY causes the Y axis only to home

The h and Input switch Home Commands differ from the "HOME" Buffer Command in that they begin a home cycle and reset the program execution pointer to the beginning of the program, whereas "HOME" does not reset the pointer.

k Kill Servo
This command sets error output to zero and opens the servo loop. It is used to disable the amplifier and allow free motion of the motor for learning or other uses. To restore normal servo operation use the "Q" Command. While the "k" Command is active, the following error limit output (i00) is disabled.

* **q** Abort Current Move
Aborts the current move and resumes with next block's move if "R" or "S" Command received. (Can also be issued via Control Panel switch.) If the "=" Command is issued next, SMCC will complete the current move.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

When this command is given from control panel ("Abort"), the program counter is reset to zero, so the program will resume from the start, not from where it left off. Deceleration time for "q" Command is set by i08 for both programmed and jog moves.

* **J(X/Y)** Jog Positive

Jog Positive at a rate specified by i06 and i31 [i51] (To stop jogging use another "J" or "Q".) Jog rate is i06 x (i31[i51]/256) x (%/65498)

NOTE: For the "J", "j", and "O" Commands the following applies:

JX, jX, OX will affect the X axis only.
J, j, O will also affect the X axis only.
JY, jY, OY will affect the Y axis only.
JJY, JjY, jJY, jjY, OOY will affect both X and Y axes.

* **j(X/Y)** Jog Negative

Jog Negative at a rate specified by i06 and i31[i51]. (To stop jogging use another "j" or "Q".) Jog rate is i06 x (i31[i51]/256) x (%/65498).

O(X/Y)n Absolute Output, Open-loop Control

The "O" Command is used to cause the SMCC to output a DC voltage of the specified value on its velocity error output to the amplifier.

The n has a range of 0 to +/- 255 where 0 produces zero volts out (+/-50 mV of offset) and 255 produces 10.0 (+/-1) volts out. The "O" Command automatically disables the servo loop and is very useful for test and calibration of the amplifier velocity or current loop.

See the note under "J" Command for axis specification.

%N,n Feedrate % Override (% Feed Pot - See also i14 and i16)

N: Range (0 - 65535), Default (value of i07, usually 65498)
n: Range (0 - 65535), Default (64)

The "%" Command is used to override the default time base number, and so to change the speed of execution of moves (feedrate %). The power-up or reset value of N is i07 (which is set at the factory to 65498).

The equivalent buffered commands are "POT N" and "SLEW n".

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The precise definition of the function of the "%" number is to control the time increment dT used to generate the next command trajectory position in the equation

$$P_{new} = P_{old} + (Vel * dT).$$

By changing the value of dT we can change how fast a move is executed (on all axes synchronously) without changing the program or altering the actual path of the trajectory.

The value of dT is determined by the equation:

$$dT \text{ (in microseconds)} = \frac{65536}{N} \times [2 \times (i05 + 1)]$$

with the default values (i05 = 487, N = 65498), dT is 976.566 microseconds or 1/1024 second (what we refer to as a "binary millisecond"). At these settings, values specified in a program for "T" and "t" define the actual execution time for a move.

If the value of N is changed (by the on-line "% N, n" Command, by the buffered "POT n" Command, by the analog input voltage with i14 = 1, or by the handwheel input frequency with i16 = 2 or 3), the speed will change proportionately (and time values dT, t, and T will change in inverse proportion). Some helpful equations are:

$$\text{Velocity (cts/sec)} = i06 \times \frac{F}{1024} \times \frac{N}{65498}$$

$$dT \text{ (microseconds)} = \frac{1{,}000{,}000}{1024} \times \frac{65498}{N}$$

$$T \text{ or } t \text{ (seconds)} = \frac{1}{1024} \times \frac{65498}{N}$$

(All these equations assume i05 = 487, for a servo cycle of 976 microseconds.)

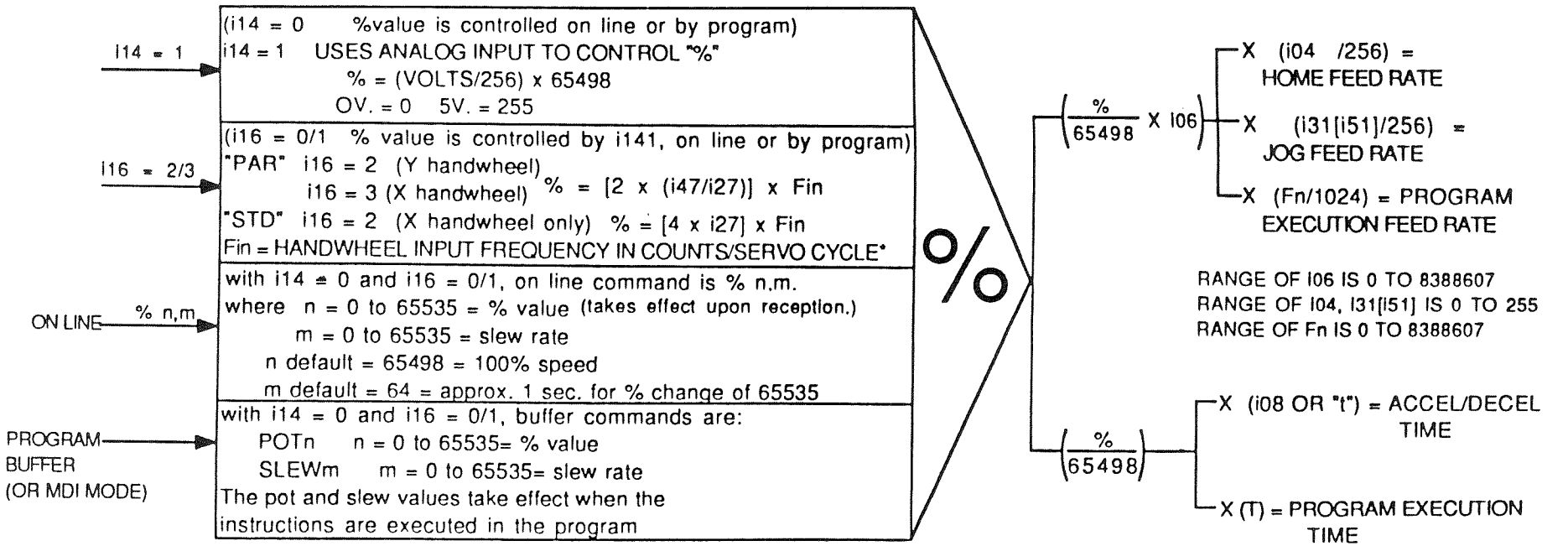
The "n" value controls the rate of change in N. It is the amount added to the current value of N each servo cycle until the new value is reached. With the default value of 64, it takes one second* to change over the full range of N. At n = 65535, any change is done in a single servo cycle. At n = 1, it takes 64 seconds to change over the full range of N.

If n is not specified, the last-specified value of n in a "%" or "SLEW" Command is used (or the default value of 64 if a value has never been specified).

If N is not specified, SMCC regards this as a Query Command and returns the present value of N.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

"%", FEEDRATE AND TIME CONTROL



NOTE: "DWELL" IS NOT AFFECTED BY % VALUE IN "PAR" BUT IN "STD" IT IS AFFECTED DUE TO DECEL TIME "t" BEING PART OF DWELL.

* ONE SERVO CYCLE IS 976 MICROSECONDS, AND IS REFERED TO AS A "BINARY MILLISECOND."

1) "SLEW" CONTROLS THE RATE OF CHANGE OF "%" AND SPECIFIES THE VALUE TO BE ADDED OR SUBTRACTED FROM "%" EACH SERVO CYCLE DURING THE CHANGE, UNTIL THE NEW SPECIFIED "%" VALUE IS REACHED.
2) A % VALUE OF 65536 IS THE SAME AS A % VALUE OF 0. IN I16 2/3 MODE, TAKE CARE THE Fin (INPUT FREQUENCY) DOES NOT CAUSE A ROLL-OVER OF THE % VALUE ABOVE 65535.
   FOR PARABOLIC, %= (65536 x Fin)/[(I27 x 32768)/I47] = [2 x (I47/I27)] x FIn
   FOR STANDARD, %= (65536 x Fin)/[32768/I27] = [4 x I27] x Fin
3) CHANGING THE % VALUE DURING A MOTION, WILL CHANGE VELOCITY BUT WILL MAINTAIN THE PROGRAMMED POSITION PATH, EVEN DURING INTERPOLATION.
4) THE ORDER OF PRECEDENCE FOR % CONTROL IS: I16 2/3, I14, ON LINE, PROGRAM BUFFER. IN I16 2/3 OR I14 1, "H" FEED HOLD WILL NOT FUNCTION. IN I16 2/3 THE "k" (KILL SERVO), "O" (OPEN LOOP) OR THE OCCURRENCE OF A SOFT FOLLOWING ERROR (I00 = -n) WILL AUTOMATICALLY SET I16 TO 0.
5) IT IS POSSIBLE TO CHANGE FROM INTERNAL TO EXTERNAL TIME BASE AND VICE VERSA DURING PROGRAM EXECUTION BY CHANGING I16 FROM (TO) 0/1 TO (FROM) 2/3 BY USING ON LINE, PROGRAM BUFFER, MDI OR PLC GENERATED COMMANDS. IF POSSIBLE, AVOID LARGE VELOCITY CHANGE DURING TRANSITION.
6) THE RANGE OF Fin (INPUT FREQUENCY) IS 0 TO 1000 COUNTS/MILLISECOND AND MAY BE EITHER AN A/B QUADRATURE OR PULSE/DIRECTION INPUT. USE AS HIGH A NUMBER AS POSSIBLE, TO AVOID VELOCITY GRANULARITY.

(PERCENT)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If i14 = 1, any % commands given are overridden by the analog input voltage, where 0 volts gives a % value (N) of 0, and 5 volts gives a % value of 65535. (See i14 definition for more details).

If i16 = 2 or 3, any % commands given are overridden by the handwheel input frequency, where the conversion from frequency to % value (N) is controlled by i27 (and i47 in the Parabolic SMCC). (See i16 definition for more details.)

Program Control Commands (Non-Buffer)

---

bnn Goto Beginning
Go to beginning of subroutine nn or main program if nn not specified ("b" only). nn is 00 to 63.

PC=v This command allows the user to control the value of the program counter, thus making it possible to execute any program line in the SMCC by first setting the PC value to the desired line number in the buffer. The range of v is 0 to 9999 and must be between 0 and the last line of the subroutines.

l Learn and Store X and Y Position
The "l" Command may be used to learn and store positions directly into the motion buffer. The SMCC may be used to go to the desired position by jogging, using the handwheel, or, if the "k" Command is used, by manually moving the motors.

If the "Program Buffer" is not full, each time an "l" Command is issued the current X or (X and Y) position is saved in the buffer. Commands such as feedrates, move times, and dwells may be sent to the SMCC at any time and will be stored in the next available buffer location.

If the "Program Buffer" is full, issuing the "l" Command causes the current position to overwrite the buffer's memory location. If a program is being executed and the SMCC is stopped at a certain position by using jog or handwheel, the position may be altered and then learned. The new learned position will now be in the buffer and next time the program is executed, the SMCC will go to the new learned position.

It is also possible, by using the PLC program, to cause one of the SMCC's inputs to be used as a "learn" input and to have the PLC generate a complete subroutine or other command each time "learn" is commanded.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

zn Purge Buffer and Reserve n Steps For Program

If n is not specified, purge main program only. If n is specified, purge entire buffer (main program, subroutines, and PLC), and reserve n lines for the main program. n is 1 to 1790. The rest of available memory can then be used to store labels (subroutines) 00 to 63 and PLC (Background Programmable Logic Controller) programs.

Examples: 1) "z50" clears the entire program buffer including the subroutines and reserves 50 lines for the new program. The PLC program is not affected. To clear PLC programs use "PUR" Command.

2) "z" clears the existing program only and retains the subroutines and the PLC program. The buffer size remains the same as last specified. Note that the buffer size can be changed by using the MAP Command.

rA,v Read Memory Data

Read v bytes of data starting from address A (Hex value). The data is sent back in a continuous string and is in hexadecimal format where two characters represent one byte.

Ranges: A (0000 - ffff) hexadecimal
v (1 - 9) If v is not specified, one byte will be sent.

Example: r f f b 8, 2 reads the value stored at address f f b 8 and f f b 9. Please note that lower-case characters a to f are used.

dA,v Read Data and Display With Format

Range: A (0000 - ffff) hexadecimal, v(1 - 65535) decimal

Read v bytes of data starting from address A (hex value) v = 1 to 65535, specified in decimal. The data is sent back in a form similar to the Intel Hex format. The base address is returned (in hex), followed by up to 16 bytes of data (in hex), separated by spaces. If more than 16 bytes are requested, more lines are returned, each starting with the base address for the line, followed by the data in the same format as the first line.

wA,d Write Memory Data

Range: A (0000 - ffff) hexadecimal, d (00 - ff) hexadecimal

Write a byte of data d to address A (hex value). Please note that d is in hexadecimal format.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Data can be written to consecutive addresses starting with the address specified with a single "w" Command by adding more data bytes separated by columns. The command line must not exceed 52 characters.

s Save Program and Parameters in Permanent Memory

The "s" Command is used to save programs and parameters in the SMCC's EAROM permanent memory. There must be a program stored in the SMCC's buffer before the "s" Command is accepted by the SMCC (the shortest acceptable program is an "END" statement). On power-up or reset, the parameters and programs in EAROM are loaded into RAM for active use (on cards without Option 5 - see below). The EAROM is not read at any other time.

With the basic 512 x 8 EAROM, the Standard SMCC stores the I, P, C, and M parameters plus six program lines, and the Parabolic SMCC stores the I parameters plus 80 program lines. With the addition of the 8K x 8 EAROM (Option 2), both Standard and Parabolic SMCC store the I, P, C, and M parameters plus 1790 program lines.

If the Option 5 (32K x 8 battery-backed RAM) is used in place of Option 2, there is no difference between permanent and active memory, so the "s" Command is meaningless.

innv Set Value of Parameter Inn to Value v

i parameters may be sent to SMCC at any time and will take effect immediately upon reception. The equivalent buffer instruction is Innv. The legal range of v is dependent on the value of nn. Please note that if inn is sent to SMCC without a value v, the SMCC will send back the value stored for the parameter inn. Also, if an ix is sent to SMCC, it will send back the current value of all the parameters it currently has.

Cmm = A Define Cmm As Address A (creates look-up table)

The equivalent buffer instruction is: SET Cmm = A.
Once an address is defined as a Cmm it can subsequently be pointed at by using Cmm.

Range: mm = 00 to 63 (decimal)
A = 0000 to ffff (hex)

Mnn = Cmm.b Define M Function Bit at Address Cmm at Bit Position b

The equivalent buffer instruction is: SET Mnn = Cmm.b.
To control the defined Mnn bit, use SET nn to turn on and RESET nn to turn off the bit in the program buffer.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Range: b = 0 to 7
nn = 00 to 63 (decimal)
mm = 00 to 31 (decimal)

NOTE: See SET Mnn and SET Cnn Commands in Buffer Instruction Section for M and C function address assignments.

Mnn = 0/1 Sets (turns on) or Resets (turns off) the Specified M Function Bit (nn has a range of 0 to 63)

Mnn = 0 Resets the specified bit
Mnn = 1 Sets the specified bit

Range: nn = 00 to 63 (decimal)

Pn = V Sets Value of a Variable Parameter
Range: V = -2 ^ 31 to +2 ^ 31 -1 (decimal)
n = 1 to 15 (decimal)

The equivalent buffer instruction is SET Pnn = V. RESETP, clears all parameters and sets them to zero.

The 15 variable parameters may be used with any buffered command noted by code "d" as a value to be used by that command.

Example: X(Pn) represents an X axis move as determined by the value of Pn.

Status Query Parameters (Non - Buffer)

---

inn Send current value of Stored Initialization/Setup parameter specified by nn.

Range: nn = 00 to 60 (Standard SMCC)
= 00 to 80 (Parabolic SMCC)

innH Send the maximum allowable value for parameter inn.

innL Send the minimum allowable value for parameter inn.

innD Send the factory-set default value for parameter inn.

ix Send all Initialization/Setup parameter values

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Control V Freeze Position Command

The SMCC accepts this command regardless of whether it is addressed or not and immediately freezes its current position in a register, enabling the user to freeze the positions on a multiple-axis system simultaneously and then read the position values of all the axes one at a time (using 'p') without time skew errors.

f Send Following Error

Following error is the difference between commanded and actual positions. The value(s) returned will be scaled by the i38 factor (for the X axis) and i58 (for the Y axis).

p Send Current Position

This returns the actual position of the active axes relative to the last zero setting, reset or power-on position. The value(s) returned will be scaled by the i38 factor (for the X axis) and i58 (for the Y axis).

v Send Current Velocity

This returns the actual velocity of the active axes. The values returned will be scaled by the i38 factor (for the X axis), i58 (for the Y axis), and i17 (affecting both axes).

NOTE: The f, p, and v Commands will send data for X axis only or X and Y axes simultaneously, if the SMCC is configured for two axes.

? Send Status and Error codes

Upon receipt of the "?" Command the Standard SMCC sends four bytes (eight digits) of data in hex format (00 to ff) to the host computer. See Parabolic Section for description eight bytes (sixteen digits) sent by the Parabolic SMCC (page 181).

1st byte BITS	.0	HOME #2 COMPLETE (MOMENTARY)
	.1	HOME #1 COMPLETE (MOMENTARY)
	.2	BLOCK READY
	.3	BLOCK REQUEST
	.4	CIRCLE IN PROGRESS
	.5	HOLD IN PROGRESS
	.6	RUNNING (ANY TYPE OF MOVE)
	.7	AUTO DIVIDE IN PROGRESS
2nd byte BITS +1	.0	TIME/FEEDRATE MODE (TIME = 1)
	.1	SINGLE STEP
	.2	P1.3 (BUFFER FULL)
	.3	CIRCLE > 180
	.4	CW/CCW CIRCLE (CCW = 1)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

	.5	CIRCLE/LINEAR (CIRCLE = 1)	
	.6	LIMIT SWITCH SET	
	.7	NOT READY FOR MOVE	
3rd byte SBITS1	.0	JOG POSITIVE	AXIS # 1 (X AXIS)
	.1	JOG NEGATIVE	
	.2	OPEN DAC MODE	
	.3	I2T (LOW TRUE)	
	.4	FOLLOWING ERROR	(i10 EXCEEDED)
	.5	NOT STOPPED	
	.6	NOT IN POSITION	(MOVING OR OPEN LOOP)
	.7	VELOCITY IS ZERO	(COMMANDED VELOCITY)
4th byte SBITS2	.0	JOG POSITIVE	AXIS # 2 (Y AXIS)
	.1	JOG NEGATIVE	
	.2	OPEN DAC MODE	
	.3	I2T (LOW TRUE)	
	.4	FOLLOWING ERROR	(i10 EXCEEDED)
	.5	NOT STOPPED	
	.6	NOT IN POSITION	(MOVING OR OPEN LOOP)
	.7	VELOCITY IS ZERO	(COMMANDED VELOCITY)

Example:

On sending a '?' to a Standard SMCC, the card returns the ASCII string '00408888'. This is a hexadecimal representation of the four status bytes (two characters per byte) starting with Byte 1. The first pair of characters is '00', so all of the status bits in Byte 1 are 0 (clear -- not set). The second pair of characters is '40'. This is a hex value that can be broken into binary as '0100 0000'. These bits correspond to the individual status bits of Byte 2 starting with Bit 7 and ending with Bit 0. In this case, only Bit 6 (limit switch set) is set. The next two bytes are '88', or '1000 1000' in binary, which mean 'Velocity is Zero' and 'I2T not set' (Bits 7 and 3, respectively) for both the X and Y axes.

% Send the value of the % Command, 0 to 65535.

Bnn Send step number or line number for Label nn or size of program buffer if nn is not specified. nn is 0 to 63. Labels are used to define subroutines.

Bx Send line numbers at which all labels, 00 to 63, are located. SMCC sends the program line numbers at which L00 to L63 are located.

Cmm Send Defined Memory Location Address for Cmm. SMCC sends the address definitions for Cmm only. mm is 00 to 63.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Cx Send all defined memory location addresses for C00 to C63. SMCC sends all addresses defined for C00 to C63.

Mnn Send M function definitions. SMCC sends address of the M function followed by the bit in that address defining the M function, followed by the value of the bit. nn is 0 to 63.

Mx Send address, bit, and value for all M functions, M00 to M63. SMCC sends data for all 63 M functions, same as Mnn.

Pn Send variable parameter values.
SMCC sends value of variable parameter n, 1 to 15.

Px Send all variable parameter values for P1 to P15.
SMCC sends a list of the 15 parameters and their current values.

PC Send value of next program line to be executed. Value of program counter.

PB Send value of program buffer size.

In general, each SMCC Buffer Command occupies one "line" in the buffer, except commands which have both an X and Y move, which occupy two buffer lines, and parabolic commands, which occupy two lines per axis.

Verification of a line number and its content can be made by using one of the editing commands, REPLACE or INSERT and then listing the program. When in edit mode, the SMCC will list the program together with the assigned buffer line number for each command.

NOTE: By interrogating the SMCC with PC and PB, the host computer can calculate the amount of data left to be executed in the SMCC buffer (PB-PC), and use this information to send more buffer data to the SMCC even as old data is being executed. The "Buff-full" status line may also be used to handshake with the SMCC in this recirculating mode; however, the PB and PC Commands simplify the operation. Using this method, an infinitely long program may be executed by the SMCC without stopping.

The SMCC V3.6 or higher (ACC-9) utility provides a "DUMP" program which makes use of this capability to download and execute programs that are larger than the SMCC buffer can hold without the need for stopping program execution at any time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Program Editing Commands (Non-Buffer)

LIST n,m
List m lines of buffer program starting at line n. If neither n nor m is specified, the whole program is listed. If n is specified but not m, only one line is listed. The SMCC sends the contents of its buffer, line by line, to the host computer.

LIST n
List line n only; LIST will list entire program.

DELETE n,m
Delete m lines of buffer program starting at line n. If n,m is not specified, no action occurs. If n is specified, but not m, only one line is deleted. When a program line is deleted, the SMCC automatically moves all subsequent lines up so as to fill the gap.

INSERT n
Insert additional line(s) of program starting at line n. To exit insert modes use <CR>. When a line insertion is made, SMCC pushes all subsequent lines down in order to make room for the insertion.

REPLACE n
Overwrite program line(s) starting at line n. To exit replacement mode, use <CR> with a blank line.

NOTE: The INSERT or REPLACE Commands cause the SMCC to output program line numbers when LIST is used, thus indicating that the SMCC is in edit mode. To exit, simply use <CR>. The program may be edited even while the SMCC is running the program. This permits on-line modifications to occur, if needed.

n and m have a range of 1 to 1790 or 5630, depending on the SMCC's options.

zn
Clear the Motion Program Buffer. The "z" Command can be used only when the SMCC is stopped. Without the n it will clear the program area while leaving the labels (subroutines) intact.

If the "zn" Command is used, the entire program and the labels are cleared and n lines (0 to 1790) are allocated to program area. The rest of the available memory can then be

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

used for Labels 00 to 63 (subroutines) and PLC program as needed.

MAP n

Reallocate n steps or lines for main program area. All labeled subroutines are moved automatically to create necessary program buffer space. This command is in effect used to alter the program area size that may have been allocated by zn.

PLC n

List the entire PLC program if n is omitted. List only the specified PLC statement if n is specified. n is 1 to 256.

PURGE n

Purge (clear out, delete) the entire PLC program if n is omitted. Purge only the specified PLC statement if n is specified. n is 1 to 256. When purging a PLC statement occurs all subsequent statements are automatically renumbered.

Control Character Commands

NOTES: All <CNTRL> type commands do not require a <CR> (carriage return) to be acted on by SMCC. They are executed immediately upon reception.

All <CNTRL> characters affect all cards on-line at once, regardless of whether they are currently addressed or not.

<CNTRL B, C, D, or E> (Parabolic SMCC Only)

These control characters are used in DMA (Direct Memory Access) mode to send and receive binary data and status information.

<CNTRL B>	[binary data]	Send binary position commands
<CNTRL C>		Request binary position data
<CNTRL D>		Request binary status data
<CNTRL E>		Request binary P1 and P2 values

Refer to Parabolic Section, DMA Communications, for a detailed explanation of the operation of these commands.

<CNTRL R> (Parabolic SMCC Only)

Upon receipt of this command, any SMCC on-line sets the value of the "%" feedrate to that in the i07 parameter. The default value of i07 is 65498. This is used to set the %

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

value to its "normal" value in a more rapid way than actually sending a % Command and its value. The rate at which the card returns to a % (i07) setting is determined by the most recent SLEW value given - either the "n" in "%N,n" or the "d" in "SLEW d" (default value is 64).

<CNTRL V>

Upon receipt of this command, any SMCC on-line will immediately "freeze" the current X and Y positions and store them for later use by the SMCC. The "frozen" position is indicated on the position display unit (Accessory 12) if one is being used and will be sent in response to a "p" Position Request Command. The "p" Command not only sends the position back to the last but also unfreezes the position display. Operation of the SMCC is not affected by the Position Freeze Command. Only position data is captured at the time of <CNTRL V> command reception. If multiple SMCCs are being used, it is possible to capture the position data simultaneously on all cards by issuing the <CNTRL V> Command capturing and freezing the position data on all SMCCs. The position data can then be read one card at a time, by using "A0p", "A1p", etc.

<CNTRL M> (Carriage Return <CR>)

This is usually sent to the SMCC by the <ENTER> or <RETURN> key on the computer keyboard. It causes the SMCC to act on the non-control characters it has received since the last carriage return (the action is immediate execution or storage in the program buffer, depending on the type of command.) As a control character, it affects all SMCCs on-line, not just the one presently addressed. This is useful for sending simultaneous commands or program lines to multiple cards, since a single <CR> operates for all cards. For instance, "A0JA1J <CR>" will cause an X axis jog on both cards to start simultaneously.

<CNTRL H> (Backspace <BS>)

This is usually sent to the SMCC by the <Backspace> key. It erases the most-recently entered character in the SMCC's line buffer (the characters awaiting a carriage return to take effect on the card). As a control character, this backspace acts on all SMCCs on-line, not just the one currently addressed. In most cases, only the addressed card will have any characters in the line buffer, but if any other cards do, the last character in their line buffer will be erased also. For instance, typing in "A0X1234A1X5678 <BS> <CR>" would leave "X123" in A0's buffer and "X567" in A1's buffer.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

6. BUFFER COMMANDS

The subsequent commands can all be stored in the SMCC's buffer and then executed with a "Step" or "Run" Command. They may also be executed on-line by using the MDI mode. (See parameter i02.)

Explanatory Codes

0 These instructions use no additional buffer space.

1 All codes greater than zero require four bytes of program memory (one block). These instructions do not require an additional "Step" Command to execute.

2 All codes greater than one cause data to be sent to the servo equations. "Step" Commands will execute these instructions one at a time.

3 This code is for X-axis moves that can be followed by a Y-axis move (code=4) on the same line. In this case only one "Step" Command is required to execute both instructions with coordinated motion.

4 This code is for Y-axis moves that can be added to the end of an X-axis move (code=3) on the same line. In this case only one "Step" Command is required to execute both instructions with coordinated motion.

a These instructions will not be executed until the beginning of an instruction that sends data to the servo (code = 2, 3, or 4).

b These instructions use i38 to scale the X-axis position data.

c These instructions use i58 to scale the Y-axis position data.

d These instructions can have their data replaced by a function of parameters. These functions must be enclosed in parentheses and have one of the following forms.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

(Pi)	(-Pi*Pj)	(Pi-Pj/Pk)
(-Pi)	(Pi/Pj)	(Pi*Pj/Pk)
(Pi+Pj)	(-Pi/Pj)	(Pi+Pj*Pk/Pn)
(Pi-Pj)	(Pi+Pj*Pk)	The ranges of i,j,k and n
(Pi*Pj)	(Pi+Pj/Pk)	are 1 to 15.

e These instructions can have an IF or IFNOT added to make them conditional.

****
NOTE: All values denoted by "nn" must be a two digit entry.
****

1ae **SET nn** Set the bit defined by Mnn to a logical one. This instruction will do nothing if the bit is an input. IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

1ae **RESET nn** Set the bit defined by Mnn to a logical zero. This instruction will do nothing if the bit is an input. IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

1e **GOTO+nn** Jump forward over nn program lines. If nn is zero then the next line is executed (GOTO+00 is a useless, but not fatal statement). IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

```
Example:  GOTO +02 IF (M11 = 1)
          P1 = 1024
          GOTO +01
          P1 = 612
          X (P1 + P2)
```

1e **GOTO-nn** Jump nn program lines backward. If nn is zero then the same line is executed. (GOTO - 00 will lock up the card and should never be used, even with IF conditions.) IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

```
Example:  U1000
          GOTO -01 IF (M11 = 0)
```

1e **GOTO nn** Jump to program line with subroutine label Lnn without saving return address. The program will terminate when a RET line is encountered. IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

NOTE: Do not use GOTO commands in such a manner that an indefinite number of these commands can be executed without an

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

intervening time-based command (move or dwell). SMCC will lock up because it cannot keep track of time. For example, do NOT use GOTO-00 IF (M11 = 0) to wait for an input. Instead use: DWELL 20
GOTO -01 IF (M11 = 0).

NOTE: Most commands take one line of program space. Two axis standard moves (e.g. X0Y0) take two lines. Parabolic (e.g. X1000:5) and 'Go Until' (e.g. X1000 ^100) moves take two lines per axis.

1e **GOSUB nn** Jump to program line with label Lnn and save the return address. When a RET instruction is executed the instruction following the GOSUB instruction is executed. IF or IF NOT can be added to the end of this instruction to make its execution conditional. The range of nn is 00 to 63.

0 **IF NOT nn = b**
IF nn = b Can be added to the end of any of the first six instructions (code=e) to make them execute only if the data at address Cnn is (not) equal to b. The range of nn is 0 to 63 and for b is 0 to 255.

0 **IF NOT nn > b**
IF nn > b Can be added to the end of any of the first six instructions (code=e) to make them execute only if the data at address Cnn is (not) greater than b. The range of nn is 0 to 63 and for b is 0 to 255.

0 **IF NOT nn + b**
IF nn + b Can be added to the end of any of the first six instructions (code=e) to make them execute only if the data at address Cnn (has/does not have) at least one bit set that is also set in b. The range of nn is 0 to 63 and for b is 0 to 255.

0 **IF NOT nn * b**
IF nn * b Can be added to the end of any of the first six instructions (code=e) to make them execute only if the data at address Cnn (has/does not have) all bits set that are also set in b. The range of nn is 0 to 63 and for b is 0 to 255.

0 **IF NOT (CC1)**
IF NOT (CC1 AND CC2)
IF NOT (CC1 OR CC2)
IF (CC1)
IF (CC1 AND CC2)
IF (CC1 OR CC2)
Can be added to the end of any of the first six instructions (code=e) to make them conditional. Conditions CCn have the following possible formats.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

	Mnn = b	True if bit defined by Mnn is equal to b. The range of nn is 0 to 63 and for b is 0 or 1.
	X > Pn	True if the X position is greater than the parameter Pn. The range of n is 1 to 15.
	X < Pn	True if the X position is less than the parameter Pn. The range of n is 1 to 15.
	X >= Pn	True if the X position is greater than or equal to the parameter Pn. The range of n is 1 to 15.
	X <= Pn	True if the X position is less than or equal to the parameter Pn. The range of n is 1 to 15.
	Y > Pn	True if the Y position is greater than the parameter Pn. The range of n is 1 to 15.
	Y < Pn	True if the Y position is less than the parameter Pn. The range of n is 1 to 15.
	Y >= Pn	True if the Y position is greater than or equal to the parameter Pn. The range of n is 1 to 15.
	Y <= Pn	True if the Y position is less than or equal to the parameter Pn. The range of n is 1 to 15.
0	**Lnn**	A label that is used by the GOTO and GOSUB instructions. The address for this label is stored in a fixed table and does not use any buffer space. A label opens up the subroutine buffer. All instructions following a label are stored in the subroutine buffer until a RETURN instruction is entered. The range of nn is 00 to 63.
1	**START**	Start continuous execution of program lines until a STOP or END is executed. One "Step" Command will execute all instructions between the START instruction and the next STOP or END instruction. START is not needed if the program is just going to be run and not stepped.
2	**STOP**	Stop continuous execution of the program.
2	**END**	Stop continuous execution of the program and move to the beginning of the main program.
1	**RETURN**	Return to the line following the last GOSUB that was executed. If no GOSUB was executed then an END statement is executed.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1d **FOR d** Start a loop and repeat it d times. If d is zero then never stop the loop. The range of d is 0 to 32767.

1 **NEXT** End of loop. If the loop has not been repeated d times then the instruction following the FOR is executed, otherwise the instruction following the NEXT is executed.

1 **FND** This instruction must be executed before jumping out of the middle of a FOR/NEXT loop to clean up the loop count and starting address. After executing this instruction the present FOR/NEXT loop is closed. If a NEXT instruction is executed after the FND, then any previous loop that is opened will be looped to. Hence care must be taken to skip the NEXT at the end of the loop that has been closed using the FND instruction. The FND instruction cannot be used to close a loop if it is within a subroutine that was called from inside the loop.

1 **CASE mm,nv** This instruction reads the location pointed to by Cmm. It then converts the data based on the letter "v" and jumps to a relative location in the table of n instructions following the CASE instruction. If the value is zero, then the next program line is executed, followed by the statement n + 1 lines down from the CASE statement. If the value is one, the second line past the CASE statement is executed, followed by the statement n + 1 lines down from the CASE statement, and so on. If the value is greater than or equal to n, then the next n instructions are skipped. The instructions in the table cannot be a CASE, RETURN, or END instruction. Any other instruction can be used, including instructions that require more than one block of memory. The range of mm is 00 to 63. The range of n depends on the conversion used and is as follows:

D - Two BCD digits. The range is 0 to 99.
H - High digit. The range is 0 to 15.
L - Low digit. The range is 0 to 15.
B - Binary value. The range is 0 to 255.
d - Same as 'D' with complemented data.
h - Same as 'H' with complemented data.
l - Same as 'L' with complemented data.
b - Same as 'B' with complemented data.

NOTE: Complemented data means that when the available data is read at the specified address, it is first complemented (inverted) before being used to jump to the specified line following the Case Command. This, for example, allows the use of thumbwheel

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

inputs at the parallel port which provide either "true" or "false" outputs, are "normally open" or "normally closed", without changing the basic Case Command written to read the thumbwheels.

1d **DIVIDE d** Divide the next move into d equal parts. If the length of the move is not divisible by d, the remainder is carried along so that the total move distance will be correct. If the move following the DIVIDE instruction is a circle, then the circle will be divided into d equal parts. However, the motion between the equal parts will be linear and not circular. The range of d is 1 to 65535.

NOTE: If the Run "R" Command is used to execute the "Divide" Command, the divide function is defeated since motion will occur continuously and will not stop at the division points. Therefore, use the Step "S" Command or input when executing the "Divide" Command.

1d **F d** Set feedrate based on the value of i06. If d is 1024, then feedrate is i06 counts/second. For motion in two dimensions the feedrate for each axis is based on the vector angle of the move. The minimum time for any move is the acceleration time set by the t instruction or i08. The feedrate is effective until a new F or T instruction is executed. The range of d is 1 to 8388607.

1d **T d** Set move time to d/1024 seconds. The move time is from the start of motion to the start of deceleration. The minimum time for any move is the acceleration time set by the t instruction. The move time is effective until a new F or T instruction is executed. The range of d is 1 to 8388607.

1d **t d** Set change-of-speed time to d/1024 seconds. The time for acceleration and deceleration is the same. This is also the minimum time for any move. Circular moves are broken up into segments of time t. This time is initialized to the value of i08 at turn-on. The range of d is 1 to 65535. On the Standard SMCC, this changes the value of the i08 parameter, and so affects future jog moves as well as programmed moves. On the Parabolic SMCC, it supersedes i08 for programmed moves, but does not change i08, and does not affect jog moves. The total time of a parabolic move segment is 't'.

2 **HOME** Go to home position. Both the X and Y axes are sent to the home position. This position can be

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

based on any combination of a home flag and an index pulse. A fixed offset is added to this position and sets the zero position for all subsequent moves. This zero position can then be changed by use of the X= or Y= instruction, or the on-line Zero Set Command. However, moves based on the home position can still be made using the AX and AY instructions.

On the Parabolic SMCC, the X and Y axes may also be individually commanded to go to home position by using the "HOME1" (X axis) or HOME2" (Y axis) Commands. On the Standard SMCC, a buffered HOME command must be for both axes (but the PLC can be made to issue "hX" or "hY" Commands).

If the "HOME" Command is preceded by a "TORQUE" Command, it is possible to do a homing cycle by running the axis into a hard stop at controlled speed (set by i04 and i06), and upon exceeding the following error limit i10, to declare that position as the home or zero position. An effort move may be made as in the normal HOME cycle.

2d **DWELL d** Set dwell time to d/1024 seconds. This instruction will stop the present motion and wait the specified time after the motion has stopped. This instruction should not be used after a parabolic move ending with a velocity other than zero as the motion will stop with no deceleration time. The range of d is 0 to 8388607. Dwell time is not affected by anything that controls the time base ("%", "POT", i16 = 2 or 3, i14 = 1).

3bd **X d** Move X axis to absolute position d. The move is based on the present zero set by the most recently executed of: the X = instruction; the on-line Zero Set Command "Z"; or a homing operation. The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863. This instruction can be followed with a Y-axis move instruction on the same line so that both axes move at the same time. If a Y-axis move is added, then two blocks of program buffer are required.

4cd **Y d** Move Y axis to absolute position d. The move is based on the present zero set by the most recently executed of: the Y = instruction; the on-line Zero Set Command "Z"; or a homing operation. The range of d depends on the scale in effect (i58). The unscaled range is -67108864 to 67108863. This instruction can be preceded by an X-axis move instruction on the same line so that both axes move at the same time. If an X-axis move is added, then two blocks of program buffer are required.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3bd **AX d** Move X axis to absolute position d relative to the last HOME or turn-on position. The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863. This instruction can be followed with a Y-axis move instruction on the same line so that both axes move at the same time. If a Y-axis move is added, then two blocks of program buffer are required.

4cd **AY d** Move Y axis to absolute position d relative to the last HOME or turn-on position. The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863. This instruction can be preceded by an X-axis move instruction on the same line so that both axes move at the same time. If an X-axis move is added, then two blocks of program buffer are required.

3bd **U d** Move X axis d distance from its present position. The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863. This instruction can be followed with a Y axis move instruction on the same line to move both axes at the same time. If a Y-axis move is added then two blocks of program buffer are required.

4cd **V d** Move Y axis d distance from its present position. The range of d depends on the scale in effect (i58). The unscaled range is -67108864 to 67108863. This instruction can be preceded by an X-axis move instruction on the same line to move both axes at the same time. If an X-axis move is added, then two blocks of program buffer are required.

1d **: d** (Parabolic SMCC Only) This instruction can be added to the end of any motion instruction (X,Y,AX,AY,U,V) to define parabolic move with ending velocity d. The value of d has the same meaning as in the "F d" Command (i06 * d/1024 cts/sec). If d is 1024, then the velocity is i06 lines per second. The range of d is -8388608 to 8388607. If :d is used in a command, F and T are inoperative for that command, and the move is executed in "t" time.

3bd **WX d** Wait the same time that would be required to move the X axis d distance from its present position. The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863. This instruction can be followed with a Y-axis move instruction on the same line. If a Y-axis move is added, then two blocks of program buffer are required.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

4cd **WY d** Wait the same time that would be required to move the Y axis d distance from its present position. The range of d depends on the scale in effect (i58). The unscaled range is -67108864 to 67108863. This instruction can be preceded by an X-axis move instruction on the same line. If an X axis move is added, then two blocks of program buffer are required.

NOTE: The WX and WY Commands are useful in multi-axis SMCC system programming when an axis on the same or another SMCC card must wait precisely for the duration of another axis motion. The WX and WY Commands will maintain correct timing even when the feedrate or % Commands are used to alter the velocity of the moving axis.

1 **WUT** (Parabolic SMCC V2.78 and newer only) Wait Until Trigger. See Software Updates section, page 414, for details.

1d **^ d** (Parabolic SMCC Only) Go Until Interrupt. This instruction can be added to the end of any motion or wait instruction (X,Y,AX,AY,U,V,WX, or WY) to define a relative move to be made after an X-axis user-flag interrupt occurs. If the interrupt does not occur before the end of the move, an error 5 is displayed and the motion stops. If a time was specified for the move, then the time will be for the total move including the move after the inter rupt. When using timed moves, care must be taken to insure that the interrupt does not occur too close to the end of the move. The minimum time for any offset move will be the acceleration time defined by the t instruction. The scale of d is always encoder counts and has a range of -8388608 to 8388607.

1d **tt d** (Parabolic SMCC V2.78 and newer only) Triggered move calculation time. See Software Updates section, page 414, for details.

0 **q...** (Parabolic SMCC V2.79 and newer only) The letter "q" on the front of a command, when given in MDI mode, will break into the current move, and execute this command (which could be another move) in its place. See Software Updates section, page 420, for details.

2d **DLY d** (Parabolic SMCC Only) Execute a delay time (no programmed move) of d/1024 seconds. This command is essentially equivalent to using the commands "T d" and "U 0", except that it does not alter the current T value. The delay time starts at the deceleration point of the previous move segment, and the t time for deceleration is included in the delay time, so the next time at zero velocity is d - t. If the delay time is shorter than the t time, it will take t time to execute, providing only the deceleration to zero velocity, with no net time at zero velocity.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The DLY Command is affected by time base changes ("%" and "POT" Commands, i14 = 1 input voltage, and i16 = 2 or 3 input frequency) just as a move would be.

The DLY Command is similar to the DWELL Command, except that the DWELL time starts after the deceleration (t) time, and DWELL is not affected

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

by time base changes. (The Standard SMCC also adds a t time between consecutive dwells.)

The wait (WX/WY) can also provide similar functionality to the DLY Command, except that the Wait Command does not specify a time directly. Rather, it specifies a distance, the F or T value in force, i06, and the % value. Its meaning is to wait for a time equivalent to a move of that distance under current conditions. The Wait Command can act on just a single axis of the card, causing the other axis to move as if it were coordinated with the one waiting.

1bd **X = d** Set the present X position to d. All X moves following this instruction are based on this new set position. (AX moves are unaffected). The range of d depends on the scale in effect (i38). The unscaled range is -67108864 to 67108863.

1cd **Y = d** Set the present Y position to d. All Y moves following this instruction are based on this new set position. (AY moves are unaffected.) The range of d depends on the scale in effect (i58). The unscaled range is -67108864 to 67108863.

1bd **RX = d** Set the present X radius to d. The range of d depends on the scale in effect (i38). The unscaled range is 0 to 67108863.

1cd **RY = d** Set the present Y radius to d. The range of d depends on the scale in effect (i58). The unscaled range is 0 to 67108863.

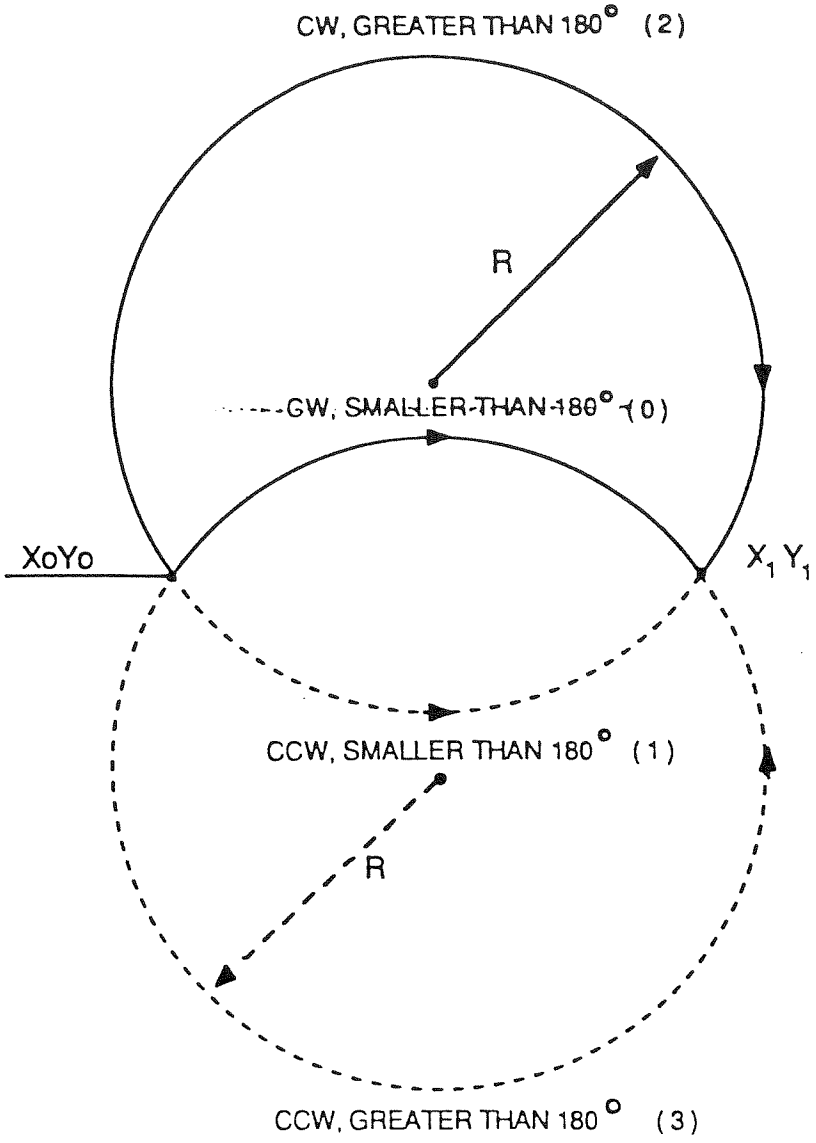
1d **R = d** Set the present X and Y radii both to d encoder counts. No scale is used and therefore d must be in encoder counts. The range of d is 0 to 8388607.

1d **C = d** (Standard SMCC only) Set offset for ellipse. The range of d is -8388608 to 8388607. This controls the angle of the ellipse by defining the Y-offset (from the center) of the maximum X-extreme of the ellipse. This offset is equal to (d * X radius/8388608). See Example 12.

1 **LINEAR** All instructions following this execute linear motion until a CIRCLE instruction is executed.

1 **CIRCLE n** All instructions following this execute circular or 3-point smoothed motion until a LINEAR instruction is executed (CIRCLE 0 to CIRCLE 3) or until an F or a T Command is executed (CIRCLE 4). The value of n determines the type of circle or 3-point smoothed motion. The 3-point smoother is in

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



PROGRAM:

CIR (0,1,2,3)	/ PLACES SMCC IN CIRCULAR AS OPPOSED TO LINEAR MODE.
R = (VALUE)	/ VALUE OF RADIUS
X (VALUE) Y (VALUE)	/ DEFINE END POINT OF CIRCLE OR DESTINATION, (X_1 Y_1).

NOTE: 1) DESTINATION-DISTANCE MUST BE SMALLER THAN 2xR
OR "ERROR 4" WILL BE GENERATED AND THE MOVE WILL
EXECUTE LINEARLY
2) IF A FULL 360° CIRCLE IS TO BE MADE, USE TWO 180° CIRCLES.

CIRCULAR INTERPOLATION

(CIRCULAR INTERPOLATION)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Parabolic only. Both an X and Y radius must be defined before executing any move instructions.

They can be defined before or after the CIRCLE instruction. The range of n is 0 to 4. The meaning of n is as follows:

n = 0 CW circle less than or equal to 180°.
n = 1 CCW circle less than or equal to 180°.
n = 2 CW circle greater than or equal to 180°.
n = 3 CCW circle greater than or equal to 180°.
n = 4 3-point smoother (S-Curve) mode (Parabolic SMCC only).

1a **TORQUE** (Parabolic SMCC Only) This puts SMCC in torque-limited mode (see page 402 for details). If it is followed by a "HOME" Command, then the home cycle changes to a torque-limited home. The home move continues until the following error on each axis exceeds i10. The normal offset move is then made and the new position is set to zero. This can be used to do a homing move up against the hard stops of your system (no home input).

If the Torque Command is followed by a Move Until Command (any move followed by a ^d), then the offset move is made after the following error on either axis is greater than i10.

If the Torque Command is followed by any other move, then the command position for that move will not be relaxed if the following error ever is greater than i10.

1ad **POT d** Sets percent feedrate to d. This instruction has the same effect as the on-line % Command. Refer to the % Command definition for details and explanation. The range of d is 0 to 65535.

1ad **SLEW d** Sets percent feed pot slew rate to d. When the percent feedrate is changed, this value is added to the current feedrate each servo cycle until it reaches the new value. The range of d is 0 to 65535.

1ad **Inn d** Sets inn to the value d. The range of nn is 00 to 60 (Standard SMCC) and 00 to 63 (Parabolic SMCC); for d it is -8388608 to 8388608. Note that the highest numbered parameters (64 through 80) on the Parabolic SMCC cannot be set using this command.

1a **m a,n** Writes the byte n to memory location a. Both a and n are hex numbers and require lower case letters a-f. The range of a is 0000 to ffff and for n is 00 to ff. This command is executed only

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

with an instruction that sends data to the servo.

1 **mm a,n** This is the same as the "m" instruction except that it does not require an instruction that sends data to the servo to execute.

1a **n a,n** Logical AND the byte n to memory location a. Both a and n are hex numbers and require lower-case letters a-f. The range of a is 0000 to ffff and for n is 00 to ff.

1a **u a,n** Logical OR the byte n to memory location a. Both a and n are hex numbers and require lower-case letters a-f. The range of a is 0000 to ffff and for n is 00 to ff.

1a **x a,n** Logical XOR the byte n to memory location a. Both a and n are hex numbers and require lower-case letters a-f. The range of a is 0000 to ffff and for n is 00 to ff.

1d **SET Pn = d** Set the variable Pn to the value d. The range of n is 1 to 15 and for d is -8388608 to 8388607 (0 - 2 ^ 23 to +2 ^ 23 - 1). While the range of values a P parameter can hold is larger than this (-2 ^ 31 to +2 ^ 31 - 1), the values that can be assigned directly as a number using this statement are limited to the smaller range. P parameters are not set synchronously with the beginning or end of any move. They are set as calculations are done for the next move block. These start right at the beginning of the current move in the Standard SMCC, or (i12) msec. before the end of the current move in the Parabolic SMCC.

1 **SET Pn = HX**
1 **SET Pn = HY**
1 **SET Pn = AX**
1 **SET Pn = AY**

These instructions set the variable Pn to home captured position (HX or HY) or to the ending position of the present move (AX or AY) based on the last home zero position. The units are encoder counts.

1 **RESET P** Sets all 15 parameters to zero.

1 **SET Cnn = a** This instruction defines Cnn to be the address a. The range of nn is 00 to 63 and for a is 0 to ffff.

1 **SET Mnn = Cmm.b** This instruction defines Mnn to be the bit position b at the address defined by Cmm. The range of nn is 00 to 63 and for mm is 00 to 31 and for b is 0 to 7.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

NOTE: SMCC makes default assignments all the M functions to "C" addresses. The assignments are different for the Standard and Parabolic SMCC and are shown on the succeeding lists. To turn on a M function (set it to a logical one state) use SETnn. To turn it off use RESnn.

PLC INSTRUCTIONS (in Buffer)

The PLC (Programmable Logic Controller) instructions of the SMCC are for the purpose of allowing the execution of instruction in the background loop of the SMCC's normal instruction execution sequence. The SMCC normally has a 500 microsecond servo loop (per axis) of which approximately 300 are spent closing the servo loop and the remaining 200 are used for general purpose background tasks such as communications, display update, velocity calculations, and PLC tasks.

The advantage of the PLC program is that it has an IF-THEN-ELSE form and can perform logical and arithmetic operations on I/O and various registers and then alter or modify a normal program's execution on an asynchronous and continuous basis, without affecting the normal program execution functions of the SMCC. Unlike motion programs which are always executed sequentially and step by step, the PLC programs are active at all times (unless disabled by command) even if a motion program is not being executed, and are executed as rapidly as leftover servo background time allows.

The execution rate of the PLC instructions is variable and depends on the number of PLC instructions, limited by memory size only, which are being used and also on the available background time and communication work load. For example the following PLC program, if it is the only one:

```
IF M01 = 0
THEN M01 = 1
ELSE M01 = 0
END
```

causes the M01 output to toggle at its maximum rate which turns out to be at 3 to 4 msec. toggle rate.

In general it is safe to assume that PLC programs will take from 3 to 5 milliseconds to execute. However, if time is critical the only positive way to determine the resulting execution rate for a particular PLC function is to measure it. PLC programs may be written at any place in the program flow so long as they start with IF (which automatically places the SMCC in the PLC programming mode) and are terminated by END (which puts the SMCC back in normal mode). Generally, it is best to place the PLC

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

programs at the end of the standard motion program so as to be able to keep track of them easily. All PLC programs are placed in a separate area of the program buffer and are not part of the normal motion program flow. They are in fact active at all times even if there is no motion program resident in the buffer and can all be turned on and off by i60 (Standard SMCC) or i62 (Parabolic SMCC). The entire PLC program is automatically disabled during the loading of a PLC branch or branches, regardless of the state of i60/i62. This occurs from the time SMCC receives an 'IF' Command until it receives an 'END' Command. No specific controls except for listing and deletion are provided for individual PLC statements or branches. However, this can easily be accomplished by using M function or other bits to control each PLC branch individually.

The number of PLC branches that may be written is 255.

Note: It is not a good idea to save PLC programs in permanent memory until they have been thoroughly debugged, particularly if they involve sending messages or issuing commands. It is easy for an improperly written PLC program to tie up operation of the card, but as long as it has not been saved to permanent memory, it can be eliminated by resetting the card or cycling power

---

DESCRIPTION OF PLC PROGRAM STATEMENTS

PLC n

Causes the SMCC to send a listing of all PLC instructions stored in buffer if n is omitted. If n is 1 to 255, only the specified PLC program will be listed. This instruction may be used only when the SMCC is outside the PLC program mode.

IF/ANDIF/THENIF/ELSEIF/AND/OR

IF: The IF statement places SMCC automatically in PLC programming mode and defines the condition for PLC action. The IF statement may be made at any place in the program but it is best to write all PLC programs immediately following the motion program.

The conditions are of the form:

1. **Mnn = 0/1** where nn is 0 to 63.
2. **Cnn = d** where d is 0 to 255, where nn is 0 to 63.
3. **AX/AY > < DATA** (X or Y absolute position)
4. **Cnn,m > < DATA** where Cnn (nn = 00 to 63) defines a address, and m (1 to 4) is the number of bytes following the address, so data is 8 to 32 bits. Note that if m is omitted, it is interpreted as Cnn,1, a one-byte address.
5. **VX/VY > < DATA** (X or Y velocity)
6. **FEX/FEY > < DATA** (X or Y following error)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

7. **Pn > < DATA**
8. **Tn = 0** True if Timer n has counted down all the way (n is 0 to 3). See detailed description of timers below.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DATA, is a decimal number in all cases, +/-[(2 EXP 31) -1] and may be substituted by Pn variable parameters. See "Variable Parameters" section below for the allowable formats.

When the specified condition is true, the PLC program proceeds to execute the "THEN" statements. If the condition is false, the PLC program executes the "ELSE" statements, if any.

NOTE: The above PLC IF conditions should NOT be applied to IF conditions used in the program buffer for the GOTO, GOSUB, etc., Commands.

Some examples are:

```
IF M01 = 1
AND AX > 10000
AND FEX<1000
THEN IF P2>1
AND C14<126
AND C52,4>69555421
AND VX>2000
AND FEY<(P2*P3/P4)
THEN P5 = 65
AND I01 = 1500

ELSE I01 = 5000    ;Alternate action for
                    THEN IF
ELSE M17 = 1       ;Alternate action for IF
END                ;End of PLC program.
```

The ">" and "<" are interpreted as "greater than" and "smaller than". "=" or "equal to" may not be used. To check for equality use, for example:

```
IF P6>4
AND P6<6
THEN.....
```

Above statement can be interpreted as:

```
IF P6 = 5      ;Illegal command, not allowed
               ;in PLC.
THEN.....
```

ANDIF: The ANDIF statement may be used only when inside a THEN or ELSE statement and will create additional conditions for action. All conditions with IF may be used with ANDIF.

THENIF: The THENIF Command allows nested conditions to be specified in PLC programs by use of conditional sub-branches. The THENIF specifies a new set of conditions after the previous IF (or ANDIF, THENIF, ELSEIF) conditions

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

are satisfied. The maximum number of successive levels that may be used is 254. The commands that can be used with the THENIF Command are the same as for the IF Command.

*See attached figure of PLC branch and sub-branch execution sequence.

ELSEIF: The ELSEIF statement begins a PLC sub-branch and may only be used when inside a THEN statement in order to add nested additional conditions before action is taken. The ELSEIF specifies a new set of conditions after the previous IF (or ANDIF, THENIF, ELSEIF) conditions are satisfied. The commands that can be used with ELSEIF are the same as for the IF statement.

Every IF or ANDIF requires a THEN or THENIF statement and an ELSE or ELSEIF to terminate or close the PLC branch or sub-Branch. If none of the ELSE statements require an action, the ELSEs may be omitted and SMCC will add them automatically. However, if one of the ELSE's among several requires an action, then all previous ELSE's must be entered in order to properly execute the PLC program.

Also note that every IF, ANDIF, THENIF, or ELSEIF statement requires a THEN and ELSE statement for completion.

AND: AND statements may be used with IF, ANDIF, THENIF, ELSEIF, and OR statements to create multiple logical conditions. In the PLC program lines:

```
IF M01 = 1
AND M02 = 1
THEN...
```

The THEN statement will be executed only if both of the conditions are true.

AND statements coupled with IF statements can take any of the conditions that an IF statement can.

If an AND statement is used in conjunction with a THEN or an ELSE statement, it carries an action with it,not a condition. See the THEN/AND section for more details.

OR: OR statements may be used with IF and AND statements to create greater varieties of multiple logical conditions. As the name implies, these perform a logical OR of conditions. For instance, with the PLC program lines:

```
IF M01 = 1
OR M02 = 1
THEN...
```

the THEN statement will execute if either of the two conditions is true,

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

ANDing is always done before ORing, which means that ORs work on blocks of ANDed conditions. For example, the following PLC program lines:

```
IF M01 = 1
AND M02 = 1
OR M03 = 1
AND M04 = 1
THEN...
```

are interpreted as: IF [(M01=1 AND M02=1) OR (M03=1 AND M04=1)]. The THEN statement will be executed if the first two conditions are both true of if the last two conditions are both true.

OR statements can take any of the conditions that an IF statement can.

THEN/AND

THEN: Defines the action to be taken by SMCC when "IF" condition is satisfied.

The "THEN" Commands take the form of actions to be taken as follows:

1. **Mnn = 0/1** Set M function to a 1 or 0.
2. **Cnn,m = DATA** Set Memory location Cnn where m specifies 1 to 4 bytes, to a value. Note that if m is omitted, it is interpreted as Cnn,1, a one-byte address.
3. **Cnn,m = Cnn +/- DATA** Add DATA to value of memory location. Do not use Cnn,m on right side of equation just use Cnn.
4. **Inn = DATA** Set servo parameter to a value.
5. **Pn = Cnn,m** Set a variable parameter to value in memory or I/O port.
6. **Pn = DATA** Set a variable parameter to DATA value.
7. **Pm = Pm +/- DATA** Add a DATA value to variable parameter. Note that same P variable must be used on both sides.
8. **SEND "Message"** Specifies that PLC should send a message to the host computer via the RS232 or parallel port, whichever is being used by the SMCC. Any ASCII character except the "Space" character may be included in the message, which must be enclosed by (") quotation characters and cannot exceed 40 characters in length. A typical message would be: ANDSEND "HELLO_1_2". NOTE: Make sure that one of the conditions that causes the message to be sent is immediately made false so that the PLC will not attempt to send the message again on the next cycle (which will garble the message and completely tie up the communications.)
9. **"Command"** The (") quotation marks without the SEND preceding them tells the PLC to send a command to the SMCC itself. Any command that SMCC can receive from the host computer may be sent via the PLC to the SMCC internally.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Typical commands would be: "q" (quit) or "i005000" (change following error limit). NOTE: Make sure that one of the conditions that causes the command to be issued is immediately set false so the PLC will not try to issue the command again on the next cycle.

10. **Tn = DATA** Assign the value in DATA to Timer n, where n has a range of 0 to 3 and DATA has a range of 0 - 65535.

See detailed description of timers below.

11. **Pn = ATAN (VALUE)**
    Assign the arctangent of value to parameter Pn (n = 0 to 15). (VALUE) must be one of the following:
    (VX/VY) or (VY/VX) or (HX/HY) or (HY/HX) or
    (AX/AY) or (AY/AX) or (FEX/FEY) or (FEY/FEX) See detailed description of the arctangent function below.
12. **Pn = VALUE**
    Assign VALUE to parameter Pn (n = 15). VALUE can take any of the following forms:
    AX, AY: Present absolute position referenced to the most recent power-on, reset, or home position.
    HX, HY: Most recently captured position referenced to previous power-on, reset, or home position.
    FEX, FEY: Present following error.
    VX, VY: Present veloctiy.

AND: AND statements can be used in conjunction with a THEN statement to define multiple actions to be taken when the IF condition is satisfied. They can take any of the actions that a THEN statement can.

An AND statement coupled with an IF-type statement carries a condition with it, not an action. See the IF section for possible conditions.

Some typical PLC statements using THEN and AND with various actions are:

```
IF M01   = 0       ;If M01 and M02 are true, then
AND M02 = 0        ;
THEN I02 = 1       ;Sets i02 parameter to a 1
AND SEND "SMCC__IS__NOW__IN__MDI__MODE"
                   ;this message is sent to the host and
                   ;printed on the screen.
AND "i0150"        ;Note that i parameters may also be
                   ;changed by sending an on-line
                   ;command, as well as using "I",
                   ;as seen above.
AND C15,4 = C15 + 15000
ELSE......
```

Like the "DATA" values in the IF statements, the "DATA" values in the THEN statement may all be replaced by a variable parameter function. For example, instead of:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

THEN/AND C23,2 = 12543
The following statement can be made:

THEN/AND C23,2 = (P3-P2/P4)

ELSE/AND

ELSE defines alternate action of PLC if the "IF" condition is not satisfied. The actions that can be taken are identical to the "THEN" Command. AND statements can be used in conjunction with ELSE to define multiple actions to be taken.

END

Must be used at the end of the PLC program and causes the SMCC to exit the PLC programming mode. However, if several IF statements are made, it is not necessary to write the END statement after each PLC branch. The SMCC will provide an automatic END for the previous branch when a new IF statement is encountered.

PURGE n

The PURGE Command, without a number, clears all PLC programs. SMCC must be outside PLC programming mode to accept the PURGE Command. When PUR n is used (n is 0 to 255), only the specified PLC branch is deleted. The PLC statements or branches all start with an IF statement and the numbering is in simple descending order of entry starting with one to be kept track of by the programmer. Please note that whenever a program containing PLC statements is loaded to SMCC, it is necessary to use the PUR Command to purge all existing statements in the buffer, otherwise the new PLC statements will simply be added to the existing statements. If you want to add PLC branches, simply type or send IF... and go on. The new PLC branches will be added to the existing PLC statements. Deleting a certain specific PLC branch simply causes the succeeding ones to be renumbered accordingly.

Variable Parameters

Variable parameter functions can have the following forms:

(+/-PI)
(PI +/- PJ)
(+/- PI*PJ)
(+/- PI/PJ)
(PI + PJ * PK)
(PI +/- PJ/PK)
(PI + PJ * PK/PN)
(PI * PJ/PK)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Where I, J, and K can be 1 through 15. Note that all "P" parameters, when they are used as "functions", must be enclosed in parentheses.

TIMERS

The SMCC's PLC has four timer ports T0, T1, T2 and T3. T0 and T1 are slow countdown timers, T2 and T3 are fast countdown timers. The clock for T0 and T1 is approximately 23 Hz. or a period of 43.69 milliseconds per count and the clock for the T2 and T3 is approximately 47 kHz or a period of 21.333 microseconds per count.

The format of the Timer Command is:

Tn=(DATA) where n is a number from 0 to 3 defining the timer to be used and DATA has a range from 0-65535 and is the multiplier for the period of the counter. (DATA) may be replaced by variable parameters (+/-P1), etc.

To use the timers it is necessary to set a timer value first by specifying, for example T1 = 500. This sets the T1 timer's time-out value to 500 x 43.69 = 21.845 seconds and to then check for time-out to occur by using:

```
IF T1 = 0
THEN.........
```

Note that the condition that caused the timer to be set in the first place can cause the timer to be set back to that value again every time through the PLC (every few milliseconds). This would prevent the timer from ever counting down. To prevent this, the same PLC branch that sets the timer should also undo the condition that caused it to be set (see example below).

Maximum time for T0 and T1 is approximately 47.7 hours. For T2 and T3 it is 1,398 milliseconds. An example of how to use the timers to turn on outputs M01 to M04 is shown below:

```
IF P1>0          ;SET P1 = 1 TO START CYCLE.
THEN T0=1000     ;SET ALL TIMERS TO VALUE OF 1000 CYCLES.
AND  T1=1000
AND  T2=1000
AND  T3=1000
AND  P1=0        ;ONCE SET, PREVENT SETTING AGAIN.
IF   T0=0
THEN M01=1       ;TURN ON M01 WHEN T0 HAS TIMED OUT.
IF   T1=0
THEN M02=1
IF   T2=0
THEN M03=1
IF   T3=0
THEN M04=1
ELSE
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The timer accuracy is dependent upon the PLC cycle time and can be considered to be -0, +3 milliseconds maximum.

If it becomes necessary to examine the progress of the timers before time-out has occurred, the following 2 byte memory locations may be read:

	STANDARD SMCC	PARABOLIC SMCC
T0	ff10	ff0a
T1	ff12	ff0e
T2	ff88	ff88
T3	ff8a	ff8a

The values in the listed addresses will indicate the progress of the countdown cycle.

Other PLC examples using timers to control M function and other outputs are shown at the end of this section and have more complex logic associated with them.

ATAN (ARCTANGENT)

PLC is also able to calculate the arc tangent of a given value and produce the corresponding angle. The PLC accomplishes this by using a variable parameter to store the result of the calculations. The command format is as follows:

```
Pn = ATAN (VALUE)

Where (VALUE) takes several forms:

(VALUE) = (VX/VY) or (VY/VX)     ;Commanded Velocity of X&Y
        = (HX/HY) or (HY/HX)     ;Captured Home Position of X&Y
        = (AX/AY) or (AY/AX)     ;Absolute Position of X&Y
        = (FEX/FEY)or(FEY/FEX)   ;Following Error of X&Y
```

The Pn parameter ends up with the 32-bit angular value of the arc tangent of (VALUE). The scaling factor is: 2 EXP. 30 = 90 degrees.

Therefore the angle value is: (Pn / 1,073,741,824) x 90 degrees.

A possible use of ATAN is to calculate the velocity vector of the X and Y axes on one SMCC and send the result to another SMCC so it can use the angular value as a command to orient a "Cutting Knife" on the Z axis in the proper direction, always pointing in the velocity vector direction of the X and Y axes.

For general information, please refer to Programming Examples 15, 16, and 17.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

NOTE: PLC programs cannot be edited after entry to SMCC except for selective purging. It is recommended that the SMCC V3.X IBM PC program be used to do all editing and changes to PLC programs, if needed.

Examples:

A few other PLC examples are given so as to provide the user with a better understanding of its usage and needed formats.

If the PLC is used to make intensive calculations with the P variable parameters, caution must be exercised in not overloading the background time since this may effect the SMCC's servo control operations capability. It is recommended that multiple divisions (e.g. P5/P4) be avoided since they are especially time consuming. The effects of PLC overload are exhibited by the SMCC's inability to execute more than 50 blocks (moves) per second and a tendency to "lock-up" and quit executing motions. The heavier the PLC load, the slower the SMCC becomes. In Parabolic version it is possible to increase the i12 value (Calculation Time) which then prevents the SMCC from locking up when calculating its first move. While SMCC is executing continuous blocks, it is the "t" or accel/decel time which controls the amount of minimum available time for more calculations. Normal PLC functions without extensive calculations have no appreciable effect on SMCC's operation.

The following example is nonsensical in terms of useful action. It does, however, present some good examples of Do's and Don'ts for PLC programs.

```
A0PURGE                  ;Clear PLC program
IF  AX > 123             ;If absolute position of X,  relative
                         ;to home zero is greater than 123,
THEN M06 = 1             ;Then turn on M06
ELSE M07 = 0             ;If AX is smaller than 123,
                         ;Then turn off M07
IF AY > 123              ;If absolute position Y is greater
THEN C06, 2 = 123        ;than 123, set 2 Bytes of memory
                         ;defined by C06, equal to 123.
ELSE C07, 4 = -123       ;If AY is smaller than 123, then
                         ;set 4 Bytes of memory
                         ;defined by C07 to be equal to -123.
IF VX > 123              ;If velocity of X is greater than 123
THEN C06, 4 = C06-123    ;Then reduce C06 value by 123
ELSE P07 = C06, 1        ;If VX is smaller than 123, set P07 to
                         ;value stored at C06's first byte only
IF  FEX  < 123           ;If following error of X is smaller
THEN P06 = 123           ;than 123, then set P06 to a value
                         ;of 123
ELSE P07 = P07 + 123     ;If FEX is greater than 123, then
                         ;increase value of P07 by 123.
IF P05 < 123             ;If Parameter P05 is smaller than 123,
THEN P06 = P07 + 123     ;(see note 1)
ELSE M07 = 1             ;If Parameter P05 is greater than
                         ;123, then turn on M07.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
AND M08 = 0                ;Also turn on M08
OR M09 = 1                 ;(see note 2)
AND M10 = 0                ;Also, turn on M10
AND P02 = ATAN (VY/VX)     ;Also, calculate  Angular value of
                           ;the commanded velocity of (VY/VX),
                           ;and store in P02
```

1 - This program line will be rejected by SMCC because P06 = P07 + 123 is invalid. The correct entry is P06 = P06 + 123, the parameter number used must be the same on both sides of the equation.

2 - The "OR" statement does not make sense when used as an action (ELSE) it will be omitted by SMCC. "OR" can be used with IF statements only.

3 - All other statements are valid.

A more useful example is to write a small PLC program that will send a message to the host each time the SMCC gets into position.

```
C01                    = ff01
M00                    = C01.7
IF M00                 = 0
THEN IF M01            = 1
THEN M01               = 0
AND SEND  "IN_POS"
ELSE
ELSE M01               = 1
END
```

Please note that M01 is used here as a "Debounce" or "Single Shot" in order to cause the message to only be sent once. Otherwise when M00 is activated, the program would send the message continuously and jam up the SMCC communication.

The following simple programming example shows the usefulness of a PLC program and suggests a way in which to organize the writing of PLC program.

This example program demonstrates the use of PLC for providing a function which allows a single switch to be used as end of Travel Limit switch and as a Home switch for precision zero setting. All that is required physically is that the switch be wired to both the "Home Flag" and the "Travel Limit" inputs of SMCC.

The concept upon which this program is based makes use of two servo status bits in order to determine whether to use the single switch as a homing flag or a Travel Limit switch. This will be explained in the program comments.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A0z                ;Address Card & Clear Program Buffer
SET C07=feba       ;Define C07 to be address feba (Status Register)
SET C08=ffca       ;Define C08 to be address ffca. (Interrupt flag
                   ;register).
SET M50=C07.6      ;Defines M50 to be bit 6 of feba (Limit status
                   ;bit, a high disables the limit.)
SET M51=C08.6      ;Defines M51 to be bit 6 of ffca (Interrupt
                   ;status bit, low if enabled).
HOM                ;Start a home cycle.  Choose and set appropriate
                   ;parameters for homing, i04 for speed, i24, 25,
                   ;and 26 X axis, i44, 45 and 46 Y axis.  Homing
                   ;should be in a direction that will drive it
                   ;toward the limit switch that is wired to both
                   ;the SMCC's "Home Flag" and Travel Limit"
                   ;inputs.
STO                ;STOP until "Run" or "Step" Command is received
                   ;again.
FOR0               ;Start a Loop forever.
F1024              ;Set feedrate to 100%, this is the beginning of
                   ;the motion program.
X100000            ;Starts a back and forth move of 100000 to 0 with
                   ;a one second Dwell at each end.
DWE1024            ;Wait one second.
X0                 ;Return to zero.
DWE1024            ;Wait one second.
NEX                ;End of Loop.
END                ;End of motion program.  Note that a program "end"
                   ;is used for the program itself,  and as will  be
                   ;seen further down, another "END" is used for the
                   ;PLC program.
A0PUR              ;Address Card and Purge all the PLC programs
                   ;stored in the PLC buffer.  If the existing
                   ;PLC program is not purged, the new PLC
                   ;program will be appended to what is already
                   ;in the PLC Buffer.
i600               ;i60 0 (i62 0 for Parabolic) turns off the
                   ;PLC so that while loading the PLC program
                   ;it does not inadvertently create undesirable
                   ;effects.  Remember, that PLC is always active
                   ;unless turned off, and that i60 or i62 simply
                   ;get executed on-line and are not stored in the
                   ;buffer.  The SMCC does turn off the PLC while
                   ;a program is being loaded, but it never hurts
                   ;to play it safe.
IFM51= 0           ;The IF statement starts the first PLC Branch
                   ;or statement which looks for condition M51=0
                   ;meaning:  Look to see if there is an interrupt
                   ;operation in progress, (The Home Command sets
                   ;ffca bit 6 low.)  If there is, then set feba
                   ;bit 6 high (M50=1) thus disabling the limit
                   ;switches.  (The bit automatically resets at
                   ;the end of each command cycle thus re-enabling
                   ;the limits.)
THENM50=1
ELSE               ;If a home cycle is not in progress then do
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ;nothing.  If the ELSE is not typed in, SMCC
                    ;will automatically insert it.
END                 ;The END statement closes PLC programming.  If
                    ;there are other PLC statements simply continue
                    ;with an "IF" which opens a new PLC statement or
                    ;branch until all PLC statements are completed.
```

The following table shows the program alterations needed for Parabolic and Y axis.

	STANDARD		PARABOLIC	
	X axis	Y axis	X axis	Y axis
Limit Status	feba.6	feba.6	feb6.4	feb6.4
Interrupt Status	ffca.6	ffc8.6	ffca.6	ffc8.6
PLC enable/disable	i60 1/0		i62 1/0	

Finally, a counter program is constructed as follows:

```
                              ;Counter, using M Function
                              ;input. (counter)
                              ;The Counter is triggered
                              ;on the trailing edge of the
                              ;MI1 input
     A0PURGE                  ;Clear PLC program
     IF M1=0                  ;Check if MI1 is toggled
                              ;M30 is used as flag to
                              ;cause the P1 counter to
                              ;only count once when MI1
                              ;toggles.
     THEN IF M30=0            ;Check if M30 Flag is set
     THEN M30=1               ;True, reset M30 Flag
     AND P1=P1+1              ;Add 1 to P1 counter
     AND "P1"                 ;Display value of P1 on
                              ;the screen
     ELSE                     ;
     IF M11=1                 ;Check if MI1 sw. is released
     THEN M30=0               ;True, set M30 Flag
     AND IF P1>999            ;Check if P1 counter has reached
     AND P1<1001              ;1000 counts
     THEN P1=0                ;True, reset counter
     AND SEND "1000-counts"   ;Send a message to the screen
     ELSE
     END                      ;End of PLC program
                              ;Note that the counter may be
                              ;preset by setting P1 to a
                              ;starting value.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PARAMETER AND COMMAND DIAGRAMS

TABLE OF CONTENTS

PAGE

i PARAMETERS 137

STANDARD SMCC M AND C FUNCTION MAP 138

PARABOLIC SMCC M AND C FUNCTION MAP. 139

M FUNCTION INPUTS AND OUTPUTS. 140

NON-BUFFER COMMANDS. 141

MOTION BUFFER COMMANDS 142

BACKGROUND PLC 143

PLC BRANCHES AND SUB-BRANCHES. 144

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i parameters

i00		SET FOLLOWING ERROR LIMIT
i01		SET IN-POSITION BAND
i02		SET MDI, MANUAL DATA INPUT MODE
i03		SET HAND SHAKE ENABLE IN HOST COMMUNICATION
i04		SET HOME FEED RATE
i05		SET SERVO TIME
i06		SET REFERENCE FEED RATE
i07		SET TIME SCALE
i08		SET ACCELERATION/DECELERATION TIME
i09		SET POSITION INTEGRATION MODE
i10		SET FOLLOWING ERROR LIMIT, INDICATOR
i11		SET "FEED HOLD" SLEW CONTROL
i12		SELECT 24/6 kHz PWM FREQUENCY
"PARB"i12		MOVE DELAY TIME OR CALCULATION TIME
"BRUSHLESS"i13		SELECT BRUSHLESS MOTOR
i14		SELECT PROCESSOR A/D CONVERTER MODE
i15		SET Y AXIS DISABLE IN BRUSHED MOTOR MODE
"BRUSHLESS"i15		SELECT LOOK-UP TABLE
i16		SELECT HANDWHEEL MODE 1 CONTROL
i17		SET VELOCITY DISPLAY TIME BASE
i18		SET DISPLAY MODE
i19		NOT USED IN STANDARD SMCC
"PARB"i19		HIGHEST CARD ADDRESSED IN SYSTEM
i20	[i40]	SET PROPORTIONAL GAIN VALUE
i21	[i41]	SET DIFFERENTIAL GAIN VALUE
i22	[i42]	SET VELOCITY FILTER TIME CONSTANT
"PARB"i22	[i42]	VELOCITY FEEDFORWARD GAIN
i23	[i43]	SET INTEGRAL GAIN CONSTANT
i24	[i44]	SELECT HOME DIRECTION
i25	[i45]	SET HOME OFFSET
i26	[i46]	SET HOME FLAG AND USER FLAG CONTROL BITS
i27	[i47]	SET HANDWHEEL SCALE FACTOR
i28	[i48]	SELECT HANDWHEEL MODE 2 CONTROL
"PARB"i28	[i48]	SELECT HANDWHEEL MODE 2 CONTROL
i29	[i49]	HANDWHEEL ENCODER MODE CONTROL BITS
i30	[i50]	SELECT GAIN MULTIPLICATION SCALER CONTROL
"PARB"i30	[i50]	ACCELERATION FEEDFORWARD
i31	[i51]	SET JOG FEED RATE
i32	[i53]	SET NORMAL PWM LIMIT CONTROL
i33	[i53]	SET PROTECTIVE PWM LIMIT CONTROL
i34	[i54]	BACKLASH
i35	[i55]	SET POSITIVE SOFTWARE POSITION LIMIT
i36	[i56]	SET NEGATIVE SOFTWARE POSITION LIMIT
i37	[i57]	SET POSITION RANGE
i38	[i58]	SET DISPLAY AND MOVE SCALE FACTOR AND FORMAT
i39	[i59]	POSTITION ENCODER MODE CONTROL BITS
i60		PLC ENABLE
"PARB" i60	[i61]	DEAD BAND
" i62		PLC ENABLE
" i63		PARABOLIC MODE CONTROL FOR DMA COMMUNICATION
" i64	[i65]	X ENCODER/RESOLVER BASE ADDRESS
" i66	[i67]	SPEED OF SECOND RESOLVER
" i68	[i69]	SECOND RESOLVER BASE ADDRESS
" i70	[i71]	ENCODER, RESOLVER BIAS
" i72	[i73]	ENCODER, RESOLVER SIZE
" i74	[i75]	SECOND RESOLVER SIZE
" i76	[i77]	ENCODER DATA/STROBE LOGIC LEVEL
" i78	[i79]	RANGE MODE CONTROL (SHORTEST DISTANCE OR ABSOLUTE)

BRUSHLESS (COMMUTATED BY SMCC)

i15	SELECT LOOKUP TABLE SINUSOIDAL OR TRAPEZOIDAL MODE
i40	SET ENCODER COUNTS PER ELECTRICAL REVOLUTION
i41	SET PHASE ADVANCE SCALE FACTOR
"PARB" i42	INDUCTION MOTOR MAGNETIZATION PWM
i43	SET LOOK-UP TABLE ADDRESS
i44	SELECT 3/4 PHASE
i45	SET HALL SENSOR PHASE OFFSET

BRUSHLESS

"PARB" i45	SLIP GAIN (NON-ZERO = INDUCTION)
i46	SET STAR-DELTA CONNECTION
i50	ENCODER/RESOLVER SELECTION
i51	SET INDUCTION MOTOR SLIP GAIN (SEE I45)
i52	SET PHASE FINDING TIME
i53	SET PHASING PWM VALUE
"PARB"i54	HALL EFFECT OR RESOLVER OFFSET

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

STANDARD

M-DEFINITIONS

M00=C00.0,1	M16=C05.1,0	M32=C00.0,1	M48=C00.0,1
M01=C01.5,0	M17=C05.2,0	M33=C00.0,1	M49=C00.0,1
M02=C02.6,0	M18=C00.0,1	M34=C00.0,1	M50=C00.0,1
M03=C02.7,0	M19=C00.0,1	M35=C00.0,1	M51=C00.0,1
M04=C03.6,0	M20=C00.0,1	M36=C00.0,1	M52=C00.0,1
M05=C03.7,0	M21=C00.0,1	M37=C00.0,1	M53=C00.0,1
M06=C00.0,1	M22=C00.0,1	M38=C00.0,1	M54=C00.0,1
M07=C00.0,1	M23=C00.0,1	M39=C00.0,1	M55=C00.0,1
M08=C00.0,1	M24=C00.0,1	M40=C00.0,1	M56=C00.0,1
M09=C00.0,1	M25=C00.0,1	M41=C00.0,1	M57=C00.0,1
M10=C00.0,1	M26=C00.0,1	M42=C00.0,1	M58=C00.0,1
M11=C04.0,1	M27=C00.0,1	M43=C00.0,1	M59=C00.0,1
M12=C04.1,1	M28=C00.0,1	M44=C00.0,1	M60=C00.0,1
M13=C04.2,1	M29=C00.0,1	M45=C00.0,1	M61=C00.0,1
M14=C04.3,1	M30=C00.0,1	M46=C00.0,1	M62=C00.0,1
M15=C05.0,0	M31=C00.0,1	M47=C00.0,1	M63=C00.0,1

SMCC TERMINAL V3.3

F1 - I-Parameters
F2 - P-Variables
F3 - C-Definitions
F4 - M-Definitions
F5 - List Program
F6 - Arrow Jog
F7 - Card Status
F8 - Display Pos
F9 - Subroutines
F10 - File Editor
NEXT MENU: ALT-Q

CARD: A0

<Esc> to EXIT

C-DEFINITIONS

C00=0000,85	C16=0000,85	C32=0000,85	C48=0000,85
C01=ff01,14	C17=0000,85	C33=0000,85	C49=0000,85
C02=ffb4,03	C18=0000,85	C34=0000,85	C50=0000,85
C03=ffb5,03	C19=0000,85	C35=0000,85	C51=0000,85
C04=ffb7,af	C20=0000,85	C36=0000,85	C52=0000,85
C05=ffbc,e0	C21=0000,85	C37=0000,85	C53=0000,85
C06=0000,85	C22=0000,85	C38=0000,85	C54=0000,85
C07=0000,85	C23=0000,85	C39=0000,85	C55=0000,85
C08=0000,85	C24=0000,85	C40=0000,85	C56=0000,85
C09=0000,85	C25=0000,85	C41=0000,85	C57=0000,85
C10=0000,85	C26=0000,85	C42=0000,85	C58=0000,85
C11=0000,85	C27=0000,85	C43=0000,85	C59=0000,85
C12=0000,85	C28=0000,85	C44=0000,85	C60=0000,85
C13=0000,85	C29=0000,85	C45=0000,85	C61=0000,85
C14=0000,85	C30=0000,85	C46=0000,85	C62=0000,85
C15=0000,85	C31=0000,85	C47=0000,85	C63=0000,85

SMCC TERMINAL V3.3

F1 - I-Parameters
F2 - P-Variables
F3 - C-Definitions
F4 - M-Definitions
F5 - List Program
F6 - Arrow Jog
F7 - Card Status
F8 - Display Pos
F9 - Subroutines
F10 - File Editor
NEXT MENU: ALT-Q

CARD: A0

<Esc> to EXIT

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

M-DEFINITIONS

M00=C00.0,0	M16=C05.1,0	M32=C00.6,0	M48=C13.4,0
M01=C01.5,0	M17=C05.2,0	M33=C00.7,0	M49=C13.5,0
M02=C02.6,0	M18=C00.0,0	M34=C11.6,0	M50=C13.6,0
M03=C02.7,0	M19=C00.0,0	M35=C11.7,0	M51=C13.7,0
M04=C03.6,0	M20=C10.0,0	M36=C12.0,0	M52=C14.0,1
M05=C03.7,1	M21=C10.1,1	M37=C12.1,0	M53=C14.1,0
M06=C00.0,0	M22=C10.2,1	M38=C12.2,1	M54=C14.2,1
M07=C00.0,0	M23=C10.3,1	M39=C12.3,1	M55=C14.3,0
M08=C00.0,0	M24=C10.4,1	M40=C12.4,1	M56=C14.4,1
M09=C00.0,0	M25=C10.5,1	M41=C12.5,1	M57=C14.5,1
M10=C00.0,0	M26=C10.6,1	M42=C12.6,1	M58=C14.6,1
M11=C04.0,1	M27=C10.7,1	M43=C12.7,1	M59=C14.7,1
M12=C04.1,1	M28=C11.0,0	M44=C13.0,0	M60=C15.0,0
M13=C04.2,1	M29=C11.1,0	M45=C13.1,0	M61=C15.1,0
M14=C04.3,1	M30=C00.2,0	M46=C13.2,0	M62=C15.2,0
M15=C05.0,0	M31=C00.3,0	M47=C13.3,0	M63=C15.3,0

SMCC TERMINAL V3.3

F1 - I-Parameters
F2 - P-Variables
F3 - C-Definitions
F4 - M-Definitions
F5 - List Program
F6 - Arrow Jog
F7 - Card Status
F8 - Display Pos
F9 - Subroutines
F10 - File Editor
NEXT MENU: ALT-Q

CARD: A0

<Esc> to EXIT

C-DEFINITIONS

C00=ff3a,00	C16=0000,22	C32=0000,22	C48=0000,22
C01=ff01,1c	C17=0000,22	C33=0000,22	C49=0000,22
C02=ffb4,2f	C18=0000,22	C34=0000,22	C50=0000,22
C03=ffb5,af	C19=0000,22	C35=0000,22	C51=0000,22
C04=ffb7,af	C20=0000,22	C36=0000,22	C52=0000,22
C05=ffbc,e0	C21=0000,22	C37=0000,22	C53=0000,22
C06=0000,22	C22=0000,22	C38=0000,22	C54=0000,22
C07=0000,22	C23=0000,22	C39=0000,22	C55=0000,22
C08=0000,22	C24=0000,22	C40=0000,22	C56=0000,22
C09=0000,22	C25=ffb8,00	C41=0000,22	C57=0000,22
C10=a3ad,fe	C26=0000,22	C42=0000,22	C58=0000,22
C11=a3ae,00	C27=0000,22	C43=0000,22	C59=0000,22
C12=a3af,fc	C28=0000,22	C44=0000,22	C60=0000,22
C13=a3b0,00	C29=0000,22	C45=0000,22	C61=0000,22
C14=a3b1,f5	C30=0000,22	C46=0000,22	C62=0000,22
C15=a3b2,00	C31=0000,22	C47=0000,22	C63=0000,22

SMCC TERMINAL V3.3

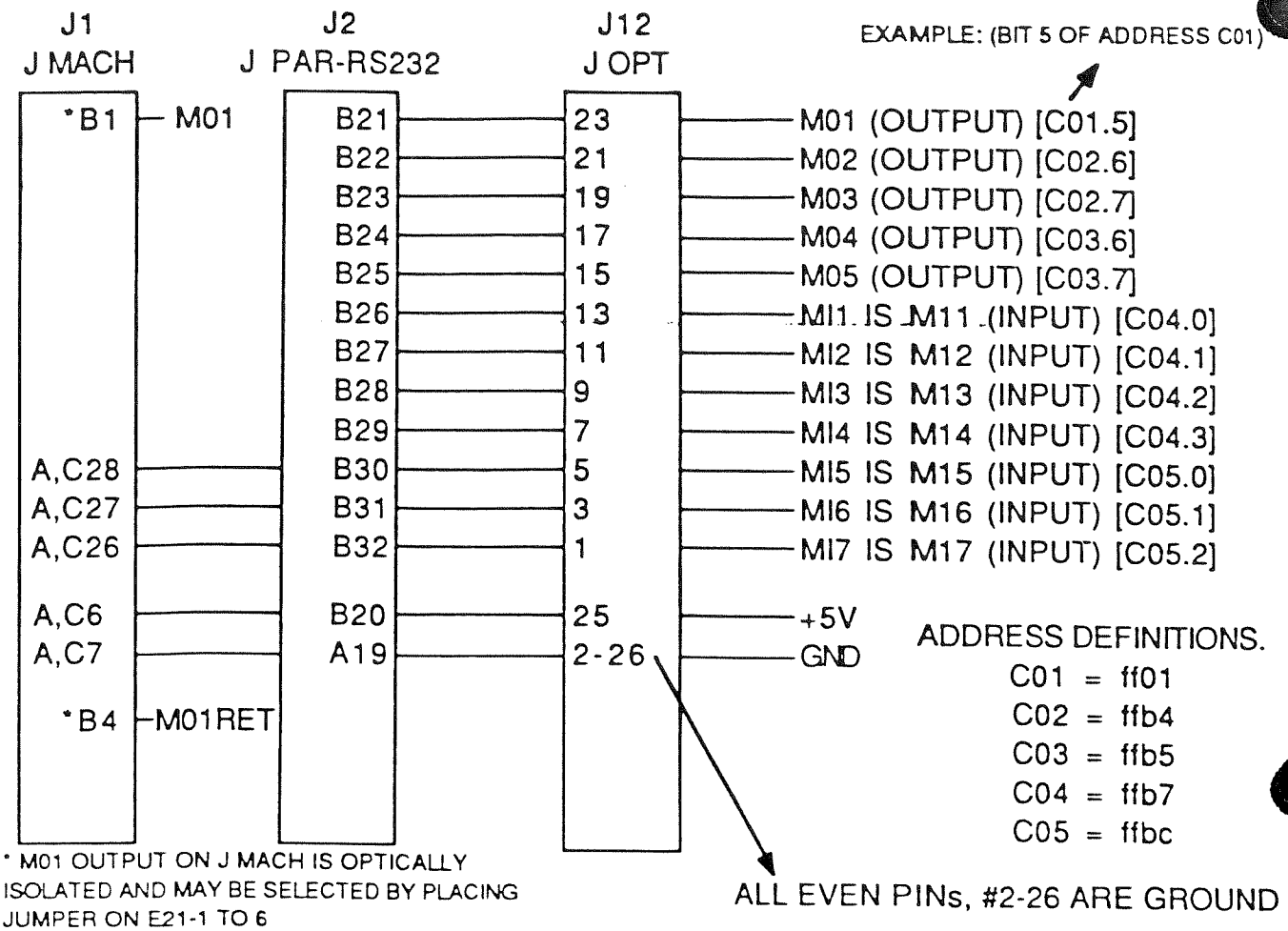
F1 - I-Parameters
F2 - P-Variables
F3 - C-Definitions
F4 - M-Definitions
F5 - List Program
F6 - Arrow Jog
F7 - Card Status
F8 - Display Pos
F9 - Subroutines
F10 - File Editor
NEXT MENU: ALT-Q

CARD: A0

<Esc> to EXIT

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC M FUNCTION INPUTS AND OUTPUTS



1) IN M- DEFINITIONS MAP FOR INPUTS
   MI1-MI4, (M11-M14), 1 IS OFF (INACTIVE), 0 IS ON (ACTIVE)
   MI5 - MI7, (MI5-M17), 0 IS OFF (INACTIVE), 1 IS ON (ACTIVE)
2) ALL INPUTS, LOW (GROUNDED) IS ACTIVE OR "ON",
   HIGH (+5 VOLTS MAX) IS INACTIVE OR "OFF".
   ALL INPUTS HAVE AN INTERNAL, 470 OHM, PULL UP RESISTOR TO +5 VOLTS.
3) ALL OUPUTS TURN ON TO GROUND WHEN ACTIVATED.
4) M02 TO M05 OUTPUTS ARE RATED FOR
   30 VOLTS MAX. AT 200 mA. M01 IS 30 V. AT 60 mA.
   IF RELAYS ARE DRIVEN, USE SUPPRESSION DIODE ACROSS RELAY COIL TO PREVENT INDUCTIVE VOLTAGE SPIKE.
5) THE J OPT CONNECTOR IS DESIGNED TO BE FLAT CABLE CONNECTED TO COMMERCIALLY AVAILABLE OPTO ISOLATION DEVICES, SUCH AS OPTO 22'S PB8 OR PB16.
6) USING ACCESSORY 14, ADDITIONAL I/O IS MADE AVAILABLE.
   3 ACC-14'S MAY BE DAISY CHAINED FOR A TOTAL OF
   48 x 3 = 164 I/O LINES.

M01 TO M05 AND M11 TO M17 ADDRESS DEFINITIONS ARE SET AT THE FACTORY. USE AS IS OR CHANGE AS DESIRED OR REQUIRED.

IN PROGRAM BUFFER, USE "SET nn" TO ACTIVATE AND "RESET nn" TO DE-ACTIVATE OUTPUTS.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

M-FUNCTION OUTPUT	JOPT.	JPAR-RS232	COMMAND
MO1	23	B21	SET 01 RES 01
MO2	21	B22	
MO3	19	B23	
MO4	17	B24	
MO5	15	B25	

* MO1-MO5 outputs low when SET.

M-FUNCTION INPUT	JOPT.	JPAR-RS232
MI1 is MI1	13	B26
MI2 is MI2	11	B27
MI3 is MI3	9	B28
MI4 is MI4	7	B29
MI5 is MI5	5	B30
MI6 is MI6	3	B31
MI7 is MI7	1	B32

* MI1-MI7 inputs are normally high when not activated and low when activated. Activated here means "GROUNDED".

NOTE: In M-definitions map, MI1-MI4 are the opposite of MI5-MI7.
When activated, MI1-MI4 is 0.
When activated, MI5-MI7 is 1.

See M&C map definitions for Address and Bit assignments.

Example 1) IF MI1=0
THEN . . .

Example 2) IF MI5=1
THEN . . .

Would both result in the "THEN" statement being executed when the respective MI1 and MI5 inputs are grounded.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SUMMARY OF DIRECT CONTROL COMMANDS.
(NON-BUFFER COMMANDS)

Q	QUIT	* THESE COMMANDS ARE ALSO AVAILAE
* R	RUN	AS CONTROL PANEL INPUT SWITCHES
* S	STEP	
* H	FEED HOLD	
$	SOFT RESET	
* $$	HARD RESET	
* =	RETURN TO PRE-JOG POSITION	
Z	ZERO SET	
* h	START HOMING CYCLE	
k	KILL SERVO	
* q	ABORT CURRENT MOVE	
* J(X/Y)	JOG POSITIVE X AND Y	
* j(X/Y)	JOG NEGATIVE X AND Y	
O(X/Y)	OPEN-LOOP CONTROL	
%N,n	FEED RATE % OVERRIDE	
bnn	GOTO BEGINNING OF PROGRAM	
l	LEARN AND STORE X AND Y	
znn	PURGE BUFFER	
rA,v or dA,v	READ MEMORY/ READ MEMORY FORMATTED	
wA,d	WRITE DATA	
s	SAVE PROGRAM	
Cmm=a	DEFINES Cmm TO BE ADDRESS a	
Mnn=Cmm.b	DEFINES M FUNCTION BIT	
Mnn = 0/1	SET M FUNCTION BIT TO ZERO OR ONE	
innv	SET PARAMETER inn VALUE	
inn	SEND VALUE OF PARAMETER	
ix	SEND VALUE OF ALL PARAMETERS	
f or p	SEND FOLLOWING ERROR OR CURRENT POSITION	
v	SEND CURENT VELOCITY	
?	SEND STATUS AND ERROR	
%	SEND VALUE OF % FEED RATE	
Bnn	SEND PROGRAM LINE NUMGER FOR LABEL	
Bx	SEND PROGRAM LINE NUMBER FOR ALL LABELS	
Cmm	SEND DEFINED MEMORY LOCATION	
Cx	SEND DEFINED MEMORY LOCATION FOR ALL ADDRESSES	
PC	SEND NEXT PROGRAM LINE POINTER VALUE	
PB	SEND SIZE OF PROGRAM BUFFER	
Pn or Px	SEND VARIABLE PARAMETER Pn OR ALL PARAMETERS	
PC = v	SET PROGRAM COUNTER TO VALUE	
Pn=v	SET VARIABLE PARAMETER	
LIST n,m	LIST M LINES STARTING AT LINE n	
DELETE n,m	DELETE m LINES STARTING AT LINE n	
INSERT n	INSERT ADDITIONAL LINES	
REPLACE n	OVERWRITE PROGRAM LINE n	
MAP n	REALLOCATE n STEPS TO BUFFER	
PLC n	LIST THE PLC PROGRAM	
PUR n	CLEAR THE PLC PROGRAM	
<CNTRL B>	DMA MODE, GET READY TO RECEIVE BINARY DATA	
<CNTRL C/D/E>	SEND BINARY POSITION/ STATUS/ PARAMETER DATA	
<CNTRL R>	SET % TO i07 VALUE	
<CNTRL V>	FREEZE POSITION	

Artisan Scientific - Quality Instrumentation ,,. Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

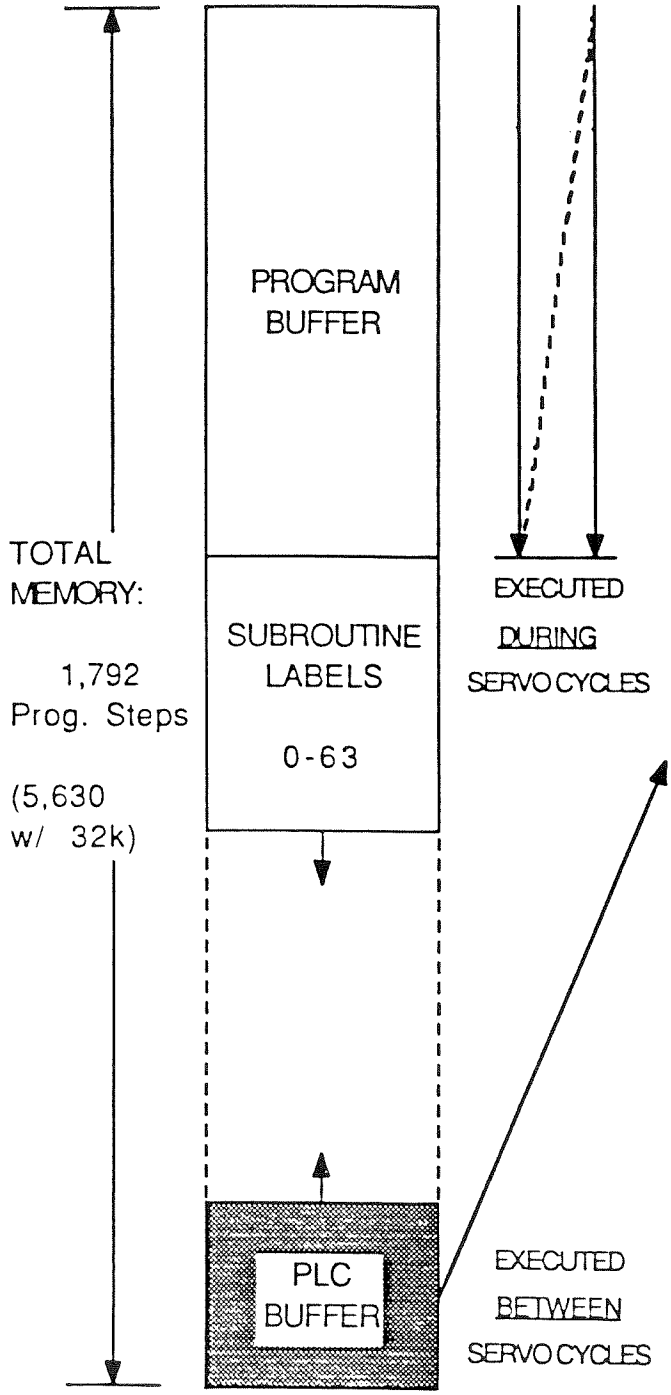
MOTION BUFFER COMMANDS

SET nn	SET THE BIT DEFINED BY Mnn
RESET nn	RESET THE BIT DEFINED BY Mnn
GOTO+nn	JUMP nn PROGRAM LINES FORWARD
GOTO-nn	JUMP nn PROGRAM LINES BACKWARD
GOTO nn	JUMP TO PROGRAM LINE WITH LABEL Lnn, DO NOT RETURN
GOSUB nn	GO TO PROGRAM LINE WITH LABEL Lnn, THEN RETURN
IF	CONDITIONAL JUMP OR GOTO
IFNOT	CONDITIONAL JUMP OR GOTO (INVERSE)
Lnn	SUBROUTINE LABEL
START	START CONTINUOUS EXECUTION OF PROGRAM
STOP	STOP CONTINUOUS EXECUTION OF PROGRAM
END	END OF PROGRAM
RETURN	RETURN AFTER EXECUTING SUBROUTINE
FORd	START A LOOP, d TIMES
NEXT	END OF LOOP
FND	JUMP OUT OF A FOR/NEXT LOOP
CASE mm,nv	JUMP TABLE, THUMBWHEEL INPUT
DIVIDE d	DIVIDE THE NEXT MOVE INTO d EQUAL PARTS
F d	SET FEED RATE
T d	SET MOVE TIME TO d/1024 SECONDS
t d	SET CHANGE OF SPEED TIME TO d/1024 SECONDS
HOME	GO TO HOME POSITION
DWELL d	SET DWELL TIME TO d/1024 SECONDS
DLY d	DELAY TIME, d/1024 SECONDS, SYNCHRONOUSLY
X d	MOVE X AXIS TO ABSOLUTE POSITION d
Y d	MOVE Y AXIS TO ABSOLUTE POSTION d
AX d	MOVE X AXIS TO ABSOLUTE POSITION d RELATIVE TO THE LAST HOME
AY d	MOVE Y AXIS TO ABSOLUTE POSITION d RELATIVE TO THE LAST HOME
U d	INCREMENTAL MOVE X AXIS d DISTANCE
V d	INCREMENTAL MOVE Y AXIS d DISTANCE
X or Y: V	PARABOLIC MOVE
WX d	X WAIT
WY d	Y WAIT
X or Y ^ d	GO UNTILTRIGGER RECEIVED, THEN MOVE d DISTANCE
X or Y = d	SET THE PRESENT X or Y POSITION TO d
RX = d	SET THE PRESENT X RADIUS TO d
RY = d	SET THE PRESENT Y RADIUS TO d
R = d	SET THE PRESENT X AND Y RADII BOTH TO d
C = d	SET OFFSET FOR ELIPSE (not in parabolic)
LINEAR	LINEAR INTERPOLATION MODE
CIRCLEn	CIRCULAR INTERPOLATION MODE/ 3 POINT SMOOTHER MODE
TORQUE	TORQUE MODE
POT d	SETS PERCENT FEED POT TO d
SLEW d	SETS PERCENT FEED POT SLEW RATE TO d
Inn d	SETS Inn TO THE VALUE d
m a,n	WRITES THE BYTE n TO MEMORY LOCATION a
mm a,n	SAME AS m, BUT EXECUTES IMMEDIATELY
n a,n	LOGICAL AND
u a,n	LOGICAL OR
x a,n	LOGICAL XOR
SET Pn = d	SET THE VARIABLE Pn TO THE VALUE d
SET Pn = AX/Y	SET THE VARIABLE Pn TO ABSOLUTE POSITION OF X OR Y
RESET P	SETS ALL 15 VARIABLE PARAMETERS TO ZERO
SET Cnn = a	DEFINE Cnn TO BE THE ADDRESS a
SET Mnn = Cmm.b	DEFINE Mnn TO BE THE BIT POSITION b AT THE ADDRESS DEFINED BY Cmm

(MOTION BUFFER COMMAND)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

BACK - GROUND PLC



PLC IS ALWAYS ACTIVE
REGARDLESS OF PROGRAM FLOW

IF (CONDITION)
AND IF
THEN IF
ELSE IF Pn ≷
position ≷
foll. error ≷
velocity ≷
Cnn,m ≷
m=1 to 4 bytes
Mnn (I/O bit) = 0/1

DATA OR PARAMETER

AND (CONDITIONS)
OR (CONDITIONS)

THEN (ACTION)

Inn = DATA (SERVO)
SET nn /RES nn (M FUNCTION)
SEND "MESSAGE" (TO HOST)
"COMMAND" (TO SMCC)
Cnn,m + DATA (+ MEMORY)
Cnn,m = DATA (C. MEMORY)
Tn= DATA (4 TIMERS)
Pn = ATAN (X/Y)
Pn = (AX/AY)
Pn = Pn + (VALUE)
Pn = DATA (VARIABLE)

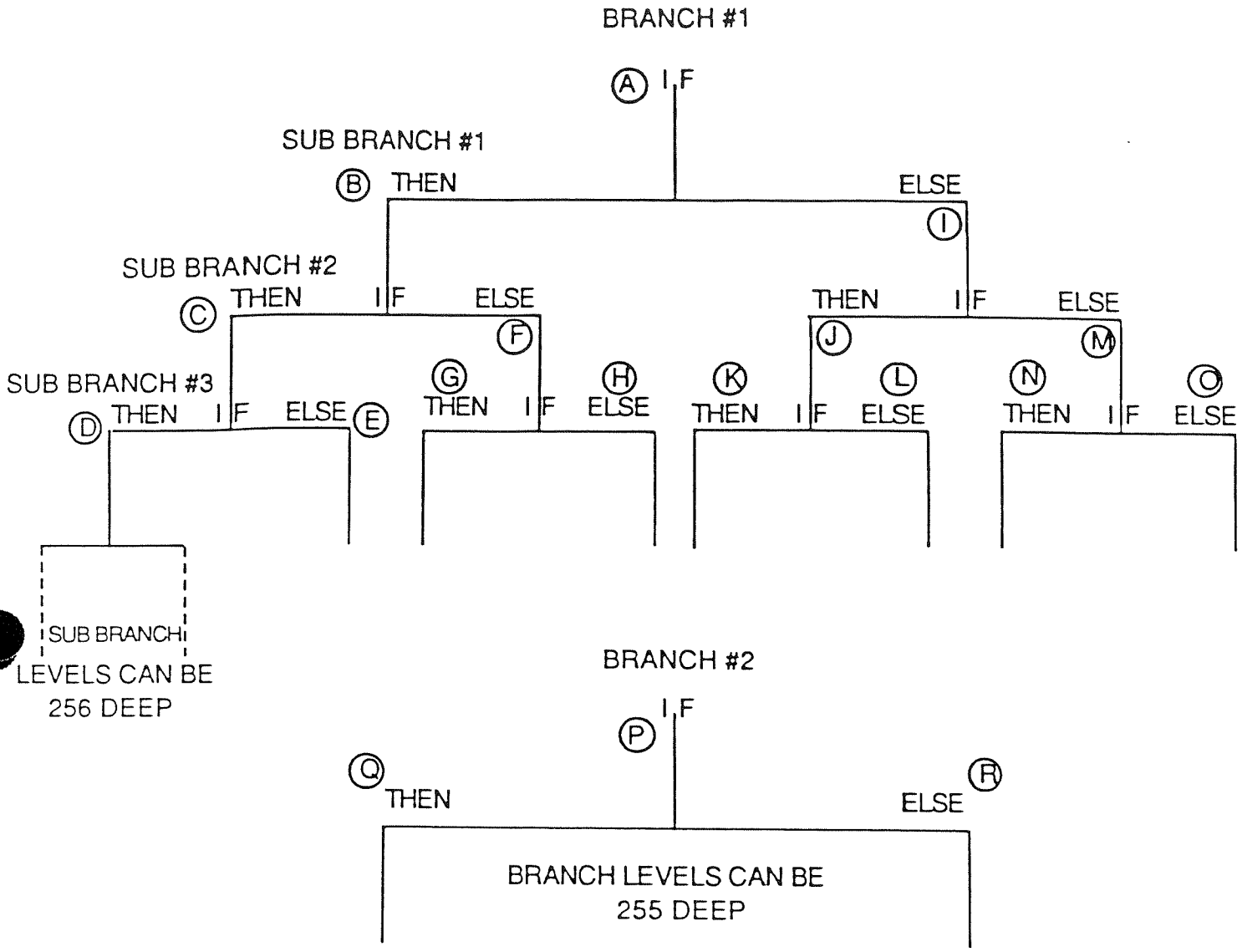
AND (ACTIONS)

ELSE (ALTERNATE ACTION)
(Same as THEN...)

OTHER BACK-GROUND TASKS
IF - THEN - ELSE
256 TOTAL

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PLC BRANCHES AND SUB-BRANCHES



PLC EXECUTION SEQUENCE IS FROM (A) TO (R) AS SHOWN

(PLC BRANCH)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC WITH PARABOLIC CAPABILITY
(SMCC - P)

TABLE OF CONTENTS

OVERVIEW 147
i PARAMETERS 148
i03,i08,i12,i16,i19,i22 148
i28,i30,i42,i45,i50,i60[i61],i62,i63. 149
i64[i65],i66[i67],i68[i69],i70[i71] 149
i72[i73],i74[i75],i76[i77],i78[i79] 149
PARABOLIC PROFILING 150
Direct Specification. 150
Mathematical Background 151
Fig. 1 Parabolic Motion Profiles. 152
Fig. 2 Point-to-Point Parabolic Move. 154
Fig. 3 Parabolic Acceleration 154
Fig. 4 Straight Line Acceleration 155
Fig. 5 Parabolic Acceleration 155
"S" Curve (Three-Point Smoother) Mode 158
Fig. 6 General Parabolic Move Segment 159
Fig. 7 Point-to-Point S-Curve Move. 159
General Algorithm for Point-to-Point S-Curve Moves. . . 161
Fig. 8 General S-Curve Acceleration 162
Fig. 9 Single Part S-Curve Move 162
Three-Point Smoothing (Cubic Fit) Applications. 164
Table 1 166
Table 2 167
Table 3 168
ACCELERATION AND VELOCITY FEEDFORWARD 169
SMCC Servo Algorithms
SMCC With Parabolic Option 169
SMCC Standard (Non-Parabolic) 170
AC INDUCTION SERVO 170
1/T ENCODER PERIOD 170
DEADBAND 171
DATA GATHERING CAPABILITY 171
Data Gathering Instructions 173
F1-COLLECT DATA 173
F2-COLLECT/STORE 174
F3-GET DISK DATA 174
F4-PLOT DATA 174
F5-DISPLAY RAW DATA 177
F6-DISPLAY DEC DATA 177
F7-PRINT RAW DATA 178
F8-PRINT DEC DATA 178
F9-SYS COMMANDS 178
F10-PLOT HELP 178
DMA COMMUNICATIONS. 178
DMA, Host to SMCC-P 179
DMA, Binary SMCC-P to Host 180

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PAGE

STATUS REQUEST 181
EAROM SAVE 182
EFFECTS OF "t" OR i08 182
"C" COMMAND 184
PARALLEL BINARY POSITION. 184
GEARED RESOLVERS 184
RANGE MODE CONTROL. 184
"GO UNTIL" 185

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC WITH PARABOLIC CAPABILITY (SMCC-P)

****

OVERVIEW

SMCC's Parabolic option uses a completely different masked CPU and software to accomplish its motion control tasks. The differences between a Standard SMCC and a Parabolic SMCC (SMCC-P) are summarized as follows:

1) Some i Parameters are different and there are more of them, but all PLC and motion commands are otherwise identical, with a few exceptions that are listed below.

2) Parabolic can generate "Cubic" profiles for path control and "Parabolic" velocity profiles of the form (X position:velocity) with specified time. Also, it provides a three point cubic fit on-the-fly, in CIRC4 Mode.

3) SMCC-P adds velocity and acceleration feedforward to the servo algorithm and changes i21, the velocity feedback term, to provide a different type of servo setup procedure.

4) AC induction servo is internal to CPU, therefore faster.

5) Provides 1/T encoder (period) measurements to improve low velocity control.

6) Adds deadband capability when in position.

7) Provides data gathering (trace) capability which allows the storing of position and following error data while a move is being executed.

8) Provides special DMA communications in binary format for high speed data transmission to and from host computer.

9) The status request from SMCC-P contains twice as many bytes of information as the Standard SMCC.

10) SMCC-P does not save the P, C, and M functions in the small serial EAROM. Instead it uses the 8K EAROM (Option 2) to save these functions. This makes it possible to save a program with approximately 80 steps in the small EAROM, as opposed to six steps for the Standard SMCC.

11) The effect of t or (i08) is changed in SMCC-P.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

12) The "C" value in circular interpolation, which allows off-axis ellipses to be made, is not available in SMCC-P. Therefore, to make such inclined (non-vertical or non-horizontal) ellipses, it is necessary to use parabolic commands directly.

13) SMCC-P is capable of reading and using parallel binary words (8 to 23 bits) for position information via the Accessory 14 Board.

14) SMCC-P is capable of handling geared resolvers for multi-speed absolute positioning including 3-, 6-, or 12-pole motors.

15) A Circular Range parameter is provided in SMCC-P. When invoked allows the ability to do incremental or absolute positioning within the rollover range.

16) A "Go Until" Command is provided for allowing an external trigger to alter the motion profile. The form is (X position^delta position).

Each of the above items will now be explained in detail.

1) i PARAMETERS

For a detailed explanation of i Parameters, please refer to the i Parameter Description Section of this manual. The differences between Standard and Parabolic are listed below:

i03 Handshake control SMCC-P adds **two new handshake modes**.

i08 Changes function - In Parabolic, i08 **controls only jog and feed hold accelerations**. "t" controls calculated move accel and decel. i08 still provides the default "t" value on power-up or reset.

i12 SMCC-P uses only 24KHz PWM frequency. i12 is now changed to specify a **move delay time or calculation time**. This is used to provide accurate multi-axis synchronization.

i16 **Handwheel mode control** - Uses i27 and i47 ratio instead of just i27 for handwheel multiplier when in i16 2 or 3 mode.

i19 Specifies **highest card address** (for DMA only).

i22 Changes meaning from velocity filter to **velocity feedforward gain**. Velocity filter is not used in

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC-P since this tends to reduce servo performance.

i28 **Handwheel mode 2 control** - Adds 1/T operating mode selection to handwheel. In 1/T i27[i47] is disabled. 1/T is used to measure the encoder's pulse period rather than frequency and thus results in smoother low velocity control.

i30 Changes meaning from gain multiplier to **acceleration feedforward gain.** Gain multiplier is not available on SMCC-P.

*i42 Is now **magnetization current.**

*i45 Is now **slip gain.**

i50 **Encoder versus resolver** selection.

i60[i61] Are now respectively the **X- and Y-axis deadband control.

i62 Is **PLC enable/disable control.**

i63 Is **parabolic mode control,** controls DMA communications format only.

i64[i65] Is **X and Y parallel encoder** and **resolver base address.**

i66[i67] Specifies the **X and Y resolver speed** (Gear Reduction).

i68[i69] Is **X and Y second resolver base address.**

i70[i71] Is **X and Y parallel encoder and resolver bias.**

i72[i73] Is **X and Y parallel encoder and resolver size.**

i74[i75] Is **X and Y second resolver size.**

i76[i77] Is **X and Y encoder data and strobe logic level, polarity control.**

i78[i79] Is **X and Y circular range mode control.**

NOTES: 1) *For brushless and induction motors only.
2) **Parameter numbers 61 and over are on SMCC-P only.
3) Only parameters 00 to 63 can be written into program buffer using "I" prefix.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2) PARABOLIC PROFILING

The term "parabolic" refers to the SMCC-P's capability to generate parabolic (quadratic, or second-order) velocity profiles, which yield cubic (third-order) position trajectories. This is one order higher than the standard SMCC (and most other controllers), which can generate linear (first-order) velocity profiles for second-order position trajectories. The extra order in the trajectories gives the SMCC-P much greater flexibility for blending of moves, smooth starting and stopping, and close control of the path of movement.

Even when executing lower-order trajectories, the SMCC-P generates the trajectories internally using third-order position equations, the low-order trajectories (e.g. trapezoidal moves) just being special cases of the general equations. The moves generated for these lower-order trajectories will be the same as those on the standard SMCC for equivalent commands, although calculated differently. For example, a standard point-to-point move of time 'T' from point A to point B with acceleration and deceleration times of 't' is broken into three concatenated parts: accel, slew, and decel. The acceleration portion lasts for time 't', with constant acceleration (dA/dt=0) up to the slew speed. The slew portion has constant velocity (accel=0; dA/dt=0) up until time 'T' (duration T-t), and the deceleration portion slows down at a constant rate to zero velocity at point B and time 'T+t' (duration t). All the key factors in the equations are calculated automatically.

It is important to note that the trajectories described in this section are all commanded trajectories, not necessarily the actual trajectories that the system will follow. It is up to the feedback and feedforward parameters, and the physical plant itself, to ensure that the actual movement of the system is as close as possible to the mathematically generated trajectories. It is very easy to generate a command trajectory that the plant is incapable of following, and this must be guarded against in writing any motion program.

There are two ways to generate parabolic velocity profiles on the SMCC-P: directly, and through use of the CIRCLE4 mode (also known as S-curve or three-point smoother mode). First, we will discuss the direct method of specifying these profiles.

Direct Specification

Because there is an extra order in the parabolic profiles, the user must specify an additional constraint to define the move completely. Therefore, in addition to the position and time specifications of a 'regular' move, the user must specify the velocity at the end of the move segment. From the constraints of time for the move, end position and velocity that were specified for the move segment, plus starting position and starting velocity, the card will calculate the unique cubic position trajectory necessary to meet the constraints.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The basic command format is as follows:

