

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*

SMART DRIVE MANUAL ADDENDUM

New Features for Smart Drives Introduced with Firmware Versions 4.10 and 5.0:

(All version 4.10 features are also included in version 5.0 firmware)

This addendum contains documentation for new IDeal™ commands and various other new features that are not included in your Smart Drive manual due to their recent release. **Please save this documentation for future reference.** The new commands covered here are: CL (Current Limit), CT (Current Hold Time), RG (Registration), SQ (Square Root), GI (Go Immediate), GP (Go Point), GZ (Home to Z Channel), TF (Torque Fall Time), TM (Torque Mode), TQ (Torque Output), and TR (Torque Rise Time). WT (Wait) is included because it has been updated.

The CL /CT features are designed for use in clamping, pressing, spot welding, nut running, drilling and other applications which require torque/force, position and velocity control all at the same time. The RG feature is designed for applications that require position registration (labeling, cutting, etc). The GI feature is designed for applications that require command processing while a distance based move is being executed. The GP feature is designed for applications that require linear interpolation. The TM, TQ, TF, AND TR commands are designed for applications that require torque control.

The CL /CT commands can only be used with B8961/2 Smart Drives. At the time of this publication the usage of the CL/CT commands require the -CL option to be included with the B8961/2 Smart Drive. An example part number would be B8961-XXXXXXXX-CL. The X's in this part number could be replaced with other letters (A-P) which call out specific Opto 22 modules. An X in an Opto slot indicates that the slot is empty. The CL option specifies that DSP version 2.0.4 or later be included with the B8961/2 Smart Drive. This DSP has been specifically designed to accommodate the CL/CT features.

Please see the following pages for descriptions of new commands and features.

P/N PCW-4916
Revised: 12/97

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

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

Table of Contents

New Command Descriptions.....1
Commands Introduced with Firmware Version 4.10 1
CL (Current Limit) 1
CT (Current Hold Time) 3
GZ (Home to Z Channel) 3
RG (Registration) 4
SQ (Square Root) 5
Commands Introduced with Firmware Version 5.0 6
GI (Go Immediate) 6
GP (Go Point) 10
TF (Torque Fall Time) 11
TM (Torque Mode) 10
TQ (Torque Output) 12
TR (Torque Rise Time) 12
WT (Wait) 13

Configuring New Setup Parameters - CLAMP 14

New Inputs & Outputs.....15

Programming Your Application.....18
New Built-in Variables 18
Programming/Application Notes 19

New RS-232C Commands.....21

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

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

New Command Descriptions

Commands Introduced with Firmware Version 4.10:

CL — Current Limit …… syntax - CLn,n

Units: Amps or % Torque (Selected from the EDIT > SETUP > CLAMP menu)
Range: Unit scaling dependent

This command is only valid with the B8961/2 SmartDrives.

The CL command sets a current limit (in user units) to "clamp" the motor current to while executing a move. Reaching this current limit is dependent on the commanded move and the encountering of a load inducing a current rise in the motor windings equal to the CL value. If the current limit is reached in the move, the control switches from position mode to torque mode on the fly and maintains a torque level (or force level if an actuator is used) proportional to the CL value. If no load is encountered in a CL defined move that induces a current rise to the CL value, the move executes like a standard position based move.

The desired torque output from a motor can be determined by multiplying the torque constant of the motor (K_T) by the motor current limit set by CL. The following table gives the nominal K_T values of IDC motors. Due to variance in individual motor windings, the K_T value can vary ±10% from the nominal.

K_T Values for IDC Motors (oz.in/Amp)				
B23	**B32**	**B41**	**H3**	**H4**
57.6	99.2	187	54	67

CL is compatible with the DC (Distance to a Change) command (Please see the DC command description in your B8961/2 manual for more details on this command). Use of CL with the DC command allows the user to specify different current limits for different sections of the move profile. The current limit will change once the motor has traveled the distance specified by the DC command which precedes the CL command. This means that one section of the move can have the full peak torque available in order to do a fast accel, while another section of the move can have a lower current limit in order to lightly press or clamp a part. When defining both a new velocity (VE) and current limit (CL) after a DC command specify the VE value first**.** This will allow the new velocity to be compared to the *Break Velocity* when determining when a torque mode runaway condition is present.

Unlike conventional position based moves which are terminated when the commanded distance is reached (DA and DI based moves), or Mode Continuous (MC) moves which are done (command processing continues after the GO) when the commanded velocity is reached, a CL move is meant to be terminated by the CT (Current Hold Time) command or an ST (Stop on Input) command (See CT and ST command descriptions for more details).

There are, however, built in safety features in the B8961/2 Smart Drive to protect your machine by terminating a CL defined move when the commanded distance is reached while in torque mode, and terminating the move when the commanded move velocity (or break velocity, whichever is higher) is exceeded while in torque mode. These features are meant to prevent machine hard stops due to broken or mis-fed parts.

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

When a CL move ends, the drive's current limit is automatically reset to its default value (full rated peak torque of the motor) to provide full available torque for the next move. This allows for a 'quick return' stroke back to home using full available peak torque after a clamping or pressing operation.

WARNING - The correct Encoder Mode setting for operating CL/CT is **SERVO-CLOSED**. This is the default operating mode for all servos. CL/CT will not work properly in other Encoder Modes.

The following are some example programs demonstrating CL syntax and conventions:

Example #1: Basic CL move which terminates on an input. This will execute a normal DA move unless the CL current limit is achieved. The system will remain clamped until input #1 is asserted. Assume CL units are Amps.

VE1	Set velocity to 1 user unit
CL2.25	Set current limit to 2.25 Amps
ST1	Terminate "clamping" when input #1 is asserted
DA2.5	Move 2.5 user units
GO	Begin the move

Example #2: Basic CL move which terminates the clamping or pressing operation after holding the CL specified current for the time specified by the CT command. Assume CL units are Amps.

