

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***

1¾ « 9ï Ê Ê Ë ì î
ï


TECHNICAL MANUAL FOR

HIGHLAND MODEL V544

HIGH SPEED QUAD MICROSTEPPER VME MODULE

*** PRELIMINARY ***

MARCH 6, 1995

COPYRIGHT 1995 HIGHLAND TECHNOLOGY, INC.

TABLE OF CONTENTS

1. PRODUCT SPECIFICATION SHEET

2. INTRODUCTION
2.1 GENERAL
2.2 PART NUMBERS AND ORDERING INFORMATION
2.4 DETAILED SPECIFICATIONS

3. THEORY OF OPERATION
3.1 V544 VME MODULE
3.2 V740 STEPPER MOTOR DRIVER
3.3 STEPPER MOTOR BASICS
3.4 CONVENTIONS AND DEFINITIONS
3.5 ENGINEERING UNIT SCALING
3.6 MOTION CONTROL PHILOSOPHY
3.7 MECHANICAL AND ELECTRICAL LIMITATIONS

4. SETUP AND INSTALLATION
4.1 VME MODULE INSTALLATION
4.3 CONNECTOR TYPES AND PINOUTS
4.4 MOTOR TYPES AND TYPICAL CONNECTIONS
4.5 LIMIT SWITCHES AND MECHANICAL STOPS
4.6 INDEX SENSORS
4.7 DIPSWITCH SETUPS

5. PROGRAMMING
5.1 DUAL-PORT MEMORY CONSIDERATIONS
5.2 VME REGISTERS
5.3 MOTOR CONTROL REGISTERS
5.3.1 MOTOR COMMAND REGISTERS
5.3.2 MOTOR STATUS REGISTERS
5.3.3 MOTOR POSITION REGISTERS
5.3.4 MOTOR VELOCITY REGISTERS
5.3.5 MOTOR ACCELERATION REGISTERS
5.3.6 MOTOR LIMIT-SWITCH DECELERATION REGISTERS
5.3.7 MOTOR AUXILIARY REGISTERS
5.4 MODULE CONTROL REGISTERS
5.5 MOTOR COMMANDS AND PROTOCOL
5.5.1 MOTOR IDLE
5.5.2 RUN CLOCKWISE
5.5.3 RUN COUNTERCLOCKWISE
5.5.4 MOVE ABSOLUTE
5.5.5 MOVE INCREMENTAL
5.5.6 SEEK INDEX
5.5.7 BACK OFF CW LIMIT SWITCH
5.5.8 BACK OFF CCW LIMIT SWITCH

CONTENTS (CONTINUED)

5.5.9 DECLARE ZERO POSITION
5.5.10 DECLARE ABSOLUTE POSITION
5.5.11 READ MOTOR THETA
5.6 FINDING ZERO
5.6.1 ZERO USING THE INDEX
5.6.2 ZERO USING LIMIT SWITCHES
5.6.3 ZERO USING HARD LIMITS
5.7 POWERUP/POWERDOWN CONSIDERATIONS
5.8 INTERRUPTS AND TIMING DETAILS

6. MICROPROCESSOR PROGRAM VERSIONS AND REVISIONS
7. ENGINEERING DRAWINGS

**********DATASHEET**********

# 2. INTRODUCTION

## 2.1 GENERAL

The Highland model V544 is a complete, 4-channel stepper motor controller packaged in a single-wide 6U VME module. It interfaces to up to four model V740 motor driver modules. The V544/V740 system was designed specifically for compatibility with the performance requirements and EMI environment associated with high-energy physics research and other critical instrumentation environments.

The design intent of the V544 was to produce a cost-effective, feature-rich motion controller which would provide a large number of features and operating modes yet be simple to install and program. The basic RUN and MOVE commands should be sufficient for many common applications, with more complex functions provided to deal with the real-world complications of motion control and stepper motor features.

The V544 is a register based VME module, and is compatible with interim VXI specifications for a "B" format VXI module. Because it is binary-data register based, access to the V544 is fast and free of message formation overhead and potential hangup states.

The V544/V740 system uses an integrated approach to microstepping motion control. The V544 "requests" motor sine and cosine winding currents from the M740 via a noise-immune analog interface, so is not subject to "losing steps" due to impulse-type noise. Users may control all motion parameters through the VME control registers, without the necessity of setting up independent switch options on both an interface card and a motor driver. The V544 connects to V740 driver modules using standard, connectorized, shielded "D9" cable assemblies, eliminating hand wiring of indexer-to-driver cables.

The V740 driver includes output lowpass filtering, reducing radiated noise signals by at least 60 dB compared to conventional stepper drivers. Since the V740 uses a noise-immune analog angular position interface, and has no internal counters or states, it is highly resistant to induced electrical noise.

## 2.2 PART NUMBERS AND ORDERING INFORMATION

The High Speed VME Quad Microstepper Module is Highland model V544-1,
part number 22A544-1.

The standard 10-amp motor driver module is model V740-2, part number 22A740-2.

A lower performance version of the V544 is available as the V540, part number 22A540-1

## 2.3 DETAILED SPECIFICATIONS

PACKAGING: Single-width 6U VME module; VXI "B" format.

POWER REQUIRED: +5 volts at 2 A max, +- 12 volts at 50 mA max.

FUNCTION: Provides independent control of one to four 2-phase stepper motors, including profiled absolute and incremental moves, continuous run operation, initialization and indexing. All motions are executed with 1/256 step resolution.

FUNCTIONS PROVIDED:
- MOTOR IDLE
- RUN CLOCKWISE
- RUN COUNTERCLOCKWISE
- MOVE ABSOLUTE
- MOVE INCREMENTAL
- SEEK INDEX
- BACK OFF CW LIMIT SWITCH
- BACK OFF CCW LIMIT SWITCH
- DECLARE ZERO POSITION
- DECLARE ABSOLUTE POSITION
- READ MOTOR THETA
- STOP MOTION IN PROGRESS

MOTION PARAMETERS: All software programmable: parameters include...

Velocity: 0.0625 to 4096 steps/sec; 16 bits

Acceleration: 256 to 17,367,040 steps/sec/sec; 16 bits

Position: +- 8.38 million steps; resolution 1/256 step; 32 bits total.

Accurate position data is maintained at all times, including continuous "run" modes and during all exception conditions. Velocity and acceleration may be varied during "run" mode. Limit switch deceleration may be separately programmed.

OUTPUTS Drives four independent stepper motors, via V740 driver modules. Motors operate in 4-wire, 2-phase

mode.

INPUTS — Each motor channel is provided with five isolated inputs, provided through the V740 drivers. Inputs are Clockwise Limit, CCW, Limit, Index, and one spare.

PARKING — Each motor may be programmed to park at reduced current when not in motion. Park factors are 100%, 75%, 50%, and 0% of normal motor drive.

# 3. THEORY OF OPERATION

## 3.1 V544 VME MODULE

## 3.2 V740 STEPPER MOTOR DRIVE ASSEMBLY

## 3.3 STEPPER MOTOR BASICS

A stepper motor is a rotary electromagnetic positioner, generally consisting of a toothed, magnetized rotor and a number of toothed electromagnetic stator sections. By alternately energizing various stator sections, it is possible to make the rotor align with the various, offset stators and thus move the rotor and output shaft in small angular increments.

The general structure of a two-phase, permanent-magnet stepper is shown in the illustration below. Depending on how the four internal coils are wired, the assembled motors may have four, six, or eight leads. Motors are available with a range of step sizes, from four steps/revolution to 200 or more. Suitable motors are manufactured by Superior Electric, Sigma Instruments, and Compumotor Corporation. Eastern Air devices and Portescap both produce stepper motors specifically designed for microstepping.

Inexpensive "pancake" type steppers generally demonstrate non-uniform angular motion when microstepped, and are not suited to applications which require very smooth motion or precise positioning.

## 3.4 CONVENTIONS AND DEFINITIONS

The following conventions and definitions are applicable to discussions of the operation and programming of the V544 High Speed Quad Microstepper:

TWO-PHASE STEPPER: The V544 is designed to operate with two-phase stepper motors; such motors achieve incremental motion through the operation of electromagnetic windings positioned at 90 degree electrical angles (3 and 5-phase steppers are also manufactured). See section 3.3 for a brief discussion of stepper motor operation.

ROTATION: When viewed from the "nameplate" end of the motor, (that is, looking into the load) rotation in a clockwise direction is considered to be positive. An absolute move to a more positive coordinate will result in positive, clockwise rotation.