```
t1024                ;specifies the time for the move
X10000:1024          ;specifies end position and velocity
```

The small-t time specifier is the same 't' that sets the accel and decel times for non-parabolic moves. For parabolic moves, it sets the time for the entire move segment. One 't' specifier can define the time for more than one move segment. As in other time specification, the units are 'binary milliseconds' (1/1024 sec),

so this example specifies a move time of one second. 'T' and 'F' settings do not affect parabolic commands.

The line specifying end position and velocity is like any of the Standard Position Move Commands (X, Y, U, V, AX, AY) with a Velocity Command added after a colon. The velocity at the end of the move segment will be:

Vend = i06 * Vspecified / 1024 (cts/sec)

where i06 is the reference feedrate. In the example above, where the specified velocity is 1024, the velocity generated at the end will be equal to i06. If you wish to specify your velocity directly in cts/sec, simply set i06 to 1024.

Note: Methods of changing speed by altering the time base, such as the '%' and 'POT' Commands, or handwheel or pot time base inputs, work independently of this, and can alter the speed from that commanded in these program lines.

Warning: If the SMCC-P leaves a program where the last motion command was a parabolic command with a non-zero velocity, the card will continue to generate a trajectory at this velocity indefinitely. This can lead to an undesired 'runaway' condition. For this reason, the last in a series of parabolic commands should always have a zero velocity Command. Also, one should not try to single-step through the parabolic steps of a program, because the card will continue on at the last commanded velocity between steps, getting completely away from the desired position.

Mathematical Background

Figures 1(a) through 1(d) show the graphs of the general case of a 'parabolic' move segment. Each graph is derivative of the one above it (or, the integral of the one below it). The key thing about these profiles is that the dA/dt is non-zero, yielding third-order equations for position.

Generally, we will be working with velocity-vs.-time graphs to describe these moves. On these graphs, the change in position is the area under the curve, and the acceleration is the slope of the curve at any point in time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

"PARABOLIC" MOTION PROFILES

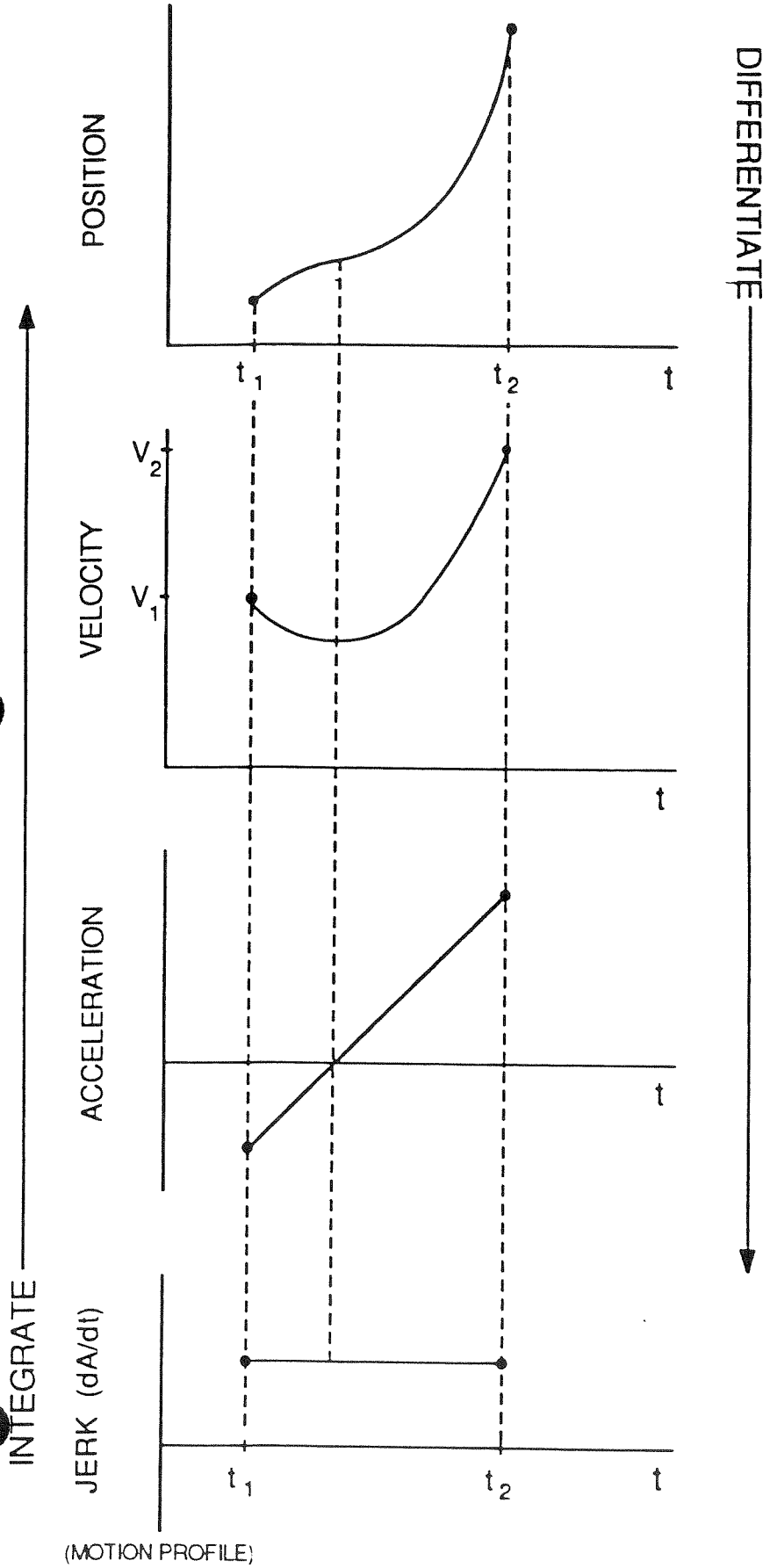


FIGURE 1(a)
POSITION vs.TIME

CUBIC (3rd ORDER)

FIGURE 1(b)
VELOCITY vs.TIME

PARABOLIC (2nd ORDER)

FIGURE 1(c)
ACCELERATION vs. TIME

LINEAR (1st ORDER)

FIGURE 1(d)
dA/dt vs. TIME
(JERK)

CONSTANT (Oth ORDER)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The first case we will examine is that of a point-to-point move using a single parabolic command specifying a zero end velocity, such as:

```
t1024              ;1 second
U10000:0           ;change in position of 10000 counts
                   ;ending with zero velocity
```

The graph of the resulting profile is shown in Figure 2. It is a symmetrical parabola, starting and stopping at zero velocity. The area under the parabola is the distance traveled -- in this example, 10000 counts. The slope of the parabola at any point is the acceleration, which is at maximum magnitude at the start and the stop, and zero at the midpoint. The two key equations for this type of move, which can be derived with simple calculus, are:

$$\text{delta P} = (2/3) * \text{Vpeak} * \text{delta t} \qquad \text{and}$$

$$\text{Apeak} = 6 * \text{delta P} / (\text{delta t})^2$$

where delta P is the change in position, and delta t is the time for the move

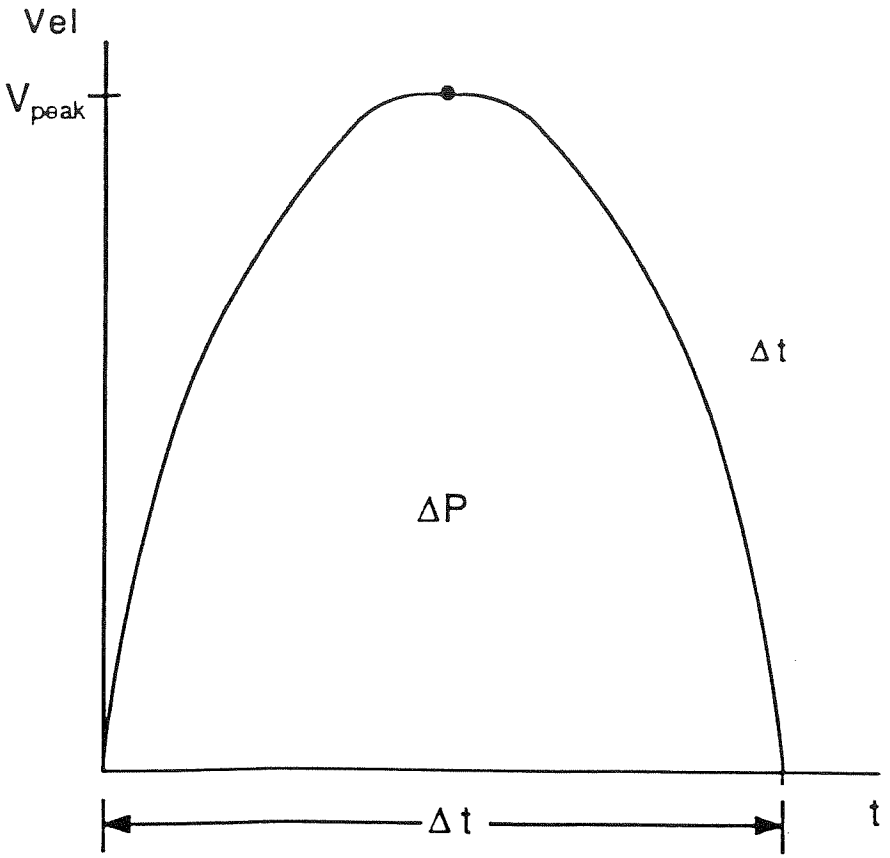
These equations can be rearranged to solve for whichever term is the unknown, as shown in the equations in Figure 2. (In solving these equations, be careful of your units, particularly distinguishing between seconds and binary milliseconds.) A common situation is that of trying to calculate a minimum-time move of a certain length, given system constraints as to maximum velocity and/or acceleration. In this case, delta P, Vmax, and Amax would be known, and the equations would be solved for delta t. If both Vpeak and Apeak were constraints, the larger of the two values calculated for delta t would have to be used, so as not to violate the other constraint.

This type of move is important because it causes the minimum energy dissipation in the motor (for covering a given distance in a given time). The mathematics to prove this is quite involved, but the idea behind it is quite simple. The power dissipated in the motor at any moment is proportional to the square of the current, and so to the square of the acceleration. The parabolic profile, by providing the maximum acceleration at the lowest velocities, enables the bulk of the distance to be covered at low values of acceleration, and therefore, low power dissipation.

Another common situation to be solved is that of accelerating to (or equivalently, decelerating from) a peak velocity from (to) a stop in a parabolic move segment. A whole family of curves can perform this task, depending on what value should be maximized or minimized. The three most useful of these curves are shown in Figures 3, 4, and 5.

The curve in Figure 3 shows the most gentle start-up possible. The equations to the right of the figure show the relations

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



$$\Delta P = \frac{2}{3} V_{peak} * \Delta t$$

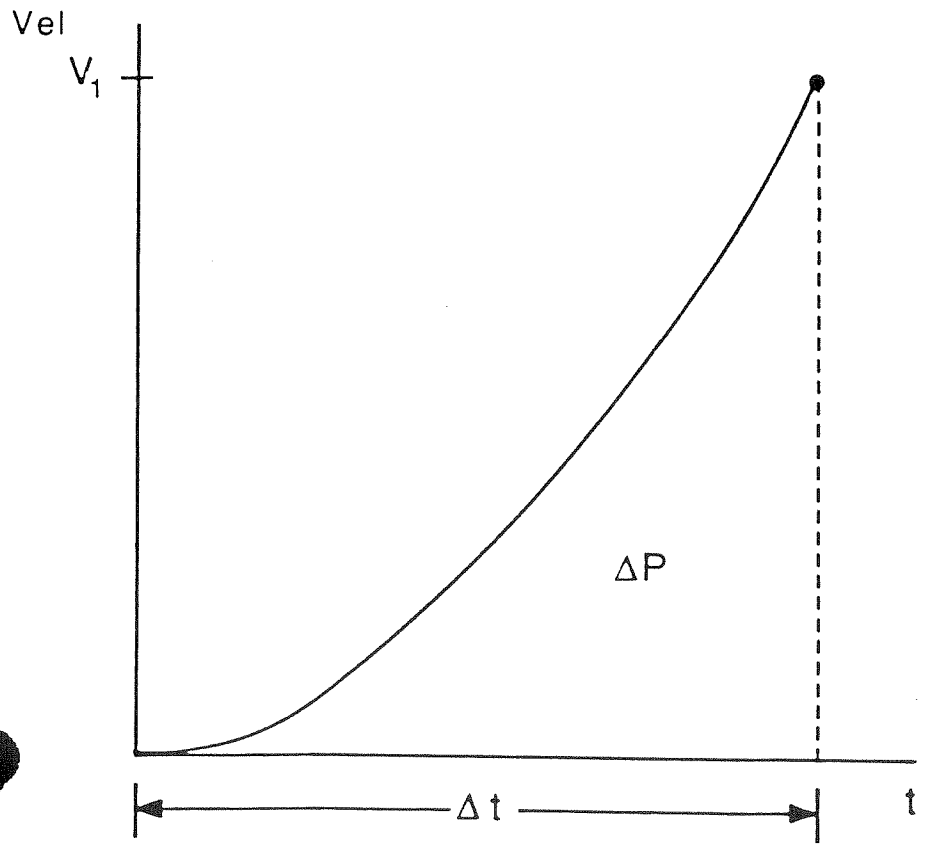
$$\text{or } V_{peak} = \frac{3\Delta P}{2\Delta t}$$

$$\text{or } \Delta t = \frac{3\Delta P}{2V_{peak}}$$

$$A_{peak} = \frac{6\Delta P}{(\Delta t)^2}$$

$$\text{or } \Delta t = \sqrt{\frac{6\Delta P}{A_{peak}}}$$

FIGURE 2. POINT-TO-POINT PARABOLIC MOVE



$$\Delta P = \frac{1}{3} V_1 \Delta t$$

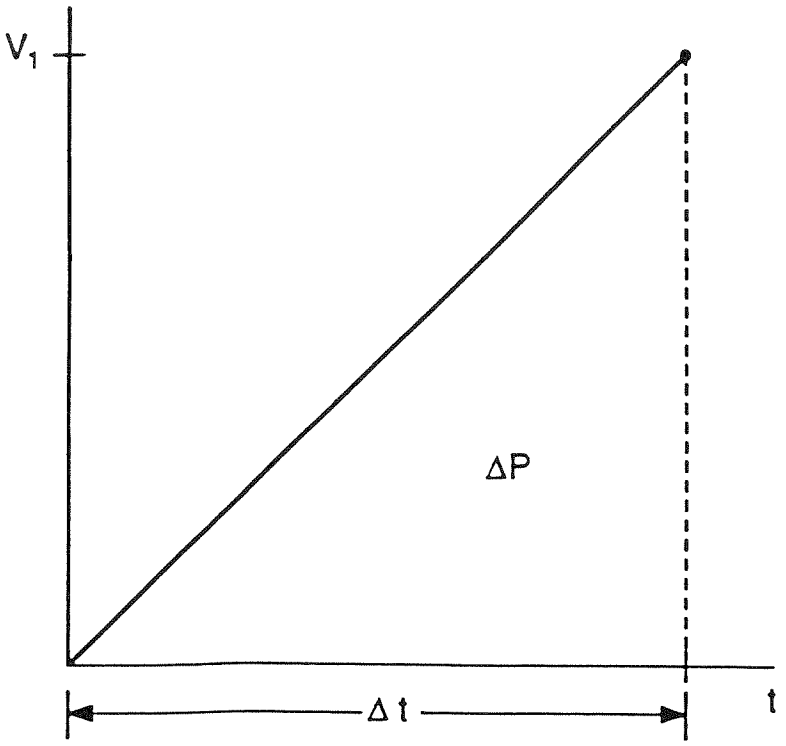
$$\text{or } \Delta t = 3 \frac{\Delta P}{V_1}$$

$$A_{peak} = A(\Delta t) = 2 \frac{V_1}{\Delta t}$$

$$\text{or } \Delta t = 2 \frac{V_1}{A_{peak}}$$

FIGURE 3. PARABOLIC ACCELERATION: MAX. NEGATIVE INFLECTION

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



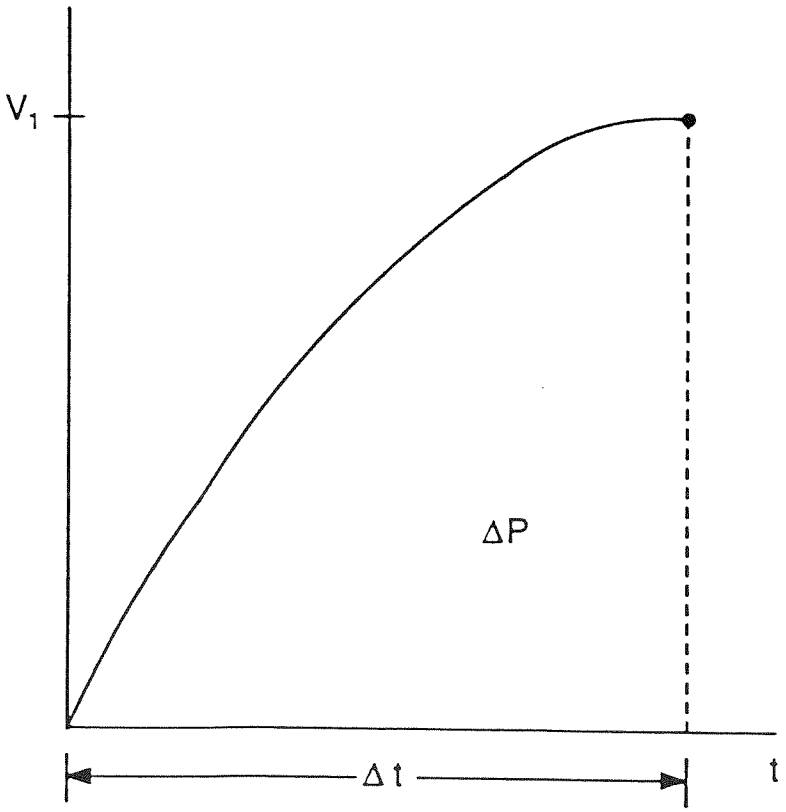
$$\Delta P = \frac{1}{2} V_1 \Delta t$$

or $$\Delta t = 2 \frac{\Delta P}{V_1}$$

$$A_{peak} = A_{avg} = \frac{V_1}{\Delta t}$$

or $$\Delta t = \frac{V_1}{A_{peak}}$$

FIGURE 4. STRAIGHT LINE ACCELERATION



$$\Delta P = \frac{2}{3} V_1 \Delta t$$

or $$\Delta t = \frac{3}{2} \frac{\Delta P}{V_1}$$

$$A_{peak} = A(0) = 2 \frac{V_1}{\Delta t}$$

or $$\Delta t = 2 \frac{V_1}{A_{peak}}$$

FIGURE 5. PARABOLIC ACCELERATION: MAX. POSITIVE INFLECTION

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

between distance, end velocity, maximum acceleration, and time. Typically, the time would be calculated from the desired end velocity and the constraint of maximum acceleration, and the distance would be calculated from that.

Figure 4 shows the case of straight-line acceleration, which is the minimum-time curve to get up to velocity if maximum acceleration is the only constraint. (The strain on the equipment that this instantaneous change to maximum acceleration causes makes many people shy away from this approach). Although parabolic commands are not required to generate this type of curve, they will work provided the constraints of the equations in Figure 4 are satisfied.

The curve in Figure 5 shows the case of the parabolic acceleration that blends most smoothly into a steady velocity move segment. The accompanying equations are identical in form to those in Figures 3 and 4.

Example: Suppose we wish to do a straight line acceleration to a speed of 200,000 cts/sec, given a maximum acceleration of 1,000,000 cts/sec/sec. From Figure 4, we would first compute delta t = 200,000/1,000,000 = 0.2 seconds. Then we would compute delta P = (1/2) * 200,000 * 0.2 = 20,000 counts. The program lines to do this would be:

```
i06 1024            ;so velocity specified in cts/sec
t205                ;0.2 seconds in binary milliseconds
U20000:200000       ;incremental move of 20000 counts
                    ;with ending velocity of 200000 cts/sec
```

Constant velocity move segments can also be done using parabolic commands, and the math is very simple. Suppose we wish to add to the above acceleration segment a constant speed segment of 400,000 counts at this maximum velocity. The time for this segment is found by simple division: 400,000 cts / 200,000 cts/sec = 2 sec. The program lines for this segment would be:

```
t2048               ;2.0 seconds in binary milliseconds
U400000:200000      ;incremental move of 400000 counts
                    ;with ending velocity of 200000 cts/sec
```

In general, any arbitrary move profile can be described using a set of parabolic commands. The only restriction is that a 't' time of at least 16 binary milliseconds be used. This is due to the fact that SMCC-P requires approximately 8 milliseconds per axis to calculate the profile parameters for the next move segment ('block'). If the move segment times are not long enough, the SMCC-P will run out of calculation time and stop executing the program. A good minimum value to use is t20, which provides an execution rate of 50 blocks/second. If t is close to the minimum value, communications and PLC program tasks may take up enough time that they may cause the card to 'hang up'. However, if t is sufficiently large, this is not a worry.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

For the general case of a parabolic move segment, it is important to check that at no point during the segment does the move exceed system capabilities. The following analysis can be used for determining whether the velocity profile during a parabolic move will generate a minimum or maximum velocity which is outside the range of beginning and ending velocities, and if so, what the magnitude of the extreme velocity is.

Figure 6 shows a parabolic move segment with the following parameters:

Vs = Starting velocity
Vf = Final velocity
dP = Distance to move (delta P)
t = Time for move
Vm = Mean velocity = dP/t

First calculate X as the time of the extreme velocity as a fraction of the time for the move:

$$X = (3Vm - 2Vs - Vf) / (6Vm - 3Vs - 3Vf)$$

If X is between 0 and +1, the extreme velocity occurs between the endpoints of the segment. If X is outside of this range, the extreme would not occur on a section of the parabola that is actually executed, and one of the end velocities would be the extreme velocity for the segment.

If X is found to be between 0 and +1, the next step is to calculate the extreme velocity:

$$Vextreme = Vs + [(3Vm - 2Vs - Vf)^2 / (6Vm - 3Vs - 3Vf)]$$

$$= Vs + [(3Vm - 2Vs - Vf)X]$$

This velocity would then be compared to system capabilities.

Example: Given
Vs = 1000 cts/sec
Vf = 2000 cts/sec
dP = 3000 counts
t = 1 sec

Then:
Vm = dP/t = 3000 cts/sec
X = (9000-2000-2000) / (18000-3000-6000)
= 5/9

Since X is between 0 and +1, a minimum or maximum velocity occurs during the move segment:

Vextreme = 1000 + [(9000-2000-2000)*5/9]
= 3777.78 cts/sec

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

S-Curve (Three-Point Smoother or CIRCLE4) Mode

The SMCC-P provides a special mode that automatically generates and combines parabolic move segments for smooth motion without the user having to specify velocities explicitly. This has several important uses. First, it can provide gentle starting and stopping with S-Curve acceleration, which can greatly reduce the strain on equipment and loads. Second, it can apply a three-point cubic fit to a series of positions received by the SMCC-P. This allows the SMCC-P to smoothly follow an arbitrary path with far fewer points than would be required if linear interpolation were used. This is particularly useful when working with complex shapes, as in cam grinding applications, where the number of points around the cam that need to be specified is an order of magnitude less with three-point smoothing than it would be with straight point-to-point interpolation.

This mode of operation is invoked by using the CIRCLE4 (CIR4) Command. Execution of this command automatically cancels the F (feedrate) or T (time move) the SMCC-P may have been using, and causes it instead to use the 't' value as the move segment time. To exit this mode, simply use a new F or T Command.

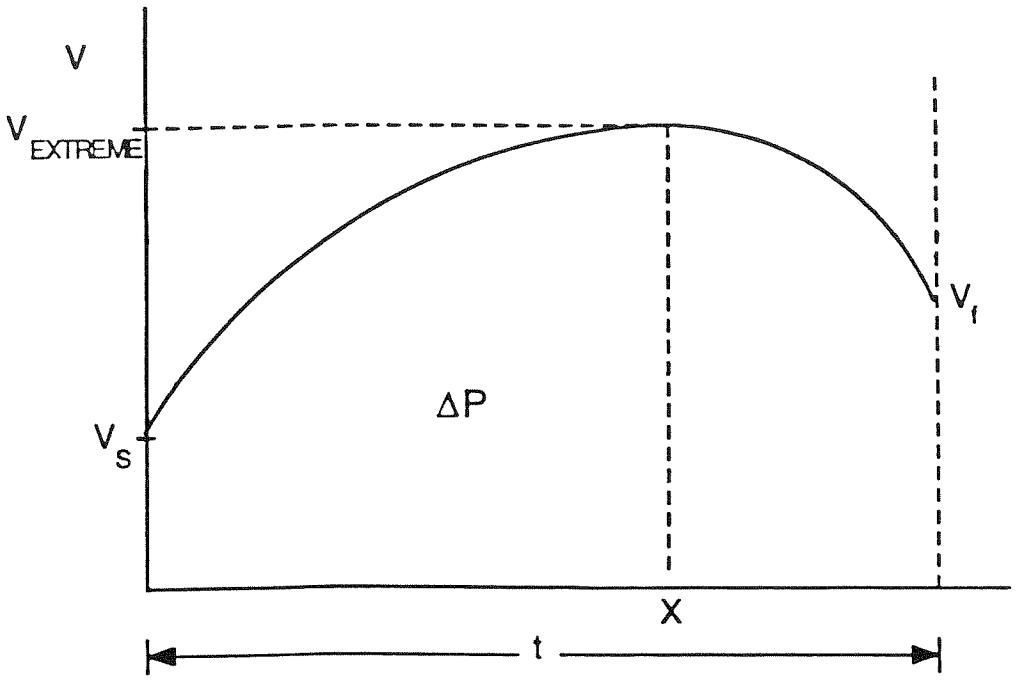
There are three basic rules for generating proper S-curve moves (these will become clear with examples):

1. If you change the time ('t') for a move segment, you must keep the ratio deltaP/t (change in position/time, or velocity) the same for the new segment as for the old segment.

2. To change velocity, keep 't' constant, and change the deltaP so that deltaP/t yields your new desired velocity. The SMCC-P will blend smoothly to the new velocity. If you change the deltaP/t ratio as you change the time t, there will be an instantaneous change in acceleration between segments (but not in velocity). Since this mode is usually used to avoid such acceleration jumps, this condition is undesirable.

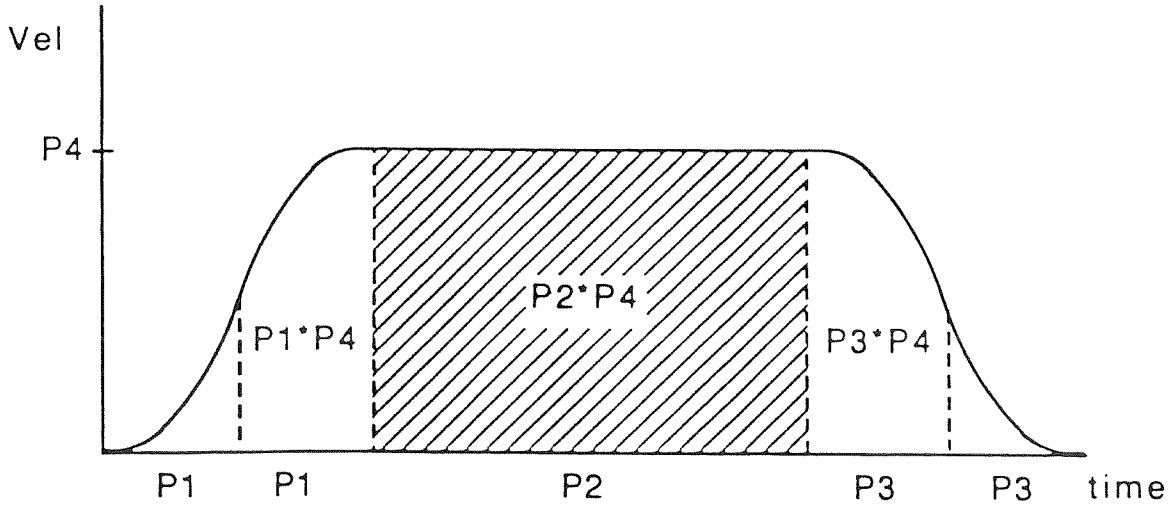
3. The SMCC-P will automatically add a move segment with a deltaP of zero in front of the first move segment you specify in a series, and also after the last segment. These move segments each take the same time 't' as the move segment to which they are appended; they do not actually appear in the program. (Do not explicitly add these to your program, or the SMCC-P will tack another segment to them.) These segments do not actually cause a zero move; as in rule 2 above, the SMCC-P provides a blending to the velocity specified by the next deltaP/t ratio.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



V_s = Starting Velocity
V_f = Final Velocity
V_m = Mean Velocity
dP = Distance to Move
t = Time to Move
X = Time of extreme vel
$V_{extreme}$ = Extreme vel

FIGURE 6. GENERAL PARABOLIC MOVE SEGMENT



CIR4
t(P1)
U(P1*P4)
t(P2)
U(P2*P4)
t(P3)
U(P3*P4)
END

FIGURE 7. POINT-TO-POINT S-CURVE MOVE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Below is a simple, general-purpose parameterized program for generating point-to-point moves with S-curve acceleration and deceleration:

```
CIR4            ;Invokes S-curve mode
t(P1)           ;Move segments take P1 binary milliseconds
U(P1*P4)        ;Make incremental X-axis move of P1*P4
                ;encoder counts
t(P2)           ;Time of P2 binary milliseconds
U(P2*P4)        ;Note that P4 factor keeps deltaP/t ratio
t(P3)           ;constant automatically
U(P3*P4)
END

Run velocity = P4 counts / binary millisecond
Total distance = (P1 + P2 + P3) * P4 counts
Total time = 2P1 + P2 + 2P3 binary milliseconds
```

The velocity profile generated by this program is shown in Figure 7. There are several important things to note about this simple program. First, since the time changes for each segment, the deltaP/t ratio must remain constant. This ratio is enforced by using the run velocity P4 to multiply by the time to give the deltaP for each segment. Second, note by looking at Figure 8 that the acceleration takes a time of 2*P1. This is due to the automatic addition by the SMCC-P of a U0 segment at the beginning of the move. By rule 2, since deltaP changes (from 0 to P1*P4) while t stays constant, the SMCC-P provides an S-curve blending from a velocity of 0 to a velocity of P4 in a time of 2*P1. The same is true (in reverse) for the deceleration. This makes the total time for the move equal to 2*P1 + P2 + 2*P3.

Figure 8 shows the general case for blending between two constant velocities in S-Curve mode. Suppose the system were moving at a constant velocity V1 and it were desired to change to a velocity of V2, and run constant at that velocity for a while. For a given t value (more on how to set this below), we must compute a deltaP1 so that deltaP1/t = V1 (deltaP1 = V1 * t), satisfying rule 1. Next we calculate deltaP2 so that deltaP2/t = V2 (deltaP2 = V2 * t), according to rule 2. If no blending were occurring, there would be a sudden jump in velocity after the first time t. But the S-curve mode provides a smooth transition between the two velocities by creating two parabolic segments, each of time t, the first one concave-up, the second concave-down.

Note that in doing this blending, the system will have moved by more than deltaP1 in the first time t, and less than deltaP2 in the second time t. However, it will move exactly deltaP1+deltaP2 in time 2t (provided the next move does not also require blending to a new velocity).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

General Algorithm for Point-to-Point S-Curve Moves

A common type of application for this mode is one that requires simple point-to-point moves with S-Curve acceleration for gentle starts and stops. It is desired to complete these moves in minimum time under the constraints of maximum allowed velocity and acceleration. Below is a general procedure for computing such moves.

First, look at the math for an S-curve acceleration. By symmetry, the area under the curve (the distance covered) is the same as for a straight-line acceleration. If the starting or ending velocity is zero this is just a triangular area, so the formula for distance covered is:

deltaP = (1/2) * 2t * V = V*t

where t is the time for each of the two parabolic segments, and V is the non-zero starting or ending velocity.

The top acceleration reached during an S-curve acceleration is exactly twice that of a constant (straight-line) acceleration:

Apeak = 2 * V / (2t) = V/t

Now if we define:

Vmax = Maximum allowed velocity (cts/sec)
Amax = Maximum allowed acceleration (cts/sec/sec)

we can compute the time value and distance required for the fastest acceleration to the maximum allowed velocity:

t = V/Apeak = Vmax/Amax (remember to convert to bin. msec.)
deltaP = V*t = Vmax*t

These values can be used for both the acceleration and deceleration portions of the move. For the constant speed (slew) portion, simply calculate:

deltaP(slew) = TotalDistance - 2*deltaP(accel)
t(slew) = deltaP(slew)/Vmax

Example: Suppose we had a system with Vmax = 50,000 cts/sec and Amax = 500,000 cts/sec/sec. We wish to do a point-to-point move of 25,000 counts in minimum time with S-curve accel and decel. First, we calculate the accel/decel 't' time as t = 50000/500000 = 0.1 sec. The accel/decel distance is deltaP = 50000 * 0.1 = 5000 counts. Next, we compute the slew distance as deltaP(slew) = 25000 - 2*5000 = 15000 counts and the slew time as t = 15000 / 50000 = 0.3 sec. The program for this move would be:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

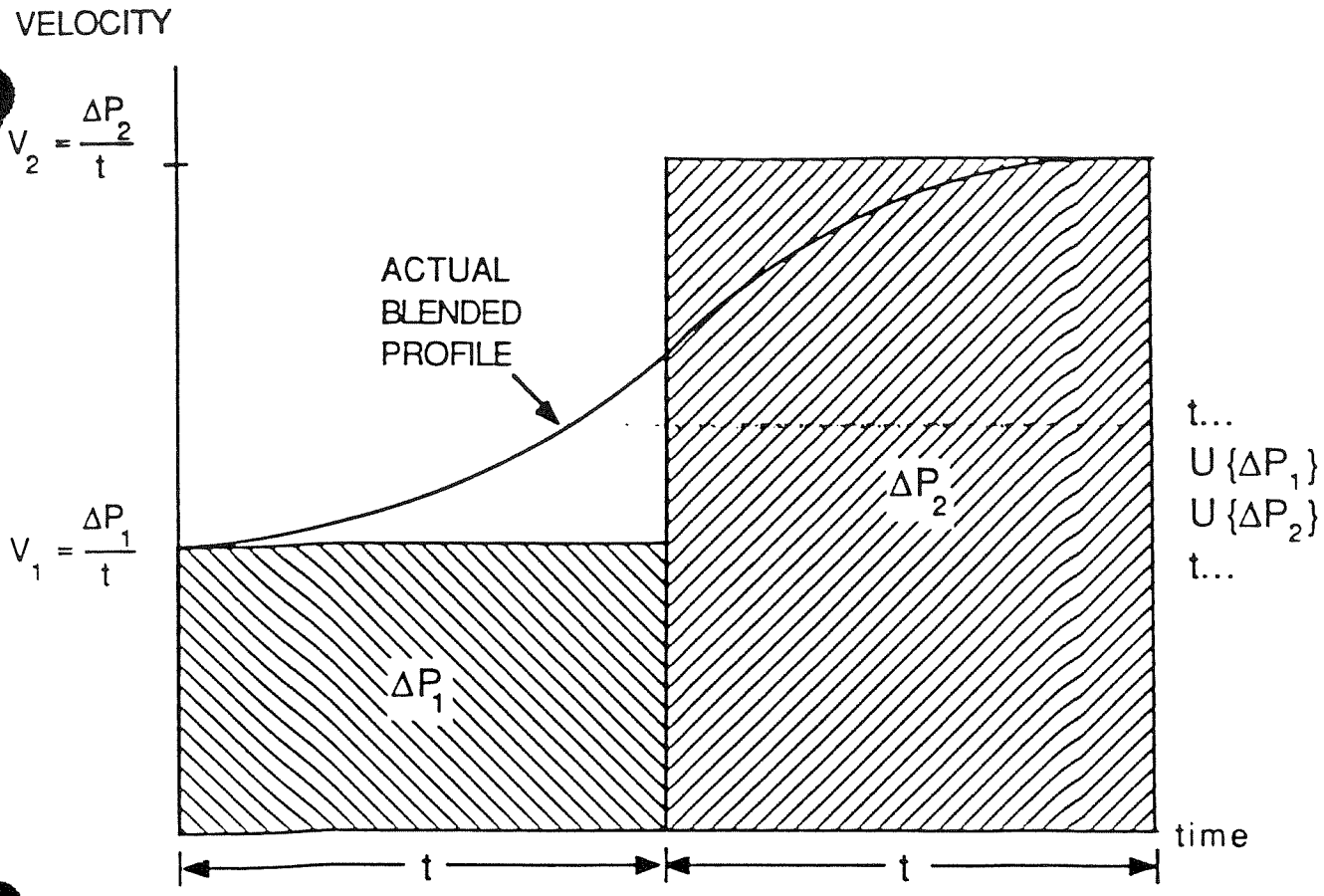


FIGURE 8. GENERAL S.-CURVE ACCELERATION

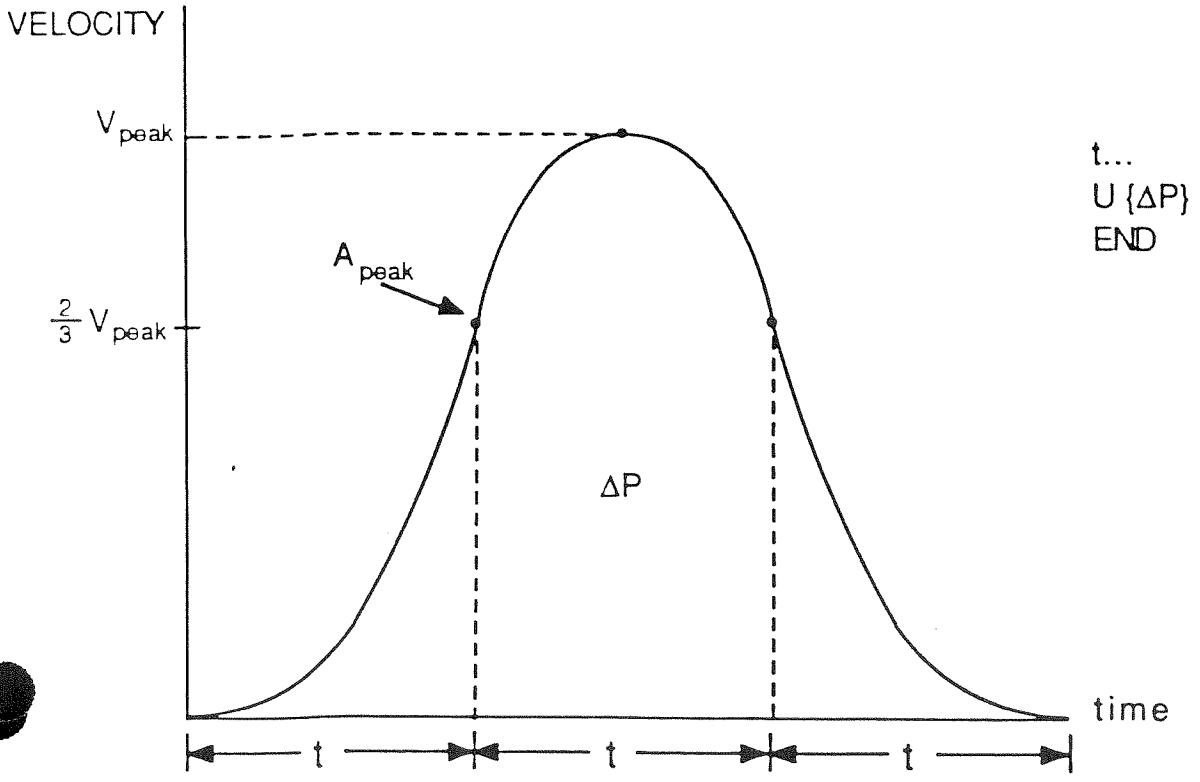


FIGURE 9. SINGLE PART S-CURVE MOVE

(S-CURVE)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
t102            ;0.1 sec in binary milliseconds
U5000           ;acceleration distance
t307            ;0.3 sec in binary milliseconds
U15000          ;slew distance -- note that ratio holds
t102            ;decel time = accel time
U5000           ;deceleration distance
END             ;or DWELL -- something to ensure stop
```

Of course, in order to do a three-part S-curve move in this manner, the total distance covered must be enough to allow this acceleration, deceleration, and a short slew. The minimum distance for which this is possible is:

```
deltaP(min3part) = 2 * Vmax*t + 0.020*Vmax
                 = 2 * (Vmax^2)/Amax + 0.020*Vmax
```

For the above example, this distance would be 11,000 counts. If the distance is shorter than this, it will be necessary to do a single-part S-curve move instead.

For a very short move, the profile will never get up to maximum allowed velocity, so maximum allowed acceleration is your only constraint for calculating a minimum-time move. Figure 9 shows a single-part S-curve move. Since there is only one position command in the move, the SMCC-P adds zero segments both before it and after it. The move takes 3*t total time to execute and the center segment blends toward zero velocity on both sides.

The peak acceleration on this move occurs right at the boundaries between segments. It is calculated by:

```
Apeak = deltaP / (t^2)
```

Setting Apeak to the maximum allowed acceleration, Amax, we can solve for the minimum-time t:

```
t = sqrt (deltaP/Amax)
```

Example: For the above system, a move of 5000 counts, which is too short for a three-part move, would have t = sqrt(5000/500000) = 0.1 sec. The program would be:

```
t102            ;0.1 sec -- total move time = 0.3 sec
U5000           ;incremental move total distance
END             ;or DWELL -- something to ensure stop
```

The maximum-length move that can be done with this calculation is the one in which the peak velocity equals the maximum allowed velocity. The equation for peak velocity is:

```
Vpeak = (3/4) * sqrt(Amax * deltaP)
```

Setting Vpeak equal to Vmax, and solving for deltaP, we get:

```
deltaP(max1part) = (16/9) * Vmax^2 / Amax
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

On our example system, this would be a move of 8888 counts. Note that this is somewhat less than the minimum three-part move, which is greater than 2 * Vmax^2 / Amax. For moves between these two limits, we must use a one-part move that is constrained by Vmax, and never reaches Amax. The equations for this type of move are:

```
deltaP = (4/3) * Vpeak * t
       = (4/3) * Vmax * t
so   t = (3/4) * (deltaP/Vmax)
```

Example: On the above system, a move of 10000 counts is too short for a three-part move, and too long for an acceleration-constrained one-part move. So we use the velocity constraint, and calculate t = (3/4) * (10000/50000) = 0.15 sec. Total move time is 0.45 sec.

These calculations can be done either in a host computer or on the SMCC-P. Example #47 in the Programming Examples section shows how these calculations can be done on the card itself. This is a sophisticated example, converting absolute positions in two axes to incremental distances, deciding which axis has the longer move, then deciding which of the three classes of S-curve moves it falls into, and calculating the time and distance parameters.

Three-Point Smoothing (Cubic Fit) Applications

Another type of application for which this mode is valuable is in providing irregular curve following capability while smoothing the trajectory between adjacent points on the curve. The SMCC-P simply requires a series of positions and the time to move between the positions. The SMCC-P, once in CIRCLE4 mode, takes each data point in turn, and does a cubic fit with it, the point before it, and the point after it -- called three-point smoothing. This process is done in real time, and it is exactly the same process as described above (just looked at slightly differently).

This smoothing may take the calculated path slightly away from the actual specified points, particularly if the velocity changes are sudden (because of the blending action). But generally, this mode allows the motion system to follow the desired position profile very closely with far fewer points than with straight-line interpolation, and most importantly, it does so without any sudden acceleration changes ('jerk') at point boundaries that could excite undesirable system dynamics.

For example, this mode is very useful for retracing a shape that has been digitized. The time 't' between digitized points would be specified, and the SMCC-P would follow the path between points, passing each new one at time 't' past the previous one.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

For this type of application, we look at the equations that derive the derived commanded positions from the numbers in the program (to repeat, the math in the SMCC-P is the same as in S-curve type applications; this is just a different way of looking at it). For a series of programmed positions r(n):

```
3-point derived pos (n) = [r(n-1) + 4*r(n) + r(n+1)] / 6
3-point derived vel (n) = [r(n+1) - r(n-1)] / 2t
```

Consider the following short program to fit a series of points:

```
X=0              ;Define current position to be zero
CIR4             ;3-point smoothing mode
t102             ;Time between points is 0.1 sec
X1000            ;Move to X=1000 (deltaP = 1000)
X2000            ;Move to X=2000 (deltaP = 1000)
X2500            ;Move to X=2500 (deltaP = 500)
END
```

The SMCC-P will automatically add zero-distance moves at the beginning and end of the sequence (rule 3). This makes the move data sequence:

```
0, 0, 1000, 2000, 2500, 2500
```

The derived positions and velocities at time intervals 't' are:

```
1)   p(1) = [0 + 4*0 + 1000]/6 = 167 cts
     v(1) = [1000 - 0]/[2*0.1] = 5000 cts/sec

2)   p(2) = [0 + 4*1000 + 2000]/6 = 1000 cts
     v(2) = [2000 - 0]/[2*0.1] = 10000 cts/sec

3)   p(3) = [1000 + 4*2000 + 2500]/6 = 1917 cts
     v(3) = [2500 - 1000]/[2*0.1] = 7500 cts/sec

4)   p(4) = [2000 + 4*2500 + 2500]/6 = 2417 cts
     v(4) = [2500 - 2000]/[2*0.1] = 2500 cts/sec

5)   p(5) = [2500 + 4*2500 + 2500]/6 = 2500 cts
     v(5) = [2500 - 2500]/[2*0.1] = 0 cts/sec
```

As explained before, the 't' time does not need to remain constant, but if it is changed, the ratio of deltaP to 't' must stay fixed (rule 1), or there will be discontinuities in velocity and acceleration, defeating the purpose of this mode of operation.

With some preprocessing in the host computer, it is possible to do a 5-point smoothing function (5th-order polynomial blend) for greater blending capability. From a series of points r(n), we calculate in the host computer intermediate points q(n):

```
q(n) = [-r(n-1) + 8*r(n) - r(n+1)]/6
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

These points q(n) are used in the SMCC-P CIRCLE4 mode program, which computes derived points p(n) as above:

$$p(n) = [q(n-1) + 4*q(n) + q(n+1)]/6$$

The net formula for the derived points p(n) from the original points r(n) is:

$$p(n) = [-r(n-2) + 4*r(n-1) + 30*r(n) + 4*r(n+1) - r(n+2)]/36$$

As a more sophisticated example, assume that we wish to make a square shape out of a rotating part, with the cutting tool moving radially in and out. With the square 100 units on a side, the equations for the radial location of the cutter are:

$$r = 50 / \cos(a) \qquad 0 \text{ deg} <= a <= 45 \text{ deg}$$
$$r = 50 / \sin(a) \qquad 45 \text{ deg} <= a <= 90 \text{ deg}$$

Symmetrical equations apply for the other four quadrants of the square.

If the controller were only capable of point-to-point moves, at least 250 points per 90 degrees would be required in order to produce a smooth shape without noticeable flats between points. However, if cubic interpolation is used, the number of points required can be significantly reduced. This can ease memory requirements, and also speed up execution if the number of move segments (blocks) per second is a limiting factor.

Below are tables showing the results of SMCC-P's CIRCLE4 mode calculations for this problem. Each table shows the actual points on the square, the three-point fit to those points with the associated error, and the five-point fit with its error. Table 1 shows the data for 10 points in 90 degrees, Table 2 shows 20 points in 90 degrees, and Table 3 shows 40 points in 90 degrees.

	Actual r	3-Point r	3-Point error	5-Point r	5-Point error	
0	50.0000	50.2078	0.2077	49.9956	-0.0045	
1	50.6233	50.8444	0.2211	50.6181	-0.0052	
2	52.5731	52.8387	0.2656	52.5652	-0.0079	
3	56.1163	56.4736	0.3573	56.1017	-0.0146	
4	61.8034	62.3401	0.5367	62.4176	0.6142	
5	70.7107	67.7416	-2.9691	69.5421	-1.1686	---- Max. error
6	61.8034	62.3401	0.5367	62.4176	0.6142	
7	56.1163	56.4736	0.3573	56.1017	-0.0146	
8	52.5731	52.8387	0.2656	52.5652	-0.0079	
9	50.6233	50.8444	0.2211	50.6181	-0.0052	
10	50.0000	50.2078	0.2078	49.9956	-0.0045	

Data for 10 points in 90 degrees

---------- TABLE 1 ----------

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

	Actual r	3-Point r	3-Point error	5-Point r	5-Point error	
0	50.0000	50.0515	0.0515	49.9997	-0.0003	
1	50.1546	50.2070	0.0523	50.1543	-0.0003	
2	50.6233	50.6781	0.0548	50.6229	-0.0003	
3	51.4208	51.4799	0.0591	51.4204	-0.0004	
4	52.5731	52.6388	0.0657	52.5726	-0.0005	
5	54.1196	54.1946	0.0750	54.1190	-0.0006	
6	56.1163	56.2044	0.0881	56.1155	-0.0008	
7	58.6414	58.7475	0.1062	58.6402	-0.0012	
8	61.8034	61.9349	0.1315	61.8016	-0.0018	
9	65.7544	65.9219	0.1676	66.0636	0.3093	
10	70.7107	69.0586	-1.6521	70.1041	-0.6066	---- Max. error
11	65.7543	65.9219	0.1676	66.0636	0.3093	
12	61.8034	61.9349	0.1315	61.8016	-0.0018	
13	58.6414	58.7475	0.1062	58.6402	-0.0012	
14	56.1163	56.2044	0.0881	56.1155	-0.0008	
15	54.1196	54.1946	0.0750	54.1190	-0.0006	
16	52.5731	52.6388	0.0657	52.5726	-0.0005	
17	51.4208	51.4799	0.0591	51.4204	-0.0004	
18	50.6233	50.6781	0.0548	50.6229	-0.0003	
19	50.1546	50.2069	0.0523	50.1543	-0.0003	
20	50.0000	50.0515	0.0515	49.9997	-0.0003	

Data for 20 points in 90 degrees

---------- TABLE 2 ----------

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

	Actual r	3-Point r	3-Point error	5-Point r	5-Point error	
0	50.0000	50.0129	0.0129	50.0000	-0.0000	
1	50.0386	50.0515	0.0129	50.0386	-0.0000	
2	50.1546	50.1677	0.0131	50.1546	-0.0000	
3	50.3490	50.3623	0.0133	50.3490	-0.0000	
4	50.6233	50.6369	0.0137	50.6232	-0.0000	
5	50.9796	50.9937	0.0141	50.9795	-0.0000	
6	51.4208	51.4355	0.0148	51.4207	-0.0000	
7	51.9505	51.9660	0.0155	51.9504	-0.0000	
8	52.5731	52.5895	0.0164	52.5731	-0.0000	
9	53.2940	53.3115	0.0174	53.2940	-0.0000	
10	54.1196	54.1383	0.0187	54.1196	-0.0000	
11	55.0574	55.0776	0.0202	55.0574	-0.0000	
12	56.1163	56.1383	0.0219	56.1163	-0.0000	
13	57.3069	57.3309	0.0240	57.3068	-0.0001	
14	58.6414	58.6678	0.0264	58.6413	-0.0001	
15	60.1345	60.1638	0.0293	60.1344	-0.0001	
16	61.8034	61.8361	0.0327	61.8033	-0.0001	
17	63.6686	63.7053	0.0368	63.6684	-0.0001	
18	65.7544	65.7960	0.0416	65.7542	-0.0002	
19	68.0900	68.1375	0.0475	68.2445	0.1545	
20	70.7107	69.8371	-0.8736	70.4036	-0.3070	---- Max. error
21	68.0900	68.1375	0.0475	68.2444	0.1545	
22	65.7543	65.7960	0.0416	65.7542	-0.0002	
23	63.6686	63.7053	0.0368	63.6684	-0.0001	
24	61.8034	61.8361	0.0327	61.8033	-0.0001	
25	60.1345	60.1638	0.0293	60.1344	-0.0001	
26	58.6414	58.6678	0.0264	58.6413	-0.0001	
27	57.3068	57.3308	0.0240	57.3068	-0.0001	
28	56.1163	56.1382	0.0219	56.1163	-0.0000	
29	55.0574	55.0776	0.0202	55.0573	-0.0000	
30	54.1196	54.1383	0.0187	54.1196	-0.0000	
31	53.2940	53.3115	0.0174	53.2940	-0.0000	
32	52.5731	52.5895	0.0164	52.5731	-0.0000	
33	51.9505	51.9659	0.0155	51.9504	-0.0000	
34	51.4208	51.4355	0.0148	51.4207	-0.0000	
35	50.9795	50.9937	0.0142	50.9795	-0.0000	
36	50.6233	50.6369	0.0137	50.6232	-0.0000	
37	50.3490	50.3623	0.0133	50.3490	-0.0000	
38	50.1546	50.1677	0.0131	50.1546	-0.0000	
39	50.0386	50.0515	0.0129	50.0386	-0.0000	
40	50.0000	50.0129	0.0129	50.0000	-0.0000	

Data for 40 points in 90 degrees

---------- TABLE 3 ----------

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Examination of the results shows that the maximum error occurs at the sharp corner of the square. This is to be expected, because traversing the corner at speed would require a path with infinite acceleration, something the blending function of the three point smoother is designed to prevent. Specifying a radius at the corner would significantly reduce the path error here. Other than at the corner, the errors were very small. As could be predicted, the five-point smoothing did better than the three point smoothing, and increasing the number of points reduced the errors. At 40 points per 90 degrees, the errors are vanishingly small except at the corners, so further increase would improve nothing -- and we are still way below the 250 points per 90 degrees required by point-to-point interpolation.

3) ACCELERATION AND VELOCITY FEEDFORWARD

SMCC-P has these two additional servo parameters in order to improve the dynamic response and reduce the following error caused by acceleration and velocity. The Standard and Parabolic SMCC servo algorithms are shown for reference.

SMCC SERVO ALGORITHMS:

SMCC WITH PARABOLIC OPTION

$$\text{Eout DAC} = i20*k0*\{[EP(N) - i21[AV(N)*(dT/[T(N) - T(N-1)])]] + i22*DV(N) + i30*k1*[DV(N) - DV(N-1)] + i23*k2*\sum_{j=0}^{n-1} EP(j)/\}$$

NOTE: The term (dT/T(N) - T(N-1) is activated only when 1/T (period) encoder count measurement is used.

Where:
- i20 = Proportional Gain
- i21 = Differential Gain (Feedback)
- i22 = Velocity Feedforward Term
- i30 = Acceleration Feedforward Term
- i23 = Integral Gain
- i09 = Integrator Mode Control
- EP(N) = Position Error
- EP(J) = Summation of Previous Errors
- AV(N) = Actual Velocity = P(N) - P(N-1)
- DV(N) = Desired Velocity = DP(N) - DP(N-1)
- (N-1) = Previous event
- k0 = 1.25V/65536
- k1 = 4
- k2 = 1/4096

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC STANDARD (NON-PARABOLIC)

$$Eout = i20 * k0 * \{EP(N) - i21 * k1 * [EP(N-i22-1)]/256 + i23 * k2 * \sum_{j=0}^{n-1} EP(j)/65536\}$$
DAC

Where:
- i20 = Proportional Gain
- i30 = Proportional Gain Multiplier 0 = x1, 1 = x256
- i21 = Differential Gain
- i22 = Velocity Smoother (Filter time constant)
- i23 = Integral Gain
- i09 = Integrator Mode Control
- EP(N) = Position Error
- EP(j) = Sum of All Position Errors
- k0 = 1.25V/65536 for i30 = 0
  = 1.25V/256 for i30 = 1
- k1 = 1/256
- k2 = 1/65536

4) AC INDUCTION SERVO

With the new CPU mask, the servo calculations for the AC induction motor phasing were moved from external to internal code, and thus made faster, which results in smoother and stiffer servo capability.

5) 1/T ENCODER PERIOD

Normally SMCC measures the number of encoder counts received during each servo cycle. At low velocities the number of encoder counts received per servo cycle is zero or very low, causing granularity and errors in actual velocity calculations. SMCC-P provides the digital tachometer (1/T) mode which uses time measurement of position changes for effective fractional resolution. The time measurement uses a 2 MHz clock and the 16-bit handwheel encoder counter to accumulate the time elapsed from one count to the next; this provides an accuracy of +/-0.5 microseconds and a maximum range of 32 milliseconds between pulses. The result of using 1/T mode is that low velocity performance and smoothness is improved by at least a factor of two or three, which allows higher gains to be used.

Since there is system dependence on the number of encoder counts per revolution for low velocity smoothness, good general rules are:

A) Use as high an encoder count/revolution as possible while keeping in mind that the high speed requirements should not exceed the maximum count rate of the encoder or the controller's position counts.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

B) Use at least 100 to 200 counts/second at the lowest RPM you will use in order to avoid velocity granularity.

C) Without 1/T use at least 300 counts/second for equivalent operation.

Lower encoder counts/second may be used if more velocity ripple can be tolerated or servo gain can be reduced for smoother velocity at the expense of stiffness.

i28[i48] is used to enable the 1/T mode and jumpers E40 and E41 must be removed so the handwheel counters can be used by the 1/T mode. Handwheel inputs are disabled and cannot be used when in 1/T mode.

6) DEADBAND

Parameters i60[i61] are now used to specify a servo deadband of 0 to 255 encoder counts. SMCC-P will maintain a zero output to the amplifier when the motor position is within +/- 0 to 255 counts of the desired position. When the following error exceeds the specified deadband limit, the SMCC-P begins to control position again. Note that amplifier's offset may drive the motor shaft to one end of the specified deadband limit. The amplifier is not disabled inside deadband.

7) DATA GATHERING CAPABILITY

The Parabolic SMCC has the ability to collect and store servo data in real time. This can be used to provide a "snapshot" of system performance during a period of interest. The servo data can come from any section of the card's memory space (it must be a continuous section of memory), but the most common data collected is position and following error. This data can then be sent to a host computer for further processing, such as graphic display.

The SMCC Terminal Program (Version 3.6 and newer) for the IBM PC is designed to facilitate this process. It has a section that allows the user to specify interactively how the data is to be gathered, to cause it to be collected and brought back to the computer, and to be plotted, printed, or displayed. Instructions for this section of the program are shown below.

The process of collecting and retrieving data requires coordination between the SMCC and the host computer, and between on-line commands and the motion program.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The SMCC-P command to set up the data gathering is:

```
     y  f ,  b ,  a ,  d1 ,  d2
```

where

y is the command itself;
f defines the sampling period (f is the log [base 2] of the sampling period in servo cycles;)
b is the number of bytes stored each cycle (1 - 32);
a is the address of the first byte to be collected each cycle (default is fe48 -- X position);
d1 is the starting address of the storage buffer (default is the address following the program buffer);
d2 is the ending address of the storage buffer (default is the end of the open memory).

On receiving this command, the SMCC-P will prepare for data storage and return the address of the pointer to the data storage buffer (in hex). Before any storage, this will be the beginning of the buffer. (A 'y' Command sent without any arguments will do nothing but return the address of the pointer. This can be used after gathering the data to find the end of the buffer used, and therefore, the length of the buffer.)

The program in the card must set and clear in the fifth bit of the byte at address febd to start and stop the data gathering. This is usually done in the following manner:

```
SET C20 = febd
SET M20 = C20.5
. . .
(instructions before gathering)
SET20
. . .
(instructions while gathering)
. . .
RESET20
(instructions after gathering)
```

After the data collection on the card has been completed, the host should send a 'y' Command without arguments to get the address at the end of the buffer. Then the 'd' Command is sent, with two arguments: first is the starting address (in hex), and the second is the length of the buffer (end address minus starting address) in decimal. See the description of the 'd' Command in the programming manual section for details on the command, and the format of the returned data.

The amount of storage space the SMCC-P has for data gathering depends on the size of your program buffer. For example, if you have a one hundred line program with no subroutines and no background programs, you will have approximately 6K of data storage space, so that if the data gathered is 16 bytes every 128 seconds = 48 seconds (approximately). When SMCC's Option 5 (32K memory) is used, about four times as much can be stored.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC TERMINAL PROGRAM DATA GATHERING INSTRUCTIONS

Once you have entered this section by pressing <ALT F10> in the SMCC Terminal main menu, you are presented with the SMCC plotter screen. Typing alphanumeric keys still lets the left window work like a standard terminal. In the right window you see the sub-menu of function key operations for this section. The menu appears as:

F1 - COLLECT DATA
F2 - COLLECT/STORE
F3 - GET DISK DATA
F4 - PLOT DATA
F5 - DISP RAW DATA
F6 - DISP DEC DATA
F7 - PRINT RAW DATA
F8 - PRINT DEC DATA
F9 - SYS COMMAND
F10 - PLOT HELP

The instructions for each of these function keys are detailed below:

<F1> - COLLECT DATA

Pressing F1 causes the SMCC program to check the card you are connected to and ensure that the card is one with the Parabolic option (Option 4). If it is not, you may not proceed any further. If it is a parabolic card, the program will ask you the period of data collection you wish to use. Your choices are in numbers of servo cycles (the default value for a servo cycle is 976 microseconds). Your choices are displayed as:

<0> 1 <1> 2 <2> 4 <3> 8
<4> 16 <5> 32 <6> 64 <7> 128

Choose the period you want by pressing your selection of 0 to 7. For example, choosing <4> would give you a period of 32 servo cycles, or 32 x 0.976 = 31.232 milliseconds per sample.

Next, the program asks you to select the type of data you wish to collect. Your choices are displayed as:

<1> X Pos. <2> X & Y Pos. <3> X Pos. & FE
<4> (X & Y) (Pos. & FE) <5> Special Data
(Vel. & Accel. data computed from position)

Enter your choice by pressing your selection of 1 to 5. If you chose one of the standard selections (n = 1 to 4), there will be 4 x n bytes collected each sample, since each value is a 4-byte number. As noted in the display of the choices, if position data is acquired for an axis, the program will automatically compute velocity and acceleration data (using forward-backward difference).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If you press <5> for special data gathering, the program will ask you for the number of bytes to store each sample. You can store from 1 to 32 contiguous bytes each sample. Next, you will be asked for the capture address, which is the address of the first byte to be collected (in the SMCC processor's memory space).

At this point, you are ready to run a program in SMCC and have it collect data for you while it runs the program. The SMCC terminal program tells you to command the card in terminal mode to perform your desired actions. When you are done performing the moves and storing data, you type CTRL-Z, and the data will be brought back into the computer.

<F2> - COLLECT/STORE

This operates identically to F1, except that it also stores the data in a disk file. You will be asked for the name of the disk file before you start the data collection.

<F3> - GET DISK DATA

This section allows to you to bring in previously collected data that had been stored on disk (see instructions under F2 and F1). You will be asked for the name of the disk file from which you wish to retrieve the data. Once the data has been brought into the computer, you may use it just as if it had been brought in directly from the card. You do not need to be connected to a card to get the data in this manner.

<F4> - PLOT DATA

This section lets you view the collected data in graphical form. You must have graphics capabilities on your PC to utilize this function. You have a great deal of choice in what data you plot, and how you plot it. You will be presented a list of choices in the left window:

```
<0> Change Plot Settings
<1> X Position vs Time
<2> X Velocity vs Time
<3> X Acceleration vs Time
<4> X Following Error vs Time
<5> Y Position vs Time
<6> Y Velocity vs Time
<7> Y Acceleration vs Time
<8> Y Following Error vs Time
<9> Special Plots
Enter your choice (ESC to exit):
```

Items <4> through <8> will only be displayed if the data for that choice has been collected.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Selecting an item <1> through <8> will cause that plot to be displayed on the screen, replacing the text screen you were using. Once in the graphics screen, typing 'P' or 'p' will cause the plot to be copied to a printer (which must be an Epson-compatible graphics-capable printer in the standard printer port). Hitting any other key will cause a return to the text screen.

If you select <0>, you will be presented with a series of questions for setting variables that control how the data is plotted, including what units will be used (which affects the tabular output as well -- see <F6> and <F8>), where the legend is located, and whether the commanded or actual data is shown.

First, you will be presented with the selection of units for time, looking much as below:

```
  Time units:
<0> (User defined)
<1> Servo Cycs
<2> Sample Pds
<3> Seconds
<4> Minutes
Current setting is <3>: Seconds.
Enter you choice (return to keep):
```

Here you enter a number from 0 through 4 to make your selection. If you choose <0> to define your own time units, you will be asked for the name of your units, and then for the number of seconds in one of your units (e.g. you could define millisecs as having .001 seconds).

Following this, you will be presented with similar selection options for position units, velocity units, and acceleration units, where the choices work identically.

Next you will be asked if you wish to show the actual data read from the encoder, or the commanded data created by the trajectory generator. You must have following error data to have the choice of command data (actual position + following error = command position). If you have set this variable to show command data, and you then bring in data without following error, the variable will automatically be set back to show actual data.

After this, you will be asked if you want to display a graticule superimposed on your plot (for aid in measurement), then in which corner of the plot you wish to show the legend. Finally, you will be returned to the Plot Options menu.

If you select item <9> from the Plot Options menu, you can build your own plot from the variables available. You will be presented with a list of these variables, some or all of the following, depending on what has been brought into the computer:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

<0> Time
<1> X Position
<2> X Velocity
<3> X Acceleration
<4> X Following Error
<5> Y Position
<6> Y Velocity
<7> Y Acceleration
<8> Y Following Error
<9> Last Special Plot

First, you will be asked which of these variables you wish to plot on the horizontal axis (all of the standard plots use time). Indicate your choice by entering the number of the choice. Second, the program will ask you which variable you wish to plot on the vertical axis (time is not available as a choice on the vertical axis). Enter the number of your choice. Entering <9> will give you the last special plot you defined, without any further questions.

Next it will ask you if you wish to plot a second variable vertically against the horizontal variable. If you answer positively ('Y' or 'y'), it will ask you what the second vertical variable is, and again you enter the number of your choice.

If your two vertical variables are of the same type of unit (e.g. both velocity, or position and following error), you will be asked how you wish to do this double plot. The choices are:

<0> Plot curves separately, same scale
<1> Plot curves separately, diff. scales
<2> Plot sum of curves
<3> Plot difference of curves

Choosing <0> will produce a plot with two curves plotted with respect to the same vertical scale. Choosing <1> will show two curves, the first plotted with respect to the left vertical scale, the second with respect to the right vertical scale. A choice of <2> will yield a single curve that is the sum of the two vertical variables. A choice of <3> will show one curve with the second variable subtracted from the first.

If the two vertical variables are of different types of units, they will automatically be plotted as separate curves on different scales.

If you have collected data using the SPECIAL DATA option, and the variables were four bytes each, you may be able to use this routine profitably, although the titles will be misleading. You may have to experiment with the unit scaling to get the output you desire.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

<F5> - DISPLAY RAW DATA

This key lets you see the collected data in 'raw' form, that is, just as it was gathered from the card, before it was separated into variables and scaled. For some SPECIAL DATA sets, this may be the only meaningful way to examine the data. It is shown in hexadecimal form with a line looking like this:

a4b4 e0 c8 13 00 20 00 00 00 80 c7 13 00 40 00 00 00

The first four digits show the starting address of the line in the data storage buffer of the card. Each pair of digits following is the hexadecimal representation of a byte of memory. For the standard collected variables of position and following error, these are grouped in sets of four bytes (least significant byte first) to form a 32-bit value.

NOTE: This routine must re-access the card or disk from which the data came, so the source must be available to the program throughout the routine.

The data is displayed one screen (twenty-three lines) at a time in the left window. At the bottom of the right window are shown the keys that allow you to move between screens of data. The keys are:

PAGE DOWN - Show the next page of data
PAGE UP - Show the previous page of data
END - Show the last page of data
HOME - Show the first page of data
ESC - Exit this display routine
(Any other key) - Same as PAGE DOWN

<F6> - DISPLAY DEC DATA

This key allows you to view the data on the screen in formatted and scaled form. Since this often requires the full screen width, it will be displayed without the standard frame. First, a page of instructions will be displayed in the left window. When you proceed, the data will be shown one screen (twenty-three samples, or lines) at a time, with a title row and a units row at the top of the screen. The keys to control movement between screens of data are the same as those for the display of raw data, but there will not be a menu continually shown, because the data takes up the full screen.

The units and scaling of the variables are the same as for plotting. These can be changed in the 'Plot Data' section (see <F4>, option <0> above).

If you have collected data using the SPECIAL DATA option, and the variables were four bytes each, you may be able to use this

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

routine profitably, although the titles will be misleading. You may have to experiment with the unit scaling to get the output you desire.

<F7> - PRINT RAW DATA

This key allows you to print the data you have collected in its original unformatted form to an ASCII printer at the standard printer port. The printout can be aborted by hitting the ESC key during the print. For more details see <F5>, above.

NOTE: This routine must re-access the card or disk from which the data came, so the source must be available to the program throughout the routine.

<F8> - PRINT DEC DATA

This key allows you to print the data you have collected in formatted and scaled form to an ASCII printer at the standard printer port. The printout can be aborted by hitting the ESC key during the print. The units and scaling are governed the same way as in the display of formatted data (see <F6> directions, above).

<F9> - SYS COMMANDS

This key allows you to perform DOS system commands without actually exiting the program. It is particularly useful here to examine directories for disk files. You can return to the program by hitting <ENTER> or <ESC> at the command prompt.

<F10> - PLOT HELP

This key lets you view a help screen that is a brief summary of the instructions in these pages.

8) DMA COMMUNICATIONS

SMCC-P provides a DMA (Direct Memory Access) Communications mode in order to allow rapid transfer of data to and from the host computer.

The DMA mode is always potentially active. Parameter i19 (Address of Last Card), whose default value is "0" and can assume a value of "0" to "f", is used by the cards during data reception and the automatic answer back cycle to receive and send "packet" data at high rates of four to five microseconds per character. Card addresses need not be consecutive but the "empty slots" will waste communications time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

All communications in DMA mode are in binary format and consist of either X and Y parabolic commands with position and velocity, or non-parabolic, position only moves including the CIR4 three-point smoothed data mode. To select Parabolic (position + velocity) versus Standard (position only) mode, use the i63 parameter as follows:

i63 0 selects the Parabolic mode
i63 1 selects the Standard mode

Note that position or status data sent from SMCC-P to host are not affected by i63.

A. DMA, Host to SMCC-P

Data can be transmitted via RS232 or parallel port by the host and may be either placed in the program buffer of SMCC-P or be executed on-line by placing SMCC-P in the MDI mode (i02 set to 1).

To start a DMA transmission assuming Parabolic position mode: position and velocity velocity data (i63=0), the host must first send a <CNTRL B> character (02H) followed by the move data for card A0 which consists of 14 bytes of data per card and represents:

4 Bytes of X Position Data
3 Bytes of X Velocity Data
4 Bytes of Y Position Data
3 Bytes of Y Velocity Data

Data format is binary LSB first, MSB last. The position data for both X and Y must be padded with five zeros in the LSB word (32 times greater) since the actual position data has 27 bits of data only. Upon completion, data for A0, data for A1, and so on is transmitted as required. The transmission is terminated when all the data has been transmitted. If the MDI mode is used (i02 set to 1), then the motion will begin upon reception of the data. If i02 is set to zero (buffer mode), the data will be placed in the buffer and may later be executed by Step "S" or Run "R" Commands. Data transmission to the card in the buffer mode may be mixed with both binary (DMA) and ASCII (non-DMA) data.

The total transmission in i63=0 mode consists of:

<CNTRLB> + (14 Bytes x Number of Cards)

In i63=1 mode, standard data with position only, velocity data is not sent, and the total transmission will consist of:

<CNTRL B> + (8 Bytes x Number of Cards)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The data transmission rate with parallel port is approximately five microseconds per character. RS232 transmissions occur at the selected baud rate, the fastest of which is 38,400 baud or approximately 250 microseconds per character. These rates apply to both reception and transmission of data. All DMA transmissions generate an internal interrupt with a latency of approximately eight microseconds.

B. DMA, Binary SMCC-P, to Host

To receive position data back from SMCC-P, the host computer issues a <CNTRL C> (03H) character. Upon reception of <CNTRL C>, the A0 card immediately begins transmitting binary data consisting of four bytes of X position data and four bytes of Y position data LSB first, MSB last. The SMCC in DMA mode generates an interrupt request internally which overrides the servo cycle in order to send data back to the host. The time needed to gather the data and begin transmission to the host is approximately eight microseconds. Each of the four bytes has five zeros padding the LSB byte so that the data the data is 32 times larger than the actual position data.

The host must receive the data and after eight bytes of data have been received, it also must issue a "space" character (or any other character) to cause card A1 to transmit its eight bytes of data. If must continue to do this until all cards have transmitted their data.

To request status data back from SMCC-P the host computer issues a <CNTRL D> (04H) character. Upon reception of <CNTRL D>, the A0 card immediately begins transmission of its eight bytes of status words. The host then sends a "space" or any other character back to SMCC-P in order to receive eight status bytes from the next card. It continues until all cards have sent their status back. See Paragraph 9 for the contents of the status word.

To request the P1 and P2 parameters back from SMCC-P, the host computer issues a <CNTRL E> (05H) character. The A0 card immediately begins transmission of eight bytes: first, four for P1, then four for P2 (least significant byte first for each), straight binary coded. The host sends a character to the SMCC's, the A1 sends eight bytes, and so on. (The SMCC-VME sends P1-P4 --16 bytes-- from each card.)

The difference in transmission of status between the DMA binary mode and the ASCII mode is that in ASCII each byte of status is sent as two 4 bit nibbles each; in hex ASCII format with most significant nibble first. Total status transmission is therefore 16 characters in ASCII mode. Also, the time needed between reception of command and transmission of data is in the order of four milliseconds and in addition each card must be addressed individually and then requested to send data back with the "?" Command.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

9) STATUS REQUEST

The SMCC-P transmits eight bytes of status information instead of four. Status transmission is initiated by the host sending "?" and <CR> characters in ASCII mode or by sending a <CNTRL D> character in binary DMA mode.

PARABOLIC STATUS BYTES

Bit	feb5	feb6
0	Home Complete (Y)	Time Mode (not feedrate)
1	Home Complete (X)	Single Step
2	Continuous Move	Buffer Full
3	Block Request (MDI mode)	Parabolic Mode
4	Auto Divide	Limit Switch
5	Feed Hold	Not Ready (X)
6	Running (any move)	Not Ready (Y)
7	Timer Active	Not Ready For Move

Bit	feb7	feb8
0	(X) Jog Pos.	(Y) Jog Positive
1	(X) Jog Negative	(Y) Jog Negative
2	(X) Limit Pos. (soft or hard)	(Y) Limit Pos. (soft or hard)
3	(X) Limit Neg. (soft or hard)	(Y) Limit Neg. (soft or hard)
4	(X) Follow. Err.(i10 exceeded)	(Y) Foll. Err. (i10 exceeded)
5	(X) I2T Limit (low true)	(Y) I2T Limit (low true)
6	(X) Not In Position	(Y) Not In Position
7	(X) Zero Velocity (commanded)	(Y) Zero Velocity (commanded)

Bit	feb9	feba
0	Edit Mode	Auto Divide Req.
1	Insert Mode	Extended Move Flag
2	Extended Read	Extended Move Flag
3	Circle Mode	Home Has Been Done Flag
4	Wait (X)	CCW Circle
5	Wait (Y)	Circle > 180
6	Backlash (X)	Posit. Frozen (after ^V)
7	Backlash (Y)	Sub. Buffer Open

Bit	febb	febc
0	Servo Pointer	Servo Pointer
1	Servo Pointer	Servo Pointer
2	Servo Pointer	Servo Pointer
3	Dwell Request	Servo Pointer
4	Home in Progress (until capture)	Servo Pointer
5	Dwell in Progress	Stop Request
6	Continuous Move Req	Servo Pointer
7	Memory Request	Servo Pointer

For comparison purposes the Standard SMCC's status bytes are shown below. The Standard SMCC does not support DMA capability so ASCII communication must be used to receive status data.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Bit	feb9	feba
0	Home Y complete	Time Mode
1	Home X complete	Single Step
2	Block Ready	Buffer Full
3	Block Request	Circle > 180
4	Circular Motion	CCW Circle
5	Feed Hold	Circle Mode
6	Running	Feed Hold
7	Auto Divide	Not Ready For Move

Bit	febb	febc
0	Jog Positive (X)	Jog Positive (Y)
1	Jog Negative (X)	Jog Negative (Y)
2	Open DAC Mode (X)	Open DAC Mode (Y)
3	I2T Limit (X)	I2T Limit (Y)
4	Following Error (X)	Following Error (Y)
5	Not Stopped (X)	Not Stopped (Y)
6	Not In Position (X)	Not In Position (Y)
7	Zero Velocity (X)	Zero Velocity (Y)

10) EAROM SAVE

The SMCC-P, unlike the Standard SMCC, does not save the assigned values of C functions, M functions, and variable (P) parameters in the 512-byte EAROM which is standard on all SMCCs. In order to save these variables and functions on the SMCC-P, Option 2 (8K permanent memory) is required. This does mean, however, that with only the 512-byte EAROM, approximately 80 lines of motion programs and subroutines can be permanently stored. In contrast, the Standard SMCC can only store seven lines of program in the small EAROM. PLC programs cannot be stored in the 512-byte EAROM.

11) EFFECTS OF "t" or i08

A. In the Standard SMCC the i08 (I08) and the "t" Commands have identical functions and their effects are as follows:

1. "t" or i08 is the accel/decel time for all calculated moves, as well as jog. A "t" Command changes the value of i08, and an i08 Command overrides the last "t" Command.

2. A t/i08 time is added between consecutive Dwell Commands (it is <u>not</u> added to a single Dwell Command).

3. All moves take a minimum of t/i08 time to get executed.

4. The "q" (Abort) Command uses the t/i08 time to decelerate to a stop if i11 (Feed Hold Slew Control) is set to a 1.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

5. If i11 = 1, the i08 time controls the deceleration on hitting a hardware or software limit or on receipt of an "H" (Feed Hold) Command, during a jog move. (The deceleration for an "H" Command received during a calculated move is controlled by the SLEW value.) If i11 = 0, all of these decelerations occur immediately.

6. The t/i08 time is used to segment circular moves.

B. The SMCC-P uses the i08 and "t" Commands differently as follows:

1. The i08 parameter is used to initialize the "t" Command on power turn-on or reset. Thereafter, i08 only affects the jog accel/decel time.

2. Once a "t" Command is executed, "t" is used to control all calculated move accelerations and decelerations.

3. There is no "t" time inserted between consecutive dwell commands.

4. All "zero" and "small" moves still require a "t" time for execution. "t" sets the minimum time in which a programmed move will take place.

5. The deceleration after "q" (Abort) and "H" (Feed Hold) Commands, and after hardware or software limits is handled the same as in the Standard SMCC (using i08).

6. The "t" Command is used to segment circular moves into parabolic move segments. This is essentially transparent to the user, and works well over a wide range of "t" values. However, the user is advised to keep "t" small, since there is a slight backup at the beginning of a circular move if t is a significant fraction of move time T. The size of this backup is R * (1 - cos0)/108, where 0 = 180 degrees * t/T. For example, if R = 10000 counts, t = 100 and T = 1000, then 0 = 18 degrees and the backup is 4.5 counts. If t were reduced to 25, the backup would be 0.3 counts (which means non-existent).

7. The "t" Command is used to define the move time in parabolic motion mode and in the CIRC4 "3-point smoother" mode.

8. Parabolic introduces i12 as "Move Delay/Calculation Time" which it uses to achieve better multi-axis synchronization and a predictable calculation time. If the "t" time is less than the i12 time, calculations between blended move segments must be done in "t" time. If they cannot be finished in this time, the card will lock up.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

12) "C" COMMAND

The "C" Command is used in the Standard SMCC to provide a means of generating off-axis (tilted) ellipses. SMCC-P does not provide this facility and therefore these types of ellipses must be generated by using either parabolic commands or be derived by a series of 3-point smoothed points using the CIRC4 Command.

13) PARALLEL BINARY POSITION

SMCC-P is capable of accepting 8 to 23 bits of parallel binary data for position information of the X and/or Y axis. This feature is primarily used with Absolute Binary Optical encoders and Temposonic Position Transducers. SMCC's Accessory 14 Board or MCC's Option 7 Board may be used to receive the parallel data from the encoder. The SMCC-P reads 16 bits of data at a time, left shifts it five places for a net reading of 12 sign extended binary bits to be used for positioning data. Only one binary position input per axis can be handled. Parameters i64[i65], i70[i71], i72[i73], i76[i77] are used to control parallel binary positioning.

14) GEARED RESOLVERS

Many systems use multiple resolvers (two resolvers with gear ratio) to determine absolute position on large machines. The resolvers are commonly referred to as coarse and fine and require individual R/D converters. The SMCC-P is capable of combining the R/D readings from each of the resolvers in order to derive an absolute position as large as 23 bits total. The gear ratio between resolvers can be from 1 to 255. Parameters i64[i65], i66[i67], i68[i69], i70[i71], i72[i73], and i74[i75] are used to control the functions of the single- or multiple- resolver system. SMCC's Accessory 14 may be used to convert two resolvers simultaneously. If the X and Y axes must both have geared resolvers, two ACC-14s may be used to satisfy the requirement. See the i parameter definitions and the "Use of Absolute Encoders and Resolvers" section of the manual for more details.

15) RANGE MODE CONTROL

The Standard SMCC has position range control parameters i37[i57] which place it in a position rollover mode where a position range can be specified and the SMCC will automatically handle all motions within that range and go from one position to another using the shortest distance. This mode is useful in rotary tables where a range of typically 360000 is specified. SMCC-P has the same capability but adds a new mode of operation within the range. i78[i79] controls whether the shortest distance inside the range will be taken when a position is specified

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

(i78[i79] = 0) or whether an absolute position move will be made with controllable motion direction (i78[i79] = 1) depending upon the sign of the specified absolute move.

16) "GO UNTIL"

The SMCC-P provides a new set of commands for X- and Y- axis moves which allows an external trigger on the X user flag input to switch the SMCC-P's move from a primary move to a secondary incremental move. If the trigger is not received during the primary move, the SMCC will generate an "Error 5" and set the "Error" line. This can be cleared by sending a <CR>. The command format is:

X primary position^Secondary delta position
Y position^delta position
X position^delta position Y position^delta position

NOTE: Both X and Y are triggered by the X user flag input, and X and Y may be absolute or incremental. X/U/AX/WX or Y/V/AY/WY may be used.

If SMCC-P is in feedrate mode ("F" Command), the entire move including the "Delta position" will be executed at the same speed.

If SMCC-P is in time mode ("T" Command), then the entire move including the "Delta position" will occur in the specified T time. Depending upon when the trigger is received, this may cause the "Delta position" move to be made slower or faster. The trigger may be received and acted upon even during the deceleration position of the primary move until "Zero Velocity" status is reached. If the trigger arrives just before the move ends, the SMCC-P will use the i12 parameter "Calculation Time" to extend the move.

Use the i26 parameter, "Set Home Flag", to choose a variety of combinations of the home flag, the encoder "C" channel, and the "X User Flag" to trigger the SMCC-P during the "Go Until" Command.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

USE OF ABSOLUTE PARALLEL BINARY ENCODER OR RESOLVERS WITH SMCC

TABLE OF CONTENTS

PAGE

OVERVIEW 187

USING ABSOLUTE PARALLEL BINARY POSITION ENCODER. . 187

Single Absolute Binary Connection to SMCC for X and Y Axis Diagram. 189

USING RESOLVERS. 190

Single Resolver Connection to SMCC for X and Y Axis Diagram. 191

Dual Resolver (Geared) Connection to SMCC Diagram 193

ACC-14 AND ACC-15. 195

POSITION FEEDBACK DEVICES. 197

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

USE OF ABSOLUTE PARALLEL BINARY ENCODER OR RESOLVERS

WITH SMCC

****

OVERVIEW

The primary position feedback for SMCC is the incremental A/B quadrature encoder. The maximum pulse rate for the position encoder's A and B inputs is 500 KHz. Even though the SMCC can decode the input by one-, two-, or four-times decoding the maximum count rate the SMCC can measure is only one MHz (not two MHz as might be expected) due to counter limitations in the CPU itself.

The SMCC has a total of four encoder channels, two of which are used for position feedback of the X and Y axes and two for "handwheel" encoder inputs. The handwheel encoder channels do not have the same count rate limitation and therefore can count at a rate of two MHz (500 KHz x 4).

All encoder channels are capable of X1, X2, and X4 multiplication with direction sense reversible, and a special mode where instead of A/B quadrature pulses, pulses are accepted in Channel A and direction control in Channel B. All of these operating modes are controlled by i parameters.

In addition to A/B quadrature encoders, the SMCC is able to use absolute parallel binary position input of 8 to 23 bits from an optical encoder or other source, or single or dual resolvers, for position feedback. To accomplish this the SMCC requires ACC-14 for Parallel Binary or ACC-14 with ACC-15 installed for resolvers. Each of these is described below.

1) USING ABSOLUTE PARALLEL BINARY POSITION ENCODER

An Accessory-14 board may be used with SMCC in order to enable it to read and use Parallel Binary data as position feedback input. The number of bits that can be handled is 8 to 23 bits and these must represent the total range of actual position travel of the axis being controlled. The absolute encoder, therefore, rotates one revolution or less for the entire travel of the machine. The source of the data can be any device which generates parallel binary bits indicating position; most commonly this is a single disk absolute binary encoder, or an encoder where two or more encoder disks are combined to provide the total number of bits needed in the application. The SMCC will not accept Gray-scale binary data; data of this form must be converted to Standard binary external to the SMCC and ACC-14. The data can be also be derived from an LVDT or a potentiometer,

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

provided that an external A/D converter is used so that the input voltage from the LVDT or the potentiometer is converted to a parallel binary word.

The assumption is made that the word "absolute" here refers to the entire travel of the axis so that the binary position data is incrementing from zero to a maximum number. Please note that the zero of the absolute encoder does not have to be mechanically adjusted to be at the actual zero position of the axis since SMCC provides a position bias offset number which can be used to create a zero anywhere along the axis. See i70[i71] for encoder bias description.

The number of parallel binary bits to be used in an application is a function of the required resolution for the axis being controlled. For example, if a 16 bit absolute encoder is used and the total axis travel is 65 inches, then the encoder will provide a resolution of just under .001 inch per bit.

In order to follow the subsequent description, please refer to the "Single Absolute Binary Encoder Connection to SMCC" block diagram. Assume we are to connect two absolute binary encoders to the SMCC for the X and Y axes.

Please note that any combination of encoders can be used for the X and Y axes, such as incremental (A/B quadrature) on X and absolute on Y, or absolute on X and resolver on Y, and so on.

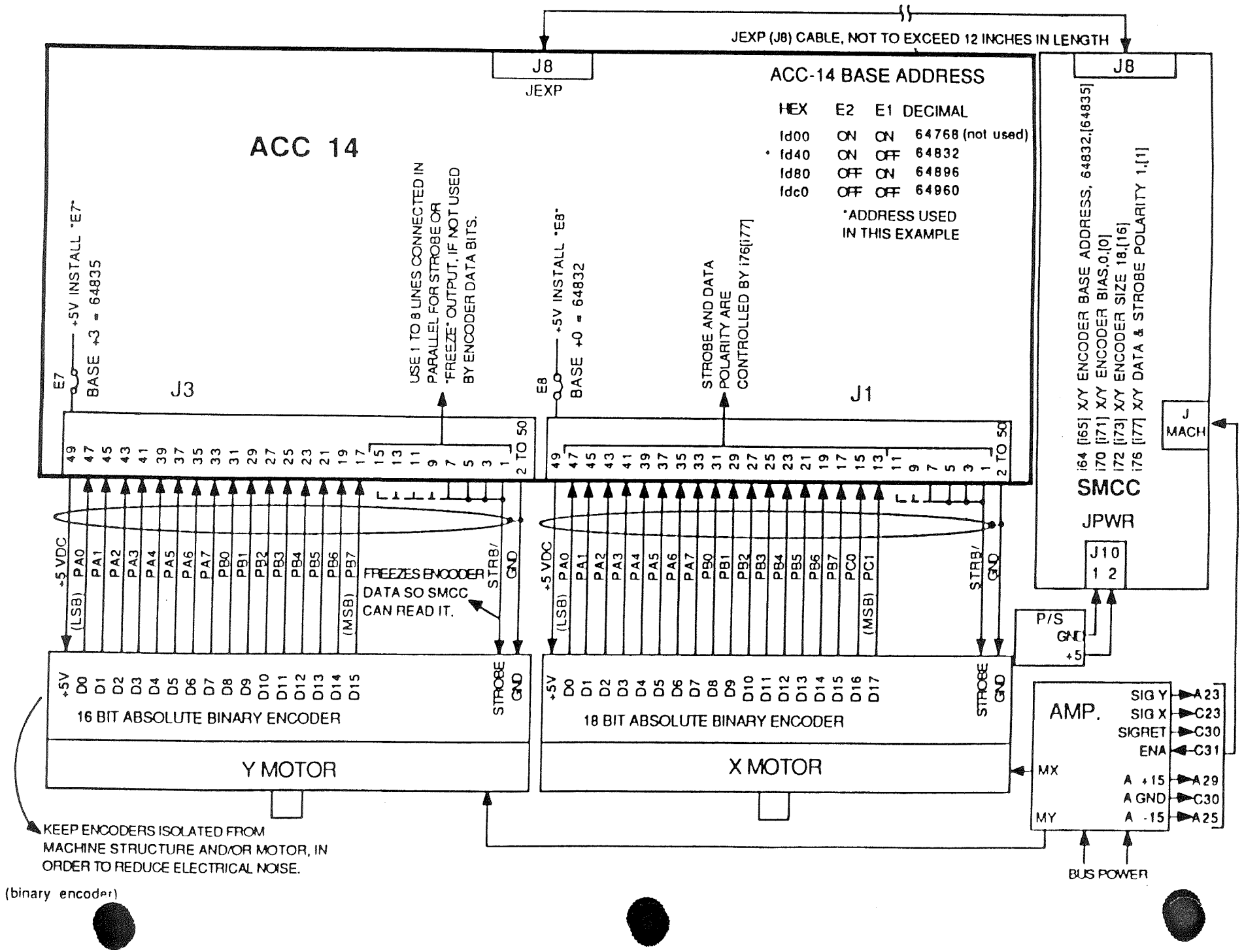
The absolute encoders must have single-ended TTL (0 to +5 volt) output data, eight bits minimum and 23 bits maximum. The data output polarity may either be true logic (1 = +5 volts) or false logic (1 = 0 volts), the SMCC i76[i77] parameters allow the user to invert the data or not before the SMCC uses it. The output data from the encoder must also be controllable by an external signal (strobe) which can be used to "freeze" the encoder data while the SMCC is reading it. i76[i77] also allows the selection of strobe output polarity. The strobe output polarity from SMCC can be specified and as many as eight lines may be paralleled at the ACC-14 in order to stiffen the strobe line's sink capability so as to provide a clean "freeze" command to the encoder. The strobe output from the ACC-14 is one line if a 23-bit encoder is used, two lines if 22 bits are used, and is a maximum of eight lines if a 16-bit or less encoder is used.