VE,1	Set velocity to 1 user unit on axis #2
CL,1.5	Set current limit to 1.5 Amps on axis #2
CT,1.2	Set "clamping" time for 1.2 seconds on axis #2
DA,2	Move 2 user units on axis #2
GO	Begin the move

Example #3: The CL command is compatible with the DC command. Assume CL units are amps. This move will not invoke the CL3 current limit until an absolute distance of 1.2 user units is reached. Therefore, there is no limit set on the motor current during the move from distance 0 to distance 1.2. This enables the motor to execute a fast accel without the CL command limiting the acceleration rate.

AC.05	Set Acceleration to .05 secs
VE1	Set velocity to 1 user unit
DA2	Move 2 user units
DC1.2 CL3	At an absolute distance of 1.2, invoke a current limit of 3 Amps
GO	Begin the move

Example #4: This program demonstrates programming with CL units of % Torque

{Move #1} CL50.5 ST1 DA2.5 GO DA0 GO
{Move #2} CL200 ST1 DA2.5 GO DA0 GO

Move #1 will clamp with a force of 50.5% of the motor's RMS current limit. For a B23, this would be 1.26 amps. For a B32, this would be 2.42 amps. Move #2 will clamp with a force of 200% of the motor's RMS limit. For a B23, this would be 5 amps. For a B32, this would be 9.6 amps.

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

CT Current Hold Time…….. syntax - CTn,n

Units: Seconds
Range: 0 to 99999.99 seconds

This command is only valid with the B8961/2 SmartDrives.

The CT command sets the time in seconds that a CL defined current is held. Once the time set by CT has been reached, the CL defined move terminates, and the control switches back to position mode automatically. Current Hold Time can be specified in increments of 0.001 seconds. The CT command can be changed as a function of distance traveled by using the DC command.

GZ Home to Z Channel ..... syntax - GZ±

This command will rotate the motor in the direction specified by ± until the encoder Z pulse is encountered. This command does not require home switches and is sometimes used in rotary applications to eliminate the need for a home switch.

The GZ command will home very slowly to ensure that the Z pulse is not missed.

Program Example: **GZ+ DA10 VE1 AC.1 GO**

GZ+	{Home to Z channel of the encoder, rotate motor in the positive direction to look for Z channel. After homing to the Z channel of the encoder, the position will be set to the Home Offset}
DA10	{Define absolute distance of 10 user units}
VE1	{Set velocity to one}
AC.1	{Set Acceleration to 0.1 second}
GO	{Start move}

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

RG

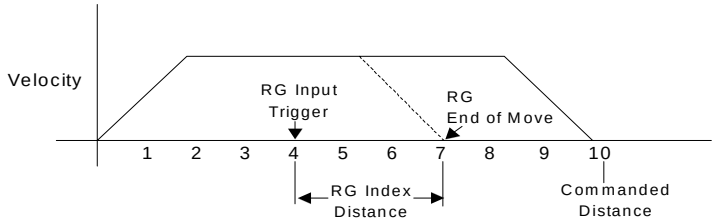
Registration .....syntax - RGn or RG,n

The Registration Command (RG) specifies a distance to be indexed from the current position - as commanded by a specific input trigger. The RG command **must appear after** a DA or DI in any program, and the RG parameter must be positive (+) regardless of the sign (+ or -) of the DA or DI command.

For example, in the program below an absolute distance of 10 user-units on axis #1 is commanded, the input trigger is received at user-unit 4, and the motor moves 3 user-units from the point where the input trigger was received. This results in a final position of 7 user units.

DA10 VE2 AC.1 RG3 GO

In the program above, assume the input was an optical sensor which triggered on a registration mark at a position of 4 user-units. The figure below shows the commanded move relative to what the registration move would be.



Accompanying the programmable Registration Command is the configurable Registration Input: G (also G in RS-232C Setup Commands). To configure a Registration input from the keypad, choose EDIT > SETUP > I/O > INPUTS. An input configured as a Registration Input will be designated by a G on the keypad input status display. The RG Command will only function if the corresponding input has been configured as a Registration Input (see note).

Note: Registration Input is only configurable on Input #1 for Axis #1, and on Input #2 for Axis #2.

System Performance when Using the RG (Registration) Command

There is a Capture Window (Position Lag) associated with the RG Command, which is a function of move velocity and the Position Capture Delay (reaction time) and can be calculated with the following equation:

Capture Window = RG Position Capture Delay * Velocity (Steps/Sec)

The Capture Window value is the number of steps accumulated between the falling edge of the Registration input and the time the current position is captured. Depending on the FPGA hardware version of your Indexer this RG Position Capture Delay will either be 164 µs or 5 µs.

The Registration Command is only available on firmware version 4.0 & higher. If you have FPGA version 5.9 or earlier, the Capture Delay will be 164 µs. If you have FPGA 6.7 or higher, the position is captured in hardware, and the only Capture Delay is the

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

input's opto-isolator (5 μs). Use the keypad's HELP key to determine your FPGA version.

Regardless of the FPGA version used, we have found both Capture Delays to be extremely repeatable. This leads to a very repeatable Capture Window (distance lag from when the registration input is made) that can be accounted for by decreasing your commanded registration distance by the Capture Window. For example, a motor traveling 240,000 steps/sec (30 rps with 8000 step/rev drive resolution) has a capture window of 39 steps for a 164 μs Capture Delay and 1 step for a 5 μs Capture Delay. Assuming the desired registration distance was 3 user units (and assuming 1 user unit is 8000 motor steps), an RG2.9795 would result in the motor traveling exactly 3 user units for a unit with a 164 μs Capture Delay.

SQ

Square Root…… syntax - SQn,(var)

Range: 0.0001 to 214748.3645

The **SQ** command calculates the square root of a number and returns the result in a user defined variable. The n parameter in the syntax can be a number or a variable parameter, however, the second parameter must be a previously defined variable for which the square root result is stored. If the second parameter is not a defined variable, you will get a **Bad Variable Name** error. Following mathematical convention, SQ will produce an **Invalid Parameter** error for negative n values. The return value is accurate to the 0.01 place.

Example: The following example program calculates the square root of 27.96 and stores the value in the user defined variable (SQRESULT).

Program: (SQRESULT)=0 SQ27,(SQRESULT)

The returned value in (SQRESULT) would be 5.28.