STEP: A step is a "full motor step", corresponding to the motion associated with a 90 degree electrical rotation of the motor phase inputs. This "full step" angle is equivalent to the manufacturer's nameplate "Steps/Rev" value divided into 360 degrees. Full stepping is accomplished by applying full rated voltage to the stepper motor windings in the proper, non-overlapping sequence. The most common stepper rating is probably 200 steps/rev, but values from 4 to several thousand are available.

MICROSTEP:An angular distance corresponding to a fraction of the full step size, achieved by applying intermediate values of current to the windings of a 2-phase stepper. For the V544, a microstep is defined to be 1/256 of a full step.

VELOCITY: The V544 treats the angular velocity of a stepper motor in steps per second. The resulting physical rotation speed of the motor in revolutions

per second is this velocity (steps/sec) divided by the motor's rated steps-
per-revolution.

POSITION: The V544 defines the absolute position of a stepper motor shaft as the
total number of steps which the motor has moved since the "zero" position
was last declared. The position may be positive or negative, with positive
being an angular position clockwise from the declared zero. Multiple
rotations will accumulate increasing angular position values, to a
maximum of about +-8 million steps.

PROFILE: A "profile" is a description of the precise dynamics of a motor motion,
usually expressed as a definition or graph of the motor velocity as a
function of time. Most "move" type commands have classic profile:
accelerate at constant acceleration to a target velocity; run at constant
velocity; decelerate at a constant rate to final stop.

INDEX POSITION: The index position is an absolute motor angular position which is the assumed zero coordinate for absolute motor move commands: When the V544 is powered up, its position registers are cleared, "zero position" is assumed, and motor electrical drive is set to the zero-step output state. The user may subsequently execute any of a number of predictable repositioning actions and subsequently declare a new zero index position.

INDEX SENSOR: The V544 may be connected to an index sensor device which defines the desired absolute zero index position. This device is commonly a mechanical limit switch or optical sensor. Commands are provided which allow the V544 to operate a motor such as to seek the index-sensor trip point and subsequently set the motor absolute position registers to a desired reference value.

LIMIT SENSORS: Systems which are capable of finite travel are often equipped with mechanical or optical limit switches which indicate to the V544 that motion past the sensed position should be stopped. Limit sensors should preferably be arranged so that the motor cannot proceed past the sensors such that the limit indication will drop out should motion continue; mechanical stops, located somewhat past the expected limit switch deceleration point, are preferred.

PARKING: In certain applications in which a high holding torque is not required, the V544 may be programmed to reduce the motor drive current after the completion of programmed motor commands. This action is referred to as "motor parking" and is discussed in section 5.3.1.

DETENT: Permanent-magnet stepper motors generally show a pronounced detent effect, attempting to align themselves to an integral step position when

coil power is removed. This may result in a small (1/2 step or less) motion when the motor is deenergized, either by turning off crate power or by setting the motor to a reduced parking current. See section 5.6 for further discussion. The less-common variable-reluctance type motors do not have a pronounced poweroff detent action.

## 3.5 ENGINEERING UNIT SCALING

The V544 communicates with the VME system computer by exchanging 16 and 32-bit binary integers. Scaling of these binary values to physical units is as follows:

### VELOCITY

Velocity is expressed as a 16-bit, binary, unsigned integer. The integer value ranges from 0 (zero velocity) to positive 65535 (maximum velocity). If the VME computer writes an integer "N" into an V544 velocity register, the corresponding motor velocity will be...

Velocity in steps/sec = N / 16

So the maximum value of N, namely 65535, will result in a motor velocity of 4095.96 steps per second. The smallest non-zero N, N=1, will then give a step rate of 0.0625 steps/sec., equivalent to 16 microsteps per second.

The actual step rate of the motor will equal the N/16 value to an accuracy of better than 0.01 percent.

### ACCELERATION

The VME computer specifies acceleration by writing the desired register with an unsigned, 16-bit integer. If the integer value is "M", in the range of 0-65535, then the corresponding motor acceleration will be...

Acceleration in steps/sec/sec = M * 256

So the maximum value of M, M=65535, gives an acceleration of 17,367,040 steps/sec/sec, and the minimum M, M=1, gives a rate of 256 steps/sec/sec.

Note that since the maximum motor speed is 4096 steps/sec, the maximum selectable acceleration value, M=65535, results in instantaneous motor starting and stopping with no discernable acceleration or deceleration limits.

In a stepper system with moderate loading and no special dynamics limits, an "M" value in the area of 0.1 to 0.002 times the velocity

value might be suitable.

POSITION

Absolute or incremental positions are specified as a pair of 16-bit values which are treated logically as a single signed, 2's complement, 32-bit integer.

Given a 32-bit integer K, the corresponding angular position is ...

Position in steps = K / 256

A few integer:position correspondences are then...

| INTEGER K | POSITION IN STEPS | |
|---|---|---|
| 0 | 0.000 | (reference position) |
| 1 | 0.0039 | (CW 0 steps + 1 microstep) |
| 256 | 1.000 | (CW one full step) |
| 65535 | 255.996 | (CW 255 steps + 255 microsteps) |
| 2,147,483,647 | 8388607.996 | (CW 8.38 million steps) |
| -1 | -0.0039 | (CCW 0 steps + 1microstep) |
| -65536 | -256.000 | ( CCW 256 full steps) |
| -2,147,483,648 | -8388608.000 | ( CCW 8.38 million steps) |

Note that the user must write the most significant 16 bits of the position value (including the sign bit) into the motor's COARSE POSITION register and the least significant 16 bits (including the microstep bits) into its FINE POSITION location. Given a position in real variable STEPS, one could convert STEPS to a 32-bit integer K and subsequently write the position to the MOTOR 1 position registers...

```
K = STEPS * 256             ; convert STEPS to 32-bit integer

I = K >> 16                 ; isolate high 16 bits as 16-bit integer
```

```
          J = K AND 65535             ; isolate steps: usteps as
16-bit integer

   CALL WVME (V544+x, I)    ; write MS position

   CALL WVME (V544+x+2, J)       ; write LS position
```

## 3.6 MOTION CONTROL PHILOSOPHY

Conventional single-channel, full-step motion controllers generally define motions in terms of discrete execution of full-step motor coil switching actions. Acceleration is thus viewed as increasing the step frequency, and running at constant velocity is expressed as the issuing of a certain number of step iterations per second. Most microstepper controllers follow this convention, but just treat the motor as if it had finer effective steps.

The V544, being a microprocessor-based, 4-channel microstep controller, uses a different motion control approach. The V544 contains four internal "motor demons", one for each of the four motors. Each demon is executed at a constant rate of 4096.000 times per second, and, at each such run, the demon computes and outputs a new angular position command to its motor, expressed as an appropriate pair of motor winding currents.

Thus the motors controlled by the V544 are "stepped" at a constant frequency but with a finely controlled angular increment, whereas a conventional stepper may be thought of as stepping at a finely variable frequency, but with a fixed, coarse angular increment. The practical consequence of this architecture is that the V544 will always microstep, producing smooth angular motion at any specified angular velocity, while still allowing the user to precisely specify motional parameters and angular positions.

The angular position of a motor controlled by the V544 may be specified to a resolution of 1/256 of a full step, or one "microstep". If the microstep part of a position value is set to zero, the motor will then come to rest at a standard "full step" angular position. The V544 may thus be instructed to make a controlled-profile move to any integral or microstep angular position, and it will perform the motion in a smooth and apparently continuous action. The V544 CANNOT be programmed to emulate a conventional stepper-controller profile, namely the execution of a profiled move in a series of jerky steps.

Note that the V544 will execute the specified acceleration, velocity, and position commands with digital accuracy, limited only by the resolution of the binary inputs and the precision of the microprocessor's clock, which is typically accurate to better than 0.01 percent.

The V544 normally operates at up to 4096 full steps per second. Since the system has a resolution of 1/256 of a full step, this output rate is equivalent to a conventional indexer/ driver system operating at a pulse rate of 1,048,576 steps per second.