Proceed to wire the parallel data lines for the X encoder to the ACC-14 J1 connector using shielded cables and the Y encoder to J3. Use the block diagram and its instructions.

It must be realized that the wiring of the absolute encoder is very critical since this type of encoder is much more subject to noise pickup than an A/B quadrature encoder. It

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SINGLE ABSOLUTE BINARY ENCODER CONNECTION TO SMCC FOR X &Y AXIS



Artisan Scientific - Quality Instrumentation Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

is important to isolate the case of the absolute encoder from the motor case and shaft as well as the machine structure in order to prevent electrical noise pickup; it may be necessary to use non-conducting materials to couple the absolute encoder to the machine's structure. It is also imperative to use well-shielded wiring between the encoder and ACC-14.

The i parameters of the SMCC must be set up to configure the system correctly. Please refer to the SMCC's i parameter section in order to set up the i64 [i65], i70[i71], i72[i73], and i76[i77] parameters properly.

In operation the SMCC will read the actual absolute position of the parallel encoder once upon turn-on or reset and apply the specified position bias to it if one has been entered. It will then set its current desired position to that obtained from the initial reading. This is what sets up the "absolute" position so that SMCC knows where it is upon power turn-on. Thereafter the SMCC will read only the sixteen least significant bits of the encoder and drive its position incrementally from these bits. The SMCC will also read all specified encoder bits and set the absolute position when it is issued the online "$" Command (Soft Reset).

In all other respects, once configured correctly, the SMCC with absolute parallel binary position feedback can be programmed and operated as if an A/B quadrature encoder were being used.

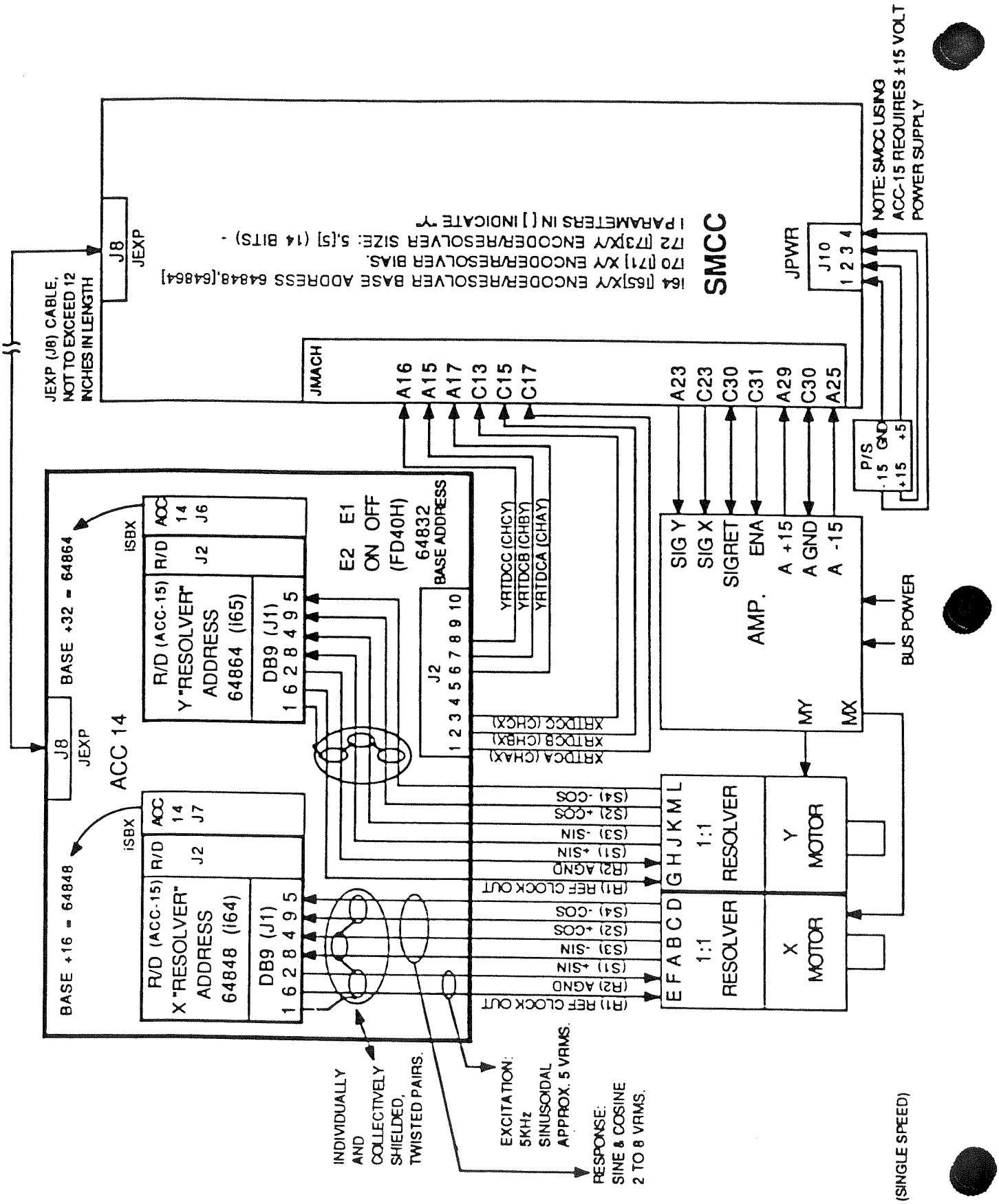
2) USING RESOLVERS

A resolver is a device which utilizes magnetic coupling principles for its operation. It requires an excitation signal in response to which it outputs sine and cosine signals (90 degrees apart) whose amplitudes are used by the R/D converter to determine the rotating shaft's angular position. Other types of resolvers exist which output phase-shifted sine/cosine where the phase shift is a measurement of angular position; these are not as common, and the SMCC's ACC-15 is designed to operate with the first type described above, where sine/cosine amplitudes are used.

The resolver's best attribute is that it is a passive device (no electronic components are used inside the resolver), so that it is very rugged and can withstand the abusive conditions on a machine, such as high temperature, shock, and vibration. Also, because its outputs are measured in a ratiometric manner, it can also withstand the electrically noisy machine environment better than an absolute encoder. Its cost is generally higher than an encoder system, especially when the conversion electronics, located at the controller, is taken into account.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SINGLE RESOLVER CONNECTION TO SMCC FOR X AND Y AXIS



The resolver is in effect a one-revolution absolute encoder since its current shaft position is immediately known upon power turn-on. Another characteristic of the resolver is that it is available in multi-pole versions such that each revolution of the shaft produces more than one cycle of sine/cosine; these are known as multi-speed resolvers. The following is a summary of commonly available resolvers:

Single-Speed, Two-Pole - Produces one sine wave cycle per revolution

Double-Speed, Four-Pole - Produces two sine wave cycles per revolution

Triple-Speed, Six-Pole - Produces three sine wave cycles per revolution

When an R/D converter is used to determine the shaft angle position of a resolver, each "cycle" or complete sine wave represents a zero-to-maximum count range. SMCC's ACC-15 can have resolutions of 10 to 16 bits selectable in increments of two bits. Therefore the resolutions can be:

10 bits - 1024 counts/cycle
12 bits - 4096 counts/cycle
14 bits - 16384 counts/cycle
16 bits - 65536 counts/cycle

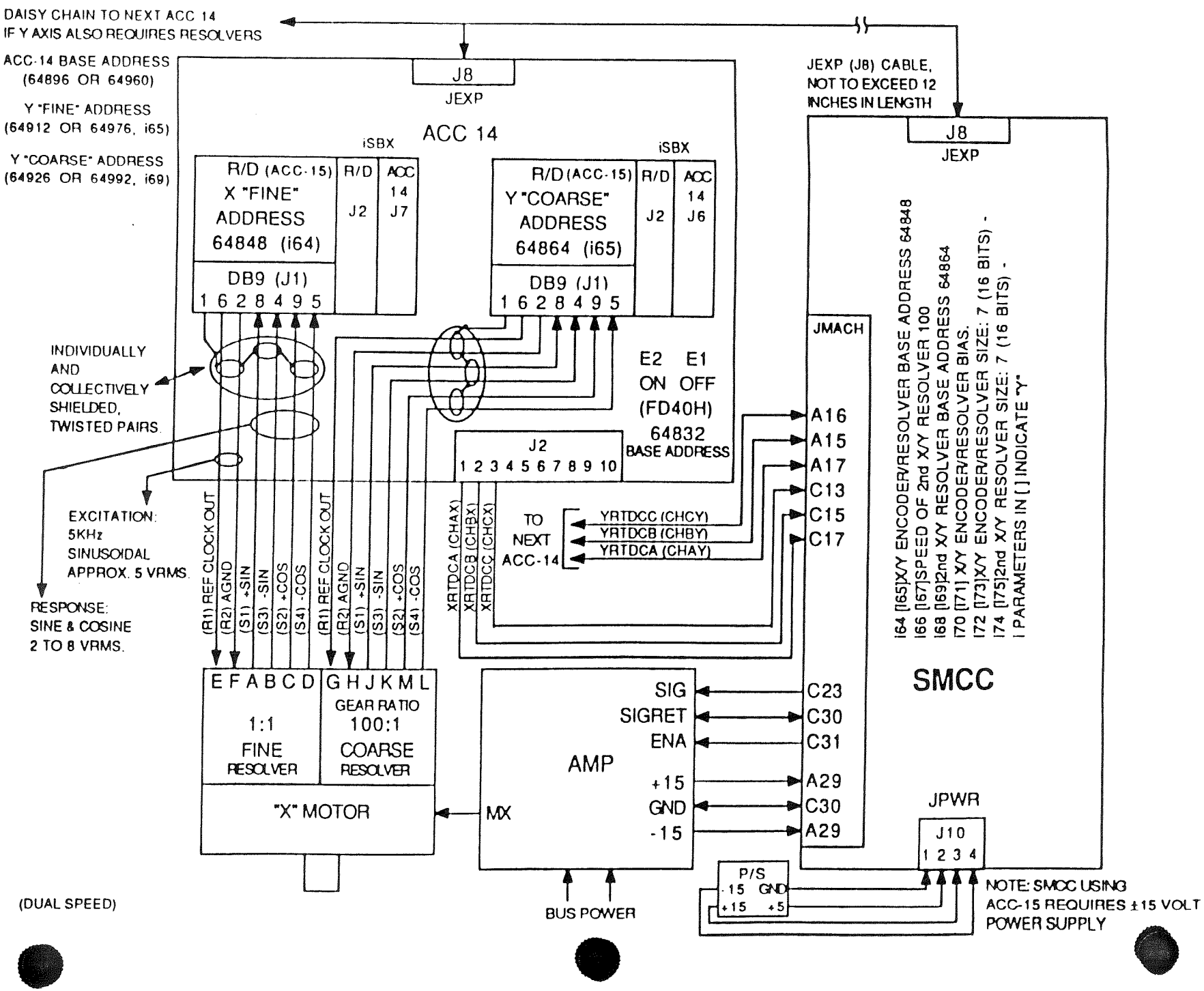
If a two-pole resolver is used, one cycle is the same as one revolution, so the count resolution is now based on one revolution of the resolver. A restriction applies to the selection of the encoder or resolver to be used if a Standard SMCC will be doing the actual phase commutation of the motor to be driven. This is that a whole, non-fractional number of encoder or resolver counts is required per electrical revolution of the motor being commutated. When driving a six-pole motor, unless a six-pole resolver is also used, the division by three makes it impossible to obtain a whole number. The Parabolic SMCC can handle this situation if you set i50 to 1 and i40 to the number of counts in <u>three</u> electrical cycles (instead of one).

In all other cases, a single- or double-speed resolver may be used. A single speed is preferred since one revolution represents one full count cycle and the resolver's zero position can be used as a once-per- revolution marker, the same as the "C" Channel in a quadrature encoder.

Another possibility exists with resolvers in that if a single, directly coupled resolver is used on the motor, it only provides absolute position for one revolution of the motor. This is sufficient when the amplifier is using the resolver to derive sinusoidal phase commutation for the motor, but it does not provide an absolute position for the machine since usually many revolutions of the motor are

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DUAL RESOLVER (GEARED) CONNECTION TO SMCC



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

required to travel the full length of the machine. In order to obtain absolute position information, a common approach is to use two resolvers with a gear reduction between the "fine" and "coarse" resolver. The "fine" resolver is the one revolving at the same rate as the motor; it is directly coupled to the motor and the "coarse" resolver is gear-reduced 2 to 256 times so that it revolves not more than once over the entire travel of the machine. The SMCC is capable of processing the data obtained from the the two resolvers and deriving an absolute position representing the machine's actual position.

Please refer to the "Single Resolver Connection to SMCC" and "Dual Resolver Connection to SMCC" block diagrams for a complete description of all necessary connections to SMCC, the ACC-14 and ACC-15, as well as the i parameters involved in the correct setup of the resolver application. Please refer to the SMCC manual for a detailed description of i parameters.

When using an absolute parallel binary encoder, only an ACC-14 is needed to read the incoming data; to use resolvers, one or more ACC-15s are needed. The ACC-15s plug into the ACC-14s' J6 and J7, iSBX connectors. They provide the necessary excitation (5 KHz sinusoidal at five volts RMS amplitude) and the decoding for the signals from the resolver (which should be a sine and cosine, at two to eight volts RMS amplitude). The decoding resolution of the ACC-15 is software controlled by the SMCC's i parameters and provides 10, 12, 14, or 16 bits of position resolution per cycle (or revolution, if a two pole resolver is used). Please note that as the resolution is increased, the maximum RPM of the resolver is reduced; at ten bits the maximum RPM is 62,000, at twelve bits it is 15,600, at fourteen bits 3,900, and at sixteen bits it is 975 RPM. All of these RPMs coincide with SMCC's maximum input data rate, which is one MHz for position pulses being received.

The ACC-15 provides the decoded position data as a binary 16-bit word so SMCC can read it and establish its absolute position. However, as is done in the case of the absolute encoder, SMCC reads the absolute data from the resolver once upon power turn-on or reset and then relies on an A/B quadrature pulse generator which is on the ACC-15 and which provides incremental pulses derived from the absolute data read by the R/D converter. The SMCC then uses these A/B quadrature pulses as if an incremental encoder were connected to it. This includes the "C" Channel or once-per-revolution pulse found on an encoder. These connections are made from the J2 connector of the ACC-15 to the J1 (J MACH) connector of the SMCC. The actual absolute position may be reloaded by issuing the online "$" Command (Soft Reset) to the SMCC. Here also SMCC provides a position bias adder, i70[i71], which can be used to set the absolute zero position read by the resolvers to coincide with the machine

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

zero without having to physically rotate the resolver to its zero position.

Once the resolver setup has been completed, all operations of SMCC are transparent to the user and the SMCC can be programmed and commanded as if a single A/B quadrature encoder were being used.

3) ACC-14 and ACC-15

Please refer to the individual manual and drawing for these devices. The following is an overview of their usage for parallel binary input data and resolvers.

Each SMCC can support up to four ACC-14s connected in daisy-chain fashion to the J8 (JEXP) connector. The ACC-14 has "E Point" jumper selections, E1 and E2, for allowing it to be placed at one of four address spaces supported by SMCC. The four base addresses are: 64768, 64832, 64896, 64960. Of these, due to hardware limitations, only the last three can be used, so do not use 64768 because it may cause erroneous data to be transmitted to SMCC. The connecting cables should be no longer than 12 inches to ensure proper communications. The SMCC will provide power to the ACC-14s over the cable. If any ACC-15s are used, the SMCC must be provided with +/-15V so these can be passed over the cables as well.

When using ACC-14 to read parallel binary data, the J1 and/or J3 connectors may be used as shown in the block diagram titled "Single Absolute Binary Encoder...". The address for J1 is Base address + 0, and for J3 it is Base address + 3. So, if the ACC-14 were located at address 64896, then the J1 connector's address would be 64896 and that of J3 would be 64899 and these would be the addresses to be used in the i parameters requiring base address specifications. The user may connect the X or Y axis to J1 or J3 as he wishes. As long as the i parameters are specified correctly, the SMCC will receive the correct data.

When using ACC-14 with ACC-15 to read resolvers, the ACC-15s should be plugged into the J6 and J7 connectors of the ACC-14 (iSBX connectors). See block diagrams titled "Dual Resolver ..." and "Single Resolver..." for actual connections. Note that the J6 connector is addressed as Base + 32 and J7 as Base + 16. Therefore, if the ACC-14 were located at 64960, the J6 connector would be at 64992 and the J7 connector at 64976. If both axes, X and Y, of SMCC are to be used with two geared resolvers each, then two ACC-14s must be used each with two ACC-15s. The i parameters are used to tell the SMCC what the configuration is and where to read data for each of the resolvers.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The following chart is a summary of all position-feedback devices that may be used with SMCC. It is provided as an overall guide for selecting a position feedback device for various applications.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

POSITION FEEDBACK DEVICES

DEVICE	TYPE	MAX. DEVICE PULSE RATES	SMCC USAGE/MERIT
A/B Quad. Rotary or Linear Encoder	Incremental	100 to/250 KHz	Direct Connect, Differential or Single Ended Lowest Cost, Most Available Most Widely Used
Absolute/ Encoder Optical	Absolute, 8 bits min. 23 bits max	100 KHz/to 1 MHz	Use ACC-14 Parallel Data Input Provides Absolute Position Information
Resolver	Absolute/ Incremental Single or Multi-Speed	1 MHz	Use ACC-14 and ACC-15 Very Rugged and Reliable
Laser Inter-ferometer	Incremental	1 to/64 MHz	Direct Connect. If A/B Quad. Output Provided; or Use ACC-14 If Parallel Data Available. SMCC limits operation to 1MHz for A/B Quad., and 2 MHz for Parallel Input.
Farrand Inductosyn	Absolute/ Incremental Single or Multi-Speed	1 MHz	Highest Resolution, Fastest Speed, Highest Accuracy. Use ACC-14 and ACC-15. May Require Pre-amplifier For Sine/Cosine Feedback Signals. Very Accurate, High Resol.
Sonic Transducer	Incremental	N/A	If Transducer has A/B Quad. Converter, Direct Connect. If Transducer has Parallel Binary Data Output, use Acc-14. SMCC does not support pulse-to-distance conversion. Very Rugged, Used Mostly in Hydraulic Cylinders, Slow Update Rate
Potentiometer	Absolute	N/A	Use Option 7 with 12-bit A/D Converter to Read Position Low Cost, Subject to Wear and Drift
LVDT	Absolute	N/A	Use Option 7 with 12-bit A/D Converter to Read Position Accurate, Reliable, Rugged

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

MEMORY AND I/O MAPS

TABLE OF CONTENTS

PAGE

SMCC I/O MAP. 199

SMCC MEMORY BIT MAP (STANDARD). 200

SMCC MEMORY BIT MAP (PARABOLIC) 201

STATUS DISPLAY INDICATOR. 202

MEMORY MAP SUMMARY. 203

STANDARD VERSION INTERNAL RAM 205

STANDARD VERSION EXTERNAL RAM 211

PARABOLIC VERSION INTERNAL RAM. 213

PARABOLIC VERSION EXTERNAL RAM. 219

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC I/O MAP

****

CPU BITS

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
ff01	IPOS/	FERR	MO1	I2T/	DREQ	XX	STEP	STRT
							ABORT	

CPU MUX BITS

SERVO 1 = 1

	P0.7	P0.6	P0.5	P0.4
ff3a	XJMI	XJPL	XPLL	XMIL
	"HOME"		HOLD	

SERVO 2 = 1

	P0.7	P0.6	P0.5	P0.4
ff3a	YJMI	YJPL	YPLL	YMIL
	"="		HOLD	

GATE XJMI and XJPL = Start Home Cycle (HOME,h)
YJMI and YJPL = Return to Prejog Position (=)
Any pair of limits together = Feed Hold
STEP and STRT = Abort at End of Move

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
ffb4	MO3	MO2	XX	XX	XX	XX	XX	XX
ffb5	MO5	MO4	XX	XX	XX	XX	XX	XX
ffb7	YUFL/	YHFL/	XUFL/	XHFL/	MI4	MI3	MI2	MI1

PARALLEL PORT BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
ffb8	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
ffb9	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

GLT BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
ffbc	XX	XX	RSL	WSL	MWAIT	MI7	MI6	MI5

BIT MAP FOR MAPPING (MOn = Mnn) AND (MIn = Mnn)
When in set-up mode, the SMCC automatically sets up the following:

```
 C01 = ff01     DEFINE
 C02 = ffb4     ADDRESSES
 C03 = ffb5
 C04 = ffb7
 C05 = ffbc

 MO1 is M01 = C01.5      DEFINE
 MO2 is M02 = C02.6      OUTPUTS
 MO3 is M03 = CO2.7      AT
 MO4 is M04 = C03.6      ADDRESS
 MO5 is M05 = C03.7      AND BIT.

 MI1 is M11 = C04.0      DEFINE
 MI2 is M12 = C04.1      INPUTS
 MI3 is M13 = C04.2      AT
 MI4 is M14 = C04.3      ADDRESS
 MI5 is M15 = C05.0      AND BIT.
 MI6 is M16 = C05.1
 MI7 is M17 = C05.2
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC MEMORY BIT MAP (STANDARD)

EXTENDED STATUS BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
fe20	BACK2	BACK1	WAIT2	WAIT1	INSERT	EDIT	EXREAD	ADREQ

STATUS BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
feb9	ADIV	RUN	FHOLD	CIRIP	BLKREQ	BLKRDY	HOME1	HOME2
feba	NOTRDY	LIMIT	CIRCLE	CIRDIR	CIR180	BFULL	STEP	TMODE
febb	RVZ1	NIPOS1	NTRDY1	FE1	I2T1/	ODAC1	JOGM1	JOGP1
febc	RVZ2	NIPOS2	NTRDY2	FE2	I2T2/	ODAC2	JOGM2	JOGP2

SERVO CONTROL BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
febd	MEMREQ	DWEREQ	STPREQ	DWELL	XX	XX	XX	XX

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC MEMORY BIT MAP (PARABOLIC)

****

STATUS BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
feb5	TIMER	RUN	FHOLD	ADIV	BLKREQ	CMOVE	HOME1	HOME2
feb6	NOTRDY	NTRDY2	NTRDY1	LIMIT	FRATE	BFULL	STEP	TMODE
feb7	RVZ1	NIPOS1	I2T1/	FE1	LMTM1	LMTP1	JOGM1	JOGP1
feb8	RVZ2	NIPOS2	I2T2/	FE2	LMTM2	LMTP2	JOGM2	JOGP2

EXTENDED STATUS BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
feb9	BACK2	BACK1	WAIT2	WAIT1	CIRCLE	EXREAD	INSERT	EDIT
feba	SBOPEN	FREEZE	CIR180	CIRDIR	HMDONE	EXMOV1	EXMOVE	ADREQ

SERVO CONTROL BITS

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
febb	MEMREQ	CMVREQ	DWELL	HOME	XX	XX	XX	XX
febc	SERVP7	SERVP6	STEP	SERVP5	SERVP4	SERVP3	SERVP2	SERVP1

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

STATUS DISPLAY INDICATOR

****

The SMCC is equipped with a one digit hexadecimal display which is used to indicate operational status to the operator.

.(Left Decimal) = In Position (i01)
.(Right Decimal) = Following Error Indicator (i10)

0 = Okay (SMCC is Happy)
1 = Hall Effect Failure (Brushless Motor)
2 = Following Error Failure (i00)
3 = TBD, Not Used Currently
4 = Circle Too Large for Radius
5 = "Go Until" Did Not Receive Trigger
6 = "Go Until" Did Not Receive Trigger
7 = Communications Error (Invalid Character)
8 = AMPlifier Failure, (AMPlifier Without Fault Data Bits)
*9 = SHUNT-FAULT
*A = GROUND-FAULT
*B = +15 OR -15 = LOW VOLTAGE
*C = Floating +15V (Bias Supply) = LOW VOLTAGE
*D = OVER-TEMP
*E = BUS-OVER-VOLTAGE
*F = DRIVER-FAULT, (OVER-CURRENT, BUS-UNDER VOLTAGE)

*NOTE: 1) Delta Tau's amplifiers provide three Fault Indication Bits which are used to isolate the cause of the problem and display it. Please note that physically the indicator may be mounted to accommodate both the stand-alone and the rack-mount configuration for visibility. An extension cable and mounting bracket are needed for the rack-mount version.

2) If both decimal points are on simultaneously, they indicate that the SMCC is not functioning properly and is unable to complete its internal initialization. Refer to "Start-Up Section" for causes and reasons.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC MEMORY MAP

****

The following pages provide complete information of the Standard and Parabolic SMCC's memory and register map. They are provided for the purpose of allowing users to have direct access to the many variables available in the SMCC so that more sophisticated programs, especially in PLC, may be written to accomplish more useful and difficult tasks. A brief summary is provided here to point our some of the more commonly used registers:

FUNCTION	PARABOLIC SMCC ADDR -# BYTES	STANDARD SMCC ADDR - # BYTES
Buffer Pointer, Next command to be executed	a3b3 - 2	a2f8 - 2
Buffer Pointer, Where data will be stored next	a3b5 - 2	a2fa - 2
Pointer to End of Main Program	a27c - 2	a2fe - 2
Library Pointer, Where next Subroutine will be stored	a27e - 2	a2fc - 2
Free Memory Space for user data. Not used by SMCC.	a3a7 - 12	a200 - 36
X Handwheel register	fe3e - 2	fe38 - 2
Y Handwheel register	feeb - 2	fe4a - 2
Absolute Position of X	fe48 - 4	fef6 - 4
Absolute Position of Y	fe4c - 4	fe46 - 4
Analog inputs filtered, 2 40 mscs. update rate	fe8c - 2x2	fe82 - 2x2
Analog inputs unfiltered, 2 40 mscs. update rate	fe90 - 2x1	fe86 - 2x1
Variable parameters P1 to P15 (P1 first, 4 bytes per var. LSByte first)	a240 - 4x15	a1a9 - 4x15
X Acceleration register, if not zero, shows SMCC accelerating. Use first four bytes.	fe58 - 6	none

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

FUNCTION	PARABOLIC SMCC ADDR - # BYTES	STANDARD SMCC ADDR - # BYTES
Y Acceleration register, if not zero, shows SMCC accelerating. Use first four bytes.	fe6e - 6	none
X axis PWM Analog output		
magnitude:	ff14 - 1	
direction:	ff00.0	
Y axis PWM Analog output		
magnitude:	ff16 - 1	ff16 - 1
direction:	ff00.1	ff00.1
Version Number Address	2034 - 5	202e - 5

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
**************************************************************************
                    INTERNAL RAM (10/06/88)
                         STANDARD
**************************************************************************

                        ORG   0FE00H

FE00                RBUFF:  DS    32
        FE00        TBUFF   EQU   RBUFF

FE20                XBITS:  DS    2          ;EXTENDED STATUS BITS
        0100        ADREQ   EQU   XBITS.0    ;AUTO-DIVIDE REQUEST
        0101        EXREAD  EQU   XBITS.1    ;EXTENDED READ FLAG
        0102        EDIT    EQU   XBITS.2    ;PROGRAM BUFFER EDIT FLAG
        0103        INSERT  EQU   XBITS.3    ;PROGRAM BUFFER INSERT FLAG
        0104        WAIT1   EQU   XBITS.4    ;X-AXIS WAIT MOVE FLAG
        0105        WAIT2   EQU   XBITS.5    ;Y-AXIS WAIT MOVE FLAG
        0106        BACK1   EQU   XBITS.6    ;X-AXIS BACKLASH FLAG
        0107        BACK2   EQU   XBITS.7    ;Y-AXIS BACKLASH FLAG
        0108        PLCFLG  EQU   XBITS+1.0  ;PLC EDIT FLAG
        0109        RESULT  EQU   XBITS+1.1  ;PLC RESULT FLAG
        010A        ELSE    EQU   XBITS+1.2  ;PLC ELSE FLAG
        010A        PLCCMD  EQU   XBITS+1.2  ;PLC COMMAND FLAG
        010B        UFLAG   EQU   XBITS+1.3  ;USER FLAG MOVE
        010C        EXMOVE  EQU   XBITS+1.4  ;EXTENDED MOVE FLAG

FE22                TPOS2:  DS    4          ;DELAYED POSITION Y-AXIS
FE26                IPOS2:  DS    4          ;ERROR INTEGRAL Y-AXIS
        FE22        WORDS   EQU   TPOS2
        FE22        DAC     EQU   TPOS2      ;DAC OUTPUT FOR PHASED MOTORS
        FE23        SLIP    EQU   TPOS2+1    ;SLIP FREQUENCY
        FE24        PHASE   EQU   TPOS2+2    ;PHASE POSITION
        FE28        PHASAD  EQU   TPOS2+6    ;PHASE ADVANCE

FE2A                TPOS1:  DS    4          ;DELAYED POSITION X-AXIS
FE2E                IPOS1:  DS    4          ;ERROR INTEGRAL X-AXIS

FE32                DPOSL1: DS    6          ;DESIRED POS.RESIDUAL X-AXIS
FE38                PHW1:   DS    2          ;PREVIOUS X-AXIS HNDW. POS.
FE3A                ACRES1: DS    2          ;ACCEL RESIDUAL X-AXIS
FE3C                DAC1R:  DS    2          ;DAC RESIDUAL X-AXIS

FE3E                EPOS2:  DS    4          ;Y-AXIS POSITION ERROR
FE42                EPOS1:  DS    4          ;X-AXIS POSITION ERROR
FE46                APOS2:  DS    4          ;Y-AXIS POSITION
FE4A                APOS1:  DS    4          ;X-AXIS POSITION
FE4E                DPOS2:  DS    4          ;Y-AXIS DESIRED POSITION
FE52                DPOS1:  DS    4          ;X-AXIS DESIRED POSITION
FE56                RUNV2:  DS    6          ;Y-AXIS COMMAND VELOCITY
FE5C                RUNV1:  DS    6          ;X-AXIS COMMAND VELOCITY

FE62                DRNV2:  DS    6          ;Y-AXIS DESIRED VELOCITY
FE68                DRNV1:  DS    6          ;X-AXIS DESIRED VELOCITY
FE6E                ACCEL2: DS    6          ;Y-AXIS COMMAND ACCEL
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
FE74            ACCEL1: DS   6          ;X-AXIS COMMAND ACCEL
FE7A            ADPOS2: DS   4          ;Y-AXIS DESTINATION POSITION
FE7E            ADPOS1: DS   4          ;X-AXIS DESTINATION POSITION

FE82            FATD0:  DS   2          ;FILTERED ATD #0 INPUT
FE84            FATD1:  DS   2          ;FILTERED ATD #1 INPUT
FE86            ATD2:   DS   1          ;RAW ATD #2 INPUT
FE87            ATD3:   DS   1          ;RAW ATD #3 INPUT

FE88            I08OLD: DS   2          ;PREVIOUS ACCEL TIME

FE8A            FDPOT:  DS   2          ;FEEDPOT
FE8C            AFDPOT: DS   2          ;FEEDPOT USED BY INTEGRALS
FE8E            DFDPOT: DS   2          ;DESIERED FEEDPOT
FE90            HFDPOT: DS   2          ;FEED HOLD FEED POT
FE92            FDSLEW: DS   2          ;FEEDPOT SLEW RATE

FE94            DWLPOT: DS   2          ;FEEDPOT FOR DWELLS (I07)

FE96            I00:    DS   2          ;FOLLOWING ERROR STOP
FE98            I01:    DS   2          ;IN POSITION
FE9A            I25:    DS   2          ;HOME OFFSET X-AXIS
FE9C            I06:    DS   4          ;FEEDRATE
FEA0            I20:    DS   2          ;X-AXIS PROPORTIONAL GAIN
FEA2            I08:    DS   2          ;ACCEL TIME
FEA4            I27:    DS   2          ;X-AXIS HNDWL. SCALE FACTOR
FEA6            I45:    DS   2          ;HOME OFFSET Y-AXIS
FEA8            I10:    DS   2          ;FOLLOWING ERROR LIGHT

FEAA            I04:    DS   1          ;HOME FEEDRATE
FEAB            I32:    DS   1          ;X-AXIS PWM LIMIT
FEAC            I51:    DS   1          ;Y-AXIS JOG FEEDRATE
FEAD            I31:    DS   1          ;X-AXIS JOG FEEDRATE
FEAE            I43:    DS   1           ;Y-AXIS INTEGRAL GAIN/PHASE
TAB.
FEAF            I41:    DS   1          ;Y-AXIS DIFF.GAIN/PHASE ADVANCE
FEB0            IBITS2: DS   1
     0580  I15      EQU  IBITS2.0  ;Y-AXIS DISABLE/PHASING CONTROL
     0581  I12      EQU  IBITS2.1  ;PWM FEQ. (24/6 KHZ)
     0582  I11      EQU  IBITS2.2  ;FEED HOLD SLEW MODE
     0583  I13      EQU  IBITS2.3  ;BRUSHLESS MOTOR ENABLE
     0584  PBFULL   EQU  IBITS2.4  ;PROGRAM BUFFER FULL
     0585  SBOPEN   EQU  IBITS2.5  ;SUBROUTINE LIBRARY OPEN
     0586  I14      EQU  IBITS2.6  ;ATD MODE (2 BITS)
     0587  I14B1    EQU  IBITS2.7
FEB1            I23:    DS   1          ;X-AXIS INTEGRAL GAIN
FEB2            I21:    DS   1          ;X-AXIS DIFFERENTIAL GAIN
FEB3            I33:    DS   1          ;X-AXIS PWM I2T LIMIT
FEB4            I22:    DS   1          ;X-AXIS AVERAGER
     05A4  I30      EQU  I22.4     ;X-AXIS GAIN SCALE
     05A5  I28      EQU  I22.5     ;X-AXIS HNDWHL. MODE (3 BITS)
     05A6  I28B1    EQU  I22.6
     05A7  I28B2    EQU  I22.7
FEB5            I42:    DS   1          ;Y-AXIS AVERAGER/ISD FLUX CURR.
     05AC  I50      EQU  I42.4     ;Y-AXIS GAIN SCALE
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        05AD    I48      EQU   I42.5      ;Y-AXIS HNDWHL MODE (3 BITS)
        05AE    I48B1    EQU   I42.6
        05AF    I48B2    EQU   I42.7
FEB6            I52:     DS    1          ;Y-AXIS PWM LIMIT/PHASE
                                          ;FINDING TIME
FEB7            I53:     DS    1          ;Y-AXIS PWM I2T LIMIT/PHASE
                                          ;FINDING  PWM
FEB8            IBITS1:  DS    1
        05C0    I24      EQU   IBITS1.0   ;X-AXIS HOME DIRECTION
        05C1    I44      EQU   IBITS1.1   ;Y-AXIS HOME DIRECTION
        05C2    I16      EQU   IBITS1.2   ;HANDWHEEL MODE (2 BITS)
        05C3    I16B1    EQU   IBITS1.3
        05C4    I02      EQU   IBITS1.4   ;MDI MODE
        05C5    I03      EQU   IBITS1.5   ;HANDSHAKE MODE
        05C6    I60      EQU   IBITS1.6   ;PLC ENABLE BIT
        05C7    I09      EQU   IBITS1.7   ;INTEGRATION MODE
 FEB9            BITS:    DS    2
        05C8    HOME2    EQU   BITS.0     ;Y-AXIS HOME COMPL.(MOMENTARY)
        05C9    HOME1    EQU   BITS.1     ;X-AXIS HOME COMPLETE
        05CA    BLKRDY   EQU   BITS.2     ;BLOCK READY
        05CB    BLKREQ   EQU   BITS.3     ;BLOCK REQUEST
        05CC    CIRIP    EQU   BITS.4     ;CIRCLE IN PROGRESS
        05CD    FHOLD    EQU   BITS.5     ;FEED HOLD IN PREGRESS
        05CE    RUN      EQU   BITS.6     ;RUN MODE
        05CF    ADIV     EQU   BITS.7     ;AUTO DIVEIDE
        05D0    TMODE    EQU   BITS+1.0   ;TIME MODE
        05D1    STEP     EQU   BITS+1.1   ;SINGLE STEP MODE
        05D2    BFULL    EQU   BITS+1.2   ;BUFFER FULL
        05D3    CIR180   EQU   BITS+1.3   ;CIRCLE > 180
        05D4    CIRDIR   EQU   BITS+1.4   ;CCW CIRCLE
        05D5    CIRCLE   EQU   BITS+1.5   ;CIRCLE MODE
        05D6    LIMIT    EQU   BITS+1.6   ;STOPPED BY LIMIT
        05D7    NOTRDY   EQU   BITS+1.7   ;NOT READY FOR MOVE
FEBB            SBITS1:  DS    1
        05D8    JOGP1    EQU   SBITS1.0   ;X-AXIS JOG PLUS
        05D9    JOGM1    EQU   SBITS1.1   ;X-AXIS JOG MINUS
        05DA    ODAC1    EQU   SBITS1.2   ;X-AXIS OPEN DAC MODE
        05DB    I2T1     EQU   SBITS1.3   ;X-AXIS CURRENT LIMIT
        05DC    FE1      EQU   SBITS1.4   ;X-AXIS FOLLOWING ERROR
        05DD    NTRDY1   EQU   SBITS1.5   ;X-AXIS NOT READY FOR MOVE
        05DE    NIPOS1   EQU   SBITS1.6   ;X-AXIS NOT IN POSITION
        05DF    RVZ1     EQU   SBITS1.7   ;X-AXIS ZERO VELOCITY
FEBC            SBITS2:  DS    1
        05E0    JOGP2    EQU   SBITS2.0   ;Y-AXIS JOG PLUS
        05E1    JOGM2    EQU   SBITS2.1   ;Y-AXIS JOG MINUS
        05E2    ODAC2    EQU   SBITS2.2   ;Y-AXIS OPEN DAC MODE
        05E3    I2T2     EQU   SBITS2.3   ;Y-AXIS CURRENT LIMIT
        05E4    FE2      EQU   SBITS2.4   ;Y-AXIS FOLLOWING ERROR
        05E5    NTRDY2   EQU   SBITS2.5   ;Y-AXIS NOT READY FOR MOVE
        05E6    NIPOS2   EQU   SBITS2.6   ;Y-AXIS NOT IN POSITION
        05E7    RVZ2     EQU   SBITS2.7   ;Y-AXIS ZERO VELOCITY

FEBD            P01BUF:  DS    1          ;P0,P1 BUFFER
FEBE            STIME:   DS    1          ;SERVO TIMER
FEBF            SFLAG:   DS    1          ;SERVO FLAG
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                05FC    DWELL   EQU   SFLAG.4    ;DWELL IN PROGRESS
                05FD    STPREQ  EQU   SFLAG.5    ;STOP REQUESTED
                05FE    DWEREQ  EQU   SFLAG.6    ;DWELL REQUESTED
                05FF    MEMREQ  EQU   SFLAG.7    ;MEMORY MODIFY REQUESTED

                                ORG   0FEC0H
FEC0                    MR0:    DS    1          ;REGISTER BANK 3
FEC1                    MR1:    DS    1
FEC2                    MR2:    DS    1
FEC3                    MR3:    DS    1
FEC4                    MR4:    DS    1
FEC5                    MR5:    DS    1
FEC6                    MR6:    DS    1
FEC7                    MR7:    DS    1
FEC8                    MR8:    DS    1
FEC9                    MR9:    DS    1
FECA                    MR10:   DS    1
FECB                    MR11:   DS    1
FECC                    MR12:   DS    1
FECD                    MR13:   DS    1
FECE                    MR14:   DS    1
FECF                    MR15:   DS    1

        FEC0    MRP0    EQU   MR0
        FEC2    MRP1    EQU   MR2
        FEC4    MRP2    EQU   MR4
        FEC6    MRP3    EQU   MR6
        FEC8    MRP4    EQU   MR8
        FECA    MRP5    EQU   MR10
        FECC    MRP6    EQU   MR12
        FECE    MRP7    EQU   MR14

FED0                    SR0:    DS    1          ;REGISTER BANK 2
FED1                    SR1:    DS    1
FED2                    SR2:    DS    1
FED3                    SR3:    DS    1
FED4                    SR4:    DS    1
FED5                    SR5:    DS    1
FED6                    SR6:    DS    1
FED7                    SR7:    DS    1
FED8                    SR8:    DS    1
FED9                    SR9:    DS    1
FEDA                    SR10:   DS    1
FEDB                    SR11:   DS    1
FEDC                    SR12:   DS    1
FEDD                    SR13:   DS    1
FEDE                    SR14:   DS    1
FEDF                    SR15:   DS    1

        FED0    SRP0    EQU   SR0
        FED2    SRP1    EQU   SR2
        FED4    SRP2    EQU   SR4
        FED6    SRP3    EQU   SR6
        FED8    SRP4    EQU   SR8
        FEDA    SRP5    EQU   SR10
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
         FEDC    SRP6    EQU   SR12
         FEDE    SRP7    EQU   SR14
         FED8    PENC2   EQU   SRP4    ;PREVIOUS AVERAGE ENCODER Y-AXIS
         FED8    PHASA2  EQU   SRP4    ;PHASE ADVANCE #2
         FEDA    PENC1   EQU   SRP5    ;PREVIOUS ENCODER Y-AXIS

FEE0             IR0:    DS    1       ;REGISTER BANK 1
FEE1             IR1:    DS    1
FEE2             IR2:    DS    1
FEE3             IR3:    DS    1       ;INPUT BUFFER LENGTH
FEE4             IR4:    DS    1
FEE5             IR5:    DS    1
FEE6             IR6:    DS    1
FEE7             IR7:    DS    1
FEE8             IR8:    DS    1
FEE9             IR9:    DS    1
FEEA             IR10:   DS    1
FEEB             IR11:   DS    1
FEEC             IR12:   DS    1
FEED             IR13:   DS    1
FEEE             IR14:   DS    1
FEEF             IR15:   DS    1

         FEE0    IRP0    EQU   IR0
         FEE2    IRP1    EQU   IR2
         FEE4    IRP2    EQU   IR4
         FEE6    IRP3    EQU   IR6
         FEE8    IRP4    EQU   IR8
         FEEA    IRP5    EQU   IR10
         FEEC    IRP6    EQU   IR12
         FEEE    IRP7    EQU   IR14
         FEE0    IOBITS  EQU   IR0
         0703    RS232   EQU   IOBITS.3  ;RS232 MODE
         FEE2    ADDRES  EQU   IR2       ;CARD ADDRESS
         FEEA    EACNFG  EQU   IR10
         FEEB    ACC     EQU   IR11
         FEE4    MOVTIM  EQU   IRP2      ;MOVE TIME

FEF0             PR0:    DS    1         ;REGISTER BANK 0
FEF1             PR1:    DS    1
FEF2             PR2:    DS    1
FEF3             PR3:    DS    1
FEF4             PR4:    DS    1
FEF5             PR5:    DS    1
FEF6             PR6:    DS    1
FEF7             PR7:    DS    1
FEF8             PR8:    DS    1
FEF9             PR9:    DS    1
FEFA             PR10:   DS    1
FEFB             PR11:   DS    1
FEFC             PR12:   DS    1
FEFD             PR13:   DS    1
FEFE             PR14:   DS    1
FEFF             PR15:   DS    1
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
FEF0    PRP0    EQU  PR0
FEF2    PRP1    EQU  PR2
FEF4    PRP2    EQU  PR4
FEF6    PRP3    EQU  PR6
FEF8    PRP4    EQU  PR8
FEFA    PRP5    EQU  PR10
FEFC    PRP6    EQU  PR12
FEFE    PRP7    EQU  PR14

FEF0    DPOSL2  EQU  PRP0      ;Y-AXIS POSITION RESIDUAL
FEF6    PHW2    EQU  PRP3      ;Y-AXIS PREV. HNDWHL POS.
FEF8    I47     EQU  PRP4      ;Y-AXIS HW SF/# OF POLES*PHASES
FEFA    I40     EQU  PRP5      ;Y-AXIS PRO.GAIN/# OF LINES/REV
FEFC    ACRES2  EQU  PRP6      ;Y-AXIS ACCEL RESIDUAL
FEFE    DAC2R   EQU  PRP7      ;Y-AXIS DAC RESIDUAL
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
**************************************************************************
                    EXTERNAL RAM 11/09/88
                         STANDARD
**************************************************************************

                        ORG   0A000H

A000                    DS    128       ;EPOS BUFFERS
A080            TPNTR1: DS    1         ;EPOS POINTER X-AXIS
A081            TPNTR2: DS    1         ;EPOS POINTER Y-AXIS

A082            DISCNT: DS    1         ;DISPLAY COUNTER
A083            LSTMOD: DS    1         ;EXTENDED READ MODE
A084            VTIME:  DS    2         ;VELOCITY DISPLAY TIME
A086            DISTIM: DS    2         ;DISPLAY TIMER
A088            DSPOS2: DS    4         ;Y-AXIS DISPLAY POSITION
A08C            DSPOS1: DS    4         ;X-AXIS DISPLAY POSITION
A090            DSVEL2: DS    4         ;Y-AXIS DISPLAY VEL
A094            DSVEL1: DS    4         ;X-AXIS DISPLAY VEL
A098            BPOS2:  DS    4         ;Y-AXIS POSITION BIAS
A09C            BPOS1:  DS    4         ;X-AXIS POSITION BIAS
A0A0            HPOS2:  DS    4         ;Y-AXIS HOME POSITION
A0A4            HPOS1:  DS    4         ;X-AXIS HOME POSITION
A0A8            CMVTIM: DS    4         ;CONTINUOUS MOVE TIME
A0AC            DTHETA: DS    4         ;DELTA ANGLE FOR CIRCLE
A0B0            THETA:  DS    4         ;ANGLE FOR CIRCLE
A0B4            THETAY: DS    4         ;Y-AXIS ANG.OFFSET FOR ELLIPSE
A0B8            RADX:   DS    4         ;X-AXIS RADIUS
A0BC            RADY:   DS    4         ;Y-AXIS RADIUS
A0C0            CTHETA: DS    4         ;COS(ANGLE)
A0C4            STHETA: DS    4         ;SIN(ANGLE)
A0C8            TCOUNT: DS    2
A0CA            ADIVP2: DS    4         ;AUTO DIVIDE POSITION Y-AXIS
A0CE            ADIVP1: DS    4         ;AUTO DIVIDE POSITION X-AXIS
A0D2            ADCNT:  DS    2         ;AUTO DIVIDE COUNTER
A0D4            LSTCTR: DS    2         ;EXTENDED READ COUNTER
A0D6            LSTPTR: DS    2         ;EXTENDED READ POINTER
A0D8            EDPTR:  DS    2         ;EDIT POINTER
A0DA            PDPTR:  DS    2         ;PREVIOUS PROGRAM POINTER
A0DC            DBUFF:  DS    12        ;DISPLAY BUFFER
A0E8            XTEMP:  DS    22
A0FE            RAMEND: DS    2         ;END OF MEMORY ADDRESS

A100            XRBUFF: DS    1         ;COM BUFFER
A101            XTBUFF: DS    53
A136            I17:    DS    4         ;DISPLAY MODE
A13A            I34:    DS    2         ;X-AXIS BACKLASH
A13C            I35:    DS    4         ;X-AXIS POS.POSITION LIMIT
A140            I36:    DS    4         ;X-AXIS NEG.POSITION LIMIT
A144            I37:    DS    4         ;X-AXIS RANGE
A148            I38:    DS    9         ;X-AXIS POSITION FORMAT
A151            I54:    DS    2         ;Y-AXIS BACKLASH
A153            I55:    DS    4         ;Y-AXIS POS.POSITION LIMIT
A157            I56:    DS    4         ;Y-AXIS NEG.POSITION LIMIT
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A15B            I57:    DS   4          ;Y-AXIS RANGE
A15F            I58:    DS   9          ;Y-AXIS POSITION FORMAT
A168            IBITS3: DS   1
A169            MTABLE: DS   64         ;SYMBOL TABLE M00-M63
A1A9            PTABLE: DS   60         ;PARAMETER TABLE P01-P15
A1E5            IEND:   DS   27

A200            FREE:   DS   36         ;FREE RAM SPACE
A224            DEPTH:  DS   1          ;PLC DEPTH COUNTER
A225            STACK:  DS   129        ;SERVO STACK
A2A6            STKEND:
A2A6            SSTPTR: DS   1          ;SERVO STACK POINTERS
A2A7            STKPTR: DS   1

A2A8            NDRNV2: DS   6          ;NEXT DESIRED VELOCITY Y-AXIS
A2AE            NDRNV1: DS   6          ;NEXT DESIRED VELOCITY X-AXIS
A2B4            NACEL2: DS   6          ;NEXT DESIRED ACCEL Y-AXIS
A2BA            NACEL1: DS   6          ;NEXT DESIRED ACCEL X-AXIS
A2C0            NDPOS2: DS   4          ;NEXT DESIRED POSITION Y-AXIS
A2C4            NDPOS1: DS   4          ;NEXT DESIRED POSITION X-AXIS
A2C8            NDCEL2: DS   6          ;NEXT DESIRED DECCEL Y-AXIS
A2CE            NDCEL1: DS   6          ;NEXT DESIRED DECCEL X-AXIS
A2D4            NMVTIM: DS   6          ;NEXT MOVE TIME
A2DA            DWLTIM: DS   4          ;NEXT DWELL TIME

A2DE            TIME:   DS   4          ;FEEDRATE/TIME
      A2DE      FDRATE  EQU  TIME
A2E2            DDPOS2: DS   4          ;DELTA DESIRED POSITION Y-AXIS
A2E6            DDPOS1: DS   4          ;DELTA DESIRED POSITION X-AXIS
A2EA            NSPEED: DS   4          ;NEW DESIRED SPEED
A2EE            TEMP:   DS   8

A2F6            BGPTR:  DS   2          ;PLC POINTER
A2F8            DTPTR:  DS   2          ;PROGRAM POINTER
A2FA            STPTR:  DS   2          ;STORAGE POINTER
A2FC            LBPTR:  DS   2          ;SUBROUTINE LIBRARY POINTER
A2FE            STEND:  DS   2          ;STORAGE END ADDRESS

A300            STABLE: DS   128        ;SYMBOL TABLE L00-L63
A380            CTABLE: DS   128        ;SYMBOL TABLE C00-C63

A400            PSTART: DS   4          ;PROGRAM BUFFER START ADDRESS


      A800      P2KEND  EQU  0A800H
      C000      P8KEND  EQU  0C000H
      C000      EARTST  EQU  0C000H
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
*****************************************************************************

                    MINI INTERNAL RAM 10/19/88
                          PARABOLIC

*****************************************************************************

                          ORG   0FE00H  ;INTERNAL RAM
FE00              RBUFF:  DS    32
          FE00    TBUFF   EQU   RBUFF
                                        ;START OF DIRECT ACCESS RAM
FE20              PTM1:   DS    2       ;PREVIOUS ENCODER EDGE TIME X-AXIS
FE22              PTM2:   DS    2       ;PREVIOUS ENCODER EDGE TIME Y-AXIS
          FE22    PPHASE  EQU   TPTM2   ;PREVIOUS PHASE ENCODER READING
FE24              PDPOS1: DS    2       ;PREVIOUS DESIRED POSITION X-AXIS
FE26              PDPOS2: DS    2       ;PREVIOUS DESIRED POSITION Y-AXIS
          FE26    PHASE   EQU   PDPOS2  ;PRESENT PHASE POSITION
FE28              DACR1:  DS    2       ;RESIDUAL DAC OUTPUT X-AXIS
FE2A              DACR2:  DS    2       ;RESIDUAL DAC OUTPUT Y-AXIS
          FE2A    DAC     EQU   DACR2   ;DAC OUTPUT FOR PHASED MOTORS
FE2C              PRES1:  DS    4       ;POSITION RESIDUAL X-AXIS
FE30              PRES2:  DS    4       ;POSITION RESIDUAL Y-AXIS
          FE30    SLIP    EQU   PRES2   ;SLIP FREQUENCY
FE34              VRES1:  DS    2       ;VEL RESIDUAL X-AXIS
FE36              VRES2:  DS    2       ;VEL RESIDUAL Y-AXIS
          FE36    PHASAD  EQU   VRES2   ;PHASE ADVANCE
FE38              ARES1:  DS    2       ;ACCEL RESIDUAL X-AXIS
FE3A              ARES2:  DS    2       ;ACCEL RESIDUAL Y-AXIS
          FE3A    PHASOA  EQU   ARES2
FE3C              HWRES1: DS    1       ;HANDWHEEL RESIDUAL X-AXIS
FE3D              HWRES2: DS    1       ;HANDWHEEL RESIDUAL Y-AXIS
FE3E              PHW1:   DS    2       ;PREVIOUS HANDWHEEL X-AXIS
FE40              PENC1:  DS    2       ;PREVIOUS ENCODER X-AXIS
FE42              IPOS1:  DS    4       ;ERROR INTEGRAL X-AXIS
FE46              PDVEL1: DS    2       ;PREVIOUS DESIRED VEL X-AXIS
FE48              APOS1:  DS    4       ;ABSOLUTE POSITION X-AXIS
FE4C              APOS2:  DS    4       ;ABSOLUTE POSITION Y-AXIS
FE50              EPOS1:  DS    4       ;ERROR POSITION X-AXIS
FE54              EPOS2:  DS    4       ;ERROR POSITION Y-AXIS
          FE54    PHRES   EQU   EPOS2   ;RESIDUAL PHASE POSITION

FE58              DPOS1:  DS    4       ;DESIRED POSITION X-AXIS
FE5C              DPOS2:  DS    4       ;DESIRED POSITION Y-AXIS
FE60              DVEL1:  DS    4       ;DESIRED VEL X-AXIS
FE64              DVEL2:  DS    4       ;DESIRED VEL Y-AXIS
FE68              ACCEL1: DS    6       ;DESIRED ACCEL X-AXIS
FE6E              ACCEL2: DS    6       ;DESIRED ACCEL Y-AXIS
FE74              JCCEL1: DS    8       ;DESIRED ACCEL DOT X-AXIS
FE7C              JCCEL2: DS    8       ;DESIRED ACCEL DOT Y-AXIS
FE84              NMVTIM: DS    4       ;NEW MOVE TIME

FE88              MOVTIM: DS    4       ;MOVE TIME

FE8C              FATD0:  DS    2       ;FILTERED ATD #0 INPUT
FE8E              FATD1:  DS    2       ;FILTERED ATD #1 INPUT
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
FE90            ATD2:    DS    1          ;RAW ATD #2 INPUT
FE91            ATD3:    DS    1          ;RAW ATD #3 INPUT

FE92            DFDPOT:  DS    2          ;DESIRED FEED POT
FE94            FDSLEW:  DS    2          ;FEED POT SLEW RATE
FE96            AFDPOT:  DS    2          ;FEED POT USED FOR INTERATION
FE98            HFDPOT:  DS    2          ;FEED HOLD FEED POT

          FF8E  I05      EQU   0FF8EH     ;SERVO TIME
FE9A            I07:     DS    2          ;DWELL TIME BASE
FE9C            I20:     DS    2          ;AXIS #1 PROPORTIONAL GAIN
FE9E            I27:     DS    2          ;AXIS #1 HANDWHEEL SCALE FACTOR
FEA0            I40:     DS    2          ;AXIS #2 PROPORTIONAL GAIN
FEA2            I45:     DS    2          ;AXIS #2 HOME OFFSET/SLIP GAIN
FEA4            I47:     DS    2          ;AXIS #2 HANDWHEEL SCALE FACTOR

FEA6            I21:     DS    1          ;AXIS #1 FEEDBACK GAIN
FEA7            I22:     DS    1          ;AXIS #1 FEEDFORWARD
FEA8            I23:     DS    1          ;AXIS #1 INTEGRAL GAIN
FEA9            I32:     DS    1          ;PWM #1 LIMIT
FEAA            I33:     DS    1          ;I2T PWM #1 LIMIT
FEAB            I41:     DS    1          ;AXIS #2 FEEDBACK GAIN
FEAC            I42:     DS    1          ;AXIS #2 FEEDFORWARD
FEAD            I43:     DS    1          ;AXIS #2 INTEGRAL GAIN
FEAE            I52:     DS    1          ;PWM #2 LIMIT
FEAF            I53:     DS    1          ;I2T PWM #2 LIMIT

FEB0            IBITS1:  DS    1
          0580  I02      EQU   IBITS1.0 ;MDI MODE
          0581  I03      EQU   IBITS1.1 ;HANDSHAKE MODE (2 BITS)
          0582  I03B1    EQU   IBITS1.2
          0583  I11      EQU   IBITS1.3 ;FEED HOLD SLEW
          0584  I13      EQU   IBITS1.4 ;BRUSHLESS MOTOR ENABLE
          0585  I15      EQU   IBITS1.5 ;Y-AXIS DISABLE/(TRAP/LOOKUP
                                        ;PHASING)
          0586  I14      EQU   IBITS1.6 ;A TO D MODE (2 BITS)
          0587  I14B1    EQU   IBITS1.7

FEB1            IBITS2:  DS    1
          0588  I24      EQU   IBITS2.0 ;HOME DIRECTION #1
          0589  I28      EQU   IBITS2.1 ;HANDWHEEL MODE #1 (2 BITS)
          058A  I28B1    EQU   IBITS2.2
          058B  I09      EQU   IBITS2.3 ;INTERGATOR MODE
          058C  I44      EQU   IBITS2.4 ;HOME DIRECTION #2
          058D  I48      EQU   IBITS2.5 ;HANDWHEEL MODE #2 (2 BITS)
          058E  I48B1    EQU   IBITS2.6
          058F  PBFULL   EQU   IBITS2.7 ;PROGRAM BUFFER CLOSED FLAG

FEB2            IBITS3:  DS    1
          0590  I18      EQU   IBITS3.0 ;DISPLAY MODE (2 BITS)
          0591  I18B1    EQU   IBITS3.1
          0592  I16      EQU   IBITS3.2 ;FEEDPOT HANDWHEEL MODE
          0593  I16B1    EQU   IBITS3.3
          0594  I19      EQU   IBITS3.4 ;HIGEST CARD IN ADDRESSED
                                        ;(4 BITS)
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
FEB3            I30:    DS   1        ;AXIS #1 DES. VEL FEEDBACK GAIN
FEB4            I50:    DS   1        ;AXIS #2 DES. VEL FEEDBACK GAIN

FEB5            BITS:   DS   2
        05A8    HOME2   EQU  BITS.0   ;HOME AXIS #2 FLAG
        05A9    HOME1   EQU  BITS.1   ;HOME AXIS #1 FLAG
        05AA    CMOVE   EQU  BITS.2   ;CONTINUOS MOVE FLAG
        05AB    BLKREQ  EQU  BITS.3   ;BLOCK REQUEST FLAG
        05AC    ADIV    EQU  BITS.4   ;AUTO DIVIDE MOVE FLAG
        05AD    FHOLD   EQU  BITS.5   ;FEED HOLD FLAG
        05AE    RUN     EQU  BITS.6   ;RUN FLAG
        05AF    TIMER   EQU  BITS.7   ;TIMER ACTIVE FLAG

        05B0    TMODE   EQU  BITS+1.0 ;TIME MODE FLAG
        05B1    STEP    EQU  BITS+1.1 ;SINGLE STEP FLAG
        05B2    BFULL   EQU  BITS+1.2 ;BUFFER FULL FLAG
        05B3    FRATE   EQU  BITS+1.3 ;PARABOLIC MOVE FLAG
        05B4    LIMIT   EQU  BITS+1.4 ;LIMIT SWITCH FLAG
        05B5    NTRDY1  EQU  BITS+1.5 ;NOT READY AXIS #1 FLAG
        05B6    NTRDY2  EQU  BITS+1.6 ;NOT READY AXIS #2 FLAG
        05B7    NOTRDY  EQU  BITS+1.7 ;NOT READY FLAG

FEB7            SBITS1: DS   1
        05B8    JOGP1   EQU  SBITS1.0 ;X-AXIS POSITIVE JOG
        05B9    JOGM1   EQU  SBITS1.1 ;X-AXIS NEGATIVE JOG
        05BA    LMTP1   EQU  SBITS1.2 ;X-AXIS POSITIVE LIMIT
        05BB    LMTM1   EQU  SBITS1.3 ;X-AXIS NEGATIVE LIMIT
        05BC    FE1     EQU  SBITS1.4 ;X-AXIS FOLLOWING ERROR WARNING
        05BD    I2T1    EQU  SBITS1.5 ;X-AXIS CURRENT LIMIT
        05BE    NIPOS1  EQU  SBITS1.6 ;X-AXIS NOT IN POSITION
        05BF    RVZ1    EQU  SBITS1.7 ;X-AXIS RUN VEL ZERO

FEB8            SBITS2: DS   1
        05C0    JOGP2   EQU  SBITS2.0 ;Y-AXIS POSITIVE JOG
        05C1    JOGM2   EQU  SBITS2.1 ;Y-AXIS NEGATIVE JOG
        05C2    LMTP2   EQU  SBITS2.2 ;Y-AXIS POSITIVE LIMIT
        05C3    LMTM2   EQU  SBITS2.3 ;Y-AXIS NEGATIVE LIMIT
        05C4    FE2     EQU  SBITS2.4 ;Y-AXIS FOLLOWING ERROR WARNING
        05C5    I2T2    EQU  SBITS2.5 ;Y-AXIS CURRENT LIMIT
        05C6    NIPOS2  EQU  SBITS2.6 ;Y-AXIS NOT IN POSITION
        05C7    RVZ2    EQU  SBITS2.7 ;Y-AXIS RUN VEL ZERO

FEB9            XBITS:  DS   2
        05C8    EDIT    EQU  XBITS.0 ;EDIT MODE
        05C9    INSERT  EQU  XBITS.1 ;EDIT INSERT MODE
        05CA    HMDONE  EQU  XBITS.2 ;EXTENDED READ FLAG
        05CB    CIRCLE  EQU  XBITS.3 ;CIRCLE MODE
        05CC    WAIT1   EQU  XBITS.4 ;X-AXIS WAIT MOVE
        05CD    WAIT2   EQU  XBITS.5 ;Y-AXIS WAIT MOVE
        05CE    BACK1   EQU  XBITS.6 ;X-AXIS BACKLASH COMP
        05CF    BACK2   EQU  XBITS.7 ;Y-AXIS BACKLASH COMP

        05D0    ADREQ   EQU  XBITS+1.;AUTO DIVIDE REQUEST BIT 0
        05D1    EXMOVE  EQU  XBITS+1.;EXTENDED MOVE FLAG (2 BITS)
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        05D2    EXMOV1  EQU   XBITS+1.
        05D3    HMDONE  EQU   XBITS+1.   ;HOME DONE FLAG
        05D4    CIRDIR  EQU   XBITS+1.4 ;CIRCLE DIRECTION FLAG
        05D5    CIR180  EQU   XBITS+1.5 ;CIRCLE GREATER THAN 180 FLAG
        05D6    FREEZE  EQU   XBITS+1.6 ;POSITION FREEZE FLAG
        05D7    SBOPEN  EQU   XBITS+1.7 ;SUBROUTINE BUFFER OPEN FLAG

FEBB            SFLAG:  DS    1
        05DB    DWEREQ  EQU   SFLAG.3   ;DWELL REQUESTED
        05DC    HOME    EQU   SFLAG.4   ;HOME IN PROGRESS
        05DD    DWELL   EQU   SFLAG.5   ;DWELL IN PROGRESS
        05DE    CMVREQ  EQU   SFLAG.6   ;CONTINUOS MOVE REQUESTED
        05DF    MEMREQ  EQU   SFLAG.7   ;MEMORY MODIFY REQUESTED

FEBC            BUFPTR: DS    1
        05E5    STPREQ  EQU   BUFPTR.5  ;STOP REQUESTED

FEBD            DBAND1: DS    1
        05ED    BDATA   EQU   DBAND1.5  ;BUFFER DATA FLAG
        05EE    DMAIO   EQU   DBAND1.6  ;DMA IO IN PROGRESS
        05EF    DMAOUT  EQU   DBAND1.7  ;DMA OUTPUT MODE

FEBE            DBAND2: DS    1
        05F5    PLCFLG  EQU   DBAND2.5  ;PLC BUFFER OPEN FLAG
        05F6    RESULT  EQU   DBAND2.6  ;PLC RESULT DECODE FLAG
        05F7    PLCCMD  EQU   DBAND2.7  ;PLC COMMAND FLAG

FEBF            P01BUF: DS    1
        05FC    UMOVIE  EQU   P01BUF.4  ;MOVE UNTIL INTERRUPT ENABLE
        05FD    TORQUE  EQU   P01BUF.5  ;TORQUE MOVE IN PROGRESS BIT

                        ORG   0FEC0H
FEC0            MR0:    DS    1         ;REGISTER BANK 3
FEC1            MR1:    DS    1
FEC2            MR2:    DS    1
FEC3            MR3:    DS    1
FEC4            MR4:    DS    1
FEC5            MR5:    DS    1
FEC6            MR6:    DS    1
FEC7            MR7:    DS    1
FEC8            MR8:    DS    1
FEC9            MR9:    DS    1
FECA            MR10:   DS    1
FECB            MR11:   DS    1
FECC            MR12:   DS    1
FECD            MR13:   DS    1
FECE            MR14:   DS    1
FECF            MR15:   DS    1

        FEC0    MRP0    EQU   MR0
        FEC2    MRP1    EQU   MR2
        FEC4    MRP2    EQU   MR4
        FEC6    MRP3    EQU   MR6
        FEC8    MRP4    EQU   MR8
        FECA    MRP5    EQU   MR10
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
        FECC    MRP6    EQU   MR12
        FECE    MRP7    EQU   MR14

FED0            SR0:    DS    1         ;REGISTER BANK 2
FED1            SR1:    DS    1
FED2            SR2:    DS    1
FED3            SR3:    DS    1
FED4            SR4:    DS    1
FED5            SR5:    DS    1
FED6            SR6:    DS    1
FED7            SR7:    DS    1
FED8            SR8:    DS    1
FED9            SR9:    DS    1
FEDA            SR10:   DS    1
FEDB            SR11:   DS    1
FEDC            SR12:   DS    1
FEDD            SR13:   DS    1
FEDE            SR14:   DS    1
FEDF            SR15:   DS    1

        FED0    SRP0    EQU   SR0
        FED2    SRP1    EQU   SR2
        FED4    SRP2    EQU   SR4
        FED6    SRP3    EQU   SR6
        FED8    SRP4    EQU   SR8
        FEDA    SRP5    EQU   SR10
        FEDC    SRP6    EQU   SR12
        FEDE    SRP7    EQU   SR14

FEE0            IR0:    DS    1         ;REGISTER BANK 1
FEE1            IR1:    DS    1
FEE2            IR2:    DS    1
FEE3            IR3:    DS    1
FEE4            IR4:    DS    1
FEE5            IR5:    DS    1
FEE6            IR6:    DS    1
FEE7            IR7:    DS    1
FEE8            IR8:    DS    1
FEE9            IR9:    DS    1
FEEA            IR10:   DS    1
FEEB            IR11:   DS    1
FEEC            IR12:   DS    1
FEED            IR13:   DS    1
FEEE            IR14:   DS    1
FEEF            IR15:   DS    1

        FEE0    IRP0    EQU   IR0
        FEE2    IRP1    EQU   IR2
        FEE4    IRP2    EQU   IR4
        FEE6    IRP3    EQU   IR6
        FEE8    IRP4    EQU   IR8
        FEEA    IRP5    EQU   IR10
        FEEC    IRP6    EQU   IR12
        FEEE    IRP7    EQU   IR14
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
            FEE0    IOBITS  EQU  IR0
            0701    AADRSD  EQU  IOBITS.1    ;ALL CARDS ADDRESSED
            0702    ADRSED  EQU  IOBITS.2    ;THIS CARD ADDRESSED
            0703    RS232   EQU  IOBITS.3    ;RS232 MODE

            FEE2    ADDRES  EQU  IR2         ;CARD ADDRESS
            FEE3    BLNGTH  EQU  IR3         ;INPUT BUFFER LENGTH

            FEE4    STIME   EQU  IR4         ;SERVO TIMER
            FEE5    ACC     EQU  IR5
            FEE6    PHW2    EQU  IRP3        ;PREVIOUS HADNWHEEL Y-AXIS
            FEE8    PENC2   EQU  IRP4        ;PREVIOUS ENCODER Y-AXIS
            FEEA    PENC4   EQU  IRP5        ;PREVIOUS AV.ENCODER Y-AXIS

FEF0                PR0:    DS   1           ;REGISTER BANK 0
FEF1                PR1:    DS   1
FEF2                PR2:    DS   1
FEF3                PR3:    DS   1
FEF4                PR4:    DS   1
FEF5                PR5:    DS   1
FEF6                PR6:    DS   1
FEF7                PR7:    DS   1
FEF8                PR8:    DS   1
FEF9                PR9:    DS   1
FEFA                PR10:   DS   1
FEFB                PR11:   DS   1
FEFC                PR12:   DS   1
FEFD                PR13:   DS   1
FEFE                PR14:   DS   1
FEFF                PR15:   DS   1

            FEF0    PRP0    EQU  PR0
            FEF2    PRP1    EQU  PR2
            FEF4    PRP2    EQU  PR4
            FEF6    PRP3    EQU  PR6
            FEF8    PRP4    EQU  PR8
            FEFA    PRP5    EQU  PR10
            FEFC    PRP6    EQU  PR12
            FEFE    PRP7    EQU  PR14

            FEF4    IPOS2   EQU  PRP2        ;ERROR INTEGRAL Y-AXIS
            FEF8    PDVEL2  EQU  PRP4        ;PREV. DESIRED VEL Y-AXIS
            FEFA    FDPOT   EQU  PRP5        ;PRESENT FEED POT
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
**************************************************************************
                        EXTERNAL RAM 05/09/89
                             PARABOLIC
**************************************************************************
                        ORG   0A000H

A000            ADPOS1: DS    4       ;NEXT DESIRED POSITION X-AXIS
A004            ADPOS2: DS    4       ;NEXT DESIRED POSITION Y-AXIS
A008            ADVEL1: DS    4       ;NEXT DESIRED VEL X-AXIS
A00C            ADVEL2: DS    4       ;NEXT DESIRED VEL Y-AXIS
A010            NACEL1: DS    6       ;NEXT DESIRED ACCEL X-AXIS
A016            NACEL2: DS    6       ;NEXT DESIRED ACCEL Y-AXIS
A01C            NJCEL1: DS    8       ;NEXT DESIRED ACCEL DOT X-AXIS
A024            NJCEL2: DS    8       ;NEXT DESIRED ACCEL DOT Y-AXIS
A02C            NMVTM1: DS    4       ;NEXT DESIRED MOVE TIME
A030                    DS    144
A0C0            NDPOS1: DS    4       ;FINAL DESIRED POSITION X-AXIS
A0C4            NDPOS2: DS    4       ;FINAL DESIRED POSITION Y-AXIS
A0C8            NDVEL1: DS    4       ;FINAL DESIRED VEL X-AXIS
A0CC            NDVEL2: DS    4       ;FINAL DESIRED VEL Y-AXIS
A0D0            CCOUNT: DS    2       ;CONTINUOS MOVE COUNT
A0D2            CDPOS1: DS    6       ;CONT.MOVE DESIRED POS. X-AXIS
A0D8            DDPOS1: DS    6       ;CONT.MOVE DELTA POS. X-AXIS
A0DE            CDPOS2: DS    6       ;CONT.MOVE DESIRED POS.Y-AXIS
A0E4            DDPOS2: DS    6       ;CONT.MOVE DELTA POS.Y-AXIS
A0EA            FDRATE: DS    4       ;FEEDRATE/TIME
      A0EA      TIME    EQU   FDRATE
A0EE            TEMP:   DS    16
A0FE            ATIME:  DS    2       ;ACCEL TIME

A100            STABLE: DS    128     ;SYMBOL TABLE L00-L63
A180            CTABLE: DS    128     ;SYMBOL TABLE C00-C63

A200            MTABLE: DS    64      ;SYMBOL TABLE M00-M63
A240            PTABLE: DS    60      ;PARAMETER TABLE P01-P15
A27C            STEND:  DS    2       ;STORAGE END POINTER
A27E            LBPTR:  DS    2       ;SUBROUTINE LIBRARY POINTER
A280            BGPTR:  DS    2       ;PLC POINTER

A282            STACK:  DS    124     ;SERVO STACK
A2FE            STKEND:
A2FE            SSTPTR: DS    1       ;SERVO STACK POINTERS
A2FF            STKPTR: DS    1

A300            XRBUFF: DS    65      ;COM BUFFERS
      A300      XTBUFF  EQU   XRBUFF
A341            DISTIM: DS    2       ;DISPLAY TIMER
A343            DISCNT: DS    1       ;DISPLAY COUNTER
A344            BDPTR:  DS    9
A34D            BPOS1:  DS    4       ;BIAS POSITION X-AXIS
A351            BPOS2:  DS    4       ;BIAS POSITION Y-AXIS
A355            HPOS1:  DS    4       ;HOME POSITION X-AXIS
A359            HPOS2:  DS    4       ;HOME POSITION Y-AXIS
A35D            LSTMOD: DS    1       ;EXTENTED READ MODE
A35E            VTIME:  DS    2       ;VELOCITY TIMER
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A360            DSPOS1: DS   4       ;DISPLAY POSITION X-AXIS
A364            DSPOS2: DS   4       ;DISPLAY POSITION Y-AXIS
A368            DSVEL1: DS   4       ;DISPLAY VEL X-AXIS
A36C            DSVEL2: DS   4       ;DISPLAY VEL Y-AXIS
A370            FXRATE: DS   4       ;PARABOLIC FEEDRATE X-AXIS
A374            FYRATE: DS   4       ;PARABOLIC FEEDRATE Y-AXIS
A378            THETA:  DS   4       ;CIRCLE ANGLE
A37C            DTHETA: DS   6       ;CIRCLE DESIRED ANGLE
A382            DDTHET: DS   6       ;CIRCLE DELTA ANGLE
A388            RADX:   DS   4       ;X-AXIS RADIUS
A38C            RADY:   DS   4       ;Y-AXIS RADIUS
A390            TCOUNT: DS   2
A392            ADIVP1: DS   4       ;AUTO DIVIDE POSITION X-AXIS
A396            ADIVP2: DS   4       ;AUTO DIVIDE POSITION Y-AXIS
A39A            ADCNT:  DS   2       ;AUTO DIVIDE COUNT
A39C            LSTCTR: DS   2       ;EXTENTED READ COUNTER
A39E            LSTPTR: DS   2       ;EXTENTED READ POINTER
A3A0            EDPTR:  DS   2       ;EDIT POINTER
A3A2            PDPTR:  DS   2       ;PREVIOUS PROGRAM POINTER
A3A4            RAMEND: DS   2       ;END OF RAM ADDRESS
A3A6            DEPTH:  DS   1       ;PLC DEPTH COUNTER
A3A7            FREE:   DS   6       ;FREE SPACE
A3AD                    DS   6
A3B3            DTPTR:  DS   2       ;PROGRAM POINTER

A3B5            STPTR:  DS   2       ;STORAGE POINTER
A3B7            I60:    DS   1       ;X-AXIS DEADBAND
A3B8            I61:    DS   1       ;Y-AXIS DEADBAND
A3B9            IBITS4: DS   1
A3BA            I80:    DS   1       ;PHASE LOCK GAIN

A3BB            I00:    DS   2       ;FOLLOWING ERROR
A3BD            I01:    DS   2       ;IN POSITION BAND
A3BF            I04:    DS   1       ;HOME FEEDRATE
A3C0            I06:    DS   4       ;SYSTEM FEEDRATE
A3C4            I08:    DS   2       ;ACCEL TIME
A3C6            I10:    DS   2       ;FOLLOWING ERROR WARNING
A3C8            I12:    DS   2       ;MOVE DELAY TIME
A3CA            I17:    DS   4       ;DISPLAY MODE
A3CE            I25:    DS   2       ;X-AXIS HOME OFFSET
A3D0            I31:    DS   1       ;X-AXIS JOG SPEED
A3D1            I34:    DS   2       ;X-AXIS BACKLASH
A3D3            I35:    DS   4       ;X-AXIS POSITIVE POSITION LIMIT
A3D7            I36:    DS   4       ;X-AXIS NEGATIVE POSITION LIMIT
A3DB            I37:    DS   4       ;X-AXIS POSITION RANGE
A3DF            I38:    DS   9       ;X-AXIS POSITION FORMAT
A3E8            I51:    DS   1       ;Y-AXIS JOG SPEED
A3E9            I54:    DS   2       ;Y-AXIS BACKLASH
A3EB            I55:    DS   4       ;Y-AXIS POSITIVE POSITION LIMIT
A3EF            I56:    DS   4       ;Y-AXIS NEGATIVE POSITION LIMIT
A3F3            I57:    DS   4       ;Y-AXIS POSITION RANGE
A3F7            I58:    DS   9       ;Y-AXIS POSITION FORMAT

A400            I64:    DS   2       ;X ENCODER/RESOLVER BASE ADD.
A402            I65:    DS   2       ;Y ENCODER/RESOLVER BASE ADD.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A404            I66:    DS   1       ;SPEED OF SECOND X RESOLVER

A405            I67:    DS   1       ;SPEED OF SECOND Y RESOLVER
A406            I68:    DS   2       ;SECOND X RESOLVER BASE ADDRESS
A408            I69:    DS   2       ;SECOND Y RESOLVER BASE ADDRESS
A40A            I70:    DS   4       ;X ENCODER/RESOLVER BIAS
A40E            I71:    DS   4       ;Y ENCODER/RESOLVER BIAS
A412            I72:    DS   1       ;X ENCODER/RESOLVER SIZE
A413            I73:    DS   1       ;Y ENCODER/RESOLVER SIZE

        A400    PSTART  EQU  0A400H
A414            XSTART: DS   1       ;PROGRAM BUFFER START ADDRESS

        C000    P8KEND  EQU  0C000H
        C000    EARTST  EQU  0C000H
        F800    P23END  EQU  0F800H
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC PROGRAMMING EXAMPLES

TABLE OF CONTENTS

		PAGE
1.	BACK AND FORTH X AXIS ONLY (WITH WAIT AT EACH END)	.224
2.	BACK AND FORTH Y AXIS	.224
3.	USING SMCC'S X AND Y AXIS	.224
4.	BACK AND FORTH, X AND Y AXIS, 45 DEGREES	.226
5.	SQUARE WITH ROUNDED CORNERS	.227
6.	SQUARE WITH SHARP CORNERS	.227
	FIGURE - TWO AXIS SYNCHRONIZED MOTION	.228
7.	SINUSOIDAL ACCELERATION	.229
8.	SQUARE WITH CIRCULAR CORNERS, SMCC	.230
9.	ELLIPSE, HORIZONTAL	.231
10.	ELLIPSE, VERTICAL	.231
11.	CIRCLE (NO STOP)	.232
12.	TILTED ELLIPSE, 45 DEGREES	.232
13.	CASE INSTRUCTION, READ ONE THUMBWHEEL	.233
	FIGURE - TYPICAL THUMBWHEEL CONNECTION TO SMCC PARALLEL PORT	.234
14.	THUMBWHEEL SWITCHES	.235
15.	CONTROLLING JOG RATE WITH PLC	.237
16.	PLC WITH MESSAGE OUTPUT/VOLTAGE CONTROLLED FEEDRATE	.237
	FIGURE - PLC PROGRAM FOR CONTROLLING i14 AND SENDING MESSAGE TO HOST	.238
17.	P VARIABLES	.241
18.	DEMO1.SMC SUBROUTINES AND GOTO	.241
	FIGURE - DEMO 1 - FLOWCHART	.242
	FIGURE - DEMO 1 - PATH	.243
19.	DEMO2: GOTO +/-	.246
20.	MULTIPLE CARD (SMCC) PROGRAMMING	.247
21.	MACHINE INPUT/OUTPUT FUNCTIONS (M FUNCTIONS)	.249
22.	"WAIT UNTIL" PROGRAM	.251
23.	LEARN COMMAND IN PROGRAMMING	.252
24.	ALTERED DESTINATION ON TRIGGER	.253
25.	CONSTANT TIME, ALTERED DESTINATION ON TRIGGER	.255
26.	DEMO 3 - "S" CURVE ACCEL AND DECEL [PARAB]	.257
	FIGURE - "S" CURVE ACCELERATION	.258
27.	CIRCULAR CUT ON THE FLY	.260
	FIGURE - CIRCULAR CUT ON THE FLY	.261
28.	INDEPENDENT X AND Y AXIS HOMING ROUTINES	.263
29.	SERPENTINE PATTERN	.265
	FIGURE - SERPENTINE	.266
30.	COUNTER USING M-FUNCTION INPUT (CORNER)	.267
31.	TIMERS (TIME1) TO CONTROL OUTPUTS	.267
	FIGURE - M FUNCTION OUTPUTS BY PLC TIMER CONTROL	.268
32.	MULTIPLE MESSAGE TRANSMISSIONS (TIME3)	.269
33.	GLUE DISPENSER PROGRAM	.270

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PAGE

34. THUMBWHEEL TEST (TESTHUMB.SMC). 272
    FIGURE - TYPICAL USE OF THUMBWHEELS 276
35. STRAIGHT MOVE USING THUMBWHEELS (XYMOVE) USE OF
    LOWER-CASE "H" AND "L" IN CAS COMMAND 277
36. TYPICAL USE OF THUMBWHEELS (TYPTHUMB1) USE OF LOWER-CASE
    "H" AND "L" IN CAS COMMAND. 281
37. PROCESSOR ADDRESS DEFINITION (BITS AND BYTES) 286
38. CONDITIONAL EXECUTION 286
39. USING M-FUNCTION INPUTS AS CONTROL PANEL INPUTS 287
40. JOYSTICK CONTROL. 288
41. HANDWHEEL PROFILE GENERATOR 289
    FIGURE - HANDWHEEL PROFILE GENERATOR. 289A
42. DETECT MARK AND CUT 290
    FIGURE - DETECT MARK AND CUT. 290A
43. SMCC AS VOLTMETER 293
44. MULTIPLE SMCC SEQUENCING METHOD 294
45. BURR-BROWN 2500, MDI MODE 296
    FIGURE - BURR-BROWN TM2500 TERMINAL CONNECTIONS 297
46. BURR-BROWN 2500 298
47. GENERAL S-CURVE PROGRAM FOR 2-AXIS MOVES. 299
48. GENERAL PROCEDURE TO READ VALUES FROM THUMBWHEEL BOARD(S)302
49. GLUE DISPENSING, THREE AXIS 304
    FIGURE - GLUE DISPENSING. 306

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC PROGRAMMING EXAMPLES

****

EXAMPLE #1

BACK AND FORTH X AXIS ONLY (WITH WAIT AT EACH END):

```
A0                      ;Address card A0
z                       ;Clears program buffer, leaves subroutines,
                        ;if any.  This command is not stored in the
                        ;buffer.
FOR 0                   ;Start infinite loop
F1024                   ;100% feedrate, as specified by i06
X100000                 ;Move to 100000 encoder counts
DWE 1024                ;Wait 1 second
X0                      ;Move back to 0
DWE 1024                ;Wait 1 second
END                     ;Closes all loops and ends program
                        ;Program is started by "R" Run Command or
                        ;"RUN" control panel switch and will run
                        ;forever.  To stop, send "S" Step, "Q"
                        ;Quit, "q" Abort, or  "H"  Feed Hold
                        ;Commands.
```

NOTE: All comments are preceded by ";". The SMCC will *NOT ACCEPT* a program written with comments so these must be stripped off before the program is sent to SMCC. SMCQ V2.7 and SMCC V3.X IBM PC programs for the SMCC automatically strip the comments off prior to transmission of a program written by the editor.

EXAMPLE #2

BACK AND FORTH Y AXIS:

Note that on any given SMCC, selecting X as opposed to Y axis simply consists of addressing the desired SMCC (A0 to Af) and then commanding an X or a Y move. The previous example shows an X axis move; to change that to a Y axis move, all that is required is to change X100000 to Y100000 and X0 to Y0.

EXAMPLE #3

USING SMCC's X AND Y AXIS:

Even though the SMCC is a two axis controller, some restrictions apply to the use of the two axes because they are not independent of each other and must be used in conjunction with each other. For example, it is *NOT* possible to tell the X axis to make a move and then later, randomly, tell the Y axis to make a move. In

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

order to move the X and Y axes non-simultaneously, programs must be written in such a way as to position or time synchronize the two moves. For example, assume we want to move X axis to 10000 first and then, without stopping, move the X and Y axis to 20000, finally, return X and Y to zero.

The program would look as follows:

```
A0z                     ;Address card and clear buffer
F1024                   ;Set feedrate
STA                     ;Start executing following blocks in
                        ;automatic "RUN" mode, without
                        ;stopping between blocks, until an
                        ;"STO" is executed (an "S"  Step
                        ;Command or switch input is received).
                        ;If the "R" Run command is used to
                        ;start this program, the STA is not
                        ;needed.

X10000                  ;Move X to 10000, Y stays at 0
X20000 Y20000           ;Move X and Y to 20000, Y starts
                        ;moving when X reaches the point
                        ;at which deceleration  would
                        ;begin for a 10000 count move.
                        ;Refer to "Synchronized Two Axis
                        ;Motion" diagram for detailed
                        ;explanation.
                        ;A slightly rounded corner will be
                        ;made since Y is accelerating
                        ;while X is moving at full speed.
                        ;It is important to note that when
                        ;X is moving alone the velocity is
                        ;i06, since F1024 is 100% of i06.
                        ;However, during the simultaneous
                        ;X and Y move, it is the diagonal
                        ;path which is at that velocity,
                        ;and the continued X move takes
                        ;place at a slower velocity.  A
                        ;general formula for velocity
                        ;calculation is
                        ;Fx or y=F([X or Y]/[X^2+Y^2] 1/2)
                        ;where F is feedrate  and X and Y
                        ;are the respective move distances.

STO                     ;Forces a stop and cancels the
                        ;STA command

X0Y0                    ;Returns X and Y axis to zero
                        ;Note that an "S" Step  or "R" Run
                        ;is required to start this last
                        ;move since STO forces a stop.
                        ;Both X and Y are now moving to
                        ;position 20000 to position 0, at a
                        ;velocity of:  F1024/1.414.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The same type of move, but with variable velocity, can be done using the T (time) Command instead of feedrate, as follows:

```
A0z
T2048                   ;Set time for all succeeding moves to
                        ;two seconds.
STA                     ;Start a continuous run
X10000                  ;X moves from 0 to 10000, then 20000,
                        ;Its velocity does not change
X20000 Y20000           ;Y axis will begin to move at
                        ;time T and will move
                        ;twice as fast as X.
STO                     ;Stops program execution
X0Y0                    ;Returns X and Y axis to zero at twice
                        ;speed
END
```

EXAMPLE #4

BACK AND FORTH, X AND Y AXIS, 45 DEGREES:

```
z 50                    ;Clears program buffer and all subroutines,
                        ;reserves 50 lines for program.
STA                     ;Start continuous motion
FOR150                  ;Start a loop for 150 cycles
F1024                   ;Set feedrate to i06 value
X100000  Y100000        ;Move X and Y to 100000 at specified
                        ;feedrate
DWE 1024                ;Wait 1 second
X0Y0                    ;Move X and Y to zero
DWE 1024                ;Wait 1 second
END                     ;End of program and end of loop.  Program is
                        ;started  by sending "S" Step  Command and
                        ;will  run  back and forth 150 times  before
                        ;stopping.
```

NOTE: The feedrate is that which applies to the diagonal path generated, the SMCC calculates the needed rates for X and Y to satisfy that requirement. In this case of a 45 degree angular move, Fx and Fy are both equal to F1024/1.414.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #5

SQUARE WITH ROUNDED CORNERS (ACCEL TIME i08 CONTROLS ROUNDING OF THE CORNERS). (START IN THE MIDDLE OF SQUARE'S SIDE.):

```
z                       ;Clear program buffer, leave subroutines
Z                       ;Set X and Y position to zero
F1024                   ;100% feedrate
X50000                  ;Move to first corner
Y100000                 ;Move to second corner, starts while X is
                        ;decelerating
X-50000                 ;Move to third corner, starts while Y is
                        ;decelerating
Y0                      ;Move to fourth corner, starts while X is
                        ;decelerating
X0                      ;Back to start position, starts while Y is
                        ;decelerating
END                     ;End of program
```

The rounded corners are obtained because the SMCC's move time for a block of data does not include deceleration time. Therefore, as one axis begins its deceleration, the other begins its acceleration; if successive X and Y moves are made, motion will overlap during decel of one axis and accel of the other, causing a grounded corner. Even though the round corner is not exactly circular, it is accurately repeatable.

EXAMPLE #6

SQUARE WITH SHARP CORNERS (START AT CORNER):

```
z50                     ;Purge buffer and all subroutines and
                        ;reserve 50 lines of program
F1024                   ;100% feedrate
X100000                 ;Move to first corner
DWE 0                   ;Force a stop
Y100000                 ;Move to second corner
DWE 0                   ;Force a stop
X0                      ;Move to third corner
DWE 0                   ;Force a stop
Y0                      ;Move to last corner
END                     ;End of program
```

In this case, since a delay of 0 is invoked, each axis must stop before the next one begins, thus avoiding rounded corners.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC PROGRAM

```
A0
* T1024/ONE SECOND MOVE
  X5000   Y5000
  X10000  Y5000
  X15000  Y15000
  X20000  Y20000
END
```

MINI-T OR SMCC

TWO AXIS SYNCHRONIZED MOTION

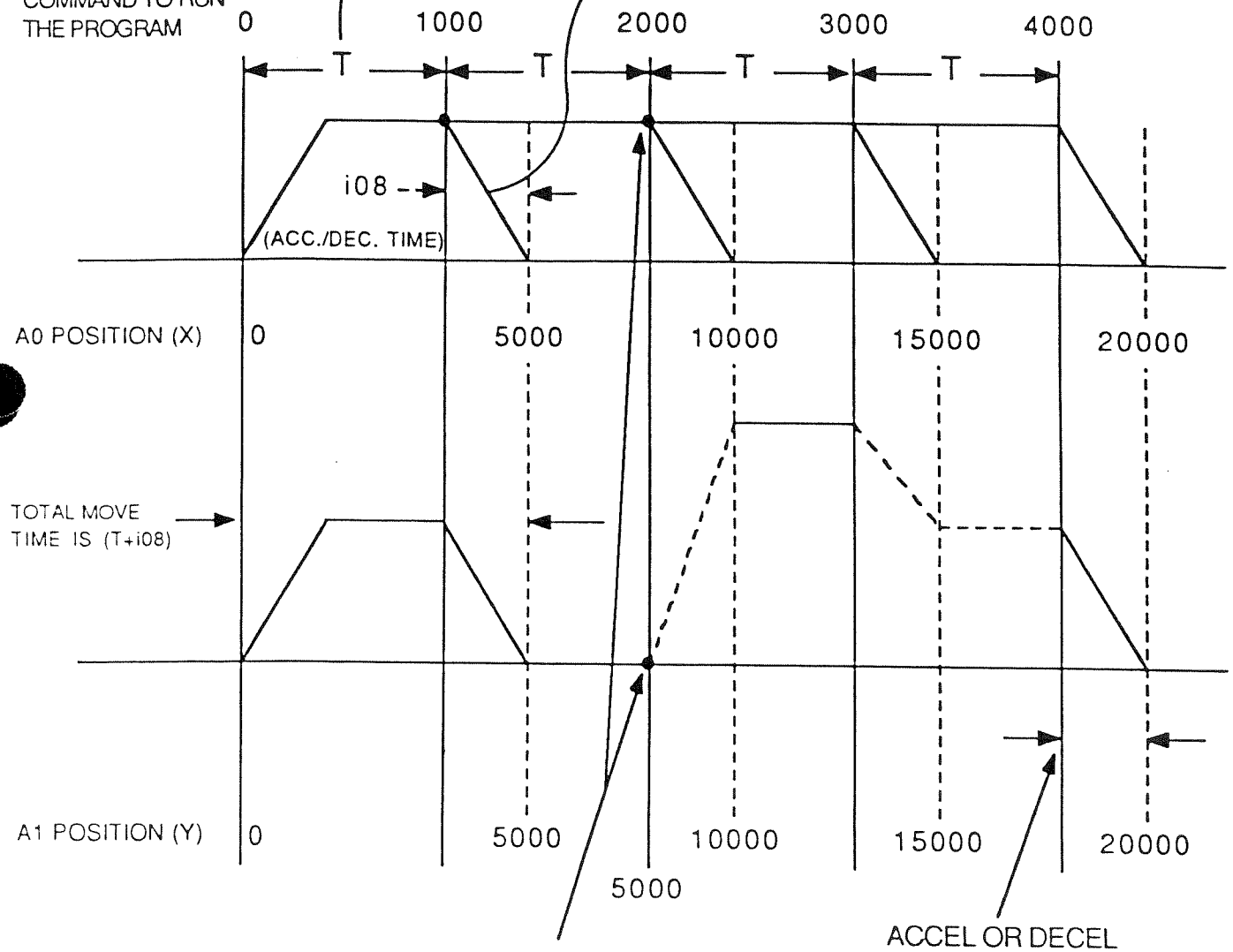
MCC PROGRAM

```
A0 (X)   A1 (Y)
T1000    T1000
X5000    X5000
X10000   X5000
X15000   X15000
X20000   X20000
M30      M30
```

MOVE TIME "T" IS FROM START OF MOTION TO BEGINING OF DECEL

PATH TO BE FOLLOWED IF A0 WAS TO STOP.

ISSUE "R" RUN COMMAND TO RUN THE PROGRAM



A1 (Y) MOTION STARTS AT BEGINING OF DECELERATION OF A0 (X).

ACCEL OR DECEL DISTANCE IS:

i06 X (F/1024)/[(i08/1024) X2] AND IS INDEPENDENT OF "%" OR "POT" VALUE.