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

Commands Introduced with Firmware Version 5.0:

GI

Go Immediate……syntax - GI

The **GI** command begins a defined move profile in the same manner as the **GO** command. Unlike the **GO** command, where program execution waits until all defined moves have terminated, **GI** allows program execution to continue once the move has begun. This allows for other program defined processes to take place while an axis is moving, such as independent multi-axis moves, **OT** commands, and conditional **IF** blocks. One axis is not required to wait for another axis to finish a move before beginning its own move.

Following is an example of a program using the **GI command**:

VE1 DI20 GI MS1, "Axis #1 is moving" TD2

In this example, once the **DI20** move begins, program execution immediately displays the **"Axis #1 is moving"** message for 2 seconds. Once the **TD2** command has executed, the program will terminate; however, axis #1 will continue to move until the **DI20** distance is reached. A Stop, Kill, or press of the ESC key will halt a **GI** based move either inside or outside program execution.

The **GI** command can cause program execution and moves to be asynchronous. In order to re-synchronize the end of a **GI** move with program execution, use the **Wait** command and its new syntax, i.e. **WT#1** will wait for only axis #1; **WT,#2** will wait for only axis #2; **WT#1,#2** will wait until both axes have stopped.

If a program error occurs during a **GI** move, the move will stop at the Stop Decel Rate.

USING THE GI COMMAND - EXAMPLES
The following examples are provided to help further explain the use of the command:

A. If a **GI** move is in progress and an additional move is commanded on the same axis, the additional move will not begin until the **GI** move has completed. For example:

VE1 DA100 GI OT1,1 DA0 GI IF1,1 MS1,"All moves done" EB

In this program, one may expect to see the message "All moves done" immediately after the DA100 move begins. In reality, the program will wait at the DA0 GI until the DA100 move has completed before continuing. More simply stated, a move cannot be commanded to begin on an axis that is already moving.

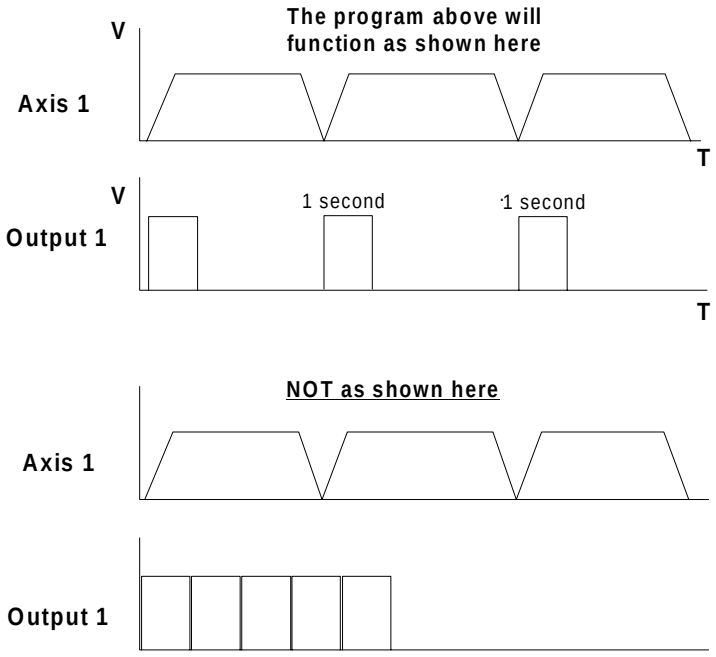
B. For multi-axis systems, **GI** allows moves within a move. For example:

VE1 DA100 GI LP5 VE,10 DI,5 GO EB

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

C. Since **GI** allows program execution to continue, there can be programming issues when using **GI**. For example, in the following program fragment:

LP VE2 DI10 GI OT1 TD1 OT0 EB

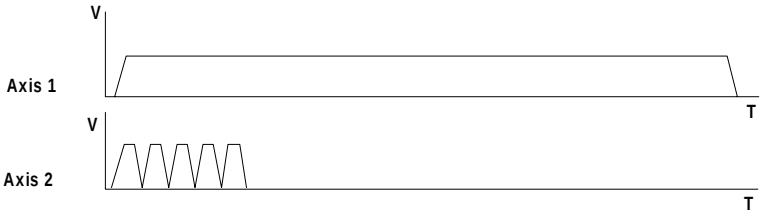


After the first pass through, the loop command (**LP**) will wait at the **GI** command since subsequent **GI** moves must wait for the present move to finish.

D. Another issue is programming conditional **GI** moves within **IF** blocks, for example:

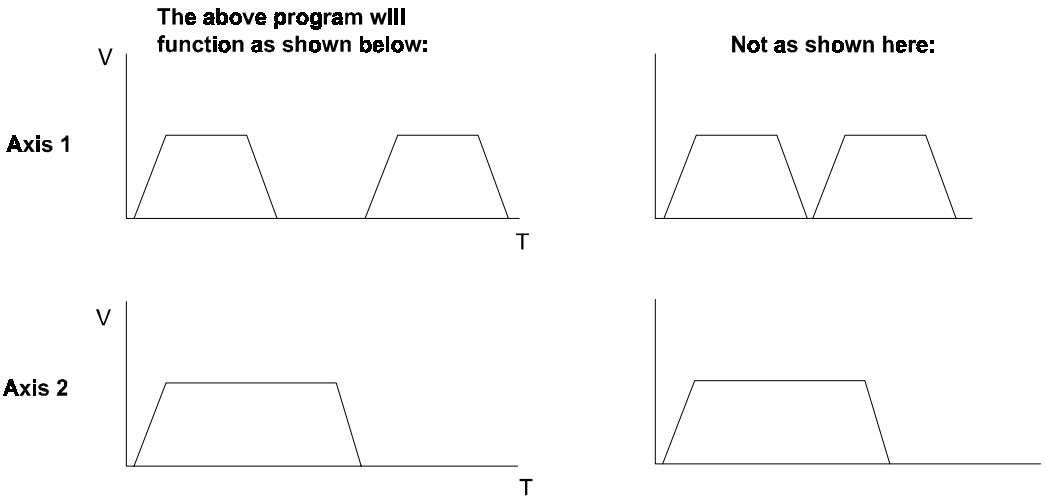
LP IF1,1 VE5 DI20 GI OTXX100 EB IF2,1 MC,+ VE,10 GO EB IF3,1 VE,0 GO EB EB

In this program, when input #1 is asserted, the **GI** move will be commanded, however, since the program will continue executing, input #1 may still be asserted on the next loop iteration. This will cause a second **GI** move to be commanded which may be undesirable. To avoid this situation, the addition of a **WT1,0** after the **IF1,1** will insure that input #1 is only seen once since it has to be deactivated to allow the program to continue.

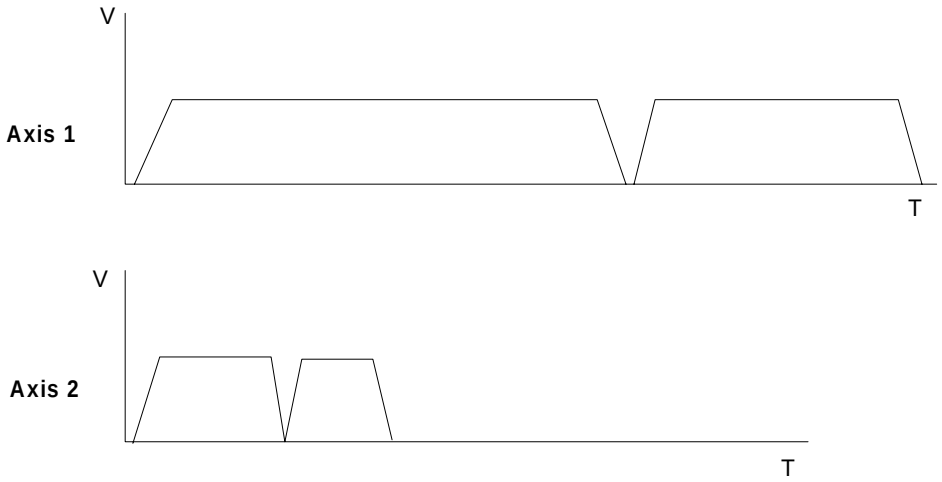


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

E. **DI1 GI DI,5 GO DI1 GI**



F. **DI10 GI DI,2 GO DI,2 GO DI10 GI**

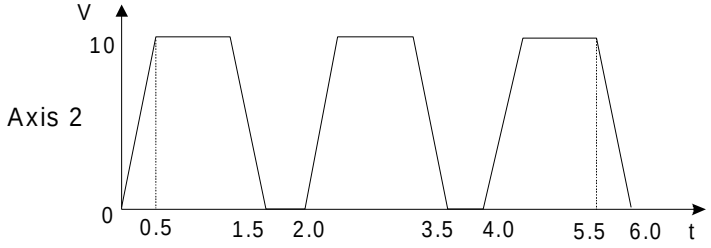


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

G.

```
(VAR1)=10                    {Initialize Variable}
LP                           {Beginning of Loop Block}
    DI55,10                  {Define Two Axis Move}
    VE10,10
    AC.5,.5
    GI                       {Start Go Immediate Move Both Axes}
    MS1,""                   {Clear Screen}
    MS1,(AI9)                {Write Analog Input 9 to the Screen While Moving}
    WT,#2                    {Wait for Axis 2 to Stop Moving}
    TD.5                     {Time Delay of 0.5 seconds}
    DI,10                    {Define Axis 2 Move of 10 units}
    GI                       {Start Axis 2 GO Immediate Move}
    IF8,1                    {If Input 8 is on}
        OT10,1               {Turn on Output 10}
        TD.1                 {Time Delay of 0.1 seconds}
        OT10,0               {Turn off Output 10}
    EB                       {End of If Block}
    WT,#2                    {Wait for Axis 2 to Stop Moving}
    TD.5                     {Time Delay of 0.5 seconds}
    DI,10
    GI                       {Start Axis 2 Go Immediate Move}
    (DIST)=(VAR1)*1000       {Do Variable Math while Moving}
    (TERM)=(DIST)            {Send Value of DIST Variable out of Serial Port}
    WT,#2                    {Wait for Axis 2 to Stop moving}
    OT2,1                    {Turn On Output 2}
    WT#1                     {Wait for Axis 1 to Stop Moving}
    OT1,1                    {Turn On Output 1}
    DA0,0
    GO                       {Move Both Axes Back to Starting Position}
EB                           {End of Loop Block, Restart Loop}
EN                           {End of Program}
```



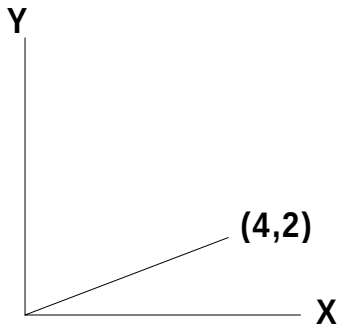


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

GP

Go Point (Linear Interpolation)……syntax - GP or GPn

The **GP** command allows the Smart Drive to execute a linear interpolated move. An example of the **GP** move is as follows:



The units and values for path velocity, acceleration and deceleration in **GP** moves are specified by the parameters traditionally defined for an axis #1 based move. As in regular moves, velocity, acceleration, and deceleration only need to be defined once. Each sequential **GP** move thereafter will use these values until new values are entered. The endpoint of the move is specified by a two axis **DA** or **DI** command which corresponds to the appropriate X and Y coordinates. The following program would execute the move in the picture:

VE2 AC.1 DA4,2 GP

The path velocity is 2 user units/sec, path acceleration is .1 sec and the X,Y position would be (4,2).

The **GPn** syntax is identical to the **GOn** command. **GPn** pre-calculates the move and waits for Input number "n" to activate before executing. For example, **GP4** would pre-calculate the move and wait for input 4 to be active before executing.

Although both axis move during a **GP** defined move, all **GP** parameters refer to the path movement rather than to the individual axis movements.

NOTES ON GP:

The largest possible **GP** moves are restricted to $X^2 + Y^2 <= (2^{31} - 1)^2$ in units of steps. For example, the longest simultaneous X,Y point move is:
Steps: **DA**1518500249,1518500249
User units with resolution 8000: **DA**189,812.5311,189,812.5311
User units with resolution 25000: **DA**60,740.0099,60,740.0099
Commanding moves larger than $X^2 + Y^2 <= (2^{31} - 1)^2$ will produce unpredictable results.

GP will work with any velocity and acceleration unit.

The **DC** command does not recognize an interpolated move as a 'single' move and will treat the axes independently. Therefore, using a **DC** during a **GP** move will cause unpredictable results unless the user has calculated the necessary values to preserve the vector move

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

TF

Torque Fall Time……..TFn,n

This command is only valid with the B8961/2 Brushless Servo Smart Drives.

The **TF** command specifies the time in seconds to ramp torque output down from a higher to lower value. If the drive has not been placed in torque mode by a **TM** command, the **TF** command is disabled. (Please see **TM** command documentation for more details).

The syntax for the command is **TFn,n** where the comma denotes the axis and the value **n** denotes the desired time in seconds of the torque fall ramp. The **TF** command is programmable in seconds only and does not support rates of change. If a **DA** or **DI** move is executed on one axis, while a **TQ** move is in effect on another, the **TF** and **TQ** parameters should be respecified on the next **TQ** command block issued.

An example of a typical program using the **TF** command is shown below:

TM1 TQ1 TR1 GO TD5 TQ0 TF0.5 GO

The **TM** command is required to place the drive in torque mode and enable the use of torque control commands. The **TQ1** command programs a constant torque output of 1 amp. The **TR1** command specifies a torque rise time to 1 amp of 1 second (Please see **TR** documentation for more details). The **TD5** command specifies a time delay of 5 seconds while a torque proportional to 1 amp is maintained. The **TF0.5** command specifies a fall time to 0 torque in 0.5 seconds.

The **TF** command cannot be used with **CL/CT** commands.

TM

Torque Mode……..TMn,n

This command is only valid with the B8961/2 Brushless Servo Smart Drives.

The **TM** command places the drive in torque mode where motor torque can be directly commanded. The syntax for the command is **TMn,n** where the comma denotes the axis. The values of **n** are defined below:

n = 1 Places the axis in torque mode and enables the torque control commands.

n = 0 Places the axis back to position mode and disables the torque control commands.

The **TM** command is used in conjunction with the torque control commands **TQ, TR,** and **TF**. These commands are analogous to the position mode commands **VE**, **AC** and **DE**. They determine the torque output level and rate of torque increase and decrease respectively. Please see documentation on these commands for further details. The **TM** command must be issued before any torque control commands become valid.

The **CL/CT** (Current Limit/Current Hold Time) commands are not considered torque control commands and represent an entirely different approach to torque/force control.

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

Please see documentation on the **CL/CT** commands for further details. It is important to note that placing the drive in **TM** mode does not invoke the features and safety protocols of the **CL/CT** commands.

TQ

Torque Output……..TQn,n

This command is only valid with the B8961/2 Brushless Servo Smart Drives.

The **TQ** command specifies the torque output of the motor when the drive has been placed in torque mode using the **TM** command (Please see **TM** command documentation for more details). If the drive has not been placed in torque mode by a **TM** command, the **TQ** command is disabled.

The syntax for the command is **TQn,n** where the comma denotes the axis and the value **n** denotes the desired torque output in units of *Amps* or *% Torque.* **TQ** units are defined under **EDIT-SETUP-CLAMP-UNITS** menu or by using the RS232 command **CU**. Please see the **CL/CT** documentation for details on torque units and restrictions.

An example of a typical program using the **TQ** command is shown below, assume units of *Amps*:

TM1 TQ0.75 GO WT1,1 TQ0 GO

The **TM** command is required to place the drive in torque mode, and enable the use of torque control commands. **TQ0.75** programs a constant torque output of 0.75 amps. When input #1 is asserted, **TQ0** brings the system down to 0 amps and 0 torque. Using

the **ESC** key, or issuing a STOP/KILL input or command, will also terminate a **TQ** move, and reset the drive to position mode.

Unlike the **CL/CT** commands, motor position is ignored while in torque mode, and more importantly, the features and safety protocols of CL/CT are not active. Therefore, if a **TQ** command is issued and the motor is not loaded or the load is “lost”, the motor will accelerate to a high velocity torque runway situation.

TR

Torque Rise Time……..TRn,n

This command is only valid with the B8961/2 Brushless Servo Smart Drives.

The **TR** command specifies the time in seconds to ramp torque output up from a lower to higher value. If the drive has not been placed in torque mode by a **TM** command, the **TR** command is disabled. (Please see **TM** command documentation for more details).

The syntax for the command is **TRn,n** where the comma denotes the axis and the value **n** denotes the desired time in seconds of the torque rise ramp. The **TR** command is programmable in seconds only and does not support rates of change. Like the **AC** command, **TR** will set both rise *and* fall ramps unless a separate torque fall time is programmed by a **TF** command (Please see **AC** and **TF** documentation for more details). If a **DA** or **DI** move is executed on one axis while a **TQ** move is in effect

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

on another, the **TF** and **TQ** parameters should be respecified on the next **TQ** command block issued.

An example of a typical program using the **TR** command is shown below:

TM1 TQ1 TR1 GO TD2 TQ0 GO

The **TM** command is required to place the drive in torque mode and enable the use of torque control commands. The **TQ1** programs a constant torque output of 1 amp. The **TR1** specifies a torque rise time to 1 amp of 1 second. The **TD2** command causes the motor torque to be held at a constant value for 2 seconds. The **TQ0** command will cause the torque to be ramped down to 0 amps in 1 second since there was no **TF** command issued.

The **TR** command cannot be used with **CL/CT** commands.

WT

Wait ..... syntax - see below

Syntax(s): WTn,xx...
WTxx... (assumes first input is input 1)
WT(AIn), expression

Range: n = starting input number, 1-16
x = 0; input high
x = 1; input low (grounded)
x = *anything but 1 or 0*; ignore the input level
expression = any valid expression as defined in the math and variables section.

This command waits for the specified condition to be true before continuing execution of a program. Either digital or analog input conditions may be used.

To increase flexibility the WT command allows you to use configured inputs in the expression. To help prevent this added flexibility from causing programming confusion, you can specify any character as an input (x). This allows you to self document your WT statements. For example, assume you configured input #3 as a "JOG SPEED" input. Programming like "WT01J10" can help remind you that you are already using input # 3.

Note: In order to synchronize program execution with the end of a **GI** move, there is new syntax associated with the **WT** (Wait) command. **WT#1,#2** will halt program execution until the respective axis has completed its move. **WT#1** will wait for only axis #1; **WT,#2** will wait for only axis #2; **WT#1,#2** will wait until both axes have stopped.

Example:

WT14,1 GO	Wait for input 14 to equal 1 before moving
WT12,010 GO	Wait for inputs 12-14 to equal 010 before moving
WT110 GO	Wait for inputs 1-3 to equal 110 before moving
WT(AI9)< 45000 GO	Wait for analog input 9 < 45000 before moving

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

Configuring New Setup Parameters - CLAMP

Configuring Clamp Units [CU]

EDIT > SETUP > CLAMP > **UNITS**

Default: Amps

You can choose between AMPS or % TORQUE as the units of the CL command. The valid parameter range for AMPS is 0 < X < (Rated RMS Current of Motor) and % TORQUE is
0 < *X* < 285%. The following table illustrates these ranges:

	B23	B32	B41	H3	H4
Rated RMS Current	2.5 Amps	4.8 Amps	4.6 Amps	2.5 Amps	5 Amps
Rated Peak Current	7.2 Amps	10 Amps	10 Amps	6 Amps	10 Amps
Max % Torque	285%	200%	200%	240%	200%

While programming CL in units of AMPS, if a number greater than the rated RMS current of the motor is commanded, the motor current is clipped to the motor's rated RMS current. Units of % TORQUE allow you to program currents above the RMS value of the motor. 100% corresponds to the rated RMS current of the motor. For example, if using a B32 motor the rated RMS current is 4.8 Amps, so a CL100 would command a 4.8 Amp clamping current from the motor. By entering values above 100% you can achieve current limit values above the motor's rated RMS current, however, clamping at values above the RMS value for extended periods of time will generate an RMS current fault. If a value over 285% is entered (2.85 times the motor's RMS rating) or [(% TORQUE)*(Rated RMS Current)] > [Rated Peak Current], the current to the motor is clipped to the Rated Peak Current.

1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to select AMPS or % TORQUE.

Configuring Break Velocity [BV]

EDIT > SETUP > CLAMP > **BRKVEL**

Default: 5 {Velocity Units}

This feature inhibits a torque mode runaway condition if the load is "lost" while at the CL limit (i.e. the object clamped breaks or slips). For moves where the break velocity is greater than the move velocity, the break velocity will be used as the trigger point where the control switches from torque mode to position mode. After this switch takes place, the motor will decelerate to the commanded move velocity. For moves with a higher move velocity than the break velocity setting, the **move velocity** will be used as the trigger point where the control switches back to position mode from torque mode. This feature allows an actuator to press smoothly at low speeds where stiction (bouncing back and forth between static and kinetic coefficients of friction) is present while still having torque mode runaway safety control.

Caution: It is recommended to set the break velocity only slightly higher than the move velocity. Too high a break velocity setting can cause the control to jump in and out of torque mode if the break velocity is exceeded while executing a CL defined move.

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

1. Use ← and → keys to select an axis.
2. Use the numeric keys to set the break velocity in VE units.

New Inputs & Outputs

Configuring Input Definition [ID]

EDIT > SETUP > I/O > **INPUTS**

Default: UUUUUUUUUUUUUUU

See Configuring Your Inputs & Outputs (I/O) in your SmartDrive manual for more details on input definition.

1. Use ← and → keys to select an Input. The function of the highlighted input will be displayed on the top line.
2. Once your cursor is on the desired input, use ↑ ↓ to select F (SET CL FORCE 1) or f (SET CL FORCE 2).

New Input Character Descriptions

F,f Set CL Force (F specifies axis 1, f specifies axis 2)
In addition to being able to set a clamping current via the CL command, the user can also set the clamp current on the fly based on an input. When the SET CL FORCE input is asserted during a CL defined move, the control will maintain the torque the motor is producing at that instant. For example, if the following move is executing: VE1 CL3 DA3 GO (CL units are Amps), and a SET CL FORCE input is activated when the motor current is at 1.5 amps, the current will be clamped to 1.5 amps. If this input is not triggered, the control will clamp the current to 3 amps. The SET CL FORCE input requires a valid CL move to be in progress otherwise the input is ignored. The SET CL FORCE input is intended for applications clamping to load cell feedback.

I Interrupt (Run 98)
When activated, motion on all axes is stopped at the stop-rate (see Edit-Setup-Misc-Stop-Rate). The current program is stopped, and processing continues with the first command in program 98. If no program is running when the input is activated, program 98 will run. This input is ignored while the keypad is in Edit mode. This is a positive edge sensitive input, rather than a level sensitive input. If multiple inputs are configured as Interrupts, only the first edge of the first activated input will be seen. If subsequent Interrupt inputs go active while the first Interrupt input is active, no additional interrupts will be seen.

Advanced Interrupt handling can be achieved using the (INT98CTRL) and (ARM INT98) variables. The (INT98CTRL) variable determines whether Interrupts can be disabled or not. The (ARM INT98) variable allows you to arm and disarm the Interrupt as desired.

When the Indexer powers up (INT98CTRL) is initialized to 0. In this mode, every interrupt results in an immediate jump to program 98, even if you just entered program 98. This means that it is possible for the interrupt service routine to be interrupted by another interrupt input. This functionality is backwards compatible with earlier versions of firmware in IDC SmartDrives. The value of (ARM INT98) is ignored.

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

When (INT98CTRL)=1 you can enable and disable Interrupts at will with the (ARM INT98) variable. Setting (INT98CTRL)=1 also initializes (ARM INT98) to 1. This means the control is watching for interrupts. When (INT98CTRL) is set to 1 an interrupt causes the program to
jump to program 98 **AND** sets (ARM INT98)=0, disabling any further interrupts until you re-enable them by setting (ARM INT98)=1. This allows you to control when you want to re-enable Interrupts in your interrupt service routine (program 98).

To summarize, when (INT98CTRL)=1:

If (ARM INT98)=0
Interrupts are ignored. (ARM INT98) is automatically set to 0 by the Smart Drive on the first edge of the input, if the previous (ARM INT98) value was 1. Interrupt processing will be suspended until (ARM INT98) is reset to 1. This allows for input debouncing and gives the user control over the ability of program 98 to interrupt itself.

If (ARM INT98)=1
The system is awaiting the first INT98 input assert edge. Once the interrupt is seen the control will go to program 98. Subsequent interrupts are ignored until (ARM INT98) is set to 1.

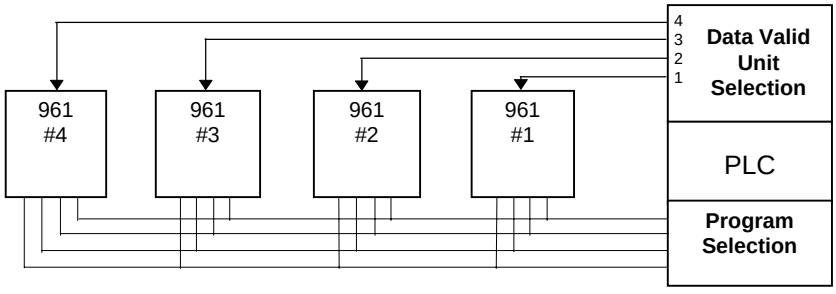
(INT98CTRL) and (ARM INT98) are reset to default values on power-up.

Note: There is a space in (ARM INT98).

V Data Valid

When this input is configured, it determines if the Binary/BCD program select lines are processed or ignored. If the input is active, program select lines are processed, otherwise they are ignored. This allows applications to be wired in a pseudo-bus architecture fashion with each unit sharing the same program select lines, and the data valid inputs determining which units should listen. Configuring this output can greatly reduce panel wiring.

In the example shown below, using the Data Valid input reduced the number of wires required for selecting programs by one-half.



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

Configuring Output Definition [OD]

EDIT > SETUP > I/O > **OUTPUTS**

Default: PPPPPPPP--------

See Configuring Your Inputs & Outputs (I/O) in your SmartDrive manual for more details on output definition.

1. Use ← and → keys to select an Output or OPTO output channel. The function of the highlighted output will be displayed on the top line.
2. Once your cursor is on the desired output, use ↑ ↓ to select K (AT CL LIMIT 1) or k (AT CL LIMIT 2).

New Output Character Descriptions

K,k At CL Limit (K specifies axis 1, k specifies axis 2)
The AT CL LIMIT output will become active when the motor current reaches the value defined by CL in the user's program. This output allows the Servo SmartDrive to indicate to a PLC when it has begun clamping, so that the user may synchronize the clamping operation with other processes in the application (for example drilling, cutting, or milling the clampedpart).

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

Programming Your Application

New Built-In Variables

The following are new pre-defined variable names that can be used in torque control applications.

Variable Name	Description	Type
(CLSTAT1), (CLSTAT2)	Value of CL termination code	Read only
(TORQLIM1), (TORQLIM2)	Displays/sets the current torque output limit of the amplifier	Read and Write

(CLSTAT1) axis 1, (CLSTAT2) axis 2
The CLSTAT variable is a status variable which reports how a CL based move terminated. The following table shows the possible values of CLSTAT and their meaning:

CLSTAT Value	Description
0	Never clamped, reached the commanded move distance
1	Clamped, stopped on a ST input
2	Clamped, stopped on a CT time out
3	Clamped then exceeded commanded move velocity or break velocity (whichever is greater) causing control to switch back into position mode
4	Clamped then exceeded commanded move distance

These status variables are useful in pressing applications (for example, pressing bearings on a journal) where part variances could effect how well two parts are pressed together. The user commands the following clamping move to repeatedly press bearings onto journals coming down an assembly line:

LP WT1 ST2 DA5 VE2 AC1 CL1.125 CT5 GO EB

In the above program the control waits for input #1 to go active before proceeding to clamp. If input #2 is ever activated during the clamping operation, the move is aborted. If both parts are within tolerance, the CL defined motor current is reached (1.125 amps), and the move terminates after the current has been held for 5 seconds. If part tolerances are too loose, the CL value is never reached and the move will terminate at the specified move distance (5 inches). If the part breaks during clamping, the motor will accelerate up to the defined move velocity, 2”/sec, and then will switch back into position mode and finish the move after the actuator has traveled 5 inches. An external event could also occur where a premature exit of the clamping operation is desired. This can be accomplished by activating input 2 which halts the clamping operation via the ST2 command in the program. Depending on how the clamping operation ended the CLSTAT value is set to the corresponding termination value.

(TORQLIM1) axis 1, (TORQLIM2) axis 2
The TORQLIM variable is for positioning applications that require a safety limit to be set on motor torque and where it is undesirable to switch to torque mode when this limit is reached. This variable allows the user to set the maximum motor current available from within a program and stay in position mode when this current is reached. The units of TORQLIM are Amps. For example, a sample program is:
(TORQLIM1)=2 DA1.125 VE3 AC1 GO

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

The above program sets the maximum motor current equal to 2 Amps. If the motor encounters a load while executing the above move that forces a current rise in the motor windings equal to 2 Amps, the motor current will be clipped to 2 Amps. However, the control will **not** switch into torque mode when 2 Amps is reached, and will instead stay in position mode.

This torque limiting feature is useful for pressing applications where the user would like to press a part with a set amount of force until a distance is reached. For example, if the user is pressing bearings onto a journal it may be desirable to press the part with a motor torque proportional to 2 Amps of current until a distance of 1.125" is reached (see the example program above).

Due to motor current (and thus available torque) being limited by the TORQLIM variable it is more likely that the user will get a Following Error when executing a move. There are two ways to solve this problem: increase the Following Error setting for the control (found in EDIT>SETUP>ENC>FOL-ERR) or increase the value of the TORQLIM variable. If neither of these solutions is appropriate for the application, it is recommended to use the CL/CT features of the product and switch into torque mode when the current limit is reached.

When TORQLIM is set to a value, the control writes the value to RAM. This value is kept in RAM until TORQLIM is redefined or until the power is cycled. The TORQLIM value is reset to the motor default when power is cycled.

Programming/Application Notes - CL/CT

- Usage of CL/CT with the DC command allows the user to maximize throughput while allowing the actuator to encounter the load at a relatively low velocity before current limiting. This minimizes impact forces and prolongs the life of the mechanics being used. After a CL defined move is complete, the available motor torque is reset to the default value of the motor to allow for an aggressive 'quick return' stroke. An example of a clamping move with a quick return stroke is:

  DA10 VE20 AC.1 DC7 VE2 DC8.2 CL2.1 CT7 GO DA0 VE30 GO

  This program commands a move distance of 10 inches, an acceleration/deceleration of 0.1 seconds, and an initial move velocity of 20 inches/sec. Since no CL value was specified before the first DC command, the full available motor torque is present to accelerate the motor up to 20"/sec in 0.1 seconds. Once the motor has reached a distance of 7 inches, the motor decelerates to a speed of 2"/sec in 0.1 sec.

  Note that no CL value was specified after the DC7 command, so full motor torque is available for decelerating the motor from 20"/sec to 2"/sec during this section of the move. While decelerating from 20"/sec to 2"/sec, the motor travels 1.1". This means that we have reached 8.1" (absolute distance from zero) when we start running at a constant speed of 2"/sec. At a distance of 8.2" the motor current is limited to 2.1 amps by the CL2.1 command. While running at 2"/sec a load is encountered and the motor current is clamped to 2.1 amps, and held for 7 seconds. After the clamp is complete the motor makes a quick return move back to the zero position.

- After completing a clamping, pressing, or fixturing operation the drive will switch from torque mode back into position mode. This switch typically causes some ringing in the motor's position as the drive is now servoing to position as opposed to torque.

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

- CL/CT commands are valid over RS-232C. This allows for a computer to stream down current limiting moves (CL/CT moves) via Terminal in 'hosted mode' without having to define a program. For example: CL5 CT.2 DI2.25 VE1 AC.3 GO is a valid set of move commands from terminal.

- When the distance commanded by a CL defined move is reached while in torque mode the B8961/2 will switch out of torque mode and back into position mode and subsequently bring the motor to a stop at the programmed decel rate. This can occur in pressing applications. This switch into position mode will result in a final position that exceeds the commanded move position. The switch from torque mode to position mode when the move distance is met is intended as a safety feature, and is not meant to bring the motor to a stop at the commanded move distance.

- Pressing the ESC key or activating a Stop, Kill, or Interrupt (Run 98) input when the system is clamped will abort the move and program as usual.

- When executing loops, CL/CT commands must appear within the LP block and within the GO block in which they are intended to be active. For example:

  LP DI5 VE2 CL3 CT.5 GO EB {valid syntax, CL/CT defined within LP and GO block}
  CL3 CT.5 LP DI5 VE2.3 GO EB {invalid, CL/CT not defined within LP and GO block}

- Two axis moves that include clamping behave just like any other two axis move: both moves must be "complete" before program flow continues after the GO command. This means that if Axis 1 has finished an indexing move, and Axis 2 is in torque mode waiting for the CT time to expire, Axis 1 must wait for Axis 2 to finish before Axis 1 can move again.

- It is recommended to slow the motor down to a low speed before encountering the load. This will minimize impact forces the mechanics are subjected to by the application, thereby prolonging their life.

- An IDC R-Pack may be required with the B8961/2 to dissipate the regenerative energy caused by CL/CT applications. The need for an R-Pack in your application will be indicated by your B8961/2 faulting while performing a move, and displaying an Overvoltage/Overcurrent Fault on the keypad display. If you do not possess a keypad, you can detect the nature of the fault by using the SA and SD immediate RS-232 commands (Please see the Immediate RS-232 Status Command section of your B8961/2 manual).

- The Force output from a screw-driven actuator while it is moving can be determined from the following equation: $F = \frac{2\pi}{16} Tm \times p \times GR \times eff_S \times eff_{GB}$, where F is force in units of lb(f), Tm is motor torque in units of oz-in, P is the pitch of the leadscrew (revs/in), GR is the gear ratio, eff_s isthe efficiency of the leadscrew, and eff_{GB} is the efficiency of the gearbox. The Force output from a screw-driven actuator that is stationary is: $F = \frac{2\pi}{16} \times Tm \times GR \times P$ (units same as above).

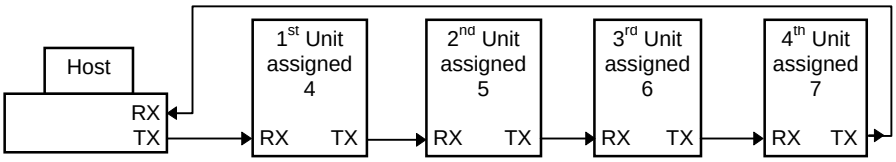
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

New RS-232C Commands - Introduced in Firmware Version 4.10

Command	Name	RS-232C Syntax

AA Auto Address AA or AAn

The AA command automatically addresses Smart Drives in a daisy chain. It assigns a device address to each unit on the daisy chain. This allows the units to be wired in a daisy chain without setting each unit's address manually. The AA command parameter n indicates the value in which the addressing sequence will begin.



In the example above, the Host issues an **AA4** and the units are addressed 4, 5, 6, 7. This offers the convenience of adding a new unit anywhere in the daisy chain without manually re-addressing all the other units. Just connect the new unit, issue an AA command from the new unit with the address of the new unit as the AA parameter , i.e. AA*n.*

BV Break Velocity nBVr,r

Example: BV3.6,2.0 (This sets Axis 1 break velocity to 3.6, Axis 2 to 2.0 in units selected by the VU command.)

CU Clamping Units nCUi,i

	Units
i=0	Amps
1	% Torque

Example: 3CU1,0 (this sets the Clamp Units for device number 3 to be % TORQUE for Axis 1, and Amps for Axis 2)

GI Go Immediate GI

Note: When a **GI** based move is running outside a program, RS232 commands can be executed immediately without the need for them to be buffered until the move is complete.

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 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