## 3.7 ELECTRICAL AND MECHANICAL LIMITATIONS

Although the V544 may be directed to issue motion commands at speeds of up to 4096 steps/second and at angular increments down to 1/256 step, several effects may limit the realizable speed and position accuracy in a real system. The major limiting effects are:

MOTOR INDUCTANCE: All motor windings have an inherent current buildup time constant, approximately equal to the L/R time constant of the motor windings. At higher step speeds, this time constant prevents rapid change of winding current and thus causes a dropoff of available torque at higher step speeds. When driven at their rated coil voltages, many small steppers are capable of maximum step speeds in the 300 - 500 step/second range when unloaded, and less when significantly loaded. Internal magnetic effects may reduce this speed limit to a lower value than expected from the L/R limits alone. Because the V740 uses constant-current drive, motor coil time constants may be effectively decreased by using a higher power supply voltage and "forcing" current into the motor windings. For best high-speed torque, always select a motor with the LOWEST rated coil voltage practical.

INERTIAL LOADING: Heavy moment-of-inertia loads will limit the maximum safe acceleration/deceleration capability of any stepper motor, such that the motor may skip steps or even fail to start if excessive acceleration is programmed.

STATIC FRICTION: Mechanical friction in the motor bearings and load will tend to make the motor "stick" in place, with the magnitude of the stick approaching one-half step (128 microsteps) as the frictional load approaches the motor's maximum torque capability. Such sticking will result in hysteresis-like positioning errors when the motor is at rest, and nonuniform (jerky) motion when the motor is running at low speeds. A resolution of one microstep (1/256 step) is probably not realizeable in most real stepper systems.

ELECTROMAGNETIC NONLINEARITIES: Numerous nonlinear effects conspire to distort the positional accuracy of a microstepping motor. They include mechanical imperfections in the shape of the rotor and stator laminations, magnetic hysteresis and saturation effects, and variations in coil resistance. These effects combine to produce a nonlinear relationship between requested and actual angular position, with the primary nonlinearity having a 4-step periodicity. As a result, position may be in error and "smooth" motion may have a periodic velocity modulation. In lightly loaded systems, such nonlinearities may be in the 5-25 microstep range.

ALGORITHMIC ERRORS: The V544 can make a number of basic rounding and quantization errors in computing motor drive currents. Further errors are introduced by the 12-bit resolution of the motor-driver DAC stages. Resulting positioning errors are generally below 2 microsteps. Steady-state velocities will be accurate to better than 0.01 per cent. Because the V544 "motor demons" run at a fixed rate, motor dynamics calculations are limited in their time resolution; small deviations from ideal acceleration/ deceleration profiles may be expected, especially at both extremes of programmed velocity/acceleration ratios. These errors are generally insignificant in independent motor applications, but may limit the accuracy of co-ordinated multi-axis moves, such as machining, plotting, or other critical vector-path situations. Any motor positioning errors are non-cumulative.

# 4. SETUP AND INSTALLATION

## 4.1 VME MODULE INSTALLATION
## 4.3 CONNECTOR TYPES AND PINOUTS
## 4.4 MOTOR TYPES AND TYPICAL CONNECTIONS

## 4.5 LIMIT SWITCHES AND MECHANICAL STOPS

Each motor may be equipped with one or two limit switches which indicate to the V544 that motion should be stopped. Separate clockwise and counterclockwise switches should be used; the V544 logic is such that, when a motor is moving in a given direction (say, CW) then the closure of the corresponding limit switch (here, the CW switch) will cause the motor to stop. Motion backing away from a closed limit switch
is permitted.

Mechanical stops should be used wherever practical to prevent any possible overrun past the limit switch sensing region, and should ALWAYS be used in situations in which
overrun could cause injury or damage.

## 4.6 INDEX SENSORS

Index sensors may be connected in a manner electrically similar to limit switches. An index will typically supply a closure when the motor is at some user-defined reference position. The V544 may be programmed to seek this index position and then align its coordinate system with the user-defined system.

Since index sensors are commonly located within the working motion range of a positioning system, they are usually designed so that they may be overrun in either direction.

## 4.7 DIPSWITCH SETUPS

A DIP switch, S1, is provided to select the VME base address for the V544 module. The module base address may be placed at any 128-byte boundary from C000h to FF00H in the VME short (16 bit) address space. S2 section bit weights are...

| S2 SECTION | BIT | HEX VALUE | |
|---|---|---|---|
| 8 | | 6 | (ignored) |
| 7 | 7 | 80 | |
| 6 | 8 | 100 | |
| 5 | 9 | 200 | |
| 4 | 10 | 400 | |
| 3 | 11 | 800 | |
| 2 | 12 | 1000 | |
| 1 | 13 | 2000 | |

The "total" module base address is C000 hex plus the hex values of all ON sections of S2. To comply with VXI conventions, hex address FF80 should not be selected as a V544 base address. For the standard version of the module, switch section 8 is inoperative.

Dipswitch S3 has four sections. Each section is associated with one motor, and is marked to show the association. When a switch section is OFF, that motor is assumed to use normally OPEN limit switches, and when the switch is ON, limit switches are asssumed to be normally CLOSED. To operate a motor with no limit switches, set its selection OFF.

Switch S4 selects the VME interrupt level at which the module operates. If all switch sections are OFF, the module will not generate VME interrupts and will pass on the interrupt grant signal with minimum delay. Any interrupt level from 1 to 7 may be elected by turning on combinations of the switch sections labeled IRQ1, IRQ2, and IRQ4.

# 5. PROGRAMMING CONSIDERATIONS

## 5.1 DUAL-PORT MEMORY CONSIDERATIONS

The primary data interchange and control mechanism between the V544 and the VME bus is the 64-word by 16-bit dual-port memory located on the module. Because both the VME bus and the microprocessor have full read/write access to most DPM locations, an "anti-collision" mechanism is needed to prevent data usage conflicts and interlock problems. In the V544, this interlocking is accomplished by the simple means of assigning, by convention, certain locations in DPM as being written into only by the VME master, and other locations reserved for writing only by the microprocessor. Although normal operation of the V544 assumes that the VME port will adhere to the documented conventions of register use, system design is such that no unusual events will occur should the VME port write to its "read only" DPM locations. If the VME-side computer were to alter a "module output data" location (say, by writing into a motor status register), the V544 will restore that data in the normal course of operation.

The V544 presents a block of 64 16-bit registers to the VME bus. All registers are located in the A16 (short) VME addressing space and respond to hex address modifiers 29 and 2C. The base (start) address of a V544 is set by an onboard DIPswitch, in the range of C000 to FF80 hex.

## 5.2 VME REGISTER USAGE

The V544 provides access to 64 separate 16-bit registers. Register usage is as follows; OFFSET is the hex VME bus address relative to the module base address.

| OFFSET | | REGISTER FUNCTION |
|---|---|---|
| 00h | VXID | VXI ID/LOGICAL ADDRESS |
| 02 | VTYPE | VXI DEVICE TYPE |
| 04 | VCSR | VXI STATUS/CONTROL REGISTER |
| 06 | PGID | FIRMWARE PROGRAM ID |
| 08 | PREV | PROGRAM REVISION |
| 0A | VSN | BOARD SERIAL NUMBER |
| 0C | OPTS | BOARD OPTIONS |
| 0E | SCAN | SCAN COUNTER |
| | | |
| 10 | MOTOR 1 | REQUESTED COARSE POSITION |
| 12 | MOTOR 1 | REQUESTED FINE POSITION |
| 14 | MOTOR 1 | REQUESTED VELOCITY |
| 16 | MOTOR 1 | ACCELERATION |
| 18 | MOTOR 1 | LIMIT SWITCH DECELERATION |
| 1A | MOTOR 1 | AUXILIARY REGISTER |
| 1C | MOTOR 1 | STATUS REGISTER |
| 1E | MOTOR 1 | COMMAND REGISTER |
| | | |
| 20 | MOTOR 2 | REQUESTED COARSE POSITION |
| 22 | MOTOR 2 | REQUESTED FINE POSITION |
| 24 | MOTOR 2 | REQUESTED VELOCITY |
| 26 | MOTOR 2 | ACCELERATION |
| 28 | MOTOR 2 | LIMIT SWITCH DECELERATION |
| 2A | MOTOR 2 | AUXILIARY REGISTER |
| 2C | MOTOR 2 | STATUS REGISTER |
| 2E | MOTOR 2 | COMMAND REGISTER |