* USE 1024 FOR SMCC, 1000 FOR MINI-T

(MINI-T)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #7

SINUSOIDAL ACCELERATION (BACK AND FORTH X AXIS MOVE WITH WAIT AT EACH END):

```
A0z                     ;Card address is A0,  clear program  buffer
                        ;only
FOR 0                   ;Start a loop forever
F1024                   ;Set feedrate to i06 value
CIR 0                   ;Set circular mode, CW
R = 10000               ;Set accel distance to 10000 lines (Radius)
X10000 WY 10000         ;Circular accel, note that Y axis is doing
                        ;a phantom move while X is moving
LIN                     ;Set linear mode
X90000                  ;Linear move to 90000
CIR 0                   ;Set circular CW mode
X100000 WY -10000       ;Circular decel to position 100000
DWE 1024                ;Wait 1 second
X90000 WY -10000        ;Circular accel
LIN                     ;Set linear mode
X10000                  ;Linear move
CIR 0                   ;Set circular CW mode
X0 WY 10000             ;Circular decel to zero position
DWE 1024                ;Wait 1 second
END                     ;End of program and end of loop
                        ;Program will repeat forever if Run Command
                        ;is issued.
```

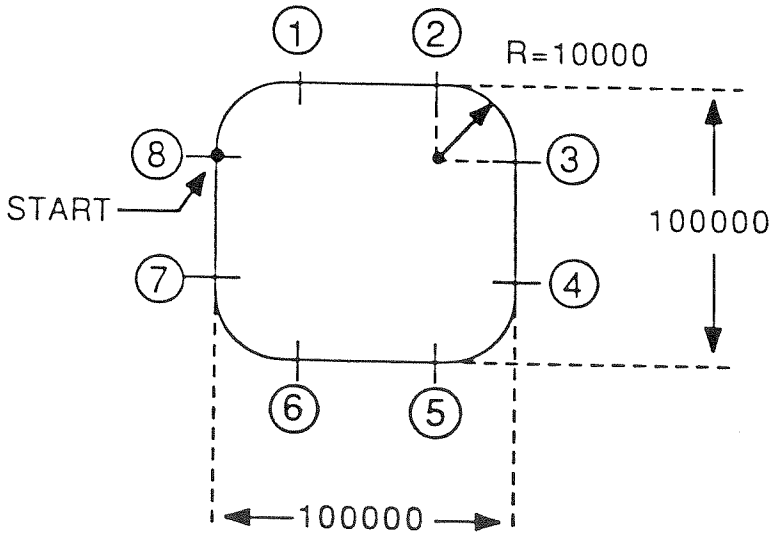
The Y axis wait moves are for making the SMCC think it's doing a circle while in reality the X axis only is moving. X does a sine move while Y does a fictitious incremental cosine move.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #8

SQUARE WITH CIRCULAR CORNERS, SMCC

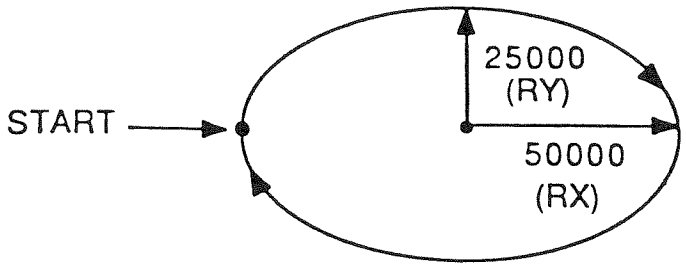
```
     Z                    /CLEARS PROGRAM BUFFER
     F1024                /SET FEEDRATE TO VALUE OF i06
     CIR0                 /CW CIRCLE, SET CIRCULAR MODE
     R=10000              /RADIUS OF 10000
     C=0                  /OFFSET=0
1)   U10000V10000         /INCREMENTAL CIRCULAR MOVE TO POINT 1
     LIN                  /SET LINEAR MODE
2)   U80000               /MOVE X TO 80000, INCREMENTAL, TO POINT 2.
     CIR0                 /SET CIRCULAR CW MODE
3)   U10000V-10000        /INCREMENTAL CIRCULAR MOVE TO POINT 3.
     LIN                  /LINEAR MODE
4)   V-80000              /MOVE Y TO-80000,INCREMENTAL TO POINT 4
     CIR0                 /CIRCULAR MODE
5)   U-10000V-10000       /MOVE TO POINT 5
     LIN                  /LINEAR MODE
6)   U-80000              /MOVE TO POINT 6
     CIR0                 /CIRCULAR MODE
7)   U-10000V10000        /MOVE TO POINT 7
     LIN                  /LINEAR MODE
8)   V80000               /MOVE TO POINT 8
     END                  /END OF PROGRAM
```



(SQUARE)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #9



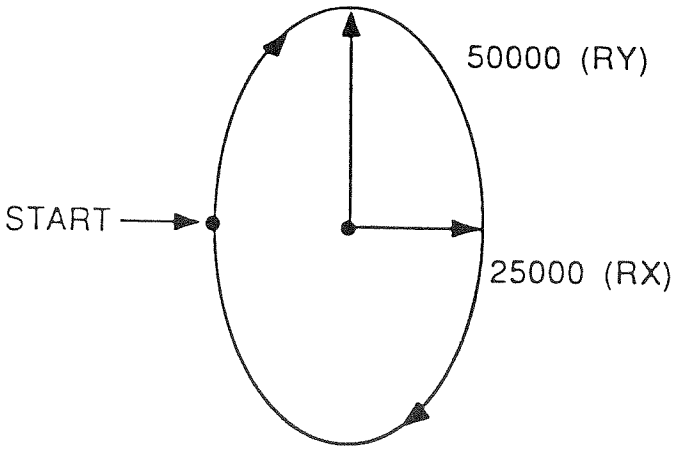
ELLIPSE, HORIZONTAL

```
F1024          /SET FEEDRATE TO i06 VALUE
CIR0           /SET CIRCULAR CW MODE.
RX=50000       /X  RADIUS=50000
RY=25000       /Y  RADIUS=25000
X100000        /MOVE TO 180 DEGREE POINT, Y NOT SPECIFIED (ZERO MOVE)
X 0            /MOVE TO ZERO
END            /END OF PROGRAM
```

EXAMPLE #10

ELLIPSE, VERTICAL

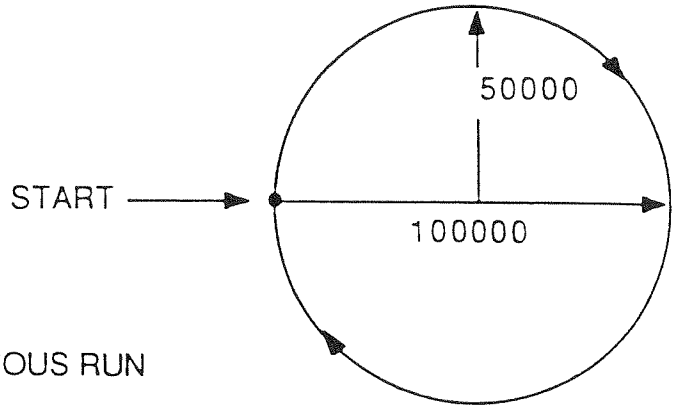
```
F1024          /SET FEEDRATE TO i06 VALUE
CIR0           /SET CIRCULAR CW MODE.
RX=25000       /X  RADIUS=25000
RY=50000       /Y  RADIUS=50000
X 50000        /MOVE TO 180 DEGREE POINT, Y NOT SPECIFIED (ZERO MOVE)
X 0            /MOVE TO ZERO
END            /END OF PROGRAM
```



(ELIPSE)

Artisan Scientific - Quality Instrumentation, Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #11




```
CIRCLE              (NO STOP)

STA                 /START A CONTINUOUS RUN
F1024               /100% FEED
CIR0                /CLOCKWIZE CIRCLE, 180 DEGREES
R=50000             /RADIUS = 50000 LINES
C=0                 /OFFSET = 0 (CIRCLE)
X 100000            /MOVE ONE DIAMETER
X 0                 /MOVE BACK ONE DIAMETER
END                 /END OF PROGRAM
```

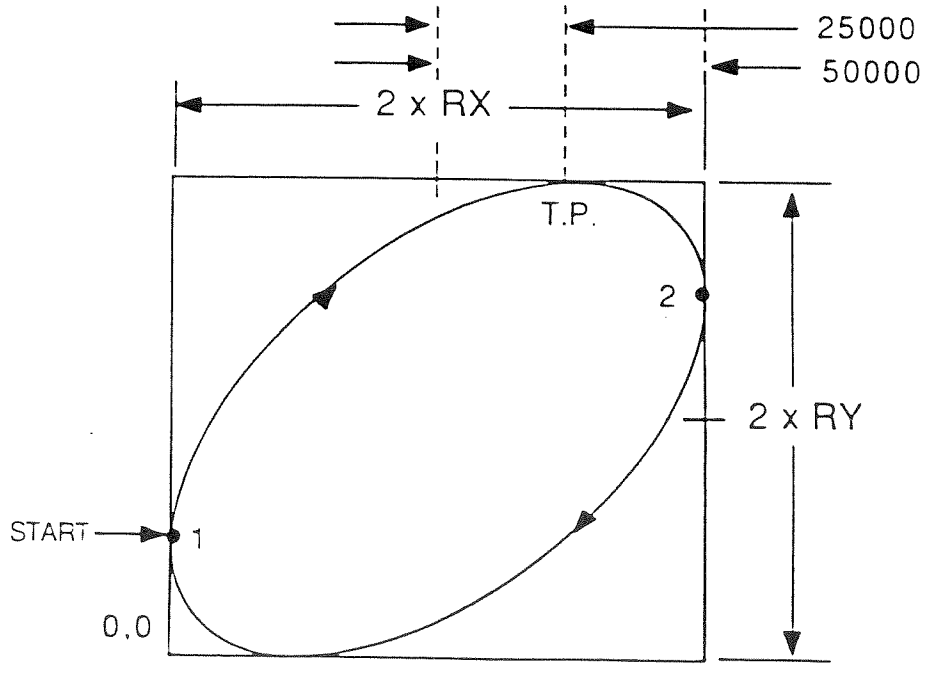
EXAMPLE #12

TILTED ELLIPSE, 45 DEGREES

$$C = \frac{2.23 \times \text{CIRCULAR OFFSET}}{RX}$$

CIRCULAR OFFSET IS THE POINT AT WHICH THE ELIPSE TOUCHES THE SQUARE. (T.P.)

```
CIR0                /MAKE CIRCLE CW.
R = 50000           /SET RADIUS TO 50000
C = 4194304
U 100000 V50000     /MOVE FROM POINT 1
                    /TO POINT 2
U-100000 V-50000    /MOVE
                    /FROM POINT 2 TO 1
END
```



(CIRCLE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #13

CASE INSTRUCTION, READ ONE THUMBWHEEL:

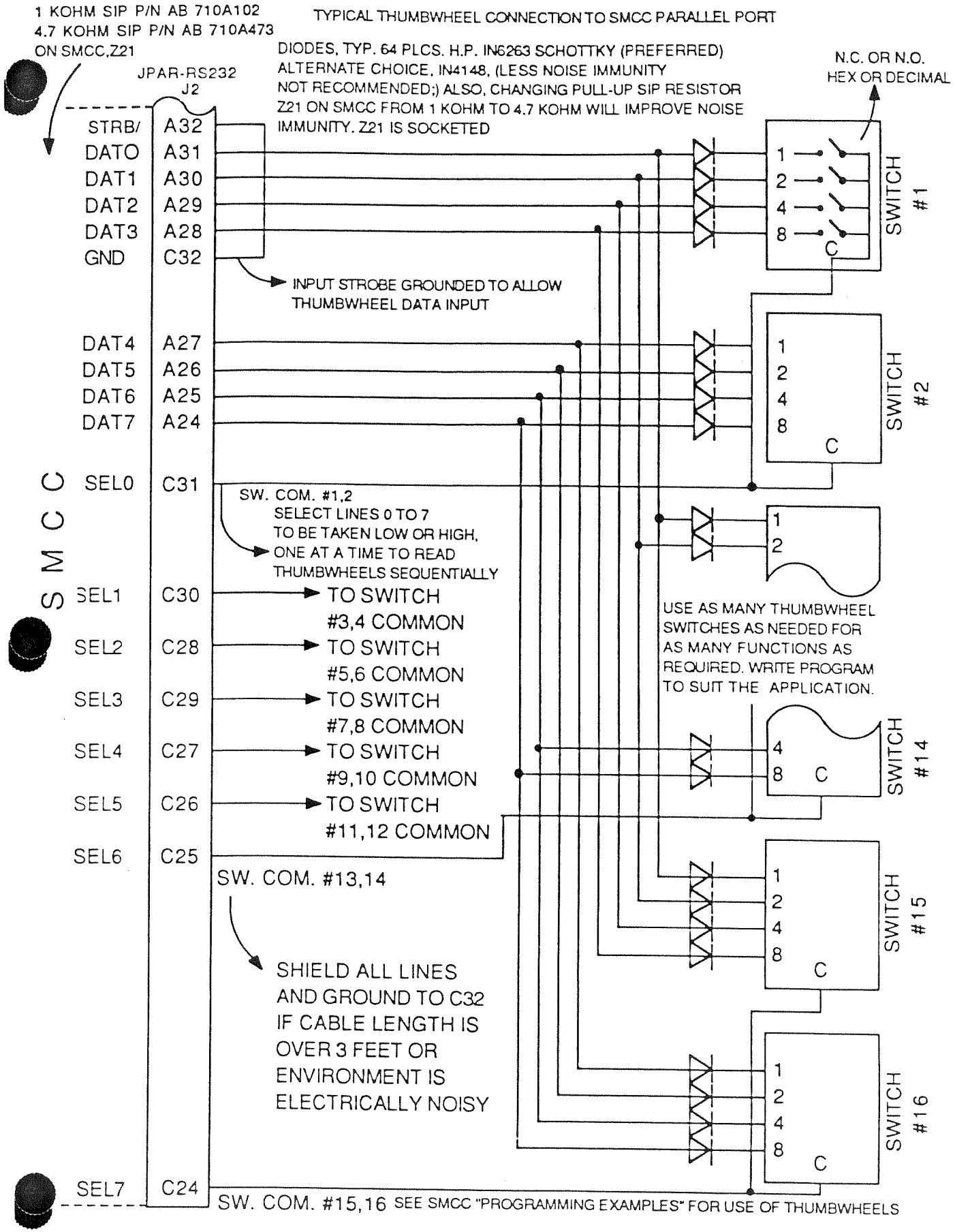
Suppose a BCD thumbwheel switch is connected to the parallel port whose address is ffb8. We want to execute different programs depending on the thumbwheel switch setting. The thumbwheel is connected to data bits 0 to 3 (Lower Nibble) and the strobe is grounded. The program can be written as:

```
SET C20 = ffb8          ;Define C20 as parallel port address ffb8

CASE  20,  10L          ;The Case Command specifies looking at
                        ;address C20 and permits a jump to a table
                        ;of 10 instructions following the Case
                        ;Command by reading the Low Nibble
                        ;(L:  Lower 4 bit) of C20 and jumping
                        ;to one of the following 10 instructions
                        ;depending on the value read from the
                        ;thumbwheel.  For example, if the thumbwheel
                        ;is set to a value of 5 (bit 0 = 1 and bit
                        ;3 = 1), then the Case instruction will
                        ;execute the 6th step or line in the table.
                        ;in this case, the GOSUB05 instruction and
                        ;will then continue with the first
                        ;instruction after the end of the table.
                        ;TABLE START
GOSUB 00                ;Execute Subroutine 00 and return to SET01
GOSUB 01                ;Execute Subroutine 01 and return to SET01
GOSUB 02                ;Execute Subroutine 02 and return to SET01
GOSUB 03                ;Execute Subroutine 03 and return to SET01
GOSUB 04                ;Execute Subroutine 04 and return to SET01
GOSUB 05                ;Execute Subroutine 05 and return to SET01
GOSUB 06                ;Execute Subroutine 06 and return to SET01
GOSUB 07                ;Execute Subroutine 07 and return to SET01
 F1024                  ;Define feedrate and do SET 01 Instruction
GOTO +10                ;Skip 10 Instructions and then proceed
                        ;The skipped instructions start with SET 01,
                        ;the 1st instruction following the "Case"
                        ;table.
                        ;TABLE END.  Note that the table may contain
                        ;any instruction (except CASE, RETURN, or
                        ;END), but only one instruction
                        ;per line may be used.
SET  01                 ;After the execution of subroutines or any
                        ;of the instructions in the table, the
   :                    ;program returns to the instruction
   :                    ;following the "Case" Table and then
                        ;continues.
```

If the thumbwheel data is out of the table's range, SMCC executes the next instruction following the table, in this case, SET 01. Please refer to "Typical Thumbwheel Connection" diagram for hardware description. Note that since only one thumbwheel is used, no diodes are needed.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #14

THUMBWHEEL SWITCHES:

This program allows two thumbwheel switches to be used for determining the size of a move. The size of the move is to be the thumbwheel value x 1000. Thumbwheels are connected to the parallel I/O port. Units at bits 0 to 3 and tens at bits 4 to 7. Thumbwheels must be "open" for a one and the wiper must be grounded. The SMCC's parallel port input strobe may be permanently grounded since only two thumbwheels are used, or may be grounded momentarily to enter new data.

```
z100                    ;Clear buffer and all subroutines, reserve
                        ;100 lines for program buffer.
Z                       ;Set current position to zero for both X and
                        ;Y axis.
SET C10 = ffb8          ;Parallel port address.  Define C10 to be
                        ;address ffb8.
SET P2  = 1000          ;Sets variable parameter P2 to 1000
F1024                   ;Set feedrate to value of i06
FOR 0                   ;Start a loop and do forever
GOS 00                  ;Go to subroutine defined by label 00 (L00)
X (P1*P2)               ;Make an X move which is the product of P1
                        ;and P2
DWE 1024                ;Wait for 1024 "Binary Milliseconds"
                        ;(1 second)
X0                      ;Return to position X0
DWE 1024                ;Wait 1 second
END                     ;End of program

L00                     ;Subroutine 00, reads thumbwheels and
                        ;determines value of P1
SET P1 = 0              ;Initialize value of variable parameter P1
                        ;to be zero
CAS 10, 10L             ;CASE command.  CAS 10 tells the SMCC
                        ;to read the value of the data at address
                        ;defined by C10 which is ffb8 or the parallel
SET  P1 = 0             ;port.  CAS 10, 10 tells the SMCC that the
SET  P1 = 1             ;CAS Command has a 10 step table of moves
SET  P1 = 2             ;following it.  CAS10, 10L tells the SMCC
SET  P1 = 3             ;to use the lower 4 bits (0 to 3) or units
SET  P1 = 4             ;thumbwheel, of the data at address C10 or
SET  P1 = 5             ;ffb8 in order to determine which step in the
SET  P1 = 6             ;table to execute.  The SMCC executes the
SET  P1 = 7             ;defined step and then is ready to execute
SET  P1 = 8             ;the instruction following the last step in
SET  P1 = 9             ;the table, in this example the next step is
                        ;another CASE instruction.
CAS 10, 10H             ;This CASE instruction tells the SMCC to use
SET P1 = P1 +  0        ;the same address, C10 or ffb8, but to use the
SET P1 = P1 + 10        ;higher 4 bits (4 to 7) or ten's thumbwheel
SET P1 = P1 + 20        ;to determine its action.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SET P1 = P1 + 30      ;The 1st CAS determines P1's value for the
SET P1 = P1 + 40      ;units and the second CAS adds the tens
SET P1 = P1 + 50      ;value.  The total value of P1 is then
SET P1 = P1 + 60      ;multiplied by the value of P2, (1000) and
SET P1 = P1 + 70      ;the result is used by the SMCC to make its
SET P1 = P1 + 80      ;move, by multiplying the values of P1  and
SET P1 = P1 + 90      ;P2.
RET                   ;End of subroutine return to program step
                      ;following GOS00.
```

To continue on usage of thumbwheel inputs, the above program only treats two thumbwheels which are read simultaneously and can always be enabled. In order to read three to sixteen thumbwheels, it is necessary to use the SEL0 to SEL7 lines on the parallel port in order to progressively select different banks of thumbwheels, please refer to the diagram "Typical Thumbwheel Connection to SMCC Parallel Port" which shows how sixteen thumbwheels can be read. Since in the previous example it was necessary to ground (take low) the common line of the thumbwheels in order to enable them, we will continue with the same assumption but with SEL0 controlling the first pair of thumbwheels. It is then necessary to write a binary value (written in hexadecimal format) to memory location ffb9 using the "w" Command (online) or the "mm" Command (in the program buffer) such that only SEL0 will be low and SEL1 to SEL7 will all be high (disabled). The binary values of SEL0 to SEL7 are:

SEL0 = 1	SEL1 = 2	SEL2 = 4	SEL3 = 8...	SEL6 = 64	SEL7 = 128
(01H)	(02H)	(04H)	(08H)	(40H)	(80H)

Writing a value of 255 (ffH) would set all SEL's to a true value (ALL high). A value of 254 (feH) would set SEL0 low and all others high. Therefore if the command mffb9,fe is inserted following the L00 statement in the example program, the SEL0 line only would be low thus enabling the first pair of thumbwheels. Similarly, successive pairs of thumbwheels can be selected by setting ffb9 to 253 (fdH), 251 (fbH), 247 (f7H), 239 (efH), 223 (dfH), 191 (bfH), and 127 (7fH), and writing appropriate subroutines to use the thumbwheel values for setting distance, velocity, accel/decel, dwell, loops, selecting subroutines, or any other variable.

If the thumbwheels and the system used require a high to be enabled, "mffb9,1" would turn on SEL0 only and leave SEL1 to 7 turned off. Successive values for ffb9 then are: 1, 2, 4, 8, 16 (10H), 32 (20H), 64 (30 H), 128 (40H).

The format for the "w" or "mm" Command is:

mm or w (address in hex), (value in hex). For example, mmffb9,d1 means write a value of 209 to address ffb9 which sets SEL0, 4, 6 and 7 high and SEL 1,2,3, and 5 low.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If more than sixteen thumbwheels are to be read, it becomes necessary to use either other controllable outputs from the SMCC such as M01 to M05 where the SET and RESET buffer commands can be used, add the Accessory 14 board which provides 48 individually programmable inputs or outputs, or to provide an external decoder for the SEL0 to SEL7 lines which, if all eight lines are used, can theoretically produce as many as 256 individually selectable outputs for reading 512 thumbwheel switches! The new Accessory 18, Thumbwheel Multiplexer Board, provides this external decoding.

EXAMPLE #15

CONTROLLING JOG RATE WITH PLC:

This PLC program causes the SMCC to jog at high speed in the position band outside +/-30000 counts and at very low speed when inside the +/-30000 count range.

```
IF M01 = 0                  ;PLC starts with IF statement.  Use M01
                            ;to toggle
AND AX > -30000             ;Check X position to be between -30000
                            ;and 30000
AND AX <  30000
THEN M01 = 1                ;Set M01 to a 1
AND I31 = 10                ;Change jog speed to low speed

IF M01 = 1                  ;Second PLC statement
AND AX < -30000             ;Check X position to be outside the
                            ;-30000 to 30000 band
OR M01 = 1
AND AX > 30000
THEN M01 = 0                ;Set M01 to a 0
AND I31 = 240               ;Change jog speed to high speed
END                         ;End PLC program
```

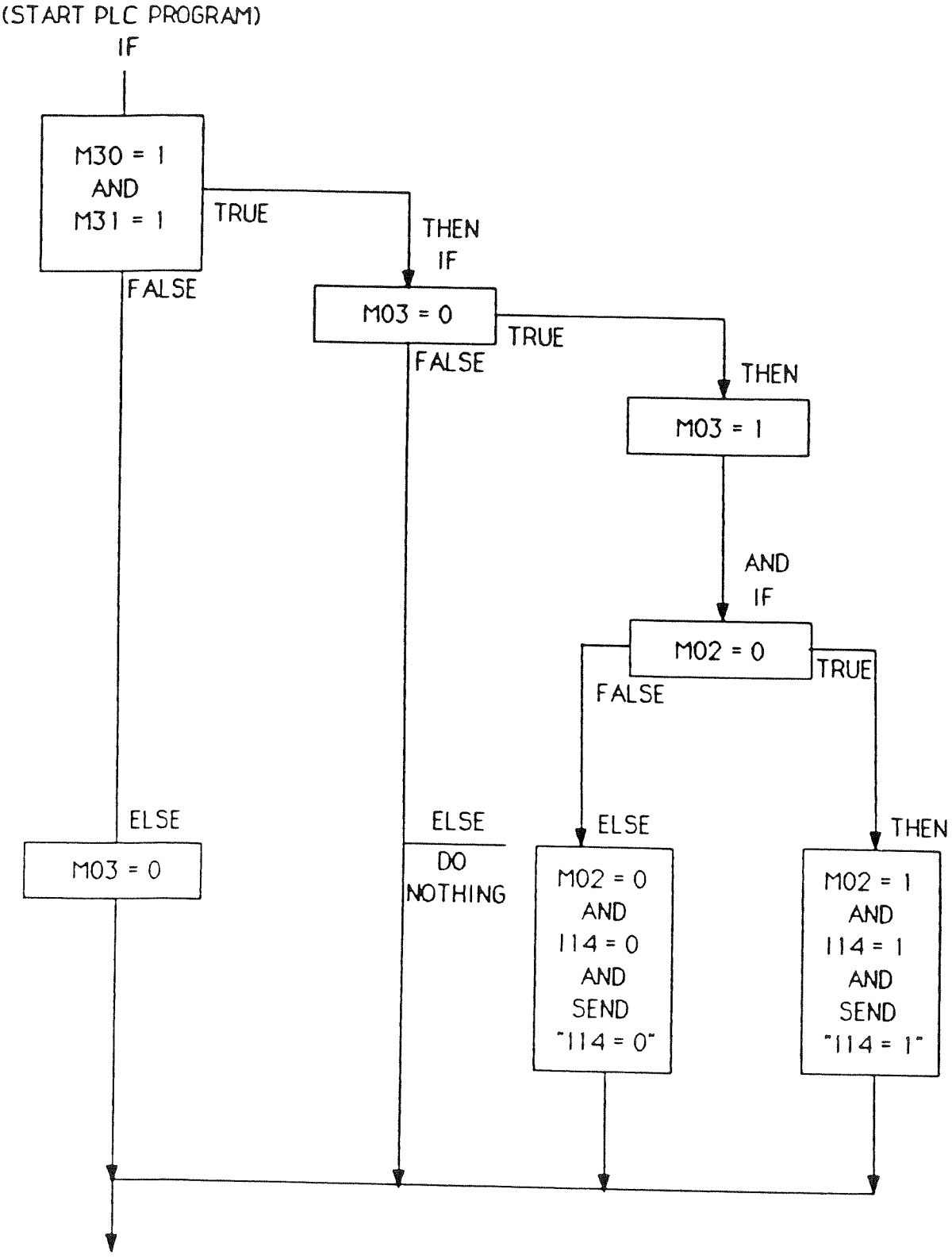
EXAMPLE #16

PLC WITH MESSAGE OUTPUT/VOLTAGE CONTROLLED FEEDRATE
(STD AND PARABOLIC SOFTWARE DATED 4/1/88 AND LATER ONLY):

The object of this PLC program is to allow an operator to selectively enable or disable the i14 parameter, used to control the source from which the % feedrate override can be issued with i14 set to zero (i140), "%" Commands can be sent via the terminal (host computer) or executed in the program buffer, "POT" Command. When i14 is set to a one (i141), the SMCC disables the "%" and "POT" Commands and accepts its feedrate from one of its analog inputs, "ANA 2" located on "J1, JMACH" pin "C1". A nominal 2K ohm pot may be used between GND and +5 volts, with the wiper attached to J1, pin C1. Shielded wires and some capacitive filtering are recommended since the feedrate is now voltage controlled and any noise on this input will cause velocity

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PLC PROGRAM FOR CONTROLLING
114 AND SENDING MESSAGE TO HOST



(PLC 1)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

fluctuations in the SMCC. 0 volts in corresponds to 0 feedrate and 5 volts in corresponds to 100% feedrate.

In order to select the source of feedrate control, we choose to use the "=" switch (also known as "prejog" switch) on the control panel to alternately change the state of i14 from a zero to a one, each time it is activated. The program proceeds as follows (please refer to accompanying figure to follow the PLC program):

```
A0z                     ;Address card, clear program buffer,
                        ;leave subroutines and PLC program
                        ;undisturbed
SET C00 = ff3a          ;Define C00 to be address ff3a
                        ;Refer to I/O map for control panel
                        ;and other input memory mapping
SET M30 = C00.2         ;Define M30 to be bit two of address
                        ;ff3a
SET M31 = C00.3         ;Define M31 to be bit three of
                        ;address ff3a. Note that when "="
                        ;switch is active, bits 2 and 3 of
                        ;ff3a become true.
F1024                   ;Set feedrate to 100%
HOM                     ;Go home
etc.                    ;Programmed moves here
 !                                !
 !                                !
 !                                !
END                     ;End of program
A0                      ;Address card
PUR                     ;Purge existing PLC program. If
                        ;PURGE is not used, the new PLC
                        ;program will append itself to
                        ;any existing PLC program in the
                        ;buffer.
IF  M30=1               ;The "IF" statement starts the
AND M31=1               ;PLC program entry. If M30 and
                        ;M31 are both true, the "="
                        ;switch is on.

THEN IF M03 = 0         ;If M03 is zero, set it to a one.

                        ;This is used as a "one shot"
                        ;flag which causes i14 to be
                        ;toggled only once, regardless
                        ;of how long the "=" switch is
                        ;held on. The "=" switch must
                        ;be released and then
                        ;reactivated before i14 will
                        ;toggle again. Also, note that
                        ;the "THEN IF" statement creates
                        ;second level PLC statement.

THEN M03   = 1          ;If M03 was zero, set it to a one.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
AND IF M02 = 0          ;This statement starts a third
                        ;level PLC which actually controls
                        ;the toggling of i14.
THEN M02 = 1            ;If M02 is zero, set it to a one.
AND I14  = 1            ;And also, set i14 to a one, this
                        ;enables the connected voltage at
                        ;the SMCC's analog input to control
                        ;the feedrate.
AND SEND "i14=1"        ;Also, send the ASCII message
                        ;"i14=1" to the SMCC's RS232
                        ;communication channel.  Please
                        ;note that if spaces are included
                        ;in the message, the SMCC
                        ;will delete them.

ELSE M02 = 0            ;If M02 was a one, then set it to
                        ;zero.  This insures that next
                        ;time the "=" switch is activated
                        ;i14 will be set to a one.  Also,
                        ;this ELSE closes the third PLC
                        ;loop.

AND I14 = 0             ;And also, set i14 to 0, this
                        ;disables the external feedrate
                        ;control and passes control back
                        ;to the program or the host computer.
                        ;Please note that the feedrate will
                        ;remain where it was last set by the
                        ;external voltage until a new host
                        ;or program originated feedrate
                        ;command is received.

AND SEND "i14=0"        ;Send "i14=0" message to host via
                        ;RS232.
ELSE                    ;Do nothing, closes PLC's second
                        ;loop.
ELSE M03=0              ;If "=" is not on, set M30=0.  Closes
                        ;first PLC loop.
END                     ;Ends the PLC program
```

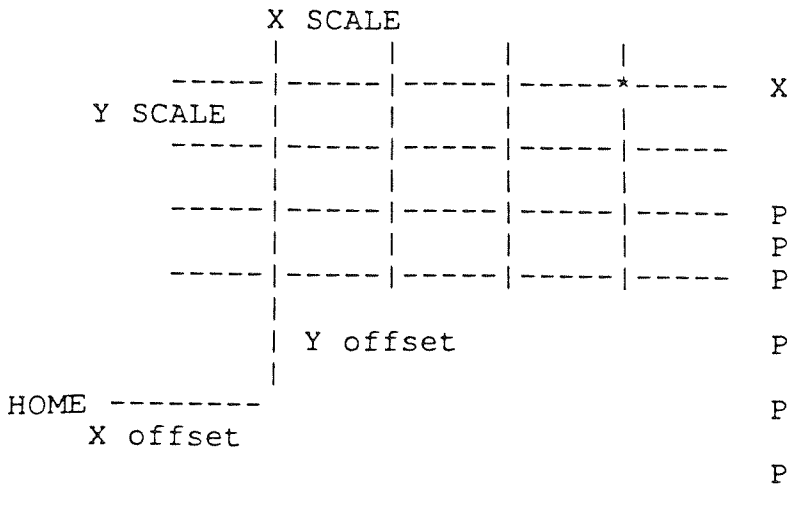
The PLC program can be written anywhere in the normal program flow, as long as it is started by an "IF" statement and ended by an "END" statement. PLC programs are stored in a separate area of the program buffer, specifically designated for this purpose.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #17

P VARIABLES:

This program shows an efficient use of P variable parameters to define an X/Y matrix.



Program:
```
F1024
X (P5 + P3 * P1) Y (P6 + P4 * P2)
END
```

To change the size and scale of the matrix simply send the pertinent parameters, the program calculates the necessary move to correctly go to the specified grid index.

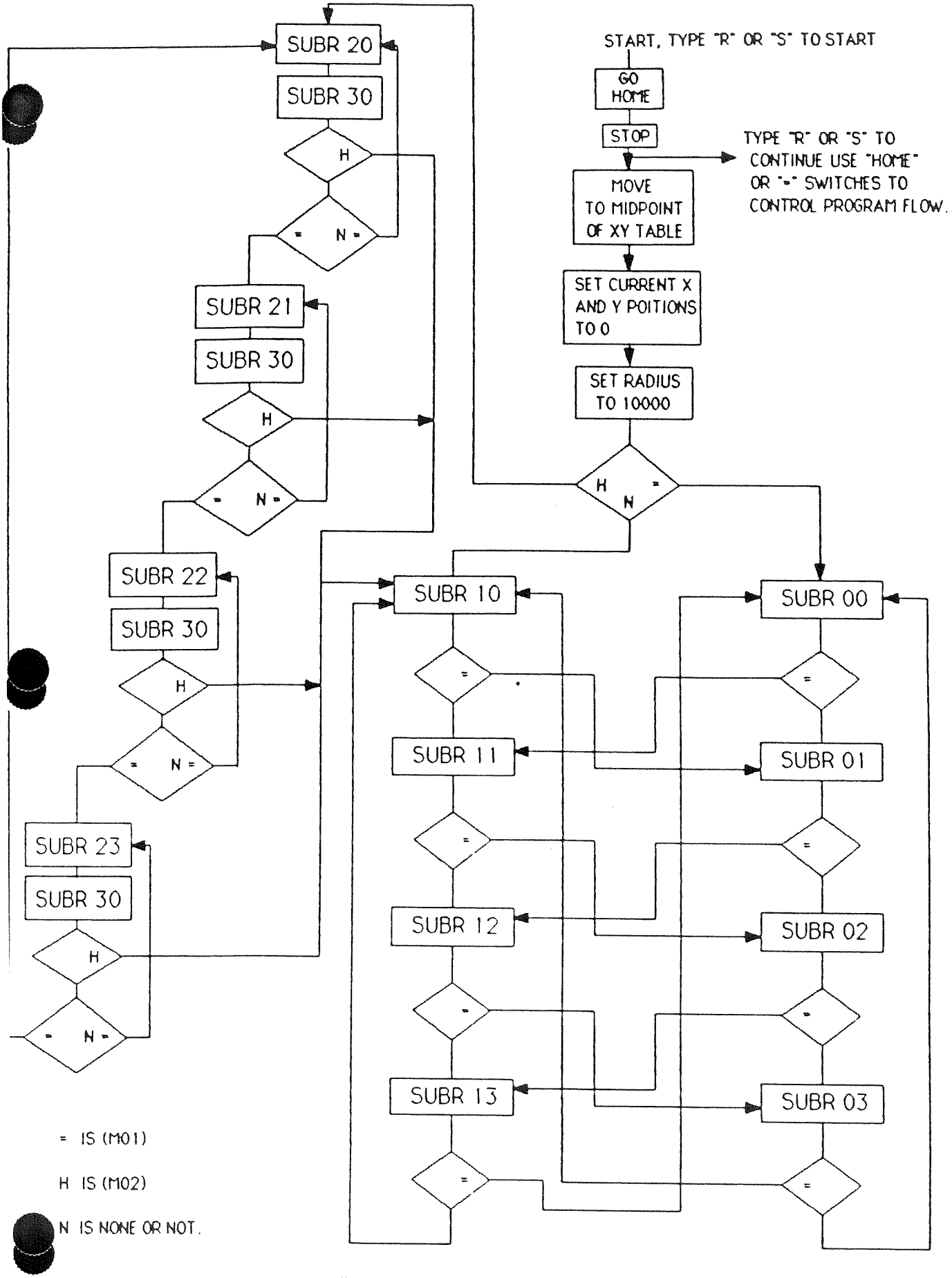
EXAMPLE #18

DEMO1.SMC SUBROUTINES AND GOTO:

This program creates a pattern as shown in the accompanying figure and uses two of the SMCC's control panel switches, the "Prejog" switch (also known as "=" switch) and the "Home" switch, to alter the program path by use of conditional GOTO statements. Note that while a program is running, the "Prejog" (or "=") and "Home" switches are inactive in terms of the SMCC's functions; they can therefore be used to alter program path without effecting normal operation. A PLC program is used to detect the presence of the "=" or "Home" switches. Note that the switches must be held long enough to be detected.

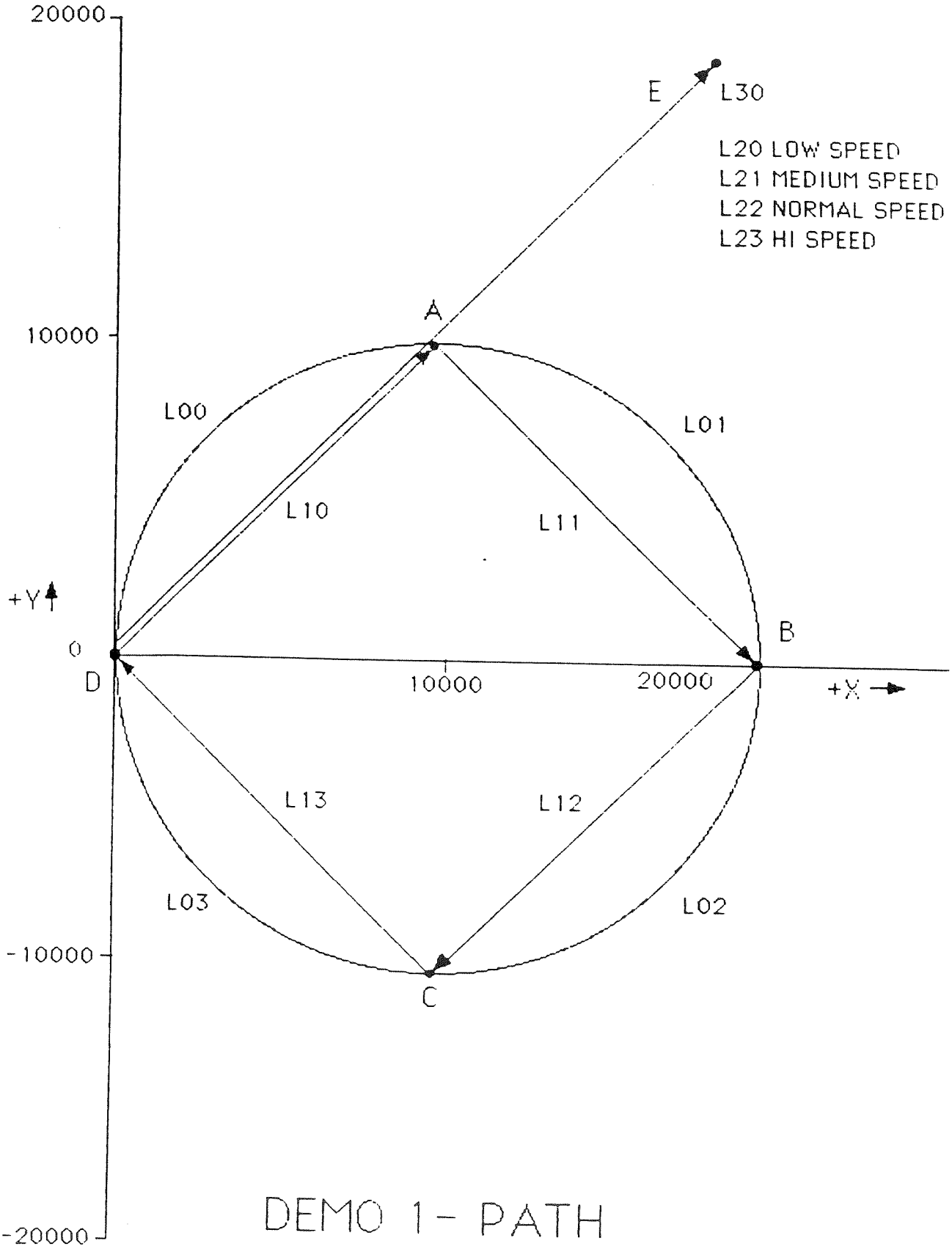
The program will trace a square pattern, with linearly interpolated moves, a circular pattern, or a longer, linear 45 degree move at four selectable speeds. Please refer to "Demo 1 - Path" and "Demo 1 - Flow Chart" for a better following of this program.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



DEMO 1 - FLOW CHART

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

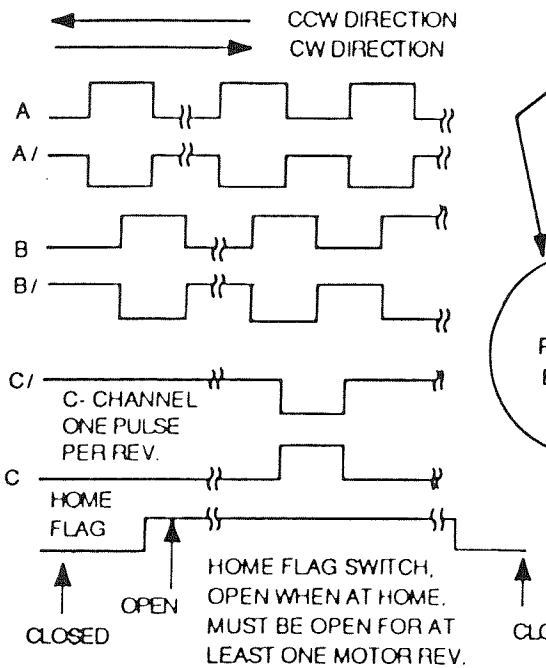


DEMO 1 - PATH

(DEMO 1 PATH)

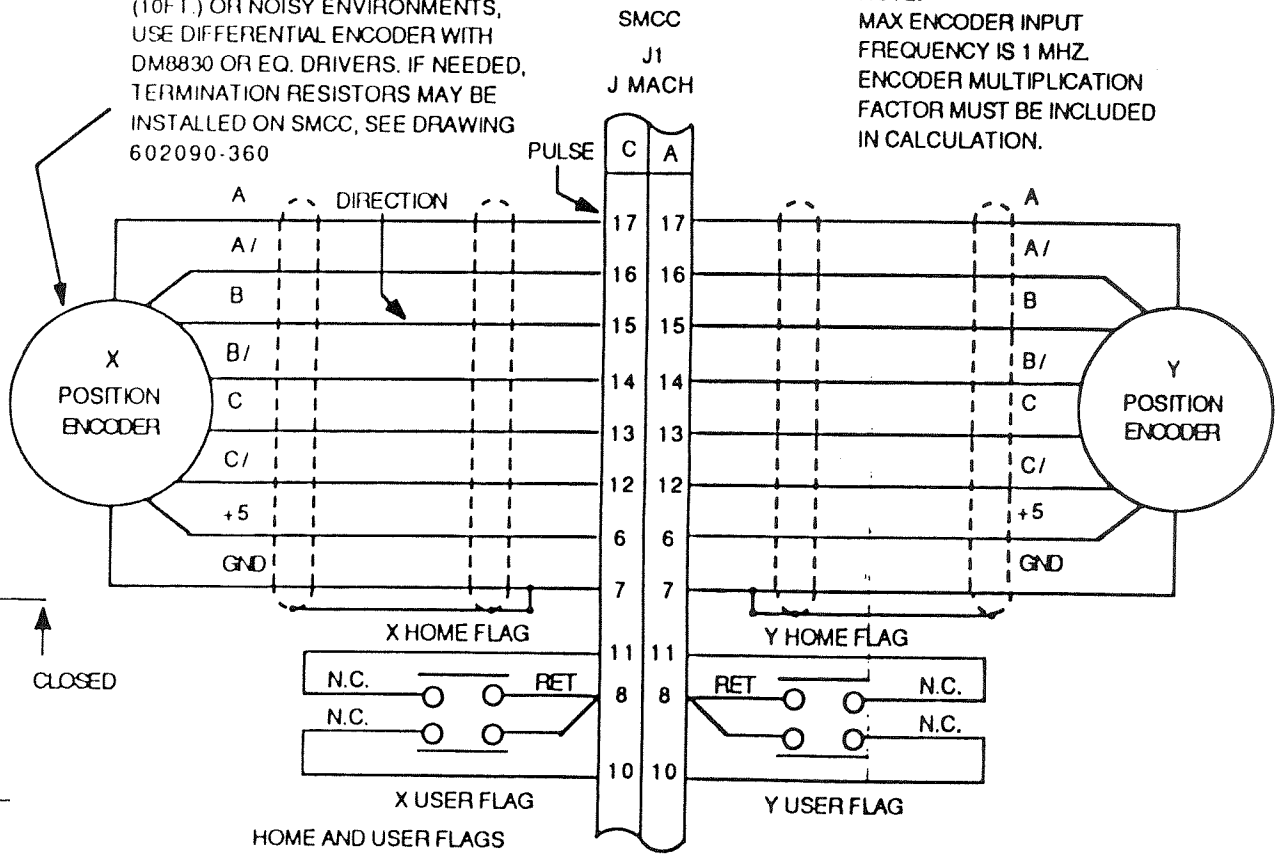
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SEE i39 FOR X,[i59] FOR Y ENCODER MULTIPLICATION FACTORS AND DIRECTION CONTROL.



FOR DISTANCES GREATER THAN 3 METERS (10FT.) OR NOISY ENVIRONMENTS, USE DIFFERENTIAL ENCODER WITH DM8830 OR EQ. DRIVERS. IF NEEDED, TERMINATION RESISTORS MAY BE INSTALLED ON SMCC, SEE DRAWING 602090-360

NOTE:
MAX ENCODER INPUT FREQUENCY IS 1 MHZ. ENCODER MULTIPLICATION FACTOR MUST BE INCLUDED IN CALCULATION.



HOME AND USER FLAGS GENERATE INTERRUPTS AND ARE USED TO CAPTURE POSITIONS ON THE FLY.

1) ENCODER LEADS "MUST BE" SHIELDED. SHIELD MUST BE INSULATED AND GROUNDED AT SMCC ONLY. HAVE SHIELD AT ENCODER END UNGROUNDED. DO NOT RUN ENCODER WIRES IN SAME CONDUIT AS A.C. POWER OR AMPL. OUTPUT WIRES.

2) JUMP E42 PIN 1 TO PIN 2 FOR DIFFERENTIAL X ENCODER INPUT
JUMP E50 PIN 1 TO PIN 2 FOR DIFFERENTIAL Y ENCODER INPUT

3) FOR SINGLE ENDED OPERATION, USE A,B, AND C INPUTS ONLY, LEAVE A/, B/, AND C/ OPEN, DO NOT TIE TO GND.
JUMP E42 PIN 2 TO PIN 3 FOR "X"
JUMP E50 PIN 2 TO PIN 3 FOR "Y"

4) SEE I24, I25 FOR X AND [I44][I45] FOR Y HOMING FUNCTION CONTROL

5) FOR ABSOLUTE POSITION ENCODER, OR RESOLVER, ACC 14 MUST BE INSTALLED.

6) REFER TO SMCC MANUAL FOR MORE DETAILS

7) WHEN ENCODER IS A PULSE AND DIRECTION DEVICE, J1C17 / J1A17 IS USED AS PULSE INPUT, AND J1C15 / J1A15 IS USED AS DIRECTION. HI= +DIRECTION:

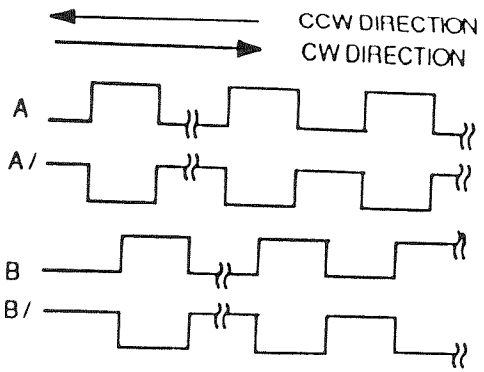
SMCC

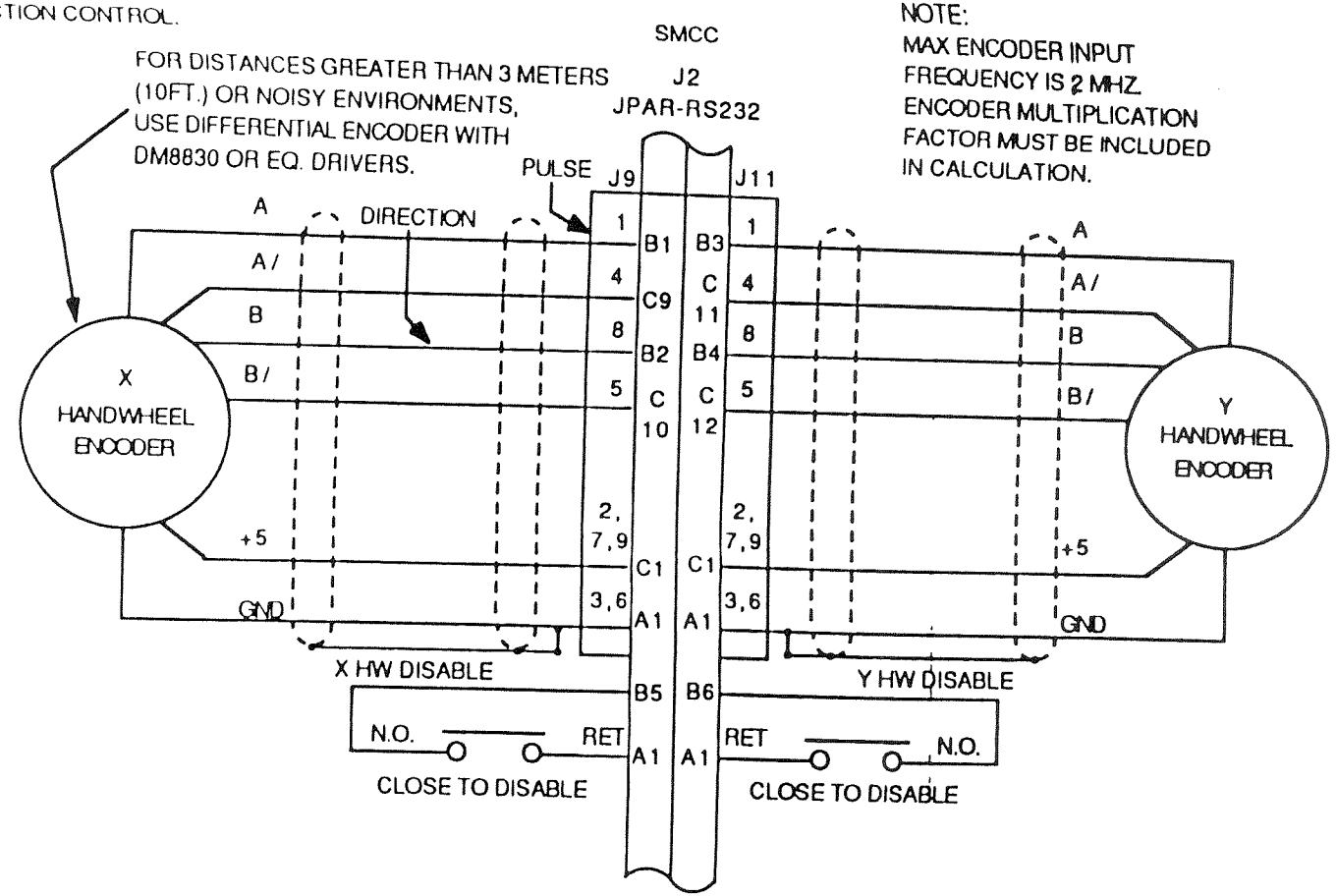
TYPICAL POSITION ENCODER CONNECTIONS DIFFERENTIAL DRIVER OR OPEN COLLECTOR OUTPUT A/B QUADRATURE

(ENCODER CONNECTIONS)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SEE i29 FOR X, [i49] FOR Y HANDWHEEL ENCODER MULTIPLICATION FACTORS AND DIRECTION CONTROL.





NOTE:
MAX ENCODER INPUT FREQUENCY IS 2 MHZ. ENCODER MULTIPLICATION FACTOR MUST BE INCLUDED IN CALCULATION.

1) ENCODER LEADS "MUST BE" SHIELDED. SHIELD MUST BE INSULATED AND GROUNDED AT SMCC ONLY. HAVE SHIELD AT ENCODER END UNGROUNDED. DO NOT RUN ENCODER WIRES IN SAME CONDUIT AS A.C. POWER OR AMPL. OUTPUT WIRES.

2) JUMP E45 PIN 1 TO PIN 2 FOR DIFFERENTIAL X HANDWHEEL ENCODER INPUT
JUMP E46 PIN 1 TO PIN 2 FOR DIFFERENTIAL Y HANDWHEEL ENCODER INPUT

3) FOR SINGLE ENDED OPERATION, USE A,B, AND C INPUTS ONLY, LEAVE A/, B/, AND C/ OPEN, DO NOT TIE TO GND.
JUMP E45 PIN 2 TO PIN 3 FOR "X"
JUMP E46 PIN 2 TO PIN 3 FOR "Y"

4) HANDWHEEL SCALE FACTOR I27, [I47] HANDWHEEL MODE CONTROL I28, [I48] AND I16

5) WHEN ENCODER IS A PULSE AND DIRECTION DEVICE, J9-11,1 AND J2-B1, B3 ARE USED AS PULSE INPUT, AND J9-11,8 AND J2-B2,B4 ARE USED AS DIRECTION. HI= +DIRECTION.

6) FOR INFORMATION ABOUT HANDWHEEL OPERATING MODES REFER TO SMCC MANUAL.

SMCC

TYPICAL HANDWHEEL ENCODER CONNECTIONS DIFFERENTIAL DRIVER OR OPEN COLLECTOR OUTPUT A/B QUADRATURE

(ENCODER CONNECTIONS)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A0z100                          ;Card address is A0, buffer is
                                ;completely cleared and 100 lines are
                                ;reserved for program space.
                                ;Subroutines will begin at line 101.
SET C00 = ff3a                  ;Selects C00 to be address ff3a
SET M30 = C00.2                 ;Sets M30 as bit 2 of ff3a
SET M31 = C00.3                 ;Sets M31 as bit 3 of ff3a
SET M32 = C00.6                 ;Sets M32 as bit 6 of ff3a
SET M33 = C00.7                 ;Sets M33 as bit 7 of ff3a
PUR                             ;Purge Command, clears all PLC
                                ;programs from buffer
IF M30=1                        ;PLC program stars with IF.  Looks
                                ;for "=" switch to
AND M31=1                       ;be on; if it is, sets M01 high.
THEN M01=1
ELSE M01=0                      ;If not, set M01 to 0.
IF M32=1                        ;Second PLC program, looks for
                                ;"Home" switch to be on, if it is
AND M33=1                       ;sets M02 high.
THEN M02=1
ELSE M02=0                      ;If not, set M02 to 0.
END                             ;Ends PLC program
F1024                           ;Set feedrate to 100%
HOME                            ;Go to home position
STO                             ;Stop at this point
LIN                             ;Linear mode
I621                            ;Turn on PLC (use I62 for Parabolic
                                ;and I60 for Standard)
X20000 Y15000                   ;Move X and Y away from home
X = 0                           ;Set X position to 0
Y = 0                           ;Set Y position to 0
R = 10000                       ;Radius of 10000
GOT20IF(M02=1)                  ;GOTO Subroutine 20 if "Home" switch is
                                ;activated
GOT00IF(M01=1)                  ;GOTO Subroutine 00 if "=" switch
                                ;is activated
GOT10                           ;GOTO Subroutine 10 if neither "Home"
                                ;or "=" switches are activated
END                             ;End of program

L10LIN                          ;Subroutine 10, linear mode
F512                            ;Half speed
DWE1024                         ;Wait 1 second
X10000Y10000                    ;Go from point D to A, linear

GOT01IF(M01=1)                  ;If "=" switch is activated, go to
                                ;Subroutine 01, if it is not continue
                                ;and go to next instruction L11.   Note
                                ;that if the "=" switch is on, the
                                ;program goes to Subroutine 01 which
                                ;alters the path from linear to
                                ;circular.
L11LIN                          ;Subroutine 11, linear mode
DWE1024                         ;Wait 1 second
X20000Y0                        ;Go to point B, linear
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
GOT02IF(M01=1)          ;If  "=" switch  is  on,  go  to
                        ;Subroutine 2, if not, continue.
L12LIN                  ;Subroutine 12, linear mode
DWE1024                 ;Wait 1 second
X10000Y-10000           ;Go to point C, linear
GOT03IF(M01=1)          ;If "=" on go to Subroutine 3,  if
                        ;not continue.
L13LIN                  ;Subroutine 13, linear mode
DWE1024
X0Y0                    ;Return to point D, linear
GOT00IF(M01=1)
GOT10                   ;Go to Subroutine 10,  repeat the above
                        ;linear cycle
L00CIRO                 ;Subroutine 00, set circular CW mode
F512
X10000Y10000            ;Go to point A,  tracing a circle with
                        ;radius of 100000 counts
GOT11IF(M01=1)          ;If   "="  switch  on, go  to
                        ;Subroutine 11 which changes
L01CIRO                 ;Subroutine 01, circular pattern, from
                        ;circular  to linear  mode.   If  not,
                        ;continue.
X20000Y0                ;Go to point B, circular
GOT12IF(M01=1)
L02CIR0                 ;Subroutine 02, circular CW mode
X10000Y-10000           ;Go to point C, circular
GOT13IF(M01=1)
L03CIRO                 ;Subroutine 03, circular CW mode
X0Y0                    ;Go to point D, circular
GOT10IF(M01=1)
GOT00                   ;Go to Subroutine 00, repeat the above
                        ;circular cycle
L20F256                 ;Subroutine 20, set quarter speed
GOS30                   ;Go to Subroutine 30 which  does  a
                        ;linear move to point E and back to D

G0T20IFN(M01=1)         ;Keep returning to Subroutine 20, as
                        ;long as "Prejog" or "=" switch has not
                        ;been energized.  If "Prejog" is turned
                        ;on, continue to next instruction.
L21F512                 ;Subroutine 21, set half speed.
GOS30                   ;Go to Subroutine 30
GOT21IFN(M01=1)
L22F1024                ;Subroutine 22,  set full speed, same
                        ;as i06
GOS30
GOT22IFN(M01=1)         ;Return to 22 if "=" is not on
L23F2048                ;Subroutine 23,  set speed to  double
                        ;value of i06
GOS30
GOT23IFN(M01=1)         ;Return to 23 if "=" is not on
GOT20

L30LIN                  ;Subroutine 30,  do a linear back and
                        ;forth move from position D to E
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
X20000Y20000
DWE1024                    ;Wait 1 second
X0Y0                       ;Return to position D
DWE1024
GOT10IF(M02=1)             ;If the "Home" switch is turned  on,
                           ;go to Subroutine 10 and start linear
                           ;square moves; if not, continue back
                           ;and forth between D and E and change
RET                        ;speeds by using "=" or "Prejog"
                           ;switch.
```

Please note that it is possible to obtain the same results with this program without the use of a PLC program by substituting:

for (M01=1) an (M30=1 AND M31=1)

and for (M02=1) an (M32=1 AND M33=1)

in all of the GOT statements. PLC was used as an example only. Also, note that the PLC program can be written at any place in the program flow since it is stored in a special buffer, separate from the normal program buffer, and that the PLC program runs continuously and is therefore able to detect presence of switch closures asynchronous to the normal program flow with a delay not greater than two to five msec.

EXAMPLE #19

DEMO2: GOTO +/-:

The following program uses the "=" switch of the control panel to select two different programs, but accomplishes this by using the GOTO + or - instead of subroutines to make the transition. Also, the "=" switch only needs momentary activation at any time in order to take effect. In this case, the motor is an AC induction type and has a 500 line per revolution encoder, multiplied by four for 2000 counts per revolution.

```
0 A0z100                   ;Address card A0 and clear buffer
                           ;Allocate 100 step buffer
SET C30 = ff3a             ;Set C30 to be memory location
                           ;ff3a which is where the "="
                           ;switch comes in.  Note that
                           ;"=" causes 2 bits to go high
                           ;bits 2 and 3.
SET M30 = C30.2            ;Set M30 to be bit 2 of ff3a
SET M31 = C30.3            ;Set M31 to be bit 3 of ff3a
PUR                        ;Purge existing PLC program if any
IFM30=1                    ;The IF signifies the start of a
                           ;PLC program
ANDM31=1                   ;If M30 and M31 ("=" switch) is on,
THENM01=1                  ;turn Set M01 to a 1. This acts as a latch.
END                        ;Ends the PLC program
t512                       ;Set accel/decel time to 1/2 second
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
HOM                              ;Go to home position
STO                              ;Stop executing program
I601                             ;Turn  on PLC program,  use  I621  for P
GOT + 10 ---------------         ;Skip 9 lines and execute line 10
RES01  <-------------    |       ;Reset M01 to zero.
I0620000 <--------   |   |       ;Set speed  to 20000 encoder cnts/sec.
t7600            |   |   |       ;Set accel/decel time to 7.421 seconds
F17136           |   |   |       ;Set speed to about 10000 RPM
X4000000         |   |   |       ;Go to position 4 million
DWE1024          |   |   |       ;Wait 1 second
XO               |   |   |       ;Return to zero position
DWE1024          |   |   |       ;Wait 1 second
GOT-07IFN(M01=1)--   |   |       ;Go back 7 lines if M01 is set to zero
RES01 <--------------|--         ;Reset M01 to zero
I0620000<--------    |           ;Set speed to 20000 encoder counts/sec.
t512             |   |           ;Set accel/decel time to 1/2 second
F100             |   |           ;Set speed to about 60 RPM
X5000            |   |           ;Go to position 5000
F900             |   |           ;Set speed to about 530 RPM
X100000          |   |           ;Go to position 100000
DWE1024          |   |           ;Wait 1 second
X5000            |   |           ;Return to position 5000
F100             |   |           ;Reduce speed to 60 RPM
XO               |   |           ;Return to position zero
DWE1024          |   |           ;Wait 1 second
GOT-21IF(M01=1)--|---           ;If M01 is one ("=" switch was seen),
                 |               ;jump back 21 lines.
GOT-12 ----------                ;If "=" not seen, jump back 12 lines.
END                              ;End of program
```

EXAMPLE #20

MULTIPLE CARD (SMCC) PROGRAMMING:

Assume two SMCCs, addressed as A0 and A1, are daisy chained via the RS232. (The parallel port may be used instead if desired.) Once the port is chosen, it is not possible to switch from one port to the other without changing the settings of the DIP switch on the SMCC card and performing either a reset or power cycle in order to have the SMCC take in the new setting. Also, one of the SMCCs must be A0 while the second can be A1 to Af. It is best to set the second to A1 and so on, for efficiency in DMA communications. Also, remember that when an SMCC is A0, it generates the sync clock, and a jumper must be on the "E2" selection. When the card is A1 to Af, it receives the sync clock and the jumper must be moved from "E2" to "E3".

It is the daisy chaining of the communication ports which automatically causes multiple cards to become time synchronized for purposes of linear, circular, and elliptical interpolation between cards; the user does not have to take any other measures to insure proper sync and interpolation. Interpolation can be done between any two or more cards (16 cards maximum) including all simultaneously. There is, however, one limitation that must

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

be observed, which is that the X and Y axes of the A0 card (or the X and Y axis residing in one SMCC) must both be used to do the same type of interpolation (either linear or circular). One SMCC *cannot* be used to do circular interpolation with its X axis and another SMCC's X or Y axis while the Y axis of the same card is used to do linear interpolation with another card's X or Y axis. The reason for this limitation is that the X and Y axis on a given SMCC are "tied" together by the internal, common feedrate control clock generator and must perform the same "type" of operation (linear or circular) in order to function correctly. The common feedrate control also limits the SMCC in that totally independent X and Y axis moves cannot be made. See previous programming Example #3.

In order to program multiple SMCCs, the procedure for linear interpolation is straight forward:

```
AAz                     ;Address all cards, clear program area,
                        ;leave subroutines, if any, undisturbed.
Z                       ;Set all current positions to zero.
T2048                   ;Set the time to be taken for any move
                        ;size on all axis to 2 seconds.
t512                    ;Set all accel and decel to 0.5 seconds.
A0X10000Y22000          ;Move the A0 card's X axis 10000 and Y
                        ;axis 22000 encoder counts.
A1X5000Y17000           ;Move the A1 card's X axis 5000 and Y
                        ;axis 17000 encoder counts.
AADWE 1024              ;All axes, wait 1 second
X0Y0                    ;Return all axes on all cards to zero
                        ;position.
END                     ;End of program
```

The above program will execute linearly interpolated moves on all four axes specified.

To execute circular interpolation more care and planning is required. The following is an example of circular interpolation being done between two separate cards. The circle starts at 0/0 goes to 10,000/0 (180 degrees) and then returns to 0/0 to complete the circle.

1) Circular interpolation between X axis on A0 and Y axis on A1:

```
AAz
F1024
R = 10000
CIR 0
A0 U20000WY0  A1 WX20000V0
A0 U-20000WY0 A1 WX-20000V0
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

For reference, to do interpolation on a single card between X and Y axis, the program would look as follows:

```
A0z                         ;Address card A0
F1024                       ;Feedrate to 100%
R = 10000                   ;Radius of 10000 encoder counts
CIR0                        ;Clockwise circle
U20000V0                    ;Incremental move of X axis to 20000 and
                            ;Y axis of 0.  Half circle to 180 degrees.
U-20000V0                   ;Return to 0
END
```

The differences therefore are the introduction of the W (Wait) Command which generates a wait time proportional to a corresponding move time, and the fact that even though the Y axis on A0 and X axis on A1 are not used, they must be treated as a "phantom" axis and be told to wait or do nothing. This is the reason why the unused axis cannot be used to do a linear interpolation while the other is doing circular or vice versa.

2) Circular interpolation between X axis on A0 and X axis on A1:

```
AAz
F1024
R = 10000
A0 CIR0 A1 CIR1                 ;Note that A1's X axis must be
                                ;reversed, using CIR1.
A0 U20000WY0   A1 U0WY20000
A1 U-20000WY0  A1 U0WY-20000
END
```

EXAMPLE #21

MACHINE INPUT/OUTPUT FUNCTIONS (M FUNCTIONS):

SMCC supports 64 M-function I/Os to interface with devices such as switches, status indicators, etc. SMCC defines M-function I/O hardware as follows:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SOFTWARE DEF.	DESCRIPTION	HARDWARE	ASSIGN. ADDRESS	PHYS. ADDRESS	PROG. ON/OFF CONTROL
M01	;Mach. Out.#1	(MAO1)	C01.5	ff01-5	SET 01/RESET 01
M02	;Mach. Out.#2	(MAO2)	C02.6	ffb4-6	SET 02/RESET 02
M03	;Mach. Out.#3	(MAO3)	C02.7	ffb4-7	SET 03/RESET 03
M04	;Mach. Out.#4	(MAO4)	C03.6	ffb5-6	SET 04/RESET 04
M05	;Mach. Out.#5	(MAO5)	C03.7	ffb5-7	SET 05/RESET 05
M11	;Mach. In. #1	(MAI1)	C04.0	ffb7-0	To Read M inputs,
M12	;Mach. In. #2	(MAI2)	C04.1	ffb7-1	in PLC or Program
M13	;Mach. In. #3	(MAI3)	C04.2	ffb7-2	flow, use M11 to
M14	;Mach. In. #4	(MAI4)	C04.3	ffb7-3	M17. Example:
M15	;Mach. In. #5	(MAI5)	C05.0	ffbc-0	IF M12=1 THEN...
M16	;Mach. In. #6	(MAI6)	C05.1	ffbc-1	or, GOTO5IFN(M15=1)
M17	;Mach. In. #7	(MA17)	C05.2	ffbc-2	etc.

Upon turn-on or reset, the SMCC automatically assigns M01 to M05 as outputs, M11 to M17 as inputs, and assumes the above definitions. Please refer to SMCC I/O Map for a better understanding. Any of the above may be reassigned as the user sees fit, but for uniformity's sake it is recommended that M01 to M05 and M11 to M17 be used as shown. As an example, to assign new M functions proceed as follows:

Assume we want to assign M functions to detect the "=" and "Home" switches on the control panel. The I/O map shows these inputs to be coming in at address ff3a bits 6 and 7 for "Home" and bits 2 and 3 for "=". ("Home" and "=" both generate a pair of bits due to the input multiplexing done to reduce individual inputs.) Let us choose C00 which is unassigned (you may check this out by simply typing C00 on line to have SMCC tell you the assignment, or even better, use SMCC V3.X or equivalent IBM PC program to receive allocations for all the C functions).

To assign C00 to be ff3a online, type C00=ff3a. To do the same in the buffer, as part of a program, type SET C00=ff3a. Next, assign an M function to the individual bits:

Online	In Buffer
M30 = C00.2	SET M30=C00.2
M31 = C00.3	SET M31=C00.3
M32 = C00.6	SET M32=C00.6
M33 = C00.7	SET M33=C00.7

Please note that when choosing Cnn to assign for M functions, Cnn may be C00 to C31 only, not C63.

Now, to use these M functions as "=" and "Home" detection, the program or conditional statements are written as:

```
GOT05 IF (M30=1 AND M31=1)
GOT62 IFN(M32=1 AND M33=1)
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Which means go to subroutine 5 if M30 and M31 are both true, (the "=" switch is on) and if not, then do not go to subroutine 62 if "Home" switch is on, but go to 62 if "Home" is not on.

To control outputs with M functions, a simple example would be SET 01 in the buffer, which would cause the M01 output to be set high. To accomplish the same online, use M01 = 1.

Type i021, this places SMCC in MDI mode where instructions do not go into the buffer, but get executed immediately upon reception of a <CR>.

Type SET01<CR> and note that this does not turn on M01.

Now type DWE0<CR> and note that M01 is now turned on. The reason is that the SET function in SMCC is executed at the beginning of the next instruction. DWE0 is used here as a harmless next instruction.

To turn off M01:

Be sure i02 is at 1 (MDI turned on).
Type RES01 <CR>.
Next type DWE0 <CR> and note that M01 now turns off.

To exit i02 1 mode, type i020, or "S" Step or "R" Run.

EXAMPLE #22

"WAIT UNTIL" PROGRAM

Wait until input is true then proceed. This program is written as a subroutine and may be called as needed.

```
    !
    !
  GOS01                    ;Go to subroutine 01
    !
    !
  L01                      ;Subroutine 01 (Label 01)
  DWE20                    ;Wait time of 20 binary milliseconds
                           ;(10 minimum)
  GOT01 IF (M11=0)         ;If M11 input (MAI1) is not
                           ;on, keep looping on subroutine
                           ;01.  When M11 becomes true,
                           ;continue.
  SET 01                   ;Turn on M01 output (MA01),
    !                      ;etc.
    !
    !
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The same result can be achieved by using a GOT-01 as follows:

```
    !
    !
DWE20                    ;Wait time of 20 binary milliseconds.
GOTO-01IF(M11=0)         ;Jump back one line
SET01                    ;as long as M11 is false.
    !
    !
```

EXAMPLE #23

LEARN COMMAND IN PROGRAMMING:

l -- Learn and Store X and Y Position

The Learn ("l") Command can be used to store positions in the program from the actual reading of the current position of X axis alone if i15 is set to a one (one axis operation), or X and Y axis together, if i15 is set to a 0 (two axis operation).

The learn function actually reads the current positions of X and Y and "writes" them into the buffer at the next available step. There are two basic operations that can be done in learn; one is to start with an empty program buffer and learn a series of points, the other is to run through a program already in the buffer and modify existing positions by doing a "learn/replace". The SMCC does a "learn" or "learn/replace" simply by examining the status of the "buffer full" bit. If the buffer is not full, the SMCC does a "learn". If the buffer is full, it does "learn/replace" at whatever point in the program buffer it may be.

An example of "learn" follows:

```
A0z                      ;Clears buffer.
j                        ;Jog to desired zero position
Q or j                   ;Quit jogging
Z                        ;Set zero position.  Note that the A0,z,j,
                         ;J,Z and Q Commands are not stored in the
                         ;program buffer.
F1024                    ;Maximum speed, note that it is possible
                         ;to enter commands in the buffer as
                         ;needed.
SET 01                   ;Turn M function M01 on
J                        ;At this point, move the X axis
                         ;to the desired position.  X only is
                         ;being moved, in the positive direction.
                         ;Stop X axis
J or Q                   ;Note that the handwheel may also be
                         ;used to move the motors for precise
                         ;positioning.  Also, it is possible
                         ;to issue the "k" Kill Command which
                         ;disables the servos and allows
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;moving the  motors manually to the
                        ;position to be learned.  After a "k",
                        ;use a "Q" to restore servo position.

                        ;Upon reception of the "1" Command,
1                       ;SMCC will learn the current position
                        ;and the appropriate instruction will
                        ;be written automatically in the
                        ;buffer.
DWELL 1024              ;Wait for 1 second.  This is typed in data.
F256                    ;Change speed
J                       ;Move to the next desired position
J                       ;Stop jogging
1                       ;Learn this point
RESET 01                ;Turn M01 off

X0                      ;Back to position zero; note that Move
                        ;Commands may also be typed in.
END                     ;End of program, sets the "buffer full"
                        ;flag on
LIST                    ;Check the program

F1024
SET01
X21276Y0                ;May need correction
DWE1024
F256
X54321Y0
RES01
XO
END
```

This program may now be edited as required, or it may be "learn/replaced" if needed by stepping through the program using jog, handwheel, or "k" Command to displace the motor and issuing an "1" Command to store the new position in place of the old.

EXAMPLE #24

ALTERED DESTINATION ON TRIGGER:

This program shows how an SMCC (either Standard or Parabolic) can be programmed to begin a move and upon reception of an external trigger, to alter its destination. The program is treated as a subroutine, which can be invoked by a GOS00 from the main program.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
GOS00                    ;Go to Subroutine 00
  !
  !
  !
LOO                      ;Subroutine OO (Label OO).
mffca,1                  ;Enable interrupts so if a trigger is
                         ;received by the "Home Flag" or "User
                         ;Flag" on the X axis, the absolute
                         ;position of X can be frozen and stored
                         ;in the Hx registers.  Note that i26
                         ;controls whether the "Home" or "User"
                         ;flag will generate the interrupt and
                         ;the polarity of the interrupt.  For
                         ;the Y axis, the control register is
                         ;ffc8 instead of ffca.

SET C20 = ffca           ;Define C20 as the address of the
                         ;capture register
SET M20 = C20.6          ;Define M20 as bit 6 of ffca.  When
                         ;this bit is set true, interrupts are
                         ;disabled.  To re-enable interrupts, use
                         ;mffca, 1.
T50                      ;Time mode moves, 50 milliseconds per
                         ;block for all succeeding moves.
L01                      ;Subroutine 01 (Label 01).
U1000                    ;Make an incremental move of 1000 encoder
                         ;counts.
GOT01 IF(M20=0)          ;Go to Label 01 if interrupt has not
                         ;occurred (trigger has not been received).
                         ;This conditional statement causes the
                         ;SMCC to keep returning to Subroutine 01
                         ;to make continuous, incremental moves of
                         ;1000 counts so long as the trigger has not
                         ;been received.  This motion will go on
                         ;forever until trigger is received.
                         ;(See alternate program for a position
                         ;limited move.)
P1=HX                    ;If interrupt has occurred, then set
                         ;parameter P1 to be equal to the frozen,
                         ;captured absolute position value of X
                         ;at the time trigger occurred.

                         ;It is important to note that if the
                         ;interrupt occurs during the approximately
                         ;10 to 20 millisecond period preceding
                         ;each move (calculation time), it will
                         ;not be detected and calculated until
                         ;the next move.  Depending upon the
                         ;offset distance to travel, this may
                         ;cause an overshoot and return to final
                         ;position.  The incremental move should
                         ;therefore be as small as practically
                         ;possible.  The Parabolic version
                         ;actually allows you to specify the
                         ;calculation time with i12, so that
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

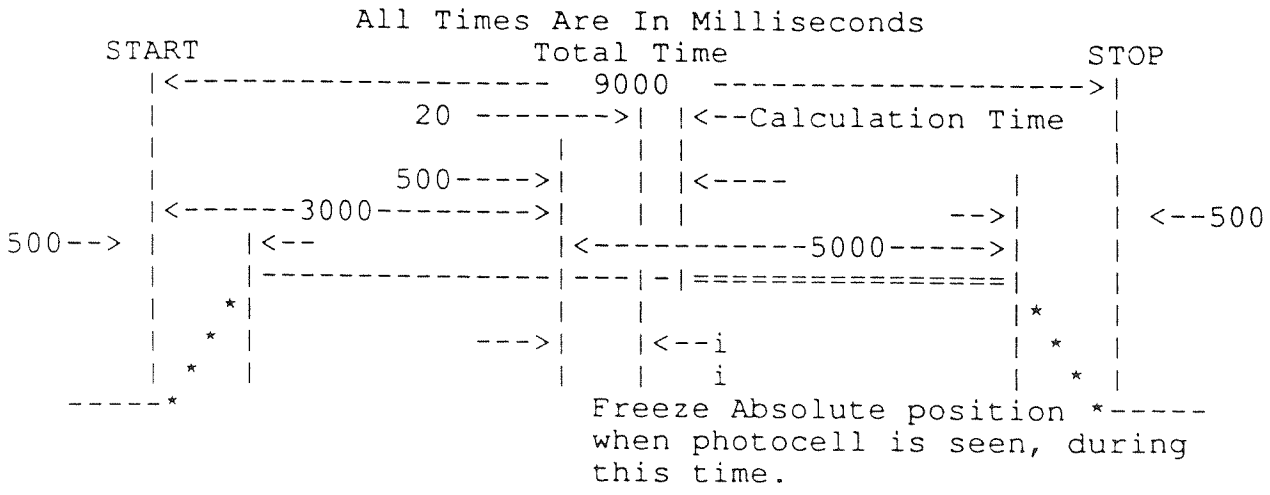
```
                              ;the "Dead Zone" can be precisely
                              ;predictable.
AX(P1 + P2)                   ;Make a move to absolute position
                              ;of HX plus an offset move of an
                              ;amount set by the value of
                              ;Parameter P2.  Note that P2 can
                              ;be positive or negative.
RET                           ;End of subroutine, return to main
                              ;program.
```

If you wish the trigger search to end at a certain position and not go forever, alter L01 and add L02 as follows:

```
L01                           ;Subroutine 01 (Label 01)
GOT02IF(X>P5)                 ;Go to Subroutine 02 if X position
                              ;is greater than Parameter P5.
U1000                         ;Otherwise, make an incremental
                              ;move of 1000 counts.
GOT01IF(M20=0)                ;Loop on Subroutine 01 until
                              ;interrupt occurs.
P1=HX                         ;
AX=(P1 + P2)                  ;
L02                           ;Subroutine 02
RET                           ;Return to main program
```

EXAMPLE #25

CONSTANT TIME, ALTERED DESTINATION ON TRIGGER:




```
A0z100                        ;Address card A0, clear buffer, and
                              ;allocate 100 lines for program.
RES 01                        ;Reset the "No Interrupt Received" flag.
SET C21=ffca                  ;Home/User interrupt register address
SET M23=C21.6                 ;interrupt disable flag.  When M23 is set,
                              ;interrupts are disregarded.
t500                          ;Set accel/decel time to 0.5 seconds
T3000                         ;Move time, 3.0 seconds
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
U3000                ;Incremental move 3000 encoder counts in 3.0
                     ;seconds, distances used are arbitrary and
                     ;are assumed to be 1 count/millisecond
mffca=1              ;enable interrupt for detecting photocell
                     ;trigger pulse during next move.
T500                 ;Move time 500
U500                 ;Incremental move 500 counts.  During this
                     ;move, the photocell is being looked at
                     ;and if a transition occurs, the absolute
                     ;position of X at that time is frozen and saved.
                     ;Later, this position is used to make a
                     ;move to the final destination  without
                     ;altering the total time.  The last 20
                     ;milliseconds of the move are used to
                     ;recalculate the new required velocity.
                     ;Photocell will not be detected during
                     ;this time, so that the actual time
                     ;window for photocell capture is 480
                     ;milliseconds.
T5000                ;Move time 5 seconds
GOS00                ;Go To Subroutine 00
END                  ;End of program

L00                  ;Subroutine 00
GOT01 IF(M23=1)      ;Go to Subroutine 01 if interrupt was
SET01                ;received or photocell was detected. If
SET 23               ;disable  interrupt, disregard other input
                     ;pulses from photocell if
U5000                ;not, move    another 5000 counts and
RET                  ;stop.  This represents the default move,
                     ;if the photocell is not detected.
                     ;Also, SET01 notifies the operator that a
                     ;photocell pulse was not received.  An LED
                     ;may be used as visible indicator.

L01                  ;Subroutine 01
SETP3=HX             ;Set Parameter P3 value to the frozen,
                     ;captured absolute position
                     ;of the X axis.  HY captures Y position.
SETP3=P3 + 5250      ;Add to the absolute position value the
                     ;distance you wish the X axis to travel
                     ;for correct position at the end of
                     ;travel.  5250 is a nominal value.
                     ;It is a good idea to keep the default
                     ;move reasonably same in size as the actual
                     ;needed move because otherwise, a violent
                     ;velocity change may be needed.
AX(P3)               ;Make a move as specified by Parameter
                     ;P3.  Note that move time is 5 seconds
                     ;and that the program calculates the
                     ;value of P3.  This move is now an
                     ;absolute position move not incremental
                     ;and it therefore keeps track of the
                     ;additional travel between the photocell
                     ;detection point and the beginning of the
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ;next move.  Also, note that velocity
                    ;during final move will change depending
                    ;upon where the photocell detector is
                    ;triggered, and the final destination
                    ;distance specified.
RET                 ;Return to program and END.
```

The object of this program is to make a precisely timed move and make an on-the-fly destination position change based on the reception of a trigger on the SMCC's X and Y "Home Flag" or "User Flag" inputs. These are special interrupt driven inputs which capture X and Y position on-the-fly with less than 100 microseconds of error. The calculation capability of the variable parameters is then used to construct an altered, absolute position move to a final desired destination. The program can be used with Standard or Parabolic SMCC. The difference is that Parabolic uses i12 to define a Move Calculation Time, thus allowing the user to define a precise trigger window. With the Standard SMCC, the calculation time varies, thus making the trigger window slightly variable.

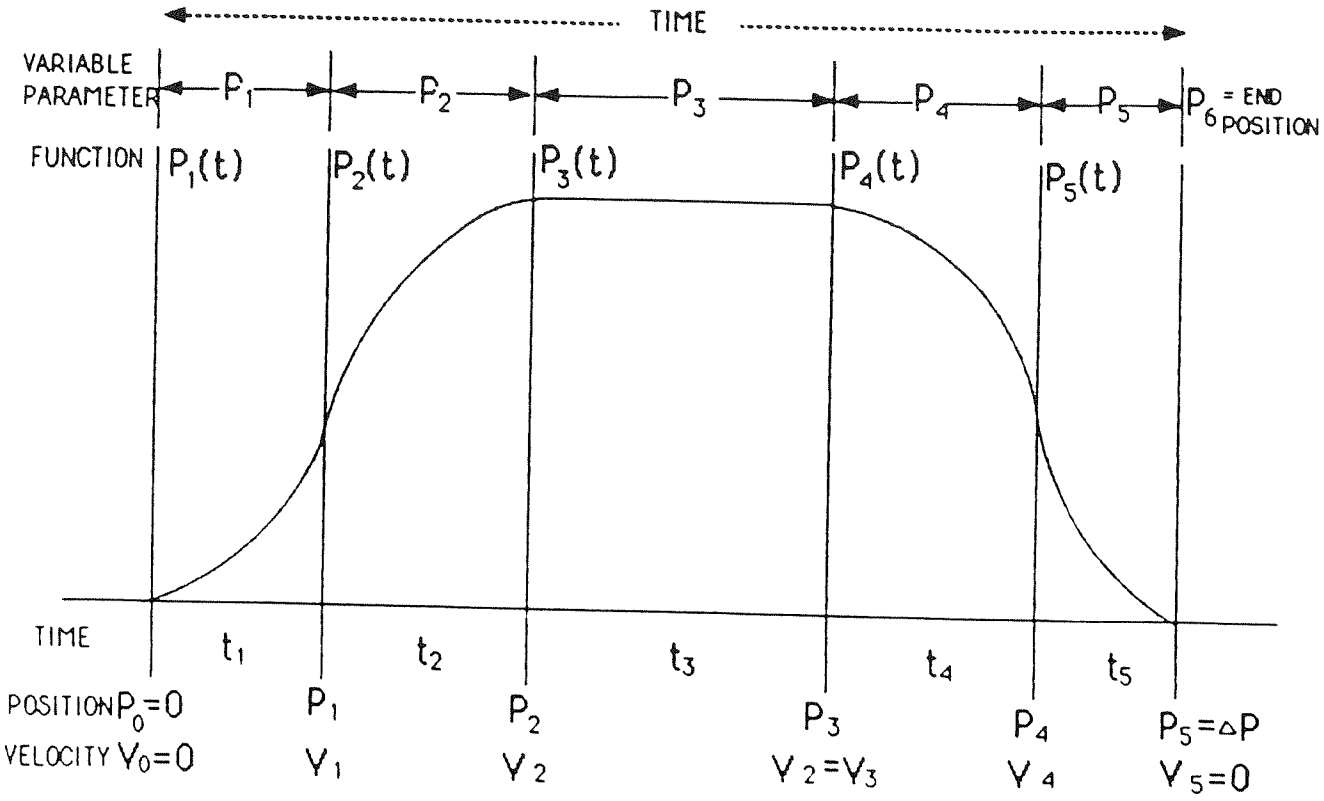
EXAMPLE #26

DEMO 3 - "S" CURVE ACCEL AND DECEL (REQUIRES PARABOLIC OPTION):

```
A0z100              ;Clear buffer
i061024             ;Set feedrate for lines/sec.
SETC20=febd         ;Set address for M20
SETM20=C20.5        ;Set M20 to save data for plot
SETP12=(P1+P2)      ;P12 = t1+t2
SETP14=(P4+P5)      ;P14 = t4+t5
SETP7 =(P1*P1)      ;P7  = t1*t1
SETP11=(P5*P5)      ;P11 = t5*t5
SETP8 =(P12+P2)     ;P8  = t1+2*t2
SETP10=(P14+P4)     ;P10 = 2*t4+t5
SETP13=(P8+P10)     ;P13 = t1+2*t2+2*t4+t5
SETP13=(P13+P3)     ;P13 = t1+2*t2+t3+2*t4+t5
SETP13=P13+(P3+P3)  ;P13 = t1+2*t2+3*t3+2*t4+t5
SETP12=(P12*P13)    ;P12 = (t1+t2)*(t1+2*t2+3*t3+2*t4+t5)
SETP14=(P14*P13)    ;P14 = (t4+t5)*(t1+2*t2+3*t3+2*t4+t5)
SETP7=(P6*P7/P12)   ;P7  = DP1=DP*t1*t1/(t1+t2)*
                    ;(t1+2*t2+3*t3+2*t4+t5)
SETP11=(P6*P11/P14);P11 = DP5=DP*t5*t5/(t4+t5)*
                    ;(t1+2*t2+3*t3+2*t4+t5)
SETP8 =(P6*P8/P13) ;P8  = (DP1+DP2)=DP*(t1+2*t2)/
                    ;(t1+2*t2+3*t3+2*t4+t5)
SETP10=(P6*P10/P13);P10 = (DP4+DP5)=DP*(2*t4+t5)/
                    ;(t1+2*t2+3*t3+2*t4+t5)
SETP9 =(P8+P10)     ;P9  = (DP1+DP2) + (DP4+DP5)
SETP9 =(P6-P9)      ;P9  = DP3 = DP-(DP1+DP2) - (DP4+DP5)
SETP8 =(P8-P7)      ;P8  = DP2 = (DP1+DP2) - DP1
SETP10=(P10-P11)    ;P10 = DP4 = (DP4+DP5) - DP5
SETP15=1024         ;Time scale in binary msec.
SETP13=(P15*P9/P3) ;P13 = V2 = V3 = DP3/t3
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE



"S" CURVE ACCELERATION

S CURVE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SETP15=3072          ;3*Time scale in binary msec.
SETP12=(P15*P7/P1)   ;P12 = V1 = 3*DP1/t1
SETP14=(P15*P11/P5);P14 = V4 = 3*DP5/t5
FOR0                 ;Start loop
STA                  ;Start run mode
SET20                ;Start data store
t(P1)                ;t1 = P1
U(P7) : (P12)        ;DP = DP1 = P7 : V = V1 = P12
t(P2)                ;t2 = P2
U(P10): (P13)        ;DP = DP2 = P8 : V = V2 = P13
t(P3)                ;t3 = P3
U(P9) : (P13)        ;DP = DP3 = P9 : V = V3 = P13
t(P4)                ;t4 = P4
U(P10): (P14)        ;DP = DP4 = P10: V = V4 = P14
t(P5)                ;t5 = P5
U(P11): 0            ;DP = DP5 = P11: V = 0
STO                  ;Stop
STA                  ;Start run mode
SET20                ;Start data store
t(P1)                ;t1 = P1
U(-P7) : (-P12)      ;DP = -DP1 = -P7 :V = -V1 = -P12
t(P2)                ;t2 = P2
U(-P8) : (-P13)      ;DP = -DP2 = -P8 :V = -V2 = -P13
t(P3)                ;t3 = P3
U(-P9) : (-P13)      ;DP = -DP3 = -P9 :V = -V3 = -P13
t(P4)                ;t4 = P4
U(-P10): (-P14)      ;DP = -DP4 = -P10:V = -V4 = -P14
t(P5)                ;t5 = P5
U(-P11):0            ;DP = -DP5 = -P11:V = 0
STO                  ;Stop
END                  ;End of loop and program
```

The program shown allows the user to utilize it as a means of making "S" curve accelerated moves. Six parameters need to be passed to the program in order to make it function properly:

Parameter P1: Defines the time to the first velocity inflection point.

Parameter P2: Defines the time from the inflection point to the point where run velocity is reached.

Parameter P3: Defines the run time.

Parameter P4: Defines the time from end of run position to decel inflection.

Parameter P5: Defines time from decel inflection point to end position.

Parameter P6: Defines total move distance.

The program is written so as to allow a completely generalized nonsymmetrical motion to occur. Parameter definitions can be

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

simplified if the application can use identical accel and decel ramps and a symmetrical inflection point, in other words, P1 = P2 = P4 = P5 and by adding the following to the beginning of the program:

```
SET P1 = VALUE
SET P3 = VALUE
SET P6 = VALUE
SET P2 = (P1)
SET P4 = (P1)
SET P5 = (P1)
```

Note: i06 must be set to 1024 in order to keep time scaling in lines/sec.

A move can then be made by defining P1 = inflection point time value in binary milliseconds, P3 = run time value, and P6 = total distance value in encoder counts. The minimum time that can be specified for any of the time parameters is 15 milliseconds. The advantage of "S Curve" acceleration is that it minimizes da/dt, rate of change of acceleration, also known as "Jerk" and as a result provides much smoother operation and reduces wear and tear on machinery.

EXAMPLE #27

CIRCULAR CUT ON THE FLY:

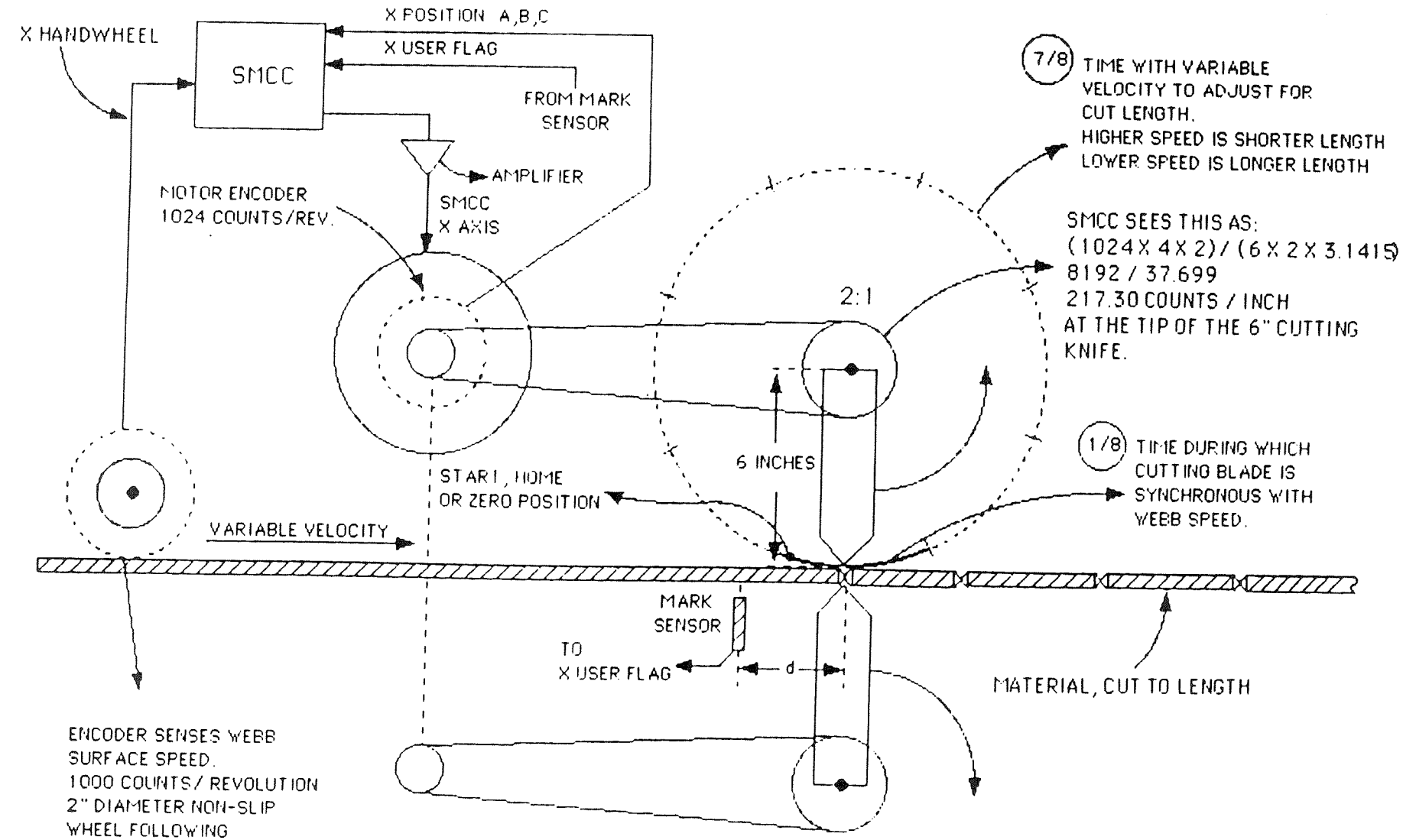
This example concerns the simple case where the web material is to be cut off without the need for compensating stretch which would require that a photocell detector be used to sync on marks made on the web in order to insure proper cutting position (see next example for this more complex case).

For the simple case of rotary cutoff, there is only a need to synchronize the tip of the cutting blade with the material's (web's) velocity when the cut occurs and then adjust the knife's velocity for proper length of material when the knife is not in contact with the material. The limitation of this system is that short lengths to be cut require a high degree of acceleration and high velocity during the nonsynchronized part of the move. It is therefore a good idea to design the program in such a way as to provide maximum "adjustment" time. Amplifier and motor sizing should be carefully chosen so specified performance can be met.

Even though the cut length will be independent of web speed, care must be taken *during* speed changes since some deviations may occur due to servo following error. The Parabolic CPU version has velocity and acceleration feedforward terms which can be used to minimize this type of error.

Let us make one further simplification in stating that for the initial program derivation we can assume that the encoder counts

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



(CIRCULAR CUT)

CIRCULAR CUT ON THE FLY

Artisan Scientific - Quality Instrumentation... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

per inch at the knife tip and that of the web velocity measuring encoder are the same.

The SMCC is equipped with a highly useful mode of operation, referred to as i16 2 mode, where the handwheel encoder is used as the time base for executing programs in the SMCC's buffer. The basic formula used to convert encoder counts to time is:

(i27/i47) x 32768 counts/binary millisecond for Parabolic CPU

and

(8 x 32768)/i27 counts/binary millisecond for Standard CPU

Please note that SMCC's Parabolic option allows for more latitude and precision by providing both i27 and i47 as variables. Also, please note that i27 or i47 may not be set to a value of one.

Assume we use parabolic, then setting i27 to 8 and i47 to 32768 yields:

(8/32768) x 32768 = 8 counts/binary millisecond.

This means that if in the SMCC program a time value of 128 is used, it will represent 128 x 8 = 1024 handwheel encoder counts.

Assume we divide the blade's move time into eight segments (see figure). The resulting program looks as follows:

```
A0z100              ;Address card, clear buffer and assign 100
                    ;program lines.
i160                ;Sets i16 to zero so SMCC can function
                    ;without web running.
F1024               ;Set feedrate to 100% of i06
HOM                 ;Do a Home Cycle to the start position.
                    ;Note that home position represents the
                    ;beginning of the synchronized blade
                    ;motion for cutting the material.  Adjust
                    ;home offset so zero position is
                    ;approximately 1/2 the sync. distance
                    ;before the cut point.
i162                ;Slave the SMCC to the web speed.  It is
                    ;assumed that i27 was set to 8 and i47 to
                    ;32768 for this application.
FOR0                ;Loop forever
T128                ;Time for move represents 1024 handwheel
                    ;counts (128 x 8)
U1024               ;Incrementally move X axis 1024 counts
                    ;which represents 1/8 of a circumference
                    ;of the blade.  This part of the move is
                    ;the "synchronized" part where the material
                    ;gets cut.
T(P1)               ;Set the time for 7/8 of the blade's
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;movement, during which velocity is
                        ;changed in order to match the desired
                        ;length.  Total material cut distance is
                        ;(P1 + 128) x 8 number of encoder counts.
                        ;Clearly, T can be made a function of
                        ;several P variables and can be set by
                        ;external thumbwheels so as to represent
                        ;the correct cut distance in inches,
                        ;millimeters, or other units.
U7168                   ;Move blade back to beginning of sync
                        ;point, 7/8 of circumference.  Total
                        ;encoder counts for one blade rotation
                        ;are 1024 x 2 x 4 = 8192,
                        ;1024 + 7168 = 8192.
END                     ;End of program
```

To adjust for the different encoder counts per inch between the web encoder and the knife encoder, program is modified as follows:

```
T128
U349                    ;(1024 x 217.30)/636.64 = 349
T(P1)
U7843                   ;8192-349 = 7843
END
```

Note that due to the encoder scaling, the synchronized versus the freewheeling distance ratio is now nearly 1 to 20. This is still okay provided the motor can accelerate or decelerate fast enough.

EXAMPLE #28

INDEPENDENT X AND Y AXIS HOMING ROUTINES:

It is often necessary to home the X and Y axes independently of each other. Even though such a procedure is possible by setting i15 to 1 then i15 to 0, the Y axis may drift while X is homing. The use of the following routines provides a general purpose manner in which to perform accurate homing cycles independently. In this example, the main program can call the X homing routine with GOSUB 00 and they with GOSUB 10.

```
C30=ffca                ;Define C30 as address ffca.
M30=C30.6               ;Define M30 as bit 6 of ffca which indicates
                        ;when the home flag interrupt has occurred.
M18=C04.4               ;M18 is defined as bit 4 or C04 which, if
                        ;undisturbed by new instructions, is
                        ;specified as address ffb7 by the SMCC upon
                        ;power turn-on or reset.  Bit 4 of ffb7 is
                        ;the X axis home flag interrupt input.  Bit
                        ;6 is the Y axis equivalent.  (See I/O Map)
L00t20                  ;Label 00 performs a homing function for X
                        ;axis alone.  The routine checks for home
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                        ;switch presence first.  If present, will
                        ;move to clear it and then will execute
                        ;a homing function which will automatically
                        ;detect the home switch, then the encoder
                        ;"C" channel, make an offset move, and
                        ;finally set its final position to a
                        ;specified value.  The t20 specifies accel/
                        ;decel times of about 20 milliseconds.
F128                    ;Set feedrate to desired home feedrate.
GOT02IF  (M18=1)        ;Go to label 02 if the home switch is on.
                        ;Label 02 will cause the axis to move in
                        ;a positive direction first until the home
                        ;switch is cleared.
L01U100                 ;Label 01, make a positive incremental 100
                        ;count move.
GOT01IF(M18=0)          ;Go back to Label 01 if home switch is still
                        ;not present.  Keep making +100 count moves.
LO2mffca,1              ;Enable X axis home flag interrupt when the
                        ;home switch goes away.
L03U-100                ;Label 03, make an incremental -100 count
                        ;move.
GOT03IF(M30=0)          ;Go to Label 03 if home/flag (encoder "C"
                        ;channel) is not present, interrupt has not
                        ;occurred.  Keep making a -100 count move
                        ;until it does.
SETP15=HX               ;If home switch is present and interrupt
                        ;has occurred by "C" channel of encoder,
                        ;capture position at which the "C" channel
                        ;occurred and assign this value to P15.
SETP15=P15+500          ;Add an offset move of 500 counts to P15.
                        ;The 500 can be any positive or negative
                        ;value or another parameter.
AX (P15)                ;Make an absolute move relative to the
                        ;captured position plus the offset move
                        ;value.  The final position will be
                        ;precisely 500 counts from the position
                        ;at which the home flag occurred.
X=0                     ;Set the final position to be 0 or any
                        ;any other value.
RET                     ;Return to main program.  Take care to
                        ;restore proper accel/decel times and
                        ;rates as required.
```

For Y axis:

1) Replace ffca with ffc8
2) M18=C04.4 with M19=C04.6
3) U with V.
4) X with Y.
5) L00, 01, 02, 03 becomes L10, 11, 12, 13.

```
C30=ffc8
M30=C30.6
M19=C04.6
L10t20
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
F128
GOT12IF (C04.4=1)
L11V100
GOT11IF (C04.4=0)
L12mffca,1
L13V-100
GOT13IF (M30=0)
SETP15=HY
SETP15=P15+500
AY(P15)
Y=0
RET
```

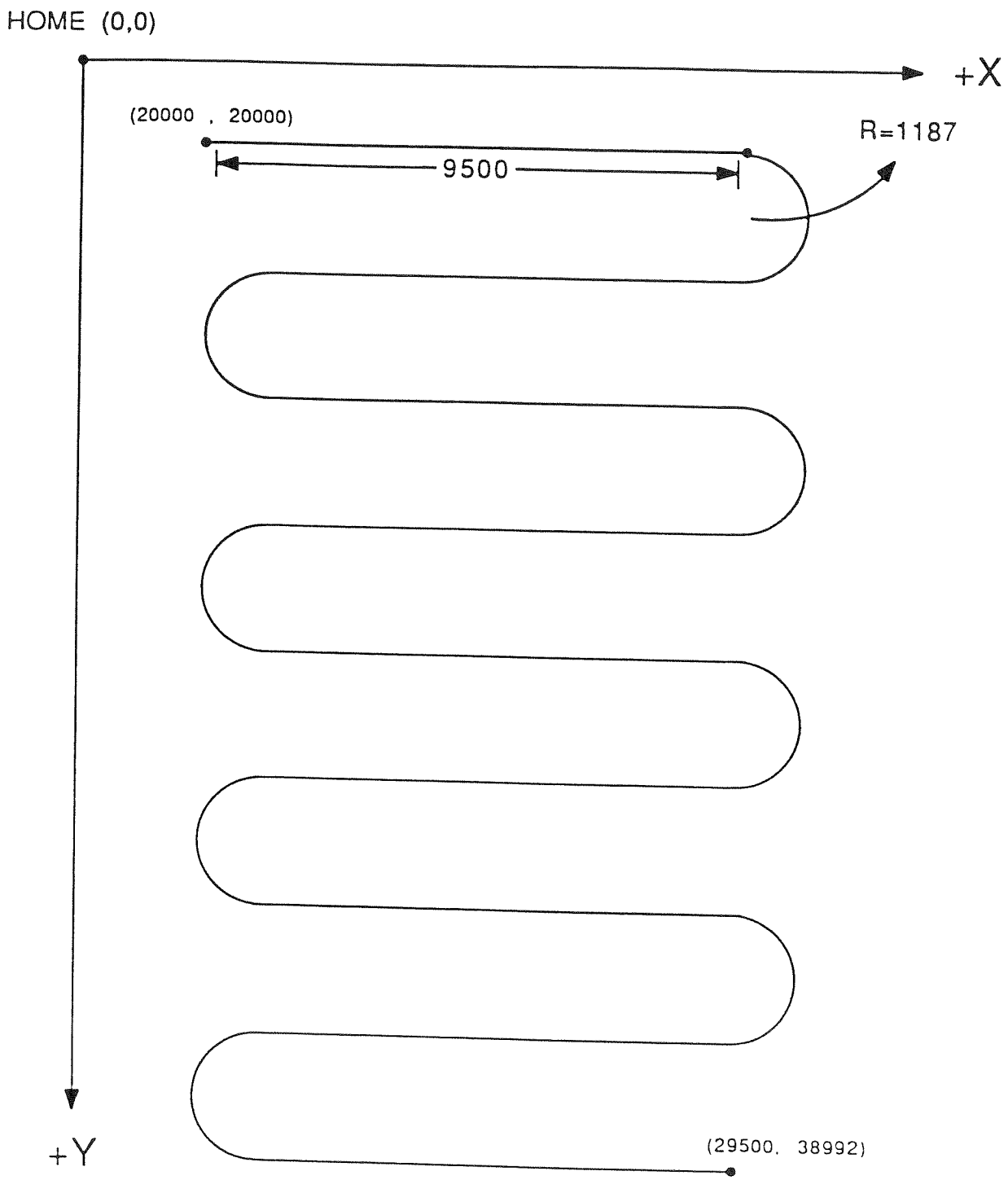
EXAMPLE #29

SERPENTINE PATTERN

This program demonstrates the use of subroutines to create a serpentine pattern which uses "Line to Circle" and "Circle to Line" connections. See figure "Serpentine" for actual path description.

```
A0z100              ;Address card 0, clear buffer
F324                ;Feedrate
HOM                 ;Go home
t150                ;Set accel/decel for 150 msec.
LIN                 ;Linear mode
X20000Y20000        ;Move X and Y to 20000 counts
DWE1024             ;Wait for 1 second
SET01               ;Turn on M01, used as Pen Up/Down control
                    ;(Pen Down)
DWE100              ;Wait for about 100 msec for pen to come down
U9500               ;Incremental move X to 9500 counts
GOS00               ;Call Subroutine 00
GOS01               ;Call Subroutine 01
GOS00               ;Call Subroutine 00
GOS01               ;Call Subroutine 01
GOS00               ;Call Subroutine 00
GOS01               ;Call Subroutine 01
GOS00               ;Call Subroutine 00
GOS01               ;Call Subroutine 01
DWE100              ;Wait 100 msec
RES01               ;Turn off M01 (Pen Up)
DWE100              ;Wait for about 100 msec. for pen to go up
X0Y0                ;Return X and Y to 0 position
END                 ;End of program
L00CIR1             ;Subroutine 00, circular CCW mode
R=1187              ;Set radius to 1187 counts
V2374               ;Incremental move one diameter
LIN                 ;Linear mode
U-9500              ;Incremental move X back 9500 counts
RET                 ;Return to main program
L01CIR0             ;Subroutine 01, circular CW mode
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



(SERP)

"SERPENTINE"

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
R=1187                  ;Set radius to 1187 counts
V2374                   ;Incremental move one diameter
LIN                     ;Linear mode
U9500                   ;Incremental move X to 9500 counts
RET                     ;Return to main program
```

EXAMPLE #30

COUNTER USING M-FUNCTION INPUT (COUNTER):

The counter is triggered on the trailing edge of MI1 input.

