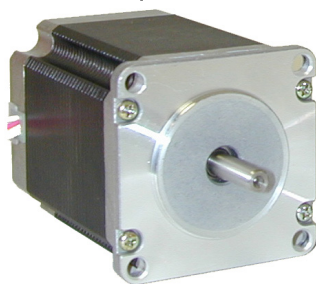
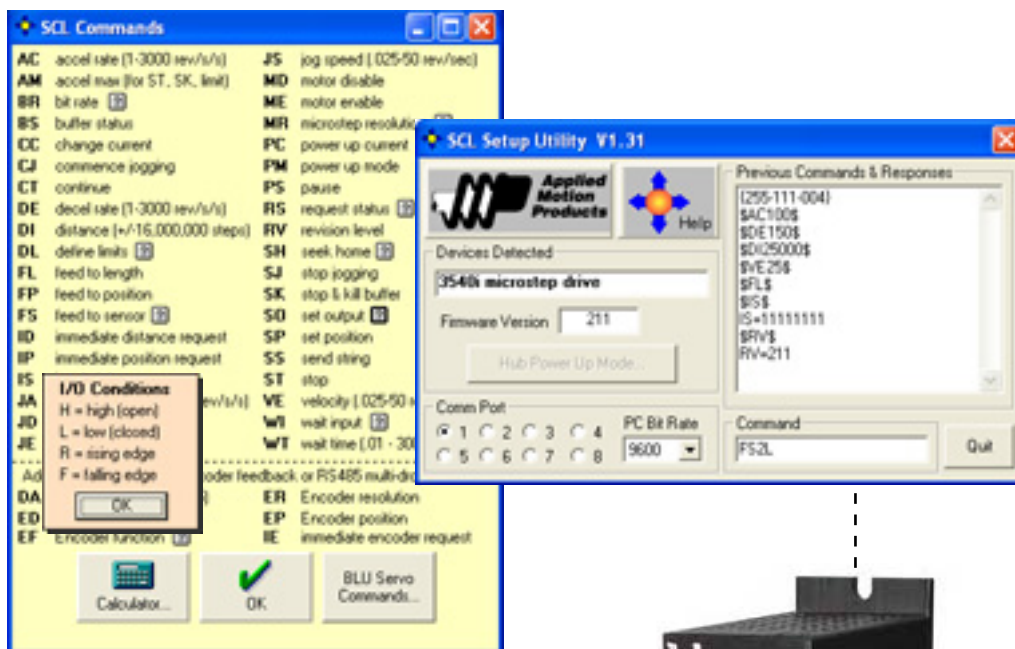


Si™ Command Language (SCL) Software Manual



for
1240i
3540i
7080i
BL7080i
BLSi7080
Si-100
Si2035
Si3540
Si5580

Contents

Getting Started.....	5
Connecting to your PC	5
Introduction	6
Communication Protocol	7
Power Up Signature	7
Using SCL with the SiNet Hub	7
Buffered or Immediate?	9
About the Buffers	9
Registers	10
Idle Current Reduction	10
Analog Inputs	10
Command Summary	11
Command Descriptions.....	12
AC – Acceleration	12
AM – Acceleration Maximum	12
BR – Bit Rate	12
BS – Buffer Status	12
CC – Change Current	12
CI – Change Idle Current	13
CJ - Commence Jogging	13
CS – Change Jog Speed	13
CT – Continue	13
DA – Define Address	13
DC - Distance till Change	13
DE – Deceleration	14
DI – Distance/Position	14
DL – Define Limits	14
ED – Encoder Dead Band	14
EF – Encoder Function	14
EP – Encoder Position	15
ER – Encoder Ratio	15
FC – Feed to Length with Speed Change	15
FD - Feed to Double Sensor	15
FL – Feed to Length	16
FM – Feed to Sensor with Mask Distance	16
FO – Feed to Length and Set Output	16
FP – Feed to Position	17
FS – Feed to Sensor	17
FY – Feed to Sensor with Safety Distance	17
HW – Hand Wheel	17
IA - Immediate Analog	18
ID – Immediate Distance	18
IE – Immediate Encoder	18
IH – Immediate High Output	18
IL – Immediate Low Output	18
IP – Immediate Position	19
IS – Input Status	19
JA – Jog Acceleration	19
JD – Jog Disable	19
JE – Jog Enable	19
JS – Jog Speed	19
MD – Motor Disable	20
ME – Motor Enable	20
MR – Microstep Resolution	20
PC – Power on Current	20
PM – Power on Mode	21
PS – Pause	21
RA - Read Analog	21
RS – Request Status	21
RV – Revision Level	21
SA – Save All	21
SH – Seek Home	22
SJ – Stop Jogging	22
SK – Stop & Kill	22
SO – Set Output	22
SP – Set Position	22
SS – Send String	22
ST – Stop	22
VC – Velocity Change	22
VE – Velocity	23
WI – Wait for Input	23
WT – Wait Time	23

Getting Started

Thank you for purchasing an Applied Motion Products Si™ Product. We hope you will find that the performance, price and ease of use make our products the best value for your application.

The Si™ Command Language (SCL) is used in several Applied Motion products, including the 1240i, 3540i, 7080i, BL7080i, BLSi7080, Si3540 and Si5580 indexer-drives and the Si-100 indexer. This manual explains how to use SCL to command a drive from a real time host such as a PC or PLC. ***If you are planning to use SCL with a BLu servo drive, there is a separate manual for that.***

If your application calls for a program to be stored in the drive and executed without a host computer, then you will want to use the *Si Programmer™* software. There is a separate manual for that.

For information regarding your specific Si™ hardware, such as wiring and mounting, please read the hardware manual that came with that product.

Using the *Si™ Command Language* requires three steps:

1. You must get the drive into SCL mode (see *Power Up Signature*, page 7).
2. You must learn to use the SCL commands. We highly recommend using the SCL Setup Utility for steps 1 and 2.
3. Develop an application program on your host computer.

If you have trouble getting your Si™ Indexer to meet your expectations, or if you want to suggest improvements to the product or this manual, give us a call at (800) 525-1609. Or, you can fax us at (831) 761-6544. If you are not calling from North America, dial (831) 761-6555.

Connecting to your PC

- Locate your computer within 6 feet of the Si™ hardware.
- Your Si™ product was shipped with a black adapter plug. It has a telephone style jack at one end and a larger 9 pin connector at the other. Plug the large end into the COM1 serial port of your PC. Secure the adapter with the screws on the sides. If the COM1 port on your PC is already used by something else, you may use the COM2 port for the Si™ Indexer. On some PCs, COM2 will have a 25 pin connector that does not fit the black adapter plug. If this is the case, and you must use COM2, you will have to purchase a 25 to 9 pin serial adapter at your local computer store.
- Your Si™ Indexer was also shipped with a 7 foot telephone line cord. Plug one end into the adapter we just attached to your PC, and the other end into the PC/MMI jack on your Si™ Indexer.

Never connect the Si™ Indexer to a telephone circuit. It uses the same connectors and cords as telephones and modems, but the voltages are not compatible.

You may also need to set the COM port in the Windows software. When the software is loaded, it looks for the first available COM port, but doesn't always find the one you've plugged into.

You can choose the port yourself by clicking on one of the "COM port" option buttons. If the port exists and is not already in use, the programming software will use it to communicate with the Si™ Indexer.

*Always apply power to Si™ hardware **after** the SCL Setup Utility™ software is running on your PC.*

Introduction

The Si™ Command Language (SCL) provides a means to control Si™ indexers and indexer-drives from a host PC or PLC. SCL allows the host to command the drive to perform a variety of motion and I/O tasks, and to provide the host with status information.

Individual drives can be directly connected to the serial port of a PC or PLC, or up to 8 drives can be connected to single port using the SiNet™ Hub-8. Up to 30 drives with the Multi-drop RS485 option can be connected to one RS485 or RS422 port.

The SCL mode firmware coexists in the drive with the *Si Programmer™* firmware. The drive determines the proper mode of operation by detecting a host signature at power up. The Windows-based *Si Programmer™* software automatically supplies the necessary signature, invoking *Si Programmer™* mode. Please see the section “Power Up Signature” for details.

SCL requires previous programming experience and custom application software. It is intended for systems developers who have the tools and knowledge to put together an application program that sends commands over a serial port.

One thing you cannot do with SCL is write a program to be stored in the drive. If you want to do that, use the *Si Programmer™* software. However, SCL provides a 128 character command buffer that allows the host to send commands ahead of time, while other commands are running.

Si™ Command Language provides more than 30 instructions. All commands begin with two upper-case ASCII characters. 9 of the commands are the same as our Windows-based Si™ language:

Feed to Length (FL)	Feed to Sensor (FS)	Feed to Position (FP)
Set Position (SP)	Seek Home (SH)	Wait for Input (WI)
Wait Time (WT)	Set Output (SO)	Change Current (CC)

Two additional instructions implement options of the Si™ Change Current instruction:

Motor Enable (ME)	Motor Disable (MD)
-------------------	--------------------

Another 4 commands are used to set up move parameters:

Accel (AC)	Decel (DE)	Velocity (VE)	Distance (DI)
------------	------------	---------------	---------------

7 commands configure the global parameters seen on the left-hand side of the *Si Programmer™* screen:

power up current (PC)	microstep resolution (MR)	define limits (DL)	
jog accel (JA)	jog speed (JS)	jog enable (JE)	jog disable (JD)

7 commands ask the drive for status information:

buffer status (BS)	input status (IS)	request status (RS)
revision level (RV)	immediate distance (ID)	immediate position (IP).

Power up mode (PM) sets the drive to power up in SCL mode, or in the factory default “auto detect mode.”

Send string (SS) tells the drive to send a text string back to the host. This is useful for detecting when a prior buffered command, typically a move, has completed.

Pause (PS) suspends execution of buffered commands until the continue (CT) command is received.

Stop (ST) stops a command in mid-execution. This is useful for killing an errant move, wait time or wait input instruction, especially during the debug phase of a program.

Stop and Kill Buffer (SK) stops any buffered command and removes all other commands from the buffer.

RS485 compatible drives are equipped with a Define Address (DA) command so that each drive can be assigned a unique address.

Drives with the encoder feedback option provide four commands for defining the encoder characteristics and functionality.

Communication Protocol

Communication between drive and host is 9600 baud, 8 data bits, one stop bit, no parity. Each command must be terminated with a carriage return (ascii 13). Drive does not echo received characters to the host. No handshaking is required. Newer firmware includes a bit rate (BR) command that allows the host to increase the communication speed.

RS232 connection is a three wire type: transmit, receive and ground. Use the cable supplied with your *Si™* drive.

RS485 connection is a five wire type: transmit+, transmit-, receive+, receive- and ground. You must provide your own cabling (Category 5 style recommended). Refer to your drive's *Hardware Manual* for more detail.

Power Up Signature

To invoke SCL mode, the user's program must detect power up of the drive and supply the signature "00" within two seconds, but not sooner than 2 milliseconds. If this is inconvenient, the drive can be set to automatically wake up in SCL mode using the PM2 command. The *SCL Setup Utility* provides an easy way to configure the power up mode of the drives and/or hub. It also and gives the user a convenient way to try out commands and gain familiarity with SCL.

When an *Si™* drive is set for "auto-detect" mode, it sends three characters to the host when power is first applied. The first character is ascii(255). The second character identifies the firmware revision. The third character of the power up packet tells the host which model *Si™* product is connected.

Note: Before installation into a multi-drop network, RS485 drives should be powered-up individually so that individual, nonconflicting addresses can be set using the "DA" command. The power up mode should be set to "2" at that time.

Using SCL with the SiNet Hub

The SiNet Hub-8 can be used to connect up to eight *Si™* drives to one PC/PLC serial port. The Hub444 can connect four drives. The hub acts a "router" sending your commands to the appropriate drive and returning information that you request from each drive. The addressing scheme is simple. If you want to route a Feed to Length command to the drive on Hub Port 4, simply send the string "4FL" followed by a carriage return.

If you want to request the input status of the drive on port 2, send the command "2IS". The drive will respond

and the hub will add an address character so that you know which drive the response came from. A typical response might be "2IS=00000000".

To send a command to all the drives that are connected to the hub, just omit the address (for example: FL).

The following SCL commands can also be used to control or query the hub itself. Address "0" is always used for commands that affect the hub.

0BR - This sets the bit rate of all the serial ports in the hub. 0BR1 specifies 9600 bits/second. 0BR2 indicates 19200 bps. The power up bit rate is 9600. You must do this in the correct sequence or communication will be lost. Think of your system as a pyramid: the PC is at the top. The next layer is the hub and the base of the pyramid consists of the drives that are connected to the hub. Always start at the bottom. For example:

- a) Make sure that all the drives in your system have firmware version 2.10 or later. The "RV" command will report the firmware of all drives connected to the hub.
- b) Send the command 1BR2. This asks drive 1 to change to 19200 bps.
- c) Repeat step (b) for each drive (2BR2, 3BR2, etc)
- d) Change the bit rate of the hub by sending 0BR2
- e) Change the bit rate of the PC. If you are trying this with the *SCL Setup Utility*, which we highly recommend, there is a drop down list for setting the PC bit rate.

Don't forget: at power down, the drive and hub change back to 9600 bps. If you send the hub the command "0BR" with no parameter, the hub will report its present bit rate, for example "0BR=1".

Requires Hub firmware version 1.14 or later, and drive firmware 2.10 or later.

0IH (Hub444 only) - Sets one of the Hub's four on-board outputs high (open) immediately. You must specify which of the four outputs you want to use: 0IH1, 0IH2, 0IH3 or 0IH4.

0IL (Hub444 only) - Sets one of the Hub's four on-board outputs low (closed) immediately. You must specify which of the four outputs you want to use: 0IL1, 0IL2, 0IL3 or 0IL4.

0IS (Hub444 only) - requests the input status of the hub's four inputs. Responds in the same format as a drive IS command. This is an immediate command.

0PM - sets the hub's power up mode. 0PM1 sets the hub for autodetect mode, where it can be used as an SCL Router or with the SiNet Hub Programmer software. A command button in the *SCL Setup Utility* can also be used to set the power up mode.

0RV - makes the hub report its firmware version.

0SO (Hub444 only) - sets one of the hub's four on-board outputs to a specified state (high or low). To set output 3 low, use the command "0SO3L".

Buffered or Immediate?

There are two basic types of SCL commands: *buffered* and *immediate*. Buffered commands execute one at a time. If you send two buffered commands to the indexer at the same time, like an FL and an SS, the SS command sits in a buffer and doesn't execute until the FL is completed. Other commands, indicated as *immediate* in the Command Summary, are executed right away, running in parallel with a buffered command if necessary. That allows you to check the buffer status (BS), or input status (IS) while the motor is moving.

About the Buffers

Like most communication devices, the SCL drives use buffers to smooth the flow of data through the drive. If you are using a fast computer to communicate with the drive, it is helpful to understand the data flow within the drive.

The Receive Buffer

When commands are sent to an SCL drive, they arrive in a piece of hardware called a UART. Every 78 microseconds, our drives run a piece of code called an interrupt service routine (ISR). The ISR handles time critical functions and looks for characters that have arrived at the UART. It moves these characters to a 32 character receive buffer.

A less time critical piece of code scans the receive buffer for new commands. If it finds an immediate command, it executes it right away. Immediate commands run very fast – in less than 1 millisecond. If the command is a buffered command, it is moved into the 128 byte command buffer.

The Command Buffer

Buffer commands are things like moves and parameter changes. These execute in order, one at a time. You can check the number of characters available in the command buffer by using the BS command.

The Transmit Buffer

Some commands, like IS, ask the drive to send information back to the PC. If the command waited for all this data to leave the drive (at 1 millisecond per character) it wouldn't be very immediate. So responses from commands like IS and SS are stored temporarily in a 32 character transmit buffer. The ISR makes sure these characters make it out of the drive.

Overflow

If you are not careful about sending characters to the drive, you can overflow these buffers. And if you do, strange things may happen.

It is useful to note the rate at which characters can enter and leave the drive: 9600 bits per second. To transmit a serial character using UARTS, 10 bits are sent. So characters enter and leave the drive at a rate never greater than 960 characters/second. That's roughly one every millisecond.

Receive Buffer Overflow

There are some instances where the receive buffer is not attended to promptly. If you stream characters into the drive at that time, the buffer can overflow. It is a circular FIFO buffer (they all are), so if it overflows, commands may be lost. They may also be misunderstood and the drive will send the host a question mark. In some cases, a command previously sent will be executed. None of this is good. And you cannot check the status of the receive buffer. Scared yet? Don't be – just be sure to observe these precautions:

After each of the following commands, you should not send any other commands to the drive for 50 milliseconds: FS, FD, FC, FY, SH, SK, ST, CJ.

During the WT command, you must use a 20 millisecond delay between commands.

Command Buffer Overflow

Since buffered commands such as moves can take a long time to execute, other buffered commands must wait. It is easy to overflow the command buffer if you don't pay attention. The easiest thing to do is to ask the drive how much space is available for buffered commands before you send one. You do this with the BS command. If the BS command indicates that less than 20 characters are available, you should not send more buffered commands to the drive.

Transmit Buffer Overflow

Some commands return more characters than you send. IS, for example, takes 3 milliseconds to send: one ms for the "I", one for the "S" and one for the carriage return that completes the command. When processed by the drive, IS will put 12 characters into the transmit buffer: for example "IS=10101010<cr>". Those characters need 12 ms to leave the drive, so if you send a continuous stream of IS commands, the transmit buffer will quickly overflow. You may see the same data come out of the drive twice, or you may get garbage. It is wise to put some delay after each status command (IS, BS, RV, IE, IP, etc.) or to wait for the response before asking for anything else.

Registers

Many SCL commands transfer information to a drive for later use. This information is stored in "registers" and remains there until a new command changes it or power is removed. For example, if you send the command "VE10", the VE register is set for a maximum move speed of 10 rev/sec. You can execute as many FL, FP or FS moves as you like without sending another VE command - the speed will remain at 10 until you change it.

In addition to the VE register, there are registers for move acceleration and deceleration (AC and DE), move distance (DI), and jog acceleration and speed (JA and JS). Besides the move parameters, there are registers for limit sensors (DL), motor current (CC), motor and encoder resolution (MR, ER), motor and encoder position (SP, EP).

Two special registers are PC (power on current) and PM (power on mode). Unlike the other registers, PC and PM retain their value even when power is removed from the drive. All other registers are "volatile", losing their information at power down, and reverting to default values at power on.

Idle Current Reduction

Any indexer-drive operating in SCL mode automatically reduces the motor current by 50% when the motor is not moving. If you need an idle current value other than 50%, you will need to use CC command before and after each move to "manually" set the current. To completely power down the motor, use the MD command.

Analog Inputs

The 1240i, 3540i and Si2035 drives each include one 10 bit, 0 to 5 volt analog input. This input can be read by the buffered command RA or by the immediate command IA. The RA command provides a decimal response in volts. The IA command responds with a hexadecimal value between 0 and 3FF that is proportional to five volts. For analog signal connections, please refer to your drive's hardware manual.

Command Summary

command	description	parameter	write only	read only	immediate	units	range	default	saved by SA
AC	accel rate	accel				rev/s/s	1 - 3000	25	•
AM	accel max (quick decel)	accel				rev/s/s	1 - 3000	500	•
BR	bit rate	rate code			•		1 - 3	1 (9600)	•
BS	buffer status			•	•	chars	0 - 128		
CC	change current	current				A	0 - imax		
CJ	commence jogging		•						
CS	change jog speed	speed	•		•	rev/sec	0 - 50		
CT	continue		•		•				
DA	define address	address					! - @		
DC	distance for FC, FM, FO, FY	distance				steps	16,000,000	20,000	•
DE	decel rate	decel				rev/s/s	1 - 3000	25	•
DI	distance or position	distance				steps	±16,000,000	20000	•
DL	define limits	limitstate					1 - 3	1	•
ED	encoder deadband	deadband				counts	1 - 255	10	•
EF	encoder function	function					0 - 3	0	•
EP	encoder position	position				counts	±16,000,000		
ER	encoder ratio	ratio					1 - 255	5	•
FC	Feed to Length with Speed Change		•						
FL	feed to length		•						
FM	Feed to Sensor with Mask Dist	inputnum,condition	•				1-8, H/L/R/F		
FO	Feed to Length & Set Output	outputnum,condition	•				1-3, H or L		
FP	feed to position		•						
FS	feed to sensor	input,condition	•				1 - 8, H/L/R/F		
FY	Feed to Sensor with Safety Dist	inputnum,condition	•				1-8, H/L/R/F		
HW	follow hand wheel	in,cond,steps/count	•			steps	1-8, H/L/R/F, 1-255		
IA	immediate analog input, in hex			•	•				
ID	immediate distance request, in hex			•	•				
IE	immediate encoder request, in hex			•	•				
IH	Immediate High output	outputnum	•		•		1 - 8		
IL	Immediate Low output	outputnum	•		•		1 - 8		
IP	immediate position request, in hex			•	•				
IS	input status request			•	•				
JA	jog accel/decel rate	accel					1 - 3000	25	•
JD	jog disable		•						
JE	jog enable		•						
JS	jog speed	speed					.025 - 50	1	•
MD	motor disable		•						
ME	motor enable		•						
MR	microstep resolution						3 - 16	8	•
PC	power up current	current				A	0 - imax		
PM	power up mode	mode (1=auto ,2=SCL)			1,2				
PS	pause		•						
RA	read analog input			•		volts	0 - 5		
RS	request status			•	•				
RV	revision level request			•	•				
SA	save all defaults		•						
SH	seek home	input,condition	•				1 - 8, H/L/R/F		
SJ	stop jogging		•						
SK	stop & kill buffer		•		•				
SO	set output	outputnum,condition	•				1-3, H or L		
SP	set abs position	position					±16,000,000		
SS	send string	text string	•						
ST	stop		•		•				
VC	velocity for speed change (FC)	speed				rev/sec	.025 - 50	1	•
VE	velocity setting	speed				rev/sec	.025 - 50	1	•
WI	wait for input	inputnum,condition	•				1-8, H/L/R/F		
WT	wait time	time	•			sec	.01 - 300		

Command Descriptions

AC – Acceleration

Sets accel rate in rev/sec/sec. Sending AC with no number causes drive to respond with present accel rate. Range is 1 – 3000.

Affects: FL, FS, FP, SH

See also: DE

Example:

You send	Drive sends
AC100	nothing
AC	AC=100

AM – Acceleration Maximum

Sets maximum accel rate in rev/sec/sec. This value is used if a move encounters an end of travel limit. It is also used by the ST and SK commands if they interrupt a move. Sending AM with no number causes drive to respond with present accel rate. Range is 1 – 3000.

Affects: FL, FS, FP, SH

See also: AC, DE

Not available with older firmware.

Example:

You send	Drive sends
AM500	nothing
AM	AC=500

BR – Bit Rate

Changes the serial port bit rate. This can be used to increase the communication speed between the host controller and the Si™ drive. You must change the bit rate of the host controller after sending this command or subsequent command will not be understood by the drive. Version 1.31 of the SCL Setup Utility has adjustable bit rates and is a useful tool for experimenting with bit rates. Drive powers up to 9600 bits/second.

Not available with older firmware. Some drives do not support BR3 (38400 bps).

Bit Rate Codes

1 = 9600 (about 1000 characters per second)

2 = 19200 (2000 chars/sec)

3 = 38400 (4000 chars/sec)

BS – Buffer Status

Drive tells you how much space is available in the command buffer.

Example:

You send	Drive sends	note
BS	BS=128	there is room for 128 additional characters

CC – Change Current

Changes current setting of drive. Also allows you to request present current setting. When you set the current with the CC, the idle current is automatically set to 50%.

Affects: FL, FS, FP, SH, WI (jogging)

See also: PC, CI

Example:

You send	Drive sends	note
CC5.1	nothing	run current set to 5.1 amps, idle current to 2.55 amps
CC	CC=5.1	current settings unchanged

CI – Change Idle Current

Changes idle current setting of drive. Also allows you to request present idle current setting. Note: this command will be overridden by any subsequent CC commands, which set the idle current to 50%. For best results, use the CI command after CC.

Affects: FL, FS, FP, SH, WI (jogging)

See also: PC, CC

Example:

You send	Drive sends	note
CI3.0	nothing	idle current set to 3 amps, run current unchanged
CI	CC=3.0	current settings unchanged

CJ - Commence Jogging

If jogging is enabled (JE command), the motor accelerates at rate set by JA command, then runs continuously at speed set by JS command. To stop jogging, use the SJ command if you want a controlled decel rate. For a faster stop, use the ST command, but beware that if the speed or load inertia is high, the motor may coast to a stop. The jogging direction is set by the sign of the last DI command. You can change the jog speed “on the fly” using the CS command.

CS – Change Jog Speed

Changes the software jog speed, on the fly. If the new speed is faster than the old one, the drive accelerates to the new speed using the speed set by JA. If the new speed is slower, the motor is decelerated using JA. CS requires a speed in rev/sec.

Not available on all models.

Example: **CS11.2**

CT – Continue

Resume execution of buffered commands.

Example:

You send	Drive sends
CT	nothing

DA – Define Address

Sets individual drive address character for multi-drop RS485 communication. This command should only be used with drives that have optional RS485 communications. Valid address characters are:

! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < > ? @

DC - Distance till Change

Sets or requests the “change” distance, in steps. The “change distance” is used by certain moves to change their behavior after this distance has been traveled. For example, FM is a Feed to Sensor move, but the sensor is ignored until the motor has moved DC steps. This is useful for “debouncing” a switch or sensor. Since DC is a relative distance, it is not necessary to provide a sign.

Range is 1 – 16,000,000.

Affects: FC, FY, FO, FM
See also: FC, FY, FO, FM

DE – Deceleration

Changes decel setting of drive. Also allows you to request present decel setting.
Range is 1 – 3000.
Affects: FL, FS, FP, SH
See also: AC

Example:

You send	Drive sends
DE125	nothing
DE	DE=125

DI – Distance/Position

Sets or requests move distance, in steps. The sign of DI indicates move direction.
Affects: FL, FS, FP, SH, CJ
See also: AC, DE, VE

Example:

You send	Drive sends	Notes
DI20000	nothing	cw direction
DI-20000	nothing	ccw direction
DI	DI=-20000	

DL – Define Limits

Sets limits to normally open (1), normally closed (2) or not used (3).
Affects: FL, FS, FP, SH, WI (jogging)

Example:

You send	Drive sends	Notes
DL1	nothing	limits are normally open
DL3	nothing	limits are not used

ED – Encoder Dead Band

On drives that have the encoder feedback option, this defines the size of the “in position” region. If static position maintenance is enabled and a motor at rest deviates from this zone, automatic correction occurs. If “end of move correction” is turned on, and the motor is outside the dead band at the end of a feed to length or feed to position move, automatic error correction occurs. The size is in encoder counts.

Example:

You send	Drive sends	Notes
ED50	nothing	dead band is 50 encoder counts
ED	ED=50	

EF – Encoder Function

On drives supporting encoder feedback, the EF command tells the drive what kind of position maintenance you want. Static position maintenance watches the encoder while the motor is at rest. If an external force moves the motor out of position, the drive tries to return it to within the dead band. “End of move correction” checks the encoder position after a feed to length or feed to position move. If the

motor is not within the dead band, corrective action is taken. For both static position maintenance and end of move correction, the corrective move length is half the distance to the ideal position. Additional corrections are made until the encoder position is within the dead band.

Example:

You send	Drive sends	Notes
EF0	nothing	all automatic encoder functions are off (but the encoder still tracks position and can be read via the EP or IE commands.)
EF1	nothing	Turns on end of move correction
EF2	nothing	Turns on static position maintenance
EF3	nothing	Enables static position maintenance and end of move correction

EP – Encoder Position

On drives supporting encoder feedback, the EP command allows the host to define the present encoder position. For example, if the encoder is at 4500 counts, and you would like to refer to this position as 0, send “EP0”. Sending EP with no position parameter requests the present encoder position from the drive.

ER – Encoder Ratio

On drives supporting encoder feedback, the ER command defines the encoder ratio. This number is the motor resolution, in steps/rev, divided by the encoder resolution, in counts/rev.

For example, if you are using a 4000 count encoder and the motor resolution is set to 20000 steps/rev (MR8), then you should set ER5. (Because $20000 / 4000 = 5$. The motor will take 5 steps per encoder count.) If your motor resolution is not evenly divisible by the encoder count, try a different motor resolution (using the MR command.) Encoders with binary resolutions, such as 512 and 1024, are unacceptable.

Note: the Si™ drive electronics use “X4” decoding, so a 1000 line encoder such as the U.S. Digital E2-1000-250-H produces 4000 counts/revolution.

FC – Feed to Length with Speed Change

Executes Feed to Length (relative move) command. Move distance and direction come from the last DI command. Accel and decel are from AC and DE commands. Initial speed is VE. After the motor had moved DC steps, the speed is reduced to VC. **DC must not be greater than VE, and DC must not exceed DI.**

Example: You’re drilling holes. You want to rapidly approach the workpiece, then slow down for drilling. Total move distance is 50000 steps, but after 45000 steps, you want to reduce the speed from 8 rev/sec to 0.5 rev/sec.

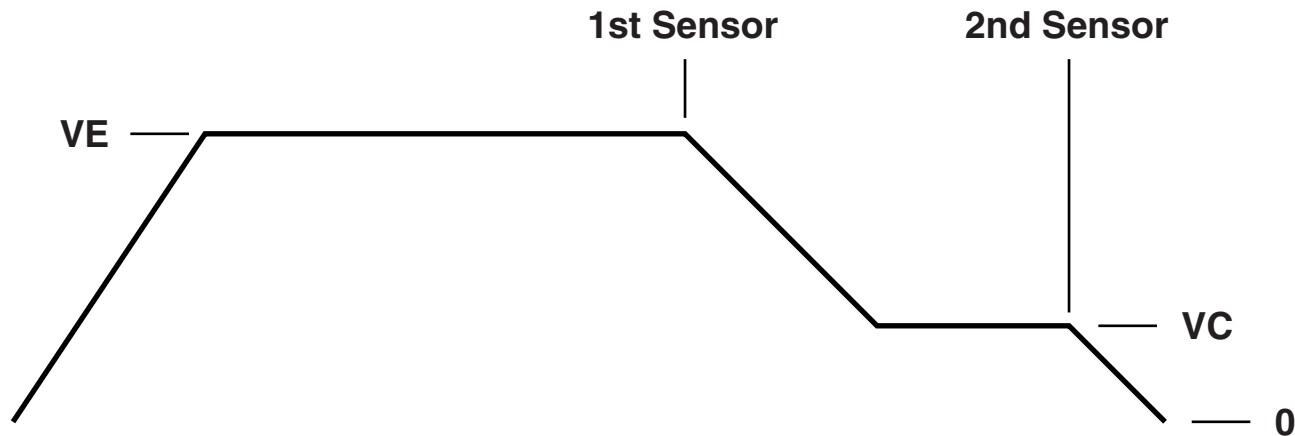
You send	Notes
VE8	initial speed is 8 rev/sec
VC0.5	change speed is 0.5 rev/sec
DI50000	total move distance is 50,000 steps
DC45000	change distance is 45,000 steps
FC	launch move

FD - Feed to Double Sensor

Accelerates the motor at rate AC to speed VE. When the first sensor “trips”, the motor decelerates (at rate DE) to speed VC. When the second sensor is reached, the motor decelerates to a stop at rate DE. The sign of the DI register is used to determine the direction.

Example:
You send
D11
VE5
VC.2
FD2F4H

Notes
move direction will be cw
speed will be 5 rev/sec until first sensor is reached
speed will be 0.2 rev/sec after first sensor is reached
move until falling edge on input 2, then decel to VC.
Decel to stop when input 4 goes high.



FD Command: Speed vs Time

FL – Feed to Length

Executes Feed to Length (relative move) command. Move distance and direction come from the last DI command. Speed, accel and decel are from VE, AC and DE commands.

FM – Feed to Sensor with Mask Distance

Executes Feed to Sensor command, but sensor is ignored for the first DC steps of the move. Useful for “debouncing” a switch or clearing a part before sensing the next one.

Example: You’re feeding parts on a conveyor. A sensor detects the leading edge of the part and stops. But if the part has a hole in it, which many objects do, then when you attempt to feed the next part into position you may in fact stop after feeding the previous part only a short distance. The solution is the use FM instead of FS and to set the DC for the size of the part.

Example: The parts are 6 inches long. Your mechanical linkage provides 20000 steps/inch. You want the part to stop moving 1 inch past the sensor. 5 inches of the part will not have gone past the sensor yet.

You send
DI20000
DC102000
FM1F

Notes
stop 20000 steps (1 inch) past sensor
ignore sensor for next 5.1 inches, allowing old part to completely clear sensor
launch move. Sensor is connected to input 1 and goes low when it sees a part

F0 – Feed to Length and Set Output

Same as Feed to Length (FL) but changes the state of an output during the move.

Example: You’re feeding parts to be cut to length. For maximum throughput, you want to trigger the cut off knife as the part is nearing the final position.

You send	Notes
DI20000	feed 20000 steps
DC15000	set output at 15000 steps
FO1L	close output 1 at 15000 steps

FP – Feed to Position

Executes Feed to Position (absolute move) command. Move position comes from the last DI command. Speed, accel and decel are from VE, AC and DE commands.

FS – Feed to Sensor

Executes Feed to Sensor command. Requires input number (1-8) and condition (H=high, L=low, R=rising, F=falling) The motor moves until the sensor state change is detected, then stops a precise distance beyond the sensor. That distance is specified by the DI command. The direction of rotation is determined by the sign of the DI command (positive is clockwise). Speed, accel and decel are from the most recent VE, AC and DE commands.

A motor moving at a given speed, with a given decel rate, needs a certain distance to stop. If you specify too short a distance, the drive may become confused and greatly overshoot the target. Use the following formula to compute the minimum decel distance, given a velocity V (in rev/sec) and decel rate D (in rev/sec/sec.). R = steps/rev.

$$DI_{\min} = R \frac{V^2}{2D} \quad \text{Example: } DI_{\min} = (20000) \frac{(1)^2}{(2)(25)} = 400 \text{ steps}$$

The Help screen of the *SCL Setup* utility contains a special calculator that computes the distance for you.

Example:

You send	Drive sends	Notes
FS1L	nothing	Feed to Sensor 1 low
FS3R	nothing	Feed to Sensor 3 rising edge

FY – Feed to Sensor with Safety Distance

Executes Feed to Sensor command, but monitors total distance traveled. If sensor is not found before distance exceeds DC, the motor is stopped and the drive sends the host and exclamation point (“!”).

This is useful for detecting machine jams or the end of a roll of labels. For example: you are feeding labels. You want to stop each label 2000 steps after the sensor detects the leading edge. The labels are 60,000 steps apart. Therefore, if you move the roll more than 60,000 steps without detecting a new label, you must be at the end of the roll.

You send	Notes
DI2000	stop 2000 steps beyond sensor
DC60000	stop after 60000 steps if no sensor detected
FY2L	feed to sensor 2 low

HW – Hand Wheel

Commands the drive to follow the position of a low speed quadrature encoder. This instruction is intended to allow the drive to be manually positioned using a CNC hand wheel. The HW command terminates when a condition on an input is met. You must include the input and condition in the command. You must also provide a move increment in motor steps per encoder (hand wheel) count.

Example:

You send	Drive sends	Notes
HW4L10	nothing	drive follows hand wheel until input 4 is low, 10 steps/count

IA - Immediate Analog

Requests a reading from the analog to digital converter connected to input pin AIN. Value is provided in hexadecimal. 0 represents 0 volts at the AIN input and 3FF represents 5 volts DC. To convert the hex value to a voltage, use the formula $v = IA * 5 / 1023$. For example, if the drive responds with "IA=2C0", which converts to 704, the voltage is $v = 704 * 5 / 1023 = 3.441$ volts.

The IA command is immediate, so it will respond even during moves, WT and WI commands. See also: RA.

ID – Immediate Distance

Requests present distance, in hex. (Distance is in hex because conversion to ascii of another format would tax the CPU enough to interfere with a move in progress. Application programs can easily convert a hex value to integer.)

Example:

You send	Drive sends	Notes
ID	ID=00002710	+10000 (10,000 steps into cw move)
ID	ID=FFFFD8F0	-10000 (10,000 steps into ccw move)

IE – Immediate Encoder

Requests present encoder position, in hex. (Distance is in hex because conversion to ascii of another format would tax the CPU enough to interfere with a move in progress. Application programs can easily convert a hex value to integer.)

Example:

You send	Drive sends	Notes
IE	IE=00002710	encoder position is +10000 counts
IE	IE=FFFFD8F0	encoder position is –10000 counts

IH – Immediate High Output

Sets output high (open) immediately. Use SO instead if you don't want the output to change until a buffered command (like a move) is complete.

You send	Drive sends
IH1	output 1 goes high immediately
IH2	output 2 goes high immediately

See also: IL, SO

IL – Immediate Low Output

Sets output low (closed) immediately. Use SO instead if you don't want the output to change until a buffered command (like a move) is complete.

Example:

You send	Drive sends
IL1	output 1 goes low immediately
IL2	output 2 goes low immediately

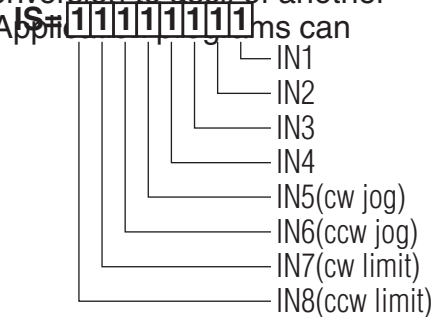
See also: IH, SO

IP – Immediate Position

Requests present absolute position, in hex. (Value is in hex because conversion to ascii of another format would tax the CPU enough to interfere with a move in program. Applying **IS=11111111** ms can easily convert a hex value to integer.)

Example:

You send	Drive sends	Notes
IP	IP=00002710	abs position is 10,000 steps



IS – Input Status

Requests immediate status of all 8 inputs.

Example:

You send	Drive sends	Notes
IS	IS=00000000	all 8 inputs are low (closed)
IS	IS=11111111	all 8 inputs are high (open)
IS	IS=00000001	input 1 is high
IS	IS=10000000	input 8 is high

JA – Jog Acceleration

Sets accel/decel rate for jog moves in rev/sec/sec. Sending JA with no number causes drive to respond with present jog accel/decel rate. Range is 1 – 3000.

Affects: WI (jogging)

See also: JS

Example:

You send	Drive sends
JA100	nothing
JA	JA=100

JD – Jog Disable

Disables jog inputs (normally active during WI instructions)

JE – Jog Enable

Enables jog inputs (active during WI instructions)

JS – Jog Speed

Sets speed for jog moves in rev/sec. Sending JS with no number causes drive to respond with present jog speed. Range is .025 – 50.

Affects: WI (jogging)

See also: JA

Example:

You send	Drive sends
JS10.35	nothing
JS	JS=10.35

MD – Motor Disable

Disables motor (cuts current to zero).

ME – Motor Enable

Restores previous motor current.

MR – Microstep Resolution

Sets, or requests microstep resolution. For a drive with built-in amplifier, like the Si5580, the range is 3 – 15, from the table below. The MR command should be used before setting the accel and decel rates and speed, because a change in motor resolution will corrupt these settings. The MR command also resets the step table, which moves the motor to the nearest pole position. The absolute position register is not changed.

Example:

You send	Drive sends	Notes
MR8	nothing	sets drive to 20,000 steps/rev
MR	MR=8	

MR Code	Steps/rev	MR Code	Steps/rev	MR Code	Steps/rev
3	2000	8	20000	13	36000
4	5000	9	21600	14	50000
5	10000	10	25000	15	50800
6	12800	11	25400		
7	18000	12	25600		

The Si-100 is a special case. It is an indexer with pulse and direction outputs that can be connected to many different drives. If you want speeds and acceleration rates to be accurate, you must set the Si-100 to match the resolution of your drive. First, enter the steps/rev using the DI command. Then send the command “MR16” to accept the contents of the DI register as the new motor resolution.

Example (a servo drive with 1024 line/4096 count encoder):

You send	Drive sends	Notes
DI4096	nothing	sets DI register to 4096
MR16	nothing	sets the motor resolution at 4096 steps/rev

If you have a servo drive like the BL7080i, the motor/encoder resolution can only be set using the *Quick Tuner™* software. MR is permanently set to 16 on the BL7080i servo drive.

PC – Power on Current

Sets power on current in amps. Also changes present current.

Example:

You send	Drive sends
PC3.2	nothing

PM – Power on Mode

Sets or requests power on mode. 1=autodetect. 2=SCL mode only. Set to 1 if you plan to use the drive with the *Si Programmer™*.

Note: RS485 drives do not work with the *Si Programmer™* software.

Example:

You send	Drive sends	Notes
PM1	nothing	drive is set to auto detect host at power on
PM2	nothing	drive is set for SCL mode at power on
PM	PM=2	

PS – Pause

Suspends execution of buffered commands until the next CT command. Useful for coordinating motion among axes by sending commands to each, while paused, then resuming all drives at once. Also can be used to suspend the operation of a machine.

RA - Read Analog

Requests the voltage on the AIN (analog input) pin. Range is 0 to 5 volts. Resolution is 10 bits. This is a buffered command, so it will wait for other buffered commands such as moves to be completed before responding. For an immediate reading, use IA.

RS – Request Status

Asks the drive to tell you what its doing. Responses are:

M = motion in progress

W = wait input command executing

T = wait time command executing

E = servo positioning fault (drive must be reset by interrupting power to clear this fault)

R = ready (none of the above happening)

RV – Revision Level

Asks the drive what firmware it has in it.

Example:

You send	Drive sends	Notes
RV RV=150	drive has firmware version 1.50	

SA – Save All

Saves the present values of AC, AM, BR, DC, DE, DI, DL, ED, EF, ER, JA, JS, MR, VC and VE as the new power on defaults.

Only available on 1240i, 3540i and Si2035. Requires firmware version 2.23A or later.

Notes: use caution when using the SA command when BR has been changed. Drives will only communicate with the Si Programmer software when BR=1 and PM=1. If BR has been changed, be sure to change the bit rate setting of the SCL Utility to match.

SH – Seek Home

Executes seek home command. Requires input number (1-8) and condition (H=high, L=low, R=rising, F=falling) Speed is set by the last VE command. Accel and decel are set by AC and DE. Direction comes from the sign of the last DI command (+ is clockwise, - is ccw).

Example:

You send	Drive sends	Notes
SH1L	nothing	Seek home 1 low
SH3R	nothing	Seek home 3 rising edge

SJ – Stop Jogging

Stops the motor when jogging (CJ starts it). Decel rate is defined by JA command.

SK – Stop & Kill

Halts any buffered command in progress. Removes any other commands from buffer.

SO – Set Output

Sets an output to a condition. Outputs: 1-3. Conditions: H=high(open), L=low(closed).

SP – Set Position

Set or request absolute position. Affects FP commands.

Example:

You send	Drive sends
SP100	nothing
SP	SP=100

SS – Send String

Drive sends a text string to the host when this buffered command is executed. The maximum string length is determined by the available space in the command buffer (the BS command can tell you what's available).

Example:

You send	Drive sends
SSMove complete	move complete

ST – Stop

Terminates any buffered command in progress.

VC – Velocity Change

Sets or requests the “change speed”, in rev/sec, for FC moves. Range is .025 - 50. Affects: FC.

VE – Velocity

Sets or requests move speed in rev/sec. Range is .025 - 50.

Affects: FL, FS, FP, SH.

Example:

You send	Drive sends
VE2.525	nothing
VE	VE=2.525

WI – Wait for Input

Waits for an input to match a condition. Inputs: 1-8. Allows very precise triggering of moves if a WI command precedes a move command in the buffer.

Conditions: H=high, L=low, R= rising edge, F=falling edge.

Jogging is active during this instruction, unless disabled by JD.

Example:

You send	Drive
WI3R	waits for rising edge on input 3 before proceeding to next buffered command.

WT – Wait Time

Causes a time delay, in seconds. Range is .01 – 300.