| | |
|---|---|
| 30 | MOTOR 3 REQUESTED COARSE POSITION |
| 32 | MOTOR 3 REQUESTED FINE POSITION |
| 34 | MOTOR 3 REQUESTED VELOCITY |
| 36 | MOTOR 3 ACCELERATION |
| 38 | MOTOR 3 LIMIT SWITCH DECELERATION |
| 3A | MOTOR 3 AUXILIARY REGISTER |
| 3C | MOTOR 3 STATUS REGISTER |
| 3E | MOTOR 3 COMMAND REGISTER |
| | |
| 40 | MOTOR 4 REQUESTED COARSE POSITION |
| 42 | MOTOR 4 REQUESTED FINE POSITION |
| 44 | MOTOR 4 REQUESTED VELOCITY |
| 46 | MOTOR 4 ACCELERATION |
| 48 | MOTOR 4 LIMIT SWITCH DECELERATION |
| 4A | MOTOR 4 AUXILIARY REGISTER |
| 4C | MOTOR 4 STATUS REGISTER |
| 4E | MOTOR 4 COMMAND REGISTER |
| | |
| 50 | MOTOR 1 ACTUAL COARSE POSITION |
| 52 | MOTOR 1 ACTUAL FINE POSITION |
| 54 | MOTOR 2 ACTUAL COARSE POSITION |
| 56 | MOTOR 2 ACTUAL FINE POSITION |
| 58 | MOTOR 3 ACTUAL COARSE POSITION |
| 5A | MOTOR 3 ACTUAL FINE POSITION |
| 5C | MOTOR 4 ACTUAL COARSE POSITION |
| 5E | MOTOR 4 ACTUAL FINE POSITION |
| | |
| 60 | MOTOR 1 ACTUAL VELOCITY |
| 62 | MOTOR 2 ACTUAL VELOCITY |
| 64 | MOTOR 3 ACTUAL VELOCITY |
| 66 | MOTOR 4 ACTUAL VELOCITY |

The remaining registers (at offset 67 through 7E) are reserved for future use.

The VXID register is the VXI-compatible ID/LOGICAL ADDRESS register. It contains the value FEEE hex, the "F" digit denoting a register based, 16-bit VME/VXI device, and the "EEE" hex digits being the registered identification of Highland Technology as the manufacturer.

The VTYPE register contains the hex value 5810, equivalent to the number 22544 decimal; this the the VXI-standard module type.

The VCSR register is the VXI-compatible control/status register. After initial powerup tests, this register will display the hex value 400C , corresponding to the "PASSED", "READY", and "MODID" VXI flags. The "FREEZE" bit (8000 hex) will allow the user program to halt data updates during a read. To use, set this bit, read the data, and then unset the bit. Note that motor parameters are frozen, but VXI registers are updated.

| BIT | NAME | FUNCTION | DECIMAL VALUE |
|---|---|---|---|
| 0 | RES | Board Soft Reset | 1 |
| 2 | SFT | Self-tests passed | 4 |
| 3 | RDY | Board ready to operate | 8 |
| 14 | MOD | Module ID 'off' flag | 16384 |
| 15 | FRZ | Freeze motor position updates | 32768 |

The PGID register identifies the microprocessor program version; it generally contains the hex value 5810, equivalent to 22544 decimal; special versions may contain other values.

The PREV register contains, in its low byte, an ASCII code identifying the revision of the microprocessor program. The typical value is 41 hex, corresponding to the character "A".

The VSN register contains a unique board serial number assigned by the factory.

The OPTS register contains board options set at the factory.

| BIT | NAME | FUNCTION | DECIMAL VALUE |
|---|---|---|---|
| 0 | ENC | Encoder option enabled | 1 |
| 1 | ACT | Encoder FPGA installed | 2 |
| 2 | SER | Serial port enabled | 4 |

The SCAN register is incremented once each time the microprocessor program scans all motor control registers, about every 260 uSec. This is useful as an indicator that the module is "alive" and operating normally.

## 5.3 MOTOR CONTROL REGISTERS

### 5.3.1 MOTOR COMMAND REGISTERS

Each motor has a dedicated 16-bit motor command register. This register is loaded from the VME bus to request initiation of motor operations, and may be read by VME to monitor the acceptance and completion of such commands. Bit assignments in the motor control registers are as follows, assuming the notation of LSB = BIT "0":

| BITS | NAME | FUNCTION | DECIMAL VALUE |
|---|---|---|---|
| 0-4 | CMD | MOTOR OPERATION COMMAND CODE | 0-31 |
| 5 | GO | INITIATES COMMAND EXECUTION | 32 |
| 6 | BUSY | INDICATES COMMAND ACCEPTED | 64 |
| 7 | DONE | INDICATES COMMAND COMPLETED | 128 |
| 9,10 | PARK | CONTROLS MOTOR POWERDOWN CURRENT | 512, 1024 |
| 14 | IEN | ENABLES INTERRUPT ON "DONE" | 16384 |
| 15 | ERR | COMMAND ERROR FLAG | 32768 |

Bits 0-4 of the motor control register define 32 possible motor commands. Currently-defined commands are...

| COMMAND | OPERATION PERFORMED |
|---|---|
| 0 | IDLE |
| 1 | RUN CW |
| 2 | RUN CCW |
| 3 | MOVE ABSOLUTE |
| 4 | MOVE INCREMENTAL |
| 5 | SEEK INDEX |
| 6 | BACK OFF CW LIMIT SWITCH |
| 7 | BACK OFF CCW LIMIT SWITCH |
| 8 | DECLARE ZERO POSITION |
| 9 | DECLARE ABSOLUTE POSITION |
| 10 | READ MOTOR THETA |

A command error will occur if...

Any motion is commanded in the direction of a closed limit switch.

Any motion command is initiated when the appropriate motor driver does not
have DC power.

If any motion command is initiated with a zero value of acceleration or velocity.

See section 5.5 for details of each motion command.

The two PARK bits determine motor idling current. If both bits are zero, the motor will remain at full current at all times. Other values of the PARK bits will result in the motor currents being reduced by a selectable factor; a motor will automatically drop to "park" current one second after any motion command is completed. Motor power dissipation drops as the square of the current cutback factor. PARK factors are...

| BIT 10 | BIT 9 | PARK CURRENT | EQUIVALENT MOTOR POWER |
|---|---|---|---|
| 0 | 0 | 100 % | 100 % |
| 0 | 1 | 75 % | 56 % |
| 1 | 0 | 50 % | 25 % |
| 1 | 1 | 0 % | 0 % |

Note that, for a permanent-magnet stepper, dropping the motor idle current may result in motor motion toward the motor's natural detent positions. The READ MOTOR THETA command can be used to "locate" these detents.

The V544 powers up with all motors parked at zero current. Executing any command will set the parking bits to the user's desired values.

### 5.3.2 MOTOR STATUS REGISTERS

Each motor has an associated status register. The register is updated with "realtime" bit states about every 250 microseconds during any motor operation, and is always updated just prior to the DONE bit being set in the motor command register, at the end of any motion command; no further updates are done once the command is completed.

All status bits are instantaneous as of the time they are posted, and none are latched. The user may issue an IDLE command (code 0) to refresh the status bits with current realtime values.

Status bit assignments are...

| BIT | NAME | FUNCTION |
|---|---|---|
| 0 | INDX | True when INDEX sensor input is active |
| 1 | CWL | True when CLOCKWISE LIMIT sensor is active |
| 2 | CCWL | True when CCW LIMIT sensor is active |
| 3 | MOTR | True if motor driver is unpowered or senses error. |
| 5 | SPRA | True if SPARE input is active |
| 15 | BADC | True if illegal command has been detected |

### 5.3.3 MOTOR POSITION REGISTERS

There are two sets of motor position registers for each motor. A SET position register used for setting the desired position and an ACTUAL position that shows the current motor absolute position. The SET position register may contain absolute and incremental position data. The user specifies motor positions as a 32-bit quantity, as detailed in section 3.5. For MOVE ABSOLUTE and MOVE INCREMENTAL commands, the position registers must be loaded just before the command is initiated. Once a command is initiated, the ACTUAL position registers will be updated regularly (at least every 260 uSec) to reflect realtime motor position, and will show the motor's final resting position at the end of any operation, including the IDLE command. The SET position registers are used only to determine the new absolute or incremental position.
Note that the actual position registers are located AFTER the normal block in locations (50h~5Fh)

Motor positions will be updated during any command execution, including the continuous RUN commands. If motor positions are read "on the fly", the user must take precautions to avoid reading "skewed" COARSE and FINE position values, caused by the fact that these values are both written (by the V544) and read (by VME) as two distinct operations. It is suggested that

the user read COARSE POSITION, FINE POSITION, and COARSE again; if the two COARSE values differ, the data is suspect and the READS should be repeated. During a RUN, the actual position registers may occasionally make a 2's complement overflow past +- 8.38 million steps. Alternately, the user may take advantage of the 'freeze' bit in the VXI CSR. See VCSR for details.