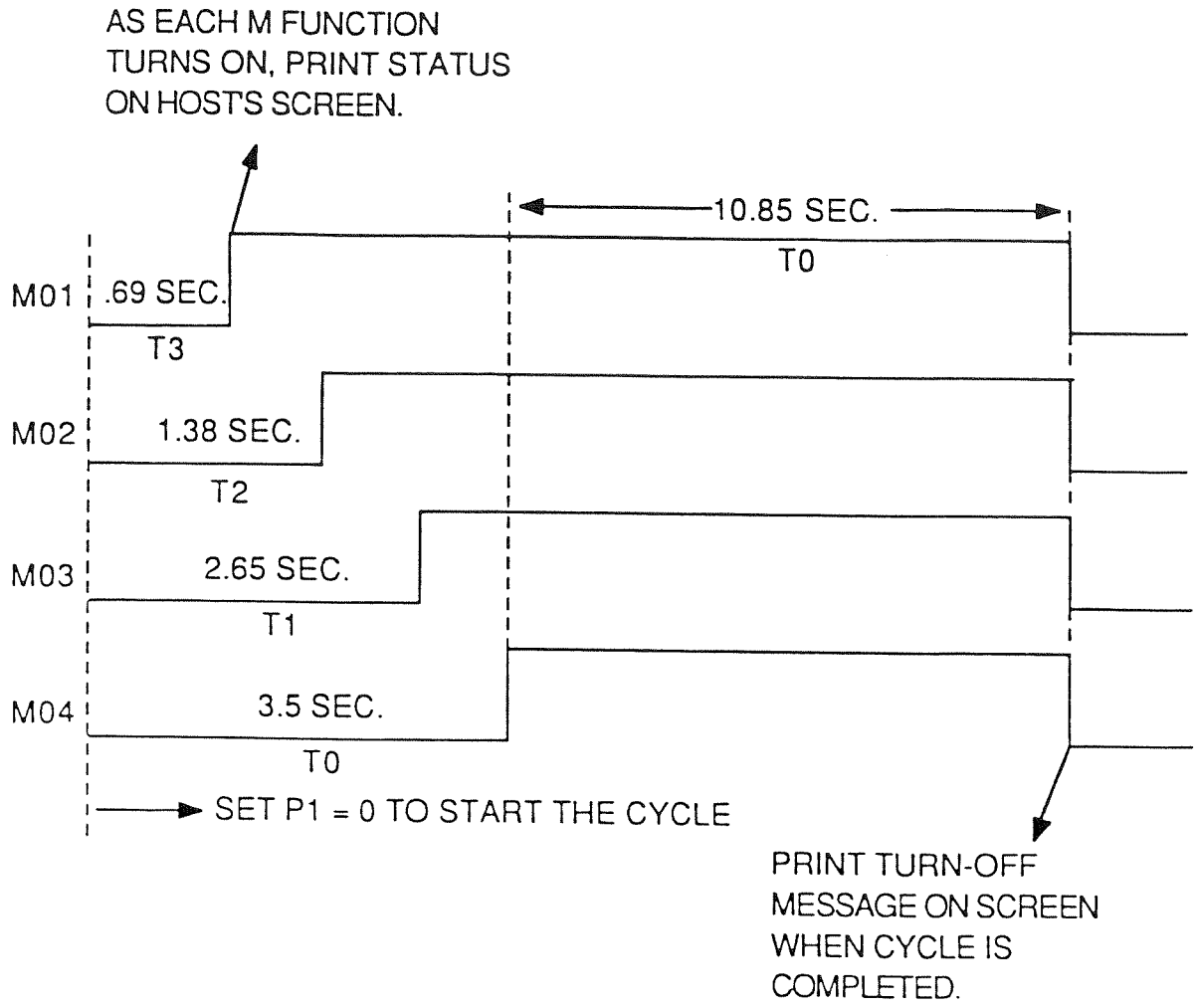
```
A0PURGE                 ;Clear PLC program
IF M11=0                ;Check if MI1 switch is toggled
                        ;M30 is used as flag to cause
                        ;the P1 counter to only count
                        ;once when MI1 switch toggles.
THEN IF M30=0           ;Check if M30 flag is set
THEN M30=1              ;True, reset M30 flag
AND P1=P1+1             ;Add 1 to P1 counter
AND"P1"                 ;Display value P1 on the screen
ELSE                    ;
IF M11=1                ;Check if MI1 switch is released
THEN M30=0              ;True, set M30 flag
AND IF P1>999           ;Check if P1 counter has
AND P1<1001             ;reached 1000 counts.
THEN P1=0               ;True, reset counter
AND SEND"1000-counts";Send a message to the screen
ELSE
END                     ;End of PLC program
```

EXAMPLE #31

EXAMPLE OF USE OF TIMERS (TIME1) TO CONTROL OUTPUTS:

```
A0z100                  ;Clear buffer and allocate 100
                        ;program lines in the buffer
                        ;program to generate 4 waveforms
                        ;by using 4 timer ports. Refer to
                        ;attached figure "M Function
                        ;Output" for shape of waveform.
                        ;NOTE: all timers are used for
                        ;countdown only.
I621                    ;Turn on PLC
RES01                   ;Reset M01
RES02                   ;Reset M02
RES03                   ;Reset M03
RES04                   ;Reset M04
RESP                    ;Reset all P (variables)
END                     ;End of program
A0PURGE                 ;Clear existing PLC program and
                        ;start new PLC program
IF P1<1                 ;Flag to activate timers
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



M FUNCTION OUTPUTS BY PLC TIMER CONTROL

(SEE ATTACHED PLC PROGRAM)

(M FUNCTION OUTPUT)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ;NOTE: To activate timers, set P1
                    ;to any value less than 1.
THEN P1=1           ;True, set P1 to 1
AND T0=80           ;Set timer 0 to 3.5 sec.
AND T1=60           ;Set timer 1 to 2.6 sec.
AND T2=60000        ;Set timer 2 to 1.38 sec.
AND T3=30000        ;Set timer 3 to 0.69 sec.
IF T0=0 AND P3<1    ;Check if T0 is finish countdown
                    ;Note that in this PLC program, P2
                    ;P3, P4, P5, and P6 are used as flags
                    ;to print each message only once
                    ;on screen.
THEN M04=1          ;True, turn on M04
AND P3=1            ;Set P3 to 1
AND SEND"M04-ON"    ;Print message on screen
AND P2=1            ;Set P2 to 1
AND T0=250          ;Set timer 0 to 10.85 sec.
ELSE IF T1=0        ;False, check T1
AND P4<1
THEN M03=1          ;True, turn on M03
AND P4=1            ;Set P4 to 1
AND SEND"M03-ON"    ;Print message on screen
ELSE IF T2=0        ;False, check T2
AND P5<1
THEN M02=1          ;True, turn on M02
AND P5=1            ;Set P5 to 1
AND SEND"M02-ON"    ;Print message on screen
ELSE IF T3=0        ;False, check T3
AND P6<1
THEN M01=1          ;True, turn on M01
AND P6=1            ;Set P6 to 1
AND SEND"M01-ON"    ;Print message on screen
ELSE IF T0=0        ;Check if T0's second countdown is
AND P2>0            ;finished
THEN M01=0          ;True, turn off M01
AND M02=0           ;Turn off M02
AND M03=0           ;Turn off M03
AND M04=0           ;Turn off M04
AND P2=0            ;Set P2 to 0
AND SEND"M01-to-M04-are-off"  ;Print message on
                              ;screen
ELSE                ;False, do nothing
END
```

EXAMPLE #32

MULTIPLE "MESSAGE" TRANSMISSIONS (TIME3):

This program prints three message lines on the screen by using a timer to generate the necessary delay of approximately 600 msec. between transmissions. Consecutive, undelayed messages may cause the host computer to lose data if it is trying to receive and display the message simultaneously. To run the program toggle the "=" switch. The three line message will be sent each time the switch is toggled.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A0z                 ;Clear buffer
SET C00=ff3a        ;Assign C00 to address ff3a
SET M30=C00.2       ;Set M30 to bit 2 of C00
SET M31=C00.3       ;Set M31 to bit 3 of C00
                    ;Note that ff3a and M30,M31 define
                    ;"=" (PREJOG) SWITCH input to SMCC
                    ;(See address map)
SET P1=1            ;Set P1 to 1
SET P2=0            ;Set P2 to 0
RES30               ;Reset M30
RES31               ;Reset M31
RES01               ;Reset M01
I621                ;Enable PLC
END                 ;End of program
                    ;Note that there is no motion
                    ;program in this PLC example.
A0PURGE             ;Clear existing PLC program and
                    ;start new PLC program
IF M30=1 AND M31=1 ;Check if "=" switch was toggled
THEN T0=15          ;Yes, set T0 timer to 600 msec.
AND P1=0            ;Set P1 to 0
IF T0=0 AND P1<1 ;Check if both T0 and P1 are 0
THEN IF M01=1       ;Yes, check if M01 is on
THEN M01=0          ;Yes, turn off M01
AND P1=1            ;Set P1 to 1
AND P2=1            ;Set P2 to 1
AND T0=15           ;Set T0 timer to 600 msec.
AND SEND"How-are-you";Print second message just once
ELSE M01=1          ;No, turn on M01
AND T0=15           ;Set T0 timer to 600 msec.
AND SEND"Hello"  ;Print first message, just once
ELSE IF T0=0 AND P2>0;No, check T0 is 0,P2 is not 0
THEN P2=0           ;YES, set P2 to 0
AND SEND"Good-bye" ;Print third message just once
ELSE                ;No, do nothing
END                 ;End of PLC program
                    ;The host computer screen should
                    ;show three message lines:
                    ;Hello/how-are-you/good-bye
```

EXAMPLE #33

GLUE DISPENSER PROGRAM (GLUE):

This program is used to put glue on four straight lines with an arc at the end of each line. The part is rotated 90 degrees each time glue is applied to one side.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A0z200              ;Clear buffer and allocate 200
                    ;program lines in the buffer
I150                ;Enable Y axis
F1024               ;Set feedrate to 100% of i06 value
FOR4                ;Start loop 4 cycles
X34000Y10552        ;Move glue gun to starting
                    ;position
SET01               ;Turn on M01 output which starts
                    ;dispensing glue
DWE256              ;Wait for 256 msec.
V750                ;Move Y axis incrementally
U320                ;Move X axis incrementally
U-2638V21360        ;Interpolated, simultaneous move
CIR1                ;Circular mode, CCW (arc setup)
R=2833              ;Radius
U-2925V2565         ;Arc destination, make the circular
                    ;move
RESET01             ;Turn off glue
LIN                 ;Linear mode, cancel circular mode
U-480               ;Wipe off glue tip by moving X
                    ;axis
SET02               ;Turn on chuck (rotate chuck 90
                    ;degrees)
DWELL51             ;Wait for 50 msec.
RESET02             ;Turn off chuck (stop chuck)
X34000Y9732         ;Move back to starting position
DWE20               ;Wait for 20 msec.
NEXT                ;End of loop, do this 4 times then
0Y0                 ;move X and Y back to 0 position
END                 ;End of program
A0PURGE             ;Clear existing PLC program and
                    ;start new PLC
IF M11=0            ;Check if Emergency Stop is on
THEN M01=0          ;True, turn off glue
AND M02=0           ;Turn off chuck
AND I03=0           ;Disable COMMUNICATIONS handshake
AND"Q"              ;Quit program
AND I15=1           ;Disable Y axis
AND"b"              ;Set pointer to top of program
AND"h"              ;Home X axis only
AND P3=1            ;Set flag 3
AND I03=1           ;Enable COMMUNICATIONS handshake

IF M13=0            ;Check if X home flag is on
AND P3>0
THEN I03=0          ;True, disable COMMUNICATIONS handshake
AND "q"             ;Quit current move
AND I15=0           ;Enable Y axis
AND"h"              ;Home both X and Y axis
AND P3=0            ;Resets flag 3
AND I03=1           ;Enable COMMUNICATIONS handshake

IF M14=0            ;Check if index chuck is on
THEN M02=1          ;True, rotate chuck
AND T3=2390         ;This cause the index chuck to
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ;stay on for 51 msec.
AND P5=1            ;Set flag 5
IF T3=0 AND P5>0    ;Check if T0 is finished with
                    ;countdown
THEN M02=0          ;True, turn off index chuck
AND P5=0            ;Reset flag 5
END                 ;End of PLC program
```

EXAMPLE #34 (Note: This program uses the obsolete method of diode-isolated thumbwheels. To see how to use the ACC-18 thumbwheel multiplexer board, refer to Examples 52 and 53 - page 383.)

THUMBWHEEL TEST (TESTHUMB.SMC):

This program is used to test all 16 thumbwheel switches, M02 to M05 outputs (LEDs), and MI1 to MI4 inputs (switches). To run this test, load "Testhumb" and issue a "R" Run Command. Observe the screen. You will see on the screen a message, "THUMBWHEEL #", then the first and the second number. The first number represents the order number of the thumbwheel switch. The second number represents the value number you set on the thumbwheel switch. A non-existing thumbwheel will show a value of 0. After all 16 thumbwheel switches are tested, it will test the M02 to M05 outputs. If you have LEDs connected to M02, M03, M04, and M05, you'll see the LEDs light up, one at a time. Finally, the last test is to test the MI1 to MI4 input switches. The SMCC will send a message to the screen to tell you to toggle the MI switch. After you toggle the MI switch you will see a message "OK" print on the screen. (If you don't have the MI switch available, you can bypass it by pressing any alpha key or numeric key on your computer's keyboard.) If you press SHIFT, CTRL, ALT, ENTER, or any special key, the program will ignore it. After all the tests are done, you will see the final message printed on the screen.

```
A0z300          ;Clear buffer, assign 300 buffer lines
I621            ;Enable PLC
SETC25=ffb8     ;Address of parallel port (data in)
SETC26=feec     ;Address of the pointer, SMCC uses
                ;this address to scan the data input
                ;from computer's keyboard
RESP            ;Reset variable parameters
RES02           ;Reset M-functions
RES03
RES04
RES05
RES30
RES31
RES32
RES33
mmffb9,fe       ;Enable switch #1 & switch #2
GOS03           ;Call Subroutine 3
mmffb9,fd       ;Enable switch #3 & switch #4
GOS03
mmffb9,fb       ;Enable switch #5 & switch #6
GOS03
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
mmffb9,f7       ;Enable switch #7 & switch #8
GOS03
mmffb9,ef       ;Enable switch #9 & switch #10
GOS03
mmffb9,df       ;Enable switch #11 & switch #12
GOS03
mmffb9,bf       ;Enable switch #13 & switch #14
GOS03
mmffb9,7f       ;Enable switch #15 & switch #16
GOS03
DWE1024         ;Wait 1 second
SET02           ;Test M02, on
DWE1024
SET03           ;Test M03, on
DWE1024
SET04           ;Test M04, on
DWE1024
SET05           ;Test M05, on
DWE1024
RES05           ;Turn off M05
DWE1024
RES04           ;Turn off M04
DWE1024
RES03           ;Turn off M03
DWE1024
RES02           ;Turn off M02
DWE1024
SET P11=1       ;Test MI1 to MI4 input switch
SET30           ;The SMCC will ask you to toggle the
STOP            ;MI switch; after you toggled the MI
SET P12=1       ;switch you will see a message "OK"
SET31           ;print on the screen.
STOP
SET P13=1
SET32
STOP
SET P14=1
SET33
STOP
SET P8=1
END

                ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 00
L00CAS25,16L    ;Subroutine 00, reads thumbwheel switch
SETP1=15        ;and determines value of P1
SETP1=14        ;CAS25 tells SMCC to read data at
SETP1=13        ;address defined by C25, in this case
SETP1=12        ;ffb8,"L" tells SMCC to read the 4
SETP1=11        ;lower bits (least significant nibble)
SETP1=10        ;and the 16 defines a 16 step table
SETP1=9         ;of commands, one of which is executed
SETP1=8         ;based on the value read at ffb8
SETP1=7         ;
SETP1=6         ;Note that the thumbwheel switch is
SETP1=5         ;being read "backward"; this is in
SETP1=4         ;order to induce an effective
```

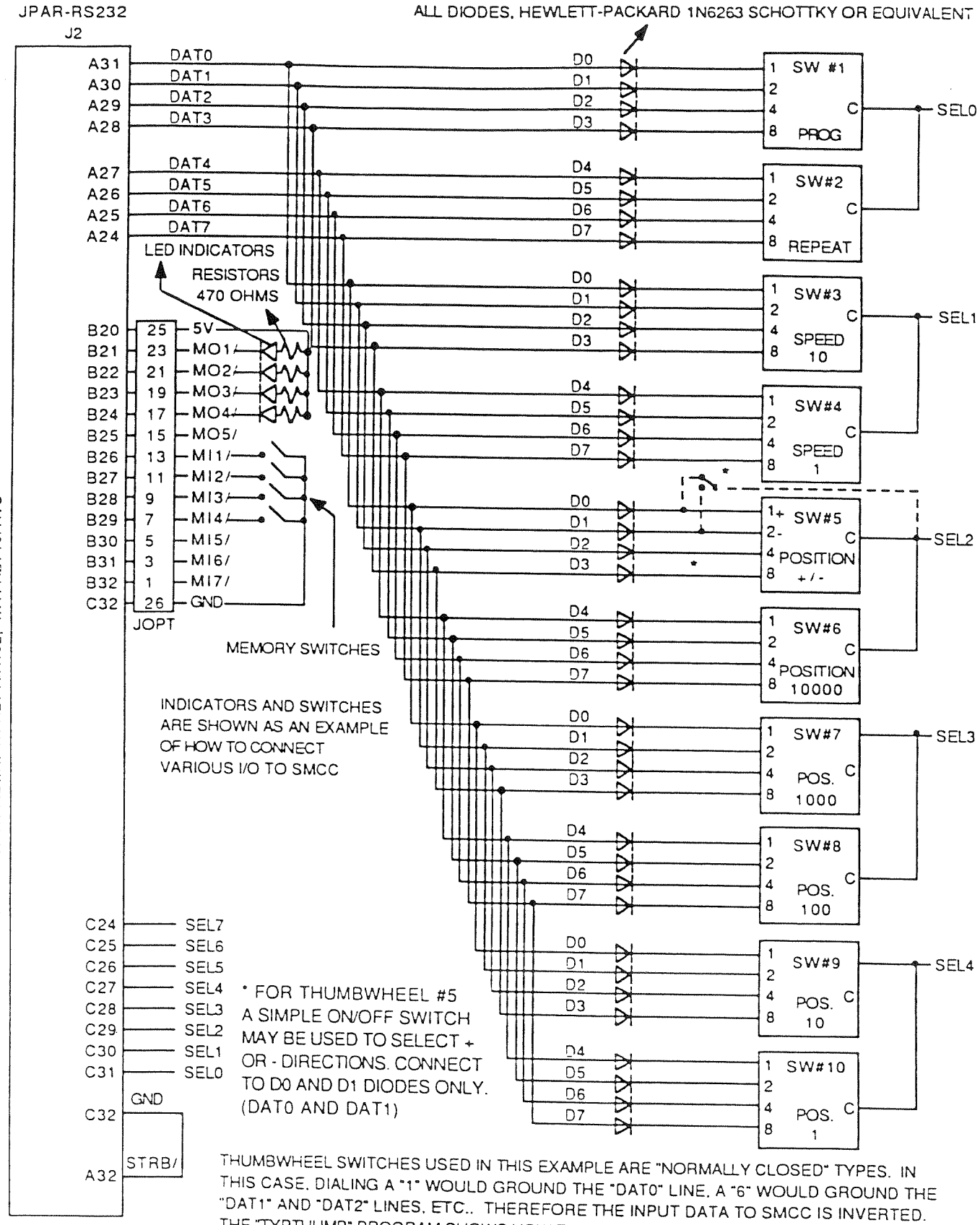

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SETP1=3          ;polarity reversal of the thumbwheel
SETP1=2          ;switch's bits.  The actual thumbwheel
SETP1=1          ;used, outputs "low true" data and
SETP1=0          ;therefore must be inverted.
RET
                 ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 01
L01CAS25,16H     ;Subroutine 01, same as Subroutine 00
SETP1=0          ;but the "H" tells SMCC to read the 4
SETP1=0          ;higher bits (most significant nibble).
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=9
SETP1=8
SETP1=7
SETP1=6
SETP1=5
SETP1=4
SETP1=3
SETP1=2
SETP1=1
SETP1=0
RET
                 ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 02
L02              ;Subroutine 02
SET P2=P2+1      ;Thumbwheel switch incremental.
SET P10=1        ;Set delay time so the SMCC can print
DWE700           ;multiple message on screen.
SET P3=1         ;The delay of 700 msec is to prevent
DWE700           ;screen overwrite.
SET P4=1
DWE700
                 ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 03
RET
L03              ;Subroutine 03
GOS00            ;Call Subroutine 0
GOS02            ;Call Subroutine 2
GOS01            ;Call Subroutine 1
GOS02            ;Call Subroutine 2
RET
A0PURGE                  ;Clear PLC_ _ start of PLC
IF P10>0 AND P10<2       ;program
THEN P10=0               ;Check if P10 is 1
AND SEND"THUMBWHEEL #"   ;True, print message
ELSE                     ;False, do nothing
IF P3>0 AND P3<2         ;Check if P3 is 1
THEN P3=0                ;True, print message
AND"P2"                  ;This causes SMCC to print the
                         ;value of P2
                         ;Note that P2 is the order
                         ;number of the thumbwheel switch.
ELSE                     ;False, do nothing
IF P4>0 AND P4<2         ;Check if P4 is 1
THEN P4=0                ;True, print value of P1
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
AND"P1"                    ;This causes the SMCC to print
                           ;the value of P1.
AND P1=0                   ;Note that P1 is the number
                           ;set on the thumbwheel switch.
ELSE                       ;False, do nothing
IF P11>0 AND M11=0         ;Check if MI1 switch is
OR C26=1 AND P11>0         ;activated or any input is
THEN P11=0                 ;received from the computer's
                           ;keyboard.
AND SEND"OK"               ;True, print message
AND P15=1
ELSE                       ;False do nothing
                           ;Note that C26 (address of the
                           ;pointer) is set to 1.  So the
                           ;SMCC knows that it only
                           ;received one single number
                           ;or one character from the
                           ;computer's keyboard.
                           ;When the SMCC detects input
                           ;from the computer's keyboard
                           ;it aborts testing that MI
                           ;switch and goes on to next MI
                           ;switch.
IF P12>0 AND M12=0         ;Check if MI2 switch is
OR C26=1 AND P12>0         ;activated or any input is
THEN P12=0                 ;received from the computer's
                           ;keyboard.
AND SEND"OK"               ;True, print message
AND P15=1
ELSE                       ;False, do nothing
IF P13>0 AND M13=0         ;Check if MI3 switch is
OR C26=1 AND P13>0         ;activated or any input is
THEN P13=0                 ;received from the computer's
                           ;keyboard.
AND SEND"OK"               ;True, print message
AND P15=1
ELSE                       ;False, do nothing
IF P14>0 AND M14=0         ;Check if MI4 switch is
OR C26=1 AND P14>0         ;activated or any input is
THEN P14=0                 ;received from the computer's
                           ;keyboard.
AND SEND"OK"               ;True, print message
AND P15=1
ELSE                       ;False, do nothing
IF P15>0 AND P15<2         ;Check if P15 is 1
THEN P15=0                 ;True, send "R", Run Command
AND"R"                     ;The "R", Run Command is
                           ;needed in this case because
                           ;the program is forced to
                           ;stop when it tests the MI
                           ;switch, so it can wait for the
                           ;MI switch to toggle.
ELSE
IF M30=1                   ;Check flag to print the
THEN M30=0                 ;message on the screen.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



THUMBWHEEL SWITCHES USED IN THIS EXAMPLE ARE "NORMALLY CLOSED" TYPES. IN THIS CASE, DIALING A "1" WOULD GROUND THE "DAT0" LINE, A "6" WOULD GROUND THE "DAT1" AND "DAT2" LINES, ETC.. THEREFORE THE INPUT DATA TO SMCC IS INVERTED. THE "TYPTHUMB" PROGRAM SHOWS HOW TO USE INVERTED DATA.

TYPICAL USE OF THUMBWHEELS (THUMBWHEEL INPUT CONN.)
(SEE PROGRAM FOR EXPLANATION)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
AND SEND"TOGGLE_MI1"
ELSE
IF M31=1
THEN M31=0
AND SEND"TOGGLE_MI2"
ELSE
IF M32=1
THEN M32=0
AND SEND"TOGGLE_MI3"

IF M33=1
THEN M33=0
AND SEND"TOGGLE_MI4"
ELSE
IF P8>0 AND P8<2        ;Check flag to print the
THEN P8=0               ;final message.
AND SEND"TEST_IS_COMPLETED,Thank you"
ELSE
END
```

EXAMPLE #35 (Note: This program uses the obsolete method of diode-isolated thumbwheels. To see how to use the ACC-18 thumbwheel multiplexer board, refer to Examples 52 and 53 - page 383.)

STRAIGHT MOVE USING THUMBWHEELS (XYMOVE) USE OF LOWER-CASE "H" AND "L" IN CAS COMMAND:

This program uses the thumbwheel switches to control position, dwell time, and feedrate. This program allows you to select X axis, Y axis, or X and Y axes moves by using the "Select" thumbwheel. One is for X axis, two is for Y axis, and three is for X and Y axes.

After a move is completed, the program waits for the MI1 switch to toggle then makes a move back to the zero position. Note that the distance set on the "Position" thumbwheel is the size of the move; also you will see a message:

"X_axis_only", "Y_axis_only", or "X_and_Y_axis"

display on the screen each time the program is run. The "X_axis_only" message tells that SMCC is running the X axis. The "Y_axis_only" message tells that SMCC is running Y axis. The "X_and_Y_axis" message tells that SMCC is running both X and Y axes.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
A0z300          ;Clear buffer
I1280           ;Set move delay to 80 msec, needed for
                ;allowing sufficient calculation time
I621            ;Enable PLC
SETC25=ffb8     ;Address of parallel port (data in)
RES01           ;Reset M01
RES02           ;Reset M02
RES03           ;Reset M03
RES04           ;Reset M04
RES05           ;Reset M05
SETP4=0         ;Set P4 to value 0
SETP5=0         ;Set P5 to value 0
SETP6=0         ;Set P6 to value 0
SETP7=11        ;Set P7 to value 11
SETP8=0         ;Set P8 to value 0
SETP10=0        ;Set P10 to value 0
SETP11=1        ;Set P11 to value 1
SETP12=2        ;Set P12 to value 2
SETP13=110      ;Set P13 to value 110
SETP14=0        ;Set P14 to value 0
mmffb9,fe       ;Enable "dwell" switch and "select" switch
SETP2=1         ;Set P2 to value 1
SETP3=10        ;Set P3 to value 10
GOS03           ;Call Subroutine 03 (read "select" switch)
SETP8=(P6*P11);Calculate P8
SETP6=0         ;Set P6 to value 0
GOS04           ;Call Subroutine 04 (read "dwell" switch)
SETP14=(P6*P13);Calculate P14 (dwell time)
DWE500          ;Wait for 500 msec
SETP6=0         ;Set P6 to value 0
mmffb9,fd       ;Enable "speed" switch
GOS03           ;Call Subroutine 03
GOS04           ;Call Subroutine 04
SETP10=(P6*P7);Calculate P10 (feedrate value)
SETP6=0         ;Set P6 to value 0
F(P10)          ;Set feedrate (depends on value of P10)
GOS00IF(M02=1AND M03=0);Select X axis
GOS06IF(M03=1AND M02=0);Select Y axis
GOS07IF(M02=1ANDM03=1) ;Select X and Y axis
SETP10=0        ;Set P10 to value 0
SETP14=0        ;Set P14 to value 0
END             ;End of program
                ;_____ 00
L00SETP6=0      ;Subroutine 00(program #1)
LIN             ;Linear mode
GOS01           ;Call Subroutine 01
SETP9=1         ;Set flag P9
DWE100          ;Wait 100 msec
U(P6)           ;Incremental move X
DWE(P14)        ;Wait
GOT-01IF(M11=1);Wait until MI1 switch toggles
X0              ;Move back to 0 position
DWE(P14)        ;Wait
GOS02           ;Call Subroutine 02
RET
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                ;_____________________________01
L01mmffb9,ef    ;Subroutine 01 (read distance switch)
                ;Enable switch #9 & #10

GOS03           ;Call Subroutine 03
GOS04           ;Call Subroutine 04
mmffb9,f7       ;Enable switch #7 & #8
SETP2=100       ;Set P2 to value 100
GOS03           ;Call Subroutine 03
SETP3=1000      ;Set P3 to value 1000
GOS04           ;Call Subroutine 04
mmffb9,fb       ;Enable switch #5 & #6
SETP2=10000     ;Set P2 to value 10000
GOS03           ;Call Subroutine 03
GOS05           ;Call Subroutine 05
SETP6=(P6*P1)   ;Calculate P6 (total distance)
RET
                ;_____________________________02
L02SETP10=0     ;Subroutine 02 (initialize variable
                ;parameter)
                ;Set P10 to value 0
SETP2=1         ;Set P2 to value 1
SETP3=10        ;Set P3 to value 10
SETP4=0         ;Set P4 to value 0
SETP5=0         ;Set P5 to value 0
SETP6=0         ;Set P6 to value 0
DWE500          ;Wait 500 msec
RET
                ;_____________________________03
L03CAS25,10h    ;Subroutine 03, reads thumbwheel switch
                ;and determines value of P1
                ;CAS25 tells SMCC to read data at
                ;address defined by C25, in this case
                ;ffb8, "h" tells SMCC to read the 4
                ;higher bits (most significant nibble)
                ;and the 10 defines a 10 step table
                ;of moves.
SETP1=0         ;Note that the lower-case "h" or lower-
SETP1=1         ;case "l", as opposed to "H" or "L",
SETP1=2         ;cause the SMCC to read the thumbwheel
SETP1=3         ;and inverting the data before using
SETP1=4         ;it to make the table jump.
SETP1=5         ;This permits the use of thumbwheels
SETP1=6         ;which invert data to be used
SETP1=7         ;without having to change the program
SETP1=8         ;to read "upside-down" data.
SETP1=9         ;The lower-case "h" and "l" Commands
                ;are available on software version
                ;2.35, 10/6/88 for Standard and 2.54P,
                ;10/6/88 for Parabolic.
SETP4=(P1*P2)   ;Calculate P4
SETP6=(P6+P4)   ;Calculate P6
RET
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                ;_____________________________ 04
L04CAS25,101    ;Subroutine 04, same as Subroutine 03
                ;but the lower-case "l" tells SMCC to
                ;read 4 lower bits (least significant
                ;nibble) by reading the thumbwheel and
                ;inverting the data before using it to
                ;make the table jump.
SETP1=0
SETP1=1
SETP1=2
SETP1=3
SETP1=4
SETP1=5
SETP1=6
SETP1=7
SETP1=8
SETP1=9
SETP5=(P1*P3)   ;Calculate P5
SETP6=(P6+P5)   ;Calculate P6
RET
                ;_____________________________ 05
L05CAS25,101    ;Subroutine 05, same as Subroutine
                ;04 but tells SMCC to determine
                ;positive or negative move.
SETP1=0
SETP1=1         ;Positive
SETP1=-1        ;Negative
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=0
RET
                ;_____________________________ 06
L06GOS01        ;Subroutine 06 (Y axis)
SETP9=2         ;Set flag P9
DWE100          :Wait 100 msec
V(P6)           ;Incrementally move Y
DWE(P14)        ;Wait
GOT-01IF(M11=1);Wait until MI1 switch toggles
Y0              ;Move Y back to 0 position
DWE(P14)        ;Wait
GOS02           ;Call Subroutine 2
RET
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 07
L07GOS01        ;Subroutine 1 (X and Y axis)
SETP9=3         ;Set flag P9
DWE100          ;Wait 100 msec
U(P6)V(P6)      ;Incrementally move X and Y
DWE(P14)
GOT-01IF(M11=1);Wait until MI1 switch toggles
X0Y0            ;Move X and Y back to 0 position
DWE(P14)        ;Wait
GOS02           ;Call Subroutine 2
RET
A0PUR           ;Clear PLC   start of PLC program
IFP8>0          ;Check if P8 (select switch) is 1 (X axis)
ANDP8<2
THENM02=1       ;true, turn on M02
ANDM03=0        ;Turn off M03
IFP8>1          ;Check if P8 (select switch) is 2 (Y axis)
ANDP8<3
THENM03=1       ;True, turn on M03
ANDM02=0        ;Turn off M02
IF P8>2         ;Check if P8 (select switch) is 3 (X and Y
ANDP8<4         ;axis)
THEN M02=1      ;True, turn on M02
AND M03=1       ;Turn on M03
IFP9>0          ;Check if flag P9 is 1
ANDP9<2
THENP9=0        ;True, reset flag P9
AND SEND"X_axis_only";Send a message to the screen
IFP9>1          ;Check if flag P9 is 2
ANDP9<3
THENP9=0        ;True, reset flag P9
AND SEND"Y_axis_only";Send a message to the screen
IFP9>2          ;Check if flag P9 is 3
ANDP9<4
THENP9=0        ;True, reset flag P9
AND SEND"X_and_Y_axis";Send a message to screen
END
```

EXAMPLE #36 (Note: This program uses the obsolete method of diode-isolated thumbwheels. To see how to use the ACC-18 thumbwheel multiplexer board, refer to Examples 52 and 53 - page 383.)

TYPICAL USE OF THUMBWHEELS (TYPTHUM1) USE OF LOWER-CASE "H" AND "L" IN CAS COMMAND:

This program uses the thumbwheel switches to control loop, position and feedrate. See attached drawing for actual connection of thumbwheel switches. To run this program, first set the program switch to one or two. If set other than one or two, program will do nothing. When program switch is set to one, then this program will run program #1; if program switch is set to two, then this program will run program #2. Program #1 allows you to set position feedrate and repeat loop. It reads the repeat switch first then the speed and distance switch. After it finishes reading the distance switch, it will move X to the

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

position set on the distance switch then wait for MI1 input switch to be activated, then it will move X back to 0 position.

NOTE: When the repeat switch is set to 0, it tells the program not to repeat. If it is set to 1, it tells the program to run one more time and so on depending on the selected repeat switch number.

Program #2 is actually two programs in one. In this program the thumbwheel switches function the same as in program #1. First it runs the circle program. (You will see the message "CIRCLE_PROGRAM" print on screen.) In the circle program, the position you set on the DISTANCE switch is the size of the circle's radius. After the circle program is done, it runs the square program. (You will see the message "SQUARE_PROGRAM" print on screen.) In the square program, the position you set on the distance switch is the size of the square's sides. After the square is done, it does a diagonal line then returns to the zero position.

```
A0z300          ;Clear buffer
I1280           ;Set move delay to 80 msec, needed for
                ;allowing sufficient calculation time
I621            ;Enable PLC
SETC25=ffb8     ;Address of parallel port (data in)
SET P14=0       ;Set P14 to value 0
RES01           ;Reset M01
RES02           ;Reset M02
RES03           ;Reset M03
RES04           ;Reset M04
RES05           ;Reset M05
SETP4=0         ;Set P4 to value 0
SETP5=0         ;Set P5 to value 0
SETP6=0         ;Set P6 to value 0
SETP7=11        ;Set P7 to value 11
SETP8=0         ;Set P8 to value 0
SETP10=0        ;Set P10 to value 0
SETP11=1        ;Set P11 to value 1
SETP12=2        ;Set P12 to value 2
mmffb9,fe       ;Enable program and repeat switch
SETP2=1         ;Set P2 to value 1
SETP3=10        ;Set P3 to value 10
GOS03           ;Call Subroutine 03 (read repeat switch)
SETP8=(P6*P11);Calculate P8
SETP8=P8+1      ;Calculate P8
SETP6=0         ;Set P6 to value 0
GOS04           ;Call Subroutine 04 (read program switch)
SETP14=(P6*P11);Calculate P14 (program select)
DWE500          ;Wait for 500 msec
SETP6=0         ;Set P6 to value 0
FOR(P8)         ;For loop (depends on value of P8)
mmffb9,fd       ;Enable speed switch
GOS03           ;Call Subroutine 03
GOS04           ;Call Subroutine 04
SETP10=(P6*P7);Calculate P10 (feedrate value)
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SETP6=0          ;Set P6 to value 0
F(P10)           ;Set feedrate (depends on value of P10)
GOS00IF(M04=1);Select program #1
GOS06IF(M05=1);Select program #2
SETP10=0         ;Set P10 to value 0
NEX              ;End of loop
SETP14=0         ;Set P14 to value 0
END              ;End of program
                 ;_____ _____ _____ _____ _____ 00
L00SETP6=0       ;Subroutine 00 (program #1)
LIN              ;Linear mode
GOS01            ;Call Subroutine 01
U(P6)V(P6)       ;Incremental move (depends on value of
                 ;P6)
DWE1024          ;Wait 1 second
SETP15=1         ;Set P15 to value 1
STO              ;Stop (wait until MI1 switch toggles)
                 ;See PLC program
SETP15=0         ;Set P15 to value 0
X0Y0             ;Move back to 0 position
DWE1024          ;Wait 1 second
GOS02            ;Call Subroutine 02
RET
                 ;_____ _____ _____ _____ _____ 01
L01mmffb9,ef     ;Subroutine 01 (read distance switch)
                 ;Enable switch #9 & #10

GOS03            ;Call Subroutine 03
GOS04            ;Call Subroutine 04
mmffb9,f7        ;Enable switch #7 & #8
SETP2=100        ;Set P2 to value 100
GOS03            ;Call Subroutine 03
SETP3=1000       ;Set P3 to value 1000
GOS04            ;Call Subroutine 04
mmffb9,fb        ;Enable switches #5 & #6
SETP2=10000      ;Set P2 to value 10000
GOS03            ;Call Subroutine 03
GOS05            ;Call Subroutine 05
SETP6=(P6*P1)    ;Calculate P6 (total distance)
RET
                 ;_____ _____ _____ _____ _____ 02
L02SETP10=0      ;Subroutine 02 (initialize variable
                 ;parameter)
                 ;Set P10 to value 0
SETP2=1          ;Set P2 to value 1
SETP3=10         ;Set P3 to value 10
SETP4=0          ;Set P4 to value 0
SETP5=0          ;Set P5 to value 0
SETP6=0          ;Set P6 to value 0
DWE500           ;Wait 500 msec
RET
                 ;_____ _____ _____ _____ _____ 03
L03CAS25,10h     ;Subroutine 03, reads thumbwheel switch
                 ;and determines value of P1.
                 ;CAS25 tells SMCC to read data at
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                ;address defined by C25, in this case
                ;ffb8, "h" tells SMCC to read the 4
                ;higher bits (most significant nibble),
                ;and the 10 defines a 10 step table
                ;of moves.
SETP1=0         ;Note that the lower-case "h" or lower-
SETP1=1         ;case "l", as opposed to "H" or "L",
SETP1=2         ;cause the SMCC to read the thumbwheel
SETP1=3         ;and inverts the data before using
SETP1=4         ;it to make the table jump.
SETP1=5         ;This permits the use of thumbwheels
SETP1=6         ;which invert the data to be used
SETP1=7         ;without having to change the program
SETP1=8         ;to read "upside-down" data.
SETP1=9         ;The lower-case "h" and "l" Commands
                ;are available on software version
                ;2.35, 10/6/88 for Standard and 2.54P,
                ;10/6/88 for Parabolic.
SETP4=(P1*P2)   ;Calculate P4

SETP6=(P6+P4)   ;Calculate P6
RET
                ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 04
L04CAS25,101    ;Subroutine 04, same as Subroutine 03
                ;but the lower-case "l" tells SMCC to
                ;read 4 lower bits (least significant
                ;nibble) by reading the thumbwheel and
                ;inverting the data before using it to
                ;make the table jump.
SETP1=0
SETP1=1
SETP1=2
SETP1=3
SETP1=4
SETP1=5
SETP1=6
SETP1=7
SETP1=8
SETP1=9
SETP5=(P1*P3)   ;Calculate P5
SETP6=(P6+P5)   ;Calculate P6
RET
                ;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 05
L05CAS25,101    ;Subroutine 05, same as Subroutine
                ;04 but tells SMCC to determine
                ;positive or negative move.
SETP1=0
SETP1=1         ;Positive
SETP1=-1        ;Negative
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=0
SETP1=0
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SETP1=0
RET
                ;_____ _____ _____ _____ 06
L06GOS01        ;Subroutine 06 (Program #2)
                ;Call Subroutine 01
SET01           ;Turn on M01
CIR0            ;Circle mode
R=(P6)          ;Set radius to value of P6
t50             ;Set accel/decel time to 50 msec
X(P6*P12)       ;Move X 1 diameter
DWE1024         ;Wait 1 second
X0              ;Move X back to 0 position
DWE1024         ;Wait 1 second
SET02           ;Turn on M02
LIN             ;Linear mode
t200            ;Set accel/decel time to 200 msec
X(P6)           ;Move X bases on P6
DWE1024         ;Wait 1 second
Y(P6)           ;Move Y bases on P6
DWE1024         ;Wait 1 second
X0              ;Move X back to 0 position
DWE1024         ;Wait 1 second
Y0              ;Move Y back to 0 position
DWE1024         ;Wait 1 second
X(P6)Y(P6)      ;Move X and Y base on P6
DWE1024         ;Wait 1 second
X0Y0            ;Move X and Y back to 0 position
DWE1024         ;Wait 1 second
GOS02           ;Call Subroutine 02
RET
A0PUR           ;Clear PLC_ _start of PLC program
IFP15>0         ;Check if P15 is not 0
ANDM11=0        ;and MI1 switch is activated
THEN"R"         ;True, send "R", Run Command
ELSE            ;False, do nothing
IFM01=1         ;Check if M01 is on
THENM01=0       ;True, turn off M01
ANDSEND"CIRCLE_PROGRAM"  ;Print message on screen
ELSE            ;False, do nothing
IFM02=1         ;Check if M02 is on
THEN M02=0      ;True, turn off M02
ANDSEND"SQUARE_PROGRAM"  ;Print message on screen
ELSE            ;False, do nothing
IFP14>9         ;Check if P14 is 10 (1 on program switch)
ANDP14<11
THENM04=1       ;True, turn on M04
ANDM05=0        ;Turn off M05
ELSE            ;False, do nothing
IFP14>19        ;Check if P14 is 20 (2 on program switch)
ANDP14<21
THENM05=1       ;True, turn on M05
ANDM04=0        ;Turn off M04
ELSE            ;False, do nothing
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #37

PROCESSOR ADDRESS DEFINITION (BITS AND BYTES):

By using Cnn and Mnn definitions, any meaningful address can be defined and used in the program. For example, corresponding definitions for M01 (M-function output #1, bit 5 of address ff01) is:

```
                          ;Definition by Direct Commands
  C01 = ff01              ;Define C01 variable as SMCC memory
                          ;address FF01
  M01  =  C01.5           ;M01 is defined as bit 5 of that address

                          ;Definition Inside Program
  SET C01 = ff01
                          ;Define C01 variable as SMCC memory
                          ;address FF01
  SET M01 = C01.5
                          ;M01 is defined as bit 5 of that
                          ;address
```

As you can see, you have to know memory addresses and bit positions of individual functions to define them (see Software Reference Manual for details). Note that since C01-C05, M01-M05, and M11-M17 are already defined internally for M-functions, it is recommended not to use these definitions so as to eliminate confusion.

EXAMPLE #38

CONDITIONAL EXECUTION:

To change program execution sequence, the following six commands may be used.

```
GOTO+ nn (IF (NOT) Condition) -- Go to forward nn program lines
GOTO- nn (IF (NOT) Condition) -- Go to backward nn program lines
GOTO  nn (IF (NOT) Condition) -- Go to label nn and end program
                                 when RETURN is executed
GOSUB nn (IF (NOT) Condition) -- Call Subroutine nn and return
SET   nn (IF (NOT) Condition) -- M-function Output Control
RESET nn (IF (NOT) Condition)
```

Logic is determined by conditions expressed after an IF or IF NOT statement. See the following syntax examples:

```
Command IFNOT 01 > 128      ;If C01 is not greater than 128
Command IF 01 * 4           ;Bit-wise logical AND
                            ;If bit 2 of C01 is 1
Command IF 01 + 16          ;Bit-wise logical OR
Command IF (M01=0 AND M02 =1)
Command IFNOT (P04 >= X)    ;If P04 Value is >= X position
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The following example demonstrates selection of a subroutine execution depending on the M-function or current axis position.

```
C01 = ffbc              ;Define C01 as machine input address
M02 = C01.0             ;Define M02 as machine input #1 (bit 0)
C03 = C01.1             ;Define M03 as machine input #2 (bit 1)
P11 = 10000             ;Define variable P11 as 10000
GOSUB 02 IF (M02=1) AND (M03=0)  ;Execute Subroutine 02
                                 ;(Defined by L02), if MI1 is
                                 ;HIGH and MI2 is LOW
GOTO+ 05 IF NOT (P11 > X)
                                 ;If present position (X axis)
                                 ;is less than 10000, skip
                                 ;5 Instructions
```

EXAMPLE #39

USING M-FUNCTION INPUTS AS CONTROL PANEL INPUTS:

This PLC program uses M-function inputs M15 (JMach, C28), M16 (JMach, C27) and M17 (JMach, C26) as "Run", "Z" and "hX" (home X axis only) Commands.

TO ACTIVATE ANY OF THESE INPUTS, GROUND THE INPUT, OTHERWISE, leave it open.

```
A0PURGE          ;Clear all existing PLC programs
IF M15=1         ;Check if M15 switch is toggled
THEN IF P1<1     ;Check if send flag 1 is set
AND P1>-1
THEN"R"          ;True, starts continuous program
                 ;execution
AND P1=1         ;Reset send flag 1
ELSE             ;
ELSE P1=0        ;Set send flag 1
IF M16=1         ;Check if M16 switch is toggled
THEN IF P2<1     ;Check if send flag 2 is set
AND P2>-1
THEN"Z"          ;True, set X and Y position to 0
AND P2=1         ;Reset send flag 2
ELSE             ;
ELSE P2=0        ;Set send flag 2
IF M17=1         ;Check if M17 switch is toggled
THEN IF P3<1     ;Check if send flag 3 is set
AND P3>-1
THEN P3=1        ;True, reset send flag 3
AND"hX"          ;Starts X axis on home cycle
ELSE             ;
ELSE P3=0        ;Set flag 3
END              ;End of PLC program
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #40

JOYSTICK CONTROL:

Connect CW of the X axis POT and CW of the Y axis to 5V, CCW of X axis POT and Y axis to GND, and wiper of X axis to C1(JMACH) and wiper of Y axis to A1(JMACH). Then connect the two wires of the joystick button, one wire to C18(JMACH), and the other wire to C8(JMACH). To jog the X motor, push and hold the joystick button then move the joystick horizontally, right is CW and left is CCW. To jog the Y motor, push and hold the joystick button then move joystick vertically; up is CW and down is CCW. The joystick is disabled when resting center or the value of the input from the joystick is within the range of the offset value.

```
A0z50              ;Address card, clear buffer
C16=fe90           ;Address of RAW ATD #2 INPUT (X axis)
C17=fe91           ;Address of RAW ATD #3 INPUT (Y axis)
P3=124             ;Offset value for - JOG
P4=129             ;Offset value for + JOG
P5=0               ;Flag to send "q"
A0PURGE            ;Clear existing PLC program
IF M15=1           ;Check if joystick button is on
THEN I14=3         ;On, enable joystick
AND P1=C16         ;Move value of C16 to P1 variable
AND P2=C17         ;Move value of C17 to P2 variable
AND P5=1           ;Enable "q" flag
AND IF P1<(P3)     ;Check if P1 and P2 are not in range
OR P1>(P4)         ;of offset value
OR P2<(P3)         ;
OR P2>(P4)         ;
THEN I14=3         ;True, enable joystick
ELSE I14=0         ;False, disable joystick
AND IF P5>0        ;Check if "q" flag is enabled
THEN "q"           ;True, send "q" Command
AND P5=0           ;Disable "q" flag
ELSE               ;
ELSE I14=0         ;
AND IF P5>0        ;
THEN "q"           ;
AND P5=0           ;
ELSE               ;
END                ;End of PLC program
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #41

HANDWHEEL PROFILE GENERATOR:

This program uses the X handwheel as input time base; the Y handwheel could also be used as input time base by changing i16 to two. Note that i27 is set to five and i47 to 32768, based on this formula :

5 counts/msec = 32768*i27/i47

See % Command in SMCC Manual for detailed explanation.

Note that for non-parabolic SMCC, i47 cannot be used and the formula is now:

4.997 counts/msec = 32768/6554

when i27 is chosen to be 6554. Other than the small error this creates, operation is identical.

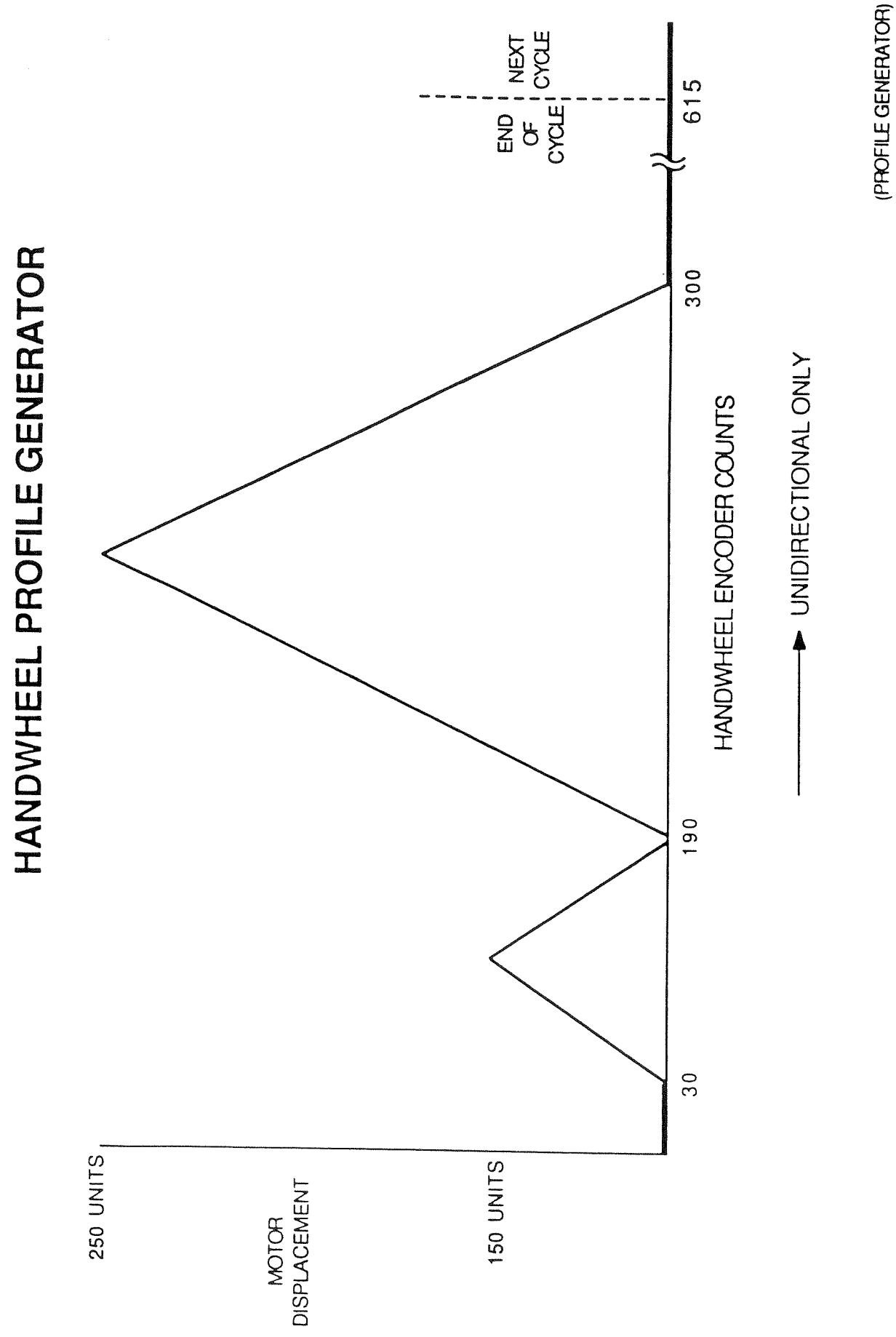
The purpose of this program is to generate an arbitrary motion profile on a motor shaft (the X axis), which is based on the handwheel encoder counts. In this particular example, the relationship between the handwheel encoder and the motor position is as follow:

a) Handwheel moves from 0 to 30 counts - X axis does not move
b) Handwheel moves from 30 to 70 counts - X axis moves 150 out and back
c) Handwheel moves from 70 to 300 counts - X axis moves 250 out and back
d) Handwheel moves from 300 to 615 counts - X axis does not move
e) Cycle repeats

Please note that the cycle will run in one direction only, if handwheel reverses the motor will not reverse direction.

```
A0z50       ;Clear buffer, reserve 50 lines
FOR0        ;Start forever loop
t1          ;Set accel/decel time to 1 msec
            ;Keep accel time as small as possible
            ;in order to minimize path distortion due
            ;to accel time.
T6          ;Set move time of 6 msec (6m sec X 5 counts = 30
            ;counts of X handwheel)
U0          ;Don't move motor
T7          ;Set move time to 7 msec (35 counts of X
            ;handwheel)
U150        ;Incrementally move motor to 150
U-150       ;Incrementally move motor back to 0
T20         ;Set move time to 20 msec (100 counts of X
            ;handwheel)
U250        ;Incrementally move motor to 250
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
U-250       ;Incrementally move motor back to 0
T63         ;Set move time to 63 msec (315 counts of X
            ;handwheel)
U0          ;Don't move motor
NEXT        ;End of loop, return to top of program
i275        ;Set parameters as required
i4732768    ;
i483        ;
i163        ;
END         ;End of program
```

EXAMPLE #42

DETECT MARK AND CUT:

The "Detect Mark and Cut" program is designed to cut on-the-fly on a marker that is on the material to be cut. The material is continuously advancing, while the knife travels back and forth. Upon detection of the photocell, the knife begins to move, matches velocity with the material, and is positioned precisely on the marker. An offset from the marker can also be easily handled. The knife is then triggered to cut and upon completion, a retrace motion is started which brings the knife to its rest position and ready to be retriggered. The rest position must be repeatable and can be the "Home" position (see attached sketch). Some of the working details are as follows:

When photocell is received, adjust i12 or t (latter, preferred) to cause X axis to move so when it matches speeds with Y it has the cutter exactly on the photocell trigger point which has also been moving. Or, if preferred, adjust DP by moving the photocell.

Choose f maximum of sync encoder for operation at maximum speed. If Y will be 20,000 counts/second *maximum*, then choose i27=20 and i47=32768, then [i27 x (32768/i47)] = 20 counts/milliseconds. **NOTE THAT THIS MAXIMUM RATE OF 20,000 COUNTS/SECOND MUST NEVER BE EXCEEDED.** Exceeding this maximum frequency results in the % value rolling over and causing the feedrate to drop to zero.

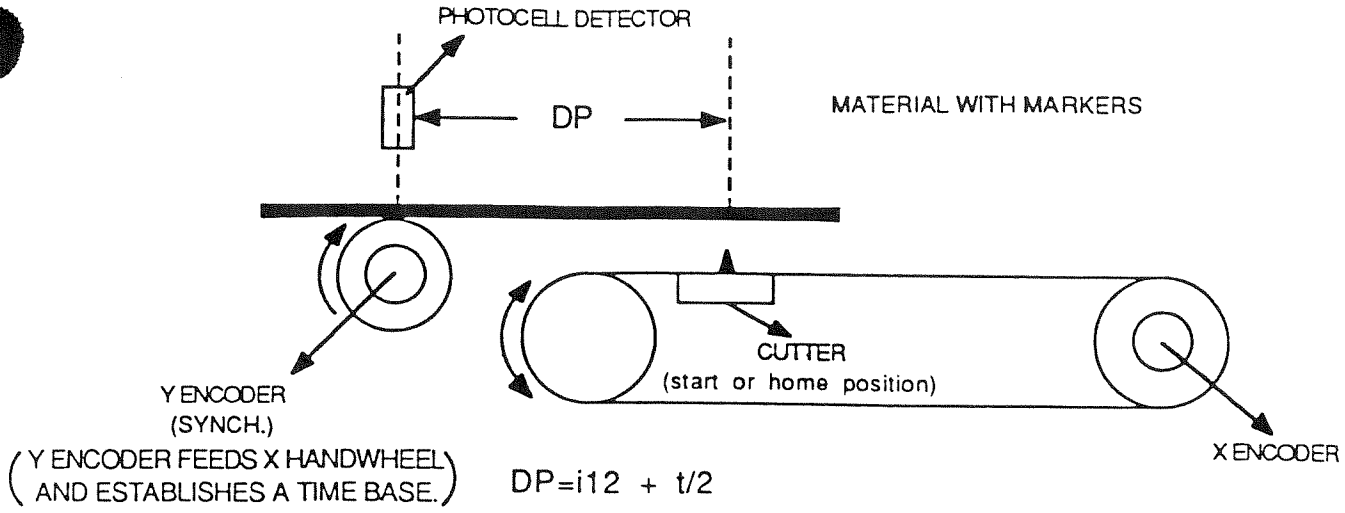
The 20 counts/milliseconds referred to here is slightly inaccurate since the 20 counts occur in a "binary millisecond", which is approximately 976 microseconds, or one SMCC servo cycle. However, since this value of 20 is used in conjunction with "T" or "t", which are also in "binary milliseconds", positioning is accurate.

Then, if DP = 500 encoder counts on X axis, DP = i12 + t/2 = DP lines/20 = 500/20 = 25 msec.

This could be too small, use bigger DP so there is enough time. If i12 = 60 and t = 140, then DP is 130/20 = 65 msec. then 65 x 20 = 1300 encoder counts in Y.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DETECT MARK AND CUT



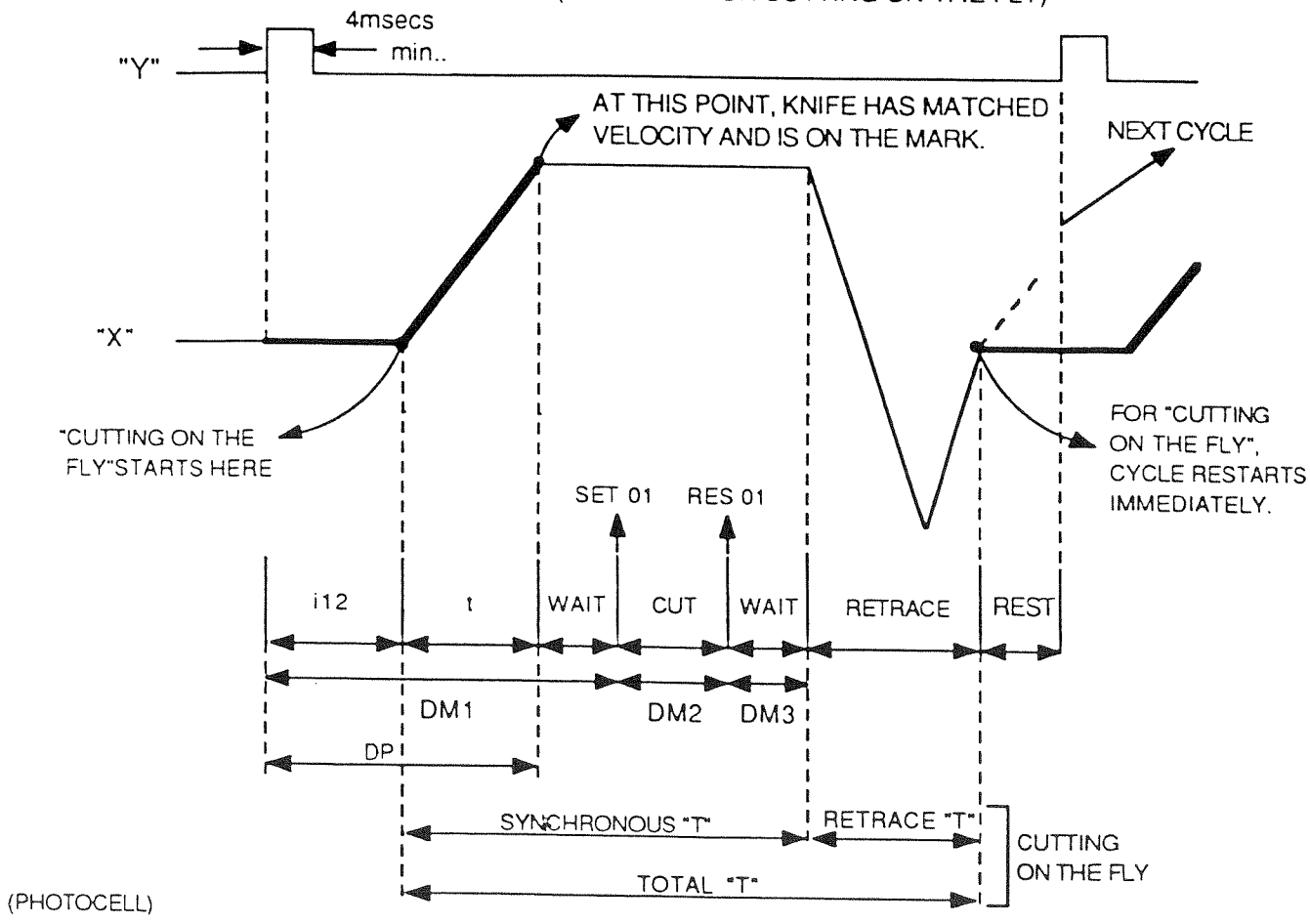
(Y ENCODER FEEDS X HANDWHEEL AND ESTABLISHES A TIME BASE.)

DP=i12 + t/2

(i27 x 32768)/i47 = # OF COUNTS/BINARY MILLISECOND FOR PARABOLIC SMCC.

(32768/i27) = #OF COUNTS/BINARY MILLISECOND FOR STANDARD SMCC.

PHOTOCELL TRIGGER (USE INTO "RUN" INPUT OF SMCC)
(NOT USED FOR CUTTING ON THE FLY)



(PHOTOCELL)

Artisan Scientific - Quality Instrumentation ..? Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Now we try to write a program as follows:

NOTE: Figures inside square brackets cannot be entered as written. The calculations should be done in another way (e.g. P parameter calculations).

```
i1260                    ;Set i12 to 60 (could be as small as 20).
                         ;i12 is the SMCC's "Move Calculation
                         ;Time".
A0z                      ;Address and clear buffer
HOM                      ;Do a homing function to establish the
                         ;accurate rest position of the knife.
FOR 0                    ;Start loop
t140                     ;Set accel/decel time to approximately
                         ;140 msec. (140 x 1000/1024) =
                         ;136.72 milliseconds.
T[(DM1/i27)x(X/Y)]       ;This is a time for the move, calculated
                         ;so that when X (the belt) moves, it will
                         ;match velocity with the part coming in.
                         ;T must be at least equal to or longer
                         ;than t, and X/Y is the ratio of encoder
                         ;counts per inch for X and Y
                         ;respectively.
                         ;
                         ;DM1 can be as large as desired, the
                         ;greater it is, the longer the wait time
                         ;while maintaining matched speed.
SET 01                   ;Turn on the cutter, solenoid, or other
                         ;device.
T[(DM2/i27)x(X/Y)]       ;Continue to maintain matched speed
U [(DM2)]                ;while the cut is occurring. Make sure
                         ;T is large enough for the cut to occur.
RES 01                   ;Turn off the cutter.
T[(DM3/i27)x(X/Y)]       ;Allow time for things to settle and
                         ;cutter to disengage.
U [(DM3)]                ;
T 500                    ;Allow 1/2 second (or as needed) to do
X0                       ;the retrace time. Note that you use
                         ;absolute positioning to return to zero,
                         ;which is the home or rest position.
STO                      ;Stop the program and wait for next
                         ;photocell pulse (Run Command).
NEX                      ;Repeat the loop
END                      ;End of program
```

Please note that since the photocell is used to trigger the SMCC's "Run" input, there is an inherent timing error due to the SMCC's response time to the trigger, which is in the order of about three milliseconds. There will therefore be a +/-1.5 milliseconds of error in the cut. If "Y" velocity is known, this can be translated into a distance error.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Also, if the "Y" material speed will be changed often as the cutting process is taking place, it is recommended that the SMCC with parable option be used in order to minimize errors due to the following error since it provides velocity and acceleration feedforward terms.

In the above example, the cycle repeats but has a "rest" period and is retriggered each time. This means that small errors in the timing (choosing i27 and i47]) will not accumulate. However, if we consider a problem which is similar but different, that of continuously cutting a given length without stopping and without retriggering on a mark, which is an "on-the-fly" cutting of a length, it then becomes very important to have an exact (non-fractional) count per millisecond time base derived from the handwheel encoder. In this example, we had chosen 20 counts/millisecond, using i27 and i47, since the Parabolic SMCC allows the use of both these parameters to derive the time base. In the Standard SMCC, more care has to be exercised, since only i27 can be used and it is therefore more difficult to come up with "whole numbers" for the time base. See "% Feedrate Chart" for details. The Standard SMCC also does not support the i12 (calculation time) parameter, so as a result there is a variable time of 8 to 12 msec. which causes some inaccuracy in the cut.

To program the "on-the-fly" cutting, it is simply necessary to write the program so that the total time "T" of the synchronous and retrace moves represents the move distance of the material to be cut. (Total time T) x 20 counts/msec. = (size of the cut), where (total time T) is the sum of the "synchronous time T", when the knife is following the web and the "retrace time T" when the knife is going back to its zero position.

An "on-the-fly" program would be as follows:

```
A0z
FOR 0       ;Start a loop forever
T1000       ;The "Synchronous Time T" here
            ;is a one second move time
            ;which translates to a distance of
            ;1000 x 20 = 20,000 encoder counts for the
            ;material to be cut.
X2000       ;This move must be long enough to allow
            ;full speed (synchronous speed) to be
            ;reached.  Additional small moves may be made
            ;in order to let the system stabilize and
            ;activate the knife.
T500        ;"Retrace Time", can be made large or small
            ;to change the length of the material to be cut.
XO          ;Return to zero position
NEX         ;Return to loop
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #43

SMCC AS VOLTMETER:

The voltage displayed on the screen consists of three digits; the first digit represents volts, the second digit represents 1/10 of a volt, and the third digit represents 1/100 of a volt. The readings will be displayed on the CRT screen each time that a change of greater than 110 mV is detected. Set P1=1 to start the display. For example: 492 is 4.92V, 12 is 0.12V, and 5 is 0.05V.

```
A0z100              ;Address card, clear buffer
I621                ;Enable PLC
RESP                ;Reset all P parameters
SETC16=fe90         ;Address of Anal input, using
                    ;Parabolic SMCC. See memory map
                    ;for other addresses.
SETP4=100           ;Scale factor
SETP6=51            ;1V=51
SETP3=1             ;
SETP8=6             ;6=110mV, sets display trigger
                    ;range
SETP10=1            ;Used to display once only
END                 ;End of program
A0PURGE             ;Clear existing PLC programs
IFP1<1              ;This flag causes PLC to look
ANDP1>-1            ;at the address of Anal input.
THENP2=C16          ;Store Anal input value in P2, ONE
                    ;BYTE VALUE.  Data is converted
                    ;from binary to decimal.  If two
                    ;bytes must be read, then use C16,2.
ANDIFP3>0           ;Check calculation flag
THENP5=(P2*P4/P6)   ;Voltage scale
ANDP7=(P2+P8)       ;Tolerance +110mV
ANDP9=(P7-P8)       ;Tolerance -110mV
ANDP3=0             ;Reset calculation flag
ANDP10=1            ;Set screen output flag
ANDT0=5             ;Set time delay for screen output
ELSEIFP2>(P7)       ;Monitor Anal input voltage
ORP2<(P9)           ;
THENP3=1            ;Set calculation flag
ELSEIFT0=0          ;Check time delay to print
ANDP10>0            ;the voltage value of Anal input.
THENP10=0           ;Reset screen output flag
AND"P5"             ;Send "P5" which is the calculated
                    ;value to the screen.
END                 ;End of PLC
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #44

MULTIPLE SMCC SEQUENCING METHOD:

```
AAq      ;It is sometimes necessary to have several
         ;SMCCs functioning together in a manner
         ;that allows various tasks to be sequenced
         ;from 1 card to the next.  The method
         ;chosen for passing sequencing information
         ;from one SMCC to the next is the use of
         ;the SMCC inputs on outputs.  On each card,
         ;Input 1 and Output 1 are tied together and
         ;then they are tied to all the card I/O 1's.
         ;The same applies to I/O 2, 3, & 4. This way
         ;any card can output a sequence number from
         ;1 to 16 (4 bits) and any card can read the
         ;data. This example is written for 3 cards
         ;but up to 16 cards can be handled.
M02=0    ;Clear all outputs
M03=0
M04=0
M05=0
i1240    ;Set servo calculation time to 40 mse c
z100     ;Clear buffer
A0GOS01  ;Start card 0
AAFOR0   ;Start loop
GOS20    ;Wait for Start Command
A0DWE19  ;Action for card 0
GOS01    ;Start card 1
A1DWE20  ;Action for card 1
GOS02    ;Start card 2
A2DWE21  ;Action for card 2
GOS00    ;Start card 0
AANEXT   ;End of loop
END      ;End of the program
L00      ;Start card A0
GOS16    ;Wait for all lines to be 1
SET02    ;Turn on Output 2
SET03    ;Turn on Output 3
SET04    ;Turn on Output 4
SET05    ;Turn on Output 5
DWE25    ;Wait 25 msec
RES02    ;Turn off Output 2
RES03    ;Turn off Output 3
RES04    ;Turn off Output 4
RES05    ;Turn off Output 5
RET      ;Return, end of subroutine
L01      ;Start card A1
GOS16
SET03
SET04
SET05
DWE25
RES03
RES04
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
RES05
RET
L02      ;Start card A2
GOS16
SET02
SET04
SET05
DWE25
RES02
RES04
RES05
RET
L16GOT21IF04*15 ;Wait for all lines to be 1
DWE20           ;Wait 20 msec
GOT16           ;Go back to L16
L20CAS04,16L    ;Wait for start code
                ;The "Case" Command is looking at
                ;Address 4 which is the location of
                ;the inputs, see memory map .
                ;The input code that makes this
                ;card do something is "0", which
                ;makes the card execute the L21
                ;routine (a simple return in this
                ;case).
A0GOT21         ;Card A0, 15 code "0", go to L21
DWE20           ;         14 wait 20 msec
DWE20           ;         13 wait 20 msec
DWE20           ;         12 wait 20 msec
DWE20           ;         11 wait 20 msec
DWE20           ;         10 wait 20 msec
DWE20           ;         9  wait 20 msec
DWE20           ;         8  wait 20 msec
DWE20           ;         7  wait 20 msec
DWE20           ;         6  wait 20 msec
DWE20           ;         5  wait 20 msec
DWE20           ;         4  wait 20 msec
DWE20           ;         3  wait 20 msec
DWE20           ;         2  wait 20 msec
DWE20           ;         1  wait 20 msec
DWE20           ;         0  wait 20 msec
A1DWE20         ;Card A1, 15 wait 20 msec
GOT21           ;         14 code "1", go to L21
DWE20           ;         13 wait 20 msec
DWE20           ;         12 wait 20 msec
DWE20           ;         11 wait 20 msec
DWE20           ;         10 wait 20 msec
DWE20           ;         9  wait 20 msec
DWE20           ;         8  wait 20 msec
DWE20           ;         7  wait 20 msec
DWE20           ;         6  wait 20 msec
DWE20           ;         5  wait 20 msec
DWE20           ;         4  wait 20 msec
DWE20           ;         3  wait 20 msec
DWE20           ;         2  wait 20 msec
DWE20           ;         1  wait 20 msec
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
DWE20           ;          0  wait 20 msec
A2DWE20         ;         15  wait 20 msec
DWE20           ;         14  wait 20 msec
GOT21           ;         13  code "2", go to L21
DWE20           ;         12  wait 20 msec
DWE20           ;         11  wait 20 msec
DWE20           ;         10  wait 20 msec
DWE20           ;          9  wait 20 msec
DWE20           ;          8  wait 20 msec
DWE20           ;          7  wait 20 msec
DWE20           ;          6  wait 20 msec
DWE20           ;          5  wait 20 msec
DWE20           ;          4  wait 20 msec
DWE20           ;          3  wait 20 msec
DWE20           ;          2  wait 20 msec
DWE20           ;          1  wait 20 msec
DWE20           ;          0  wait 20 msec
AAGOT20         ;Go back to L
L21RET          ;Return
```

EXAMPLE #45

BURR-BROWN 2500, MDI MODE:

To use The BURR-BROWN terminal in MDI mode for the SMCC, it is not necessary to write a program for the SMCC but only to program the BURR-BROWN's function keys as follows:

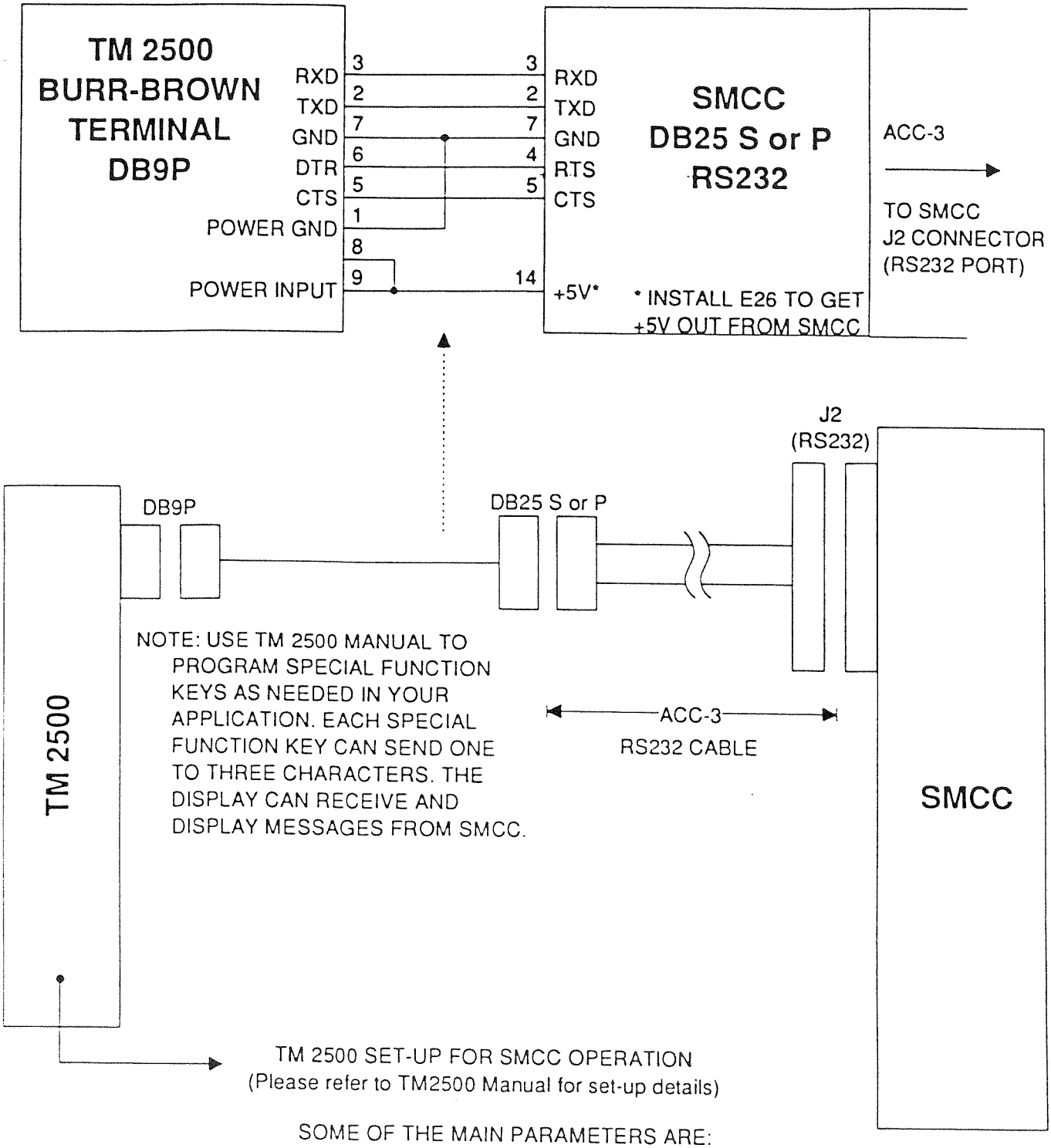
1) Assign "A0i" to the first function key.
2) Assign "F" to the second function key.
3) Assign "X" to the third function key.
4) Assign "Y" to the fourth function key.

The fifth and sixth function key may be programmed for other simple functions such as "h" for homing, "p" for display position, etc.

After successfully programming the function keys, then connect the BURR-BROWN terminal to the SMCC. Now to enter the MDI mode, push the "A0i" key, enter 021 to enable the MDI mode, and push the enter key. Any of the i parameters can be entered or displayed. Push the "F" key, enter the desired feedrate, and push the enter key. To move the X motor, push the "X" key, enter the desired position, and push the enter key. To move the Y motor, push the "Y" key, enter the desired position, and push the enter key. To move both X and Y motors, push the "X" key, enter the desired position for X, push the "Y" key, enter the desired position for Y, and push the enter key. To disable the MDI mode, push the "A0i" key, enter 020 to disable, and push the enter key.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

BURR-BROWN TM2500 TERMINAL CONNECTIONS



TM 2500 SET-UP FOR SMCC OPERATION
(Please refer to TM2500 Manual for set-up details)

SOME OF THE MAIN PARAMETERS ARE:

TM →	1	Block Mode
MA →	00	
BR →	2	9600 Baud
DF →	4	8 bits/No Parity
HS →	0	Block Mode
EN →	3	CRLF

(BUR-BRWN)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #46

BURR-BROWN 2500:

Please refer to BURR-BROWN 2500 Manual to program the function keys on the BURR-BROWN terminal.

First connect the BURR-BROWN terminal to COM1 (RS232) of any PC, then run any terminal software such as FREEWAY, ACE, etc. Secondly, configure the PC and the BURR-BROWN terminal, so both can communicate with each other, thirdly follow the BURR-BROWN Manual on how to program the function keys. Assign "P2=" to the first key, "P1=" to the second key, "RUN" to the third key, and "Q" to the fourth key.

After successfully programming the function keys, connect the BURR-BROWN terminal to the SMCC. To enter the desired position of the SMCC, push the first function key, enter the desired distance, and push enter. To enter the desired speed of the SMCC, push the second function key, enter the desired velocity, and push enter. To run the program, push the third function key and then push enter on the BURR-BROWN terminal. To stop the program, push the fourth function key and then push enter. Similarly, the fifth and sixth function keys may be programmed for other single functions such as "h" for homing, "Z" for zero setting position, etc.

This program uses BURR-BROWN terminal to input the position and the feedrate. Also, at the end of the move, the SMCC will send the position of the motor to the BURR-BROWN terminal. Before running this program, enter the position as the P2 variable and feedrate as the P1 variable.

```
A0z100      ;Clear buffer
I621        ;Enable PLC
RESP        ;Reset all variable parameters
HOM         ;Do a home
DWE1024     ;Wait 1 second
FOR0        ;Begin loop
F(P1)       ;Feedrate
X(P2)       ;Move X to its destination
SETP3=1     ;Set send position flag
DWE1024     ;Wait 1 second
X0          ;Move X back to 0 position
DWE1024     ;Wait 1 second
NEXT        ;End of loop
END         ;End of program
A0PURGE     ;Clear existing PLC programs
IFP3>0      ;Check if send position flag is set
ANDP3<2     ;This issues that the PLC sends the "p"
            ;only once per occurrence.
THEN"p"     ;True, display position
ANDP3=0     ;Reset send position flag
END         ;End of PLC program
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE #47

GENERAL S-CURVE PROGRAM FOR 2-AXIS MOVES:

User must define maximum velocity and accel at start. Then, to make a move, he simply must set P1 to the new X location, and P2 to the new Y location (this can be done during the previous move). When the "R" (Run Command) is sent, all of the move parameters will be calculated, but the move will not occur until a CONTROL-R is sent.

```
A0z200                      ;Clear buffer, reserve 200 lines for
                            ;main program. This setting of
                            ;parameters done in a subroutine
                            ;that is only called when M63 = 0 -
                            ;this should be first time only,
                            ;because subroutine sets M63 = 1.
                            ;User should make sure M63=0 before
                            ;first run or whenever P14 or P15 is
                            ;changed; also, i12 should be set to 80
                            ;or greater whenever the subroutine is
                            ;to be run -- the subroutine resets it
                            ;to a lower value to speed
                            ;subsequent moves.
i12 100                     ;Non-buffered: allows large calc time
L63                         ;User sets these two values for a
                            ;given system
SET P15=100000              ;Max velocity in cts/sec
SET P14=4000000             ;Peak accel in cts/sec^2
                            ;Other P parameters are automatically
                            ;computed during first run of program.
                            ;P10 thru P15 should not be overwritten
                            ;outside of this subroutine.
SET P13 = 1024              ;Accel time for 3-part move
SET P13 = (P13*P15/P14)     ;In (1/1024 sec)
SET P11 = (P15*P15/P14)     ;Accel/decel distance
SET P12 = (P11+P11)         ;At both ends
SET P10 = 50                ;1/50 sec is min slew time
SET P10 = (P15/P10)         ;Min slew distance
SET P12 = (P10+P12)         ;Min 3-part move distance
SET P8 = 16                 ;P10=16/9*Vmax^2/A max...
SET P9 = 9                  ;The minimum 1-part move that
SET P10 = (P8*P11/P9)       ;hits max velocity
                            ;Define bits for program logic
C23 = a24b                  ;MSByte of P3
M23 = C23.7                 ;MSBit (sign bit) of P3
C24 = a24f                  ;MSByte of P4
M24 = C24.7                 ;MSBit (sign bit) of P4
C27 = a25b                  ;MSByte of P7
M27 = C27.7                 ;MSBit (sign bit) of P7
C29 = a263                  ;MSByte of P9
M29 = C29.7                 ;MSBit (sign bit) of P9
C08 = a25d                  ;2nd LSByte of P8 (dt^2)
C20 = febd                  ;Data storage bit is address
M20 = C20.5                 ;febd, bit 5
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SLEW 65535                  ;For max change in rate
CIRCLE4                     ;Put into S-curve mode
SET 63                      ;So won't be called next time
RET                         ;Main program starts with call to subroutine
GOSUB 63 IF (M63=0)         ;This will be executed each time Run
                            ;Command given
SET P3=AX                   ;Present commanded X position
SET P4=AY                   ;Present commanded Y position
SET P3=(P1-P3)              ;Delta X move
GOTO +02 IF (M23=0)         ;Jump if positive
SET P5=(-P3)                ;abs (delta X) = -delta X
GOTO +01
SET P5=(P3)                 ;abs (delta X) = delta X
SET P4=(P2-P4)              ;Delta Y move
GOTO +02 IF (M24=0)         ;Jump if positive
SET P6=(-P4)                ;abs (delta Y) = -delta Y
GOTO +01
SET P6=(P4)                 ;abs (delta Y) = delta Y
SET P7=(P5-P6)              ;abs(delta X) - abs(delta Y)
GOTO +02 IF (M27=0)         ;Jump if X move longer
SET P8=(P6)                 ;Delta P = abs(delta Y)
GOTO +01
SET P8=(P5)                 ;Delta P = abs(delta X)
SET P9=(P8-P12)             ;Compare to shortest 3-part
GOTO +32 IF (M29=1)         ;Jump if short (1-part) move
```

This section computes the parameters for a three-part S-curve move:

```
SET P9=1024
SET P9=(P9*P8/P15)          ;Slew time is total time
SET P9=(P9-P13)             ;minus accel time at each end
SET P9=(P9-P13)
GOTO +06 IF (M27=0)         ;Jump if X move is longer
SET P7=(P3*P11/P6)          ;Scale X accel dist to Y move
GOTO +02 IF (M24=0)         ;Jump if positive Y move
SET P8=(-P11)               ;Y accel dist, neg move
GOTO +01                    ;
SET P8=(P11)                ;Y accel dist, pos move
GOTO +05
SET P8=(P4*P11/P5)          ;Scale Y accel dist to X move
GOTO +02 IF (M23=0)         ;Jump if positive X move
SET P7=(-P11)               ;X accel dist, neg move
GOTO +01
SET P7=(P11)                ;X accel dist, pos move
SET P5=(P3-P7)              ;X slew dist is total dist
SET P5=(P5-P7)              ;minus dist at each end
SET P6=(P4-P8)              ;Ditto for Y slew
SET P6=(P6-P8)

                            ;Now do the 3-part S-curve move
                            ;Total move time is 4*P13 + P9 (in
                            ;1/1024 secs)
POT 0                       ;To delay move until ^R
t(P13)                      ;Accel time
U(P7) V(P8)                 ;Accel distances
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
t(P9)                       ;Slew time
U(P5) V(P6)                 ;Slew distances
t(P13)                      ;Decel time
U(P7) V(P8)                 ;Decel distances
DWELL 1                     ;Make sure it finishes
GOTO +46                    ;Done with move -- go to end
```

The following section does a one-part S-curve move:

```
SET P9 = (P8-P10)           ;Compare to shortest max vel.
GOTO +03 IF (M29=1)         ;Move: jump if shorter
                            ;Longer 1-part move: time set to hit max
                            ;velocity
SET P9 = 768                ;dt= 1024 * (3*Len)/(4*Vmax)
SET P9 = (P9*P8/P15)
GOTO +36                    ;Jump to exec of 1-part move
                            ;Short 1-part move: time set to reach
                            ,max accel
                            ;The only tough thing is to determine
                            ;the move time which is the square root
                            ;of (distance/peak accel)
SET P9 = 1048576            ;1024^2
SET P8=(P8*P9/P14)          ;(delta t)^2  in bin msec ^2
SET P9 = 125                ;Safe value if lookup fails
CASE 08, 32B                ;Find delta t by lookup
SET P9 = 16                 ;C08=0: dt^2 < 256
SET P9 = 22                 ;C08=1: dt^2 < 512
SET P9 = 27                 ;C08=2: dt^2 < 768
SET P9 = 31                 ;C08=3: dt^2 < 1024
SET P9 = 35                 ;C08=4: dt^2 < 1280
SET P9 = 39                 ;C08=5: dt^2 < 1536
SET P9 = 42                 ;C08=6: dt^2 < 1792
SET P9 = 45                 ;C08=7: dt^2 < 2048
SET P9 = 48                 ;C08=8: dt^2 < 2204
SET P9 = 50                 ;C08=9: dt^2 < 2560
SET P9 = 53                 ;dt^2 < 2816
SET P9 = 55                 ;dt^2 < 3072
SET P9 = 58                 ;dt^2 < 3328
SET P9 = 60                 ;dt^2 < 3584
SET P9 = 62                 ;dt^2 < 3840
SET P9 = 64                 ;dt^2 < 4096
SET P9 = 66                 ;dt^2 < 4352
SET P9 = 68                 ;dt^2 < 4608
SET P9 = 70                 ;dt^2 < 4864
SET P9 = 72                 ;dt^2 < 5120
SET P9 = 74                 ;dt^2 < 5376
SET P9 = 75                 ;dt^2 < 5632
SET P9 = 77                 ;dt^2 < 5888
SET P9 = 78                 ;dt^2 < 6144
SET P9 = 80                 ;dt^2 < 6400
SET P9 = 81                 ;dt^2 < 6656
SET P9 = 83                 ;dt^2 < 6912
SET P9 = 84                 ;dt^2 < 7168
SET P9 = 86                 ;dt^2 < 7424
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
SET P9 = 89                ;dt^2 < 7936
SET P9 = 90                ;dt^2 < 8192
                           ;More entries could be added here if
                           ;necessary (e.g., for low acceleration
                           ;relative to velocity)
POT 0                      ;Don't move until get ^R
t(P9)                      ;Delta t (1/3 total move time)
U(P3) V(P4)                ;Incremental position moves
DWELL 1                    ;Make sure it finishes
I12 38                     ;Make sure calc time is right
END
```

EXAMPLE #48

TO READ VALUES FROM THUMBWHEEL BOARD(S)

This procedure allows the user to specify the address of the least significant pair of thumbwheels (highest address) and the number of thumbwheel pairs to comprise the value to be read. The number must consist of pairs of thumbwheels. The procedure is contained in subroutines and PLC so it can simply be added to a user program. It uses parameters P5, P6, P7, P8, & P9; C06, C09, C10, & C21; M21 & M22.

Below is a very brief program to demonstrate thumbwheel decoding.

```
A0z40
PURGE
                           ;Parameter definitions
SET C10=ffb8               ;Parallel port input address
SET C09=ffb9               ;Parallel port output address
SET C06=a254               ;LSByte of P6
SET C21=a3a7               ;Empty memory location
SET M21=C21.0              ;Control bit to start read
SET M22=C21.1              ;Control bit to increment read
I12 40                     ;Default -- shorter than delay
RESETP
                           ;Get values from thumbwheels
SET P5=2                   ;Address 2 is lowest digits
SET P6=3                   ;3 pairs (6 digits) in value
GOSUB 00                   ;Thumbwheel value --> P7
SET P1=(P7)
SET P5=6                   ;Address 6 is lowest digits
SET P6=3                   ;3 pairs (6 digits) in value
GOSUB 00                   ;Thumbwheel value --> P7
SET P2=(P7)
t256
F1024
X(P1)Y(P2)                 ;Go to position on thumbwheels
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Main thumbwheel decoding routine:

```
L00
SET 21              ;Cause PLC to write base address
                    ;to parallel port output
DWELL100            ;To make sure address takes...
                    ;this must be 10 longer than i12
SET P7=0            ;To start
SET P9=1            ;Multiplier: 1's digit to start
L01
GOSUB 02            ;Read hi nibble (lo addr) -> P8
SET P7=(P7+P8*P9)   ;Add into value
SET P8=10           ;To get to next digit
SET P9=(P9*P8)      ;P9 now set for next digit
GOSUB 03            ;Read lo nibble (hi addr) -> P8
SET P7=(P7+P8*P9)   ;Add into value
SET P8=10           ;To get to next digit
SET P9=(P9*P8)      ;P9 now set for next digit
SET P6=P6-1         ;To mark next pair of digits
GOTO +03 IF NOT 06 > 0    ;Jump if no more digits
SET 22              ;Cause PLC to decrement address
                    ;at parallel port output
DWELL100            ;make sure new address takes...
                    ;this must be 10 longer than i12
GOTO 01             ;Back to L01 for next pair
RET                 ;End of main subroutine
```

This subroutine decodes the value of the right-hand digit of a pair of thumbwheels (high address) and returns the value in P8:

```
L02
SET P8=0
CASE 10, 10H        ;Digit value is in C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

This subroutine decodes the value of the left-hand digit of a pair of thumbwheels (low address) and returns the value in P8:

```
L03
SET P8=0
CASE 10, 10L      ;Digit value is in C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET
```

PLC routines are used to write to the parallel output port because only they can do so parametrically.

```
IF M21=1          ;Start control bit set
THEN C09=(P5)     ;Write base address to port
AND M21=0         ;Done; reset control bit

IF M22=1          ;Decrement control bit set
THEN P5=P5-1
AND C09=(P5)      ;Decrement address at port
AND M22=0         ;Done; reset control bit
END
```

EXAMPLE #49

GLUE DISPENSING, THREE AXIS

This program demonstrates how to use two SMCCs to dispense glue around the periphery of a part in such a way as to maintain the third axis (A1), the "glue gun" orthogonal to the part's contour. Even though it is possible to take advantage of the SMCC's PLC "ATAN" arctangent calculation capability to generate the third axis movements automatically, this program instead uses direct programming for the third axis and demonstrates how three axes may be coordinated via a simple program. Please note that even though not shown, it is possible to use the fourth axis (Y on A1), as a simple D/A converter which generates a voltage based on the actual feedrates and where the same output voltage can be used to control the glue's flow rate.

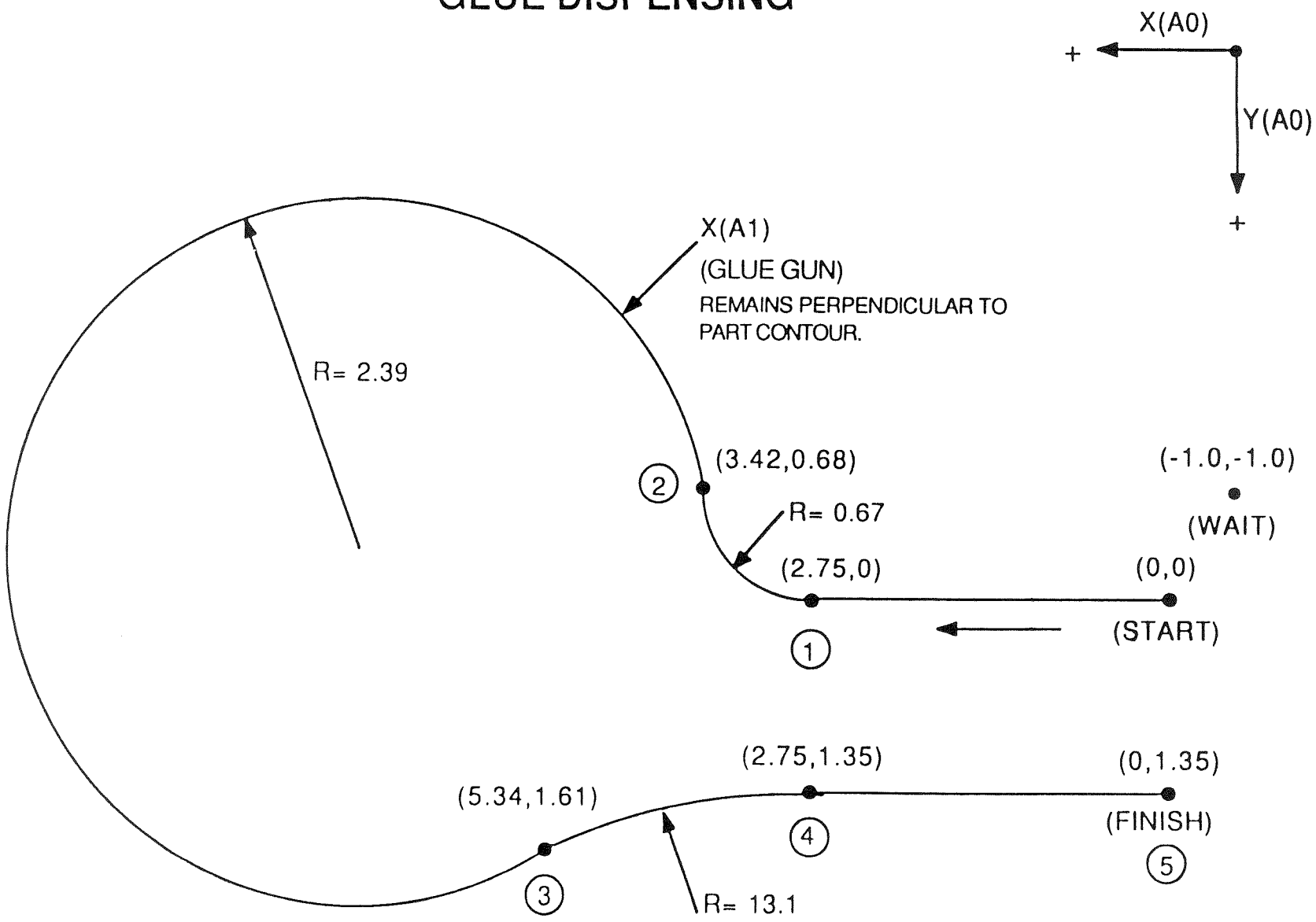
```
AAz100            ;Address all cards & reserves
                  ;100 buffer lines
F1024             ;Feedrate,100 % of i06
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
t30                 ;Acc/dec time
HOME                ;Homing,set i04,i25[i45] & i26[i46]
X10.00 Y10.00       ;to suit
                    ;Axis move to wait position,
                    ;issue "RUN" Command to continue
A0X=-1.00           ;Set position for X & Y
Y=-1.00
A1X=0               ;Set position for Z (glue gun axis).
AASTOP              ;Wait until "RUN" Command is
                    ;issued
FOR0                ;Begin loop
T689                ;Time to move
A0X0Y0              ;Move X & Y to corner of part
A1X0                ;Z wait, no move
AASET01             ;Turn on glue,use M01 output
DWE20               ;Wait for 20 msec, this forces a stop
                    ;Adjust dwell so glue starts
                    ;flowing at the right place
LIN                 ;Linear mode
T805                ;Time to move
A1X0                ;Z wait,no move
A0X2.75Y0           ;Move X to point # 1
RX=0.67             ;X radius of the arc
RY=0.67             ;Y radius of the arc
CIR0                ;Circular mode (angle<=180),CW
AAT313              ;Time to move
A1X-91.2            ;Rotate glue head to -91.2 degrees
A0X3.42Y-0.68       ;Arc move to point #2
RX=2.39             ;X radius of the circle
RY=2.39             ;Y radius of the circle
CIR3                ;Circular mode (angle >180),CCW
AAT3449             ;Time to move
A1X191.4            ;Rotate glue head 191.4 degrees
A0X5.34Y1.61        ;3/4 of circle move to point #3
RX=13.1             ;X radius of the arc
RY=13.1             ;Y radius of the arc
CIR0                ;Circular mode (angle<=180),CW
AAT761              ;Time to move
A1X180.0            ;Rotate glue head 180 degrees
A0X2.75Y1.35        ;Arc move to point #4
LIN                 ;Linear mode
AAT805              ;Time to move
A1X180.0            ;Wait, don't rotate glue head
A0X0Y1.35           ;Move X to point #5
AARES01             ;Turn off glue (M01 off)
DWE20               ;Wait for 20 msec, adjust to suit
T689                ;Time to move
A1X0                ;Rotate glue head back to original
                    ;position
A0X-1.00Y-1.00      ;Move X & Y back to wait position
AASTOP              ;Wait until "RUN" Command is
                    ;issued
AANEXT              ;End of loop
END                 ;End of program
                    ;Assuming 10,000 counts per inch
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

GLUE DISPENSING



(GLUE GUN)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ;on A0 X & Y axis and 2,000 counts
                    ;per 360 degrees glue gun rotation
                    ;Scale factors are set as follows:
                    ;A0 is i3810000:10000.4 & i5810000
                    ;:10000.4
                    ;A1 is i382000:3600.1, change to
                    ;suit
                    ;Limit & Home switches must be
                    ;installed & wired
AAPURGE             ;Clear all existing PLC program
IF M15=1            ;Check if M15 switch is toggled
THEN IF P1=0        ;Check if send flag is set
THEN"R"             ;If true, send "RUN" Command
AND P1=1            ;Reset send flag
ELSE
ELSE P1=0           ;Switch isn't toggled, set send
                    ;flag
END                 ;End of PLC program
                    ;This PLC program can be used to
                    ;provide a remote "Run" trigger
                    ;for the machine.
                    ;To implement, connect the M15
                    ;input, (JMach. C28) on both card
                    ;A0 & A1 to the switch.
                    ;When switch is grounded, both cards
                    ;will start running.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E - POINT (JUMPER) DESCRIPTIONS

TABLE OF CONTENTS

PAGE

E-POINTS, TEST POINTS, AND ADDRESS SELECTIONS. . . 309

E-POINT AND TEST-POINT DIAGRAM 317

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC E POINTS, TEST POINTS, AND ADDRESS SELECTIONS (HARDWARE CONFIGURATION)

****

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E1	2 1 * *	D-8	JUMP Pin 1 to Pin 2 to increase gain of phase 1 output. No short is +/-5 volts out, short is +/-10 volts out. See E9 also. Normally used at +/-10 volts out for single ended output and +/-5 volts out when output is used differentially.	Jumper installed +/-10 volts out
E2	1 2 * *	D-6	JUMP Pin 1 to Pin 2 for Sync/Out (use on A0 card only.) Note that if E2-1,2 is installed, E3-1,2 must be removed.	Jumper installed Card is set up as A0.
E3	1 2 * *	D-5	JUMP Pin 1 to Pin 2 for input sync (use on all cards other than A0). Also see E2.	No jumper (open)
E4	3 2 1 * * *	D-4	JUMP 1 to 2 to allow input bit from JEXP (J8 Pin 15) to be read by CPU. JUMP 2 to 3 to allow "servo 2" to go to JEXP (J8 Pin 15) to be used for R/D conversion.	No Jumper (open)
E5			NOT USED	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E6	1 2 * *	D-3	JUMP Pin 1 to Pin 2. With jumper installed, EA/ (Pin 51) of CPU is grounded, which causes the CPU to use external memory. Otherwise, EA/ (Pin 51) is pulled up to +5 volts, CPU then uses internal memory only.	No Jumper (open)
E7	1 2 * * * * 3 4	D-3	JUMP 1 to 2 to allow RXD/ as output J2 Pin all (J PAR-RS232)	1-2 Jumper installed
			JUMP 3 to 4 to allow TXD/ as input J2 Pin a12 (J PAR-RS232)	3-4 Jumper installed
			JUMP 1 to 3 to allow RXD/ as input J2 Pin a12 (J PAR-RS232)	RXD/ is output
			JUMP 2 to 4 to allow TXD/ as output J2 Pin a12 (J PAR-RS232)	TXD/ is input
			E7 is used to in effect swap pins 2 and 3 on a standard RS232 connector to accommodate various host configurations as Data Set or Date Terminal. Also see E8.	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E8	1 2 * * * * 3 4	D-3	JUMP 1 to 2 to allow RTS to be output, J2 pin a9 (J PAR-RS232)	1-2 Jumper installed
			JUMP 3 to 4 to allow CTS to be input, J2 pin a10 (J PAR-RS232)	3-4 Jumper installed
			JUMP 1 to 3 to allow RTS to be input, J2 PIN a9 (J PAR-RS232)	RTS is output
			JUMP 2 to 4 to allow CTS to be output, J2 PIN a10 (J PAR-RS232)	CTS is input
			Use E8 in conjunction with E7 for proper configuration.	
E9	1 2 * *	C-7	JUMP Pin 1 to Pin 2 to increase gain of Phase 3 (See E1)	Jumper installed +/-10 volts out
E10			NOT USED	
E11	*1 *2	C-6	JUMP Pin 1 to Pin 2 to allow -14 VA to come from J11 (JPWR) (Ties Amplifier and SMCC power supply together. Defeats OPTO coupling).	No jumper (open)
			Note that if E11 is changed, E15 and E24 must also be changed.	No jumper

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E12	*3 *2 *1	C-4	JUMP Pin 2 to Pin 3 to allow U17 (Pin 26) SRAM and U17A (Pin 26) EAROM to input +5V. for 8KX8 parts. JUMP Pin 1 to Pin 2 to allow U17 (Pin 26) SRAM and U17A (Pin 26) EAROM to input A13 for 32KX8 parts.	2-3 Jumper installed card is configured for 8KX8 parts.
E13	1 2 3 * * *	B-8	When short removed from 1 & 2 and placed on 2 and 3, provides X axis unipolar analog error output with direction bit on BSINX at J1C18. (See E18 for Y axis.)	1-2 Jumper installed
E14	3 2 1 * * *	B-7	JUMP Pin 1 to Pin 2 to allow AMPENA/signal (low true) JUMP Pin 2 to Pin 3 to allow AMPENA/signal (High true)	1-2 Jumper installed
E15	*1 *2	C-7	JUMP Pin 1 to Pin 2 to allow analog GND to come from J PWR CONN J11 (Ties amplifier and SMCC GND together. Defeats OPTO coupling.) Note that if E15 is changed, E11 and E24 must also be changed.	No jumper (open)
E16			NOT USED	
E17			NOT USED	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E18	*1 *2 *3	B-8	When short removed from 1-2 and placed on 2-3, provides Y axis unipolar error output with direction bit on BSINY at J1A18. (See E13 for X axis)	1-2 installed
E19	2 1 * *	B-6	X axis selects the SIP Polarity. I2T polarity selector open = positive input shorted = negative input	No jumper installed positive polarity
E20	2 1 * *	B-6	Y axis selects the SIP Polarity. I2T polarity selector open = positive input shorted = negative input	No jumper installed positive polarity
E21	6* *1 5* *2 4* *3	B-5	JUMP Pin 1 to Pin 6 to send optically isolated BM01 to JMACH (J1) & JPWR (J11)	2-5 jumper installed IPOS/output
		B-4	JUMP Pin 2 to Pin 5 to send optically isolated BIPOS/ to go to JMACH (J1) & JPWR (J11)	
		B-4	JUMP Pin 3 to Pin 4 to send optically isolated BIPOS to go to JMACH (J1) & JPWR (J11)	
E22	1 2 * *		When installed, allows soft following error (i10) to also drive the Following error output, FEOP/ (J1-3)	No jumper installed. Only i00 drives the FEOP/ output.
E23	1 2 3 * * *	B-4	JUMP Pin 2 to Pin 3 to allow phasing without hall effects. JUMP Pin 1 to 2 to phase with hall effects.	2-3 jumper installed

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E24	*2 *1	A-5	JUMP Pin 1 to Pin 2 to allow +14 VA to come from JPWR CONN. J11 (Ties Amplifier and SMCC power supply together. Defeats OPTO-coupling). Note that if E24 is changed, E11 and E15 must also be changed.	No jumper (open)
E25	3 2 1 * * *	A-3	JUMP Pin 1 to Pin 2 to obtain +5V to top of P01, R17 (5K) for VEE of LCD display's positive bias. JUMP Pin 3 to Pin 2 to obtain -5V to top of P01, R17 (5K) for VEE of LCD display's negative bias.	1-2 jumper installed (positive bias)
E26	*1 *2	A-3	JUMP Pin 1 to Pin 2 to allow +5V to go to J2 Pin a20 & c13 (JPAR-RS232) CONN.	No jumper installed
E27	*3 *2 *1	A-1	JUMP Pin 1 to Pin 2 to connect A5 to R/W displays & JPAR/RS232.	2-3 installed
E28 - E39			NOT USED	
E40	2 1 * *	C-6	JUMP Pin 1 to Pin 2 to obtain normal X encoder operation. Remove Jumper for digital tach (1/T). X encoder operation.	1-2 installed (X encoder not 1/T)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E41	1 2 * *	C-6	Same as E40, but for Y axis.	1-2 installed (Y encoder not 1/T)
E42	*1 *2 *3	A-6	JUMP Pin 1 to Pin 2 to obtain differential X position encoder input mode. JUMP Pin 2 to Pin 3 to obtain non-differential X encoder input mode, this will bias X encoder negative inputs To 1/2 VCC = 2.5 V. (See E50)	2-3 (X encoder non-diff.)
E43			NOT USED	
E44			NOT USED	
E45	*1 *2 *3	B-3	JUMP Pin 1 to Pin 2 to obtain differential X handwheel input mode. JUMP Pin 2 to Pin 3 to obtain non-differential X handwheel input mode, this will bias X-HW negative inputs to 1/2 VCC = 2.5V	2-3 (X - HW enc non-diff.)
E46	*1 *2 *3	A-3	Same as E45, but for Y handwheel.	2-3 (Y - HW Enc non-diff.)
E47	2 1 * *	B-3	Jumper Pin 1 to Pin 2 to tie X-handwheel and Y-handwheel together for input from a front panel at J7-16 for Chan. A. (Same as E48 for Channel B)	No jumper
E48	*1 *2	B-1	Jumper Pin 1 to Pin 2 to tie X-handwheel and Y-handwheel together for input from a front panel at J7-22 for Chan. B. (Same as E47 for Chan. A.) Note: Both E47 and E48 must be changed for proper operation.	No jumper

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E POINT	PHYSICAL LAYOUT	LOC.	DESCRIPTION	DEFAULT
E49			NOT USED	
E50	*3 *2 *1	A-7	JUMP Pin 1 to Pin 2 to obtain differential Y Position encoder input mode. (See E42)	2-3
TP1		D-7	Watchdog timer test point, Low = O.K. High = watchdog Timer time out, SMCC not functioning.	
TP2		D-6	12 MHZ Clock	
TP3		B-7	YENCA TP (Y ENCODER CHANNEL A TP)	
TP4		B-6	XENCB TP (X ENCODER CHANNEL B TP)	
TP5		B-6	XENCA TP (X ENCODER CHANNEL A TP)	
TP6		A-7	YENCB TP (Y ENCODER CHANNEL B TP)	
TP7		A-6	YENCC TP (Y ENCODER CHANNEL C TP	
TP8		A-8	PHASE 1 TP = X AXIS COMMAND SIGNAL or X AXIS PHASE 1 SIGNAL	
TP9		A-8	PHASE 3 TP = Y AXIS COMMAND SIGNAL or X AXIS PHASE 3 SIGNAL	
TP10		A-6	XENCC TP (X ENCODER CHANNEL C TP)	
TP11		A-5	INIT/ RESET INPUT TP, LOW = RESET CONDITION HIGH= NON-RESET CONDITION	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DWG. NO.	SH	REV.

REVISIONS

REV.	DESCRIPTION	DATE	APPROVED

SWITCH

CLOSED = L = 0 (PUSH TO CLOSE)

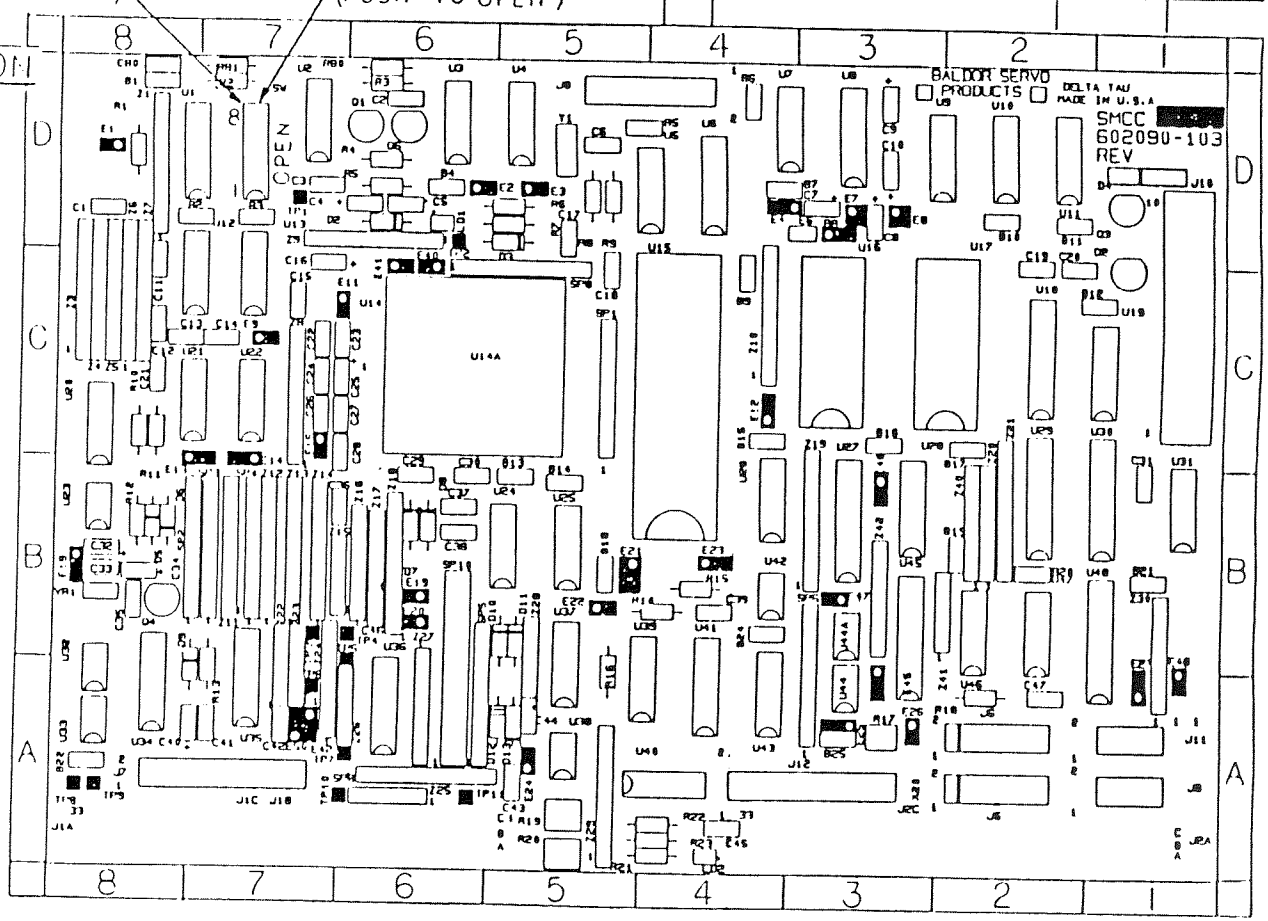
OPEN = H = 1 (PUSH TO OPEN)

CARD ADDRESS	SW1	SW2	SW3	SW4
0	1	1	1	1
1	0	1	1	1
2	1	0	1	1
3	0	0	1	1
4	1	1	0	1
5	0	1	0	1
6	1	0	0	1
7	0	0	0	1
8	1	1	1	0
9	0	1	1	0
10	1	0	1	0
11	0	0	1	0
12	1	1	0	0
13	0	1	0	0
14	1	0	0	0
15	0	0	0	0

BAUD	SW6	SW7	SW8
9600	0	0	0
4800	1	0	0
2400	0	1	0
1200	1	1	0
PARALLEL	0	0	1
300	1	0	1
38400	0	1	1
19200	1	1	1

FUNCTION	SW5
SAVE ENABLE	0
SAVE DISABLE	1

E POINT	LOCATION
1. E1	D6
2. E2	D6
3. E3	D5
4 E4	D4
5 E6	D3
6 E7	D3
7 E8	D3
8 E9	C7
9 E11	C6
10 E12	C4
11 E13	B6
12 E14	B7
13 E15	B6
14 E16	B6
15 E19	B6
16 E20	B6
17 E21	B5
18 E22	B5
19 E23	B4
20 E24	A5
21 E25	A3
22 E26	A3
23 E27	A1
24 E40	C6
25 E41	C6
26 E42	A6
27 E45	B3
28 E46	A3
29 E47	B3
30 E48	B1
31 E50	A7
32 TP1	D7
33 TP2	D6
34 TP3	B7
35 TP4	B6
36 TP5	A7
37 TP6	A7
38 TP7	A6
39 TP8	A8
40 TP9	A6
41 TP10	A6
42. TP11	A5



QTY REQD	FSCM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL SPECIFICATION

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS DECIMALS ANGLES
.XX ±
.XXX ±

CONTRACT NO.

APPROVALS	DATE
DRAWN Ano	10.25.22
CHECKED	
ISSUED	

MATERIAL

FINISH

DO NOT SCALE DRAWING

NEXT ASSY	USED ON

APPLICATION

DELTA TAU DATA SYSTEMS

SMCC E-POINTS AND TEST POINTS

SIZE	FSCM NO.	DWG. NO	REV
B		602154	B

SCALE

SHEET 1 of 1

E-PT. DESCRIPT

BISHOP GRAPHICS/ACCUPRESS REORDER NO. A-4343

SMCC CONNECTORS LISTING

TABLE OF CONTENTS

	PAGE
J1 (JMACH)	319
J2 (JPAR-RS232)	322
J5 (JDISPX)	325
J6 (JDISPY)	325
J7 (JPAN)	326
J8 (JEXP)	327
J9 (JXHW)	327
J10 (POWER CONNECTOR)	328
J11 (JYHW)	328
J12 (JOPT)	329
LIST OF MATING CONNECTORS	330

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

****

SMCC CONNECTORS LISTING

****

J1 (JMACH) Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "A"				
A1	ANA3	INPUT	ANALOG INPUT 3	{Could be used {as analog speed {command. Use 0 {to +5V for 0 to {full scale.
A2	AVSS	INPUT	ANALOG RETURN	
A3	CUR1	INPUT	CURRENT 1 FEEDBACK	Current limiting
A4	YPLL	INPUT	Y AXIS + LIMIT SW.	Low enables the move.
A5	YMIL	INPUT	Y AXIS - LIMIT SW.	Low enables the move.
A6	PL5V	I or O	+5VDC POWER SUPPLY	Could be used as input to card or output to encoders.
A7	GRND	I or O	+5 VDC RETURN	Logic GND
A8	GRND	I or O	+5 VDC RETURN	Logic GND
A9	SPAR	SPARE		
A10	YUFL/	INPUT	Y AXIS USER FLAG	Programmable polarity
A11	YHFL/	INPUT	Y AXIS HOME FLAG	Programmable polarity
A12	CHCY/	INPUT	ENCODER C CH. NEG	Y axis (Do not GND if not used)
A13	CHCY	INPUT	ENCODER C CH. POS	Y axis
A14	CHBY/	INPUT	ENCODER B CH. NEG	Y axis (Do not GND if not used)
A15	CHBY	INPUT	ENCODER B CH. POS	Y axis
A16	CHAY/	INPUT	ENCODER A CH. NEG	Y axis (Do not GND if not used)
A17	CHAY	INPUT	ENCODER A CH. POS	Y axis
A18	BSNY	OUTPUT	Y PWM SIGN	Programmable level
A19	BPO4/	OUTPUT	PWM OUT PHASE 1, 2, 3, or 4	Nonbuffered pulse
A20	BPO6/	OUTPUT	PWM OUT	
A21	BPO5/	OUTPUT	PWM OUT	
A22	SPAR	SPARE		
A23*	PHA3/	OUTPUT	PHASE 3 RETURN	{Analog differential
A24	PHA3	OUTPUT	PHASE 3 COMMAND	{Signal Out Y axis
A25	M15A	INPUT	-15V SUPPLY	From power amplifier
A26	MAI7/	INPUT	MACHINE INPUT 7	Low is a True
A27	MAI6/	INPUT	MACHINE INPUT 6	Low is a True
A28	MAI5/	INPUT	MACHINE INPUT 5	Low is a True
A29	P15A	INPUT	+15 V SUPPLY	From power amplifier
A30	AGND	INPUT	+/- 15V COMMON	From power amplifier
A31	AENA	OUTPUT	AMPLIFIER ENABLE	Programmable polarity
A32	FALT/	INPUT	AMPLIFIER FAULT	Low to A30 is Fault

*NOTE: If PHA3/ is not needed for Servo Amplifier, do not tie to AMP-common (AGND).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J1 (JMACH) Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "B"				
B1	IPOP/	OUTPUT	(see NOTE 1)	
B2	FEOP/	OUTPUT	FOL.ERR OPTO.ISOLATED	
B3	FERT	OUTPUT	ERROR RETURN	Transistor Emitter
B4	IPRT	OUTPUT	IN POSITION RETURN	Transistor Emitter
B5	SPAR	SPARE		
B6	PL5V	INPUT	+5 VDC POWER SUPPLY	
B7	GRND	INPUT	5 VDC COMMON	
B8	GRND	INPUT	5 VDC COMMON	
B9	M15V	INPUT	-15 V POWER SUPPLY	Could be used to
B10	P15V	INPUT	+15 V POWER SUPPLY	power ACC14, ACC15 boards if connected to SMCC
B11	F1LD/	OUTPUT	FOLLOWING ERROR 1	Low lights LED
B12	F2LD/	OUTPUT	FOLLOWING ERROR 2	Low lights LED
B13	ITLD/	OUTPUT	CURRENT LIMIT IND.	Low lights LED
B14	SPAR	SPARE		
B15	ERLD/	OUTPUT	ERROR INDICATOR	Low lights LED
B16	EPLD/	OUTPUT	END OF PROGRAM IND.	Low lights LED
B17	IPLD/	OUTPUT	IN POSITION IND.	Low lights LED
B18	BFLD/	OUTPUT	BUFFER FULL IND.	Low lights LED
B19	INIT/	INPUT	RESETS SMCC	Low is a Reset
B20	SPAR	SPARE		
B21	HWYD/	INPUT	HANDWH. Y DISABLE	Low disables
B22	HWXD/	INPUT	HANDWH. X DISABLE	Low disables
B23	HOME/	INPUT	GO HOME COMMAND	Low is Go Home
B24	HOLD/	INPUT	HOLD MOTION	Low is Hold
B25	STEP/	INPUT	STEP THROUGH PROGRAM	Low is Step
B26	STOP/	INPUT	STOP PROGRAM RUN	Low is Stop
B27	PREJ/	INPUT	RET. TO PREJOG POS.	Low is Return
B28	STRT/	INPUT	START PRORGRAM RUN	Low is Start
B29	YJPL/	INPUT	JOG Y AXIS IN +DIR.	Low is Jog Y+
B30	XJPL/	INPUT	JOG X AXIS IN +DIR.	Low is Jog X+
B31	YJMI/	INPUT	JOG Y AXIS IN -DIR.	Low is Jog Y-
B32	XJMI/	INPUT	JOG X AXIS IN -DIR.	Low is Jog X-

NOTE 1: Optically isolated IPOP/ function could be changed by E21 to the following:

a) In position for this card "Low is true".
b) In position for this card "High is true".
c) MO1/ output "Low is true".

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J1 (JMACH) Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "C"				
C1	ANA2	INPUT	ANALOG INPUT 2	{Could be used as {analog speed command. {Use 0 to +5V for 0 {to full scale.
C2	AVSS	INPUT	ANALOG RETURN	
C3	CURO	INPUT	CURRENT 0 FEEDBACK	Current Limiting
C4	XPLL	INPUT	X AXIS + LIMIT SW.	Low enables the move.
C5	XMIL	INPUT	X AXIS - LIMIT SW.	Low enables the move.
C6	PL5V	I or O	+5 VDC POWER SUPPLY	Could be used as input to card or output to encoders.
C7	GRND	I or O	+5 VDC RETURN	Logic GND
C8	GRND	I or O	+5 VDC RETURN	Logic GND
C9	SPAR	SPARE		
C10	XUFL/	INPUT	X AXIS USER FLAG	Programmable polarity
C11	XHFL/	INPUT	X AXIS HOME FLAG	Programmable polarity
C12	CHCX/	INPUT	ENCODER C CH. NEG	X Axis (Do not GND if not used)
C13	CHCX	INPUT	ENCODER C CH. POS	X Axis
C14	CHBX/	INPUT	ENCODER B CH. NEG	X Axis (Do not GND if not used)
C15	CHBX	INPUT	ENCODER B CH. POS	X Axis
C16	CHAX/	INPUT	ENCODER A CH. NEG	X Axis (Do not GND if not used)
C17	CHAX	INPUT	ENCODER A CH. POS	X Axis
C18	BSNX	OUTPUT	X PWM SIGN	Programmable level
C19	BPO3/	OUTPUT	PWM OUT PHASE 1, 2, 3, or 4	Nonbuffered pulse
C20	BPO2/	OUTPUT	PWM OUT	
C21	BPO1/	OUTPUT	PWM OUT	
C22	SPAR	SPARE		
C23*	PHA1/	OUTPUT	PHASE 1 RETURN	{Analog differential
C24	PHA1	OUTPUT	PHASE 1 COMMAND	{signal out X axis
C25	M15A	INPUT	-15 V SUPPLY	From power amplifier
C26	MAI7/	INPUT	MACHINE INPUT 7	Low is a True
C27	MAI6/	INPUT	MACHINE INPUT 6	Low is a True
C28	MAI5/	INPUT	MACHINE INPUT 5	Low is a True
C29	P15A	INPUT	+15 V SUPPLY	From power amplifier
C30	AGND	INPUT	+/- 15 V COMMON	From power amplifier
C31	AENA	OUTPUT	AMPLIFIER ENABLE	Programmable polarity
C32	FALT/	INPUT	AMPLIFIER FAULT	Low to C30 is Fault

*NOTE: If PHA1/ is not needed for Servo Amplifier, do not tie to Amp-Common (AGND)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J2 JPAR-RS232 Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "A"				
A1	GRND	COMMON	SMCC COMMON	
A2	GRND	COMMON	SMCC COMMON	
A3	GRND	COMMON	SMCC COMMON	
A4	INIT/	INPUT	RESETS SMCC	Low is a "Reset"
A5	EROR/	OUTPUT	SIGNALS ERROR	Low is an "Error"
A6	BUFU/	OUTPUT	BUFFER FULL	Low is "Buffer full"
A7	GRND	COMMON	SMCC COMMON	
A8	DASR	RS232	DATA SET READY	
A9	CLTS	RS 232	CLEAR TO SEND	9A & 10A functions
A10	BUFU/	OUTPUT	BUFFER FULL	Low is "Buffer full"
A11	RXD/	RS 232	RECEIVE DATA	(E7 could exchange
A12	TXD/	RS 232	TRANSMIT DATA	(11A and 12A functions
A13	CHAS		CHASSIS	Not tied to common
A14	SPAR	SPARE		
A15	BUFU/	OUTPUT	BUFFER FULL	Low is "Buffer full"
A16	CHAS		CHASSIS	Not tied to common
A17	GRND	COMMON	SMCC COMMON	
A18	IPOS	OUTPUT	IN POSITION (BUSSED)	High is "In position"
A19	GRND	COMMON	SMCC COMMON	
A20	SP5V	OUTPUT	+5 VDC SUPPLY	Deactivated by E26
A21	IPOS/	OUTPUT	IN POSITION (BUSSED)	Low is "In position"
A22	BUFU	OUTPUT	BUFFER FULL	High is buffer full
A23	ACKN/	OUTPUT	HANDSHAKING	Low is handshake
A24	DAT7	BIDIRECT.	DATA 7	Parallel Interface
A25	DAT6	BIDIRECT.	DATA 6	Parallel Interface
A26	DAT5	BIDIRECT.	DATA 5	Parallel Interface
A27	DAT4	BIDIRECT.	DATA 4	Parallel Interface
A28	DAT3	BIDIRECT.	DATA 3	Parallel Interface
A29	DAT2	BIDIRECT.	DATA 2	Parallel Interface
A30	DAT1	BIDIRECT.	DATA 1	Parallel Interface
A31	DAT0	BIDIRECT.	DATA 0	Parallel Interface
A32	STRB/	INPUT	STROBE	Low is a "Strobe"

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J2 JPAR-RS232 Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "B"				
B1	ENAX	INPUT	X HANDHW ENC.A CH.	5V TTL square pulse
B2	ENBX	INPUT	X HANDWH ENC.B CH.	5V TTL square pulse
B3	ENAY	INPUT	Y HANDWH ENC.A CH.	5V TTL square pulse
B4	ENBY	INPUT	Y HANDWH ENC.B CH.	5V TTL square pulse
B5	HWXD/	INPUT	X HANDWH DISABLE	Low is "Disable"
B6	HWYD/	INPUT	Y HANDWH DISABLE	Low is "Disable"
B7	DAB7	BIDIRECT.	DISPLAY DATA 7	5V TTL
B8	DAB6	BIDIRECT.	DISPLAY DATA 6	5V TTL
B9	DAB5	BIDIRECT.	DISPLAY DATA 5	5V TTL
B10	DAB4	BIDIRECT.	DISPLAY DATA 4	5V TTL
B11	DAB3	BIDIRECT.	DISPLAY DATA 3	5V TTL
B12	DAB2	BIDIRECT.	DISPLAY DATA 2	5V TTL
B13	DAB1	BIDIRECT.	DISPLAY DATA 1	5V TTL
B14	DAB0	BIDIRECT.	DISPLAY DATA 0	5V TTL
B15	EX	OUTPUT	X DISPLAY ENABLE	High is "Enable"
B16	RDWR	BIDIRECT.	READ OR WRITE	E27 selects function
B17	RSTR	OUTPUT	READ STROBE	TTL
B18	CONT	OUTPUT	CONTRAST ADJUST.	0 to 5 VDC
B19	EY	OUTPUT	Y DISPLAY ENABLE	High is "Enable"
B20	PL5V	OUTPUT	+5V	Power supply out
B21	MAO1/	OUTPUT	MACHINE OUTPUT 1	Open collector TTL
B22	MAO2/	OUTPUT	MACHINE OUTPUT 2	Low is a true
B23	MAO3/	OUTPUT	MACHINE OUTPUT 3	Low is a true
B24	MAO4/	OUTPUT	MACHINE OUTPUT 4	Low is a true
B25	MAO5/	OUTPUT	MACHINE OUTPUT 5	Low is a true
B26	MAI1/	INPUT	MACHINE INPUT 1	5V TTL Low is a true
B27	MAI2/	INPUT	MACHINE INPUT 2	5V TTL Low is a true
B28	MAI3/	INPUT	MACHINE INPUT 3	5V TTL Low is a true
B29	MAI4/	INPUT	MACHINE INPUT 4	5V TTL Low is a true
B30	MAI5/	INPUT	MACHINE INPUT 5	5V TTL Low is a true
B31	MAI6/	INPUT	MACHINE INPUT 6	5V TTL Low is a true
B32	MAI7/	INPUT	MACHINE INPUT 7	5V TTL Low is a true

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J2 JPAR-RS232 Connector (96 Pin DIN Connector)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
Row "C"				
C1	PL5V	OUTPUT	+5V DC	Power supply out
C2	SYNC/	OUTPUT	SYNCHRONIZER	Low True from A0 card
C3	BRTS	OUTPUT	BUFFERED RTS	For RS232 daisychain
C4	BTXD	OUTPUT	BUFFERED TXD	For RS232 daisychain
C5	IPOS	OUTPUT	IN POSITION (BUSSED)	High is "In Position"
C6	SPAR	SPARE		
C7	DATR	RS232	DATA TERMINAL READY	
C8	SPAR	SPARE		
C9	ENAX/	INPUT	X HWL ENC.A CH NEG	5V TTL
C10	EWBX/	INPUT	X HWL ENC.B CH NEG	5V TTL
C11	ENAY/	INPUT	Y HWL ENC.A CH NEG	5V TTL
C12	ENBY/	INPUT	Y HWL ENC.B CH NEG	5V TTL
C13	SP5V	OUTPUT	+5 VDC SUPPLY	Deactivated by E26
C14	SPAR	SPARE		
C15	SYNC/	OUTPUT	SYNCHRONIZER	Low true from A0 card
C16	HSB/	INPUT	READ STROBE	Low is "Read Strobe"
C17	ORDY/	OUTPUT	OUTPUT READY	Low is "Read Strobe"
C18	GRND	COMMON	SMCC COMMON	Low is "Read Strobe"
C19	EROR/	OUTPUT	SIGNALS ERROR	Low is an "Error"
C20	INIT/	INPUT	RESETS SMCC	Low is a "Reset"
C21	GRND	COMMON	SMCC COMMON	
C22	GRND	COMMON	SMCC COMMON	
C23	GRND	COMMON	SMCC COMMON	
C24	SEL7	OUTPUT	DATA7 OUTPUT	Scanner output for reading thumbwhl switches
C25	SEL6	OUTPUT	DATA6 OUTPUT	" " "
C26	SEL5	OUTPUT	DATA5 OUTPUT	" " "
C27	SEL4	OUTPUT	DATA4 OUTPUT	" " "
C28	SEL3	OUTPUT	DATA3 OUTPUT	" " "
C29	SEL2	OUTPUT	DATA2 OUTPUT	" " "
C30	SEL1	OUTPUT	DATA1 OUTPUT	" " "
C31	SEL0	OUTPUT	DATA0 OUTPUT	" " "
C32	GRND	COMMON	SMCC COMMON	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J5 JDISPX Connector (14 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	PL5V	OUTPUT	+5V POWER	
2	GRND	COMMON	SMCC COMMON	
3	RSTR	OUTPUT	READ STROBE	TTL
4	CONT	OUTPUT	CONTRAST ADJUST. VEE	0 to 10 VDC
5	EX	OUTPUT	X DISPLAY ENABLE	High is "Enable"
6	RDWR	BIDIRECT.	READ OR WRITE	E27 selects func.
7	DAB1	BIDIRECT.	DISPLAY DATA1	5V TTL
8	DAB0	BIDIRECT.	DISPLAY DATA0	5V TTL
9	DAB3	BIDIRECT.	DISPLAY DATA3	5V TTL
10	DAB2	BIDIRECT.	DISPLAY DATA2	5V TTL
11	DAB5	BIDIRECT.	DISPLAY DATA5	5V TTL
12	DAB4	BIDIRECT.	DISPLAY DATA4	5V TTL
13	DAB7	BIDIRECT.	DISPLAY DATA7	5V TTL
14	DAB6	BIDIRECT.	DISPLAY DATA6	5V TTL

J6 JDISPY Connector (14 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	PL5V	OUTPUT	+5V POWER	
2	GRND	COMMON	SMCC COMMON	
3	RSTR	OUTPUT	READ STROBE	TTL
4	CONT	OUTPUT	CONTRAST ADJUST.	0 to 10 VDC
5	EY	OUTPUT	X DISPLAY ENABLE	High is "Enable"
6	RDWR	BIDIRECT.	READ OR WRITE	E27 selects func.
7	DAB1	BIDIRECT.	DISPLAY DATA1	5V TTL
8	DAB0	BIDIRECT.	DISPLAY DATA0	5V TTL
9	DAB3	BIDIRECT.	DISPLAY DATA3	5V TTL
10	DAB2	BIDIRECT.	DISPLAY DATA2	5V TTL
11	DAB5	BIDIRECT.	DISPLAY DATA5	5V TTL
12	DAB4	BIDIRECT.	DISPLAY DATA4	5V TTL
13	DAB7	BIDIRECT.	DISPLAY DATA7	5V TTL
14	DAB6	BIDIRECT.	DISPLAY DATA6	5V TTL

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J7 JPAN Connector (26 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	PL5V	OUTPUT	+5V POWER	For remote panel
2	GRND	COMMON	SMCC COMMON	
3	YJMI/	INPUT	JOG Y AXIS IN -DIR	Low is Jog Y-
4	XJMI/	INPUT	JOG X AXIS IN -DIR	Low is Jog X-
5	YJPL/	INPUT	JOG Y AXIS IN +DIR	Low is Jog Y+
6	XJPL/	INPUT	JOG X AXIS IN +DIR	Low is Jog X+
7	PREJ/	INPUT	RET. TO PREJOG POS.	Low is Return
8	STRT/	INPUT	START PROGRAM RUN	Low is Start
9	STEP/	INPUT	STEP THROUGH PROGRAM	Low is Step
10	STOP/	INPUT	STOP PROGRAM RUN	Low is Stop
11	HOME/	INPUT	GO HOME COMMAND	Low is Go Home
12	HOLD/	INPUT	HOLD MOTION	Low is Hold
13	HWYD/	INPUT	HANDWH. Y DISABLE	Low disables
14	HWXD/	INPUT	HANDWH. X DISABLE	Low disables
15	INIT/	INPUT	RESETS SMCC	Low is a Reset
16	ENAX	INPUT	X HANDWH ENC. A CH.	5V TTL sq.pulse
17	IPLD/	OUTPUT	IN POSITION IND.	Low lights LED
18	BFLD/	OUTPUT	BUFFER FULL IND.	Low lights LED
19	ERLD/	OUTPUT	ERROR INDICATOR	Low lights LED
20	EPLD/	OUTPUT	END OF PROGRAM IND.	Low lights LED
21	ITLD/	OUTPUT	CURRENT LIMIT IND.	Low lights LED
22	ENBX	OUTPUT	Y HANDWH ENC. B CH.	5V TTL sq.pulse
23	FILD/	OUTPUT	FOLLOWING ERROR 1	5V TTL sq.pulse
24	F2LD/	OUTPUT	FOLLOWING ERROR 2	5V TTL sq.pulse
25	PL5V	OUTPUT	+5V POWER	For remote panel
26	GRND	COMMON	SMCC COMMON	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J8 JEXP Connector (20 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	-15V	OUTPUT	-15V POWER	For extension card
2	M RESET	OUTPUT	RESET	For extension card
3	+15V	OUTPUT	+15V POWER	For extenstion card
4	IOWR/	OUTPUT	WRITE DATA	Low is Write
5	GRND	OUTPUT	SMCC COMMON	
6	MD3	BIDIRECT.	ADDRESS DATA3	
7	GRND	COMMON	SMCC COMMON	
8	MD2	BIDIRECT.	ADDRESS DATA2	
9	+5V	OUTPUT	+5V POWER	
10	MD4	BIDIRECT.	ADDRESS DATA4	
11	+5V	OUTPUT	+5V POWER	
12	MD1	BIDIRECT.	ADDRESS DATA1	
13	IORD/	OUTPUT	READ DATA	Low is Read
14	MD5	BIDIRECT.	ADDRESS DATA5	
15	MWT/SV2	INPUT/ OUTPUT	GENERAL INPUT/ SERVO OUTPUT 2	(Not used in SMCC) (Controlled by E4)
16	MD0	BIDIRECT.	ADDRESS DATA0	
17	MALE	OUTPUT	ADDRESS LATCH ENABLE	
18	MD6	BIDIRECT.	ADDRESS DATA6	
19	MMCS	OUTPUT	BOARD SELECT	
20	MD7	BIDIRECT.	ADDRESS DATA7	

J9 JXHW Connector (10 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	ENAX	INPUT	X HANDWHEEL A CH.	5V TTL sq. pulse
2	PL5V	OUTPUT	+5V POWER	
3	GRND	COMMON	SMCC COMMON	
4	HWAX/	INPUT	X HNDWH. A CH. NEG.	5V TTL sq. pulse
5	HWBX/	INPUT	X HNDWH. B CH. NEG.	5V TTL sq. pulse
6	GRND	Common	SMCC COMMON	
7	PL5V	OUTPUT	+5V POWER	
8	ENBX	INPUT	X HANDWHEEL B CH.	5V TTL sq. pulse
9	PL5V	OUTPUT	+5V POWER	
10	SPAR	SPARE		

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J10 Power Connector (10 Pin Terminal Strip)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	GRND	INPUT	5V DC COMMON	Power input
2	PL5V	INPUT	+5V DC SUPPLY	Power input
3	P15V	INPUT	+15V DC SUPPLY	Power input
4	M15V	INPUT	-15V DC SUPPLY	Power input
5	IPOP/	OUTPUT	{MULTIPLE FUNCTION	Opto-isolated
6	IPRT	OUTPUT	{see NOTE 1	Opto-isolated
7	FEOP	OUTPUT	FOLLOWING ERROR	Opto-isolated
8	FERT	OUTPUT	FOLL. ERROR RETURN	Opto-isolated
9	SPAR	SPARE		
10	SPAR	SPARE		

J11 JYHW Connector (10 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	ENAY	INPUT	Y HANDHWEEL A CH.	5V TTL sq. pulse
2	PL5V	OUTPUT	+5V POWER	
3	GRND	COMMON	SMCC COMMON	
4	HWAY/	INPUT	Y HANDWHEEL A CH.	5V TTL sq. pulse
5	HWBY/	INPUT	Y HANDWHEEL B CH.	5V TTL sq. pulse
6	GRND	COMMON	SMCC COMMON	
7	PL5V	OUTPUT	+5V POWER	
8	ENBY	INPUT	Y HANDWHEEL B CH.	5V TTL sq. pulse
9	PL5V	OUTPUT	+5V POWER	
10	SPAR	SPARE		

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J12 JOPT Connector (26 Pin Header)

PIN #	SYMBOL	FUNCTION	DESCRIPTION	NOTES
1	MAI7/	INPUT	MACHINE INPUT 7	5V TTL, HIGH is True
2	GRND	COMMON	SMCC COMMON	
3	MAI6/	INPUT	MACHINE INPUT 6	5V TTL, HIGH is True
4	GRND	COMMON	SMCC COMMON	
5	MAI5/	INPUT	MACHINE INPUT 5	5V TTL, HIGH is True
6	GRND	COMMON	SMCC COMMON	
7	MAI4/	INPUT	MACHINE INPUT 4	5V TTL, LOW is True
8	GRND	COMMON	SMCC COMMON	
9	MAI3/	INPUT	MACHINE INPUT 3	5V TTL, LOW is True
10	GRND	COMMON	SMCC COMMON	
11	MAI2/	INPUT	MACHINE INPUT 2	5V TTL, LOW is True
12	GRND	COMMON	SMCC COMMON	
13	MAI1/	INPUT	MACHINE INPUT 1	5V TTL, LOW is True
14	GRND	COMMON	SMCC COMMON	
15	MAO5/	OUTPUT	MACHINE OUTPUT 5	Open Collector, True LOW true
16	GRND	COMMON	SMCC COMMON	
17	MAO4/	OUTPUT	MACHINE OUTPUT 4	Open Collector, True LOW true
18	GRND	COMMON	SMCC COMMON	
19	MAO3/	OUTPUT	MACHINE OUTPUT 3	Open Collector, True LOW true
20	GRND	COMMON	SMCC COMMON	
21	MAO2/	OUTPUT	MACHINE OUTPUT 2	Open Collector, True LOW true
22	GRND	COMMON	SMCC COMMON	
23	MAO1/	OUTPUT	MACHINE OUTPUT 1	Open Collector, True LOW true
24	GRND	COMMON	SMCC COMMON	
25	PL5V	OUTPUT	+5V POWER	
26	GRND	COMMON	SMCC COMMON	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC CONNECTORS	LIST OF MATING CONNECTORS
J1 (JMACH) -	AMLAN P/N C96F3LA+B+C DELTA TAU ASSEMBLY P/N 602065-101 or P/N 602065-100
J2 (JPAR-RS232) -	AMLAN P/N C64F3SCHM.Z.A+C 3M P/N 6964-0000-EF or equivalent
J3 -	NOT USED
J4 -	NOT USED
J5 (JDISPX) -	H and T P/N 4003-14-00-P5 3M P/N 3385-6600 or equivalent
J6 (JDISPY) -	H AND T P/N 4003-14-00-P5 3M P/N 3385-6600 or equivalent
J7 (JPAN) -	H and T P/N 4003-26-00-P5 3M P/N 3399-6600 or equivalent
J8 (JEXP) -	H and T P/N 4003-20-00-P5 3M P/N 3421-6600 or equivalent
J9 (JXHW) -	H and T P/N 4003-10-00-P5 3M P/N 3473-6600 or equivalent
J10 (POWER) -	NONE
J11 (JYHW) -	H and T P/N 4003-10-00-P5 3M P/N 3473-6600 or equivalent
J12 (JOPT) -	H and T P/N 4003-26-00-P5 3M P/N 3399-6600 or equivalent

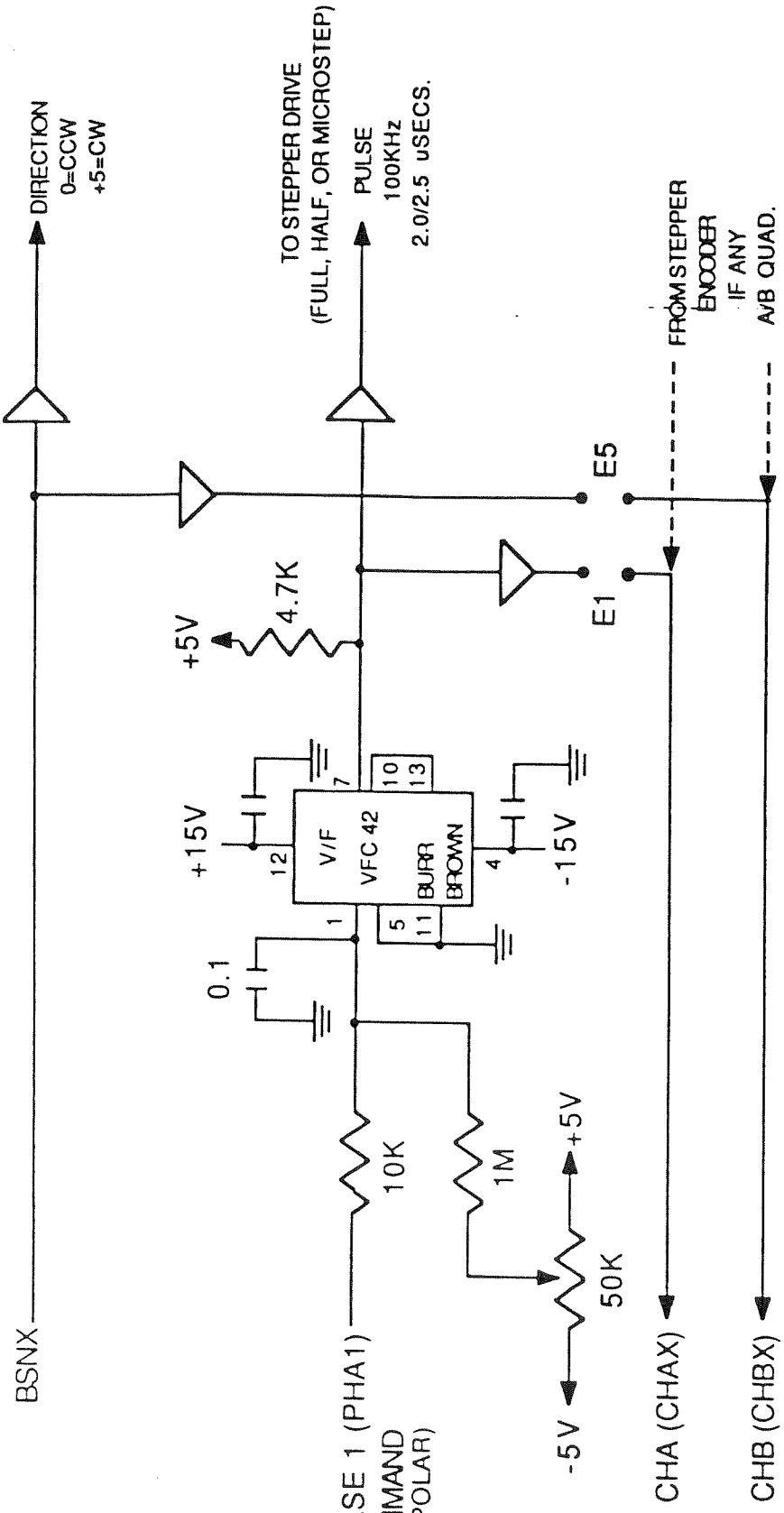
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

CONNECTION DIAGRAMS

TABLE OF CONTENTS

	PAGE
STEPPER MOTOR DRIVER CONVERTER	.332
RS232 I/O CONNECTOR (FLAT CABLE)	.332A
PARALLEL I/O CONNECTOR WIRING.	.333
"JMACH" TERMINAL BLOCK	.334
SMCC CONNECTOR LOCATIONS	.335
TWO-AXIS POSITIONING SYSTEM.	.336
TWO-AXIS BRUSH TYPE POSITIONING SYSTEM	.337
SINGLE AXIS 3-PHASE POSITIONING SYSTEM	.338
LCD DISPLAY.	.339
SMCC TO SERVO AMPLIFIER CONNECTIONS.	.340
HANDWHEEL ENCODER CONNECTIONS.	.341
MOTOR ENCODER CONNECTIONS.	.342
ANALOG INPUT POTENTIOMETER CONNECTIONS	.342A

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



SMCC SERVO COMMAND OUTPUT
TO
STEPPER MOTOR DRIVER CONVERTER

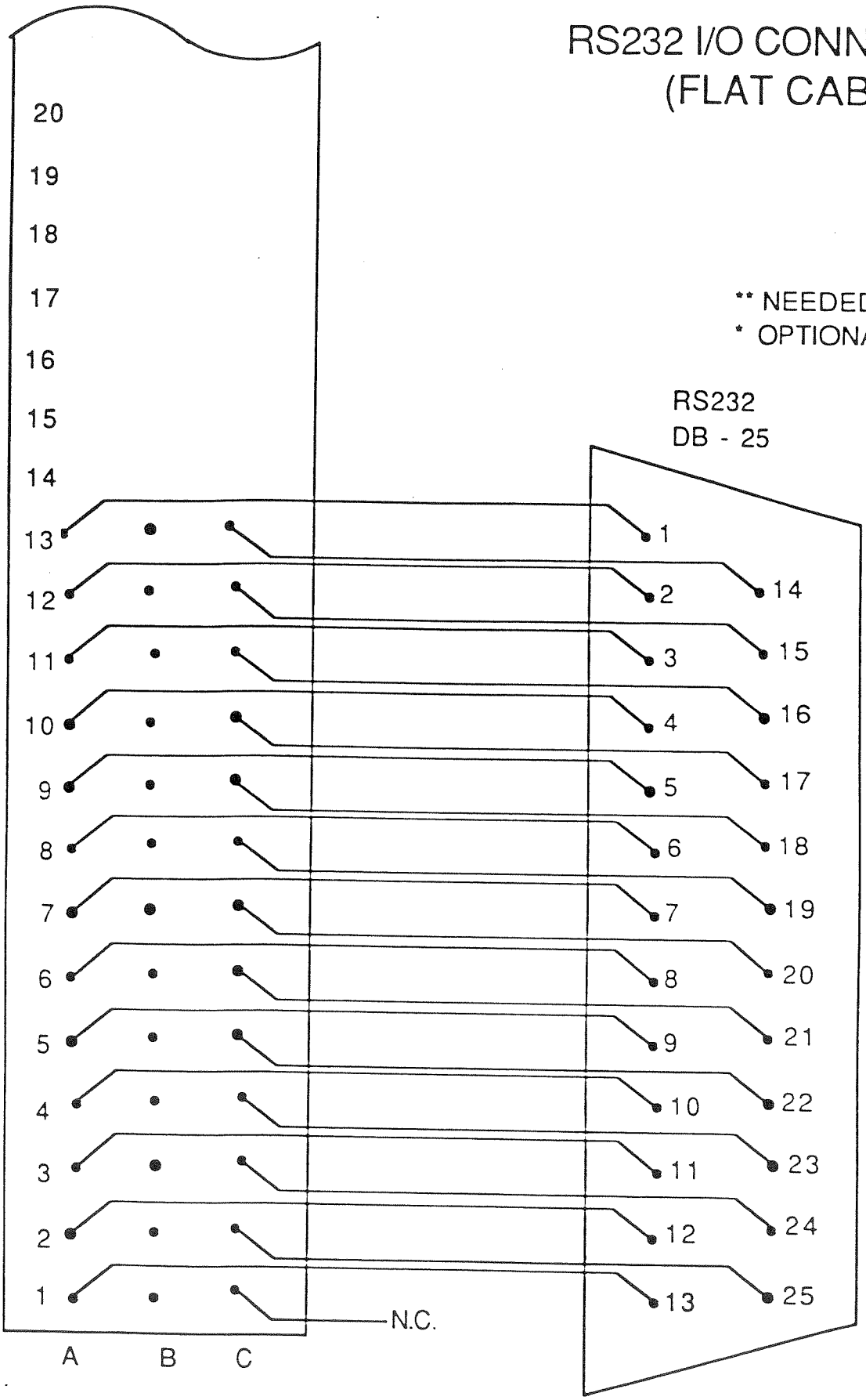
(STEPPER)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

JPAR - RS232

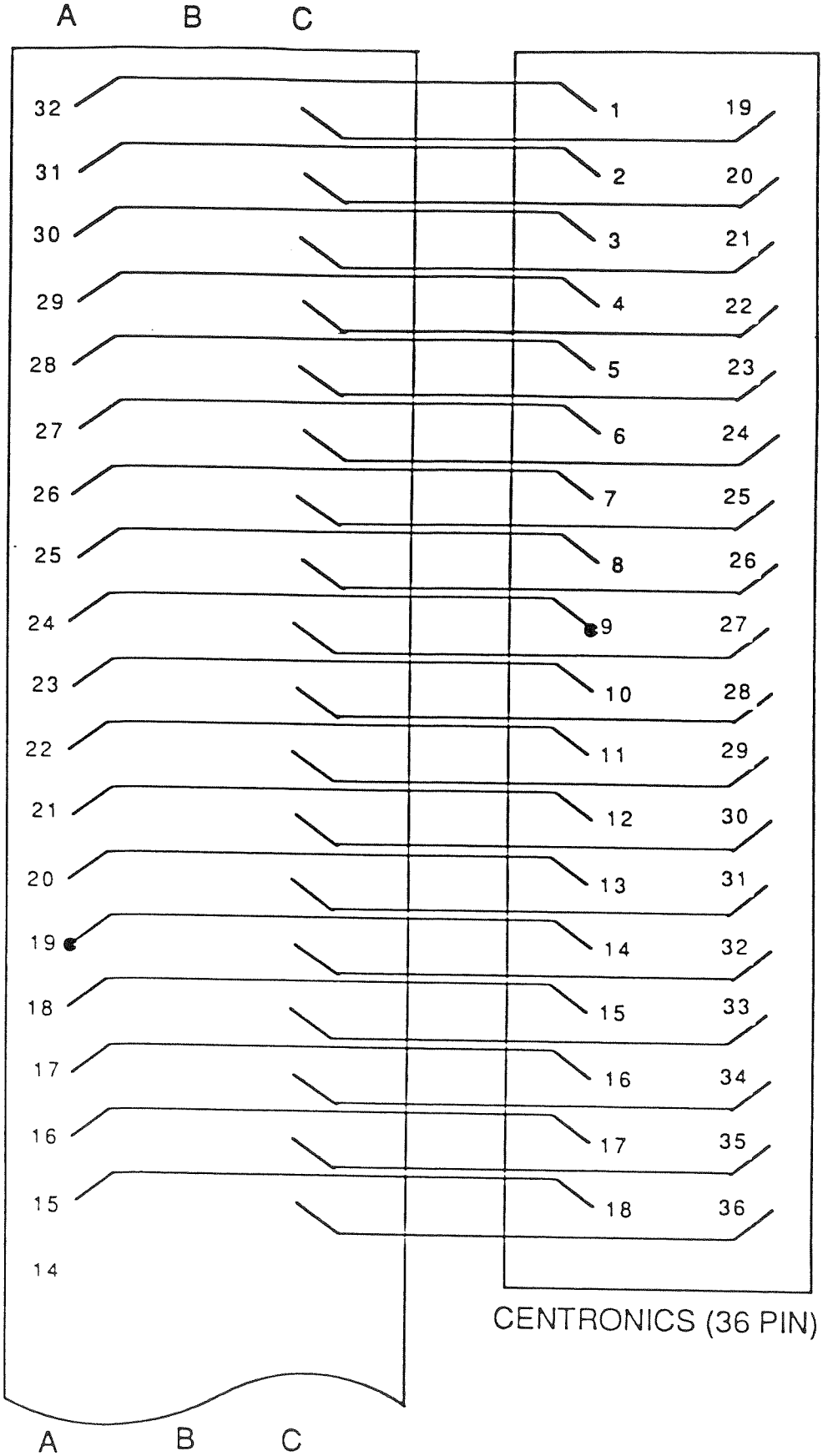
RS232 I/O CONNECTOR (FLAT CABLE)

** NEEDED FOR RS232
* OPTIONAL FOR HOST ONLY.



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PARALLEL I/O CONNECTOR WIRING

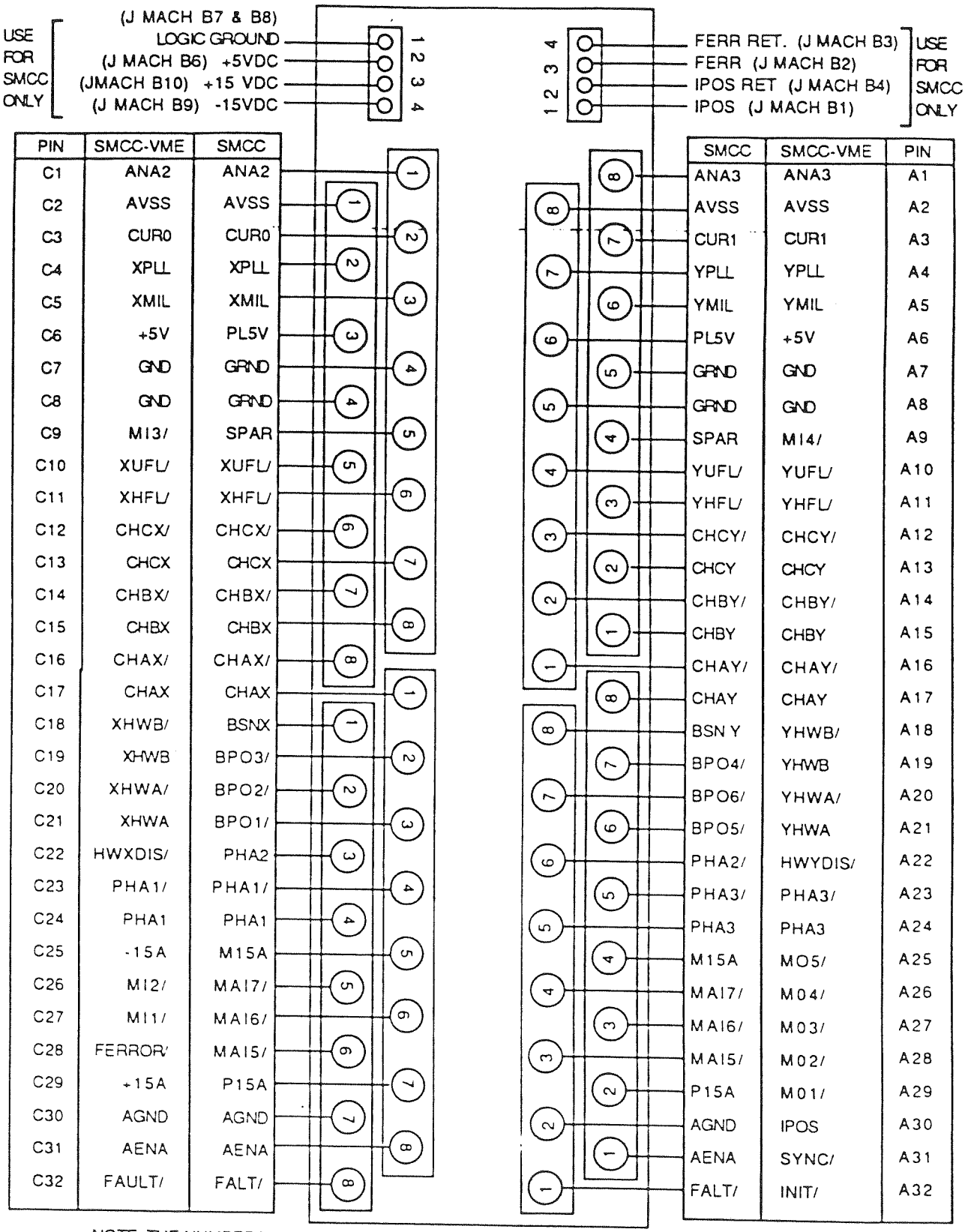


1 = STRB/
2 = DAT0
3 = DAT1
4 = DAT2
5 = DAT3
6 = DAT4
7 = DAT5
8 = DAT6
9 = DAT7
10 = ACKN/
11 = BUFU
12 = IPOS/
13 = SPSU
14 = GND
15 = IPOS
16 = GND
17 = CHAS
18 = BUFU/
19 = GND
20 = SEL0
21 = SEL1
22 = SEL2
23 = SEL3
24 = SEL4
25 = SEL5
26 = SEL6
27 = SEL7
28 = GND
29 = GND
30 = GND
31 = INIT/
32 = EROR/
33 = GND
34 = ORDY/
35 = HSB/
36 = SYNC/
36 = SYNC/

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC ACCESSORY 8H (JMACH)

(VIEWED FROM TERMINAL BLOCK SIDE)

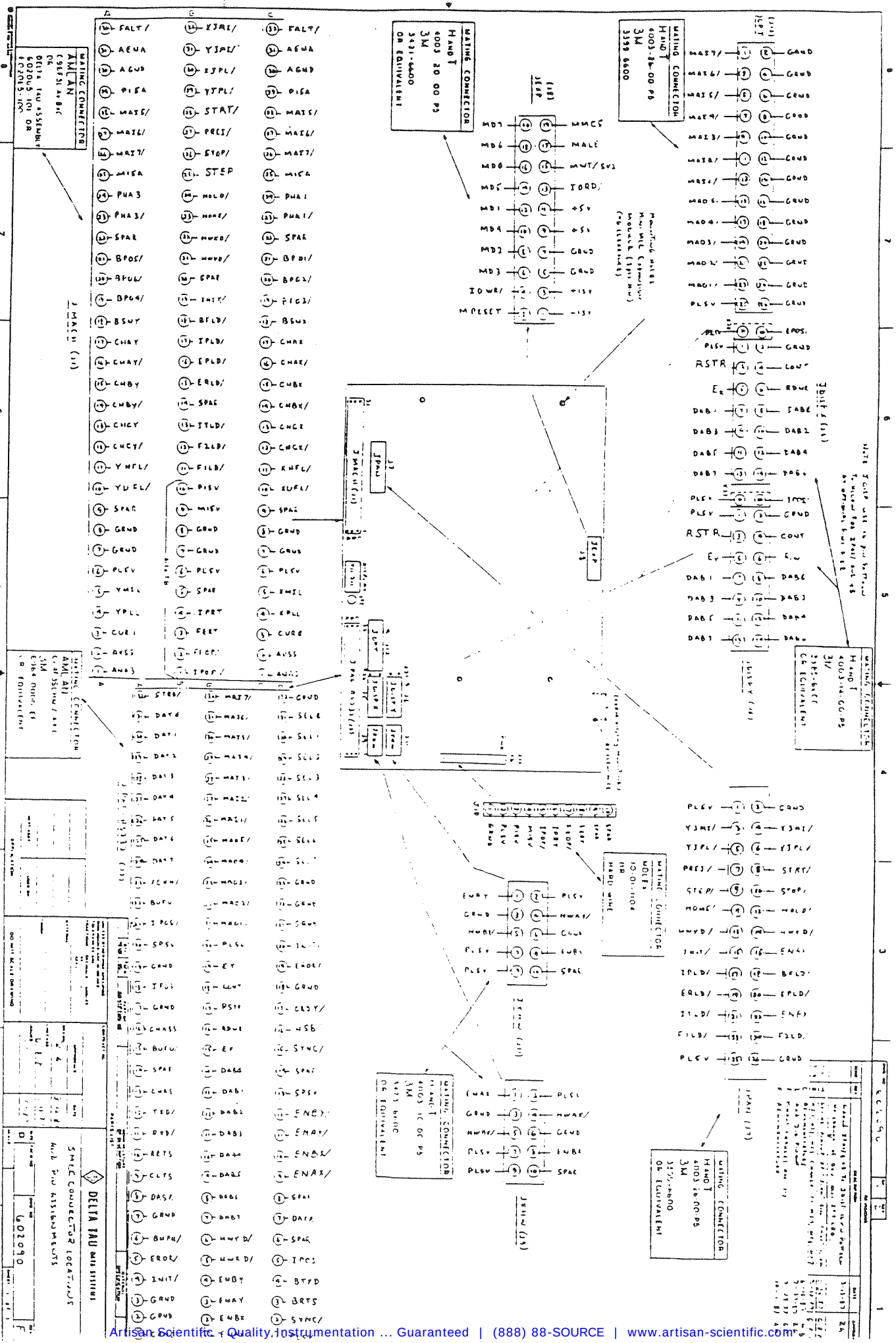


PIN	SMCC-VME	SMCC
C1	ANA2	ANA2
C2	AVSS	AVSS
C3	CUR0	CUR0
C4	XPLL	XPLL
C5	XMIL	XMIL
C6	+5V	PL5V
C7	GND	GRND
C8	GND	GRND
C9	MI3/	SPAR
C10	XUFL/	XUFL/
C11	XHFL/	XHFL/
C12	CHCX/	CHCX/
C13	CHCX	CHCX
C14	CHBX/	CHBX/
C15	CHBX	CHBX
C16	CHAX/	CHAX/
C17	CHAX	CHAX
C18	XHWB/	BSNX
C19	XHWB	BPO3/
C20	XHWA/	BPO2/
C21	XHWA	BPO1/
C22	HWXDIS/	PHA2
C23	PHA1/	PHA1/
C24	PHA1	PHA1
C25	-15A	M15A
C26	MI2/	MAI7/
C27	MI1/	MAI6/
C28	FERROR/	MAI5/
C29	+15A	P15A
C30	AGND	AGND
C31	AENA	AENA
C32	FAULT/	FALT/

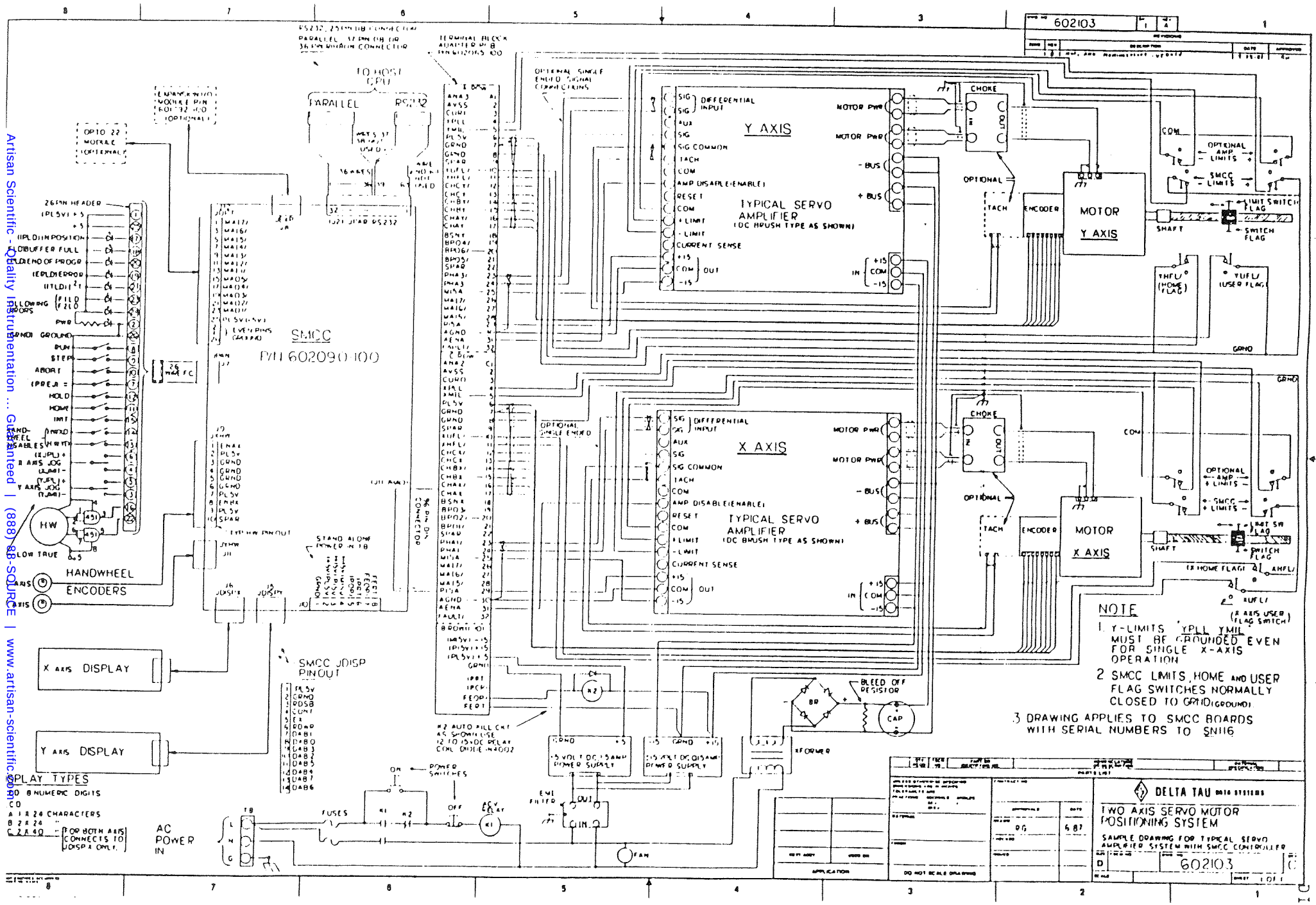
SMCC	SMCC-VME	PIN
ANA3	ANA3	A1
AVSS	AVSS	A2
CUR1	CUR1	A3
YPLL	YPLL	A4
YMIL	YMIL	A5
PL5V	+5V	A6
GRND	GND	A7
GRND	GND	A8
SPAR	MI4/	A9
YUFL/	YUFL/	A10
YHFL/	YHFL/	A11
CHCY/	CHCY/	A12
CHCY	CHCY	A13
CHBY/	CHBY/	A14
CHBY	CHBY	A15
CHAY/	CHAY/	A16
CHAY	CHAY	A17
BSN Y	YHWB/	A18
BPO4/	YHWB	A19
BPO6/	YHWA/	A20
BPO5/	YHWA	A21
PHA2/	HWYDIS/	A22
PHA3/	PHA3/	A23
PHA3	PHA3	A24
M15A	MO5/	A25
MAI7/	M04/	A26
MAI6/	M03/	A27
MAI5/	M02/	A28
P15A	M01/	A29
AGND	IPOS	A30
AENA	SYNC/	A31
FALT/	INIT/	A32

NOTE: THE NUMBERS IN CIRCLES ARE THE USUAL NUMBERS THAT APPEAR ON THE SCREW TERMINALS. IT IS POSSIBLE THAT OTHER NUMBERING, OR NO NUMBERING MAY BE USED

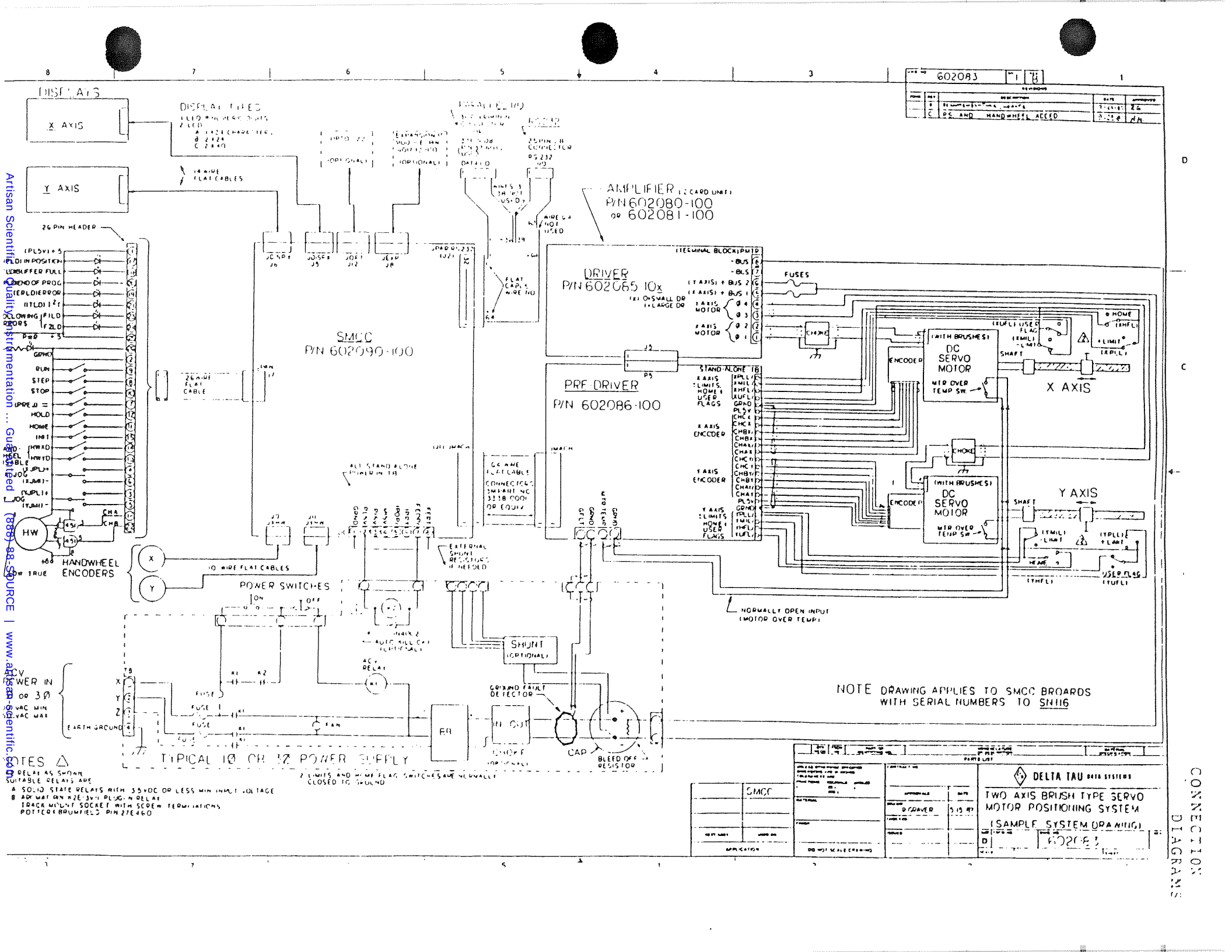
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



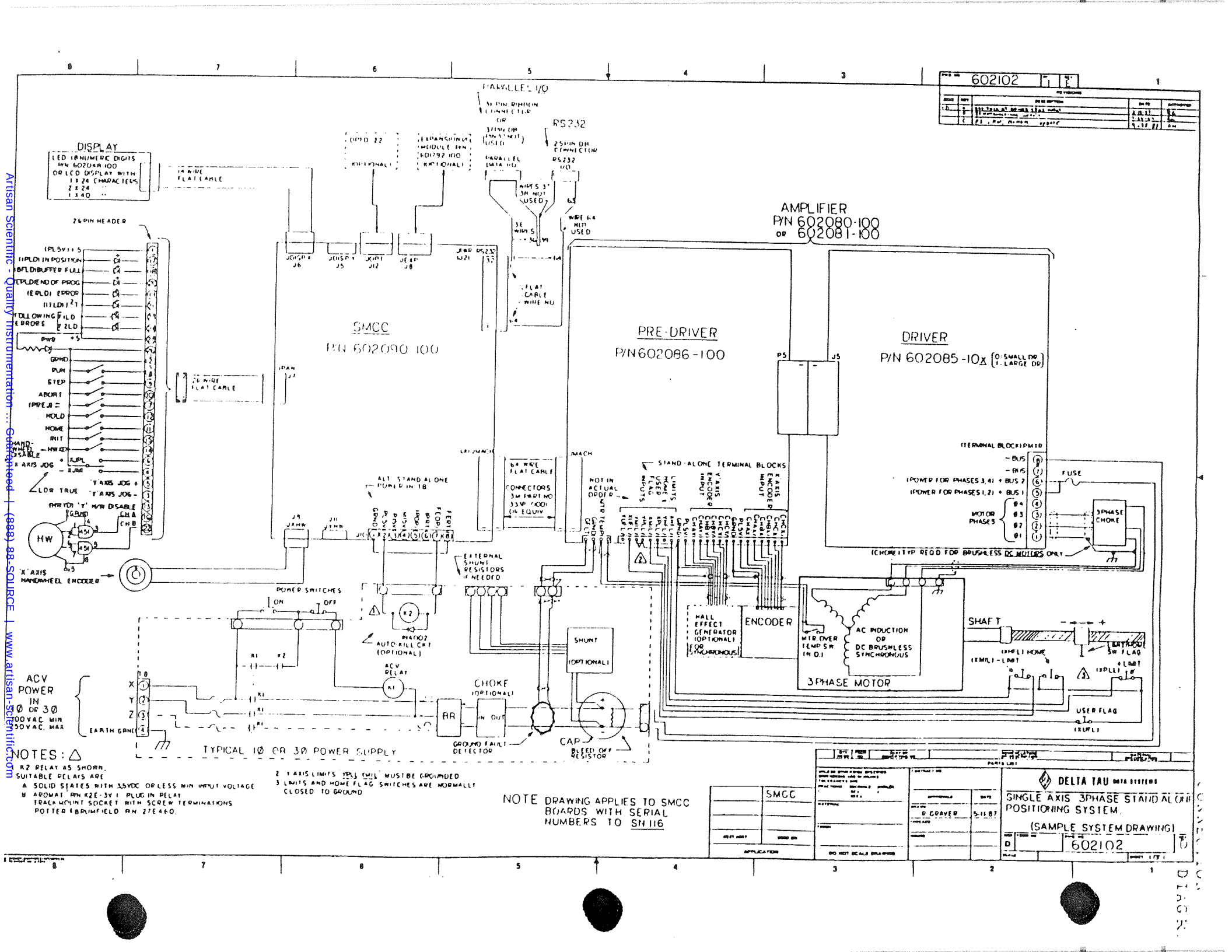
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



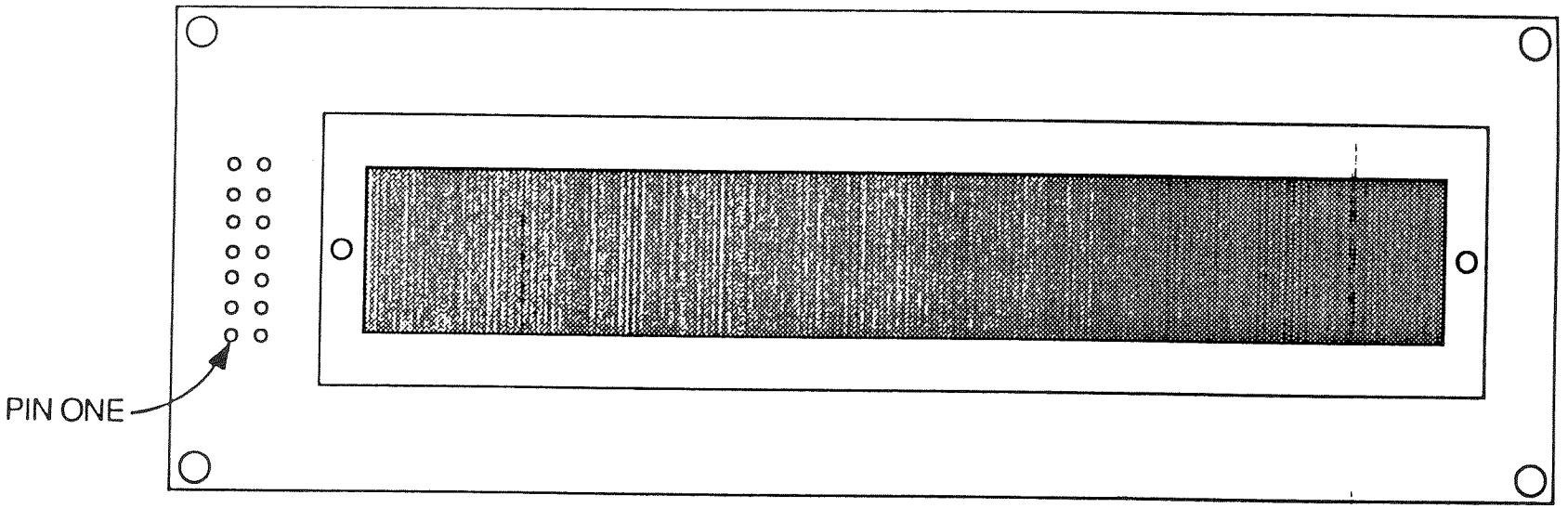
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



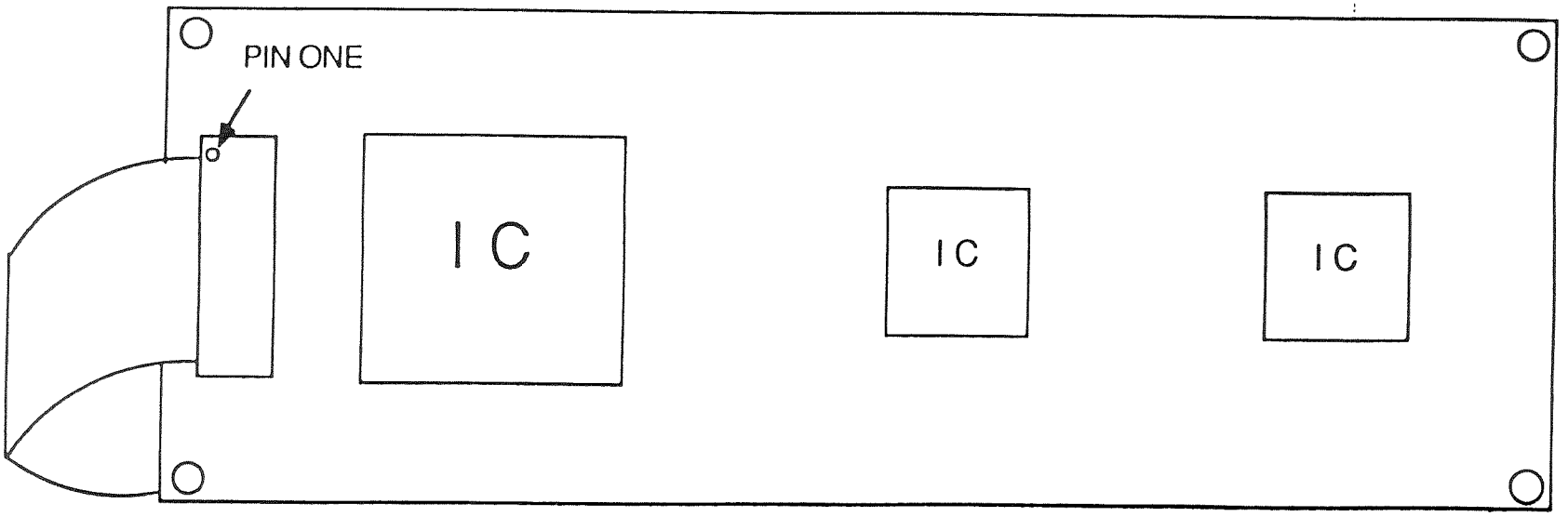
Artisan Scientific ... Quality Instrumentation ... Guaranteed (888) 88-SOURCE | www.artisan-scientific.com


602102
PARALLEL I/O
RS232
DISPLAY
AMPLIFIER
P/N 602080-100
OR 602081-100
SMCC
P/N 602090 100
PRE-DRIVER
P/N602086-100
DRIVER
P/N 602085-10X (0: SMALL DR 1: LARGE DR)
STAND-ALONE TERMINAL BLOCKS
FUSE
3PHASE CHOKE
ENCODER
HALL EFFECT GENERATOR (OPTIONAL) FOR SYNCHRONOUS
AC INDUCTION OR DC BRUSHLESS SYNCHRONOUS
3PHASE MOTOR
SHAFT
SHUNT (OPTIONAL)
CHOKE (OPTIONAL)
CAP
BLEED OFF RESISTOR
GROUND FAULT DETECTOR
AUTO-FILL CRT (OPTIONAL)
ACV POWER IN
ØØ OR 3Ø
TYPICAL 1Ø OR 3Ø POWER SUPPLY
HW
'X' AXIS HANDWHEEL ENCODER
NOTES: △
1 R2 RELAY AS SHOWN, SUITABLE RELAYS ARE
A SOLID STATES WITH 3.5VDC OR LESS MIN INPUT VOLTAGE
B AROMAT P/N RZE-3Y-1 PLUG IN RELAY
TRACK MOUNT SOCKET WITH SCREW TERMINATIONS
POTTER & BRUMFIELD P/N 27E460.
2 Y AXIS LIMITS (PEL) (MEL) MUST BE GROUNDED
3 LIMITS AND HOME FLAG SWITCHES ARE NORMALLY CLOSED TO GROUND
NOTE DRAWING APPLIES TO SMCC BOARDS WITH SERIAL NUMBERS TO SN 116
SMCC
D GRAVER 5-11-87
DELTA TAU DATA SYSTEMS
SINGLE AXIS 3PHASE STAND ALONE POSITIONING SYSTEM.
(SAMPLE SYSTEM DRAWING)
602102
D

ACC 12 OR 12A
2 x 24 OR 40 LINE LCD DISPLAY
(FRONT SIDE)

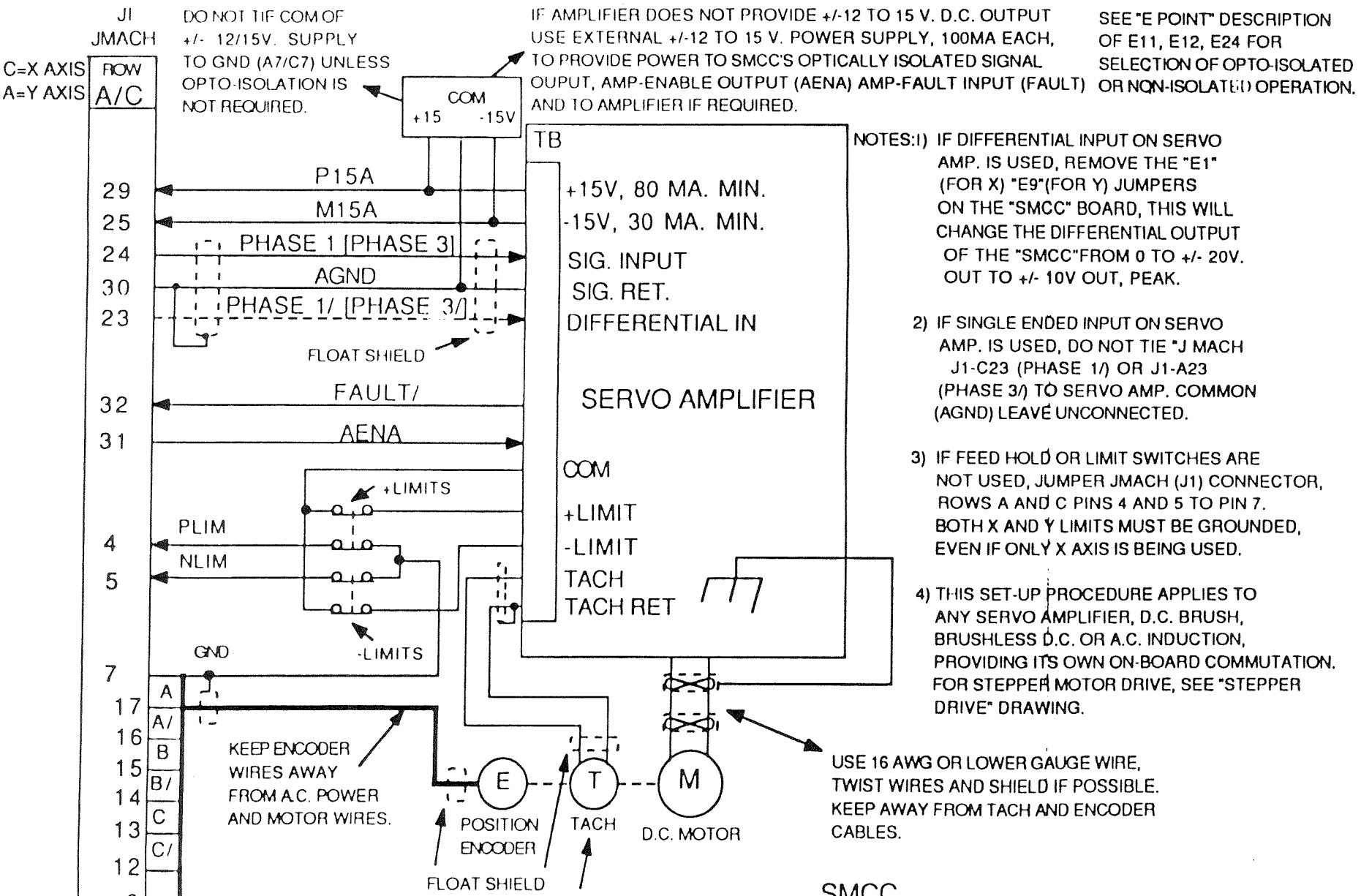


(BACK SIDE)



CONNECTION DIAGRAMS

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

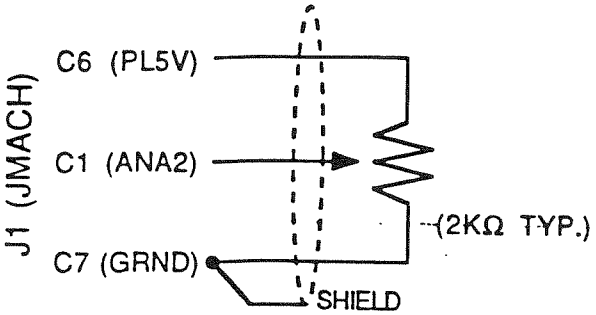


SMCC
SERVO AMPL. SIGNAL, POWER AND MOTOR, TACH & ENCODER INPUT/OUTPUT CONNECTIONS

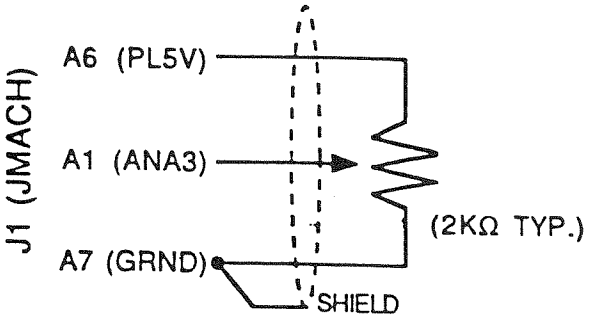
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

ANALOG INPUT POTENTIOMETER CONNECTIONS

1) WITH ANALOG CONDITIONING (NO OPTION 3)

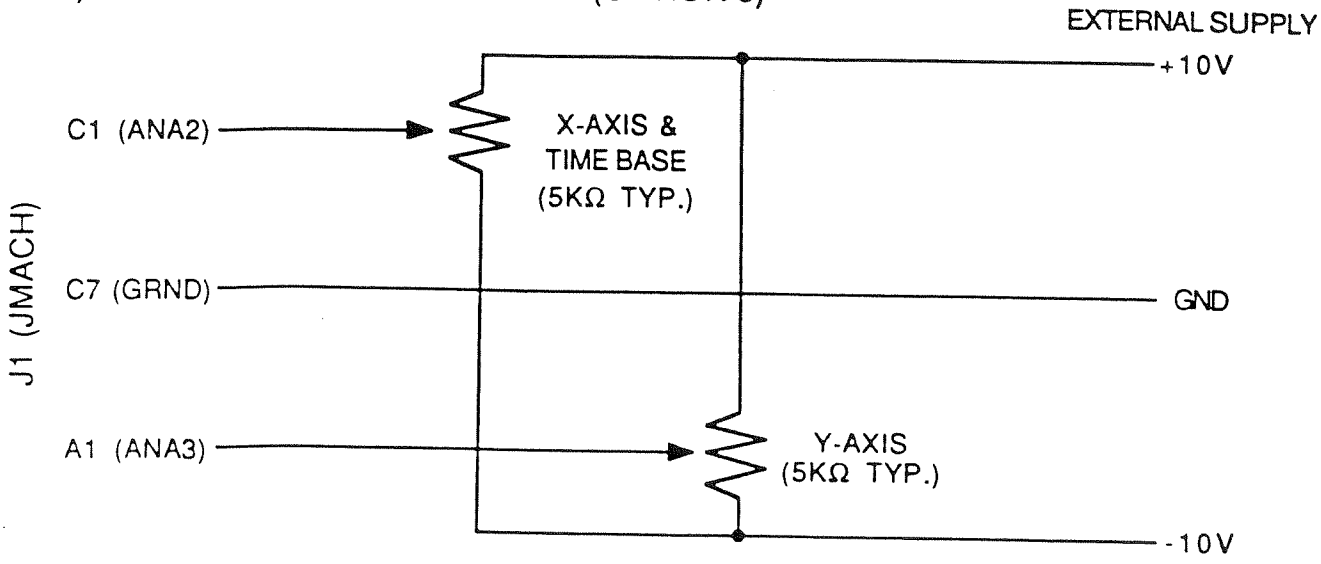


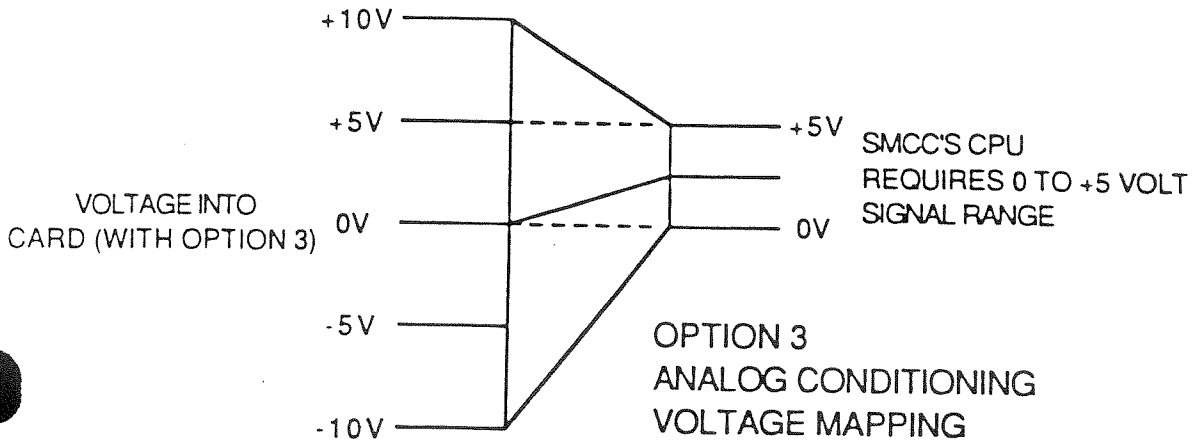
X-AXIS & TIME BASE



Y-AXIS

2) WITH ANALOG CONDITIONING (OPTION 3)





OPTION 3
ANALOG CONDITIONING
VOLTAGE MAPPING

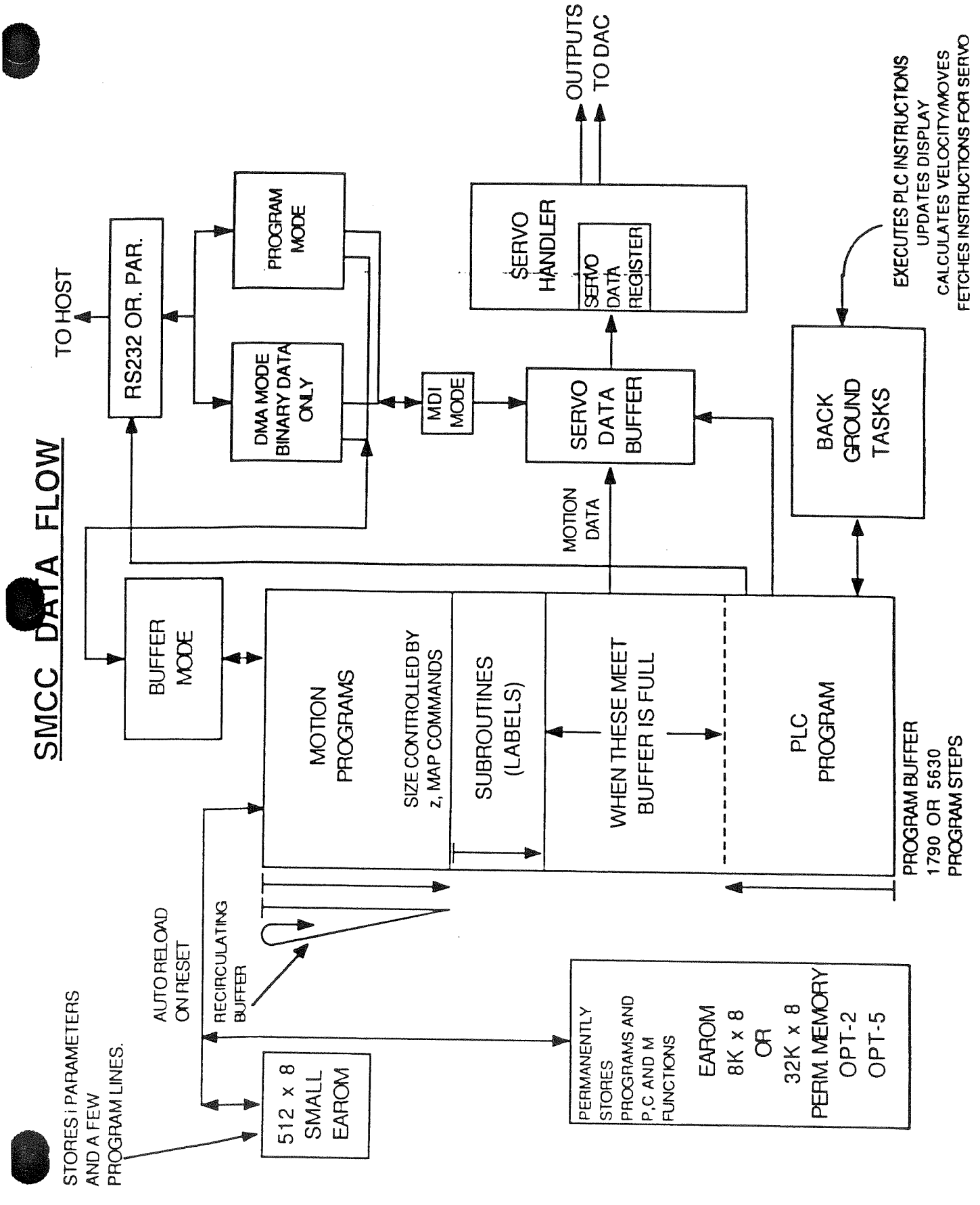
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TIMING DIAGRAMS

TABLE OF CONTENTS

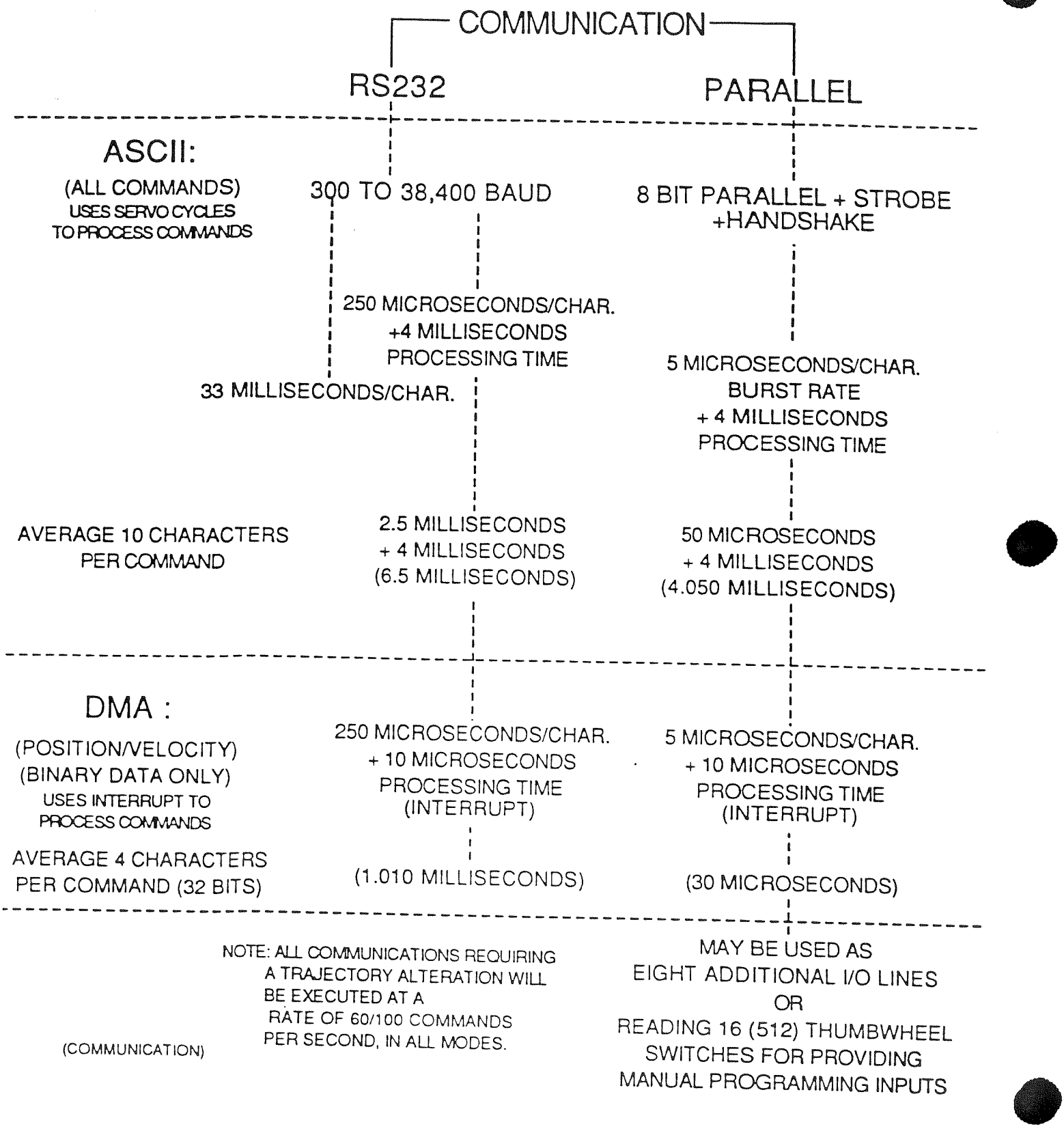
	PAGE
DATA FLOW	.344
COMMUNICATION	.345
BUFFER FULL TIMING.	.346
PARALLEL PORT TIMING.	.347
RS232 TIMING.	.348

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

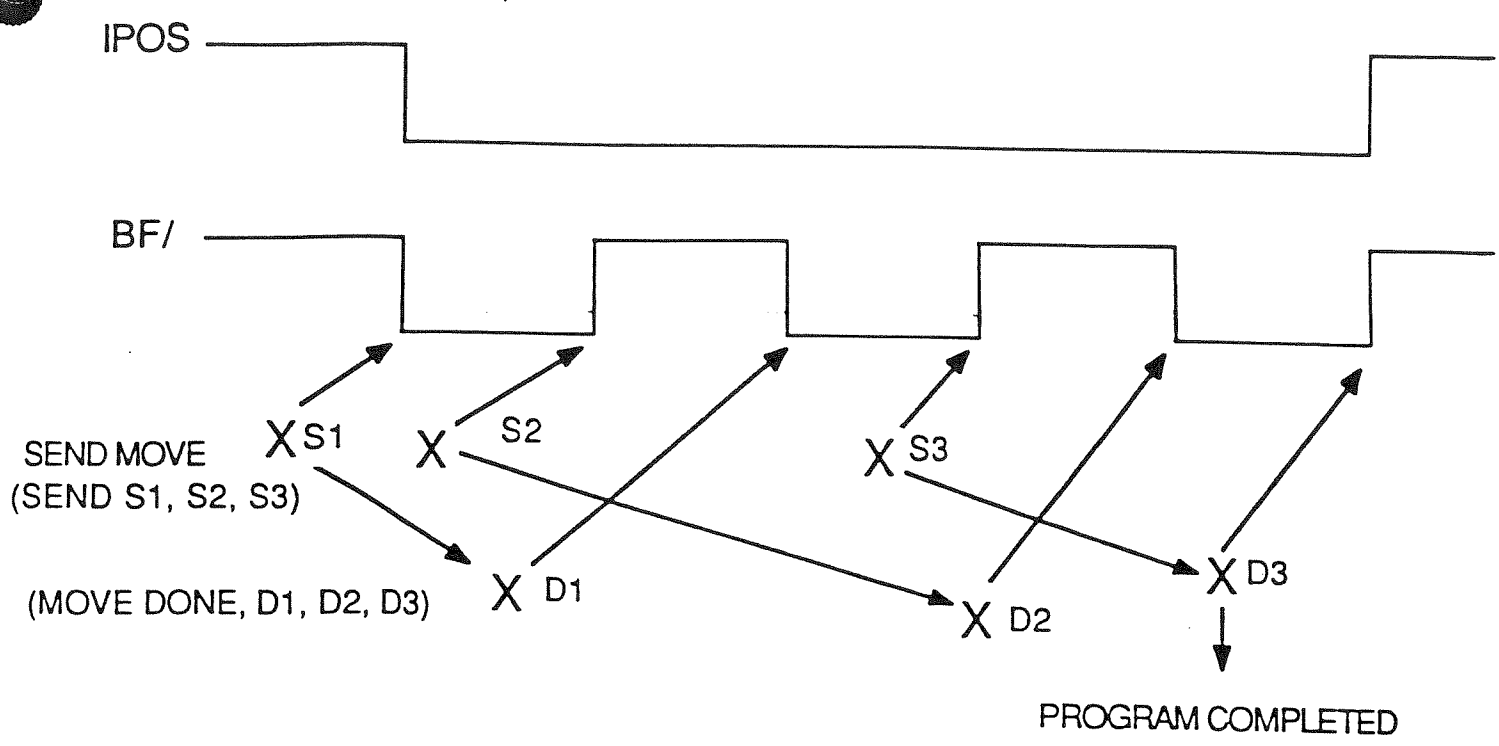


(DATA FLOW)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

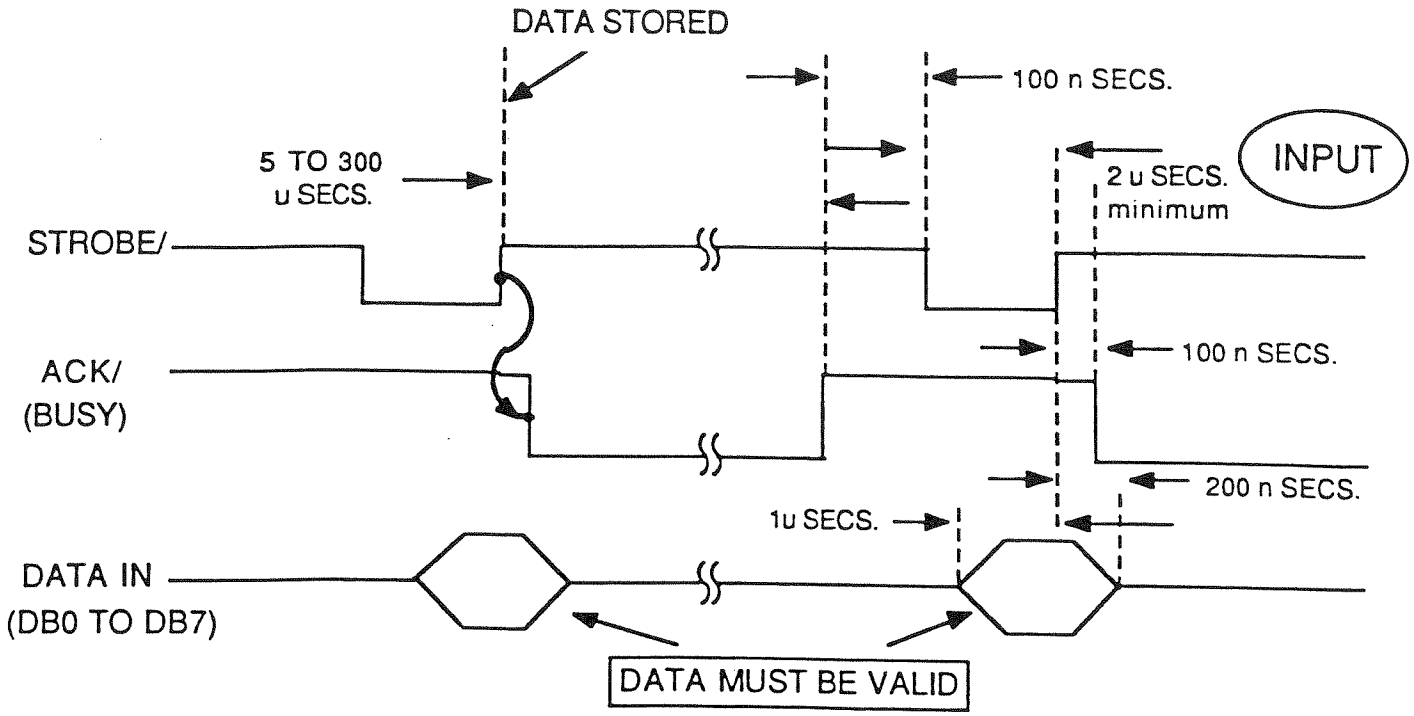


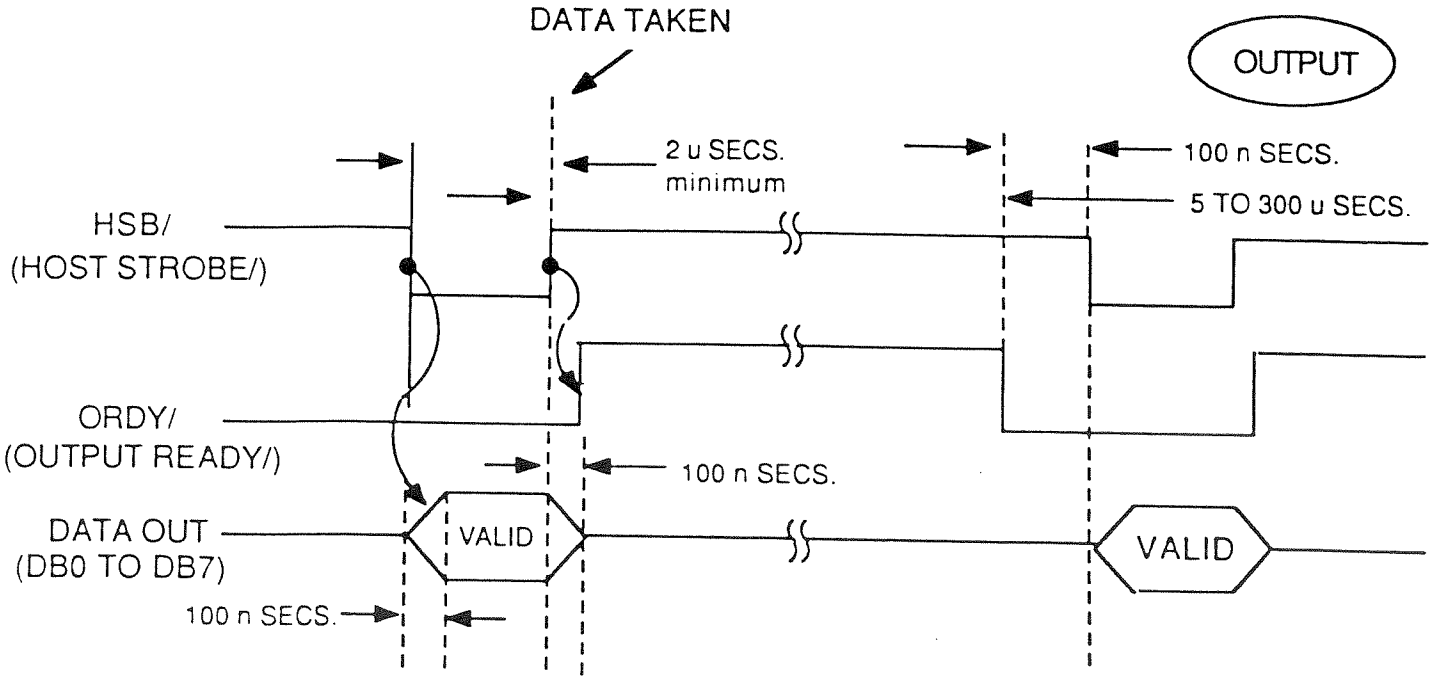
TIME BETWEEN SENDING OR COMPLETING MOVE AND BF/ CHANGE IS UP TO 10MS (CALCULATION TIME FOR 2 AXIS MOVE)

BUFFER FULL AND IN-POSITION TIMINIG WHEN IN MDI MODE

(MDI)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



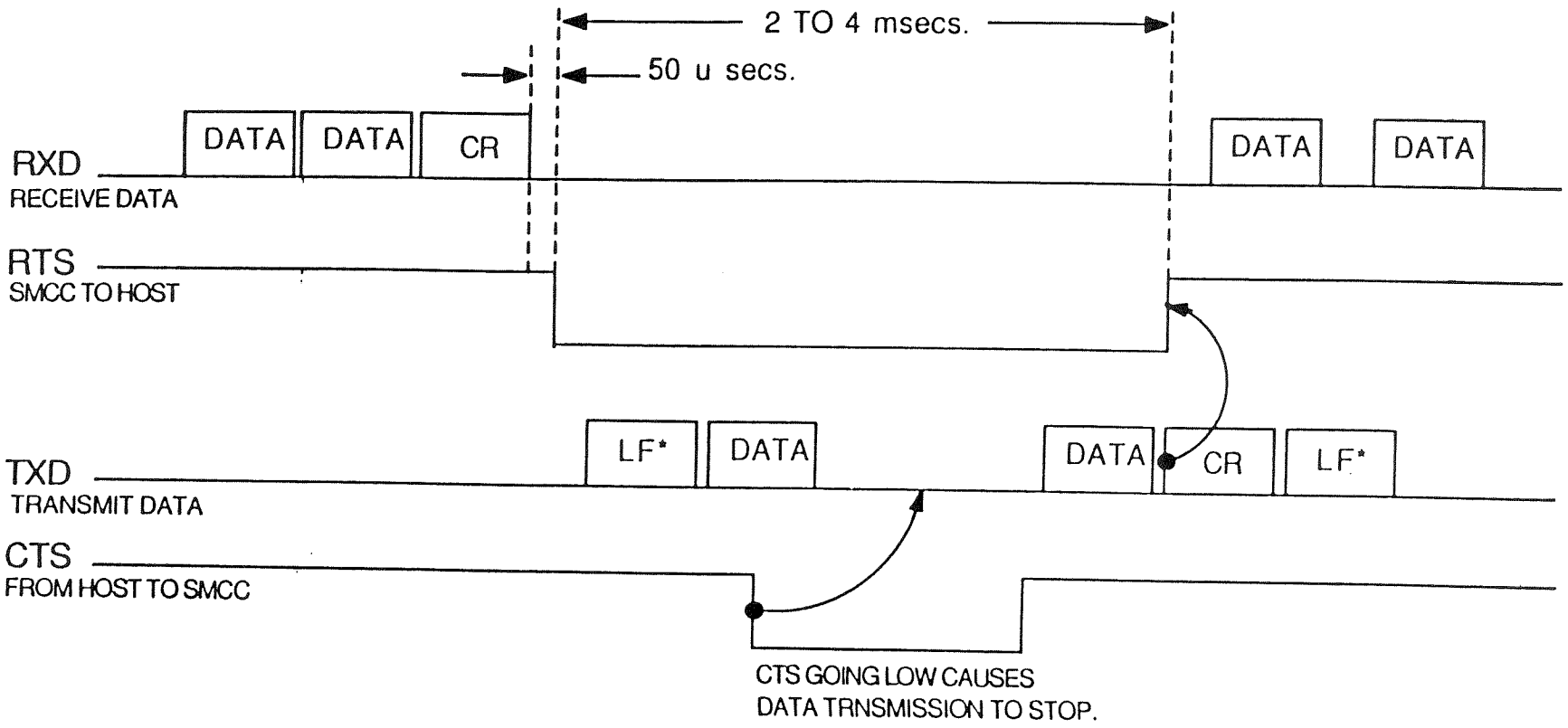


SMCC PARALLEL PORT TIMING DIAGRAM.