In summary, the V544 module keeps the set position register intact and updates the actual position register. Any additional writes to the set position register after the move has started will be ignored. Any writes to the actual position register will be ignored and overwritten with current actual position data.

### 5.3.4 MOTOR VELOCITY REGISTERS

Each motor has an associated SET velocity register and an ACTUAL velocity register, with scaling as noted in section 3.5. Set velocity must be set to a nonzero value prior to initiating any motion command. Certain commands may be prematurely terminated by setting the velocity value to zero during the command execution, as described in section 5. The ACTUAL register will display current realtime velocity updated approximately every 260 microseconds.

Very low velocity values will result in extended-duration commands which may not display obvious motion. A full-span MOVE of 8.38 million steps, with a velocity of "1", will require about 4 years to complete.

### 5.3.5 MOTOR ACCELERATION REGISTERS

Each motor has an acceleration register which must be loaded prior to initiation of any motion command. See section 3.5 for engineering unit scaling.

### 5.3.6 MOTOR LIMIT-SWITCH DECELERATION REGISTERS

Prior to issuing any motion command, the user may optionally declare a LIMIT SWITCH DECELERATION value. If this value is zero, the motor, upon hitting a limit switch, will decelerate at the normal ACCELERATE rate. If this value is nonzero, the specified LS DECEL rate will be used for limit-switch deceleration. Note that although the maximum deceleration rate (65535 decimal) will theoretically result in a "brickwall" stop, such a stop command may, in many systems, cause the motor to slip one or more steps and actually overrun the limit switch further than would a more controlled stop.

The LIMIT SWITCH DECEL rate is scaled identically to ACCELERATION values.

The LIMIT SWITCH DECELERATION rate also applies to stopping the motor when the index input is sensed during a SEEK INDEX command. See section 5.5.6.

### 5.3.7 MOTOR AUXILIARY REGISTERS

Each motor has one associated general-purpose register which may

be used to pass or receive parameters for specialized commands. Specific uses of these registers will be detailed in individual command descriptions.

## 5.5 MOTOR COMMANDS AND PROTOCOL

Because the motor command/status registers are located in dual-port memory, they may be read or written at any time by either the V544 internal microprocessor or by the VME control computer. It is necessary that the VME computer abide by certain simple protocol conventions to avoid timing conflicts with the microprocessor. The required sequence is...

a. The VME computer should always read and check the command/status register before attempting to write to it; the DONE bit must be true in the motor command register before ANY new command code can be written.

b. Desired motion parameters (velocity, acceleration, LS decel) must be set up prior to issuing a motion command. If the previous motion parameters are acceptable, they need not be reloaded. For MOVE-type commands, POSITION must be loaded immediately before initiating the command.

c. The VME program must execute a command by a single 16-bit bus write operation containing the function code desired, with the GO bit set, and with the desired PARK bit pattern included.

d. The BUSY bit may be checked to verify that the requested command has been initiated. BUSY may not appear immediately after the GO bit is asserted.

e. No parameter changes are generally allowed during a motor motion command execution, except for those explicitly permitted during the RUN and MOVE commands, as described below.

f. The various status and error bits will generally not be valid until the requested operation is completed and the DONE bit is raised by the V544 microprocessor. At the end of any command, the POSITION registers will contain the final motor position.

### 5.5.1 COMMAND 0 : IDLE

The command register will display the IDLE command code following powerup. This state indicates that no motor activity is in progress. The DONE bit will be set.

The VME system may issue the IDLE command (write just the GO bit into a motor command register) in order to change the states of the PARK bits or to update the motor status or position registers.

### 5.5.2 COMMAND 1 : RUN CLOCKWISE

The VME computer may request that a motor enter the RUN CLOCKWISE state by loading the decimal value 33 (eg, CMD 1 + GO) into the command register. A typical sequence would be...

1. Check the control register for DONE asserted; loop if DONE is off.

2. Write the desired velocity to the ACTUAL velocity register.

3. Write the desired acceleration to the acceleration register.

4. Write decimal 33 ("RUNCW" + "GO") into the command register.

The motor will then shortly go "BUSY" and the motor will begin to move clockwise at the specified velocity. The motor speed will ramp up to the specified speed at the specified acceleration, and then continue at a constant velocity.

To stop the RUN CW motion command, the user may at any time declare a velocity of zero; the motor will then decelerate to a stop, at which time the RUN mode will terminate and the command register "DONE" bit will rise. An ACCELERATION value of zero is illegal; if such a value is written prior to executing the RUN CW command, the command will be refused and the DONE and SUMMARY ERROR bits will be set in the command register, with COMMAND ERROR set in the motor status register.

Absolute motor position is updated during a RUN, and will reflect the motor's final stopping position at the end of the RUN.

RUN CW will also be terminated by the actuation of the CW limit switch, and will generate a limit switch error if the command is issued while the limit switch is actuated. Again, the POSITION registers will be current at stop.

Before issuing a RUN command, a LIMIT SWITCH DECELERATION value may be specified; see section 5.3.6.

### 5.5.3 COMMAND 2 : RUN COUNTER-CLOCKWISE

Except for the direction of rotation, this command is identical to the RUN CW command. The command will terminate upon sensing an active CCW limit switch.

### 5.5.4 COMMAND 3 : MOVE ABSOLUTE

This command will cause a motor to move to a specified absolute step position. The setup sequence is...

1. Check the command register for DONE asserted; loop if DONE is off.

2. Write the desired velocity into the REQUESTED velocity register.

3. Write the desired acceleration into the acceleration register.

4. Write the desired 32-bit destination position into the REQUESTED position register.

5. Write decimal 35 ("MOVE ABS" + "GO") into the motor command register.

The motor will then move to the specified absolute position, using the specified velocity and acceleration/deceleration rate, following the standard "accelerate-cruise-decelerate" profile. As in the RUN commands, the move may be stopped prematurely by changing the velocity to zero. The move will also be terminated by encountering an appropriate limit switch. In any case, the final motor position will be posted in the two ACTUAL position registers after the motor comes to rest and just prior to the DONE bit being asserted; barring a premature stop or limit switch hit, this value will be exactly equal to the value asserted by the VME computer prior to issuing the MOVE command.

Note that the MOVE command may result in clockwise or counter-clockwise motor motion, depending on the previous and requested motor positions. Remember, the 32-bit requested motor position is a 2's complement, signed quantity, as noted in section 3.5.

### 5.5.5 COMMAND 4 : MOVE INCREMENTAL

The MOVE INCREMENTAL command is similar to MOVE ABSOLUTE, but the declared move distance is assumed to be relative to the present motor position. Load the REQUESTED position register with the incremental distance. At the end of a normal MOVE INCREMENTAL operation, the ACTUAL position registers will reflect the current

absolute position. Note that a negative 2's complement MOVE INCREMENTAL distance will result in CCW rotation.

Any incremental move will NOT result in loss of any previously declared absolute position origin.

### 5.5.6 COMMAND 5 : SEEK INDEX

This command is identical to MOVE ABSOLUTE, except that motor motion will stop if the INDEX sensor is actuated. The LIMIT SWITCH DECEL rate will be applied to stopping motion. The user may then read the ACTUAL position registers to determine the position at which the motor came to rest. If extreme index-position accuracy is required, the user may wish to approach the index at a very low velocity, with perhaps two or more progressively slower seeks in succession.

If the INDEX sensor is not detected, the operation will terminate as if it were a regular MOVE ABSOLUTE.

Refer to section 4.6 for notes on index timing.

### 5.5.7 COMMAND 6 : BACK OFF CW LIMIT SWITCH

This command behaves identically to MOVE ABSOLUTE, but motor motion will stop if the V544 senses that the CW limit switch is no longer actuated. For this command, the only legal direction of motion is CCW.

### 5.5.8 COMMAND 7 : BACK OFF CCW LIMIT SWITCH

This command behaves identically to MOVE ABSOLUTE, but motor motion will stop if the V544 senses that the CCW limit switch is no longer actuated. For this command, the only legal direction of motion is clockwise.

### 5.5.9 COMMAND 8 : DECLARE ZERO POSITION

Execution of this command will clear all 32 ACTUAL position bits for the associated motor, effectively declaring that the motor's current position is the absolute zero reference for future motion commands.

### 5.5.10 COMMAND 9 : DECLARE ABSOLUTE POSITION

This command allows the user to assign an absolute angular position value to the current motor position. With the motor in the DONE state, load the two REQUESTED position registers with the desired values and issue the DECLARE ABSOLUTE command, with the "GO" bit set. The V544 will then set its ACTUAL position registers equal to the specified values, and then assert the DONE bit.

No motor motion results from this command. Execution of this command will generally result in the user-visible "logical" angular coordinate values becoming decoupled from the motor's "physical" detent-oriented position system by as much as a half

step.

### 5.5.11 COMMAND 10 : READ MOTOR THETA

This command permits the caller to read the "physical" motor position, namely the angular position of the motor relative to its physical rotor and stator structure and detent positions. The command results in 10 bits of data being returned in the motor auxiliary data register. The low 8 data bits represent the current microstep drive position (with all 0's meaning the motor is at a "full step" location) and the high two bits indicate the active quadrant of the motor's 4-step cyclic drive sequence. If the user desires the V544 positional system to be synchronized to the motor's physical (detent-indexed) step orientation, it is necessary to declare a motor zero position which corresponds to a full step position as indicated by the READ THETA command.

## 5.6 FINDING ZERO

At powerup, a stepper motor is considered by the V544 to be at position zero, but may actually be at some nonzero position relative to the reference system associated with the mechanism which the motor drives. It is generally necessary to align the stepper's coordinates with those of the physical environment.

### 5.6.1 ZERO USING THE INDEX

The user may issue a SEEK INDEX command to the V544, with a position specified such that the motor is likely to actuate the index sensor. The motor will decelerate to a stop when the sensor is hit, with the status INDEX bit true if the motor comes to rest with the index sensor still actuated. The user may then do a DECLARE ZERO or DECLARE ABSOLUTE command to initialize the V544 motor position to the known INDEX value. For highest accuracy, one may wish to make two approaches to the index sensor, the second seek performed at low velocity with a high LS DECEL value.

### 5.6.2 ZERO USING LIMIT SWITCHES

A user may choose to "run into" a limit switch and then DECLARE a position, perhaps after issuing a BACK OFF LIMIT SWITCH command.

### 5.6.3 ZERO USING HARD LIMITS

A user may initialize the motor's initial position by running up against a mechanical stop, if system conditions permit this (neither the V544 nor the motor itself will be damaged by sustained motion against a stop). Just program an "impossibly long" MOVE or a RUN, wait longer than the maximum possible transit time, stop the motor motion, and then DECLARE a position.

## 5.7 POWERUP/POWERDOWN CONSIDERATIONS

When an V544 module is powered down, connected motors will attempt to move to their nearest natural detent position. There is one such natural detent per step; these "poweroff detents" do not generally align with integral V544 positions or MOTOR THETA values.

On powerup the V544 assumes all motors are at POSITION = 0, THETA = 0, and are all "parked" at zero current, so no motor motion takes place. When any motor commands are issued (including IDLE), the motors will be powered up for at least one second (and longer if parking is disabled) and the motors will move such as to align with the THETA = 0 position. This may result in an unpredictable motor motion of as much as two full steps.

## 5.8 INTERRUPTS AND DETAILED TIMING

When any command is issued to a motor channel, users have the option to assert the IEN bit in the motor command register. IEN should be set in the same VME word write operation which loads the new command code and GO bit. The timing of a command execution is...

A. The V544 microprocessor continually scans all four motor channels.If any motor demon is still BUSY, the VME position and status registers are reloaded but the command register is ignored.

B. When a demon is not busy, the scanner checks to see if a new command plus the GO bit are loaded into the command register. If so, the motion parameter registers are read, the BUSY bit is set in the command register, and the command is initiated.

C. When a command is done, the DONE bit is asserted in the motor command register. If the IEN bit had been set when the command was originally issued, an interrupt will be generated.

Note that there may be a delay (up to several hundred microseconds) between the recognition of a new command (with GO set) and when the BUSY bit comes up. A command may then take anywhere from several hundred microseconds (for, say, IDLE) to as much as 4 years (for a long move) before DONE is asserted. If IEN is set and a non-zero interrupt level is selected, the interrupt will be asserted about 8 microseconds

after DONE is posted.

The V544 contains a single internal interrupt request flipflop. It is set by a DONE bit rising on any motor channel which executes a command with its IEN bit set. Even "non-motion" commands (such as DECLARE ZERO will interrupt if so programmed.

The V544 interrupt circuit operates in ROAK (release on acknowledge) mode, and reads the setting of the ADDRESS dipswitch (S1) as the value of the 8-bit interrupt vector. Note that ALL eight sections of S1 are used to form the vector, even though the LS section (section 8) is not used to determine the VME bus address of the module. The V544 follows VXI standards, wherein a device located at address V * 64 + 49154 (or V * 40h + C000h) generates an interrupt vector V.

Since all four motor DONE events logically OR to set the single IRQ flipflop, users should check all four motor channels for DONE whenever any interrupt ocurrs.

## 6. MICROPROCESSOR PROGRAM VERSIONS AND REVISIONS

## 7. ENGINEERING DRAWINGS

to make two approaches to the index sensor, the second seek performed a€ „ w … r Dp p Gp l 9Ï j Ï €²ß4 óâ[›
r l Lžr 0gt“p▯ è€▯ iþ™§-
\▯▯ !¦i▯ 4
7!Ï▯ ¦i¢i▯ jªi§-▯ \à 7!▯ 4▯ ▯ ▯ ▯ ▯ ▯▯ ▯@▯▯ ▯▯€ ‡ i ‰
W ‹ ÿÿ ÿÿ ÿÿ‘ ÿÿ“ ÿÿ• ÿÿ ÿÿ¯ ÿÿ± ÿÿÈ ÿÿ
▯ ÿÿ

□ ÿÿ¢i□□< □ □ ð □□< □ □ ð □

□ 7□ ÿÿ9□ ÿÿ;□ ÿÿR□ ÿÿT□ ÿÿV□ ÿÿX□ ÿÿZ□ ÿÿk□ ÿÿm□ ÿÿo□
ÿÿq□ ÿÿs□ ÿÿu□ ÿÿw□ ÿÿy□ ÿÿ{□ ÿÿ±□ ÿÿ³□ ÿÿ¶□ ÿÿ
□¶□ ß□ ÿÿá□ ÿÿ□ ÿÿ
ÿÿ" ÿÿ6 ÿÿh ÿÿŒ ÿÿŽ ÿÿ ÿÿÉ ÿÿï ÿÿ□□ ÿÿ8□ ÿÿ]
□ ÿÿƒ□ ÿÿµ□ ÿÿ ·□ ÿÿÙ□ ÿÿý□ ÿÿ □ý□ %
□ ÿÿU□ ÿÿ…□ ÿÿŸ□ ÿÿµ□ ÿÿ ·□ ÿÿÎ□ ÿÿú□ ÿÿ□□ ÿÿ8□ ÿÿ]
□ ÿÿ□□ ÿÿ□ ÿÿÍ□ ÿÿ÷□ ÿÿ.□ ÿÿU□ ÿÿy□ ÿÿ □ ÿÿ·□ ÿÿ □·
□ Ñ□ ÿÿò□ ÿÿ

ÿÿ)  ÿÿ@  ÿÿe  ÿÿ‹  ÿÿŽ  ÿÿ¦  ÿÿ¨  ÿÿª  ÿÿÌ  ÿÿñ  ÿÿ
▯ ÿÿ%▯ ÿÿF▯ ÿÿl▯ ÿÿ▯▯ ÿÿ»▯ ÿÿà▯ ÿÿ ▯à▯ â▯ ÿÿ▯ ÿÿ;
ÿÿ> ÿÿ@ ÿÿB ÿÿD ÿÿF ÿÿH ÿÿJ ÿÿL ÿÿN ÿÿP ÿÿ€
ÿÿ, ÿÿ… ÿÿ ÿÿŸ ÿÿ² ÿÿ´ ÿÿ ▯´ °
ÿÿ<
ÿÿ>
ÿÿÖ

ÿÿØ

ÿÿÛ
ÿÿÝ
ÿÿ3 ÿÿ5 ÿÿg ÿÿj ÿÿ›
 ÿÿ  ÿÿå  ÿÿü  ÿÿÞ  ÿÿN  ÿÿP  ÿÿ¨  ÿÿÌ  ÿÿ
 Ì  Î  ÿÿ   ÿÿ
    ÿÿI  ÿÿK  ÿÿ•  ÿÿ  ÿÿM  ÿÿO  ÿÿo  ÿÿƒ  ÿÿ´  ÿÿÞ
 ÿÿ
 ÿÿ2  ÿÿg  ÿÿ  ÿÿÏ  ÿÿ  ÿÿ2  ÿÿ
 2  f  ÿÿh  ÿÿ¯  ÿÿ±  ÿÿõ  ÿÿ÷  ÿÿB  ÿÿD  ÿÿ
€  ÿÿ«  ÿÿ-
 ÿÿ  ÿÿE  ÿÿ†  ÿÿÃ  ÿÿÅ  ÿÿ  ÿÿL  ÿÿN  ÿÿ   ÿÿ
    ß  ÿÿ
 ÿÿ  ÿÿ_  ÿÿž  ÿÿ½  ÿÿ¾  ÿÿÀ  ÿÿß  ÿÿá  ÿÿü  ÿÿ)
 ÿÿ+  ÿÿK  ÿÿM  ÿÿ´  ÿÿ¶  ÿÿ°  ÿÿ²  ÿÿw  ÿÿ
 w  z  ÿÿ¡  ÿÿ£  ÿÿ7  ÿÿ9  ÿÿ…  ÿÿÑ  ÿÿ !  ÿÿh!  ÿÿ'!
ÿÿ"!  ÿÿë!
ÿÿ<"  ÿÿ…"  ÿÿœ"  ÿÿž"  ÿÿö"  ÿÿF#  ÿÿ #  ÿÿÝ#  ÿÿ  Ý#
$  ÿÿi$  ÿÿ¬$  ÿÿ®$  ÿÿ %  ÿÿN%  ÿÿž%  ÿÿŠ%  ÿÿ©%  ÿÿú%
ÿÿH&  ÿÿ—&  ÿÿª&  ÿÿ¬&  ÿÿ '  ÿÿN'  ÿÿ '  ÿÿæ'  ÿÿ-
(  ÿÿU(  ÿÿ  U(  W(  ÿÿ¬(  ÿÿö(  ÿÿ<)  ÿÿŠ)  ÿÿÄ)  ÿÿÅ)  ÿÿÇ)
ÿÿ *  ÿÿc*  ÿÿ³*  ÿÿ +  ÿÿI+  ÿÿ-+  ÿÿ˜+  ÿÿê+  ÿÿ3,  ÿÿ~,  ÿÿÊ
,  ÿÿ -  ÿÿ  - 0-  ÿÿ2-  ÿÿ†-  ÿÿÕ-
ÿÿ .  ÿÿg.  ÿÿ°.  ÿÿö.  ÿÿ /  ÿÿ /  ÿÿw/  ÿÿ¾/  ÿÿ  0  ÿÿ?
0  ÿÿA0  ÿÿ‘0  ÿÿÞ0  ÿÿ.1  ÿÿy1  ÿÿÉ1  ÿÿ  É1  <2  ÿÿ=2  ÿÿ?
2  ÿÿc2  ÿÿe2  ÿÿ 3  ÿÿ 3  ÿÿ 3  ÿÿ 3  ÿÿn3  ÿÿ 4  ÿÿ.4  ÿÿ04  ÿÿ
S4  ÿÿU4  ÿÿ®4  ÿÿò4  ÿÿC5  ÿÿE5  ÿÿ-5  ÿÿ  -5  ²5  ÿÿ´
5  ÿÿÂ5  ÿÿÄ5  ÿÿ!6  ÿÿ«6  ÿÿ-
6  ÿÿß6  ÿÿá6  ÿÿ,7  ÿÿ×7  ÿÿÃ7  ÿÿ]8  ÿÿ_8  ÿÿ¯
8  ÿÿ 9  ÿÿ 9  ÿÿ 9  ÿÿ 9  ÿÿ 9  ÿÿ
 9  f9  ÿÿ²9  ÿÿø9  ÿÿú9  ÿÿ :  ÿÿ":  ÿÿW:  ÿÿY:  ÿÿŠ:  ÿÿÇ:  ÿÿ
;  ÿÿ ;  ÿÿB;  ÿÿD;  ÿÿ ;  ÿÿ„;  ÿÿÌ;  ÿÿÎ;  ÿÿ

< ÿÿ < ÿÿ < Q< ÿÿS< ÿÿ'< ÿÿ"< ÿÿÖ< ÿÿü< ÿÿ"> ÿÿ•>
ÿÿØ> ÿÿÚ> ÿÿ$? ÿÿ&? ÿÿq? ÿÿs? ÿÿ¢? ÿÿ¤? ÿÿÖ? ÿÿÙ?
ÿÿü? ÿÿþ? ÿÿ þ?
±@ ÿÿ¾A ÿÿÀA ÿÿlC ÿÿnC ÿÿnE ÿÿpE ÿÿ'F ÿÿÍG ÿÿÕG ÿÿ×G
ÿÿùH ÿÿûH ÿÿøI ÿÿûI ÿÿ,J ÿÿ.J ÿÿ4K ÿÿ6K ÿÿªN ÿÿ
ªN ¬N ÿÿ-O ÿÿ˜O ÿÿ¦Q ÿÿ¨Q ÿÿšR ÿÿ,T ÿÿ-
T ÿÿ/T ÿÿšW ÿÿœW ÿÿ¾W ÿÿÀW ÿÿãW ÿÿ
X ÿÿ9X ÿÿ;X ÿÿjX ÿÿlX ÿÿ
Z ÿÿ
Z Z ÿÿ Z ÿÿÇZ x îZ x ðZ ÿÿ [ ÿÿ
[ ÿÿ=\ ÿÿ?\ ÿÿí\ ÿÿð\ ÿÿ ] ÿÿ ] ÿÿõ] ÿÿ÷] ÿÿ ^ ÿÿ9
^ ÿÿc^ ÿÿu^ ÿÿ‡^ ÿÿ <‡^ ™^ ÿÿ«^ ÿÿ½^ ÿÿÏ^ ÿÿá^ ÿÿã^
ÿÿç_ ÿÿé_ ÿÿEa ÿÿGa ÿÿœb ÿÿŸb ÿÿÅb ÿÿÇb ÿÿòb ÿÿôb ÿÿŒe
ÿÿ f ÿÿæf ÿÿèf ÿÿ <èf h ÿÿ h ÿÿ"h ÿÿ
$h ÿÿÄh ÿÿÆh ÿÿíh ÿÿ-
i ÿÿRi ÿÿoi ÿÿ˜i ÿÿ¹i ÿÿ×i ÿÿøi ÿÿ j ÿÿ-
j ÿÿ/j ÿÿWj ÿÿ}j ÿÿ j ÿÿ <j ·
j ÿÿÞj ÿÿþj ÿÿ k ÿÿ9k ÿÿ;k ÿÿbk ÿÿ‡k ÿÿ‰k ÿÿÁk ÿÿèk ÿÿ
l ÿÿ%
l ÿÿCl ÿÿDl ÿÿFl ÿÿml ÿÿ'l ÿÿ²l ÿÿÌl ÿÿ <Ìl ól ÿÿ m
ÿÿ0m ÿÿNm ÿÿPm ÿÿwm ÿÿœm ÿÿ¼m ÿÿÖm ÿÿým ÿÿ n ÿÿ:n ÿÿXn
ÿÿZn ÿÿ n ÿÿ¢
n ÿÿÇn ÿÿên ÿÿ o ÿÿ2o ÿÿ <2o Wo ÿÿzo ÿÿ|o ÿÿšo ÿÿ¸o
ÿÿÖo ÿÿôo ÿÿöo ÿÿFp ÿÿGp ÿÿIp ÿÿDq ÿÿZq ÿÿ
\q ÿÿÜq ÿÿÞq ÿÿ‹s ÿÿŽs ÿÿÏs ÿÿ t ÿÿ <t 4t ÿÿ[t ÿÿ t
ÿÿ‰t ÿÿÖt ÿÿØt ÿÿŒu ÿÿŽu ÿÿCv ÿÿEv ÿÿÓv ÿÿÕv ÿÿ w ÿÿ w
ÿÿWw ÿÿ-
w ÿÿÁw ÿÿìw ÿÿ x ÿÿ x ÿÿ <x ðx ÿÿñx ÿÿóx ÿÿ y ÿÿ y
ÿÿ;y ÿÿ=y ÿÿæy ÿÿ†z ÿÿ^z ÿÿÌz ÿÿ { ÿÿK{ ÿÿ {ÿÿÑ{ ÿÿ |
ÿÿT| ÿÿ'| ÿÿÓ| ÿÿÕ| ÿÿ <Õ| B} ÿÿD} ÿÿb} ÿÿ } ÿÿž}
ÿÿ±} ÿÿÅ} ÿÿß} ÿÿü} ÿÿ ~ ÿÿ8~ ÿÿ^~ ÿÿ
€~ ÿÿ¦~ ÿÿÂ~ ÿÿÅ~ ÿÿç~ ÿÿé~ ÿÿ. ÿÿ0 ÿÿ <0 ~ ÿÿ
ÿÿ' ÿÿå ÿÿç ÿÿ € ÿÿ
€ ÿÿÅ ÿÿÇ ÿÿ , ÿÿY, ÿÿ¡, ÿÿé, ÿÿ1ƒ ÿÿyƒ ÿÿ{ƒ ÿÿT„ ÿÿ
V„ ÿÿá„ ÿÿâ„ ÿÿ <â„ ä„ ÿÿ … ÿÿ … ÿÿZ† ÿÿ\† ÿÿ
‡ ÿÿ ‡ ÿÿ ‡ ÿÿ<‡ ÿÿ>‡ ÿÿf‡ ÿÿ¢
‡ ÿÿà‡ ÿÿ"ˆ ÿÿ^ˆ ÿÿ¨ˆ ÿÿÝˆ ÿÿ ‰ ÿÿ ‰ ÿÿD‰ ÿÿ <D‰ F‰
ÿÿ‹Œ ÿÿîŒ ÿÿðŒ ÿÿä ÿÿ×Ž ÿÿÕ ÿÿØ ÿÿ ' ÿÿ ' ÿÿ:' ÿÿ<
' ÿÿ " ÿÿ " ÿÿØ" ÿÿÚ" ÿÿ ” ÿÿ ” ÿÿ›” ÿÿ ” ÿÿ <” Ò”
ÿÿÔ” ÿÿ$— ÿÿ&— ÿÿq— ÿÿs— ÿÿ ˜ ÿÿ
˜ ÿÿ0˜ ÿÿ2˜ ÿÿ
™ ÿÿ
™ ÿÿ4™ ÿÿ6™ ÿÿ˜š ÿÿšš ÿÿìš ÿÿ@› ÿÿŠ› ÿÿŒ› ÿÿ <Œ› é›
ÿÿ:œ ÿÿ€œ ÿÿÉœ ÿÿËœ ÿÿ ÿÿu ÿÿ¢ ÿÿ
¤ ÿÿú ÿÿEž ÿÿVž ÿÿXž ÿÿªž ÿÿôž ÿÿ%Ÿ ÿÿ'Ÿ ÿÿ‰
Ÿ ÿÿÍŸ ÿÿ ÿÿ <A ÿÿC ÿÿ_ ÿÿa ÿÿ ¡ ÿÿ ¡ ÿÿÊ¡
ÿÿË¡ ÿÿÍ¡ ÿÿò¡ ÿÿô¡ ÿÿ©¢ ÿÿ«¢ ÿÿþ¢ ÿÿ £ ÿÿL£ ÿÿN
£ ÿÿ›£ ÿÿ £ ÿÿê£ ÿÿ <ê£ ì£ ÿÿ×¤ ÿÿÙ¤ ÿÿË
¥ ÿÿ¼¦ ÿÿ¾¦ ÿÿ<§ ÿÿ>§ ÿÿ"¨ ÿÿ
$¨ ÿÿŠ¨ ÿÿ ¨ ÿÿº¨ ÿÿ¼¨ ÿÿX© ÿÿZ© ÿÿ © ÿÿ © ÿÿê© ÿÿì© ÿ
ÿ <ì© 4ª ÿÿ6ª ÿÿ ª ÿÿ ª ¯ÿÿÌª ¯ÿÿÎª ¯ÿÿ« ¯ÿÿ+« ÿÿ-
« ÿÿx« ÿÿz« ÿÿ'® ÿÿ)® ÿÿ& ÿÿ( ÿÿP ÿÿR ÿÿ ± ÿÿ ± ÿÿ

h±  ÿÿ <▯▯h±  i±  ÿÿk±  ÿÿ ±  ÿÿ ±  ÿÿc³  ÿÿe³  ÿÿÏ³  ÿÿÑ³  ÿÿ ´
 ÿÿ▯´  ÿÿ4´  ÿÿ6´  ÿÿ▯µ  ÿÿ▯µ  ÿÿ9µ  ÿÿ;
µ  ÿÿ▯¶  ÿÿ▯¶  ÿÿA¶  ÿÿC¶  ÿÿ <▯▯C¶  ▯·  ÿÿ▯·  ÿÿE·  ÿÿG·
  ÿÿÆ¸  ÿÿÈ¸  ÿÿÆ¹  ÿÿÇ¹  ÿÿÉ¹  ÿÿò¹  ÿÿô¹  ÿÿ£»
  ÿÿö¼  ÿÿø¼  ÿÿ▯½  ÿÿ▯½  ÿÿK¾  ÿÿM¾  ÿÿn¾  ÿÿp¾  ÿÿ <▯▯p¾  ¬À  ÿ
ÿ®À  ÿÿÓÀ  ÿÿÕÀ  ÿÿYÁ  ÿÿ[Á  ÿÿ~Á  ÿÿ€Á  ÿÿóÂ  ÿÿöÂ  ÿÿ#Ã  ÿÿ%
Ã  ÿÿ*Ä  ÿÿ,Ä  ÿÿ Å  ÿÿàÅ  ÿÿâÅ  ÿÿ▯Æ  ÿÿ▯Æ  ÿÿ▯Ç  ÿÿ <▯▯▯Ç  ▯Ç
ÿÿiÇ  ÿÿÚÇ  ÿÿÜÇ  ÿÿ/È  ÿÿ È  ÿÿËÈ  ÿÿãÈ  ÿÿåÈ  ÿÿ4É  ÿÿ^É  ÿÿ§É
 ÿÿ©É  ÿÿHË  ÿÿ_Ë  ÿÿbË  ÿÿXÌ  ÿÿZÌ  ÿÿ4Î  ÿÿ6Î  ÿÿ <▯▯6Î  ØÎ  ÿÿ
ÚÎ  ÿÿÜÎ  ÿÿ▯Ï  ÿÿ▯Ï  ÿÿ7Ï  ÿÿ9Ï  ÿÿ;Ï  ÿÿåÈ  ÿÿ4É  ÿÿ^É  ÿÿ§É  ÿ
ÿ©É  ÿÿHË  ÿÿ_Ë  ÿÿbË  ÿÿXÌ  ÿÿZÌ  ÿÿ4Î  ÿÿ6Î  ÿÿ <▯▯f  à=Ð/▯  8▯p
5 ▯▯
$ ▯8▯▯;Ð  8▯È(
                        ÿÿ~,  ÿÿÊ,  ÿÿ▯-  ÿÿ
▯  ¹Î  ▯  å  °Î  ÿ ÿÿÿÿ  ÿÿ▯.  ÿÿg.  ÿÿ°.  ÿÿö.  ÿÿ /  ÿÿ /  ÿÿ
w/  ÿÿ¾/  ÿÿ  0  ÿÿ?0  ÿÿA0  ÿÿ‘0  ÿÿÞ0  ÿÿ.1  ÿÿy1  ÿÿÉ1  ÿÿ
▯  ▯  4▯  ▯

□ ¼□ □ □ □ è□ )□ □ >□ ø
E)
½1

„8
W? ▯ yI ▯ S ▯ n
\ ▯ b ▯ ,g ▯ Äk ▯ Ço ▯ qx ▯ C~ ▯ b„ ▯ V ▯ ‹˜ ▯ K¡ ▯
¨ é° G¹ tÂ àÊ ÿÿx7 ÿÿÃ7 ÿÿ]8 ÿÿ_8 ÿÿ¯
8 ÿÿ▯9 ÿÿ▯9 ÿÿ▯9 ÿÿ▯9 ÿÿ▯9 ÿÿ ▯▯
Arial ÿÿø9 ÿÿú9 ÿÿ : ÿÿ": ÿÿW: ÿÿY: ÿÿŠ: ÿÿÇ: ÿÿ▯; ÿÿ
▯; ÿÿB; ÿÿD; ÿÿ ; ÿÿ„; ÿÿÌ; ÿÿÎ; ÿÿ

< ÿÿ < ÿÿ

**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