(PARALLEI

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

RS232 TIMING & HANDSHAKE



(*LF IN i03 1 MODE)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SCHEMATICS

TABLE OF CONTENTS

PAGE

J1, J2, J7, AND J8 SCHEMATIC 350

J2 AND J12 SCHEMATIC 351

J1 SCHEMATIC 352

J1 AND J10 SCHEMATIC 353

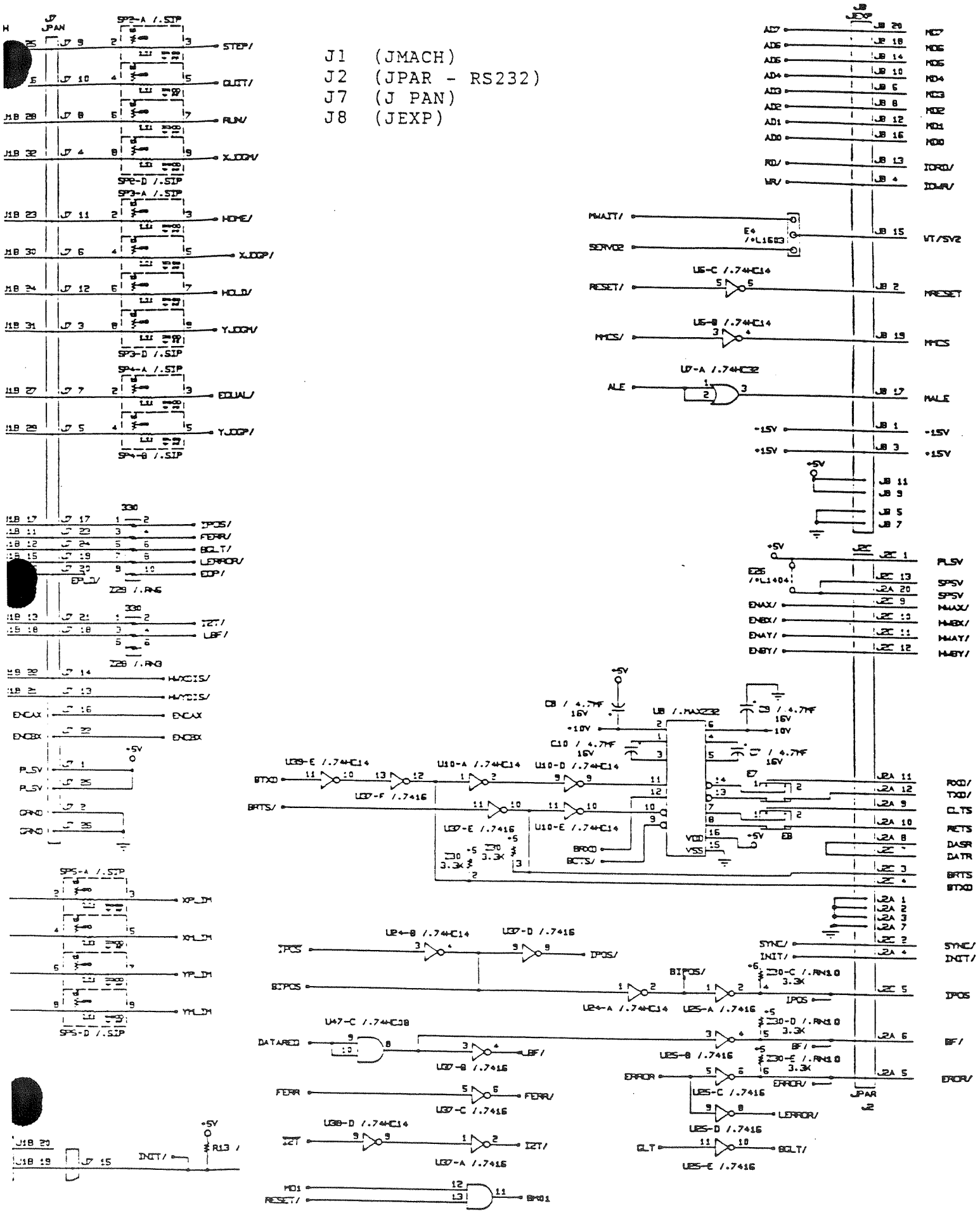
J2, J5, J6, J9, J11 SCHEMATIC. 354

J2 SCHEMATIC 355

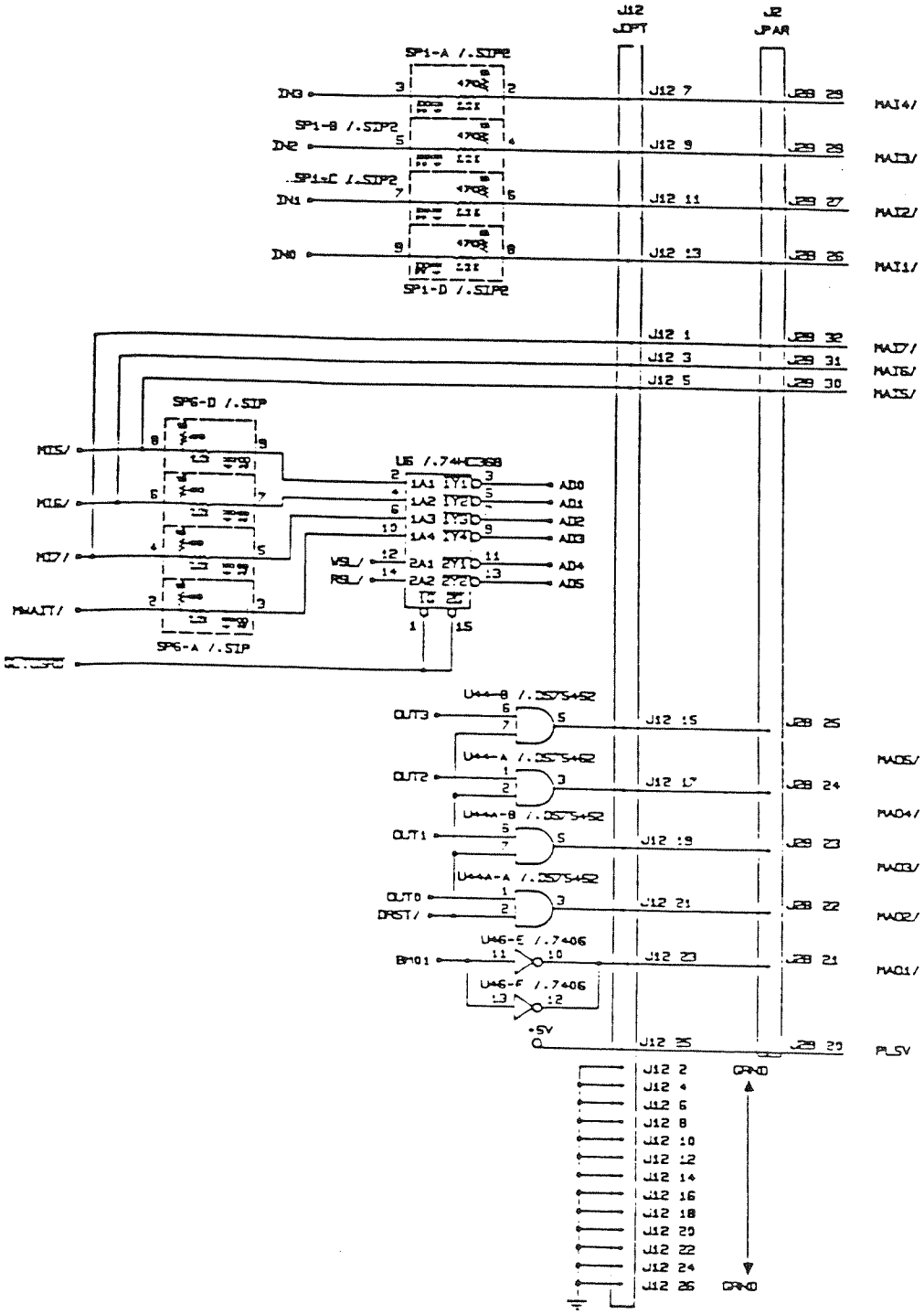
ANALOG SIP (OPTION 3) SCHEMATIC. 355A

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J1 (JMACH)
J2 (JPAR - RS232)
J7 (J PAN)
J8 (JEXP)

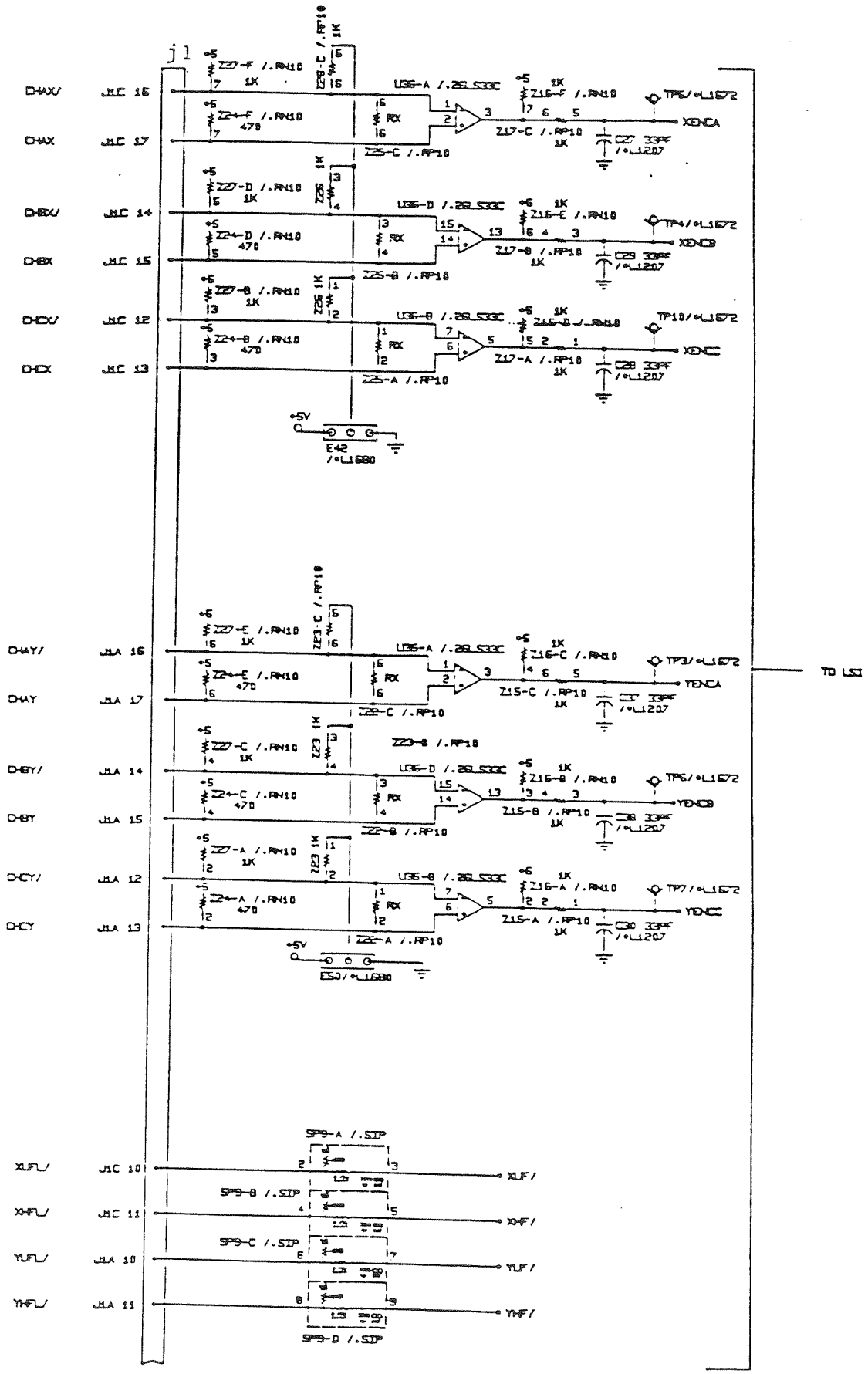


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



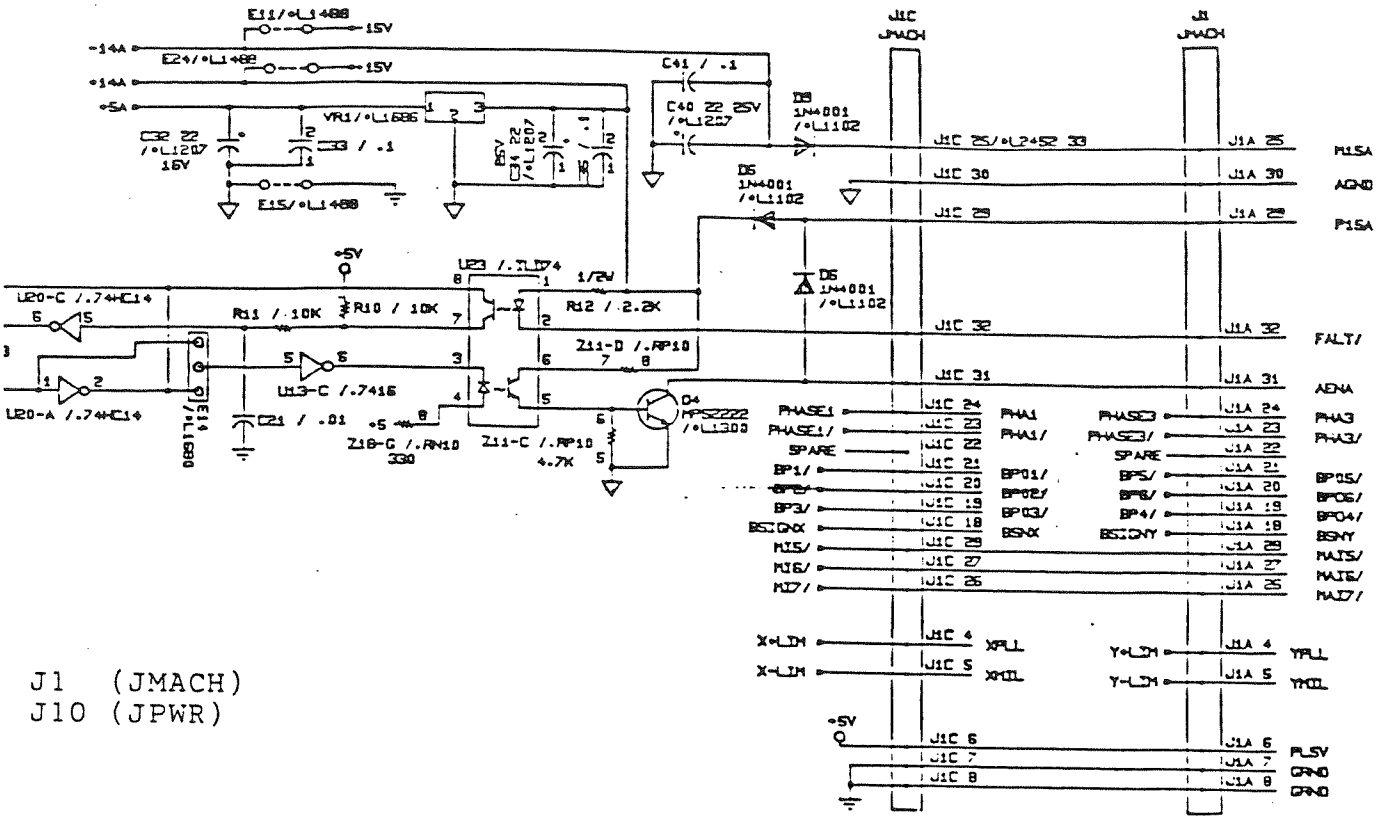
J2 (JPAR - RS232)
J12 (JOPT)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

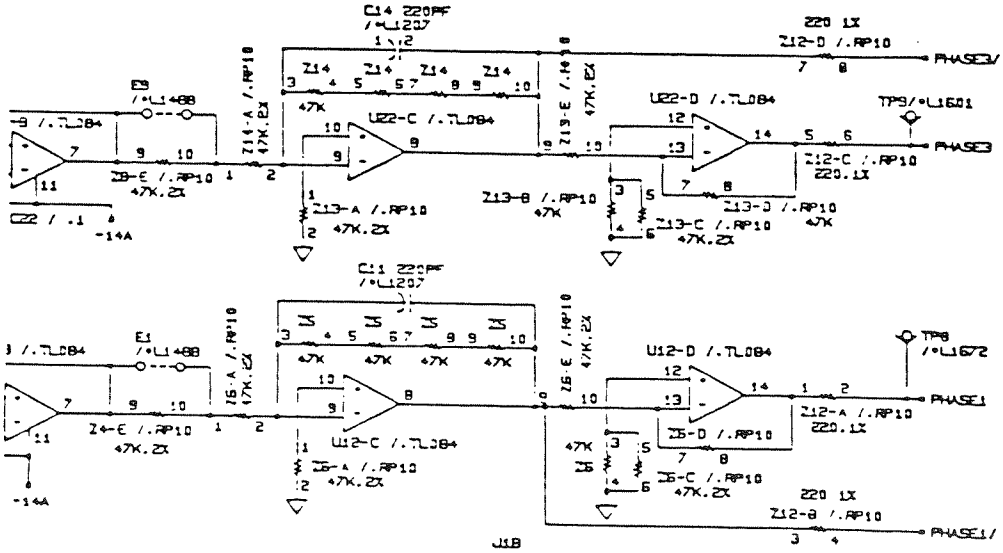


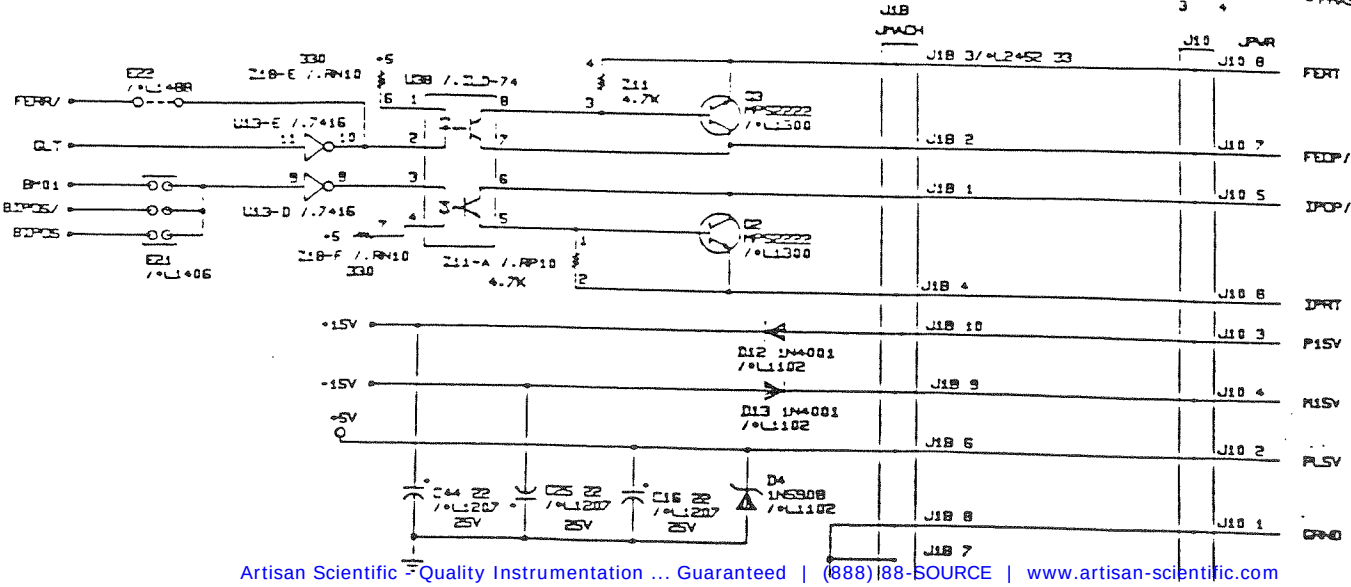
J1 (JMACH)

Artisan Scientific - Quality Instrumentation, Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

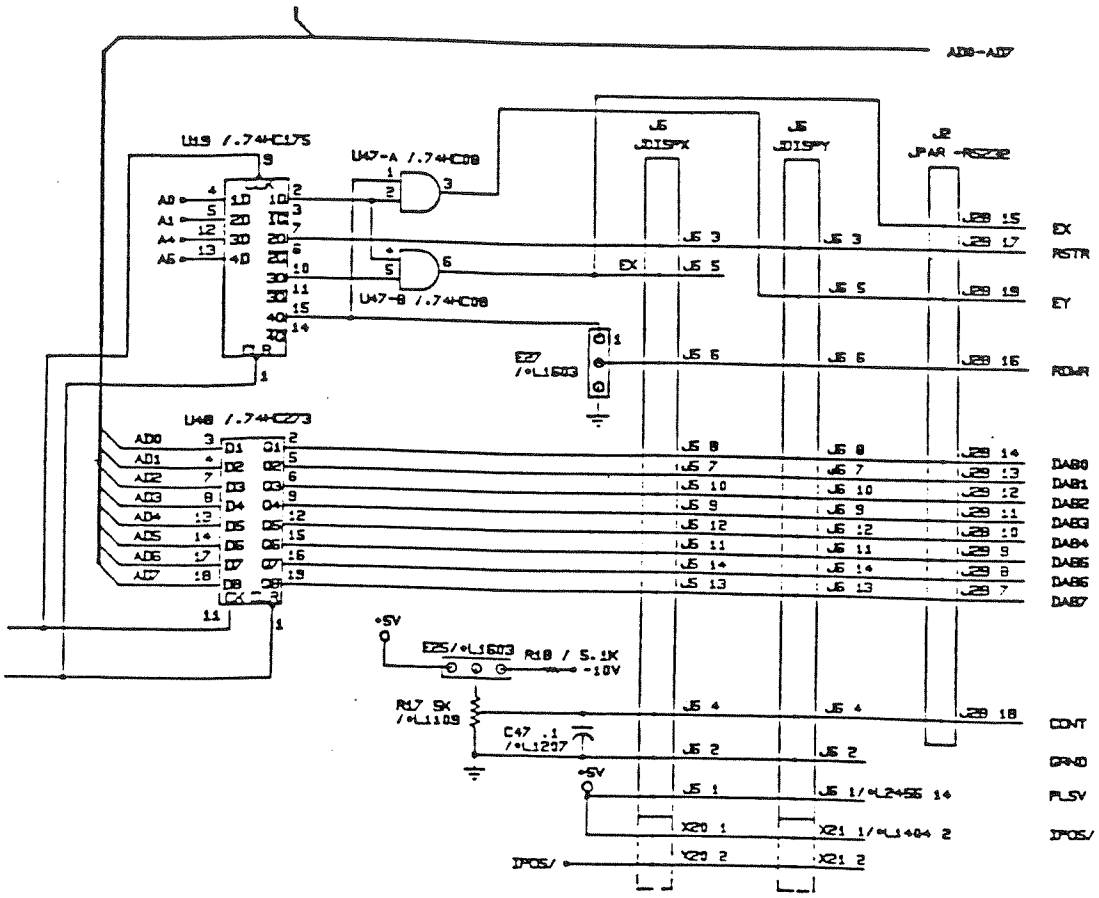


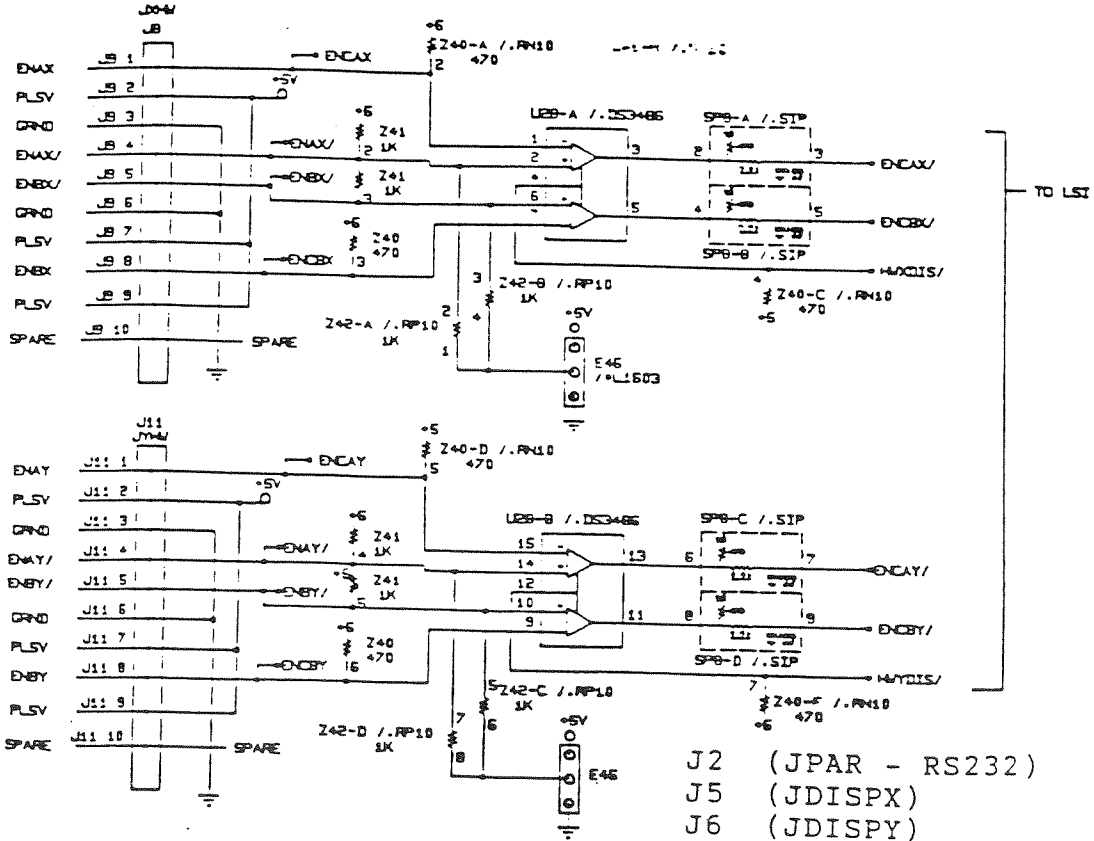
J1 (JMACH)
J10 (JPWR)





Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

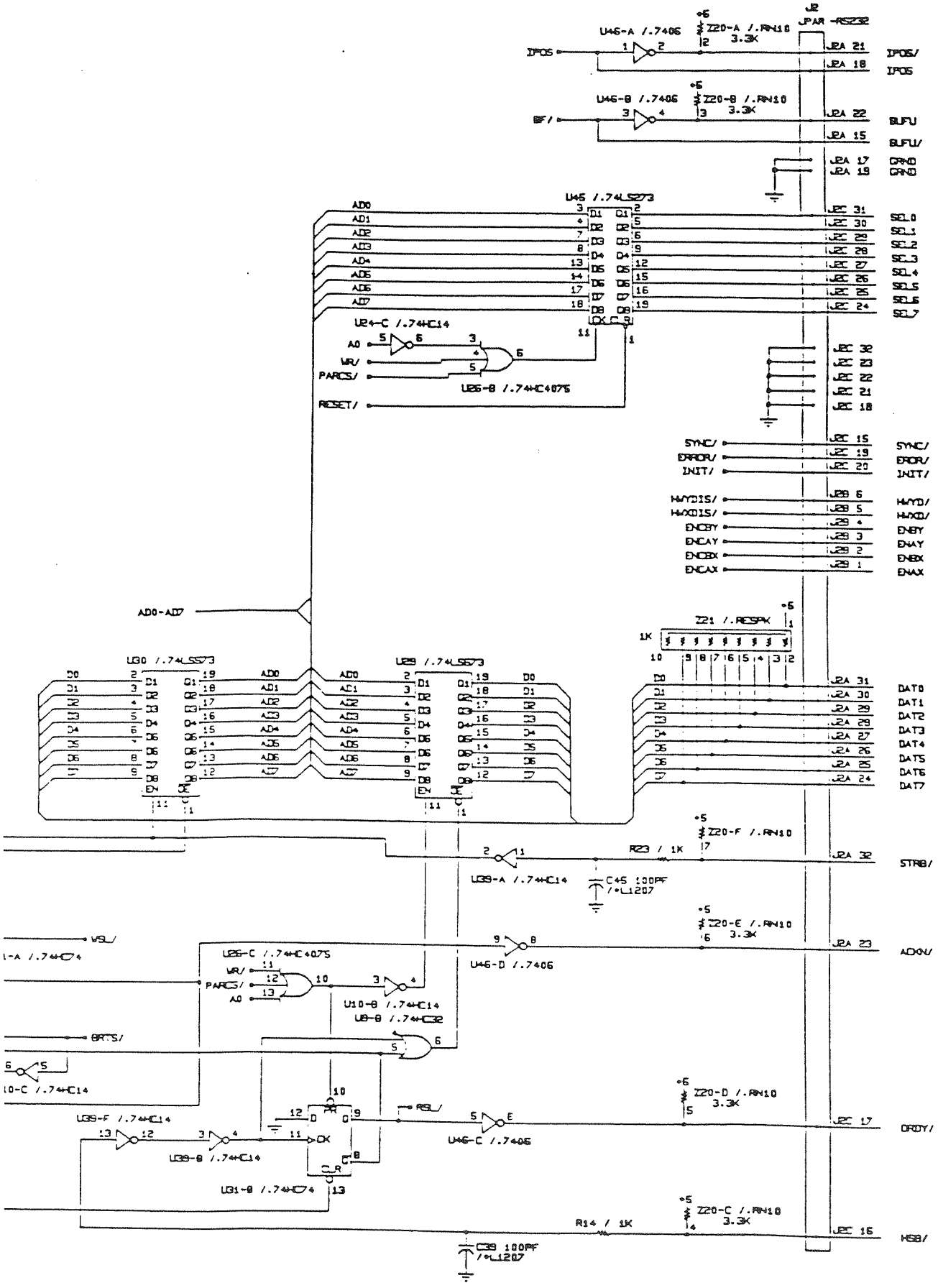




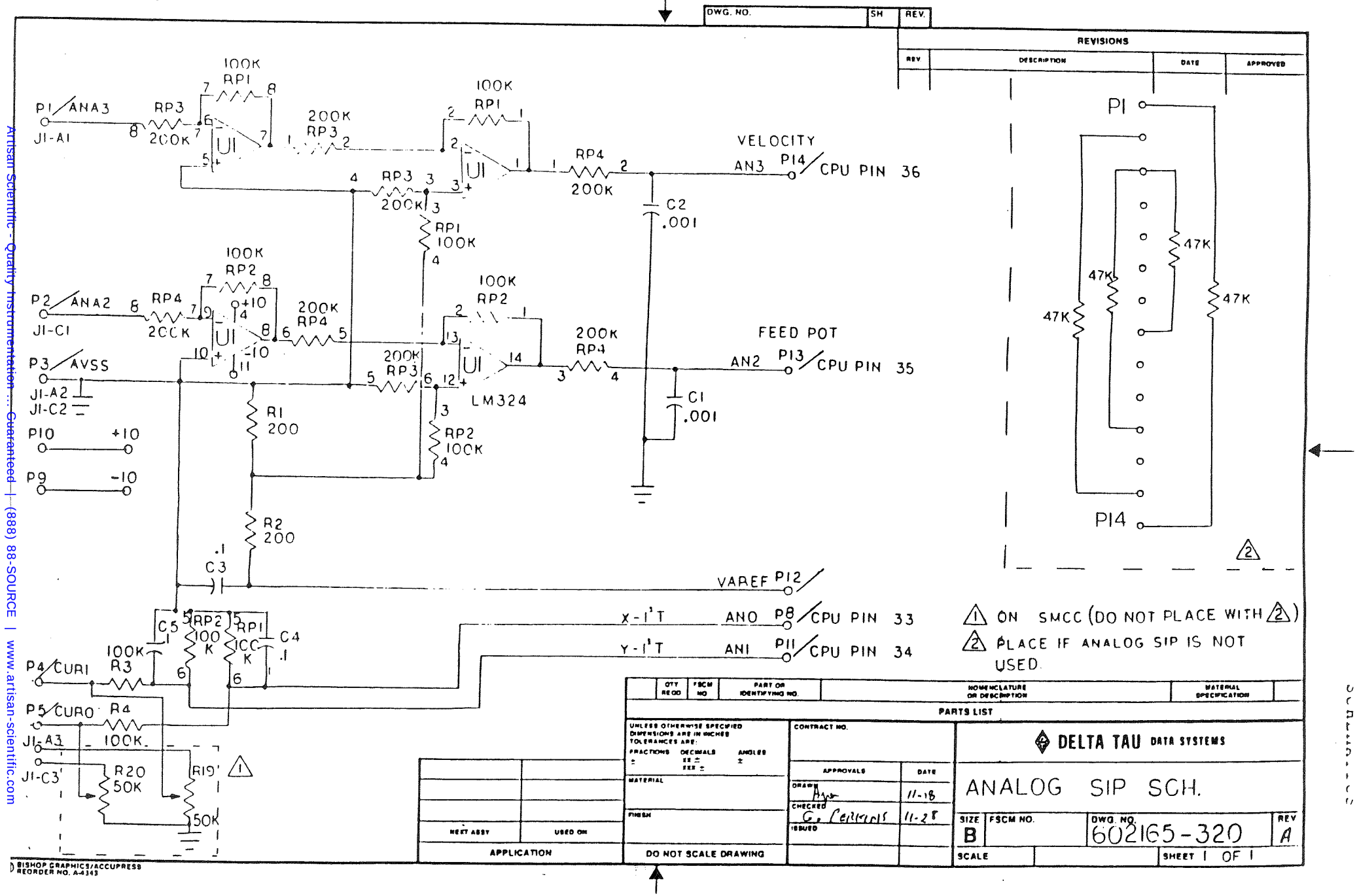
J2 (JPAR - RS232)
J5 (JDISPX)
J6 (JDISPY)
J9 (JXHW)
J11 (JYHW)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

J2 (JPAR - RS232)



Artisan Scientific - Quality Instrumentation. Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

ACCESSORIES AND OTHER INTERCONNECTION DIAGRAMS

TABLE OF CONTENTS

PAGE

SMCC TO THUMBWHEEL 357

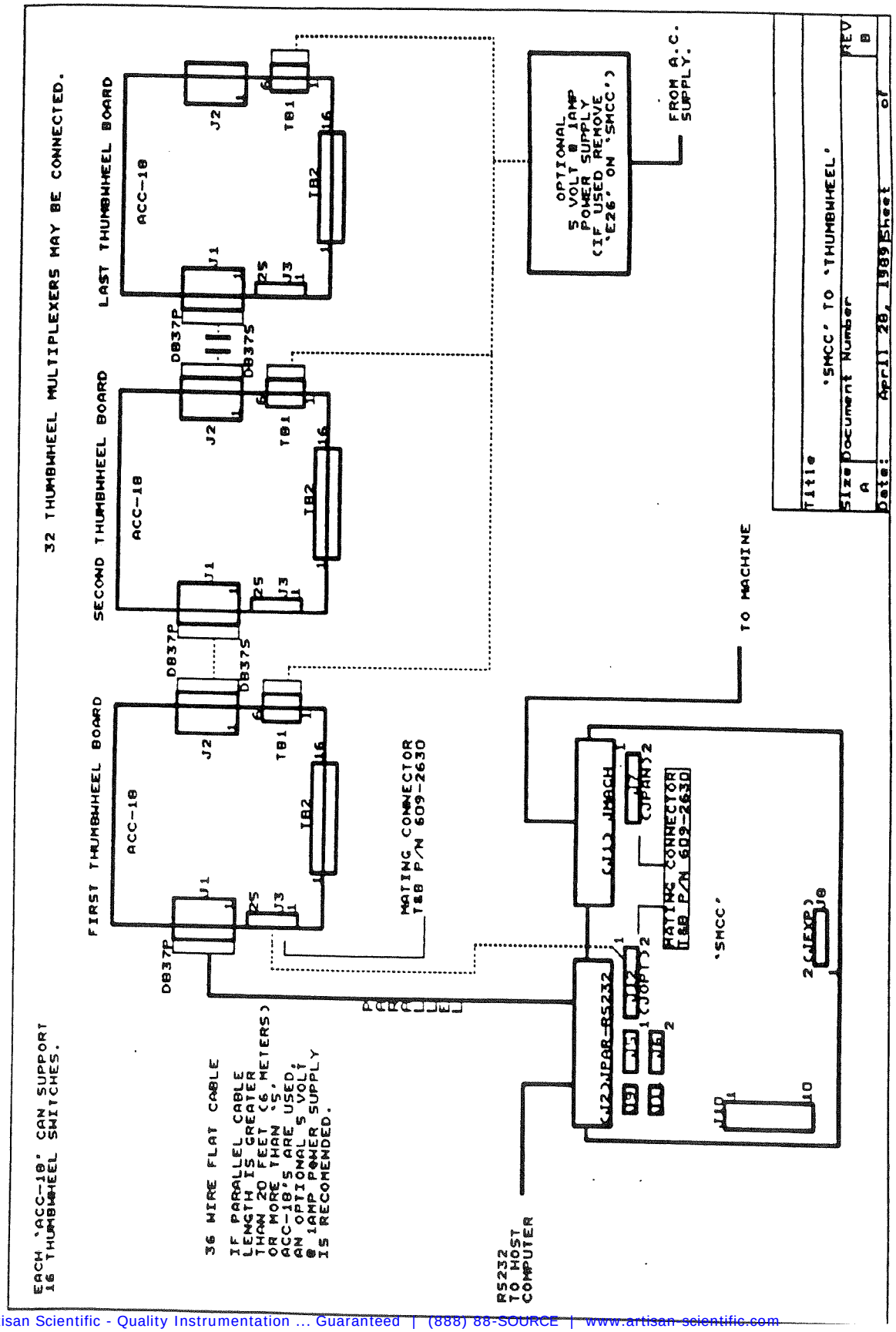
SMCC TO OPTO22 358

ACC14 TO OPTO22. 359

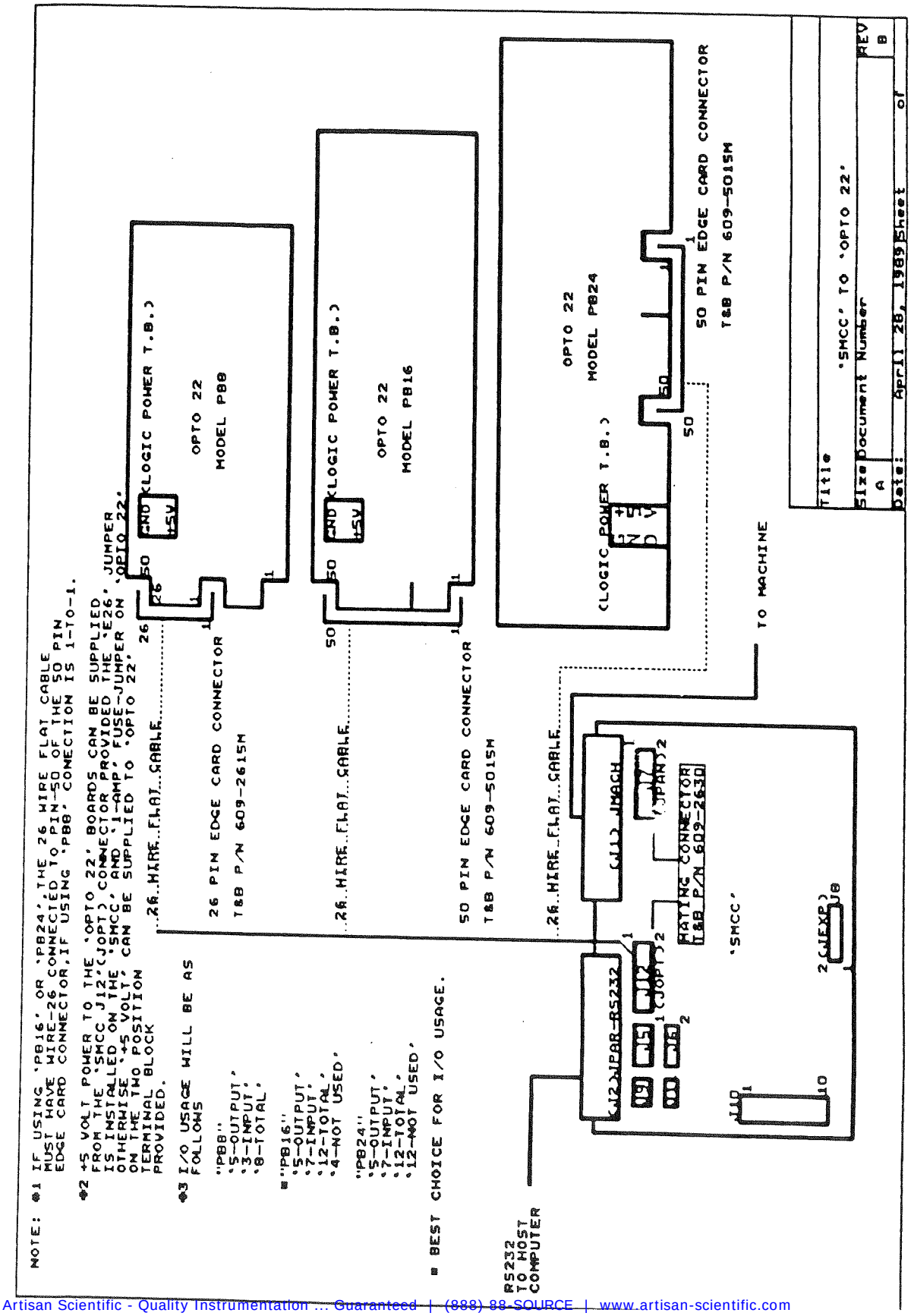
SMCC TO HYDRAULIC CYLINDER 360

TERMINAL BLOCK FAN-OUT FOR SMCC CONNECTORS . . 361

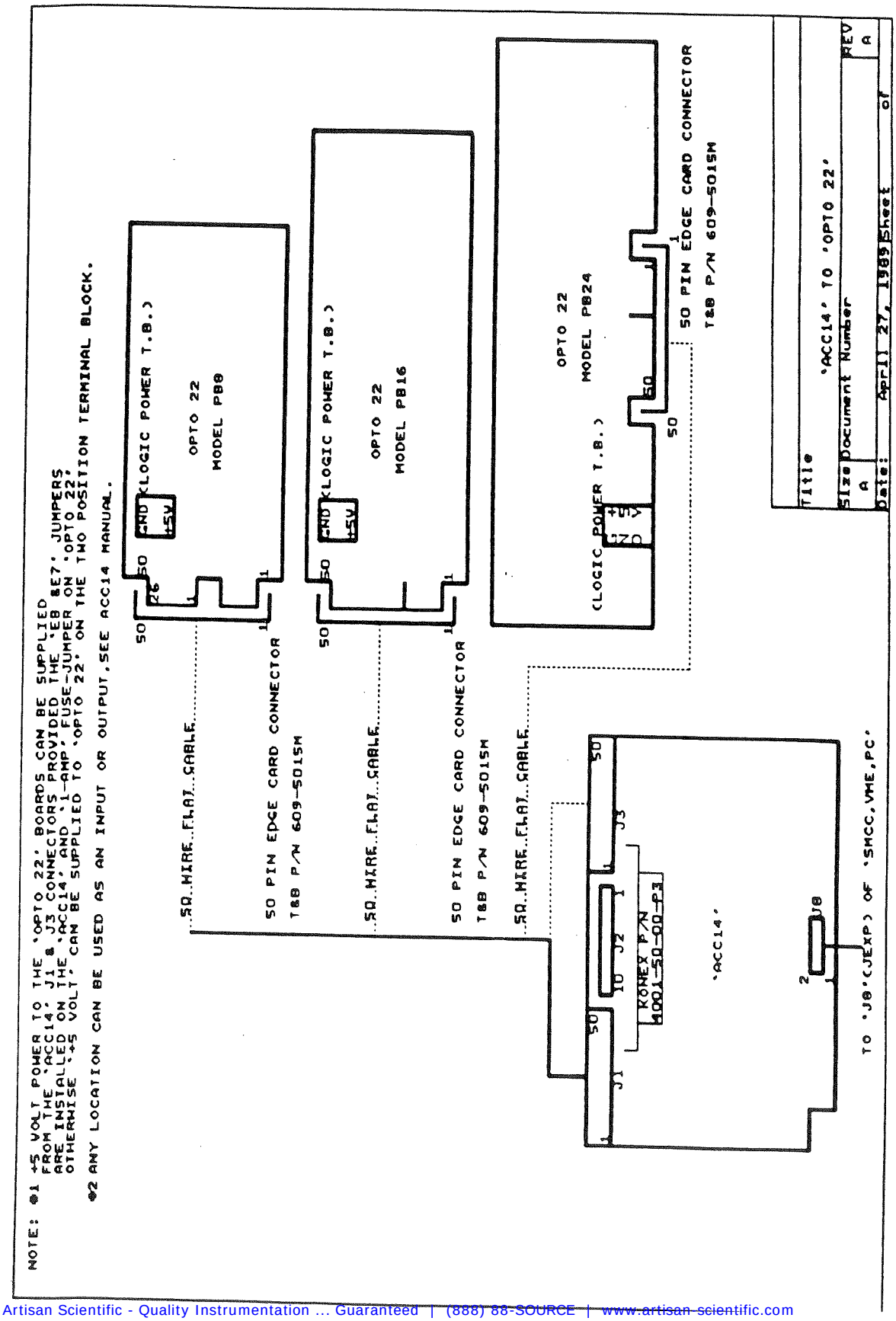
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



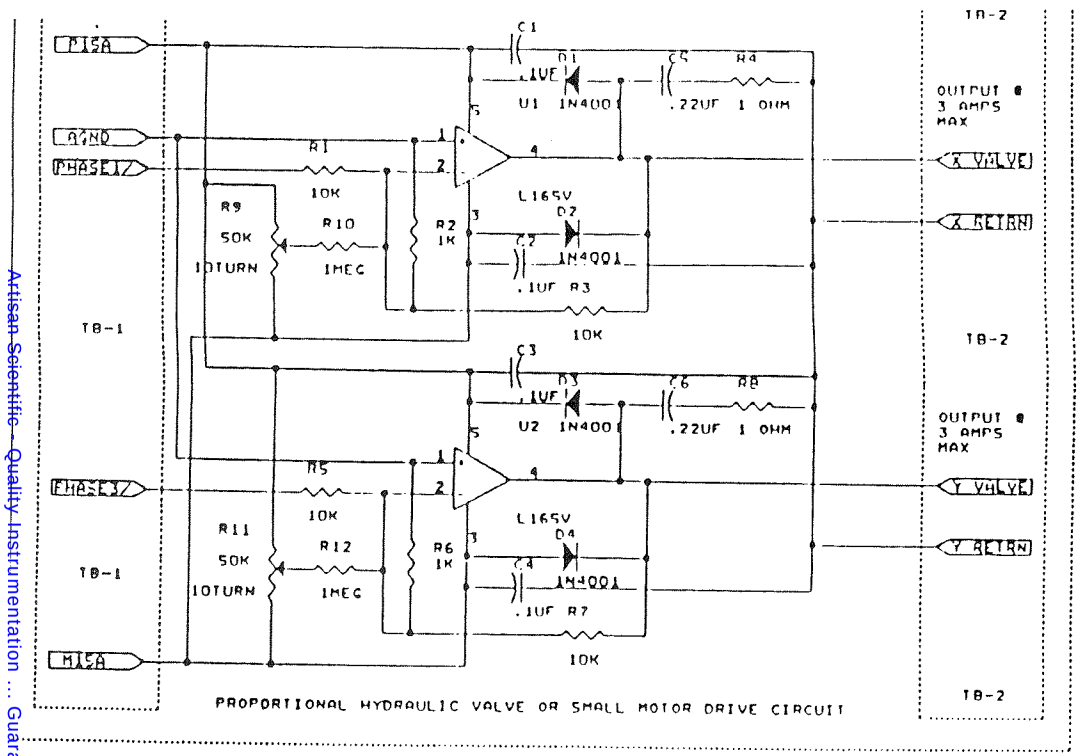
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

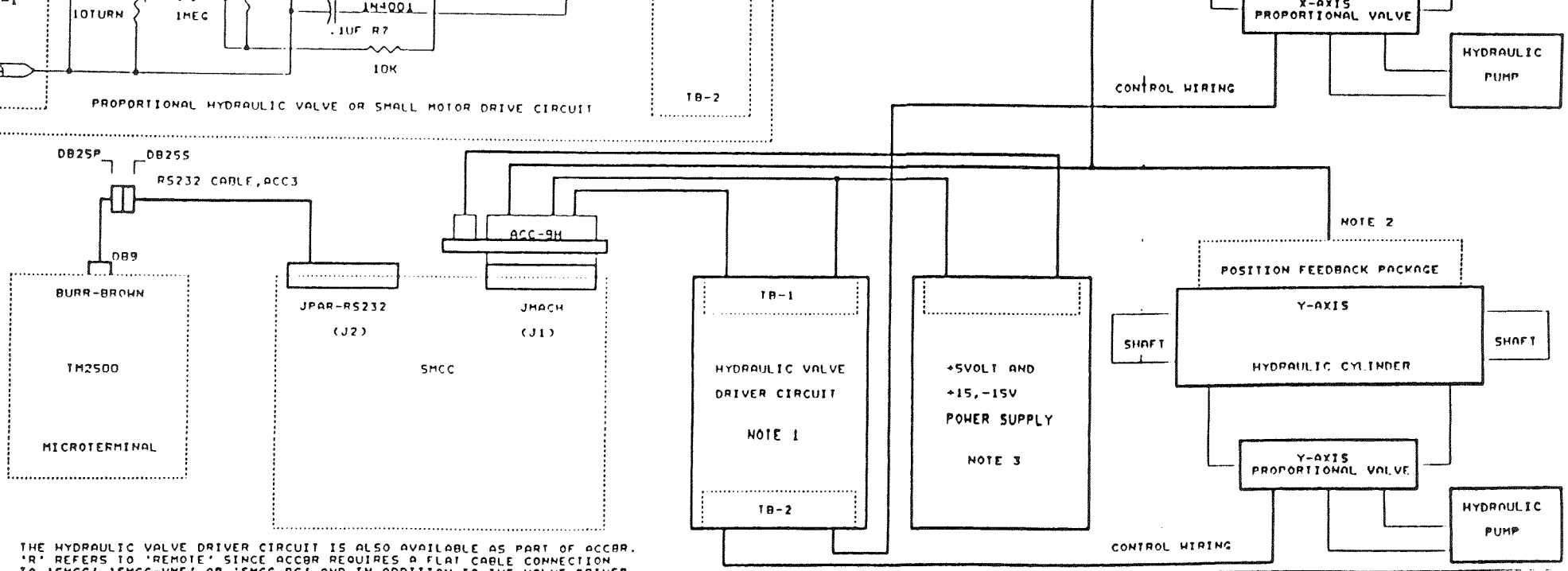


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com





NOTE: #1 THE HYDRAULIC VALVE DRIVER CIRCUIT IS ALSO AVAILABLE AS PART OF ACC8R. 'R' REFERS TO 'REMOTE' SINCE ACC8R REQUIRES A FLAT CABLE CONNECTION TO 'SMCC','SMCC-VME' OR 'SMCC-PC' AND IN ADDITION TO THE VALVE DRIVER, PROVIDES ALL NECESSARY TERMINAL STRIPS FOR CONNECTING ALL I/O FUNCTIONS TO SMCC.

#2 A/B QUADRATURE ROTARY OR LINEAR ENCODER OR RESOLVER OR TEMPOSONIC OR EQUIVALENT WITH A/B QUADRATURE CONVERSION CIRCUITS.

#3 THE +15V,-15V SUPPLY COULD BE AN UNREGULATED +12V,-12V SUPPLY AS LONG AS THE VOLTAGE DOES NOT EXCEED +18V,-18V.

DELTA TAU DATA SYSTEMS

Title
'SMCC' TO 'HYDRAULIC CYLINDER'

Size	Document Number	REV
B	6021A4-340	A

Date: April 24, 1995 Sheet 1 of 1

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TERMINAL BLOCK FAN-OUT FOR SMCC CONNECTORS

PHOENIX CONTACT TERMINAL BLOCK		ROBINSON NUGENT	3 M Co.	USAGE				
# OF TERMINALS	PART NUMBER	MATING CONNECTOR	MATING CONNECTOR	SMCC	SMCC-VME	SMCC-PC	SMCC-STDbus	ACC-14
10	2281018	IDS-C10PK-TR	3473-6600	J9 = JXHW J11 = JYHW				
26	2281050	IDS-C26PK-TR	3399-6600	J7 = JPAN J12 = JOPT	J1 = JPAN J3 = JRS232	J1 = JPAN J3 = JRS232	J1 = JPAN J3 = JRS232	
50	2281089	IDS-C50PK-TR	3425-6600					J1,J3 OPTO-22 I/O
60	2281092 or ACC-8R	IDS-C60PK-TR	3334-6600			P2=PMACH	P2=PMACH	
64	2970057 or ACC-8H,V or R	PIN-IDC-64CSB-TG	6964-0000-EF	J1 = JMACH	P2=PMACH			

"PHOENIX CONTACT" TERMINAL BLOCKS CAN BE USED TO BRING "HEADER CONNECTOR" SIGNALS OUT TO SCREW TYPE TERMINALS FOR EASY WIRING IN MACHINE ENVIRONMENT. THE CONNECTION BETWEEN THE SMCC CONNECTOR "HEADER" AND THE TERMINAL BLOCK IS DONE BY USING APPROPRIATE FLAT CABLE OF DESIRED LENGTH WITH MATING CONNECTORS. ACCESSORY 8 Horizontal, Vertical or Remote MAY ALSO BE USED FOR THE J OR PMACH CONNECTOR.

(FAN-OUT)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

APPLICATION NOTES AND FURTHER EXAMPLES

TABLE OF CONTENTS

PAGE

APP. NOTE: CUSTOMER CONVERSION FROM STANDARD TO PARABOLIC SMCC 363

APP. NOTE: DETERMINING POSITION AND VELOCITY AT SEGMENT BOUNDARIES IN 3-POINT SMOOTHER (CIR4) MODE . . . 364

APP. NOTE: USING SMCC'S PROGRAM BUFFER IN RECIRCULATING AND OTHER MODES. 365

APP. NOTE: TRAPEZOIDAL MOVE PROGRAMMING ON THE SMCC. . . 367

Feedrate-Based Move Diagram. 369
Time-Based Move Diagram. 369
Concatenated Move Diagram. 371
Two-Axis Concatenated Move Diagram 371

APP. NOTE: CONVERSION OF SMCC RAW HEXADECIMAL DATA TO DECIMAL 374

EXAMPLE 50: PARAMETERIZED MOVE WITH ANALOG SPEED CONTROL AND LEARNED POSITIONS. 375

EXAMPLE 51: TIME-BASED FOLLOWING PROGRAM WITH THUMBWHEEL-CONTROLLED RATIO AND BACKGROUND CORRECTION. . 378

EXAMPLE 52: USE OF PARABOLIC MODE FOR TRAPEZOIDAL MOVES. 383

Move Performed at Half Speed (Thumbwheels at 4999) . 385
Move Performed at Full Speed (Thumbwheels at 9999) . 386

EXAMPLE 53: THUMBWHEEL-CONTROLLED S-CURVE MOVES 387

Move Performed with Thumbwheel Set to 30.00" 390
Move Performed with Thumbwheel Set to 15.00" 391

APPL. NOTE: UNDERSTANDING SMCC'S TIME BASE CONTROL . . . 392

APPL. NOTE: USING SMCC'S DATA GATHERING FOR INTERMITTENT PROBLEMS OR SIMULATNEOUS MULTIPLE-CARD GATHERING 399

APPL. NOTE: SMCC'S TORQUE-LIMIT MODE 402

APPL. NOTE: USING THE SMCC DUMP UTILITY. 405

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TABLE OF CONTENTS (CONT.)

PAGE

APPL. NOTE: RE-INITIALIZING SMCC 406

APPL. NOTE: CONSTRUCTIVE USE OF EXTRA SMCC MEMORY SPACE. 409

APPL. NOTE: SOFTWARE UPDATES FOR SMCC "PARABOLIC". . . . 413

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

CUSTOMER CONVERSION FROM STANDARD TO PARABOLIC SMCC

To change a standard SMCC to parabolic SMCC, please follow these steps:

1. Remove the standard CPU (MC2A) and install the parabolic CPU (parabolic).

2. Remove the standard EPROM and install the parabolic EPROM.

3. Apply power to the SMCC. Note that the red error LED should light up due to the mismatch in system parameters from standard to parabolic.

4. In order to get the SMCC back to normal, the system parameters have to be reinitialized. To reinitialize, connect the RESET signal (J7 pin 15) to one leg of a normally open momentary push button switch, and GND (J7 pin 2 or 26) to the other leg of the switch. Connect the STEP/ signal (J7 pin 9) to one leg of a normally open momentary toggle switch and GND (J7 pin 2 or 26). Do the same thing to the RUN/ (J7 pin 8) and ABORT/ (J7 pin 10). If you have an ACC-16 control panel, all the switches needed for manual reset are available.

5. After all the switches are hooked up, power up the SMCC with a computer connected to the SMCC running a terminal emulation program such as the SMCC TERMINAL program, and complete the following steps:

   A. Push and hold the ABORT/ switch while pushing and releasing the reset button. Look at the computer screen where two lines of message will be displayed on the screen. When the message is displayed, release the ABORT/ switch.

   B. Follow the prompts on the screen and choose the default parameters and the system type (brush, brushless, or AC induction).

   C. After the parameters and system are chosen, the SMCC will do a series of switch tests. To bypass the switch test, toggle the ABORT/ switch until you are out of the switch test. The screen now displays encoder and handwheel encoder counts. If you have the motor encoder and the handwheel encoder hooked up to the SMCC, rotate the motor shaft to see the encoder counts change on the screen. Rotate the handwheel to see the handwheel encoder counts change. To get the SMCC back to normal operation, toggle the ABORT/ switch and then save the system parameters ('s' Command). You must save the parameters at least once; otherwise the SMCC will not function correctly.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DETERMINING POSITION AND VELOCITY AT SEGMENT BOUNDARIES IN 3-POINT SMOOTHER (CIR4) MODE

For the most general case of CIRCLE4 mode moves, with t, deltaP (dP), and deltaP/t all changing, the user may want to compute the state of the system at boundaries of the move segments. The equations below define the position and velocity at segment boundaries. Note that if t and deltaP/t change simultaneously (as in the example), there are acceleration discontinuities, but the equations are still valid.

$$f(n) = [U(n+1) \times t(n) - U(n) \times t(n+1)] / 3[t(n) + t(n+1)]$$

(POSITION CORRECTION FACTOR)

$dP(n) = U(n) + f(n) - f(n-1)$ = DELTA POSITION FOR SEGMENT n

$V(n) = [U(n) + U(n+1)] / [t(n) + t(n+1)]$ = VELOCITY AT END OF SEGMENT n

For the following set of move segments:

```
CIR4        [t(0) = t(1), U(0) = 0] MUST BE ADDED TO BEGINNING
t160        t(1)
U11000      U(1)
t200        t(2)
U29000      U(2)
t88         t(3), t(4), t(5)
U13333      U(3)
U13333      U(4)
U13333      U(5)
t200        t(6)
U29000      U(6)
t160        t(7)
U11000      U(7)
            [t(8) = t(7), U(8) = 0] MUST BE ADDED TO END
```

We can compute the conditions at segment boundaries:

n	t(n)	U(n)	f(n)	dP(n)	V(n) (lines/sec)
					0
0	160	0	1833	1833	35200
1	160	11000	2259	11426*	113778
2	200	29000	133	26874	150517
3	88	13333	0	13200	155148
4	88	13333	0	13333	155148
5	88	13333	-133	13200	150517
6	200	29000	-2259	26874	113778
7	160	11000	-1833	11426	35200
8	160	0	0	1833	0

* 11426 = 11000 + 2259 - 1833, for example

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

USING SMCC'S PROGRAM BUFFER IN RECIRCULATING AND OTHER MODES

The SMCC board has a very flexible program buffer that permits many ways of executing motion programs. The standard buffer is large enough to hold 1790 program lines; with extended memory (Option 5), it can hold 5630 lines (most commands are one line; dual axis standard moves are two lines; parabolic and 'Go Until' moves are two lines per axis). PLC programs are stored in the same buffer, but are stored at the opposite end.

SMCC is capable of continuous execution of programs that are larger than the board's memory. It can do this because its program buffer is dynamic and recirculating: that is, you can write to the buffer while the program is executing, and you can overwrite lines that have already been executed. By monitoring the status of the program buffer properly, you can send new lines in far enough in advance that you ensure continuous execution, but not so far ahead that you overwrite lines that have not yet been executed. This allows the SMCC to execute programs of unlimited size. Of course, a host computer such as a PC must be frequently in contact with the SMCC for this technique.

The technique is very straightforward. The main program buffer was allotted a certain number of lines the last time a "zn" command was given to the card. For instance, "A0z1000" allots 1000 lines to the main program buffer (lines 0-999). Subroutines start on line 1000, and the PLC buffer starts from the opposite end of the buffer memory.

If you load the first 750 lines of a long program (no "END" statement) into the card, you have 250 lines remaining. If you send the card a "PB" (Program Buffer Space) command at this point, the card will return "250", denoting the 250 available lines in the program buffer.

Now, if you start to run the program, and you send "PB" while the card is executing line 50, the card will return "300", where the 300 lines available comprise the 250 at the end plus the 50 already executed. The basic idea here is to continuously monitor the number of available lines with the "PB" command. When the number of available lines reaches a certain number, say 500, you can load another block of commands, say 250 lines, and start monitoring the available buffer again. This process can be repeated indefinitely until the program has been completely sent to the card.

You can tailor the size of the buffer, the threshold size when you start to load more lines, and the size of each block loaded to your own application, but two things should be kept in mind. First, you should start loading more lines when there are still a substantial number of lines left to execute (250 in the above example) so that you do not risk coming to a stop in the middle of the program by letting the card executed the last line already loaded. Second, you should load fewer lines in each block than the space available (250 fewer in the above example) so that you

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

do not try to overwrite an unexecuted line. The SMCC will not allow you to do this: it will return an error and simply ignore that line. The numbers shown above as examples are used in the "DUMP" utility program (which utilizes this technique) and have been found to give very robust operation.

Of course, if you are overwriting program lines, you should not have any branching or looping in the main program: the arrow of execution should move straight through the program. Looping and branching may occur in the subroutines, which are not overwritten. The subroutines should be loaded first, after the main program buffer size has been set with the "zn" command, but before any of the main program buffer has been loaded.

If the program to be executed is small enough to fit in the allotted area, there is no need to use these recirculating techniques. The entire program can be loaded, then run. In this case, the program can also be stored in the card's EAROM (permanent memory), so it will be automatically loaded into active memory (ready for execution) upon power-up or reset.

Alternately, programs may be executed without using the buffer at all by using SMCC's MDI mode (i02=1). In this mode, buffer commands are executed immediately upon receipt by SMCC (or when the previous command is finished). Since the buffer is not being used, only one command in advance may be sent to the card. The BF/ signal ("Buffer" full -- low when not ready to accept a command) or status bit should be monitored in this mode to see when SMCC is ready to accept another command. You do want to stay one block (move segment) ahead so moves can be blended together properly, but you are not allowed to get more than one move ahead. The block execution rate is roughly the same as in non-MDI mode: safely 50 blocks per second for two axes, and 65 blocks per second for one.

For rapid block execution, communications speed may be a problem. If it is, binary (DMA) commands may be used to cut the communications load roughly in half. Binary commands may be used in either MDI (where they are executed promptly) or non-MDI mode (where they are stored in the buffer), but they are most useful in rapid execution in MDI mode. Refer to the DMA communications chapter of the Parabolic section of the manual for details.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

TRAPEZOIDAL MOVE PROGRAMMING ON THE SMCC

The SMCC card offers several different ways of specifying the speed and duration of moves. These moves can be simple point-to-point moves or complex sequences of concatenated segments. The profiles of these moves (velocity vs. time plots) can either be trapezoidal or parabolic. Parabolic moves (direct parabolic or S-Curve) are available only on the Parabolic SMCC. Calculations for these moves are covered in the Parabolic section of the manual. The trapezoidal moves, covered here, can be specified either in terms of feedrate (speed) or move time. It is important to remember that all of the moves discussed here are series of commanded positions. The actual move of the system is also dependent on the physics of the system, along with the feedforward and feedback parameters.

A feedrate-based move is specified by giving an 'F' command before the move that sets the speed. A time-based move is specified by giving a 'T' command before the move that defines the move time. 'F' and 'T' are modal commands; that is, the last one executed stays in force until a new 'F' or 'T' is given. In either type of move, a 't' command sets the time for acceleration between speeds, and to or from a stop. The units of 'F' are related to the reference feedrate (see below). The units of 't' and 'T' are 1/1024 second (referred to as binary milliseconds).

Note: On the Standard SMCC, the 't' and 'i08' values are identical: changing one changes the other. On the Parabolic SMCC, i08 merely provides the power-up value for 't'. On both cards, i08 is used for non-programmed stops and moves (jogs).

Feedrate-Based Moves

In a feedrate-based move, the speed for the constant-velocity portion of the move is given by the expression

V (cts/sec) = i06 * F# / 1024

where i06 is the reference feedrate, also in cts/sec. For a two-axis move, V is the combined vector velocity of both axes. Therefore, the velocity of each axis is

V_x (cts/sec) = i06 * cos a * F# / 1024
V_y (cts/sec) = i06 * sin a * F# / 1024

where

a = arctan (deltaY / deltaX).

The time for a move segment, from the beginning of acceleration to the end of the constant velocity portion, is

T (sec) = deltaP (cts) / V (cts/sec).
T (bin msec) = 1,048,576 * deltaP (cts) / (i06 * F#)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

(For the two axis case, deltaP is the vector distance: sqrt(deltaX^2 + deltaY^2.) The relationship is illustrated in Figure 1. Of course, if the distance is too short to get up to the specified velocity, the move becomes triangular in profile.

A simple feedrate-based point-to-point move can be specified as follows:

```
X=0         ; Define starting position as zero
t256        ; Accel/decel time is 256/1024 sec = 1/4 sec
F512        ; Speed is one-half of i06 reference
X50000      ; Distance is 50000 - 0 = 50000 counts
```

If i06 is 40000 (cts/sec), the time from start of motion to the end of constant velocity (start of deceleration) is 50000 / (40000/2) = 2.5 seconds. Adding the quarter-second deceleration, the total move time is 2.75 seconds.

Time-Based Moves

In a time-based move, the user directly specifies the time (T) from beginning of acceleration until the end of constant velocity. SMCC computes the velocity from this by the equation

V (cts/sec) = deltaP (cts) / T (sec)

(The essential difference between feedrate-specified and time-specified moves is that in the first case, SMCC computes move time from the velocity command, and in the second case, it computes velocity from the time command.)

A simple time-based point-to-point move can be programmed as follows:

```
X=0             ; starting position is zero
t128            ; 128/1024 = 1/8 second accel/decel
T1024           ; 1 second
X10000          ; move to X=10000
```

The card will accelerate up to top speed in 1/8 second (t), stay at top speed until 1 second (T) has elapsed from the beginning of the move, and then decelerate back to zero velocity in 1/8 of a second. The total move thus takes T+t time. The magnitudes of acceleration and velocity are automatically calculated to satisfy the required conditions of the move. The top (slew) velocity is simply the distance over the move time (delta X / T); in this case, 10000 cts/sec. The rate of acceleration is the slew velocity divided by the acceleration time (V_{slew} / t), for this example 80000 cts/sec/sec. The distance covered while accelerating or decelerating is V_{slew} * t / 2, which is 625 counts here. Note that the deceleration must begin 625 counts before the destination. The plot of this move is shown in Figure 2.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

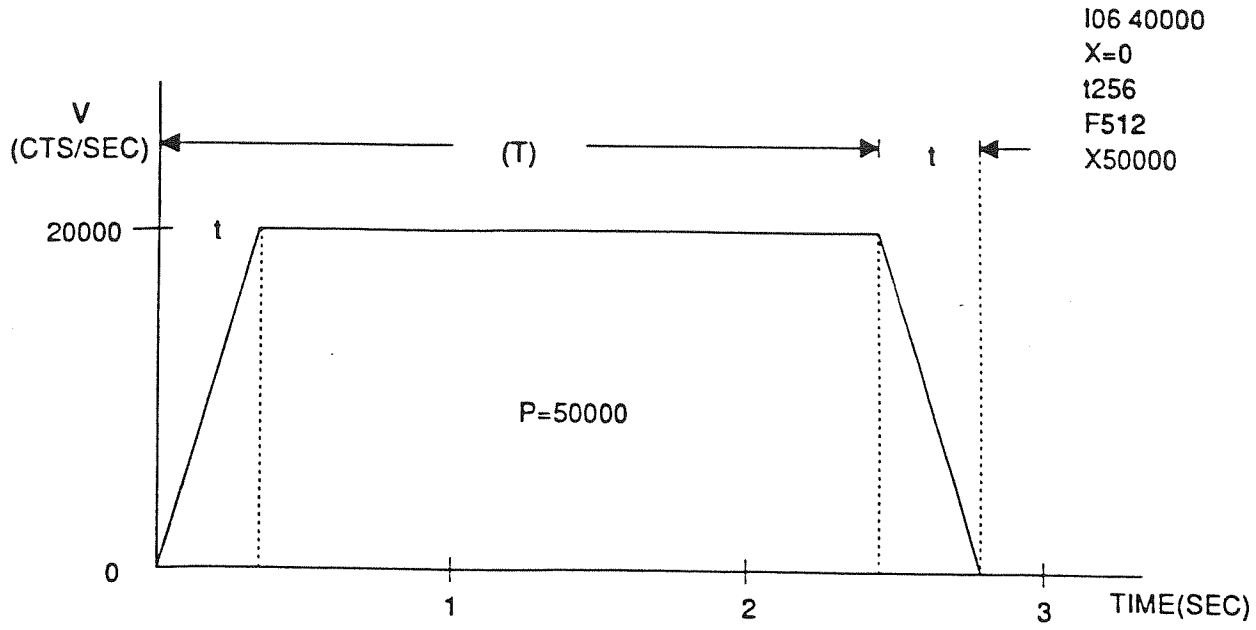


FIGURE 1. FEEDRATE-BASED MOVE

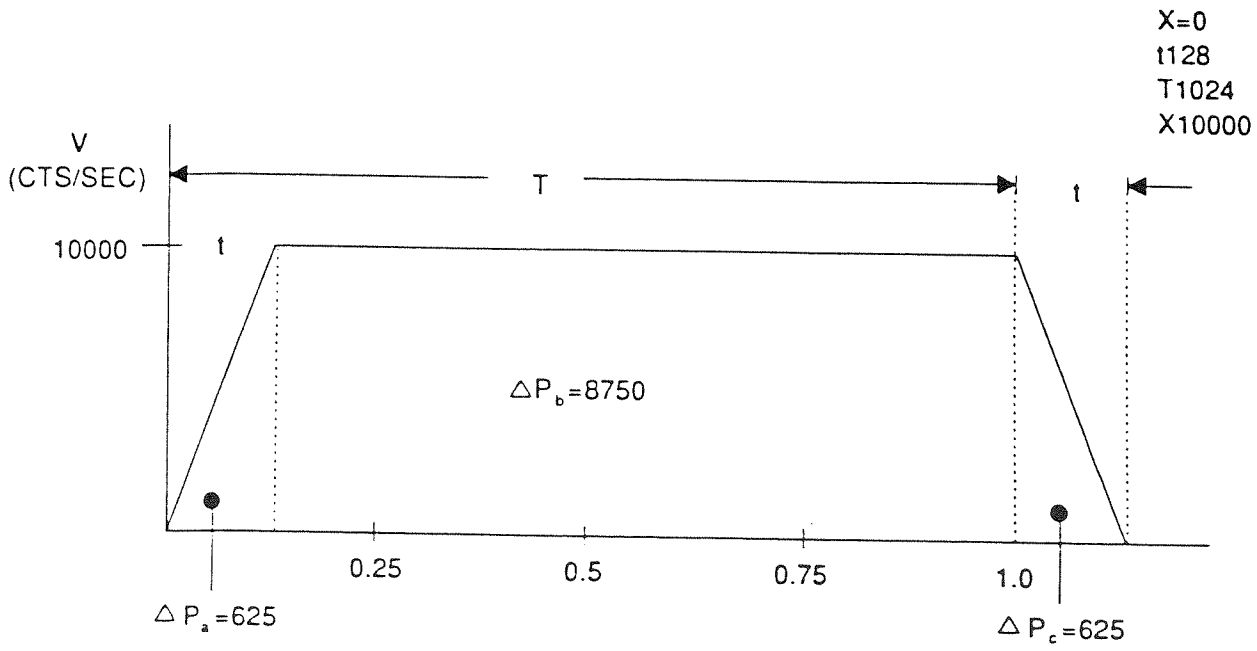


FIGURE 2. TIME-BASED MOVE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Concatenated Moves

What if we wish to add another move segment onto this first one, to make a continuous move? We program this as follows:

```
X=0             ; start at zero
t128            ; accel/decel time is 1/8 sec
T1024           ; move segment time is 1 sec
X10000          ; delta X = 10000
X30000          ; delta X = 20000
```

This move starts just as the first one did: accelerating in 1/8 second to a speed of 10000 cts/sec, and continuing at that speed until 1 second has elapsed from the beginning of the move. We are still 625 counts short of 10000 (X=9375). If no more move segments were specified, it would simply decelerate to a stop.

However, since a new segment follows, the card will accelerate to the new velocity in time t, and remain at that velocity until time T has elapsed from the start of the second acceleration, and then either decelerate to a stop in time t, or accelerate to the slew velocity of a third segment. The slew velocity for the second segment is again delta X / T, or 20000 cts/sec here; the acceleration is $(V_{slew2} - V_{slew1}) / t$, again 80000 cts/sec/sec. When the second T time has elapsed, we are $V_{slew2} * t / 2$ (1250) counts short of our new destination (X=30000). Since there are no more move segments, we decelerate at V_{slew2} / t (160000) cts/sec/sec to our final position. The total move time is 1024 + 1024 + 128 = 2176 bin msec = 2.125 sec. This full move is shown in Figure 3.

Does the math all work correctly for position, velocity, and time? Working from Figure 3, taking delta position as the area under the velocity vs. time curve, we get:

$$\text{delta } P = V_{slew1}*t/2 + V_{slew1}*(T-t) + (V_{slew1}+V_{slew2})*t/2 + V_{slew2}*(T-t) + V_{slew2}*t/2$$

$$\text{delta } P = 625 + 8750 + 1875 + 17500 + 1250$$

$$= 30000 \text{ counts}$$

This method of joining moves is very useful in a lot of applications, particularly in two axes, because it allows blending of moves without stopping, with a slight rounding of the corners near the designated points during the time in which one axis accelerates and the other decelerates. During the slew portions of the moves, there is a straight line move directly between the specified points.

For instance, take the following two-axis program, using incremental X and Y moves:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

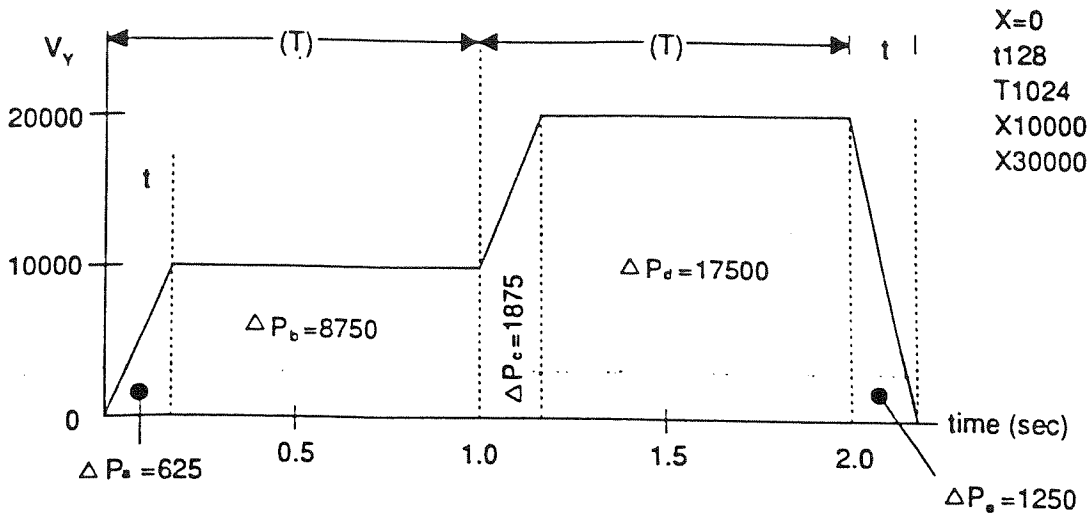


FIGURE 3. CONCATENATED MOVE

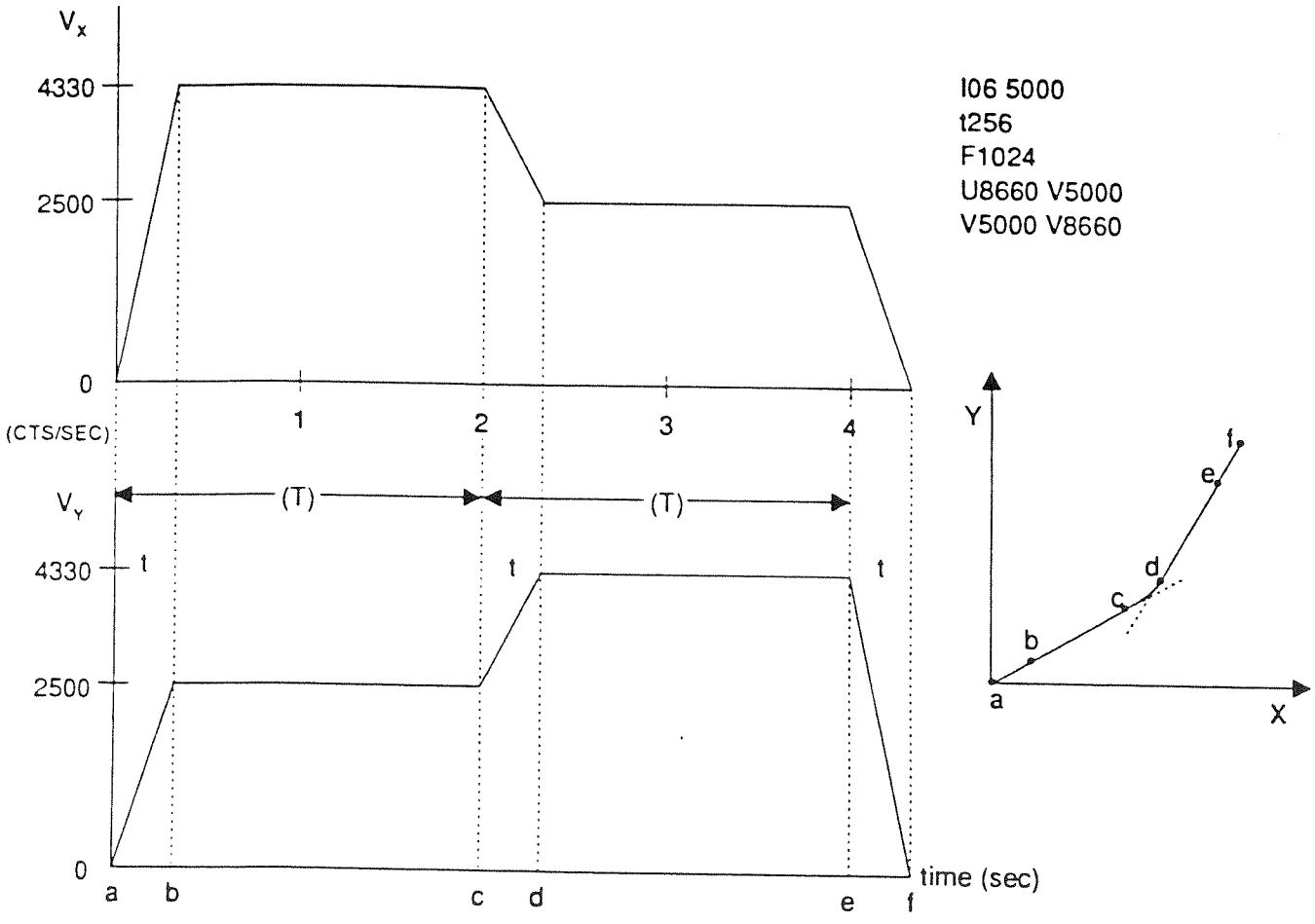


FIGURE 4. TWO-AXIS CONCATENATED MOVE

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
I06 5000          ; reference feedrate = 5000 cts/sec
t256              ; 1/4 second acceleration
F1024             ; vector speed equal to i06
U8660 V5000       ; incremental move at 30 deg (atan[5000/8660])
U5000 V8660       ; incremental move at 60 deg (atan[8660/5000])
```

The length of each move segment is sqrt(8660^2+5000^2) = 10000 cts. Since the F value is 1024, the vector speed equals i06, or 5000 cts/sec. The time from the beginning of each segment to the point where it starts accelerating into the next is 10000 cts / 5000 cts/sec = 2 seconds. The total move time is 2 + 2 + 0.25 = 4.25 sec. Looking at the individual axes, the velocity of X is 5000 cos 30° = 4330 cts/sec in the first segment, and 5000 cos 60° = 2500 cts/sec in the second. The Y velocity is 2500 cts/sec in the first segment and 4330 in the second. These profiles are shown in Figure 4.

Starting from a stopped position, both axes will ramp up to their first velocity in 1/4 second. They will continue along at these velocities until two seconds have elapsed from the beginning, following a straight line path. At this time they will still be short of the first programmed position, but here both axes start to change speed to their new velocities simultaneously, causing the path to curve smoothly. This speed change is performed in the 't' time of 1/4 second. Then both axes proceed at their new constant velocity until 2 seconds have passed from the start of the speed change, during which time they are on a straight-line path directly between the first and second destination points. The curve preceding this caused a smooth blending between the first straight-line path and the second. The result of this blending, however, is that the first destination point was never reached. (It is physically impossible to keep moving around a sharp corner -- infinite acceleration would be required. To actually reach the corner point, a full stop (DWELL) would be required.)

At the end of the second 'T' time, the axes are short of the second destination. Since there are no more moves to be executed, they decelerate to a stop in time 't' (1/4 sec) right on the second destination point.

If you are blending to a move in the opposite direction, you will turn around before reaching the first destination point. For example, the program

```
t128
T1024
X=0
X10000
X0
```

will never reach X=10000. The greatest X position it will reach is 9687 (calculation is left as an exercise).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Transition Point Programming

However, there are some types of move segments where some external calculation will need to be done. For instance, if you want to get to a given position while still at the slew velocity, and then immediately start to accelerate to the new speed (and not pass the point sometime during the acceleration, as this mode does naturally), you will have to specify a different destination position in the program, calculated externally. The calculations are straightforward, as shown below.

The key thing to remember is that the acceleration to a new speed for the next move segment begins at the position where the old move would have to start decelerating to a stop if there were not another move segment. In the first concatenated-move example above, the acceleration to higher speed starts at X=9375, with the first command position at X=10000. So, if you are interested in defining the position where the ramp to a new speed begins, the commanded position is:

$$P_{commanded} = P_{rampstart} + (V_{slew1} * t) / 2$$

The position where the ramp ends is related to the commanded intermediate position by:

$$P_{commanded} = P_{rampend} - (V_{slew2} * t) / 2$$

If you are programming the moves incrementally, the change in position (deltaP) that you specify for a move segment is related to the actual move distance (from start of one ramp to start of next ramp) by:

$$deltaP_{commanded} = deltaP - (V_{slew1}*t_1)/2 + (V_{slew2}*t_2)/2$$

In general, the SMCC's logic for these moves is structured so that the move segment is considered 'over' at the point where it starts to ramp to a new velocity or to a stop. Any M functions that are to be changed between segments are changed at this point. This includes the case where you are decelerating to a stop -- the functions get changed at the beginning of the ramp, not when the destination is reached.

If you wish to run an axis at constant speed, but are breaking the move into pieces to do different things with the other axis, you should work in time-based mode, and keep the deltaP/T ratio constant. However, if you change the little t time, this will cause a speed change on the axis, because the start of the next acceleration period has changed relative to the commanded position, yet the distance to it must be covered in the same time 'T'. To compensate for this, change the programmed distance by adding the term $V_{slew1} * (t_{new}-t_{old}) / 2$.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

CONVERSION OF SMCC RAW HEXADECIMAL DATA TO DECIMAL

When data is taken in raw form from an SMCC memory location, it is returned as ASCII characters representing hex bytes (2 characters per byte), with the least significant byte first for multiple byte values. Data is returned in this form when the 'r' and 'd' commands are used. Below is an example showing how to convert this data into decimal value.

The user requests the X-axis position from a Parabolic SMCC:

r fe48,4 (Return 4 bytes starting at memory location fe48)

The SMCC will return a string such as:

a0c30900

This is to be interpreted as four bytes of hexadecimal data: the least significant byte is a0; the next byte is c3; the next byte is 09; and the most significant byte is 00. Written out in standard form, this value is:

0009c3a0 (base 16)

The conversion from base 16 to base 10 is then a standard mathematical procedure, shown as follows:

```
0   0   0   9   c   3   a   0
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   1 x 0 =          0
|   |   |   |   |   |   16 x 10 =          160
|   |   |   |   |   256 x 3 =              768
|   |   |   |   4096 x 12 =              49152
|   |   |   65536 x 9 =                 589824
|   |   1048576 x 0 =                       0
|   16777216 x 0 =                          0
268435456 x 0 =                             0
                                       --------
                                         639904
```

The decimal value is then 639,904. However, position data on the SMCC has 5 bits of fraction, so we divide by 32 to get the value of the position as an integer number of counts and get 19997 counts.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE 50

PARAMETERIZED MOVE WITH ANALOG SPEED CONTROL AND LEARNED POSITIONS

This program is designed to allow a stand-alone SMCC to do a move of a fixed pattern but with all of its parameters variable by the operator. This example uses the inputs from the standard control panel (the ones that would not affect normal operation of the single axis) for demonstration purposes. An actual application would use the general purpose inputs MI1-MI7. To set the speed transition positions for the move the system is moved by jogging or handwheel following to the desired location, and the proper switch is toggled. This stores the location in a P parameter that the programmed move uses. There are four potentiometers hooked up to the SMCC's analog inputs. Three of these control the speed of a part of the move; the fourth controls the dwell time at the end of the move.

Note the use of "marker bits" in the PLC. These are used to make sure a given action only occurs once per switch toggle. Because the PLC program cycles repeatedly and quickly, actions can occur multiple times on a pressed switch (for example) unless one of the conditions that caused the action is changed with the action. This is what the marker bits are for. (In this case, it is not too important that the action only be taken once, but it could be critical in other applications.) The SEND commands are for debugging purposes only.

```
A0z50
PURGE
C21=ff3a              ; CPU MUX byte
M21=C21.0             ; YMIL (Y minus limit)
M22=C21.1             ; YPLL (Y plus limit)
M23=C21.2             ; YJPL (Y jog plus)
M24=C21.3             ; YJMI (Y jog minus)
C22=a3a7              ; Empty area in memory
M31=C22.0             ; Marker bit for M21 & M22
M33=C22.2             ; Marker bit for M23
M34=C22.3             ; Marker bit for M24
M35=C22.4             ; Controls when pots are read
SET35                 ; Always set (for now)
C10=fe8c              ; A/D 0 input register
C11=fe8e              ; A/D 1 input register
C12=fe90              ; A/D 2 input register
C13=f391              ; A/D 3 input register
P8=40                 ; So max P7 of 255 gives 10240

t64                   ; Accel/decel time
F1024                 ; Full speed for rapid move
X(P2+P10)             ; Rapid move
F(P13)                ; Feedrate for coarse move
X(P3+P10)             ; Coarse move
F(P14)                ; Feedrate for medium move
X(P4+P10)             ; Medium move
F(P15)                ; Feedrate for fine move
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
X(P5+P10)              ; Fine move
DWE(P6)                ; Wait in final position
F1024                  ; Full speed for retract
X(P1+P10)              ; Retract move
END

************************************************************************
PLC branches to set the P parameters for the above move. Control
panel switches (to be replaced by buttons into Machine Inputs),
analog pots, and a handwheel encoder form the user interface.
************************************************************************

IF M21=1               ; YMIL & YPLL both set
AND M22=1              ; means hold switch set
AND M31=0              ; Not set last cycle
THEN P2=AX             ; Param. gets current position
AND M31=1              ; Mark that has been set
AND SEND "RAPID_END_POSITION_SET"

IF M21=0               ; If hold switch released
AND M22=0
AND M31=1              ; Not released last cycle
THEN M31=0             ; Mark that has been released

IF M23=1               ; YJPL set
AND M24=0              ; but YJMI not set
AND M33=0              ; YJPL not set last time
THEN P3=AX             ; Param. gets current position
AND M33=1              ; Mark that has been set
AND SEND "COARSE_END_POSITION_SET"

IF M23=0               ; YJPL released
AND M33=1              ; Not released last cycle
THEN M33=0             ; Mark as released

IF M24=1               ; YJMI set
AND M23=0              ; but YJPL not set
AND M34=0              ; YJMI not set last time
THEN P4=AX             ; Param. gets current position
AND M34=1              ; Mark as set
AND SEND "MEDIUM_END_POSITION_SET"

IF M24=0               ; YJMI released
AND M34=1              ; Not released last cycle
THEN M34=0             ; Mark as released

IF M23=1               ; Both YJPL and YJMI set is
AND M24=1              ; Prejog (=) switch set
AND M33=0              ; Not set last time
AND M34=0              ; Ditto
THEN P5=AX             ; Param. gets current position
AND M33=1              ; Mark as set
AND M34=1              ; Ditto
AND SEND "FINE_END_POSITION_SET"
                       ; This branch converts potent.
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
                    ; voltages into speed and time
                    ; parameters.
IF M35=1            ; Always true (for now)
THEN P7=C10,2       ; Prelim value for dwell time
AND P13=C11,2       ; Coarse speed
AND P14=C12         ; Medium speed
AND P15=C13         ; Fine speed
AND P6=(P7*P8)      ; Dwell time in bin msecs
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE 51

TIME-BASED FOLLOWING PROGRAM WITH THUMBWHEEL-CONTROLLED RATIO AND BACKGROUND CONTROL

This program controls two axes that are slaved to an open-loop master axis. This is for a printing application, where the printing roll is run open loop. The pull roll, which maintains tension on the paper as it comes out of printing, and the knife roll, which cuts the paper to size (one cut per revolution), are both slaved to the print roll. The pull roll must match surface speed with whatever size print roll is being used, so the circumference of the print roll is entered on thumbwheels, and the following factor computed from the thumbwheel value. The knife roll matches the print roll revolution for revolution, regardless of print roll size, so no thumbwheel is needed. However, because of the gear ratios and encoder resolutions, the following factors do not work out exactly right and there is some drift. The PLC program monitors this drift and issues corrections repeatedly as needed so there is no net drift. To bring the knife roll into phase with the print roll, there are pushbuttons to advance and retard the knife.

```
; X-axis controls knife wheel and Y-axis controls
; pull roll.
; This program assumes:
; ** 2500 line per rev encoder on print roll;
;    quadrature channels into X handwheel input;
;    index pulse into X user flag input;
;    encoder rotates 2x for each print roll rev;
;    4x decoding used (i29)
; ** 1024 line per rev encoders on both slave axes
;    4x decoding used for both (i39, i59)
; ** Knife roll (X axis) motor/encoder rotates 3x
;    for each knife roll rotation
; ** Pull roll (Y axis) motor/encoder rotates 72/28x
;    for each pull roll rotation
; ** Pushbutton wired into MI5: when pressed, causes
;    thumbwheels to be read again
; ** Thumbwheels only used for Y axis now (pull
;    roll): left 5 digits of 2nd row of board
;    (ffb9 addresses 4,5,&6). Entry should be
;    circumference of print roll to 1/1000 inch.
; ** Pushbuttons to phase in X axis (knife roll)
; ** Coarse advance (+10 counts) into MI1
; ** Fine advance (+1 counts) into MI2
; ** Coarse retard (-10 counts) into MI3
; ** Fine retard (-1 counts) into MI4
; ** Default assignments for M11-M15 as matching up
;    to MI1-MI5 have not been disturbed
%65498          ; to make sure beginning gets done
i160            ; ditto
A0z100
PURGE
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
; Parameter definitions
I16 0
t100
T100
DWE200
C10=ffb8        ; Parallel port input address
C09=ffb9        ; Parallel port output address
C29=ffca        ; User flag interrupt address
M29=C29.6       ; True when has been interrupted
C21=a3a7        ; Empty memory location (PARAB)
C22=a3a8        ; 2nd empty memory location (PAR)
M21=C21.0       ; Control bit to start read
M22=C21.1       ; Control bit to increment read
M31=C21.2       ; Marker bit for MI1
M32=C21.3       ; Marker bit for MI2
M33=C21.4       ; Marker bit for MI3
M34=C21.5       ; Marker bit for MI4
M35=C21.6       ; Marker bit for MI5
M36=C21.7       ; Bit to control parameter change
M30=C22.0       ; Marks any pushbutton pressed
M63=C22.1       ; True after first index captured
i00 0               ; No following error shutdown
i12 60              ; Default calc time -- <  delay
i27 10              ; Following factor
i47 2048            ; Following factor
i26 3               ; proper capture of interrupt
i14 0               ; Make sure no analog time control
i28 3               ; X position following inactive
i48 3               ; Y position following inactive
i62 1               ; Turn PLC on
RESET P             ; Make all zero to start
SET P10=96          ; damping & counts -> frac
SET P13=32          ; counts -> frac
SET P3=147456       ; Numerator term for pull roll
SET P4=815885       ; Denominator term for pull roll
SET P1=4915         ; X-axis move in 50 milliseconds
RESET 31            ; To mark as not pushed
RESET 32            ; Ditto
RESET 33            ; Ditto
RESET 34            ; Ditto
SET 35              ; M35 set so will read thumbwheels
RESET 36            ; So i47 not changed immediately
RESET 63            ; M63=0 to mark first index
GOSUB 00            ; Initial read of thumbwheels
I16 3               ; Put in X time base following

DWE200              ; Make sure all is calculated
mm ffca,1           ; Enable interrupts
t1                  ; Accel/decel time: instant at 1st
T50                 ; Move time
FOR 0               ; loop until external stop
GOSUB 00 IF (M35=1) ; To read thumbwheel value
SET P11=(P1+P12/P10)
SET P12=0
U(P11) V (P2) ; move both axes in sync
NEXT
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
;
;*******************************************************
; Main thumbwheel decoding routine
;*******************************************************
L00
; This subroutine reads the left 5 thumbwheels of
; the bottom row of the first thumbwheel board
; and assembles into a single value -- possible
; range of 0 -- 99,999
mm ffb9,6           ; Set address 6 (1's digit)
SET P7=0            ; To start
SET P9=1            ; Multiplier: 1's digit to start
GOSUB 03            ; Read lo nibble  -> P8
SET P7=(P7+P8*P9)   ; Add into value
mm ffb9,5           ; Set address 5 (10's & 100's)
SET P9=10           ; For 10's digit
GOSUB 02            ; Read hi nibble  -> P8
SET P7=(P7+P8*P9)   ; Add into value
SET P9=100          ; For 100's digit
GOSUB 03            ; Read lo nibble  -> P8
SET P7=(P7+P8*P9)   ; Add into value
mm ffb9,4           ; Set address 4 (1K's & 10K's)
SET P9=1000         ; For 1000's digit
GOSUB 02            ; Read hi nibble  -> P8
SET P7=(P7+P8*P9)   ; Add into value
SET P9=10000        ; For 10000's digit
GOSUB 03            ; Read low nibble  -> P8
SET P7=(P7+P8*P9)   ; Add into value for final value
RESET 35            ; So will not read again
SET P2=(P3*P7/P4)   ; New Y delta move
RET                 ; End of main subroutine

; This subroutine decodes the value of the right-
; hand digit of a pair of thumbwheels (high nibble)
; and returns the value in P8
L02
SET P8=0
CASE 10, 10H        ; Digit value is in C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET

; This subroutine decodes the value of the left-hand
; digit of a pair of thumbwheels (low nibble) and
; returns the value in P8
L03
SET P8=0
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
CASE 10, 10L      ; Digit value is in C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET

;*****************************************************
; Pushbutton routines for temp. advance of pull roll
; Pressing a button changes speed -- releasing it
; changes it back.  Other buttons are 'locked out'
; until the first one pressed is released.
; M11 is MI1 -- coarse advance button
; M12 is MI2 -- fine advance button
; M13 is MI3 -- coarse retard button
; M14 is MI4 -- fine retard button
; M31,M32,M33, &M34 are marker bits for M11-M14
; M30 marks if any buttons are pressed
IF M11=0           ; Coarse advance pushed
AND M30=0          ; but none already down
THEN P15=P15+320   ; 10 count advance
AND M31=1          ; To mark that it is pushed
AND M30=1          ; To mark that any is pushed
AND SEND "Coarse advance Pushed"

IF M12=0           ; Fine advance pushed
AND M30=0          ; but none already down
THEN P15=P15+32    ; 1 count advance
AND M32=1          ; To mark that it is pushed
AND M30=1          ; To mark that any is pushed
AND SEND "Fine advance Pushed"

IF M13=0           ; Coarse retard pushed
AND M30=0          ; but none already down
THEN P15=P15-320   ; 10 count retarding
AND M33=1          ; To mark that it is pushed
AND M30=1          ; To mark that any is pushed
AND SEND "Coarse Retard Pushed"

IF M14=0           ; Fine retard pushed
AND M30=0          ; but none already down
THEN P15=P15-32    ; 1 count retarding
AND M34=1          ; To mark that it is pushed
AND M30=1          ; To mark that any is pushed
AND SEND "Fine Retard Pushed"

; These PLC branches mark release of a button
IF M11=1           ; button not pressed
AND M31=1          ; but just was
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
THEN M30=0          ; To mark no button pushed
AND M31=0           ; To mark as not pushed
AND SEND "Coarse advance Released"

IF M12=1            ; button not pressed
AND M32=1           ; but just was
THEN M30=0          ; To mark no button pushed
AND M32=0           ; To mark as not pushed
AND SEND "Fine advance Released"

IF M13=1            ; button not pressed
AND M33=1           ; but just was
THEN M30=0          ; To mark no button pushed
AND M33=0           ; To mark as not pushed
AND SEND "Coarse Retard Released"

IF M14=1            ; button not pressed
AND M34=1           ; but just was
THEN M30=0          ; To mark no button pushed
AND M34=0           ; To mark as not pushed
AND SEND "Fine Retard Released"

; This branch acts on push of button to MI5
; Since MI5 has reverse polarity of MI1-MI4 (high
; true) and button is low when pushed
IF M15=1            ; 'Reread' button pressed
AND M35=0           ; but not already
THEN M35=1          ; Set so thumbwheels will be read
AND SEND "Thumbwheel Read Pressed"

; This branch causes startup of index-triggered
; position correction routine.  It captures the
; X position and sets the 'desired position' equal
; to it.  The factor of 32 (P13) is used to make
; sure overflow is handled correctly.
IF M29=1            ; User flag tripped
AND M63=0           ; for the first time
THEN P14=HX         ; Captured X position
AND P15=(P14*P13)   ; Desired position = this
AND P12=0           ; No incremental move to catch up
AND M63=1           ; No longer first time thru
AND SEND "UserFlagTripped1stTime"

; This branch acts on subsequent index-pulse cap-
; tures.  It compares the actual captured position
; to the desired position, and causes an incremental
; move to make up the distance.
IF M29=1            ; User flag tripped
AND M63=1           ; not for the first time
THEN P14 = HX       ; Captured X position in counts
AND P15=P15+196608  ; Add 6144 counts * 32
AND P14=(P14*P13)   ; To bring to fractional counts
AND P12=(P15-P14)   ; # of fractional counts behind
AND "wffca,1"       ; To re-enable interrupts
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE 52

USE OF PARABOLIC MODE FOR TRAPEZOIDAL MOVES

This program provides a profiled back and forth move which is done at a rate set by a 4-digit thumbwheel value. Two things are noteworthy about this program: first, the use of parabolic mode to do trapezoidal moves. This is done so that the transitions start and stop at exactly the points specified in the program (if the moves were defined by standard trapezoidal commands, the specified positions would be passed sometime during the transitions). What has to be kept in mind here is the relation between distance (U), velocity (Vn) and time (t) to keep the desired profile for a segment: U = V * t for a constant velocity segment; and U = 1/2 * (V1 + V2) * t for an acceleration segment. Second, the program can perform the move at different speeds by allowing the feedrate override value (% value) to be set by the thumbwheels. A reading of 9999 gives full speed; 0001 gives the minimum speed (0000 means zero speed, which means you get stuck in the program by executing it at zero rate). The % value is actually set using the buffered POT command.

```
A0z100                ; Clear program buffer & subrout.
PURGE                 ; Empty the PLC buffer
SET C10=ffb8          ; Parallel port input address
SET C20=febd          ; Address of control bits
SET M20=C20.5         ; Data gathering control bit
SET P10=64            ; Scales thumbwheel to % value
SET P11=10            ; Ditto
I06 69905             ; So feedrates can be set as RPM
SLEW 65535            ; Immediate change in % value
GOSUB 00              ; Read thumbwheels  -> P7
SET P1=(P7*P10/P11)   ; Convert to % value
POT (P1)              ; Set execution rate
SET20                 ; Start data gathering
t100                  ; Accelerate segment
U10000:3000           ; 2.5 revs ending at 3000 rpm
t120                  ; Slew segment
U24000:3000           ; 6 revs at constant 3000 rpm
t60                   ; Decelerate segment
U8400:1200            ; 2.1 revs ending at 1200 rpm
t100                  ; Slew segment
U8000:1200            ; 2 revs at constant 1200 rpm
t17                   ; Accelerate segment
U1724:1800            ; 0.431 revs ending at 1800 rpm
t81                   ; Slew segment
U9772:1800            ; 2.443 revs at constant 1800 rpm
t52                   ; Decelerate to stop
U3104:0               ; 0.776 revs ending at 0 rpm
t100                  ; Accelerate back
U-10000:-3000         ; -2.5 revs ending at -3000 rpm
t183                  ; Slew segment
U-36600:-3000         ; -9.15 revs at constant -3000 rpm
t40                   ; Decelerate segment
U-6400:-1800          ; -1.6 revs ending at -1800 rpm
```

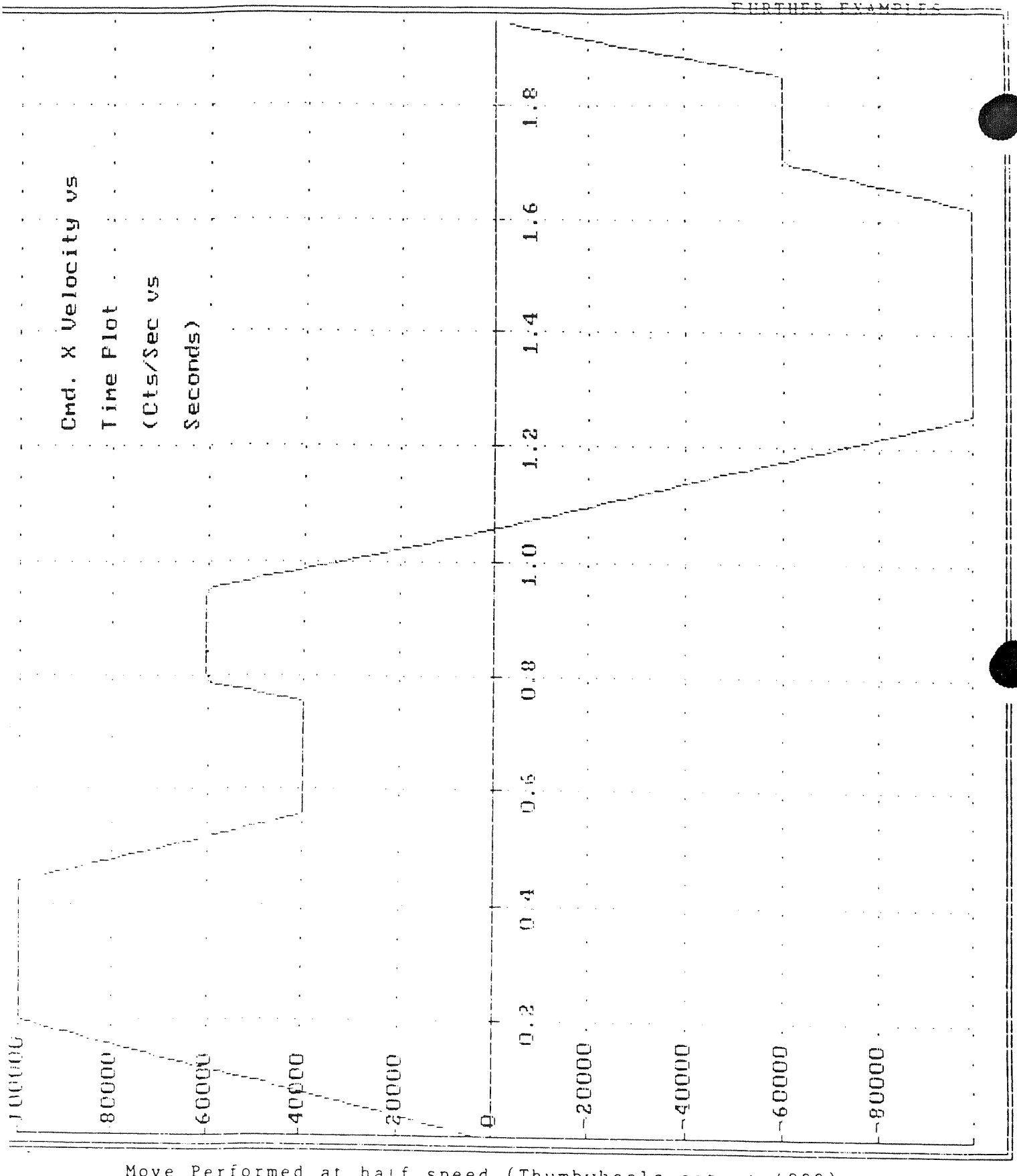

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
t75                    ; Slew segment
U-9000:-1800           ; -2.25 revs at constant -1800 rpm
t50                    ; Final deceleration
U-3000:0               ; -1.5 revs ending at 0 rpm
RES20                  ; End data gathering
END
L00                 ; Subroutine to read thumbwheels
mm ffb9,1           ; 2nd pair of TWs on top row
SET P7=0            ; To start
SET P9=1            ; Multiplier: 1's digit to start
GOSUB 02            ; Read hi nibble -> P8
SET P7=(P7+P8*P9)   ; Add into accumulated value
SET P9=10           ; 10's digit
GOSUB 03            ; Read low nibble -> P8
SET P7=(P7+P8*P9)
mm ffb9,0           ; 1st pair of TWs on top row
SET P9=100          ; 100's digit
GOSUB 02            ; Read hi nibble -> P8
SET P7=(P7+P8*P9)   ; Add into accumulated value
SET P9=1000         ; 1000's digit
GOSUB 03            ; Read low nibble -> P8
SET P7=(P7+P8*P9)   ; Add into accumulated value
RET

L02      ; Subroutine to decode right digit of pair
SET P8=0
CASE 10, 10H      ; value in hi nibble of C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET

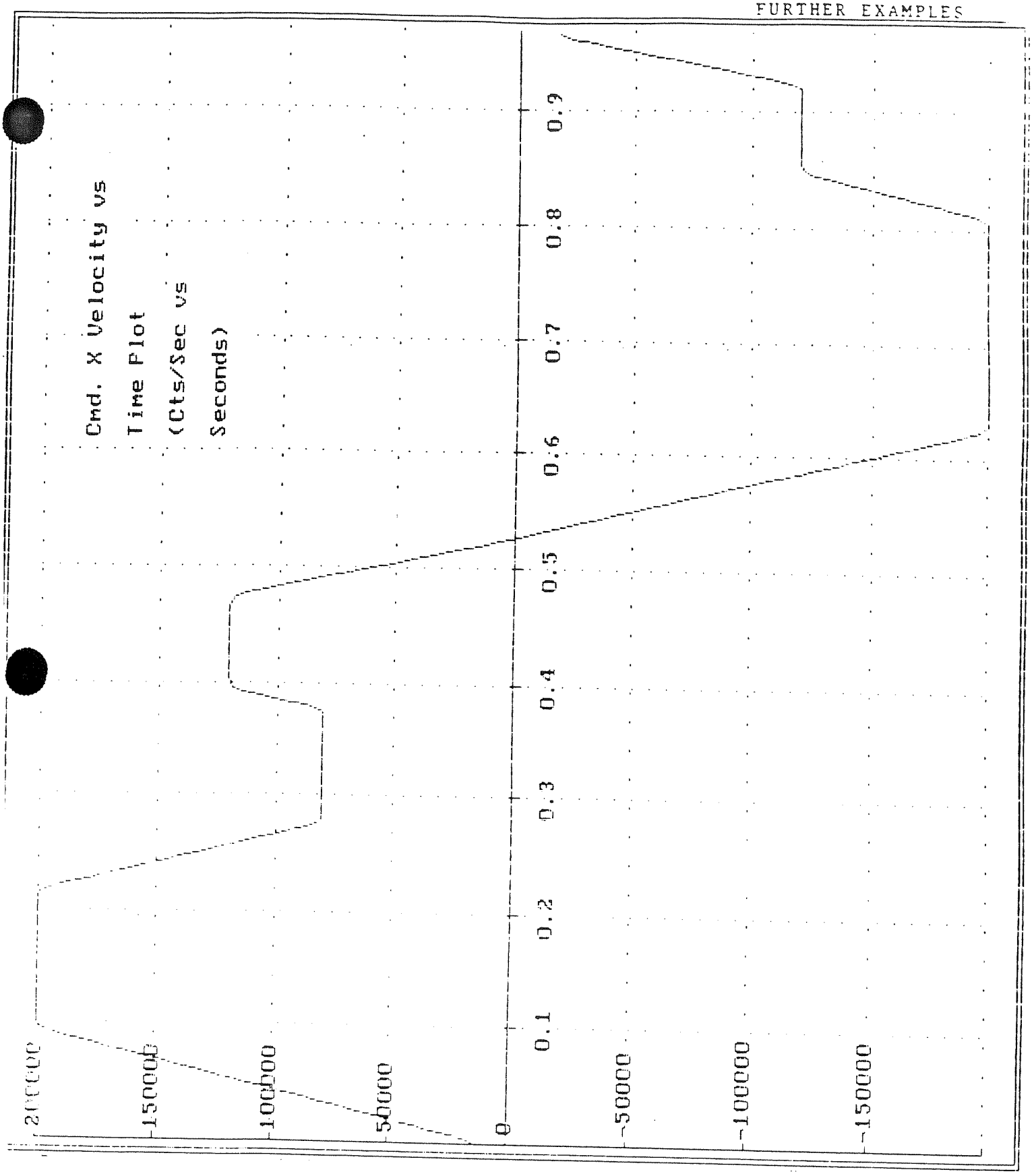
L03      ; Subroutine to decode left digit of pair
SET P8=0
CASE 10, 10L      ; value in lo nibble of C10
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Move Performed at half speed (Thumbwheels set at 4999)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Move performed at full speed (Thumbwheels set at 9999)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

EXAMPLE 53

THUMBWHEEL-CONTROLLED S-CURVE MOVES

This program performs an index move whose length is determined by 4-digit thumbwheel input. The thumbwheel digits are located at addresses 6 & 7 of the thumbwheel multiplexer board. The TW value will be interpreted as hundredths of an inch if Machine Input 5 (MAI5/ -- pin A28 of JMACH) is not forced low (normal state), or as tenths of a millimeter if it is forced low. The thumbwheels are read the first time thru the program; thereafter they are only read if Machine Input 6 (MAI6/ -- pin A27 of JMACH) has been taken low since the beginning of the last move (use a normally open switch that closes to ground). A move is done each time the STEP/ input (pin 9 on JPAN) is taken low. Each move has S-curve accel. and decel. achieved achieved using CIRCLE4 mode. The profile is different depending on whether the move is less than or greater than 21 inches. On a given side of this boundary, different move lengths are done by changing the length of the constant-speed part of the move.

```
A0z100
PURGE
; Parameter setting and definition
C25=ffb8              ; parallel port input
C21=a3a7              ; unused memory space
M26=C21.1             ; Marker bit to read TW's
C10=a267              ; MSByte of P10
M10=C10.7             ; Sign bit of P10
C20=febd              ; For data storage
M20=C20.5             ; Data storage control bit
P11=96911             ; Conversion from mm to counts
P12=100000            ; Conversion divisor
P13=246154            ; Conversion from in to counts
; Buffer program starts here
START                 ; Step input causes run til STOP
SET 26                ; So thumbwheels read 1st time
DWELL 80              ; So the setting of bit takes
FOR 0                 ; Loop indefinitely
GOSUB 00 IF (M26=1) ; Read thumbwheels if told to
SET 20                ; Start data storage
CIRCLE4               ; Put into S-curve mode
t(P2)                 ; 1/2 of accel time
U(P3)                 ; accel distance
t(P4)                 ; slew time
U(P5)                 ; slew distance
t(P2)                 ; 1/2 of decel time
U(P3)                 ; decel distance
DWELL 0               ; To catch end of move for store
RESET 20              ; End data storage
STOP                  ; To await next run signal
START                 ; Step input causes run til STOP
NEXT
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
L00             ; Thumbwheel reading routine
mm ffb9,07      ; Address low two digits of thumbwheel
GOSUB 02        ; Read low digit (hi nibble) -> P8
SET P7=(P8)     ; Accumulate value in P7
SET P9=10       ; Multiplier for 10's digit
GOSUB 03        ; Read hi digit (lo nibble) -> P8
SET P7=(P7+P8*P9) ; Add into P7
mm ffb9,06      ; Address hi two digits of thumbwheel
SET P9=100      ; Multiplier for 100's digit
GOSUB 02        ; Read low digit (hi nibble) -> P8
SET P7=(P7+P8*P9) ; Add into P7
SET P9=1000     ; Multiplier for 1000's digit
GOSUB 03        ; Read hi digit (Lo nibble) -> P8
SET P7=(P7+P8*P9) ; Add into P7
GOSUB 04 IF (M15=1) ; For metric scaling
GOSUB 05 IF (M15=0) ; For English scaling
RESET 26            ; Thumbwheels have been read
RET

L02             ; Decodes right digit of TW pair
CASE 25, 10H    ; Branch on high nibble of C25
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET

L03             ; Decodes left digit of TW pair
CASE 25, 10L    ; Branch on low nibble of C25
SET P8=0
SET P8=1
SET P8=2
SET P8=3
SET P8=4
SET P8=5
SET P8=6
SET P8=7
SET P8=8
SET P8=9
RET

L04             ; Metric scaling routine
SET P10=2250    ; Shortest OK move (in mm/10)
SET P10=(P7-P10)    ; Compare to specified length
GOTO 40 IF (M10=0)  ; Not too short
SET P7=2250         ; Too short -- make OK
L40
SET P10=8915    ; Longest OK move (in mm/10)
SET P10=(P10-P7)    ; Compare to specified length
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

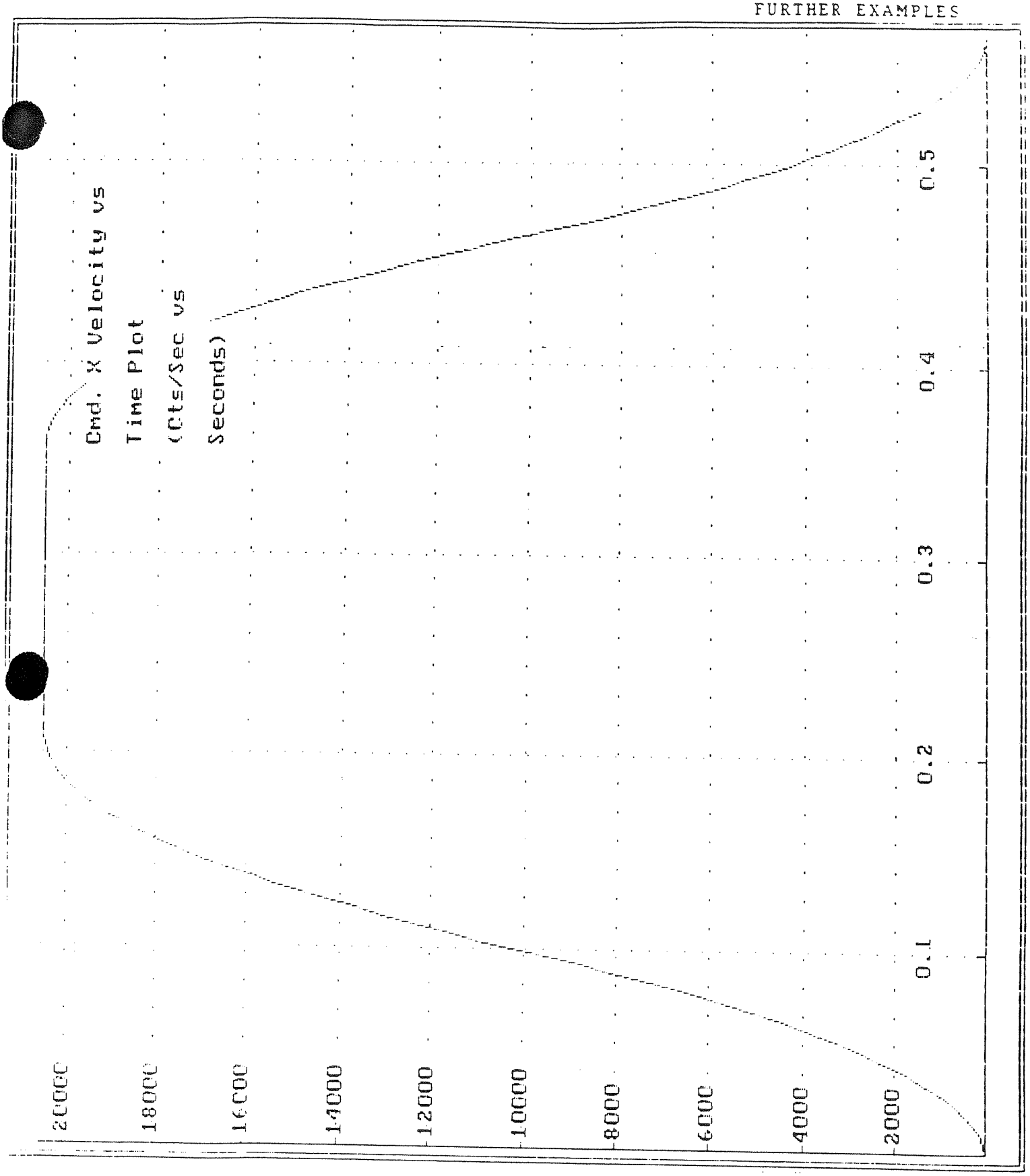
```
GOTO 41 IF (M10=0)   ; Not too long
SET P7=8915          ; Too long -- make OK
L41
SET P10=5334  ; Longest 'short' move
SET P10=(P10-P7)     ; Compare to specified length
SET 02 IF (M10=1)    ; Mark for long move
RES 02 IF (M10=0)    ; Mark for short move
SET P1=(P7*P11/P12)  ; Convert to encoder counts
GOSUB 06             ; Calculate move parameters
RET

L05            ; English scaling routine
SET P10=886    ; Shortest OK move (in .in/100)
SET P10=(P7-P10)     ; Compare to specified length
GOTO 50 IF (M10=0)   ; Not too short
SET P7=886           ; Too short -- make OK
L50
SET P10=3510  ; Longest OK move
SET P10=(P10-P7)     ; Compare to specified length
GOTO 51 IF (M10=0)   ; Not too long
SET P7=3510          ; Too long -- make OK
L51
SET P10=2100         ; Longest 'short' move
SET P10=(P10-P7)     ; Compare to specified move
SET 02 IF (M10=1)    ; Mark for long move
RES 02 IF (M10=0)    ; Mark for short move
SET P1=(P7*P13/P12)  ; Convert to encoder counts
GOSUB 06             ; Calculate move parameters
RET

L06            ; Long move calculation routine
GOTO 60 IF (M10=1) ; For long move
SET P2=72          ; Short accel time
SET P3=962         ; Short accel distance
GOTO 61
L60
SET P2=108         ; Long accel time
SET P3=2160        ; Long accel distance
L61
SET P5=(P1-P3)     ; Total length minus accel len
SET P5=(P5-P3)     ; & minus decel length
SET P4=(P5*P2/P3)  ; Slew time propor. to slew dist
RET

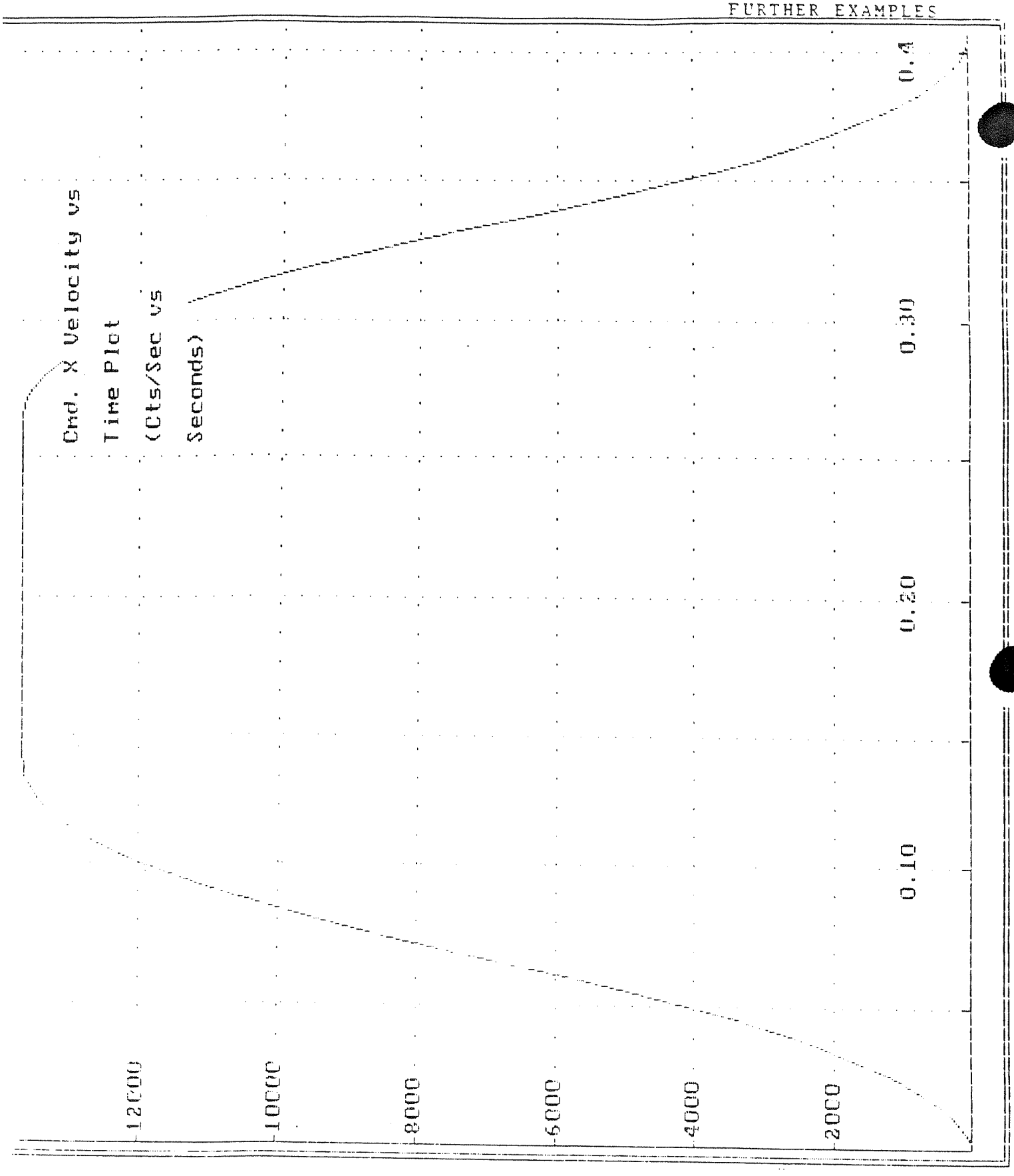

IF M16=1           ; TW read button pressed
AND M26=0          ; but wasn't pressed before
THEN M26=1         ; then set marker bit
END
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Move performed with thumbwheels set to 30.00"

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



Move performed with thumbwheels set to 15.00"

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

UNDERSTANDING SMCC'S TIME BASE CONTROL

The SMCC has sophisticated and flexible control of the time base used in servo programs. For the novice user, the default settings are satisfactory for a wide variety of applications, and there is no need to worry about an understanding of how SMCC handles these issues. For the more advanced user, the card provides many capabilities to tailor the card to special applications requiring various rates of execution. For these people, some understanding of the methods SMCC uses is essential to achieving the desired results. This application note is intended to provide that understanding.

Servo Cycle Time

As sent from the factory, the SMCC has a servo cycle time of 976 microseconds (usec). That is, every 976 usec, SMCC does the calculations to close the feedback loop, sending a new command to the amplifier, and also updates the trajectory position. The basic unit of time in the motion program is the binary millisecond (1/1024 sec). These are the units of the 't' and 'T' commands, and implicitly the time set by 'F' commands. Many users need to know no more than this to operate their systems.

Now we look at how SMCC sets these values. Parameter i05 controls the servo cycle time. The value of i05 is the starting value of a counter that counts down once every microsecond. When it gets to zero, an interrupt is generated that starts the servo cycle calculations for an axis, and the counter is reset to i05. At i05's default value of 487, this happens every 488 usec, with alternating cycles for the X and Y axes. This means that each axis has a 976 usec servo cycle time. The general formula is

$$\text{Servo cycle time (usec)} = 2\ (i05 + 1). \qquad (1)$$

If you have a multiple-card system, the i05 value on card A0 controls the servo cycle time for all of the cards. It sends out the servo inpterrupt signal on the daisy-chained SYNC line. On each card jumper E2 controls whether the card sends or receives this signal. This jumper should be installed on card A0 (send), and removed on all the rest (receive).

Programming Time Units

The "percent" (%) value, together with i05, controls what the programming time units are (units for T and t commands). The formula for the time units is

$$dT\ (\text{usec}) = \frac{65536 * \text{Servo time(usec)}}{\%\ \text{value}} = \frac{65536 * 2^{(i05+1)}}{\%\ \text{value}}. \qquad (2)$$

For the default values of i05 = 487 and dT = 1/1024 sec (=976.56 usec), we get a % value of 65498. (65498 is to 65536 as the

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

servo cycle time of 976 usec is to the time unit of 976.56 usec.) The parameter i07 sets the % value on power-up or reset; the value of i07 as sent from the factory is 65498. The range of % and i07 values is 0 to 65535.

The time units for DWELL commands are determined by i07, not the % value (substitute 'i07' for '% value' in Equation 2). This allows DWELL times to stay the same even as the % value is varied. (However, the Standard SMCC inserts a 't' time between consecutive DWELLs, and this time varies with the % value.) If you want your standing-still time to vary with the % value, use a U0 command, a WX0 command, or, on the Parabolic SMCC, a DLY (delay) command.

By changing the % value, we can change the rate of execution of a program without changing the path it follows at all. For example, the program

```
X=0
Y=0
t256
T768
X1000 Y1000
```

takes 1024 time units (T + t) to execute. At the default values of i05 = 487 and % = 65498, the time units are 1/1024 sec, and the move takes 1 second. If the % value were instead 32749 (half of 65498), each time unit would be 1/512 second, and the move would take 2 seconds.

*** Problem #1: If you wished your time units always to be 1/1000 sec (1 millisecond), with i05 = 487, what value would you want for i07? (Solution at end of note.)

On-Line Time-Base Control

There are four ways to change the % value after power-on or reset. First is the on-line (non-buffered) %N,n command. The 'N' argument is the new desired % value. The 'n' value is the amount that the current % value is incremented each servo cycle (called the slew rate). If no 'n' argument is given, the last commanded slew rate is still in effect. The default slew rate value on power-up or reset is always 64. For the standard servo cycle time, this corresponds to changing over the full range of % values in one second. The range of slew rate values is 0 (no change -- a value of 1 means 64 seconds to cover the full % range at standard servo cycle) to 65535 (which always causes a change to the new desired % value in one servo cycle). Execution of a %N,n command begins immediately upon rreceipt by the card. A % command with no arguments is a request for the current % value.

Motion Buffer Time-Base Control

The second way to change the % value is through the use of the motion buffer commands POT N and SLEW n. The POT N command, when

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

encountered during the execution of a program, causes the % value to move toward N at a slew rate controlled by the most recent SLEW n command or % command with an 'n' argument.

Voltage Time-Base Control

The third way to control the % value is through an external analog voltage level (usually from a potentiometer) with i14 = 1. This voltage is input to the card at the ANA2 pin (J1-C1), and processed through an 8-bit A/D converter, whose digital value is multiplied by 256 to give the % value. Zero volts in corresponds to a % value of zero; +5 volts in corresponds to a % value of 256*255 = 65280. (With the analog input conditioning option (#3), the full range of inputs is -10V to +10V.) This mode is intended for manual control of speed of program execution; it is not intended to be very precise on the control of time. Do not try to control the speed of execution of multiple cards this way; the A/D converters on the different cards cannot be guaranteed to convert exactly alike, and loss of synchronization will result.

Frequency Time-Base Control

The final way to change the % value is through an external digital frequency (as from a "master encoder") brought in at the handwheel input with i16 = 2 or 3. This input signal can either be quadrature or pulse and direction; parameter i29[i49] controls how SMCC will decode the signal. In either case, you should ensure that the counter always counts up: counting down causes a negative time increment that cause the algorithm to blow up. Parameters i27 and/or i47 control the conversion of the input frequency to a % value.

This mode is very useful for synchronizing the moves of a servo system to some external speed, usually the speed of some material flow. An encoder is simply geared to the flow, and its signal is fed to SMCC(s) -- multiple cards stay synchronized. It also can be used to control the speed of execution of multiple cards synchronously off a single potentiometer by passing the pot output through a voltage-to-frequency (V-to-F) converter and feeding the resulting frequency into the handwheel inputs of all the cards; since this is fundamentally a counting operation, all the cards can perform this identically.

In most master-slave synchronization applications, it is not necessary to think in terms of actual % values; one instead concentrates on the proper i27 [i47] values to keep the system properly synchronized. The procedure for setting these values is quite straightforward. (The procedure, as explained below, assumes that i05 = 487. If i05 is set to another value, i27[i47] scale with i05+1.) First, select an input count frequency that you are sure will never be exceeded in your application. It is best to pick this value so that when it is expressed in counts/bin. msec., it is an integer for the Parabolic SMCC, and a power of two for the Standard SMCC. For instance, if your maximum input frequency is 50,000 cts/sec, you could select a

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

value of 51,200 cts/sec (50 cts/bin msec) for the Parabolic SMCC, or 65,536 cts/sec (64 cts/bin msec) for the Standard SMCC.

The equation for selecting the conversion parameters for Parabolic SMCC is

$$\frac{\text{Max. input freq. (cts/bin msec)}}{32768} = \frac{i27}{i47}. \qquad (3)$$

If our maximum input frequency is 50 cts/bin msec, we could use i27 = 50 and i47 = 32768, or i27 = 25 and i47 = 16384. It is the ratio between the parameters that is important. The legal range of i27 and i47 is 0 to 65535.

On the Standard SMCC, the conversion equation is

$$\frac{\text{Max. input freq. (cts/bin msec)}}{8 * 32768} = \frac{1}{i27} \qquad (4)$$

If we have set our maximum input frequency at 64 cts/bin msec, we get an i27 value of 4096. With only one conversion parameter, the Standard SMCC has less flexibility in setting conversion factors.

Once these conversion factors have been determined, the user simply writes his program as if the system were always running at this maximum frequency (running in "real time"). For instance, if at this maximum input frequency, the slave system needed to do a back and forth move once per second to stay properly synchronized, we could write a program such as

```
T512            ; 1/2 second in "real time"
FOR 0           ; infinite loop
X10000          ; forward move
X0              ; backward move
NEXT            ; end of loop
```

This program would then stay properly synchronized to the input frequency at all frequencies lower than this maximum. (At this maximum, the % value would average to be 65498, thus running in "real time" with time units in binary milliseconds.)

For smooth operation, the input frequency should be several tens of thousands of counts per second. What the card actually does in this mode is to take the number of counts that have occurred during the last servo cycle (usually about a millisecond), and scale this value by i27 (and i47 in the Parabolic SMCC) to yield a % value. The time unit for the next servo cycle, which is proportional to the commanded position increment, is scaled to this % value. For example, if the input frequency is between 3 KHz and 4 KHz, on some servo cycles 3 counts will have passed; on others 4 will have passed (even at constant frequency). There will be a 33% difference in the position increment between the 3-

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

count cycles and the 4-count cycles. However, if your input frequency is between 50 and 51 KHz, there will only be a 2% difference between the 50-count cycles and the 51-count cycles. The higher the input frequency, the smoother the operation -- maximum input count frequency is 2 MHz.

***Problem #2: You wish to synchronize an operation on a Parabolic SMCC to a linear material flow. The material can potentially move at up to 150 in/sec. As the material moves, it turns an 8 inch circumference wheel attached directly to a 1000 pulse per rev. encoder. You wish to complete an operation cycle for every 75 inches of material that passes. How should you set i27, i47, and the time for an operation cycle?

Changing Servo Cycle Time

The vast majority of SMCC users can operate satisfactorily at the default servo cycle time of 976 usec (i05 = 487). There are two basic reasons why someone may wish to change this. The first is in a system where this cycle time is too long for adequate stability or smoothness of the servo loop closure. This usually occurs in systems that have very fine-resolution feedback devices, such as micron-pitch linear encoders, or laser interferometers. In this case, the user would reduce i05 to try to get the desired performance.

Reduction of i05 must be done with care. The more it is reduced, the more often background tasks are interrupted to perform the servo cycle calculations, and the higher the percentage of time (servo duty cycle) spent on these calculations. The background tasks that get interrupted include trajectory planning, status monitoring, PLC programs, and communications, all of which take longer as i05 is reduced. It is very possible to take i05 low enough so that near 100% duty cycle is reached, and no communications with the card is possible, even to increase i05! In this case, the card must be reset, restoring the default value of i05, in order to restore communications. It is not possible to store new values of i05 in permanent memory, because that would make it possible to get into a no-communications state that even a reset would not fix!

If only one axis is being controlled, i15 should be set to 1 so that no Y-axis servo calculations are performed. All background tasks should be kept to a minimum: a low communications rate, a small PLC, and longer-time moves (if the card cannot complete the planning calculations for a move -- done in background -- before the move starts -- the card will cease program execution and lock up). The exact limits are very application dependent: the user must experiment in his own environment. With light background load, it may be possible to get down to i05 = 250 for a single-axis application, or to i05 = 400 in a dual-axis situation.

The other common reason to change servo cycle time is to provide more time for background tasks (for instance, very heavy communications and/or a huge PLC program). In this case, i05

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

should be increased. The limit to the amount of increase will be determined by the maximum cycle time that gives satisfactory servo performance, which is system dependent.

With a different i05, the user must also change i07 to keep the desired time units. Equation (2) still governs the time units. To keep the time units as binary milliseconds, i07 changes in proportion to i05 + 1. This is possible if i05 is reduced from 487, but not if it is increased, because of the limited upper range of i07. At higher values of i05, it will be necessary to define a different basic time unit: say, 1/512 second.

In addition, the servo filter parameters will need to be changed. In general, as i05 decreases, i20[i40] (proportional gain) can be increased slightly, i21[i41] (derivative gain) and i22[i42] (feedforward gain) increase inversely proportional to i05, i30[i50] (acceleration feedforward gain) increases inversely proportional to the square of i05, and i23[i43] (integral gain) decreases in proportion to i05.

***Problem #3: You are running a (single-axis) SMCC system that requires a very tight servo loop, and you have had to reduce i05 to 300 to get adequate stability. What should your i07 value be to keep your time units in binary milliseconds?

Solutions

***Solution #1: Rearranging Equation 2, and solving for the % value (and so for i07), we get i07 = % value = 65536 * 2 * (i05+1) / dT = 65536 * 976 / 1000 = 63963.

***Solution #2: First we calculate the maximum input count rate:

$$\frac{150\ \text{in}}{\text{sec}} * \frac{\text{rev}}{8\ \text{in}} * \frac{1000\ \text{pulses}}{\text{rev}} * \frac{4\ \text{cts}}{\text{pulse}} = \frac{75000\ \text{cts}}{\text{sec}}.$$

We use 4x decode because we want as high an input frequency as possible. The maximum rate is equivalent to 73.23 cts/bin msec. Since we need an integer, we will use 75 cts/bin msec (= 76800 cts/sec). From Equation 4, 75 / 32768 = i27 / i47; so we set i27 = 75, and i47 = 32768.

Since we rounded up our maximum input frequency from 75000 to 76800 cts/sec to make the math work, we must adjust our maximum (real time) material speed to match: 150 in/sec * 76800 / 75000 = 153.6 in/sec. We wish to know to the time taken to move 75 inches at this speed:

$$\frac{75\ \text{in}}{153.6\ \text{in}} * \frac{\text{sec}}{\text{sec}} * \frac{1024\ \text{bin msec}}{\text{sec}} = 500\ \text{bin msec}.$$

We would write our program so that the total time (probably using T times) for an operation cycle is 500 bin msec.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

***Solution #3: We can use a rearranged Equation 2 to solve this:

$$i07 = \% \text{ value} = \frac{65536 * 2 * (301)}{1000000 / 1024} = 40340.$$

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

USING SMCC'S DATA GATHERING FOR INTERMITTENT PROBLEMS OR SIMULTANEOUS MULTIPLE-CARD GATHERING

Sometimes in an SMCC application, there is an intermittent problem that it would be desirable to catch with the data gathering problem without personally supervising every machine cycle. If the problem is such that SMCC will automatically stop running the program, or it triggers some status (e.g. following error warning limit) that the PLC can notice and stop the program, you can catch the problem with data gathering.

This is done by gathering data on every cycle through the problem area. Of course, each time data is gathered, it overwrites the last cycle's data, but if you can make the problem time the last time, you will have the problem data.

A similar technique works for gathering data on multiple cards simultaneously, which the standard procedure on the PC is not able to do directly.

Repeated Data Gathering

A simple example of repeated data gathering is shown in the SMCC program HOLDPLOT.SMC. Each time through the FOR-NEXT loop, M21 is set. This is a bit in otherwise unused memory whose sole purpose is to get the PLC program to issue the 'y' command that sets up data gathering. This command is normally sent from the host program, such as the SMCC Terminal Program, but to do it repetitively, we must do it from inside the card.

The first argument of the 'y' command (which is 4 in this example) sets the period of the data gathering. Technically, it is the log base 2 of the gathering period in servo cycles -- 0 means every cycle, 1 every second cycle, 2 every 4th, and so on up to 7 for every 128th cycle. Here we are gathering every 4th cycle. The second argument is the number of bytes to gather each time. 4 bytes gathers X position only, 8 bytes gets X and Y position, 12 bytes gets X and Y position plus X following error, and 16 gets position and following error for both X and Y axes. In this example we want X position and following error, so we use 12.

There is a DWELL command between the setting of the control bit and the setting of the data gathering bit. This is to allow the PLC time to act. When the data gathering bit is set (M20 here), SMCC starts gathering data and storing it in empty buffer space starting after the program storage, overwriting what was there befcre.

Look at the dialog terminal screen for the running of the program. After the program is started with an "R", every time the "y" command is given from the PLC, SMCC tells the host what the starting storage address is. This is a two-byte hexadecimal number, least significant byte first. In our example, it comes

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

out as "b4a4". Reversing the bytes, we see that storage starts at address a4b4.

When the SMCC program stops running (here because it has completed all four loops, but it could be because of a problem), the user should type "y" into the terminal program. SMCC will return the address of the end of the gathered data (in this example it returns "34ac"). The bytes of this address must be flipped (here it gives ac34). From this subtract the beginning location to get the length (ac34(hex) - a4b4(hex) = 780(hex) = 1920(decimal)).

Next, turn on the disk storage option with ALT-F6. Now bring the data back to the computer and store on disk with:

d {start(hex)},{length(dec)}

In our example, the command is: da4b4,1920

SMCC will respond by sending all of its data back. It will scroll through the screen and get stored on disk. Now hit the ESCAPE key and answer 'y' to the question to exit the Terminal program.

In response to the DOS prompt type:

COPY TERMDATA.SMC HOLDPLOT.PLT

(The second file name can be whatever you want it to be, but the first must be TERMDATA.SMC.) Now reenter the SMCC Terminal program, go into the Editor with F10, and load HOLDPLOT.PLT by hitting F1, 1, and typing HOLDPLOT.PLT for the file name to be loaded. Now we must add the file header info so the plotting routines can make sense of the raw data. Hit Alt-I four times to add four blank lines to the top of the file. In the first line, type the second argument of the 'y' command the PLC gave (in this case, 12). In the second line, type the first 'y' argument (here, 4). In the third line, type the value of i05 (usually 487). Make sure you add an empty fourth line. Now re-save the file with F1, 2 <ENTER>.

Finally, hit ESCAPE to get out of the Editor, and ALT-F10 to get into the gather and plot routines. Hit F3 to load the file into the plotting buffers, specifying HOLDPLOT.PLT as the name of the file. Now you are ready to plot, print, or display the data just as if it had come directly from the card.

Simultaneous Data Gathering In Multiple Cards

A similar technique can be used to gather data in multiple cards simultaneously and bring it back to the computer. The Data Gathering and Plotting Section of the SMCC Terminal Program does not allow this directly, so we must use the terminal mode to do this manually.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The cards should each have a motion program in them that sets and resets the data gathering bit simultaneously. Before running these programs, give each card individually the 'y' command to set up the data gathering -- something like the following:

```
A0y4,12

A1y4,12
```

To each line, the addressed card will send back the starting storage address, least signficant byte first. Write these values down. Now run the programs on all the cards simultaneously, with an 'AAR'. When the programs have finished, find out where the end of the storage on each card is with 'A0y', 'A1y', and so on. Calculate the length of each storage buffer as shown above.

Now turn on the disk storage option with the ALT-F6 key. Next bring back the captured data from the first card with:

```
A0d {start(hex)}, {length(dec)}
```

When the transfer to computer and disk is complete, exit the program. Change the name of the file as done above, something like:

```
COPY TERMDATA.SMC HOLDPLT0.PLT
```

Re-enter the SMCC program, turn on the disk storage option, and bring back the data from the second card with:

```
A1 {start(hex)}, {length(dec)}
```

Exit the program and copy theinformation in TERMDATA.SMC to a different file name. Repeat these steps for as many cards as you had storing data.

When this is done, re-enter the SMCC program and call up the program editor. Add the header information to each file as shown in the previous section, and re-store the modified files. You will now be able to plot, display, or print the data from any of these files.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC'S TORQUE-LIMIT MODE

An SMCC motion controller with the Parabolic software option has a special torque-limiting mode that has several important uses. This mode is entered by using the TORQUE command (short form: TOR). TORQUE is a buffered program command, so it will be used in either of two ways. If i02=0 (not MDI mode) and there is an open motion program buffer, this command will be entered into the buffer and executed at the proper point in the program flow when the program is run. If i02=0 and the motion program buffer is not open, the TORQUE command will not be accepted and SMCC will report an error. If i02=1 (MDI mode), the TORQUE command is taken immediately and the card is put in torque-limiting mode coincident with the start of execution of the next programmed or homing servo command (but not a jog -- to do a torque-limited jog in MDI mode, first command a DLY 20, for instance, to enter torque-limited mode).

The TORQUE command works by preparing the torque-limit mode bit to be set at the beginning of the next programmed servo command. This bit is at SMCC memory address **febf**, bit 5. To do torque-limited jogs, it may be easier to set this bit directly, which allows the user not to have to enter MDI mode and not to have to give a programmed servo command before the jog to enter torque mode. For example, once an M-function has been assigned to this bit by "C25=febf" and "M25=C25.5", then a torque-limited jog can be performed by the on-line commands "M25=1" and "J". Having this M-function declared also facilitates status checking and exiting torque-limit mode (see Torque-Limited General Moves, below).

Torque-Limited Homing Moves

The first use of torque-limiting mode is with a homing move. Normally, a homing move looks for an external signal or combination of signals (home switch and/or C-channel) to find the home reference position. However, in torque-limiting mode, it looks for the condition where the magnitude of the following error has exceeded the warning following error limit (i10). It takes the actual position when this condition occurs as the home reference signal. This allows homing into a hard stop. The following error will be small as the axis is running freely, because the actual position is following the commanded position quite closely. When it hits the stop, the commanded position continues to increment, but the actual position stops changing. This causes the following error to build up until the i10 limit is exceeded. The actual position at this time is the home reference position (the zero position after home can be different by use of the i25[i45] parameter -- home offset). Tripping the following error limit automatically takes the card out of torque-limiting mode.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Torque-Limited "Go-Until" Moves

The second use of torque-limiting mode is with a "Go Until" move (e.g. X100000^50). When not in torque-limiting mode, this type of move is looking for an external trigger -- usually from a registration mark -- just like a homing move. In torque-limiting mode, this is looking for the i10 following error limit to be exceeded -- just like a homing move in torque-limiting mode. The actual position of the axis at the time when i10 is exceeded is the reference position to which the value after the ^ symbol is relative (in our example, the axis will try to control to 50 counts past this position).

This type of move allows the axis to reach a position relative to an unknown hard stop. If the relative value is past the hard stop, the axis will push against the stop with a steady torque. (In general, there should be no integral gain when doing this, or the command output will quickly saturate.) If the relative value is not past the hard stop, the axis will hit the stop, then back off from it. Tripping the i10 following error limit automatically takes the card out of torque-limiting mode.

Torque-Limited General Moves

The last use of torque-limiting mode works with general moves, programmed and jog. In these cases, if the normal increment of commanded position according to the desired trajectory causes the i10 following error to be exceeded, the increment is reduced so that the following error is only equal in magnitude to i10. This slows down or even stops the move if excess resistance is found, and also has the effect of limiting the torque that the axis will use. This type of move is very useful in pulling material, such as a web or strand, that has limited strength; or in dynamometer testing. In this use, the card stays in torque-limiting mode indefinitely, and it must be manually removed from the mode. The torque-limit mode bit is bit 5 of SMCC memory location **febf** (hex). This bit should be set to zero to take SMCC out of torque-limiting mode (assign an M-function to the bit, and set it to zero at the desired time, either on-line, in the motion program, or in the PLC).

What Determines the Torque Limit?

In a torque-limited move, the SMCC output (which is directly related to torque in a current-loop amplifier) is actually limited by one of two factors. First, there is an output limit directly defined by i32[i52]. This limit is always active, even in non-torque-limited moves. Second, the following error limit, i10, can limit the output because it limits the size of the error that enters the PID loop (it really limits the output only if the integral gain is zero, which it always should be in torque-limiting mode).

Which of these limits controls the output first depends on the actual parameter values. The command output value, in bits, is

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
Out(n) = (i20/2048) * [FE(n) - i21*AV(n) + i22*DV(n) +
         (i30/4)*DA(n) + (i23/4096)*IE(n)]
```

where
- i20 is the proportional gain (i40 for Y)
- FE(n) is the following error for cycle n
- i21 is the differential gain (i41 for Y)
- AV(n) is the actual velocity for cycle n
- i22 is the velocity feedforward gain (i42 for Y)
- DV(n) is the desired velocity for cycle n
- i30 is the acceleration feedforward gain (i50 for Y)
- DA(n) is the desired acceleration for cycle n
- i23 is the integral gain (i43 for Y)
- IE(n) is the integral of the following error at cycle n

In most torque-limit cases this relationship can be significantly simplified. First, the integral gain can be zero. Second, the feedforward gains can often be ignored, either because they are zero, or because the desired velocity and acceleration are low. In the case of running into a stop, the actual velocity is zero and the equation reduces to:

```
Out(n) = (i20/2048) * FE(n)
```

You can substitute i32 for Out(n), and see what the maximum following error that increases output is. Then you can substitute i10 for FE(n) and see what the maximum output that can be produced is. Whichever has the lower level is the controlling factor. In these types of moves, it is often desired that i10 limit the output, because higher outputs may be desired when not against a stop, and it is nice not to have to change i32 depending on the situation.

The case of a running torque-limit move is basically the same for a current-loop amplifier when i21 and i22 are equal (this is usually the case on these systems), because the actual and desired velocities almost cancel. Usually in these moves, the user will use i32 to limit the torque directly. The above equation can be used just to make sure that i10 does not limit the output to a lower value. (What often happens here is that higher load torque causes the PID loop to ask for a higher output than i32; i32 limits the output, which causes the axis to fall behind until the i10 limit is exceeded, which reduces the requested velocity).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

USING THE SMCC DUMP UTILITY

The SMCC "Dump" utility program for the IBM PC and compatible computers is designed to allow the SMCC to continuously run programs bigger than its program buffer. Dump takes advantage of the dynamic and recirculating nature of the SMCC program buffer by overwriting already executed program lines. By repeatedly polling the status of the buffer, it makes sure it keeps feeding new commands in advance of when they are needed, but does not try to overwrite any commands not yet executed.

Using the Dump program is very easy. The program to be dumped must be in an ASCII disk file on the PC. The file could have been created by the SMCC Terminal Program Editor, by the SMCCAD conversion program, by a word processing program that can create ASCII files, or by a custom program.

In response to the DOS prompt, simply type **dump file.ext**, where "file.ext" is the name of the program file to be dumped (these usually will have the .SMC extension). No further user action is needed. The program will immediately start to load the first block of program lines and command the card to start running the program. Then it will monitor the buffer until a good portion of the block of program lines has been executed, at which point it will load the next block. This process is repeated until the entire program has been loaded. Dump reports to the PC screen as it loads each block and as it waits until the next block. An explanation of how Dump does this is contained in the applications note "Using SMCC's Program Buffer in Recirculating and Other Modes".

The program being dumped should not have any branching or loops in the main body of the program. If any of this is required, it should be put in subroutines and placed at the top of the file. Dump automatically adds the line "A0z1000" to the top of the program file as it sends it to the SMCC(s). The timing of sending subsequent blocks is dependent on having the main program buffer near 1000 lines long. Remember that any subroutines and PLC programs must reside in the remaining memory -- 790 lines with the standard memory. Programs can be dumped to more than one card (if daisy-chained together), but the user must include the program lines to clear the other cards (e.g. A1z1000) and control all of the card addressing in the file.

The Dump program will communicate over the COM1 port at 9600 baud and a x1 clock speed multiplier unless there is a file SMCC.DAT in the default directory. This file is created by an SMCC Terminal program Version 3.5 or older when the communications configuration is changed and saved. If you wish to use another communications configuration, you must set it up using one of these older versions of the SMCC Terminal program. Dump can communicate over the COM1 or COM2 RS-232 interfaces at various baud rates and clock speed multipliers (which determine how long to wait for the interface to be ready again), or over Delta Tau's bidirectional parallel interface at various PC bus addresses.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

RE-INITIALIZING SMCC

There may come a time when a user must essentially "empty out" SMCC and start over again. The most likely thing to cause this condition is an errant PLC program saved in permanent memory that dominates the communications channel so much that the user cannot disable or erase the program. Typically a PLC program that triggers a message to the host (SEND "___") or a command for the SMCC ("___") repeatedly (i.e. the condition that caused this action was not immediately set false) will virtually lock up communications with the host. (It is good practice not to save a PLC program in permanent memory until it has been thoroughly debugged. A problem PLC program that is only in active memory can be eliminated by a hardware reset or by cycling power.) This note explains the procedure for re-initializing SMCC, which can eliminate the existing programs, including the offending PLC program.

This re-initialization procedure may also be used to set the I-parameters to factory default values and/or to check the operation of the inputs to SMCC. In short, when things are thoroughly messed up and nothing else seems to work, this is a good procedure to use, in order essentially just to start over.

The procedure involves hardware lines on the 26-pin JPAN connector (J7 on the standalone SMCC, J1 on the SMCC-PC SMCC-VME and SMCC-STD), particularly the INIT/ line (Pin 15 -- the button into this line is usually called RESET) and the STOP/ line (Pin 10 -- the switch into this line is usually called ABORT). Delta Tau's ACC-16 control panel is ideally suited for this task; however, it is not difficult to rig low-true switches to the connector if there is no control panel. To perform this procedure, there should be either a 2 x 40 character LCD display attached to the JDISP connector, or a terminal emulator program such as the SMCC Terminal Program on a host PC ready to talk to the card. SMCC messages are sent to both the display and the communications port during the initialization process.

Starting the Procedure

Start the procedure by holding the STOP/ (Abort) line low, bringing the INIT/ (Reset) line low and releasing it. SMCC will enter a special mode with its outputs disabled (for the duration of the initiailization procedure). It will briefly display the PROM version and date with a message like this:

```
DELTA TAU DATA SYSTEMS
11/15/89 VERSION 2.80P
```

Then SMCC will prompt with this message:

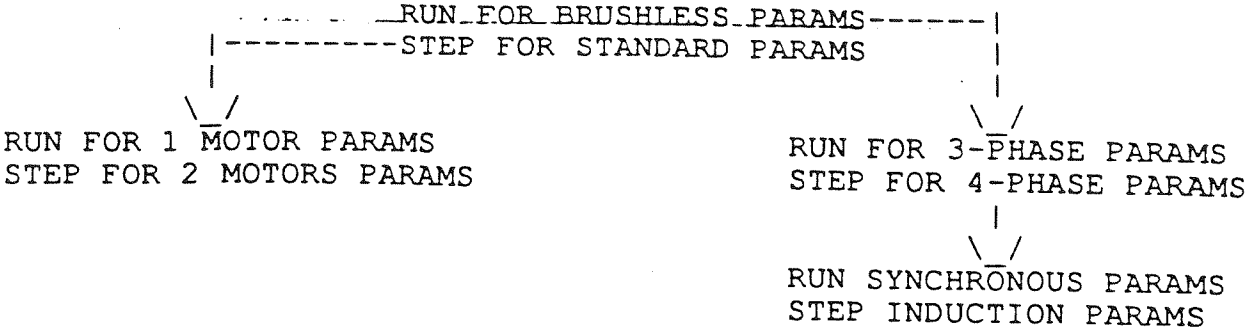
```
RUN FOR PRESENT PARAMS
STEP FOR DEFAULT PARAMS
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Selecting Re-initialization or Not

If you wish to keep the I-parameters and the programs that you currently have saved in permanent memory, take the STRT/ (Run -- Pin 8) line low and release it. If you wish to set up the card with the factory default parameters and no programs, take the STEP/ (Step -- Pin 9) line low and release it.

If you have selected the default parameters, you will be presented with several more choices, selectable in the same manner as above:



(When you are done with this re-initialization of the SMCC, you will probably want to change many of these default values. Hopefully, you have already stored your values in a file on disk in the computer so they can be downloaded easily. You may want to make sure your motors cannot be enabled after the initialization if you are taking the default parameters, because the default i39 and i59 values may set you up for runaway conditions.)

Switch Test

After you have set the I-parameters, you enter the "switch test" portion of initialization, which allows you to check the functionality of the hardware inputs to SMCC. You will be prompted with a line telling you which input is expected. You advance to the next stage either by taking that input low (the limits are to be taken high) or by taking the STOP/ line low, which circumvents that test. You will see the following prompts:

SWITCH TEST	RUN	(STRT/ -- Pin 8)
SWITCH TEST	STEP	(STEP/ -- Pin 9)
SWITCH TEST	HOLD	(HOLD/ -- Pin 12)
SWITCH TEST	HOME	(HOME/ -- Pin 11)
SWITCH TEST	EQUAL	(PREJ/ -- Pin 7)
SWITCH TEST	JOG +X	(XJPL/ -- Pin 6)
SWITCH TEST	JOG -X	(XJMI/ -- Pin 4)
SWITCH TEST	JOG +Y	(YJPL/ -- Pin 5)
SWITCH TEST	JOG -Y	(YJMI/ -- Pin 3)
SWITCH TEST	LIMIT +X	(XPLL on JMACH)
SWITCH TEST	LIMIT -X	(XMIL on JMACH)
SWITCH TEST	LIMIT +Y	(YPLL on JMACH)
SWITCH TEST	LIMIT -Y	(YMIL on JMACH)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Once this part of the test is finished, you will be able to test the encoder and handwheel inputs. You will be shown a display looking something like this:

```
 ENC-X ENC-Y  HW-X  HW-Y
     0     1     3 65535
```

which displays the least significant sixteen bits of the counter registers for each input. The motors are disabled, so they may be turned by hand at will and the results observed on the display. Likewise, any handwheel encoders may be turned and the results observed. When you are done with this portion of the test, take the STOP/ line low momentarily, and the SMCC will commence normal operation.

After the Procedure

At this point, you may have changed the parameters and programs in active memory (RAM), but the non-volatile memory (EAROM) has not been changed. If you want to change the EAROM, you must "save" the contents of RAM into EAROM with the "s" command. First you will probably want to re-load certain parameter values and programs (although not any offending program segments, of course).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

CONSTRUCTIVE USE OF EXTRA SMCC MEMORY SPACE

Some sophisticated users of SMCC may find themselves limited in number of variables, and want to effectively extend the number using empty memory space in the SMCC. This application note is a guide to this technique.

Where to Look for Space

The first place to look is the space explicitly reserved for this purpose. In the Standard SMCC, this is 36 bytes starting at address a200 (a200 -- a223); in the Parabolic SMCC, this is 12 bytes starting at a3a7 (a3a7 -- a3b2). This space is always free.

The second place to look is in the program buffer. This can have two areas of unused memory: first, between the end of the main motion program and the beginning of the subroutines; and second, between the end of the subroutines and the end of the PLC program, which starts at the other end of the buffer space.

This memory must be used with more care. First expanding the motion program, subroutines, or PLC program could overwrite the memory you are using. Second, in the Parabolic SMCC, the data gathering routine uses the area between subroutines and PLC to hold the sampled data before it is sent to the computer.

The motion program memory starts at address a400 in the Standard SMCC, and at a414 in the Parabolic SMCC. There are four pointer variables that tell you where the unused memory is past this address. STPTR (storage pointer) holds the address where the next line of the main motion program would go. STEND (storage end pointer) holds the address of the end of the space reserved for the main program; it is the address of the first line of the subroutines. The address held by STEND is set by the most recent 'zn' command, which reserves 'n' lines (4 bytes per line) for the main program (e.g. 'z40' reserves 160 bytes (a0 hex) before the subroutines, so STEND would be at a414 + a0 = a4b4 on the Parabolic SMCC). LBPTR (subroutine library pointer) holds the address where the next subroutine line would go. Finally, BGPTR (background PLC pointer) holds the address where the next PLC program line would go.

Thus, the two spaces for possible use are 1) in between the addresses held by STPTR and STEND, and 2) in between the addresses held by LBPTR and BGPTR (remember, the PLC program extends beyond the BGPTR location to the end of the buffer).

All of the pointers are two-byte values (to hold a sixteen-bit address), with the least significant byte of the address held in the least significant byte of the pointer. The addresses of the pointers themsleves are (their contents hold the addresses of these key buffer points):

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Pointer Name	Standard SMCC Address	Parabolic SMCC Address
STPTR	a2fa,a2fb	a3b5,a3b6
STEND	a2fe,a2ff	a27c,a27d
LBPTR	a2fc,a2fd	a27e,a27f
BGPTR	a2f6,a2f7	a280,a281

These addresses can be read using the 'r' on-line command. Two bytes can be read at once with this command. For instance, 'ra27c,2' on the Parabolic SMCC will return the address of the beginning of the subroutines. However, it will do so LSB first; so if it returns 'b4a4', the address is really 'a4b4'.

What region of this usable space you use is dependent on your application. If you plan to do data gathering, use the area in between STPTR and STEND. If you plan to expand one of the three program areas, stay away from that end of the empty space.

How to Use This Space

The first and most common use of this space is for user flags for control logic. To do this, simply define a C-function to an unused byte (e.g. C21=a4b4), and then up to eight M-function to bits of this C-function (e.g. M21=C21.0, M22=C21.1). These bits can then be set and cleared at will in the motion and/or PLC programs.

The second use is for temporary storage of P-variable values so that the P-variables themselves may be freed up for other use. P-variables are 4-byte values, so the C-functions to do this should be defined four bytes apart (e.g. C22=a4b8, C23=a4bc, C24=a4c0). The only place the values of C and P variables can be exchanged is the the PLC program. Typically, a flag bit is set in the motion program to trigger this branch of the PLC. It is imperative to allow time for the PLC program to make the switch, which means making sure that the PLC program can go through a full cycle before the motion program uses the new values. In the Standard SMCC, there should be two servo moves or dwells in between the time that this bit is set and the time that the new P-variables are to be used, because the calculations will start immediately after the beginning of the second move. In the Parabolic SMCC, only one move in between is needed if the move time is enough longer than the calculation time (i12) to allow the PLC to cycle fully before the calculations start. An example of how this might work is shown below:

```
     Motion Program Segment
SET21                ; This is set at start of decel or dwell
DWELL20              ; No calcs done during this move
DWELL20              ; Calcs below start at the beginning of this
P4=(P1*P2/P3)        ; This will use the new P values
   ...
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PLC Program Segment

```
IF M21=1
THEN C22,4=(P1)      ; Store P values in C-functions
AND C23,4=(P2)
AND C24,4=(P3)
AND P1=C25,4         ; Put previously stored values back in P
AND P2=C26,4
AND P3=C27,4
AND M21=0            ; So this is only done once at a time
```

There would be a similar routine to make the switch the other way. This technique could, of course, be extended to handle a lot more storage locations. It is even possible to keep re-defining C-function locations during the execution of the program.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SOFTWARE UPDATES FOR SMCC "PARABOLIC"
AVAILABLE ON ROM VERSIONS
(11/7/89 V2.78 P & V), (11/14/89 V2.79 P & V), AND GREATER

Several updates (corrections and improvements) have been made to the parabolic version of SMCC as follows:

Changes for SMCC V2.78 P and V and Higher

1) A new buffer command, "tt," was added which controls the Go Until, "^" command's calculation time independently of i12.

2) To eliminate the initial "i12" calculation delay time before a move is started, a new buffer command, "WUT" is provided.

3) A new backlash take-up mode is provided where the backlash is taken up for any operating mode including jog and hand-wheel. Parameter i80 specifies the slew rate.

4) For 180° (half circle) circular interpolation, where the specified radius is too small, an error band is provided for rejection of the move, parameter i81 specifies the tolerance band.

5) The MDI mode has been improved to provide better handshake timing and prevent lost data in transmission.

6) An automatic reset cycle was added for the ACC-14 and OPT-7 piggyback boards.

7) Tests were run on a six-pole motor with two-pole resolver versus six-pole motor with six-pole resolver, and it is established that PLC and background calculation time consumption is increased by a factor of slightly less than two for the two-pole resolver. Effects of using a display unit (ACC-12) was also examined.

Changes for SMCC V2.79P and Higher

8) Allows the 'q' command to be followed by any other command on the same line. This is to be used in the MDI mode to alter destination without stopping. Example: "X10000 Y10000" followed by "qX5000 Y5000" during the first move will break into that move and towards the second destination.

Each of the above is now described in detail:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1) "tt" COMMAND

In normal operation, the SMCC uses an i12 time of 20 to 50 msec. or more, depending upon the application. The go until command "^" currently uses i12 time to calculate its new destination after the external trigger is received. The "tt" command is specifically provided for setting the calculation time of the "^" command, and is used as a "buffer" command by the SMCC.

Upon power turn-on or reset, the value of i12 is copied into "tt," subsequently the program can set the value of "tt" to that appropriate in the application. For example, "tt25" sets the "^" command calculation time to 25 binary msec. or (25 x 1000)/1024 msec. If "tt" is not used, its value will remain that of i12.

The minimum allowable value for "tt" is 10 msec. A value of 20 msec. is recommended for general purpose use. Please note that if heavy use of background time is made, either because of multiple PLC statements or other reasons, the "tt" time may have to be extended.

CAUTION - Erratic motion may result if too small a value is used for "tt."

2) "WUT" COMMAND

The SMCC, in its normal operating mode, requires i12 time after it receives an "S" (step) or "R" (run) command before a motion command is sent to the amplifier. This is undesirable in some applications. The "WUT" command is specifically designed to provide a means to overcome this problem and make it possible to start motion nearly instantaneously by transporting the required calculation time to a more convenient time where motion does not have to start.

"WUT" is a buffer command and can be used between buffer instructions, but only when velocity is known to be zero. The reason for this restriction is that "WUT" actually causes the % feedrate to be instantly set to zero. As a result, the following move sequence, executed with step command "S":

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
:
.           /"S" (step) command is issued
X5000       /Move starts but ends abruptly because
WUT         /WUT is executed at the beginning of decel.
X10000      /This move is calculated but not executed.
:
.
```

will cause a very abrupt stop during the X5000 move since the end of the move is actually the beginning of the deceleration point, where velocity is still high.

The "WUT" should therefore be used only when there is no motion (velocity is zero). For example:

```
:
.           /"S" (step) command is issued
X5000       /Move completes then "S" issued again
DWE 20      /Dwell is completed, "S" issued again
            /Use a dwell of at least 20 to provide
            /enough calculation time.
WUT         /% is set to zero, X10000 move is calculated,
            /but not executed.
X10000      /X10000 move can be started by issuing a trigger
            /pulse at the SMCC's Y user flag input.  Motion will
            /start within one to two msec. after trigger is
            /received.
:
.
```

This sequence would work correctly, and not cause any unwanted jerks since the DWE 20 (DWELL) forces the X5000 move to be completed and then allows the "WUT" to be executed. Note that if instead of "S" (step) commands, an "R" (run) command was used at the start of the sequence, then X5000, DWELL 20, and the WUT commands would execute automatically and the X10000 move would be precalculated and ready for execution using the Y user flag input, or if preferred, by sending a "CNTRL R" command via RS232, the parallel port, or the "bus" as the case may be, since the "CNTRL R" command automatically sets the value of the % command to 65498 (the value of i07) and thus causes the move to occur at full speed.

To summarize the "WUT" command's function:

1) It is a buffer command and should be used at a point in the program where velocity is zero to avoid instantaneous velocity change. Its function is to calculate the succeeding move and passing the results to the servo routine thus making it ready to make a move, while at the same time automatically setting the value of "%" to zero and preventing the motion from actually occurring.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2) To activate the "WUT" command (cause it to precalculate the next move), an "S" (step) or "R" (run) command is required, either via communication or control panel input.

3) To cause the activated "WUT" command's move to actually occur, use a hardware input trigger at the Y user flag input (J MACH Pin A10, use i26 to configure trigger polarity, etc.) which causes the % value to be set to i07 (default 65498) instantly since the YUFL/ input causes an interrupt at the CPU. Motion occurs within one to two msec. of trigger reception.

Instead of the YUFL/ trigger, a "CNTRL R" command may be used to get motion started.

It is also possible to abort the move by using "q" (abort) or "Q" (quit) commands.

3) BACKLASH TAKE-UP

Currently the SMCC provides backlash control by using two parameters, i34 for the X axis and i54 for the Y axis, which specify the amount and direction of backlash motion (extra motion) to be made when the direction of motion changes anytime during program execution. Backlash is not active when jogging or handwheeling.

A new backlash mode is now provided where the specified amount of backlash is taken up whenever the direction of the servo velocity is changed. This method allows backlash to be active at all times during commanded motions, during jog, and during handwheel operations.

Parameter i80 has been added in order to provide control in choosing the old or new backlash modes and to provide a "slew" control in the rate of backlash take-up. i80 has a range of 0 to 15. A value of zero (i80 0) selects the old backlash mode and is the default value for i80. Values of 1 to 15 activate the new backlash mode and also specify the number of encoder counts per background cycle (about every four msec.) that will be used to take up the backlash specified by i34 [i54]. For example if i34 [i54] are set to 50 and i80 is set to 10, then the 50 counts of backlash will be taken up in approximately 20 msec. after velocity reversal.

In order to avoid oscillations during reversal and for other reasons, in addition to velocity reversal, a minimum of four encoder counts of direction-reversal is also required before backlash will activate.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

i80 cannot be used as I80 (i parameter in the buffer), therefore this parameter cannot be changed by the program itself and must be set up and saved with off-line commands.

4) HALF-CIRCLE INTERPOLATION TOLERANCE

When using the SMCC's "CIR 0,1,2, or 3" command (circular interpolation) with a specified radius and a 180° circular arc, the destination point must be exactly two times the radius or smaller otherwise, even an excess of one count in the destination point would cause the SMCC to declare an error and do a linear move instead of a circular one. Even though a solution for this problem exists in that two 90° arcs can be made instead of one 180° arc, a better solution is provided by specifying an error band tolerance and aborting the move rather than doing a linear move.

A new parameter, i81, with a range of 0 to 15 and a default value of 0 is implemented. The value of i81 is used to provide a tolerance in encoder counts which is checked before the 180° circular move is made.

If the destination is smaller or equal to two times the radius, the move is made as specified. The effect of a smaller destination is that it causes the SMCC to make an arc smaller or larger than 180° depending upon which circle has been defined, CIR 0,1,2, or 3.

If the destination is greater than two times the radius but less or equal to the tolerance set by i81 in encoder counts, one half the excess value is added to the radius and the move is then made. This means that the resulting circle will have a radius larger than that specified in the program, but a perfect circle will be made.

If the destination is greater than two times the radius and greater than the i81 encoder count tolerance allowed, then the move is aborted, no motion occurs, and the error flag is set. In the standalone SMCC, "Error 4" is displayed.

Please note that if engineering unit scale factors are used and RX (radius of X) and RY (radius of Y) are used, i81 will still be compared as encoder counts regardless of scaling.

i81 cannot be used as I81 (i parameter in the buffer), therefore this parameter cannot be changed by the program itself and must be set up and saved with off-line commands.

5) MDI HANDSHAKE TIMING IMPROVEMENT

In Manual Data Input mode (MDI: i02 = 1), the timing of the handshake signal has been improved to allow more rapid block execution. This is important only for those users who need

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

to send rapid sequences of moves (less than 100 msec.) in real time (when MDI mode is used). In MDI mode, the Buffer Full (BF) line is used to provide the Block Request (BLKREQ) signal, indicating when it is valid to send the next move block.

In MDI mode, the host may send one move (block) ahead of the move currently executing. If this next move is received and processed by SMCC in time, it will blend the moves smoothly together. This means that in order to do a blended series of moves, the host must send a move after the previous move has started, but enough before the end of this move that SMCC can calculate how to blend into the new move. "Enough before" means i12 time (the move calculation time) before the time where SMCC would start to decelerate to a stop if no further moves were received.

The BF (BLKREQ) line tells the host whether it is valid to send the new command or not. During a sequence of moves, BLKREQ will be high (BF/ -- BLKREQ/ -- will be low) when it is okay to send the next move. When the card has already been given one move ahead, BLKREQ will be low (BLKREQ/ high), signaling that the card is not ready to accept another move command. If one is sent, the card will consider it an illegal command and ignore it.

When the card is not executing any moves, BLKREQ will be low, even though it will accept a move command. If the host program cannot tell from its internal logic that it is in a state before a sequence of moves has begun, it must look at the In-Position (IPOS) signal as well as BLKREQ. IPOS is high only when the card is not executing a move and it is within the in-position band set by i01 (it is a good idea to set a sufficiently large i01 in this case). It is all right for the host to send a move with BLKREQ low if IPOS is high, subsequently, only a BLKREQ high state needs to be looked at for sending data.

The procedure for sending a series of moves is as follows. The first move will be sent with BLKREQ low and IPOS high. As soon as SMCC recognizes this as a move command (two to three msec.), it will take BLKREQ high. (The host should never try to send commands faster than this -- it could catch SMCC faster than the card could respond to changes in its own status.) As soon as it starts this move (i12 time later), it will take IPOS low, where it will stay as long as moves are being executed. With IPOS low, and BLKREQ high, SMCC is executing a move, but it is okay to send the next move (the host will not need to look at IPOS any more during this continuous sequence of moves).

Now the logic goes in a loop. Having determined it is valid to send the next move, the host sends it. When SMCC recognizes it as move (two to three msec.), it takes BLKREQ low, telling the host not to send any more moves. At this point,

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SMCC is still executing a move and has a second move in the MDI buffer that it is ready to execute. When the present move finishes, SMCC starts the second move and takes BLKREQ high, signaling that it is ready to receive the next move. The host had been polling BLKREQ, waiting for it to go high again before sending the next command.

If the host is sending the commands as ASCII strings, it can (and should) send all of the command except for the carriage return before BLKREQ goes high. When it sees BLKREQ go high, it need only send the carriage return. If the commands are binary (DMA) the host should wait for BLKREQ high to send the whole command.

If the host has not sent the next command in advance of i12 time before SMCC would start decelerating, SMCC takes BLKREQ low, signaling that it is not ready to take a command and blend it to the present move, because there is insufficient calculation time. If this happens the host must wait until the move is completely done, and start up again as a new sequence.

If the host sends data only after BLKREQ goes high, it should wait about four milliseconds (data processing time) after the completion of data transmission before it polls BLKREQ again. This gives the SMCC the time it needs to set BLKREQ properly. Polling too quickly can cause the host to think the SMCC is ready when it really isn't.

6) AUTOMATIC RESET CYCLE FOR ACC-14 AND OPT-7

When the Accessory 14 (ACC-14) or Option 7 (OPT-7) boards are used with SMCC to expand I/O capability, the SMCC currently does not "zero" initialize these automatically and it is therefore necessary to perform the reset operation via the user program or a PLC program.

The initialization procedure of the SMCC has been changed to automatically set all ACC-14 I/O lines as inputs and the Option 7 D/A converter to zero upon power turn-on or reset.

7) SIX-POLE MOTOR OPERATION

Tests were conducted to determine the effect of six-pole motor commutation with a two- or six-pole resolver on background calculation time. The test consisted of writing a single PLC loop statement which toggled an output at maximum rate and measuring the output frequency. The test results are as follows:

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

(i50 = 1) six-pole motor, two-pole resolver -- Output period 7.5 msec.

(i50 = 0) six-pole motor, six-pole resolver -- Output period 4.0 msec.

These results indicate a two to one improvement in PLC operation time and are consistent with the fact that the six-pole motor with two-pole resolver has to perform a division by three (i50 = 1) which requires more time to calculate.

The effect of this is that when using the 6/2 combination, all minimum recommended times for "tt," i12, "t," "T," etc., have to be nearly twice as large since all recommended times are based on a four to five msec. background cycle time.

Also, the effects of using various displays (ACC-12, 12A, 12B) were examined and it was determined that if no display unit is to be used in SMCC, it is best to select i18 = 0 (default value is three) which selects the eight digit LED display ACC-12B. The reason for this is that SMCC performs the display calculations at all times regardless of whether a display is actually used or not, and the ACC-12B has the fastest and shortest display calculation time. The improvement in background time is in the order of three percent shorter when using i18 0 instead of i18 3.

8) ALTERED DESTINATION MOVE IN MDI MODE

Sometimes when commanding moves for immediate execution, it is desired to break into a move before it is finished and give the card a new destination. This can be done by commanding the new move preceded by a lowercase "q." As soon as SMCC receives this new command, it will start calculating the altered trajectory to the new destination. It will begin this altered trajectory i12 time after receiving the command.

Example:

```
i021        ; MDI mode - execute moves immediately
F100
X10000      ; start move toward X = 10000
qX-5000     ; if issued while previous move is still
            ; executing, this will cause SMCC to break
            ; into that move and alter the trajectory
            ; to proceed to this new destination
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1/T Velocity Measurement Info 78 79 170 314 315
Absolute Encoder Base Address (i64[i65]) 89
Absolute Encoder Data & Strobe Polarity (i76[i77]) 93
Absolute Encoder Parameters (Parab SMCC) 42 89
Absolute Encoder Setup 187
Absolute Encoder/Resolver Offset (i70[i71]) 91
Absolute Encoder/Resolver Size (i72[i73]) 91
Absolute Movement Program Commands 118
Absolute Position Sensor Overview 187
AC Induction Motor Mag. Current (i42/i52) 85 87
AC Induction Motor Setup 38
AC Induction Motor Slip Gain (i45/i51) 87
Acceleration Feedforward Gain (i30[i50]) 80
Acceleration Time Parameter (i08) 62
Acceleration Time, Differences in Parabolic 182
Accel/Decel Time Set Program Command (t) 116
Accessories Listing 13
Address Map (See Memory Map, I/O Map)
Address of Card -- Selection 53 317
Address of Highest Card Parameter (i19) 72
Altered Destination Moves 119 420
Amplifier Connection Diagrams 336 337 338 340
Amplifier Connection Information 29 309 311 312
Amplifier/Motor Connection Procedure 26
Analog Control Output Description 8
Analog Input Conditioning Option (#3) Desc. 12 355A
Analog Input Control Mode Parameter (i14) 66
Analog Input Description 8
Analog Input Potentiometer Connections 342A
Analog Input Register Locations 203
Arctangent Function for PLC 131
A/D Converter Mode Parameter (i14) 66
Background PLC (See PLC)
Backlash Compensation Parameters (i34[i54], i80) 81 94 416
Basic Operational Checks 23
Baud Rate Selection 54 317
Beginning Examples 224
Bit Definitions (See M Functions)
Block Diagram 3
Branching Program Commands 112 115
Brushless DC Motor Phase Finding Time (i52) 87
Brushless DC Motor Select Parameter (i13) 65
Brushless DC Motor Setup 36 313
Brushless DC Motor Star/Delta Connection (i48) 86
Buffer Commands (See Motion Buffer, PLC Commands)
Buffer Purge (Motion Program) Command 102 108
Buffer Purge (PLC Program) Command 109 124
Burr-Brown Terminal Use Examples 296 298
Byte Definitions (See C Functions)
C Definition Default Assignments 138 139

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

C Function Definition Commands 103 122
C Function Query/Evaluation Commands 106 113 114 124
C Function Value Setting Commands 127
CAD Conversion Program (SMCCAD -- ACC-17) 17
Calculation Time Parameter (i12) 65
Card Address Select Command 53
Circle Error Tolerance Limit (i81) 94 417
Circle4 Mode Profiling Information 158 364
Circle4 Mode Program Examples 299 386
Circular Movement Commands 120
Circular Movement Examples 230 232 265
Circular Position Range Mode (i78[i79]) 184 94
Circular Position Range Parameter (i37[i57]) 184 82
Command Format Description 53
Command Summary 141
Communication with Host -- Selection 54 317
Communications Connector Descriptions 322 333 351 354 355
Communications Problems 21
Commutation Cycle Size Parameter (i40) 84
Commutation Encoder/Resolver Selection (i50) 86
Commutation Hall Sensor Offset (i45/i54) 85 88 313
Commutation Lookup Table Address (i43) 85
Commutation Lookup Table Param. (i15) 67
Commutation Offset (Hall Effect or Resolver) (i54) 88
Commutation On/Off Parameter (i13) 65
Commutation Phase Advance Parameter (i41) 84
Commutation Phase Finding Output Value (i53) 88
Commutation Phase No. (3/4) Select (i44) 85
Commutation Triple Cycle Definition (i50) 86
Conditional PLC Commands 124
Conditional Program Commands 113
Connecting SMCC to Amplifier/Motor 26
Connecting SMCC to Host Procedure 21
Connecting SMCC to Machine 29
Connector Listings 318
Connector Locations 335
Control Panel 10 15 16
Control Panel Connector Descriptions (J7; JPAN) 326 350
Control Panel Function I/O Locations 199
Control-Character Commands 109
Cubic Fit Profiling Information 158
Current Limit (See Output Limit)
Data Gathering Information 171 399
Data (Memory) Read/Write Commands 102 121
DC Brushless Motor (See Brushless DC Motor)
DC Motor (Non-tach) System Setup 35
Deadband Size Parameter (i60[i61]) 88
Deceleration Times Chart 96
Derivative Gain Constant (i21[i41]) 75
Differential Gain Constant (i21[i41]) 75

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Differential Gain Time Delay Term (i22[i42])				76
Digital Control Output Description				8
DIP Switch Settings		54	55	317
Direct Command Summary				141
Direct Control Commands				95
Direct Motion Control Commands				95
Display Connectors Descriptions (J5, J6)			325	354
Display Panel Status Bit Description				72
Display Position Scale & Format (i38[i58])				82
Display Type Parameter (i18)				71
Display Types (ACC-12)			11	15
Display Velocity Scaling Parameter (i17)				71
DMA Commands			109	178
DMA Communications Information				178
DMA Parabolic Mode Control (i63)				89
DUMP Program Load Utility			365	405
EAROM Storage Option (#2) Description				11
EEPROM Storage Option (#2) Description				11
Electrical Revolution Size Parameter (i40)				84
Electrical Specifications				4
Elliptical Movement Commands				120
Elliptical Movement Examples			231	232
Encoder Connection Information	315 316	319	321	352
Encoder Data & Strobe Polarity (i76[i77])				93
Encoder Jumper Descriptions (E40,E41,E42,E50)		314	315	316
Encoder (Handwheel) Decoding Param. (i29[i49])				79
Encoder (Position) Decoding Param. (i39[i59])				83
Encoder/Resolver Base Address (i64[i65])				89
Encoder/Resolver Select for Commutation (i50)				86
Encoder/Resolver Size (i72[i73])				91
Error/Status Display Description			10	202
Event Response Program Examples		251	253	255
Expanded Program Memory Option (#5)				12
Expansion Connector Descriptions (J8; JEXP)	5	10	327	350
E-Point (Jumper) Descriptions				309
E-Point (Jumper) Locations				317
Feed Hold Slew Control Parameter (i11)				64
Feedback Damping Term (i21[i41])				75
Feedback Equations			169	73
Feedback Gain Multiplication Factor (i30[i50])				79
Feedback Integral Gain (i23[i43])				76
Feedback Integration Mode Parameter (i09)				63
Feedback Proportional Gain (i20[i40])				73
Feedforward Acceleration Gain (i30[i50])				80
Feedforward Velocity Gain (i22[i42])				76
Feedrate Command for Programmed Moves (F)				116
Feedrate Override Control Chart				100
Feedrate Override Default Parameter (i07)			62	393
Feedrate Override Query Command			106	393
Feedrate Override (%) Commands		98	121	393

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Feedrate (Homing) Parameter (i04) 60
Feedrate (Jog) Parameter (i31[i51]) 80
Feedrate (Reference) Parameter (i06) 61
Following Error Limit Outputs 10 313 320 326 328
Following Error Limit Parameters (i00,i10) 57 64
Following Mode Control Parameter (i28[i48]) 78
Free Memory Locations 203
Freeze Position for Read Command 105 109
Frequency (Time Following) Calculations 68 394
Gain Multiplication Factor (i30[i50]) 80
Go Until Command Trigger Control (i26) 77 403
Go Until Movement Program Command (^) 119 185 403
Go Until: New Destination Calculation Time (tt) 119 414
Hall Sensor Phase Offset Parameter (i45/i54) 85 88 313
Handshake Enable Parameter (i03) 59
Handwheel Connectors Descriptions (J9,J11) 315 327 328 354
Handwheel Encoder Decoding Param. (i29[i49]) 79
Handwheel Following Mode Parameter (i28[i48]) 78
Handwheel Operating Mode Parameter (i16) 67
Handwheel Position Register Locations 203
Handwheel Scale Factor Parameter (i27[i47]) 78
Hexadecimal to Decimal Conversion 374
Highest Card Address Parameter (i19) 72
Home Feedrate Parameter (i04) 60
Home Flag/User Flag Control Parameter (i26[i46]) 77
Homing Commands 97 117
Homing Direction Parameter (i24[i44]) 77
Homing Offset Parameter (i25[i45]) 77
Homing Special Routine Examples 263
Host Communication Mode Selection 54 317
Host Connection Procedure 21
Hydraulic Cylinder Connection Example 360
i Parameter Descriptions 56
i Parameter Query Commands 104
i Parameter Summary 137
i Parameter Value Setting Commands 103 121 127
i08, Differences in Parabolic 182
I2T Limits (See Output Limits)
IBM PC Bi-directional Parallel Port (ACC-10) 14
IBM PC Software Executive Program (ACC-9) 14 171
IEEE-488 Interface (ACC-13) 15
If-type Statements (Motion Buffer) 113
If-type Statements (PLC) 124
Incremental Movement Program Commands 118
Induction Motor (See AC Induction Motor)
Initialization Procedure 406
Initialization/Setup Command Description 56
Integral Gain Constant (i23[i43]) 76
Integration Mode Parameter (i09) 63
In-Position Band Parameter (i01) 58

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

I/O Expansion Connector Descrip. (J8;JEXP)	5	10	327	350
I/O Map				199
I/O Specifications				8
JDISPX, JDISPY (J5,J6) (See Display Connectors)				
JEXP (J8) (See Expansion Connector)				
JMACH Connector (J1) (See Machine Connector)				
Jog Commands			97	98
Jog Feedrate Parameter (i31[i51])				80
JOPT (J12) (See Opto-22 Connector)				
JPAN (J7) (See Control Panel Connector)				
JPAR-RS232 (J2) (See Parallel, RS232)				
JPWR (J10) (See Power Connector)				
Jumper (E-Point) Descriptions				309
JXHW, JYHW (J9,J11) (See Handwheel Connectors)				
Labels for Subroutines				114
Learn Command				101
Learn Command Program Example				252
Limit Switch Information	24	29	319	321
Limit (Software) Parameters (i35[i55],i36,[i56])				81
Logical Operation Program Commands				122
Lookup Table Address Parameter (i43)				85
Lookup Table Select Parameter (i15)				67
Loop Control Program Commands				115
M Definition Default Assignments			138	139
M Function Definition Commands			103	122
M Function Query/Evaluation Commands		107	113	124
M Function Value Setting Commands		104	112	127
Machine Connection Procedure				29
Machine Connector Descrip. (J1;JMACH)	319	332	350	352	353
Machine Input/Output Mapping		138	199	249
Machine Input/Output Program Example				249
Magnetization Current Parameter (i42/i52)			85	87
Manual Control Description				10
Manual Data Input Mode Parameter (i02)				58
MDI Mode Handshake Timing			346	417
MDI Mode Program Example				296
Memory Capacity				5
Memory Map -- RAM (Parabolic SMCC)				213
Memory Map -- RAM (Standard SMCC)				205
Memory Map -- Status & Control Bits				200
Memory Map -- Useful Locations				203
Memory -- Free Locations			203	409
Motion Buffer Commands			111	142
Move Calculation Time Parameter (i12)				65
Move Programming Explanations			150	367
Move Until -- See 'Go Until'				
Movement Program Commands			117	118
Multiple Card Program Example		247	294	304
Multiple Card Setup Information		20	29	309
No Motion Condition				27

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Non-Buffer Commands				95	141
Offset from Home Parameter (i25[i45])					77
Open Loop Output Commands					98
Optical Isolation of Outputs		4	311	312	314
Options Listing					11
Opto-22 Connector Descriptions (J12;JOPT)	329	351	358	359	
Output Frequency Parameter (i12)					64
Output Limit Information			8	80	313
Output Limit Parameter (i32[i52])					80
Output Optical Isolation		4	311	312	314
Output Protective Limit (i33[i53])					81
Overview					2
P Parameter Function Formats				112	129
P Parameter Memory Locations			203	212	219
P Parameter Query/Evaluation Commands			107	113	124
P Parameter Value Setting Commands			104	122	127
Panel Connector Descriptions (J7; JPAN)				326	350
Parabolic Card i Parameter Overview					148
Parabolic Card Overview					147
Parabolic Mode Control for DMA (i63)					89
Parabolic Move Examples				257	382
Parabolic Move Profile Explanation					150
Parabolic Movement Program Command					118
Parabolic Path Generator Option (#4) Desc.					12
Parabolic Profiling Information					150
Parallel Bi-directional Port for IBM PC (ACC-14)					14
Parallel Encoder Parameters (Parab SMCC)					42
Parallel Interface Description		5	11	322	333	347
Parallel I/O Option (#1) Description					11
Parallel Port Addresses					199
Parameter Summary					55
PC Bi-directional Parallel Port (ACC-10)					14
PC Software Executive Program (ACC-9)				14	171
Percent Command/Value (See Feedrate Override)					
Performance Specifications					6
Permanent Memory Save Command					103
Permanent Memory, Differences in Parabolic					182
Permanent Program Storage Options (#2,#5)				11	12
Phase Advance Scale Factor (i41)					84
Phase Finding Current Value (i53)					88
Phase Finding Time (i52)					87
Phase Number (3/4) Select Param. (i44)					85
Physical Specifications					4
PLC Action Statements					127
PLC Buffer List Command				109	124
PLC Buffer Purge Command				109	124
PLC Commands				123	143
PLC Conditional Statements					124
PLC Diagrams					143
PLC Enable Parameter (i60/i62)					88

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

PLC Program Examples 237
PLC Timer Program Examples 267 269
PLC Timers 130
Position Encoder Decoding Param. (i39[i59]) 83
Position Feedback Device Summary 197
Position Following Scale Factor (i27[i47]) 78
Position Following (See Handwheel Following)
Position Freeze for Read Command 105 109
Position Integration Mode Parameter (i09) 63
Position Offset from Home Parameter (i25[i45]) 77
Position Offset -- Abs. Encoder/Resolver (i70[i71]) 91
Position Range Mode Control (i78[i79]) 184 94
Position Range Parameter (i37[i57]) 184 82
Position Register Locations 203 205 213
Position Scale and Format Param. (i38[i58]) 82
POT Motion Buffer Command 121 393
Potentiometer Connections 342A
Potentiometer Use Descriptions 8 66 375 393
Power Connector (J10; JPWR) 311 312 314 328
Power-up Procedure 20
Program Branching Commands 114
Program Branching Examples 241 246
Program Buffer Commands 11[illegible]
Program Buffer Purge Command 106
Program Command Summary 142
Program Control Commands 101
Program Editing Commands 107
Program Pointer Locations 203
Programming Manual 49
Programming Tutorial 43
Proportional Gain Constant (i20[i40]) 73
PWM Frequency Parameter (i12) 64
PWM Output Limit Parameter (i32[i52]) 80
PWM Output Protective Limit (i33[i53]) 81
Reset Commands 95 97
Resolver Commutation Offset (i54) 88
Resolver Parameters (Parabolic SMCC) 42
Resolver Setup 190 309
Resolver (Coarse) Size (i74[i75]) 93
Resolver -- Fine to Coarse Ratio (i66[i67]) 90
Resolver -- 'Coarse' Base Address (i68[i69]) 91
Resolver-to-Digital Converter (ACC-15) 16 195
Resolver/Absolute Encoder Offset (i70[i71]) 91
Resolver/Absolute Encoder Size (i72[i73]) 91
Resolver/Brushless Motor System Setup 39
Resolver/Encoder Base Address (i64[i65]) 89
Resolver/Encoder Select for Commutation (i50) 86
RS232 Interface Description 5 21 310 322 348
Runaway Condition 27
Sampling Time Parameter (i05) 61

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Save (to Permanent Memory) Command 103
Schematics 349
Second Resolver Base Address (i68[i69]) 91
Second Resolver Size (i74[i75]) 93
Second Resolver Speed (i66[i67]) 90
Serial Interface Description 5 21 310 322 348
Servo Algorithms 169 73
Servo Time Parameter (i05) 61 392 396
Setup of AC Induction Motor 38
Setup of Brushless Motor Systems 36
Setup of Non-Tach DC Systems 35
Setup of Tachometer DC Systems 32
Sinusoidal Acceleration Example 229
Slave Following Scale Factor (i27[i47]) 78
Slave Mode Control Parameter (i28[i48]) 78
SLEW Motion Buffer Command 121 393
Slip Gain Parameter (i45/i51) 87
SMCCAD Conversion Program (ACC-17) 17
Software Limit Parameters (i35[i55],i36[i56]) 81
Speed Set Command for Programmed Moves (F) 116
Status Bits (Display Panel) Description 72
Status Bytes Description (Parabolic) 181 201
Status Bytes Description (Standard) 105 200
Status Display Indicator (Board Digit) 10 202
Status Query Commands 104
Status/Error Display Description 10
Stop Condition Slew Control Param. (i11) 64
Stop Times Chart 96
Stopping Commands 95 97 114
Subroutine Labels 114
Synchronization Calculations 68
S-Curve Mode Profiling Information 158 364
S-Curve Mode Program Example 299 386
t Time, Differences in Parabolic 182
Tachometer DC Motor System Setup 32
Terminal Block Pinout Diagram 334
Terminal Block (ACC-8) 13
Terminal Program 14 171
Test Point Descriptions 316
Test Point Locations 317
Then-type Statements (PLC) 127
Three-Point Smoother Mode Profiling Info 158 364
Thumbwheel Multiplexer Board (ACC-18) 17
Thumbwheel Multiplexer Program Examples 302 378 382 386
Thumbwheel Read Examples 233 235 272 277 281
Time Base Control (See Feedrate Override, Time Following)
Time Following Calculations 68 392
Time Following Program Example 378
Time Following Scale Factor (i27[i47]) 78 394
Time for Accel/Decel Command 116

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Time Scale Parameter (i07)	62	393	397
Time Set Command for Programmed Moves (T)			116
Timer PLC Program Examples		267	269
Timers for PLC Programs			130
TORQUE Motion Program Command		121	402
Torque-Limited Mode		121	402
Trapezoidal Move Profile Explanation			367
Trigger Calculation Time Command (tt)		119	414
Trigger Response Program Examples	251	253	355
Triple Cycle (Commutation) Definition (i50)		41	86
User Flag Move Adjustment Command (^d)		119	185
User Flag Program Examples		253	255
User Flag/Home Flag Control Parameter (i26[i46])			77
Velocity Command for Programmed Moves (F)			116
Velocity Display Time Base Parameter (i17)			71
Velocity Feedforward Gain Constant (i22[i42])			76
Velocity Filter Time Delay Term (i22[i42])			76
Velocity of Homing Parameter (i04)			60
Velocity of Jogging Parameter (i31[i51])			80
Verification of Operation			23
Voltage-to-Frequency Converter (ACC-8V)		13	332
Wait Commands (WX, WY)			118
Wait Until Trigger Command		119	414
Wrong Direction Condition			26
WUT Command		119	414
Y-Axis Disable Parameter (i15)			66A
Zero Velocity Program Commands	117	118	119

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC.
OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CALIFORNIA 91324

(818) 998-2095 FAX: (818) 998-7807

RETURN MATERIAL FOR CREDIT POLICY

When a Delta Tau product has been purchased and received by the customer and the customer wishes to return the product and receive credit for it, the following applies.

1. During the first 30 days after receiving the product, the customer may return the product to Delta Tau:

    a. At no charge, if the purchase order was a 30-day consignment.

    b. If the purchase order was a standard purchase, then a 25% restocking charge will be applied.

2. During the first 30 to 90 days after receiving the product, the customer may return the product with a 35% restocking charge applied.

3. After the first 90 days, the product may not be returned for credit. In general, no product will be accepted by Delta Tau once the 90 day limit is expired. Special circumstances will be reviewed and the product may be approved for return, but only with written authorization from Delta Tau specifying the applicable terms and conditions.

NOTE: ALL RETURNED MATERIAL OR PRODUCT MUST BE ACCOMPANIED BY AN RMA (RETURNED MATERIAL AUTHORIZATION) NUMBER AND MUST BE SHIPPED PREPAID. DELTA TAU WILL NOT ACCEPT C.O.D. OR FREIGHT COLLECT SHIPMENTS.

CALL THE FACTORY AND ASK FOR CUSTOMER SERVICE, TO RECEIVE AN RMA NUMBER, BEFORE YOU SEND BACK YOUR PRODUCT.

cm\document\policy-1

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC.
OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CALIFORNIA 91324

(818) 998-2095 FAX: (818) 998-7807

LIMITED WARRANTY
AND
RETURN MATERIAL FOR REPAIR POLICY

WARRANTY STATEMENT

Delta Tau warrants its products (SMCC and PMAC) to be free of material, workmanship, and **software defects** for a period of one year from date shipped from Delta Tau.

Delta Tau Data Systems, or its designated representative, will repair, or at Delta Tau Data Systems' option, replace any product that has been returned by the purchaser and is confirmed by Delta Tau Data Systems to be defective.

Misapplication or misuse in excess of specification are not warranted and are the customer's responsibility. This warranty shall be null and void if this product has been altered without authorization, misused,or damaged by accident.

If Delta Tau receives a board for repair that is under warranty, and after testing nothing is found wrong with it, a standard minimum charge will apply. For our PMAC series the charge will be $210.00, for the SMCC series $125.00, and for the MCC series $90.00, plus any shipping charges incurred.

DELTA TAU DATA SYSTEMS SHALL NOT BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR OTHER TYPES OF DAMAGES RESULTING FROM THE USE OF THIS PRODUCT OTHER THAN THE LIABILITY STATED ABOVE. THIS WARRANTY IS IN LIEU OF ANY OTHER WARRANTIES EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

REPAIR PROCEDURE

Should you require either warranty or non-warranty repair, proceed as follows.

1. Call the factory, ask for the Repair Department, and obtain a Return Material Authorization Number (RMA#). You may then request a repair by either sending your product in or requesting an **Exchange/Repair** (see 5. below).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2. A. Always place or write the RMA# on the shipping label and on the box. This will prevent unintentional misplacement or unnecessary delays.

B. On a piece of paper inside the box provide the following information:

1) As good a description as possible of the problem the product has. Preferably a technician or engineer should provide this data. (PLEASE INCLUDE OUR FORM ENTITLED "FAULTY GOODS REPORT" FILLED OUT TO THE BEST OF YOUR ABILITY.)

2) The name and telephone (and/or fax) number of the person who can authorize the repair, and shipping & billing addresses.

C. Send the unit to be repaired, freight prepaid, to Delta Tau. C.O.D. or freight collect shipments will not be accepted by Delta Tau Data Systems.

3. If the repair of the product is Delta Tau's responsibility per warranty statement, there will be no charge for the repairs, and the product will be returned to customer, freight prepaid.

4. If the repair of the product is not under warranty or is the customer's responsibility, Delta Tau will call you with the amount of the repair charges for your authorization. You will then have to issue a purchase order for the repair of your product.

Should you choose to not have the unit repaired, there will be a charge for estimating and handling. The product will be returned to you and you will be billed for freight charges and repair or estimate charges.

If the board is deemed irreparable, Delta Tau will provide you with an excellent discounted price for an available board.

5. Normal repair time is one to two weeks. If the unit is needed in less than one week, and a loaner board is not available, a charge of one half the minimum amount will apply for expedited repair service.

6. TEMPORARY REPLACEMENT

For emergency situations and in order to reduce machine downtime, you may request an **Exchange/Repair.** You must then specify precisely the product you have with Part Number, Serial Number, Options, and Accessories. Failure to do so correctly may result in Delta Tau sending you a wrong or incomplete replacement product.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Delta Tau's minimum charges will apply for this service. Minimum charges being for the PMAC - $210, SMCC - $125, and for the MCC $90, plus shipping charges.

A. For warranty repairs, Delta Tau will ship you a replacement the following day, using second day delivery service, freight prepaid. Faster service will incur special handling and shipping charges.

Upon receipt of the exchange, the customer must return the defective product to Delta Tau, freight prepaid, within two weeks or less otherwise the customer will be billed for a new product shipment.

B. For non-warranty repairs the procedure is the same as for warranty repairs except that the customer will be billed for shipping charges and a flat-rate exchange/repair fee.

7. When a non-warranty repair is performed, the repair itself is warranted for 90 days for the same **identical** problem.

Warranty repairs are not to exceed the original warranty date, no matter when a warranty repair is made last.

cm\document\policy-2

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC.
OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CALIFORNIA 91324

(818) 998-2095 FAX: (818) 998-7807

==========

PURCHASE ORDER CANCELLATION POLICY

If a customer has issued a purchase order for the purpose of purchasing Delta Tau's products and wishes to cancel or modify the subject purchase order, the following applies.

1. If the purchase order is a single order (one non-scheduled delivery):

   a. Cancellation is free of charge if it occurs one week (7 days) prior to scheduled delivery date.

   b. A 5% charge is applied if cancellation occurs within one week of scheduled ship date.

   c. Modification of the order may be made up to the scheduled ship date and may result in rescheduling the shipment date.

2. If the purchase order is a multiple scheduled delivery order:

   a. Cancellation within two weeks (14 days) of order placement is free of charge.

   b. Thereafter and before the first shipment is made, a cancellation charge will be made which is in proportion to the material ordered for the subject purchase order. Generally a charge of 0% to 25% of the total order will be applied, depending upon circumstances.

   c. If cancellation occurs after the first or subsequent shipment(s) are made, then the terms in 2b above will apply to the balance of the order as well as a back-bill for the difference in the price charged for the units shipped versus Delta Tau's price list and normal quantity discounts of the actual units shipped.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

For example, if the order consists of 100 units at $500 each and 30 units have been delivered, the cancellation charges could be as follows:

Price of unit per Delta Tau's standard list price: $550

Cancellation charge:

= [(550-500) x 30] + [2 x 70 (500/100)]
= 1500 + 700
= $2200.00

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC. OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CALIFORNIA 91324

(818) 998-2905 FAX: (818) 998-7807

SCHEDULED ORDER POLICY

In order to allow customers to take advantage of quantity discounts for scheduled orders, certain requirements must be met:

- The product to be purchased must be delivered within a period of one year from the order date. Other arrangements may be made, but will require special agreement.

- A reasonably firm delivery schedule, specifying dates and product quantities to be delivered per shipment must be specified at the time the order is placed. The schedule may be changed during the course of the year, however, the specified product quantity must be shipped.

- If less than the specified total quantity is taken, then the discounted price will be changed to the price indicated on Delta Tau's standard "price list" for the subject product at the quantity level which was actually used. A "bill back" invoice will then be issued for the difference in price.

cm/document/policy-4

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC.
OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CA 91324

(818) 998-2095 / FAX: (818) 998-7807

SHIPPING POLICY

Delta Tau's method for shipping within the United States, for either a new order or for an RMA, will be to ship the merchandise UPS BLUE (or second day delivery by other carriers). The product will arrive in two days and will be shipped by air, thus decreasing the possibility of damage to the shipment. Delta Tau will also insure the merchandise for its full value.

The customer may specify another shipping method, if he chooses, and we will do our best to comply.

Delta Tau always insures any shipment leaving from our facilities to protect ourselves and the customer. If the customer does not want us to insure his package, we must receive complete prepayment of the invoice's full amount prior to shipment, or a letter stating the customer's wish for Delta Tau to waive the insurance and that the customer is accepting full responsibility of the shipment once it leaves our premises. All shipments are F.O.B. Northridge, CA and automatically imply transfer of product responsibility from Delta Tau to carrier and customer upon shipment pick-up by carrier.

The customer essentially takes possession of the shipped products once they leave our loading dock. Should the product get lost during an uninsured shipment, the customer is responsible for the product and must pay the invoice amount in full in thirty days. Therefore, to protect Delta Tau and its customers, Delta Tau insures all shipments or requests prepayment.

When shipping outside of the United States, Delta Tau will use its own discretion as to what carrier to use (usually DHL, UPS, or equivalent) unless otherwise requested by the customer.

The customer will be invoiced for charges incurred in shipping and handling, such charges are usually listed as a separate item on the original invoice. All shipments are F.O.B. our plant.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DELTA TAU DATA SYSTEMS, INC.
OF CALIFORNIA
9036 WINNETKA AVENUE
NORTHRIDGE, CA 91324

(818) 998-2095 / FAX: (818) 998-7807

SOFTWARE UPDATES

Delta Tau is happy to supply a software update, at no charge to the customer (except for shipping charges) for a period of six months after initial purchase. The update will consist of an EPROM, User's Manual, and Accessory 9D (if it was purchased). It will be the customer's responsibility to call Delta Tau to inquire if an update has been made and is available since his purchase. Not more than one free update will be provided to a given customer.

If the customer wishes to receive an update after the six month period (or after the first free update), it may then be purchased for a cost of $395.00. If the customer has multiple cards, extra proms may be purchased at a cost of $55.00 each.

Existing customers who have already received an update will also be required to pay the update fee, should they wish to receive the update.

NOTE: Corrections to legitimate errors found by the customer and verified by Delta Tau will, of course, be provided at no charge.

cm/document/policy-6

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com







Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP