SVX Servo Suite

Software Manual

Contents

1 Introduction

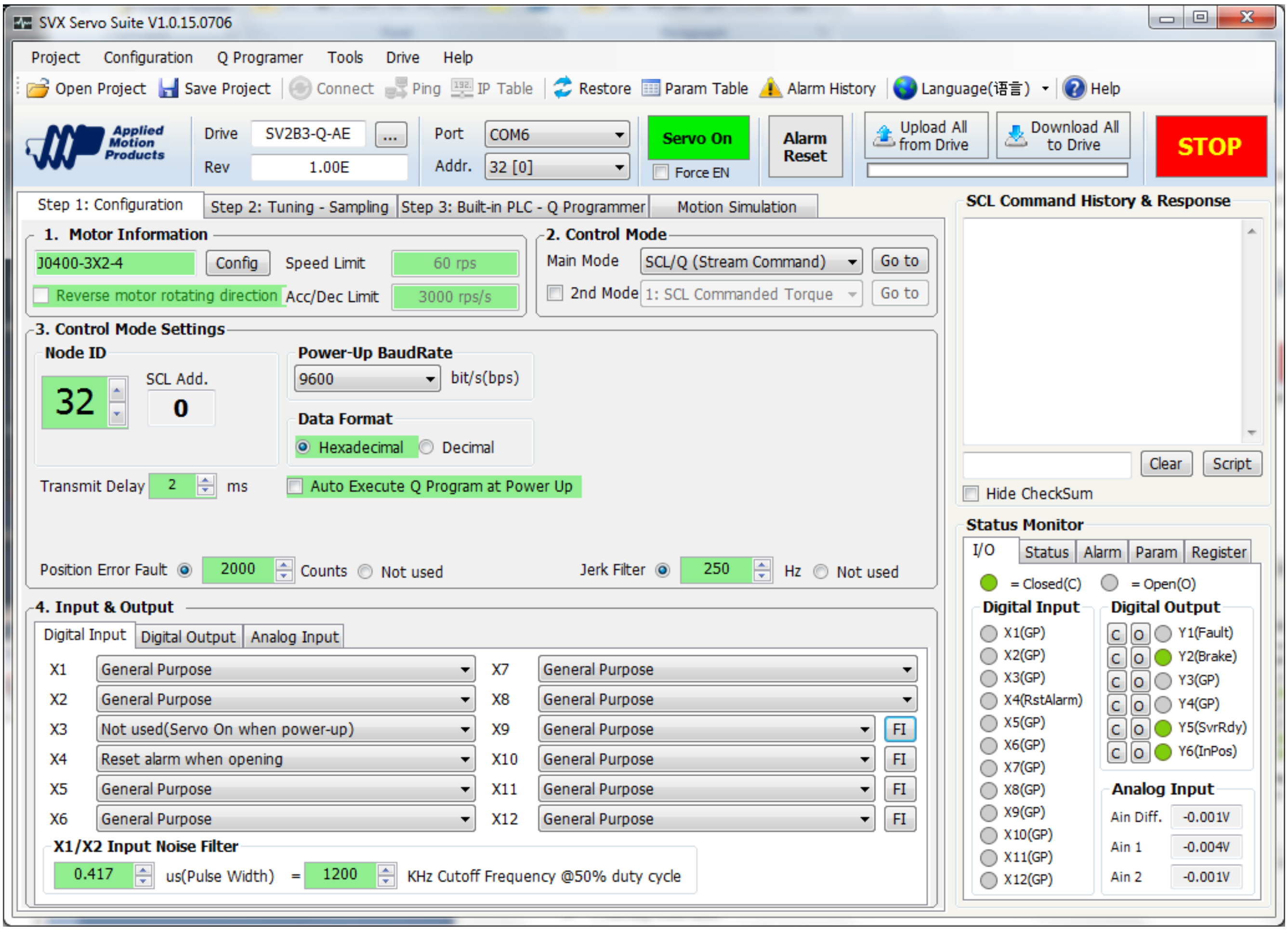
Thank you for purchasing an Applied Motion Products SV200 series product. We hope our dedication to performance, quality and economy will make your motion control project successful.
The SV200 Series, a new generation AC Servo system features

- excellent frequency response and setting time
- fully compatible 40, 60 and 80mm servo motors (with fail safe brake option)
- wide selection of control modes including
    - pulse & direction
    - analog torque & velocity
    - streaming commands over RS-232, RS-485, CANopen, Ethernet, EtherNet/IP and Modbus
    - stored programming using Q language
    - position table selected by digital inputs
- available in 120 and 220VAC versions
- three power levels:
    - 200 watts (100W at 120 VAC)
    - 400 watts (200W at 120 VAC)
    - 750 watts (400W at 120 VAC)
- easy to use online auto-tuning
- advanced anti-vibration with two notch filters

The SV200 is particularly suitable for high speeds, high torque, high accuracy, safety and long-life applications such as: factory automation, semiconductor manufacturing equipment, SMT, PCB, LED, packaging, labeling, and food processing equipment, robots and custom machinery.

1.1 SVX Servo Suite Overview

The SVX Servo Suite is a Windows PC based software application used to configure, tune, and program your servo system. This document explains how to install the SVX Servo Suite and how to configure and tune your servo system. For information regarding your specific hardware, such as wiring and mounting, please read the **SV200 User Manual** which can be downloaded from our website.



The features of SVX Servo Suite include:

- Friendly Interface
- Easy setup within just three steps
- Drive setup and configuration
- Servo control gains Auto-tuning
- Servo tuning and sampling
- Built-in Q programmer
- Motion testing and monitoring
- Write and save SCL command scripts
- Online help integrated

If you get in trouble with using our driver or software, or if you have any suggestions about our products and this manual, please call (800) 525-1609 or send an email to support@applied-motion.com to let us know.

Our software is supported on PC's running Microsoft Windows XP (Service Pack 3), Vista, 7 or 8, 32 or 64 bits.

The SVX Servo Suite requires Microsoft .Net Framework 2.0

1.2 SVX Servo Suite Setup

SVX Servo Suite can be downloaded from our website's software page. After downloading, open the SVX Servo Suite setup file and follow the on screen instructions.

1.3 Install the USB Serial Driver

For configuration and tuning, you'll need to connect your SV200 to your PC using the included mini USB cable for communication. USB serial drivers are normally installed automatically when you install the SVX Servo Suite software.

If your PC asks you to install the hardware driver when you connect the SV200 to your PC, you can find the USB driver file on your system here:

C:\Program Files (x86)\Applied Motion Products\SVX Servo Suite\Driver Installation Tool

In this folder, you can choose:

"x86" for 32 bit systems

"x64" for 64 bit systems

After opening one of the folders above, double click **MCP2200DriverInstallationTool** to install the USB serial driver.

2 Using the SVX Servo Suite software with your drive

SVX Servo Suite offers two types of communication: serial communication over USB and Ethernet communication.

2.1 Connecting Your Drive to the SVX Servo Suite

Connect to an SV200 drive via serial communication over USB:

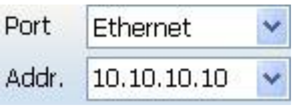
- Connect the drive to your PC's USB port
- Launch SVX Servo Suite
- Select the COM port that the USB serial driver is using (see picture below)
- Power up the drive
- SVX Servo Suite will recognize the drive model and revision

When launching SVX Servo Suite, the software will search all COM ports available and load the drop down list.

For SV200 Ethernet drives, follow these steps:

- Connect the drive and PC to your switch or router using CAT5 or CAT6 patch cables. Or you can connector the drive directly to the Ethernet port of a PC. A crossover cable is not required.
- Launch SVX Servo Suite
- Switch to Ethernet and enter the IP address of the drive as shown below (a factory fresh drive will be preset to 10.10.10.10)
- Power up the drive

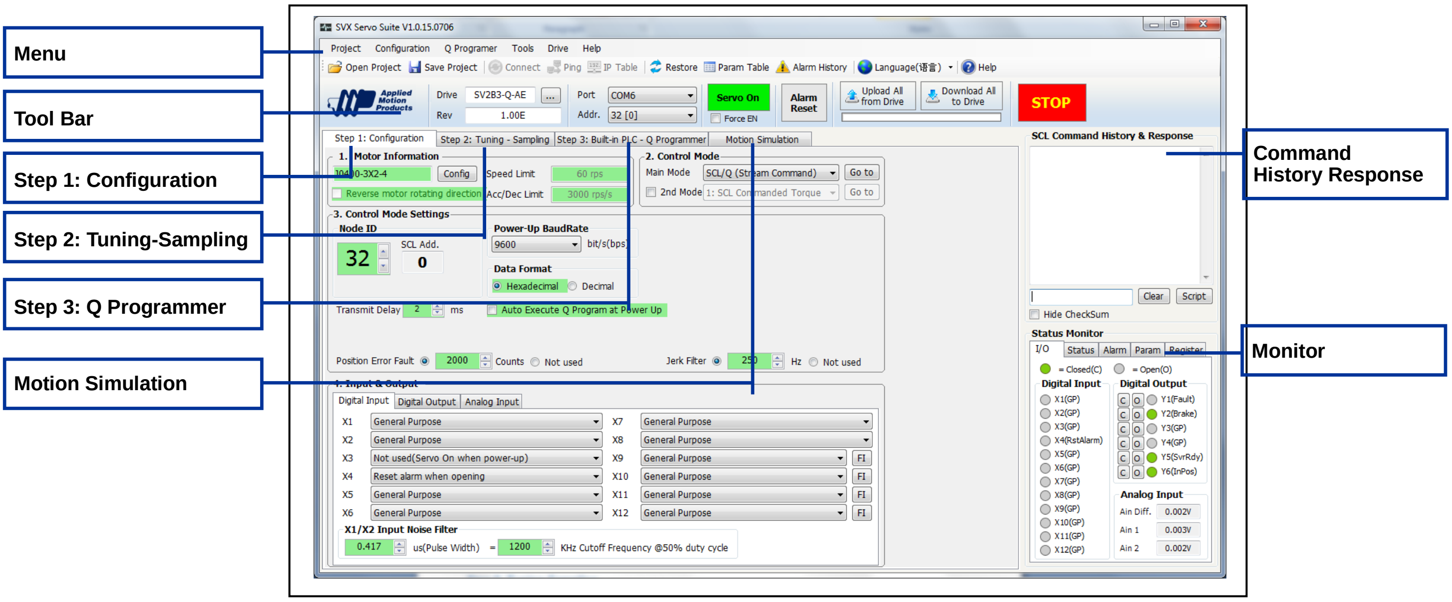
SVX Servo Suite will not detect the drive information automatically, you need to click the Upload button or the Connect button in the main screen to get the drive model and revision.



2.2 User Interface

To launch the ***SVX Servo Suite,*** select it from the Start menu: ***Start*** → ***Programs*** → ***Applied Motion Products*** → ***SVX Servo Suite*** → ***SVX Servo Suite.***

The Main screen includes: Menu, Tool Bar, Step 1: Configuration, Step 2: Tuning-Sampling, Step 3: Q Programmer (only for –Q/-C models) and Motion Simulation as shown below.



Menu
The main menu provides some frequently-used operations for Project, Configuration, Tools, Q Programmer, Drive and Help.

Tool Bar
Tool Bar is used to set the communication, Open Project, Save Project, Connect, Ping, IP Table, Restore, Parameter Table, Alarm History, Change Language, drive model, Servo status control, Alarm Reset, Upload & Download, Emergency stop.

Step 1: Configuration
This tab provides the drive configuration settings, such as 1 Motor Information, 2 Control mode, 3 Control mode settings, Input & Output.

Step 2: Tuning-Sampling
This tab provides the Auto-tuning and sampling settings, start sample and display sampling curve diagram.

Step 3: Q Programmer
This tab provides some functionality to program environment, test, save and download or upload the Q program. It is only shown for –Q and –C models.

Motion Simulation
This tab provides motion test, such as point to point motion, jogging, homing etc...

SCL Terminal
The SCL Terminal allows you to send SCL commands to the drive.

Status Monitor
Status Monitor can display I/O status, Drive status, Alarm, Parameters and Register monitor.

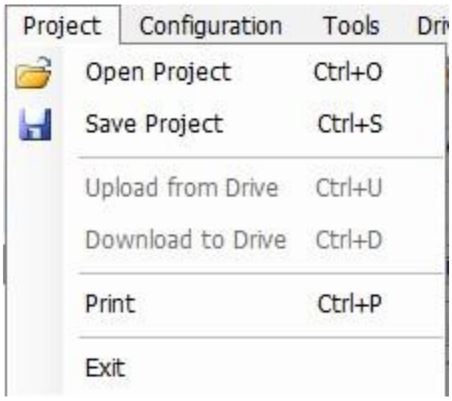
2.3 Menu

Project Configuration Tools Drive Help

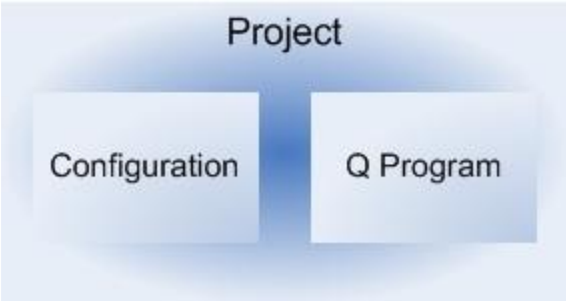
1st Stage Menu	2nd Stage Menu	Hot Key	Function
Project	Open	Ctrl+O	Open project file (.mdvprj format)
	Save	Ctrl+S	Save project file (.mdvprj format)
	Upload from Drive	Ctrl+U	Upload project from the drive
	Download to Drive	Ctrl+D	Download project to the drive
	Print	Ctrl+P	Print current project
	Exit		Exit SVX Servo Suite application
Config	Open Config	Ctrl+Shift+O	Open configuration file (.mdvcfg format)
	Save Config	Ctrl+Shift+S	Save configuration file (.mdvcfg format)
	Upload from Drive	Ctrl+Shift+U	Upload configuration from the drive
	Download to Drive	Ctrl+Shift+D	Download configuration to the drive
	Print	Ctrl+Shift+P	Print current configuration
Tools	Firmware Downloader		Upgrade the drive's firmware
	Calibration		Calibration for non-Applied Motion motor
	Move Profile Calculator		Pilot motion profile based on target distance, velocity, acceleration/deceleration, etc.
	Export CANopen Parameters		Export CANopen Parameters to file
	CANopen Test Tool		Run CANopen Test Tool application (require pre-installation)
Q Program	Open Q Program		Open Q program file (.qpr format)
	Save Q Program		Save Q program file (.qpr format)
	Open Segment		Open Q segment file (.qsg format)
	Save Segment		Save Q segment file (.qsg format)
	Upload from Drive		Upload Q program from the drive
	Download to Drive		Download Q program to the drive
	Clear Q Program		Clear Q program
	Set Password		Set password to secure Q program
	Print Q Program		Print Q program
Drive	Connect	Ctrl+R	Connect or Re-connect to the drive
	Stop	Ctrl+F5	Emergency Stop
	Ping		Ping to the Ethernet drive
	IP Table		Edit user defined IP address through IP table No. 1to E
	Param Table		Display the Parameter Table
	Script		Run the SCL Script
	Option		Set Alarm, Regen, Communication and other options
	Restore		Configure the drive to Factory Default Setting
	Alarm History		Record drive's alarm history
Help	About		Get the software version
	Help Content		Open online help

2.3.1 Project

In the Project menu, the SVX Servo Suite can allow you to upload and download both configurations and Q programs. Drive configurations and Q programs can be saved as a single project file (.mdprj) to your local disk. SVX Servo Suite can also download the project files to a different drives directly from the hard disk. In addition, it can also print out the detailed project files.



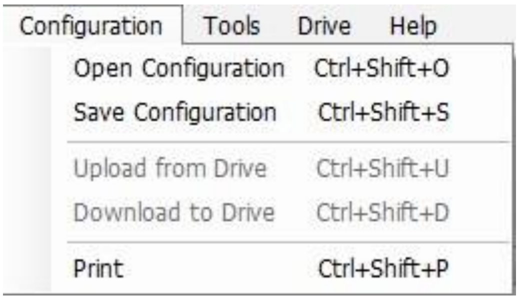
For the drive models that support Q programming capability, the project includes the configuration and Q program , see picture below:



For the drive without Q programming capablility, the project file only contains the configuration.

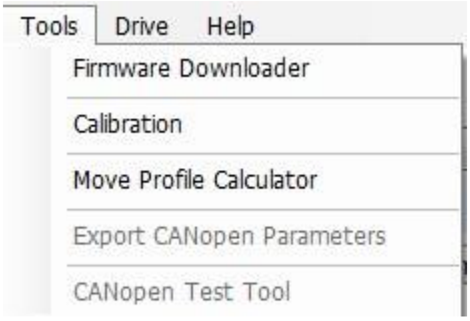
2.3.2 Configuration

In the Configuration menu, the SVX Servo Suite allows you to upload and download drive configurations. It can also save as the configuration file (.mdcfg) to your local disk and download configurations to different drives directly from the hard disk. In addition, it can print out the configuration details.



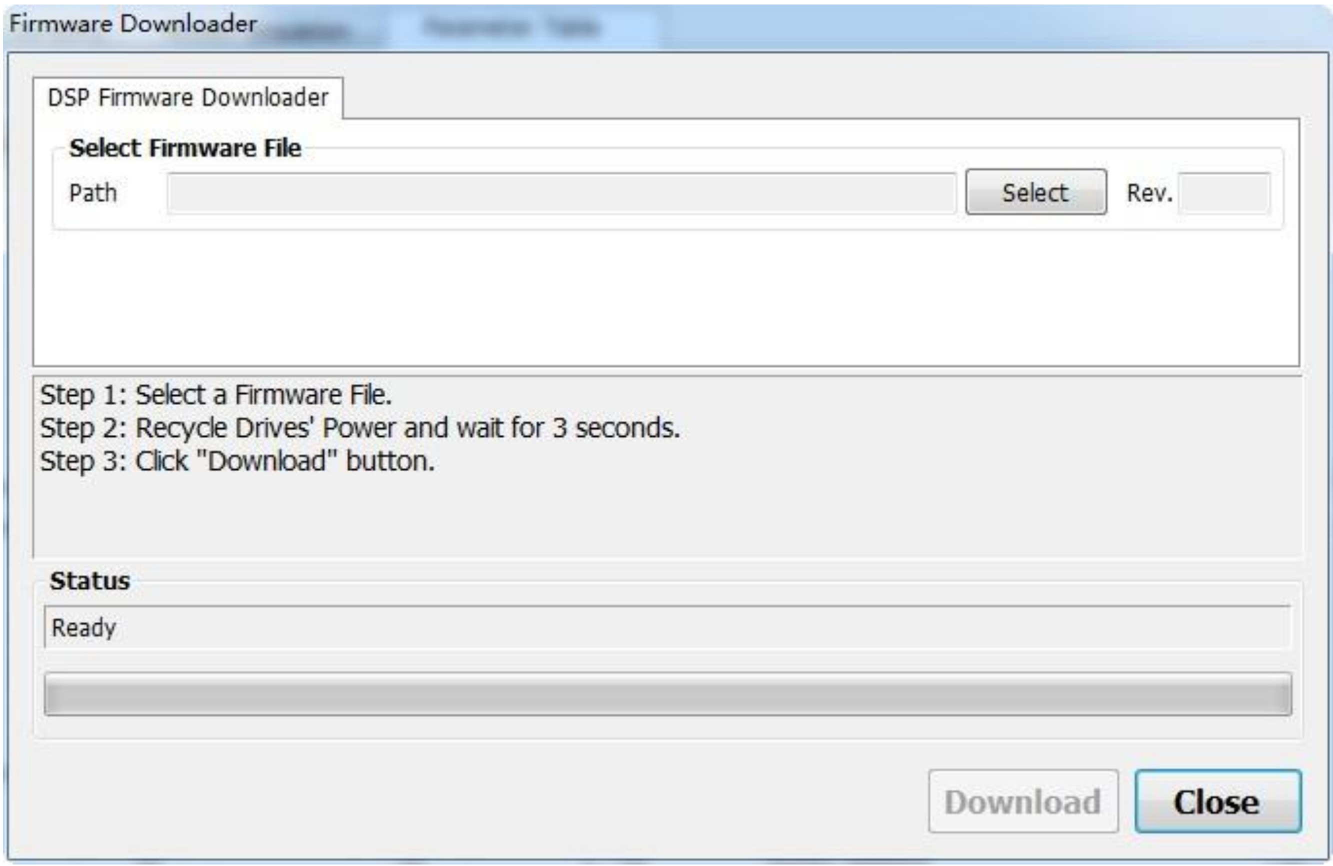
2.3.3 Tools

The Tools menu includes Firmware Downloader, Calibration, Motion Profile calculator, Export CANopen Parameters and CANopen Test Tool, see picture below:



2.3.3.1 Firmware Downloader

The Firmware Downloader can used to upgrade your drive's firmware. Before upgrading please contact the Applied Motion Products tech support team to obtain the latest firmware version.



Please follow this sequence to perform a firmware update:

Step 1: Select a firmware file

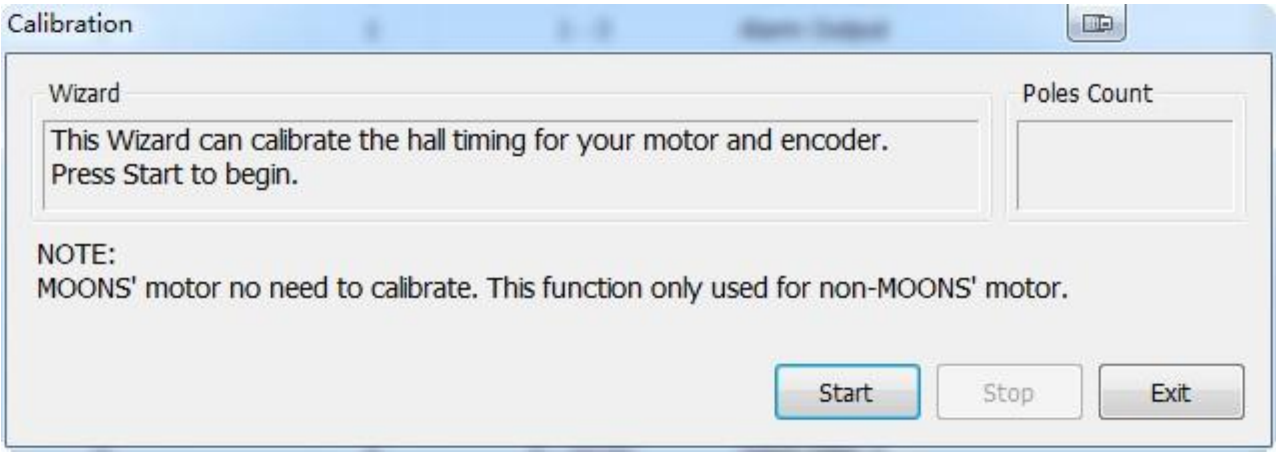
Step 2: Recycle drive's power and wait for 3 seconds

Step 3: Click the "Download" button.

Note: SV200 drives do not support multi axis networking firmware updates for RS-485 field bus. You can only do the firmware updates for each single axis which must be offline from the network.

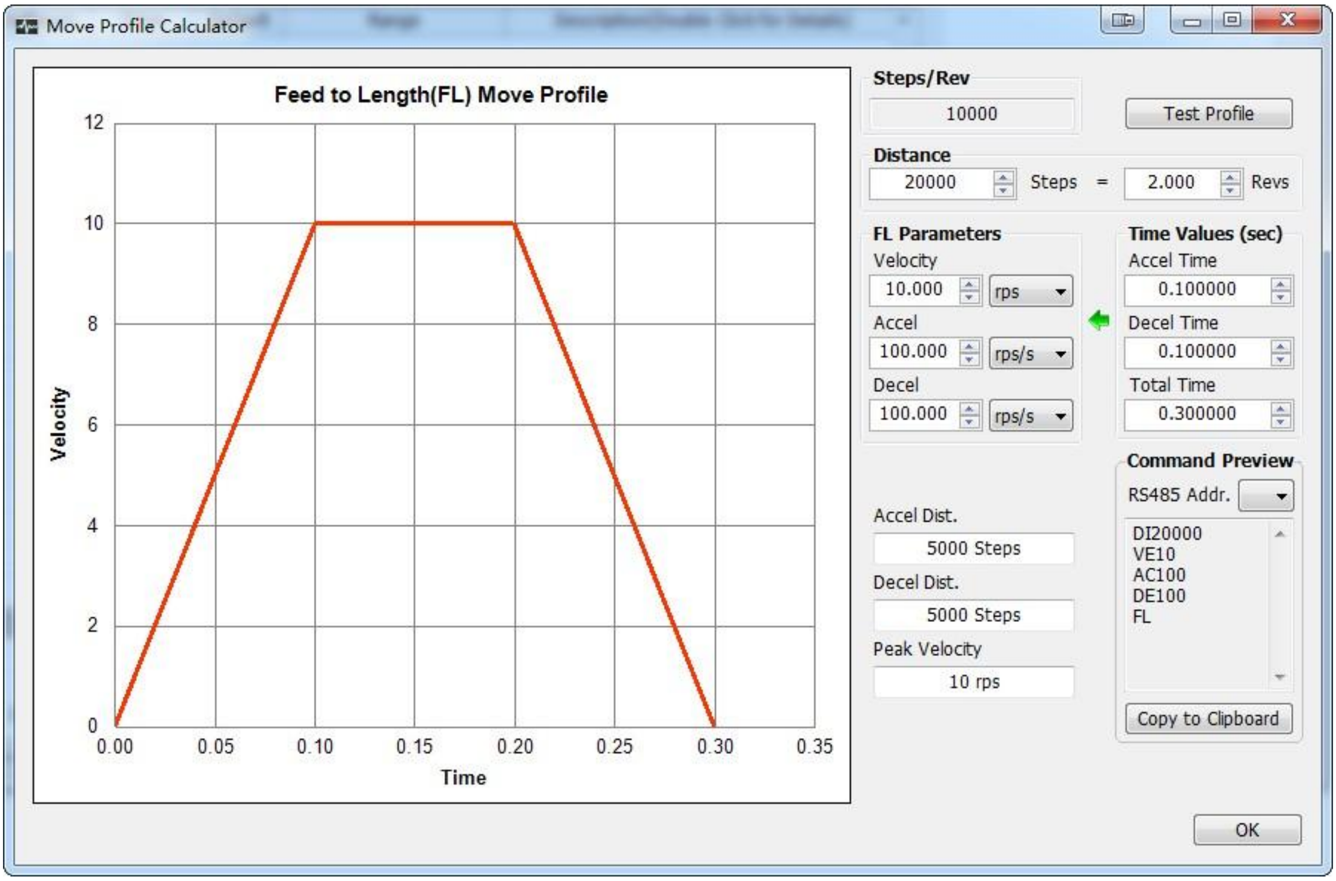
2.3.3.2 Calibration

This tool helps you calibrate a servo motor which is not made by Applied Motion Products. In most cases, it can automatically detect your motor timing pattern and configure the drive settings for it.



2.3.3.3 Move Profile Calculator

The Move Profile Calculator provides an excellent tool for simulating a motion profile. You can enter SCL commands for acceleration, deceleration, speed and distance and see a plot of velocity versus time and a listing of time to speed, time at speed and total move duration. Or you can specify the timing and the calculator will tell you the necessary parameters to make it happen. When a drive is connected with the software, you can click "Test Profile" to try an actual move with the parameters you've entered.

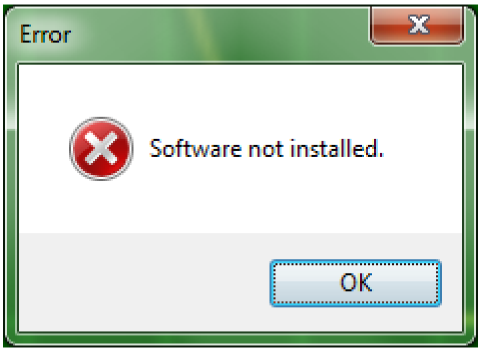


2.3.3.4 Export CANopen Parameters

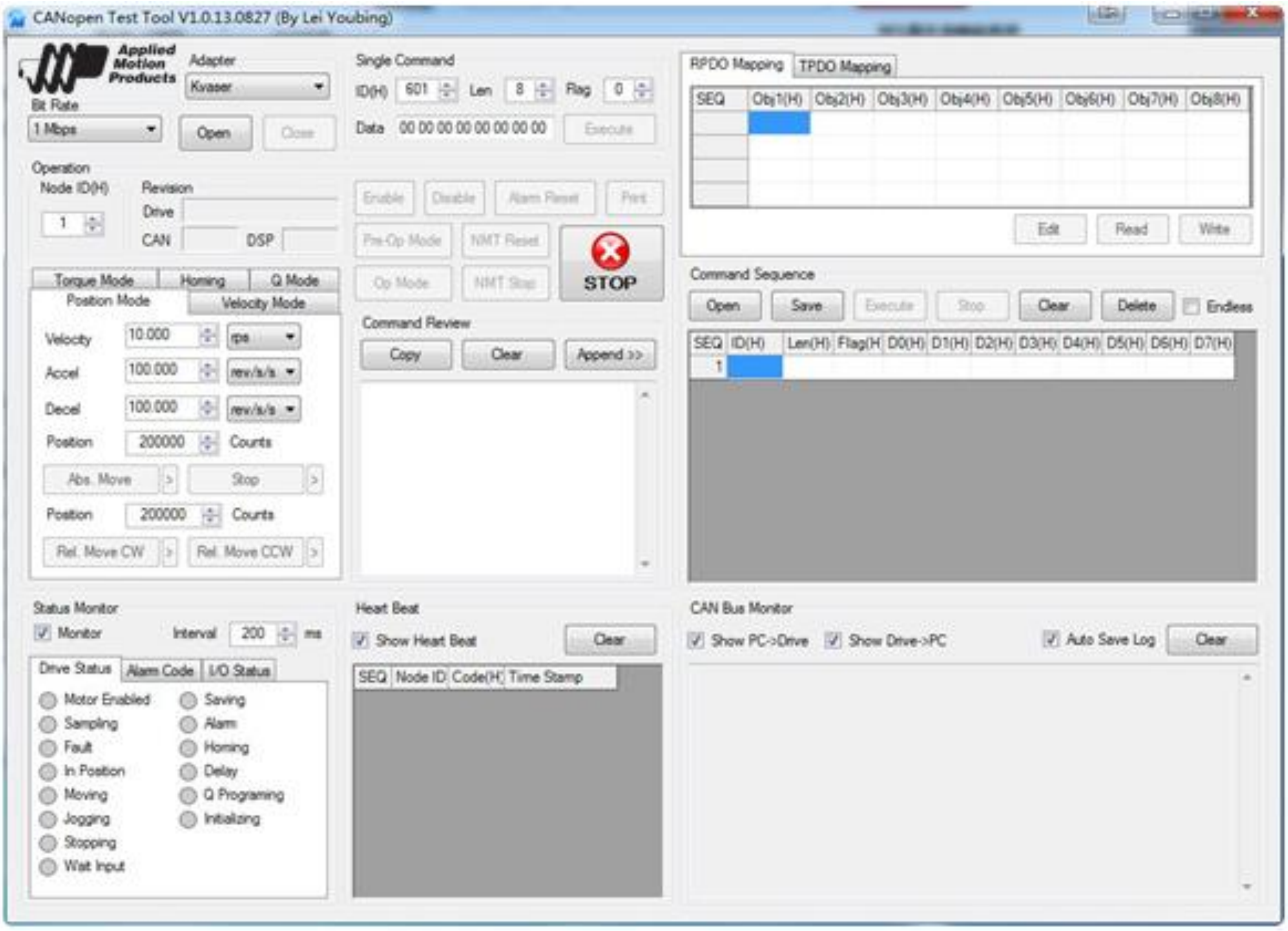
After tuning is done, you can use the Export CANopen Parameters tool to export tuning parameters such as KP, KD, VP, VI to a text file in a data format that is easy for the customer to immigrate to their CANopen program. Below is an example of a saved file.

2.3.3.5 CANopen Test Tool

This provides a quick link to the installed CANopen Test Tool software. If you have installed the CANopen Test Tool, click this will launch “CANopen Test Tool” software. If you see this message, you need to download and install the CANopen Test Tool:

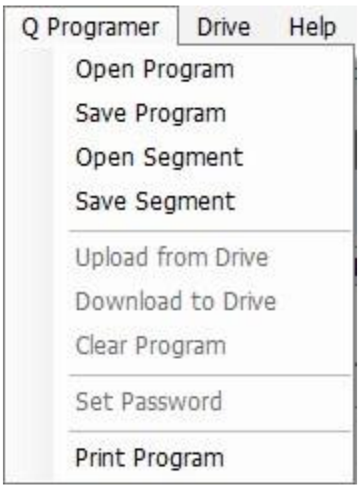


If the CANopen test tool has been installed, you should see a screen like this:



2.3.4 Q Programmer

If your drive is a Q model, the Q Programmer menu can save drive's Q program to your local disk as a ".qpr" file. It can also download the Q program to a different drive directly from the hard disk. In addition, it can print out the your Q program.



2.3.5 Drive Menu

The Drive menu has the following functions:

Menu	Name	Hot keys	Description
Drive	Connect	Ctrl+R	Connect Drive
	Stop	Atl+F5	Emergency stop
	Ping		Ping for Ethernet driver
	Edit IP table		Edit IP table for Ethernet drive
	Parameter Table		Display parameter
	Script		Run script file
	Restore Factory default	Ctrl+Shift+D	Restore drive to factory default mode
	Restore tuning parameters		Restore drive's tuning default setting
	Alarm history	Ctrl+Shift+A	Check alarm history
	Misc. Settings		Settings for alarm mask, regen resistor, communication, and other settings

2.3.5.1 Connect

Searches for a drive and if it finds one, establishes communication with the SVX Servo Suite.

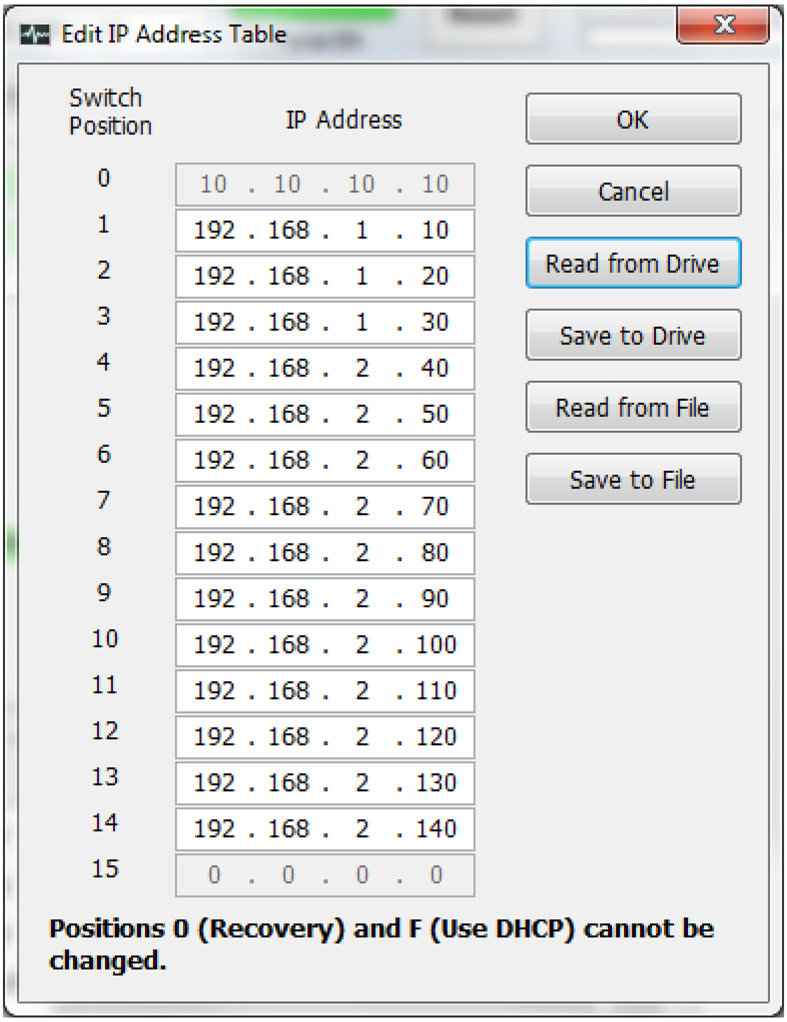
2.3.5.2 Ping

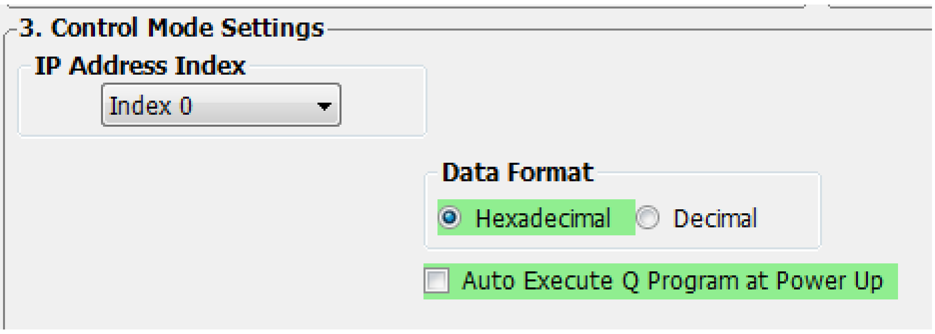
Clicking Drive…Ping will verify your network configuration and ensure that the software can communicate with the drive. If the SVX Servo Suite is able to connect with your drive, the drive's communication processor firmware version (also known as the ARM build number) will be displayed along with the Ethernet MAC ID.

2.3.5.3 IP Table

IP Table is used to edit the IP addresses for the drives with Ethernet ports. Then select which address you want according to its index as shown below.

Note: When downloading the IP address to the drive, the IP address does not take effect until you power cycle the drive.


Edit IP Address Table
Switch Position
IP Address
OK
Cancel
Read from Drive
Save to Drive
Read from File
Save to File
0 10 . 10 . 10 . 10
1 192 . 168 . 1 . 10
2 192 . 168 . 1 . 20
3 192 . 168 . 1 . 30
4 192 . 168 . 2 . 40
5 192 . 168 . 2 . 50
6 192 . 168 . 2 . 60
7 192 . 168 . 2 . 70
8 192 . 168 . 2 . 80
9 192 . 168 . 2 . 90
10 192 . 168 . 2 . 100
11 192 . 168 . 2 . 110
12 192 . 168 . 2 . 120
13 192 . 168 . 2 . 130
14 192 . 168 . 2 . 140
15 0 . 0 . 0 . 0
Positions 0 (Recovery) and F (Use DHCP) cannot be changed.


3. Control Mode Settings
IP Address Index
Index 0
Data Format
Hexadecimal
Decimal
Auto Execute Q Program at Power Up

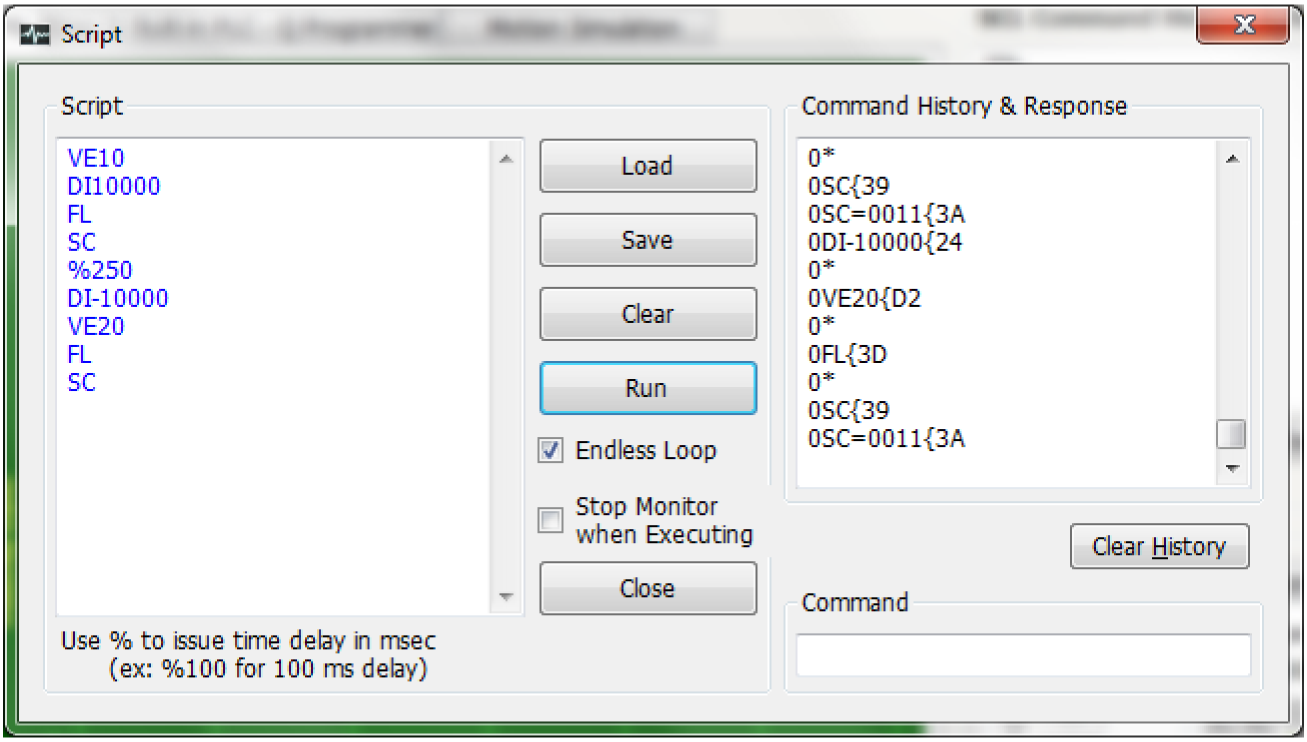
2.3.5.4 Parameter table

This handy feature displays all drive operating parameters in a table.

SEQ	Category	Command	Unit	Software	Drive	Default	Range	Description(Double Click for Details)
000	PID	KP		10000	10000	8000	0 - 32767	Global Gain 1
001	PID	KG		12000	12000	10000	0 - 32767	Global Gain 2
002	PID	KF		22500	22500	6000	0 - 32767	Proprotion Gain
003	PID	KD		16200	16200	2500	0 - 32767	Deriv Gain
004	PID	KV		25000	25000	8000	0 - 32767	Damping Gain
005	PID	KI		360	360	200	0 - 32767	Integrator Gain
006	PID	KK		22681	22681	0	0 - 32767	Inertia Feedforward Constant
007	PID	KJ		5000	5000	5000	0, 10 - 5000	Jerk Filter Frequency
008	PID	VP		15000	15000	15000	0 - 32767	Velocity Loop Proportional Gain
009	PID	VI		600	600	1000	0 - 32767	Velocity Loop Integral Gain
010	PID	KE		15000	15000	15000	0 - 32767	Deriv Filter Gain
011	PID	KC		20000	20000	25000	0 - 32767	PID Filter
012	Control Mode	CM		7	7	21	1-8,11,12,15-18,21,22,25	Main Control Mode
013	Control Mode	CN		21	21	21	1-6,8,11,12,15-18,21	Second Control Mode
014	Control Mode	PM		2	2	2	2, 5, 7, 8, 9	Power-up Mode
015	Control Mode	JM		1	1	1	1 - 2	Jog Mode
016	Current Config	GC	0.01A	0	0	0	-180 - 180	Current Command
017	Current Config	CC	A	1.800	1.800	0.500	0.000 - 1.800	Max Current
018	Current Config	CP	A	5.400	5.400	1.500	0.000 - 5.400	Peak Current
019	Current Config	HC	A	1.800	1.800	1.500	0.000 - 1.800	Current in Hard Stop Homing
020	Trajectory	VM	rps	60.000	60.000	60.000	0.025 - 100	Max Velocity
021	Trajectory	AM	rps/s	3000.000	3000.000	3000.000	0.167 - 5000	Max Accel
022	Trajectory	JS	rps	1.000	1.000	10.000	0.025 - 100	Jog Speed

2.3.5.5 Script

You can use the scripting to write sequences of SCL commands and execute them as if they were streaming from a host controller. The Endless Loop function allows you the run the script continuously, until stop is clicked. This is a good way to learn about SCL streaming commands before writing your host software



Checking "Stop Monitor when Executing" will decrease software delays on your PC and make the communication timing behave more like a real host program running on a PC or PLC. It will also disable the real time display of registers and I/O on your SVX Servo Suite screen.

2.3.5.6 Restore Factory Default

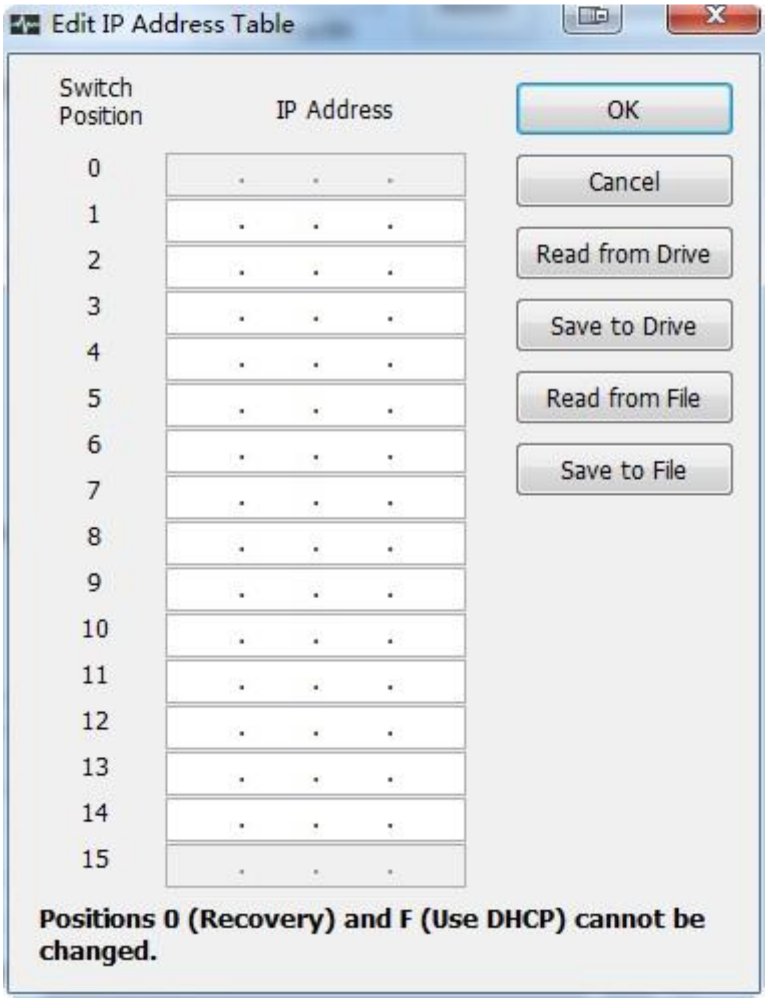
The Restore button will reset all the parameters on the drive to the default factory settings. If you get into a situation where your drive is acting strangely, restoring factory defaults can make your drive work the way it did when it first came out of the box.
Note: This will erase all the parameters you have changed, so you may need to save them to a file first.

2.3.5.7 Restore Tuning Default

This Restore button will reset all the tuning parameters of the drive, but leave other settings unchanged.

2.3.5.8 Alarm History

SV200 series drives store a log of alarm conditions. Each time there is an alarm, the drive stores the information of which alarms were triggered at that time. Since a fault may trigger more than one alarm condition, the drive stores all of them for reference. This information can then be extracted using SVX Servo Suite or the Host Language to help with drive and system problem solving. The drive stores up to 8 sets of alarm conditions.

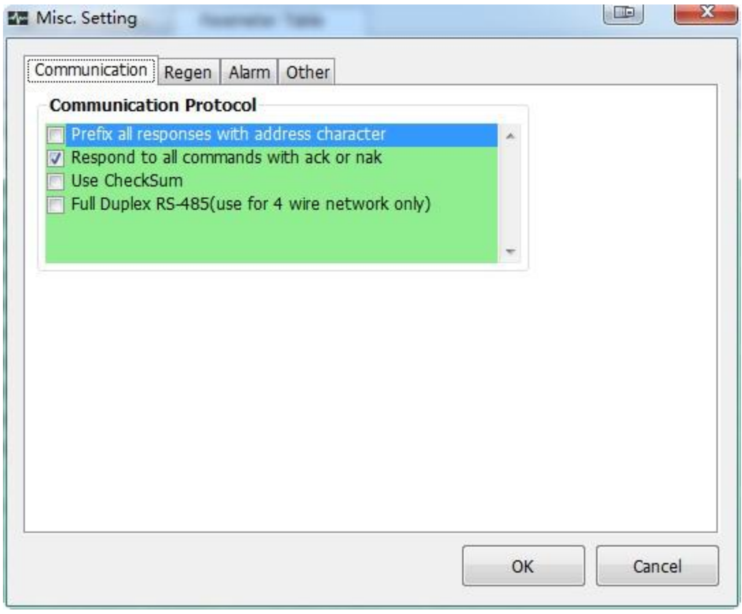


2.3.5.9 Misc. Setting

This dialog is used for defining the alarm mask, regeneration resistor, communication and other parameters

A. Communication

This tab defines specific communication settings between the host controller and your SV200 Series servo drive.



Prefix all responses with address character: Instructs driver to prefix all responses to SCL commands with its address character. Useful for multi-axis networks.

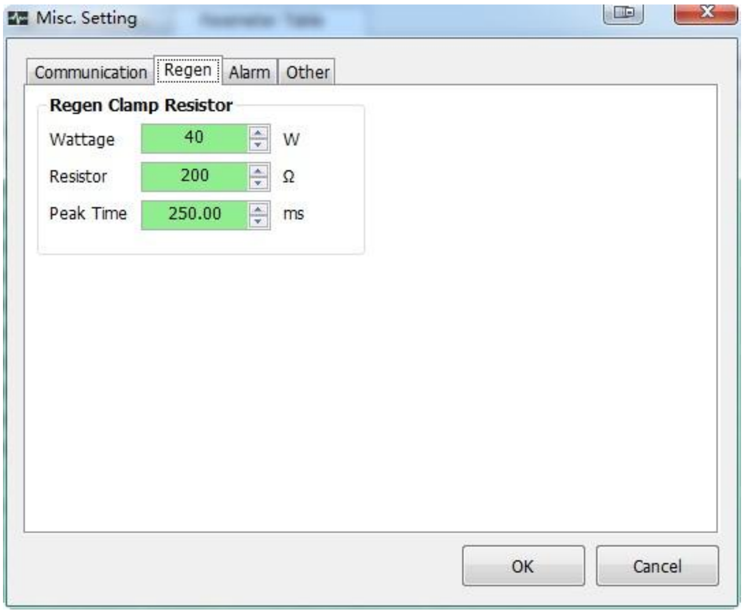
Respond to all commands with Ack or Nack: Respond to all commands with Ack or Nack so you know if the drive received the command and understood it.

Use Checksum: Use Checksum during communication

Full Duplex RS-485: Use this setting with a 4 wire connection network to allow full duplex communication (sending while receiving). For two wire RS-485 networks, do not check this box.

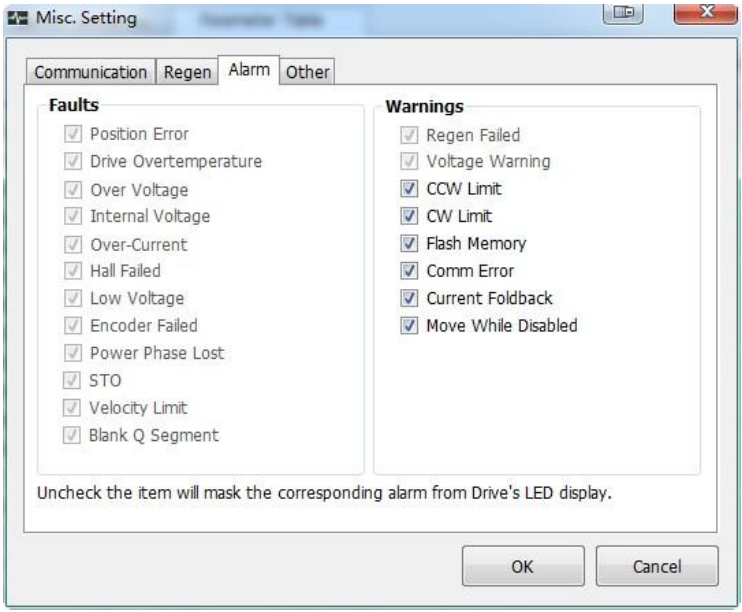
B. Regeneration Resistor

This page will help you to setup an external regeneration resistor.

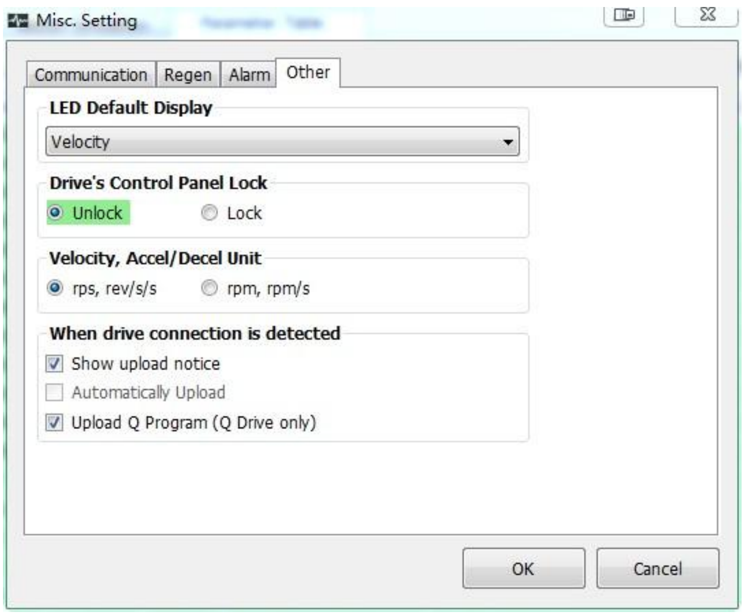


C. Alarm Menu

There are two kinds of alarms: warnings and faults. Faults are critical errors that will disable the drive for protection and show an error pattern on the LED display. Warnings do not disable the drive and are less critical, but still show up on the LED display. You might consider some warnings, such as occasionally tripping and end of travel limit, to be a normal part of your systems operation. In that case, you can uncheck the alarms you want to inhibit; if the drive encounters such alarms, it will not display a warning LED. However, the drive will still store save them in the alarm history for future examination.



D. Other

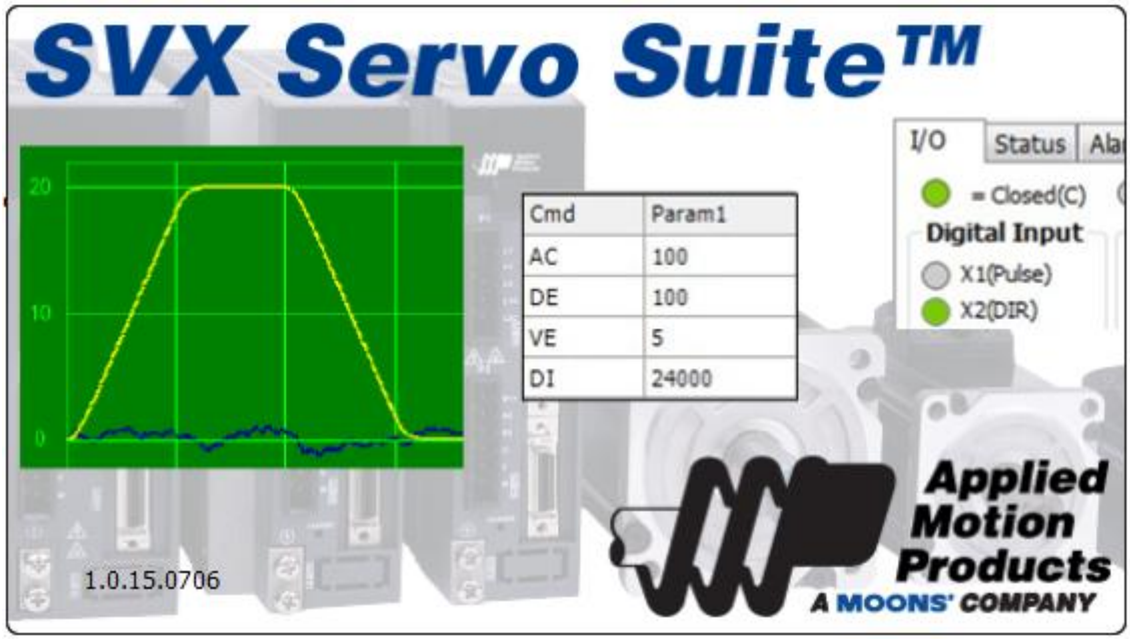


LED Default Display: Sets the default power-up LED Display of the drive.
Drive's Control Panel Lock: Prevents the user from changing any settings thru the front panel controls.
Velocity, Accel/Decel Unit: Defines the units used for velocity, acceleration and deceleration. You can choose: rps (revolutions per second) and rev/s/s or rpm and rpm/s/s.
When drive is connected: If you want to automatically upload your drive's configuration and tuning parameters when the drive is connected to SVX Servo Suite, check the box marked **"**Automatically upload". If you also want your drive's Q program to automatically upload upon connection, check "Upload Q Program".

2.3.6 Help

A. About

Displays the software splash screen and revision.



B. Help Content

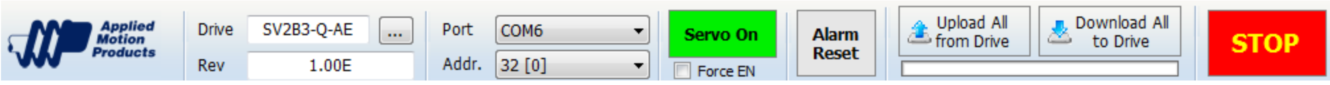
Click for help.

2.3.7 Language

The Language button allows you to choose English or Chinese.

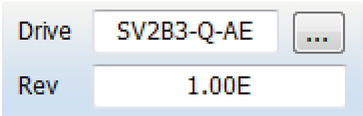
2.4 Tool Bar

The Tool Bar includes Drive Model, Firmware Revision, Communication Port and Address, Servo Status control, Alarms, Upload & Download buttons and the Stop button.



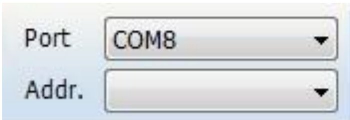
2.4.1 Drive Model

The Drive drop-down list shows all of the available SV200 Series AC Servo drive model numbers.
The Revision window will display a drive's firmware version once the drive is properly connected to the PC and power is supplied.

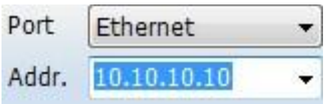


2.4.2 Communication Port

You must choose the communication port to which you have connected the drive before any configuration, tuning or programming can take place. For RS-485 networked drives, it allows you to specify the address of the drive to connect with. For RS-232 drives, select the “blank” address at the top of the list.



For Ethernet drives, the drive’s IP address need to be entered.



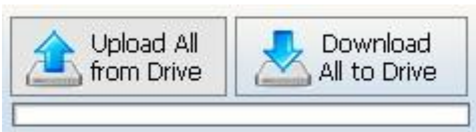
2.4.3 Servo Status

The servo enable switch enables and disables the servo system. When green color is shown, the motor is enabled and the drive’s servo system is active.

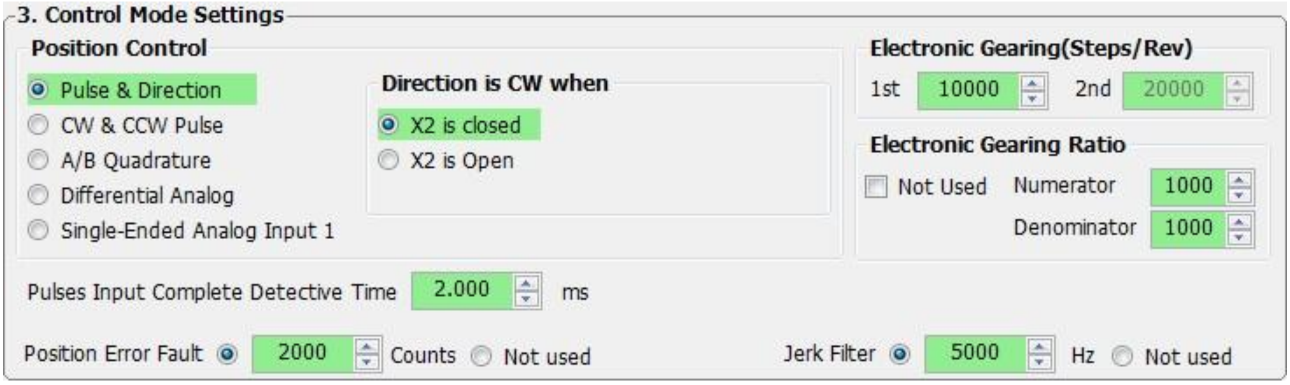
Force enable allows you enable the drive when drive is connected regardless of the external enable switch status.
Alarm reset If an alarm occurs, you can click “Alarm Reset” to clear it.
NOTE: Alarms can only be cleared when drive’s warning or fault problems are solved.

2.4.4 Upload and download

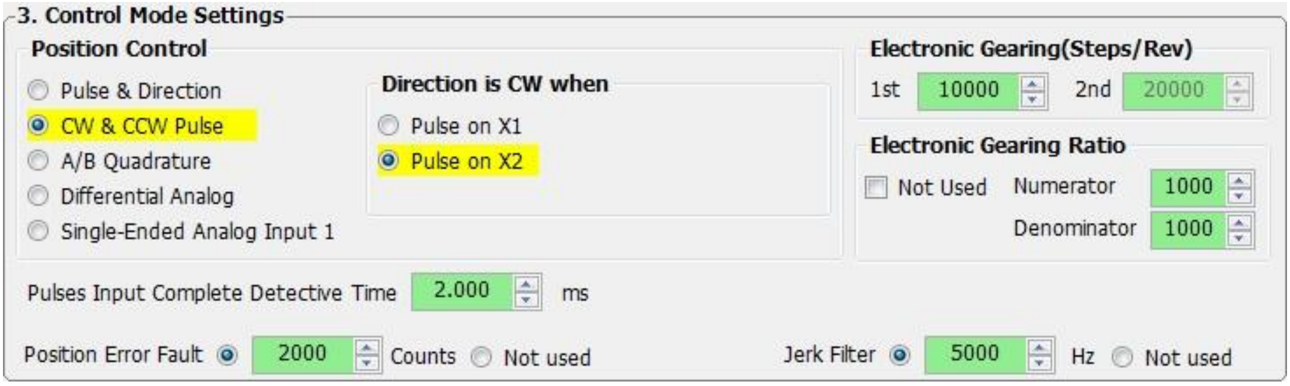
Upload lets you copy the set up and tuning parameters from your dive into SVX Servo Suite software. This is useful if you want to make changes to a system that has already been configured and tuned.
The Download button is used to send the settings from the SVX Servo Suite software to your drive. Use this if you make a change to a drive setting and want to transfer the information back to the drive.



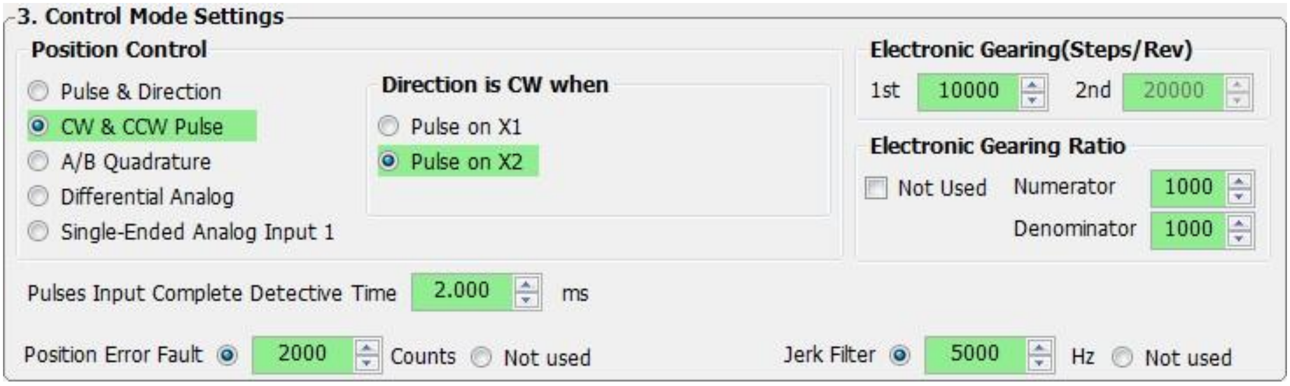
"Upload All from Drive" and "Download All to Drive" will upload or download the whole project (tuning, configuration and Q program). After performing an upload or download, the background of each parameter will turn green. This indicates the parameter in software matches what's in the drive.. See below.



If you then change a parameter in the software, the background of that parameter will change to yellow, indicating that the value of the parameter in the software and drive are now different. See below.



Performed a download after any parameter changes will transfer those values to the drive and the background of the parameter will turn back to green in the SVX Servo Suite, as shown below.



If the driver is not powered up or not connected to the software, or a single upload or download is not performed after the driver is powered up or connected to the software, the background color of parameter is transparent or white, in which means software and driver has not been synchronized by an Upload or Download.

2.4.5 Stop

Stop drive's motion immediately.

3 Using SVX Servo Suite for Configuration

Steps for drive configuration with SVX Servo Suite

Step 1: Configuration

Configure motor information, control mode, as well as I/O settings

Step 2: Servo tuning

Tuning the driver tuning parameters to fit your motion requirements

Step 3: Q programming

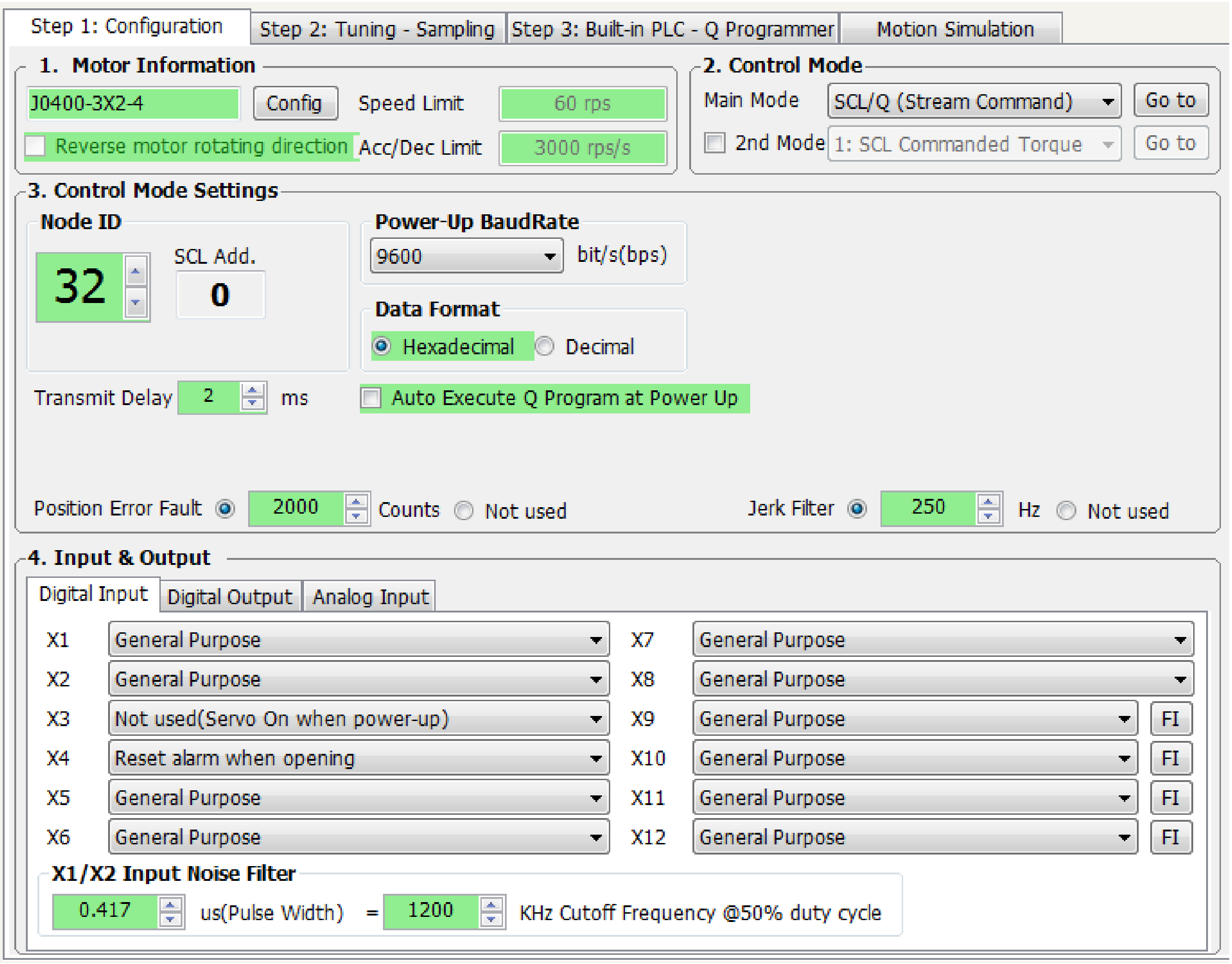
Q programming and debugging

Step 4: Motion simulation

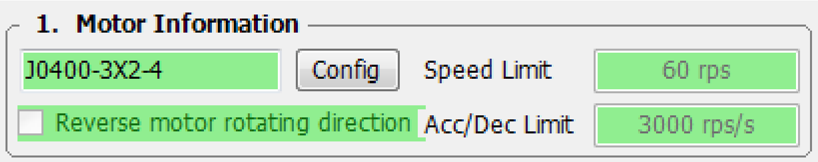
Use to simulate motion, including jog mode, P-to-P motion and homing mode.

3.1 Configuration

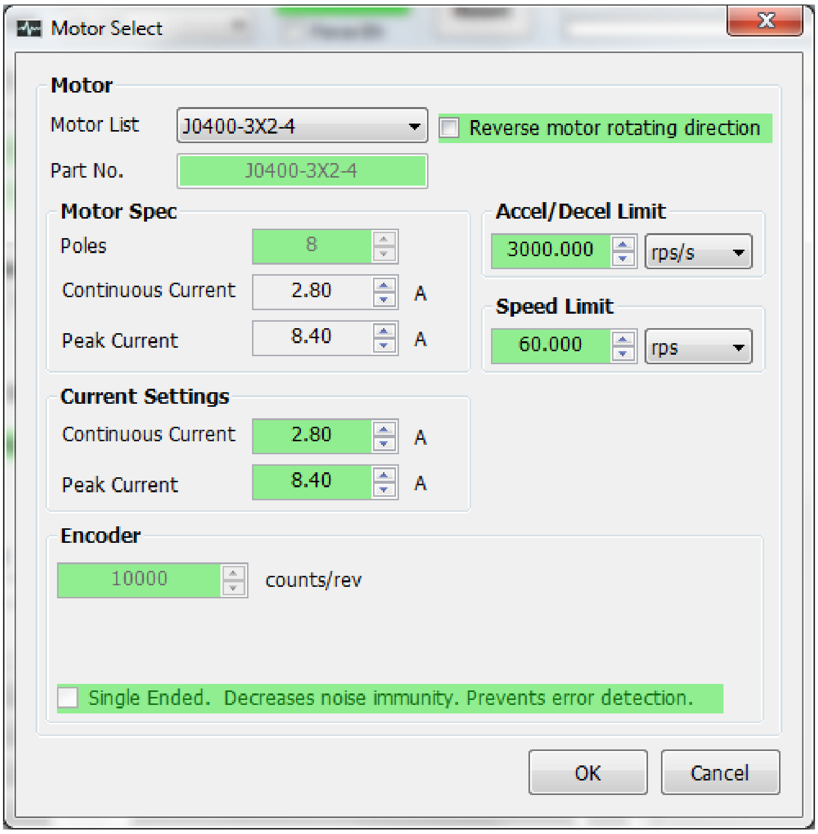
In this tab, you can configure the drive for your motor, select a control mode and define the functionality of inputs and outputs.



3.1.1 Motor Configuration



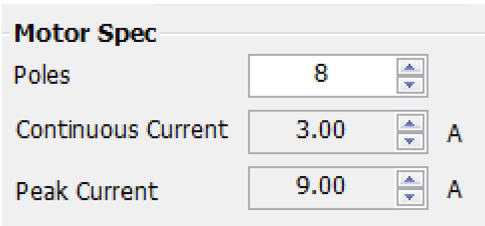
To select the motor, click the Config button. You will a dialog that allows you to choose a motor and set the maximum current, speed limit, and accel/decel limit. If you need the motor to rotate in the direction opposite to normal, check the box marked “Reverse motor rotating direction”. (A power cycle is necessary before the direction change is valid)



NOTE: The motor model number must be the same as the motor that is connected or damage could occur.

3.1.1.1 Current Settings

If you are using an Applied Motion products J series servo motor, the current setting will be populated automatically when you select the motor from the list. If you are using a motor that is not on the list, choose Custom, and then enter the motor’s rated peak and continuous currents. You’ll also need to enter the motor’s pole count. This value should be total poles, not “pole pairs”.



The SV200 servo drive provides a peak current momentarily if needed to achieve the required torque to move the load that is attached to your motor. This will provide greater acceleration rates than would otherwise be possible.

To assure reliable motor operation, the drive will automatically ramp the current down after one second so that the average current does not exceed the motor's rating. Never continuously operate a servo motor above its rated current.

3.1.1.2 Maximum Speed

Here you can enter the maximum speed allowable in your application. If your maximum speed is set below the speed your command signal can demand, the final speed achieved will be the speed set in the Maximum Speed parameter.

Note: Maximum Speed works with Velocity Mode and Torque Mode Only.

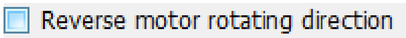
In Pulse Input Mode these values will be limited in your controllers' software.

3.1.1.3 Maximum Acceleration/Deceleration Limit

This will set the maximum level of acceleration for the motor. Even if the command input tries to demand a higher level of acceleration, the drive will only accelerate at the maximum set level.

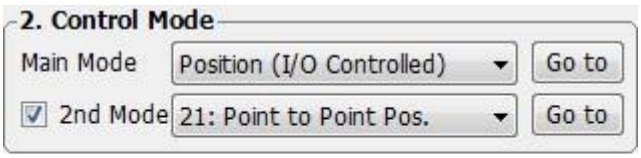
3.1.1.4 Reverse motor rotating direction

If this is checked, the motor rotating direction will be reversed without any other changes. Changes to this value only take place after you power cycle the drive.



3.1.2 Control Mode Selection

For –S and - Q drive models, you can choose two types of control mode, based on your needs.



Main mode options:

- Position (I/O Controlled)
- Velocity (I/O Controlled)
- SCL (Streaming Commands)
- Torque
- Position Table (-S type only)
- Stored Q program (Q models only)
- Modbus (Q models only)

Second mode options:

- SCL Commanded Torque
- Analog Torque
- Analog Torque & Direction
- Analog Torque+R/S
- Analog Torque+R/S+Dir

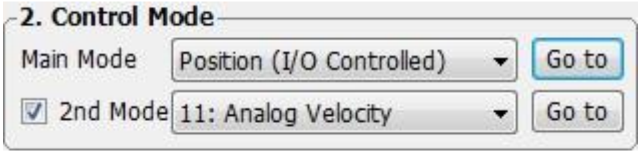
- Analog Velocity
- Analog Velocity+R/S
- Fixed Velocity
- Fixed Velocity+R/S
- Fixed Velocity+CS
- Fixed Velocity+R/S+CS
- Point to Point Positioning

Control mode switch

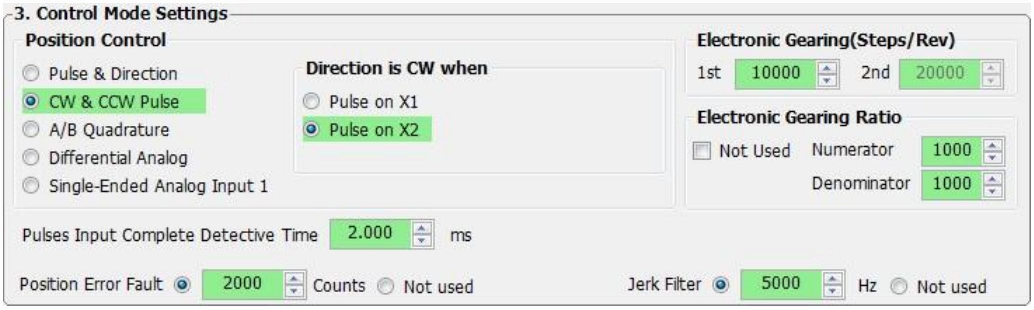
On drive where a second control mode is allowed, input X8 is used to switch between two main and 2nd control modes. In the software, click "Go to" for mode selection.

For example:

1) Set the main mode as Position (I/O Controlled) and set the second mode to Analog Velocity.



2) Click the "Go to" button beside the main mode list and you will see these options for the main control mode:

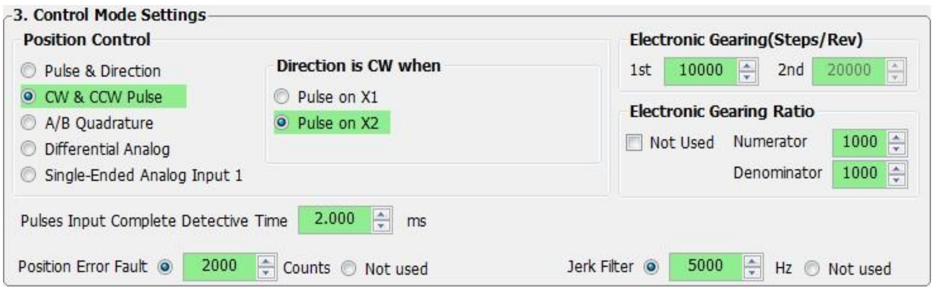


3) Click the "Go to" button next to the 2nd mode list and you can see these settings for the secondary mode

3.1.3 Control Mode Configuration

3.1.3.1 Position Mode (I/O Controlled)

Position mode has five control inputs: Pulse & Direction, CW&CCW Pulse, A/B Quadrature, Differential Analog, and Single-Ended Analog Input 1.



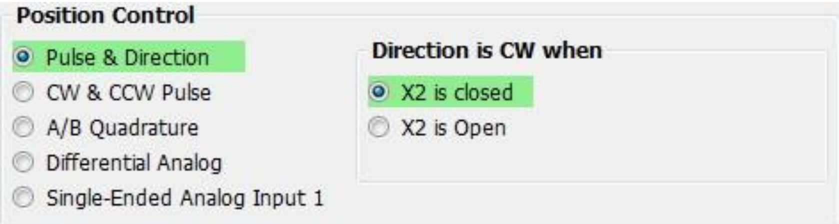
3.1.3.2 Position Control - Pulse Input

Pulse Input Mode is for systems whereby the position of the motor is determined by a digital input signal in the form of pulses.

The three modes available are:

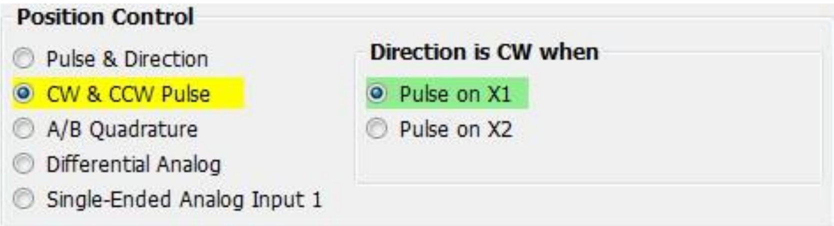
Pulse & Direction

This mode accepts step and direction signals from a PLC or motion controller. The frequency of the pulses determines the motor speed while the number of pulses controls the position; the direction of rotation is determined by a signal fed into another input. You can configure whether X2 signal closed or open represents clockwise motion.



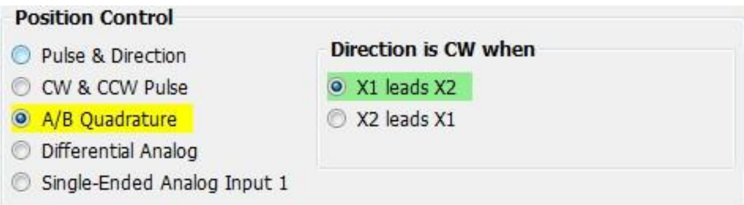
CW & CCW Pulse

The motor will move CW or CCW depending on which input the pulse is fed into. The drive has two inputs allocated to this feature, pulses fed into one input will generate CW motion, and pulses fed into the other input will generate CCW motion.



A/B Quadrature

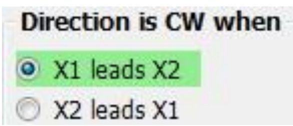
This is sometimes called "slave mode" or "mast encoder following". The motor will move according to signals that are fed to the drive from a master encoder. This encoder can be mounted on a shaft on the machine or it can be another motor in the system. Using quadrature input mode it is possible for a number of motors to be "daisy chained" together with the encoder output signal from each drive being fed into the next.



For all the pulse input modes you will need to determine a value to enter into the Electronic Gearing Box. An explanation on how to do this is given in the next section.

Direction is CW when

If the motor's direction of rotation in A/B quadrature mode is the opposite of what you want, select the other option for direction.



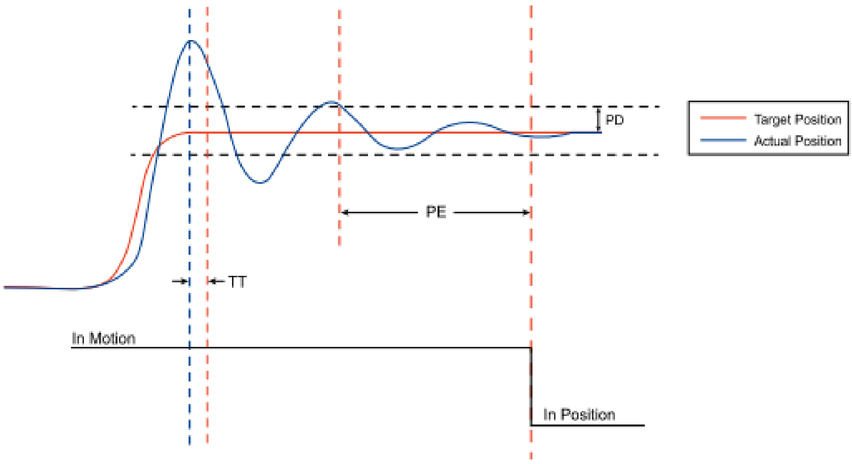
Jerk Filter

In pulse & direction mode, if the pulse train from the controller suddenly changes from a constant speed (or no speed) to acceleration mode, the abrupt change causes a “jerk” in the mechanical system. For many applications this is undesirable because it increases wear and tear on the mechanical linkage between motor and load. If you are handling fluids, jerk can cause them to “slosh around” or spill. It might even tear the backing on a roll of labels. Enabling the SV200’s jerk filter can prevent all these problems by providing a smooth transition between acceleration and constant speed. The jerk filter is specified in hertz (Hz). Smaller values provide smoother motion.

The jerk filter will introduce some time delay into your motion profile, but it doesn’t affect the positioning accuracy. The jerk filter setting is located on the Configuration tab, as part of the Control Mode Settings.

Pulse Input Complete Detection Time

When a servo drive generates its own motion profile like it does in point to point position mode, it knows when the move is over and if the target position has been reached. But when the motion is commanded from an external controller using pulse & direction, how does it know? For the SV200 servo drive, you can define a “Pulse Input Detection Time”. If the drive doesn’t receive any pulses during this period of time, the move is considered to be over and the SV200 can determine whether the motor is at the target position or not. This parameter can also be set using the SCL “TT” command. For details, please see the Host Command Reference.



3.1.3.3 Position Control - Analog

Positioning mode using an analog input causes the motor to position the motor relative to the analog input value.

Analog positioning allows you to move the motor a relative distance according to the value of an analog input. For example, the configuration below would move the motor +/-8000 counts from its current position according to the voltage applied, e.g. a signal of +5 volts would move the motor 8000 counts clockwise.
There is also an option for an offset voltage and a dead-band. The offset can be used to offset the position in case the 0 volt signal from your analog command does not represent the zero position on your application.

3.1.4 Velocity Mode (I/O Controlled)

In Velocity Mode the drive uses command signals to determine the motor speed.

Accel: Sets the acceleration to be used in velocity mode.

Decel: Sets the deceleration to be used in velocity mode.

There are several other options for velocity mode.

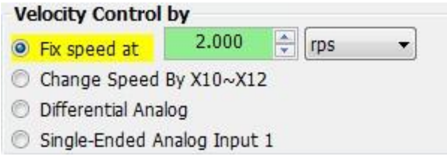
Velocity Control Type: Options are Speed Only (true velocity mode) or Position Over Time (where position mode used to run the motor at a constant speed).

- **Speed Only:** Velocity proportional gain and velocity integral gain are used in this mode and must be set. Position error limit is not used so position lag will not fault the drive
- **Position over Time:** Position error limit must be set because excessive position lag can fault the drive.

Velocity Control By: Four velocity sub modes can be used as explained the next sections.

3.1.4.1 Fixed Speed

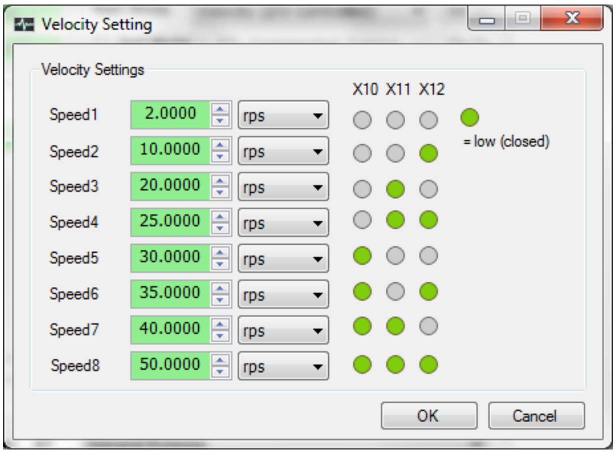
The motor will run at a fixed speed. Run/stop and direction are controlled by digital inputs.



3.1.4.2 Change speed level by X10~X12

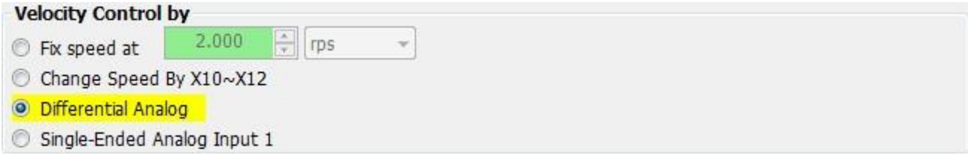
Three digital inputs are assigned and allow you to select from 8 speeds

Click the Velocity Setting button to enter the 8 speeds.



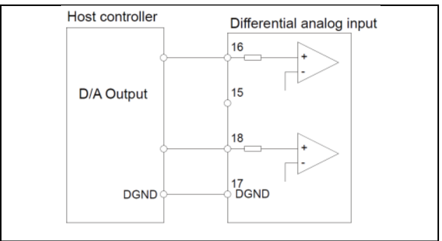
3.1.4.3 Analog Velocity Mode

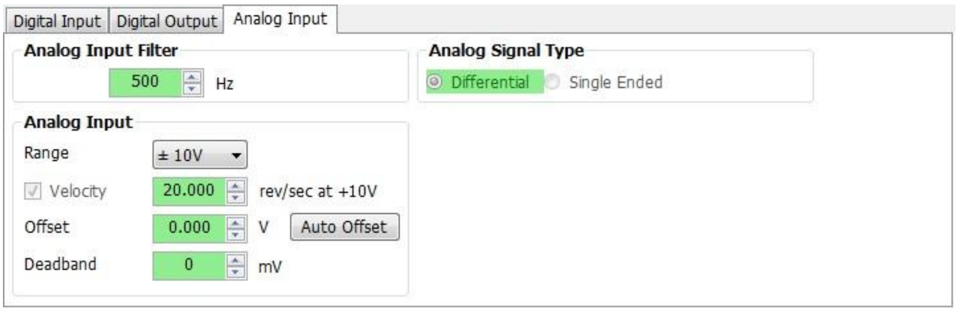
Analog velocity mode has two options: Differential Analog and Single-Ended Analog Input 1.



1) Differential Analog

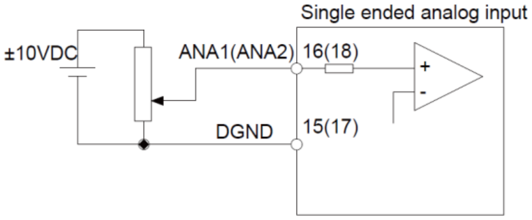
In this mode, the motor speed is determined by the difference between the voltages applied to analog inputs 1 and 2. Using a differential signal can reduce interference from electrical noise and make the speed more accurate.

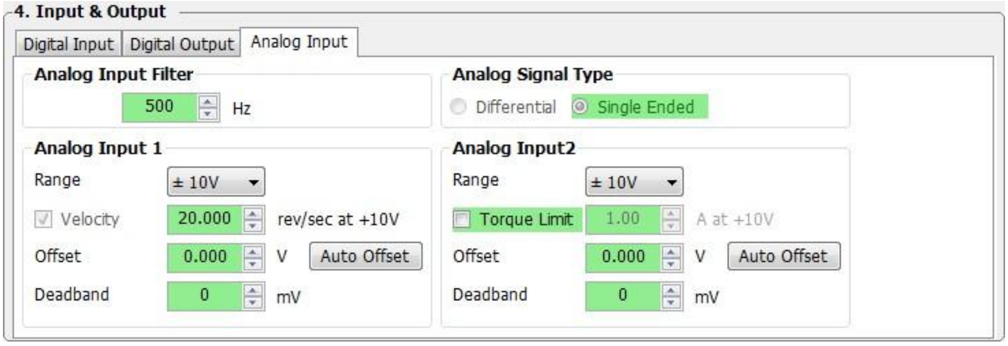
Input Type	Type	Pin NO.	Functions	
Analog Input	ANA1	16	Differential Analog Input	
	ANA2	18		
	DGND	15	Digital Ground	



2) Single-Ended Analog Input 1

In this mode, the motor speed is determined by the voltage applied to analog input 1



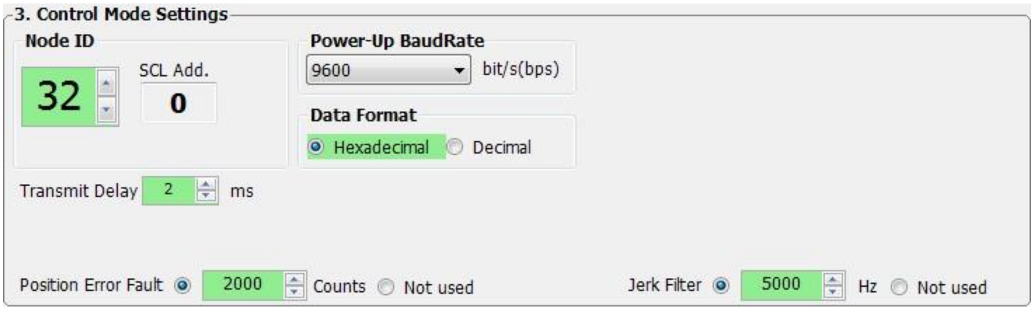


3.1.5 SCL /Q Mode (Streaming Commands and/ or Stored Program)

3.1.5.1 SCL

SCL or serial command language, gives users a simple way to control a motor drive via a serial port. This eliminates the need for separate motion controllers or to supply control signals, like Pulse & Direction, to your step and servo motor drives. It also provides an easy way to interface to a variety of other industrial devices like PLCs, industrial computers, and HMIs, which most often have standard or optional serial ports for communicating to other devices.

SCL is our host command language for applications that require the drives to be sent instructions by a host controller in real time. SCL command can be sent to SV200 series drives with RS-232, RS-485 or Ethernet communication ports. The RS-485 option allows you to have multi-axis multi-drop applications with the drives "daisy chained" on one serial link. When this option is selected you will need to set an address for each drive you are working with. Refer to Setting the Address in the next section.



Node ID

In SCL mode with RS-485 communications you will need to set the address for each drive in your system. Simply select the address character and perform a download, in this way up to 32 drives can be connected together on a single serial link.

Transmit delay

This sets up the transmit delay for communications between host controller and the drive. This is highly necessary for 2 wire configurations for RS-485 communication. The host must disable its transmitter before it can receive data. This must be done quickly before a drive begins to answer a query.

Baud rate

At power up a drive will send its power-up packet detected after 1 second and the drive is configured for SCL or Q operation (see PM command) the drive will set the baud rate according to the value stored in the Baud Rate NV parameter. This parameter will not effect immediately, it will only effect at next drive power up.

Data format

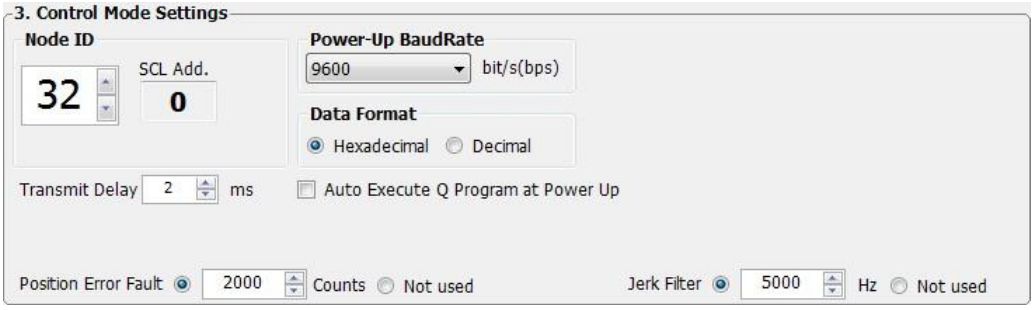
To setup data transmit type between Hexadecimal and Decimal.

3.1.5.2 Q Program

Q takes the SCL language to an entire new level by allowing users to create programs that can be stored in the drive's flash memory. The drive can be set to automatically executed programs at power up or it can wait for the programs to be started and stopped using streaming commands over RS-232, RS-485, Ethernet, Modbus, CANopen or EtherNet/IP connections. Q programs can also be triggered using the drive's built-in I/O.

Q programs offer

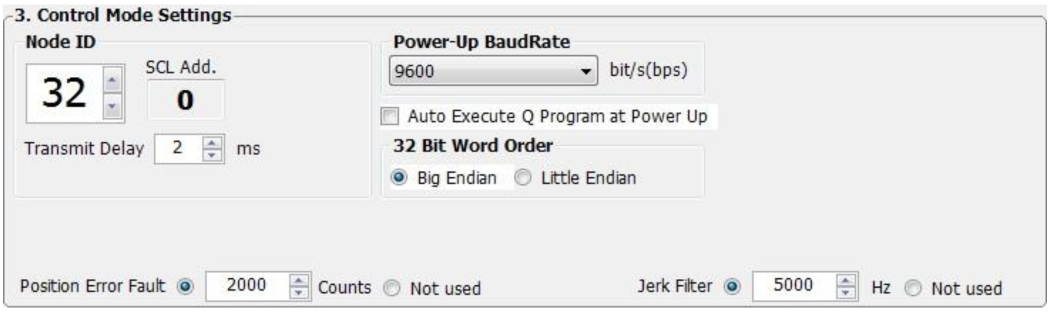
- Single-axis motion control
- Stand alone or networked operation
- Multi-tasking
- Conditional processing and program flow control
- Math calculations
- Data register manipulation
- User interaction via touch screen HMI's



Auto Execute Q Program at Power Up

If this box is checked, the drive will execute a stored Q program from segment 1 automatically at power up.

3.1.6 Modbus/RTU



Node ID

Each device on a Modbus network requires a unique address. Only the device with the matching address will respond to a given host command. In a Modbus network, address "0" is the broadcast address. It cannot be used for an individual drive's address. Modbus/RTU allows drive addresses to be set from 1 to 32. The SCL address is an ASCII code. The relationship between the Modbus Node ID and the SCL address is shown in the table below. We're telling you this in case you want to set or check the drive address using the SCL "DA" command. That command works with ASCII SCL addresses, so you'll need to know the corresponding Modbus address. If you use addresses 1 through 9 it keeps things simple because the Modbus and SCL addresses in that range are the same.

Node ID	1	2	3	4	5	6	7	8
SCL Address	1	2	3	4	5	6	7	8
Node ID	9	10	11	12	13	14	15	16
SCL Address	9	:	;	<	=	>	?	@
Node ID	17	18	19	20	21	22	23	24
SCL Address	!	"	#	$	%	&	'	(
Node ID	25	26	27	28	29	30	31	32
SCL Address	)	*	+	,	-	.	/	0

Auto Execute Q Program at Power Up

If this is checked, the drive will execute a stored Q program from segment 1 automatically at power up.

32 bit word order

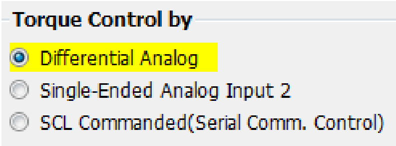
Big-endian: The most significant byte (MSB) value is stored at the memory location with the lowest address; the next byte value in significance is stored at the following memory location and so on. This is akin to Left-to-Right reading in hexadecimal order.

Little-endian: The most significant byte (MSB) value is stored at the memory location with the highest address; the next byte value in significance is stored at the following memory location and so on. This is akin to Left-to-Right reading in hexadecimal order.

3.1.7 Torque Mode

When the drive is set for Torque mode, it allows you to define the current that will be delivered and thus the torque generated by the motor and the direction of rotation. In this mode the motor speed will depend on the load applied to the motor.

WARNING - If the motor is not connected to the load or has no load applied, downloading this mode while a command signal is present may cause the motor to accelerate to a high speed.

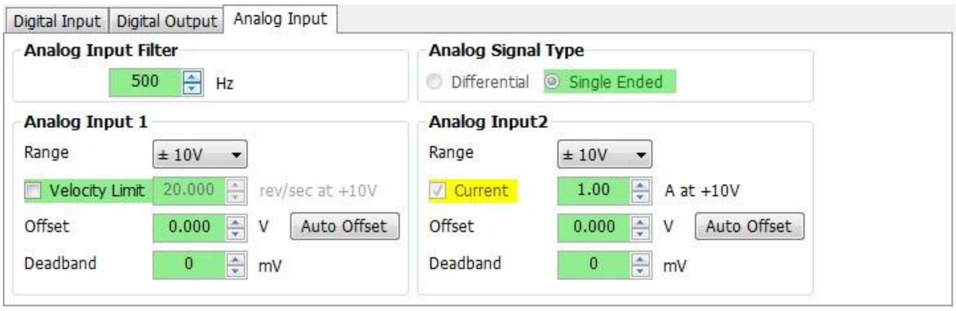


Torque mode has two control options: Analog and SCL Commanded.

3.1.7.1 Analog

There are four settings that are required for getting the analog inputs to control the desired mode output:

- Range: ±10V is the only choice.
- Current: Establishes a gain value that scales the output from the input. For example, if current is set to “1A at +10V”, a +10 volt input signal will tell the drive to apply 1 amp to the motor. A 2 volt input would apply 0.2 amps to the motor.
- Offset: Sets an offset value to the input that can null out a voltage bias or it can shift the input voltage value as needed. Often in analog systems it is very difficult to get a true “0” value. Using the offset feature allows adjusting out any unwanted offsets that disturb the desire for a true 0 volt input from an external controller. The “Auto Offset” function can automatically detect and correct voltage biases on the input. Click the button and follow the instruction to accomplish this task.
- Deadband: Defines a voltage region where the input is seen as “0”. Because of the sometime imprecise nature of analog signals and inputs there may be a need to create a “dead” zone where the analog input has no effect on the output. This is normally needed around the “0” input. For example, when using a joystick to operate the motor the user may not want any torque output when the Joystick is at its “null” position. Most joysticks are not that precise and may still output a small voltage, adding the dead band can eliminate the effect of the small voltage.



Analog Torque mode has two analog input options: single-ended input 2 and differential mode.

1) Differential Analog Input: command voltage is the difference between the signal at Analog Input 1 and that of Analog Input 2. This provides more immunity from electrical noise.
2) Single-ended Analog Input 2: the command voltage comes from Analog Input 2 only.

3.1.7.2 SCL Commanded

SCL Commanded Mode allows a host controller to send the SCL “GC” command to control the motor’s torque.

3.1.8 CANopen

CANopen is a communication field bus standardized by the CAN in Automation Group (CiA). SV200 servo drives are compliant to CiA 301 and CiA 402 and use the CAN 2.0B passive physical layer. Detailed information on our CANopen implementation can be found on our website.

Node ID

In the CANopen network, each drive needs to have a unique NODE-ID. CANopen node ID addresses are 7 bit binary numbers in the range from 1~127 (hexadecimal 0x01~0x7F).

CAN Bit Rated

SV200 series CANopen drives can support these 8 CAN communication bit rates:

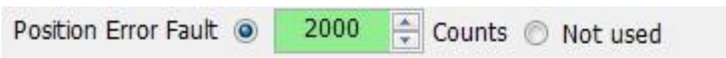
CAN Bit Rated
1Mbps
800Kbps
500Kbps
250Kbps
125Kbps
50Kbps
25Kbps
12.5Kbps

3.1.9 Positioning Error Fault & Electronic Gearing

3.1.9.1 Positioning Error Fault

Positioning error is the difference, in encoder counts, between the actual position and the commanded position of the motor. A small amount of positioning error is a normal part of a servo system. But sometimes the unexpected can happen. A wire might break, a sensor could fail or the motor may encounter a physical obstruction. You might even one day forget to set up and tune a drive before installing it into a system. In all of these cases, you'll want to know that something is wrong as soon as possible and without damaging anything. For this reason, the SV200 servo drives include a position error fault limit. Anytime the position error (as reported by the encoder) exceeds this limit, the drive cuts power to the motor.

You can set the fault limit to as little as 10 encoder counts, or as much as 32000. When you're first tuning the system, you should set this value high or select Not Used so that the drive doesn't shut down as you experiment with tuning parameters. Once the drive is properly tuned and you know how much error to expect during normal operation, you can set an appropriate fault limit. For example: set Quick Tuner's scope to plot position error. Execute some aggressive sample moves, using the maximum speed and acceleration that you plan to use in your application. If the maximum position error is, say, 50 counts, then you could safely set the fault limit at 100.

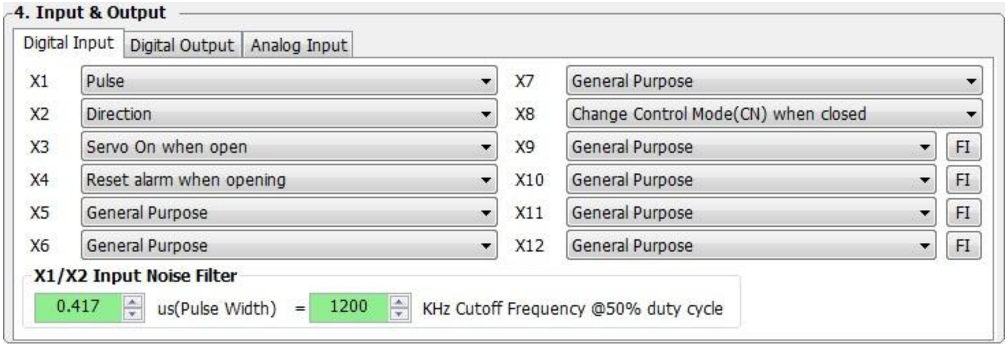


3.2 I/O Configuration

I/O Configuration includes configuration options for digital inputs, digital outputs and analog inputs, each on a separate tab..

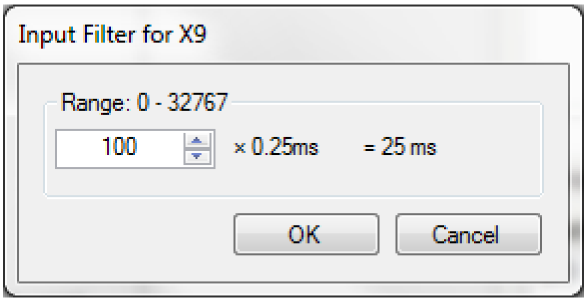
3.2.1 Digital Input Configuration

This tab is used to configure the digital inputs(X1..X12). Please refer to the SV200 User's Manual for details.



3.2.1.1 FI Input filter

This setting defines a low pass digital firmware filter for a given input. The digital input must be at the same level for the time period specified by the FI command before the input state is updated. For example, if the time value is set to 100 the input must remain high for 100 processor cycles before high is updated as the input state. One processor cycle is 250µsec on an SV200 servo drive. A value of "0" disables the filter.



3.2.1.2 Input Noise Filter

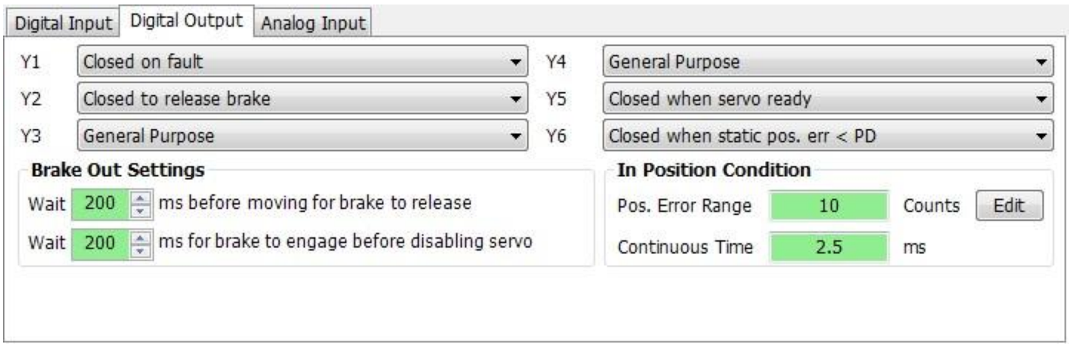
The Input Noise Filter is a low-pass digital hardware filter, rejecting noise above the specified frequency. Or you can set the Pulse Width, and the software will calculate the filter frequency.

3.2.1.3 Digital Input Functions

Signal	Symbol	Pin NO.	Details
X1	X1+	3	This input has three functions: ● In digital position mode, accepts STEP pulses, CW pulses, or Ch. A quadrature signal ● Run/Stop input in torque or velocity mode. ● General purpose input.
	X1-	4	
X2	X2+	5	This input has three functions: ● In digital position mode, accepts DIR signal, CCW pulses, or Ch. B quadrature signal ● Direction input in torque or velocity mode. ● General purpose input.
	X2-	6	
X3	X3+	29	● Enable/Disable input.
	X3-	31	● General purpose input.
X4	X4+	35	● Alarm Reset Input, used to reset drive alarm.
	X4-	34	● General purpose input.
X5	X5+	8	● Limit Sensor Input.
	X5-	2	● General purpose input.
X6	X6+	9	● Limit Sensor Input.
	X6-	1	● General purpose input.
X7	X7+	39	● Gain Select Input in all control mode.
	X7-	38	● General purpose input.
X8	X8+	12	● Switch Control mode between main mode and second mode. ● General purpose input.
	X8-	32	
X9	X9	26	● Dividing Switch, change the pulses per revolution for electronic Gearing. ● General purpose input.
X10	X10	27	● Pulse Inhibited Input. Ignore the pulse input when this input is activated in position mode. ● Speed Selecting Input 1 in change Speed mode. ● General purpose input.
X11	X11	28	● Speed Selecting Input 2 in change Speed mode. ● General purpose input.
X12	X12	30	● Speed Selecting Input 3 in change Speed mode. ● General purpose input.

3.2.2 Digital Output Configuration

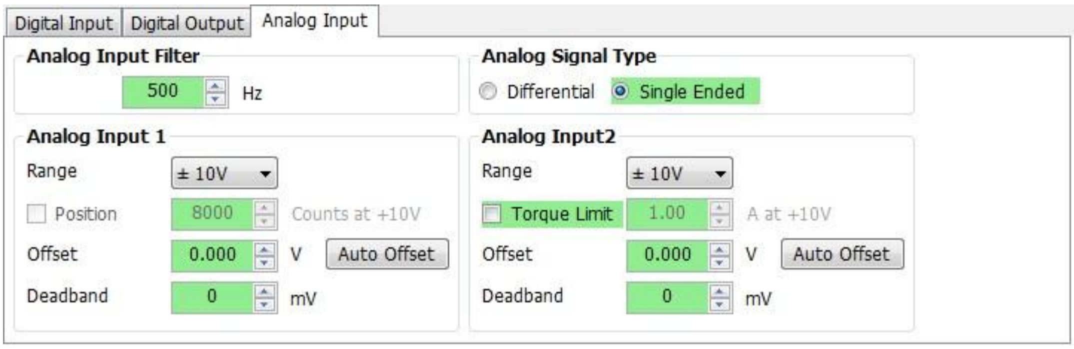
This tab is used to configure the digital outputs(Y1..Y6). Please refer to the SV200 User's Manual for details.



3.2.2.1 Digital Output Functions

Signal	Symbol	Pin NO.	Details
Y1	Y1+	37	This output has two functions: ● Alarm Output. ● General purpose output.
	Y1-	36	
Y2	Y2+	11	This output has two functions: ● Motor brake control output. ● General purpose output.
	Y2-	10	
Y3	Y3+	42	● Torque Reached Output. ● General purpose output.
	Y3-	33	
Y4	Y4+	43	● Moving signal output, output signal when dynamic position error less than set value in position mode. ● Velocity reach output. Output signal when actual speed is same as the target speed and the speed ripple less than ripple range. ● General purpose output.
	Y4-	33	
Y5	Y5+	40	● Servo ready output. Output servo ready signal when the drive is ready to be controlled and without alarm. ● General purpose output.
	Y5-	41	
Y6	Y6+	14	● In position signal output, output signal when in position, and the position error less than set value in position mode. ● Tach out output. Tach output produces pulses relative to the motor position with configurable resolution. ● General purpose output.

3.2.3 Analog Input Configuration



3.2.3.1 Analog Input Filter

The analog input filter sets the frequency in hertz (Hz) of the roll off point of a single pole low pass filter. When using any of the Analog Input modes, this filter can be used to reduce the effects of analog noise on the mode of operation.

3.2.3.2 Analog Input Settings

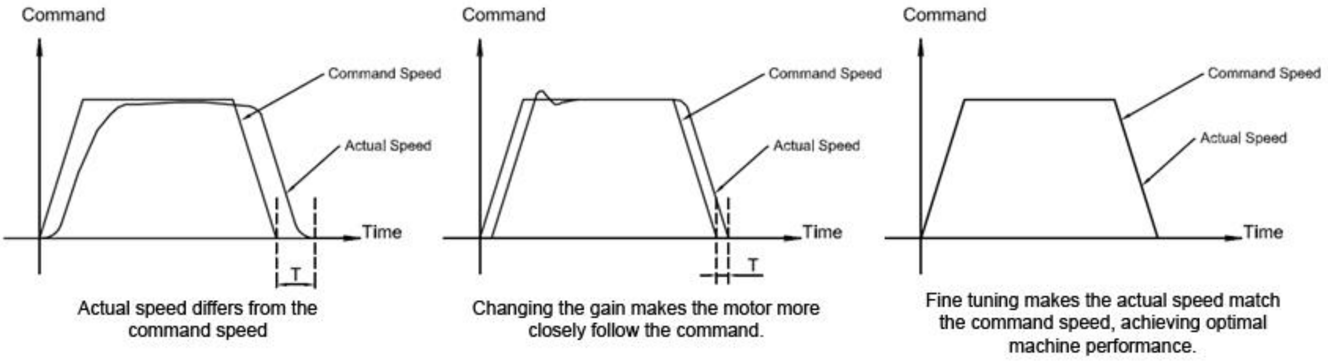
1. Range –±10V.
2. Offset – Sets an offset value to the input that can null out a voltage bias or it can shift the input voltage value as needed. Often in analog systems it is very difficult to get a true "0" value. Using the offset feature allows adjusting out any unwanted offsets that disturb the desire for a true 0 volt input from an external controller. The "Auto Offset" function can automatically detect and correct voltage biases on the input. Just click the Auto Offset button and follow the instructions to accomplish this task.
3. Deadband – Inserts a voltage region where the input is seen as "0". Because of the sometime imprecise nature of analog signals and inputs there may be a need to create a "dead" zone where the analog input has no effect on the output. This is normally needed around the "0" input. For example, when using a joystick to operate the motor the user may not want any torque output when the Joystick is at its "null" position. Many joysticks are not precise and may still output a small voltage, adding the dead band can eliminate the effect of the small voltage.

4 Step 2: Tuning - Sampling

Like most modern servo drives, the SV200 series employs sophisticated algorithms and electronics for controlling the torque, velocity and position of the motor and load.
Feedback sensors are used to tell the drive what the motor is doing at all times. That way the drive can continuously alter the voltage and current applied to the motor until the motor meets the commanded torque, velocity or position, depending on the control mode selected. This form of control is called "closed loop control."
One of the loops controls the amount of current in the motor. This circuit requires no adjustment other than specifying the maximum current the motor can handle without overheating.
The PID loop compares the intended motor position to the actual motor position as reported by the encoder. The difference is called error, and the PID loop acts on this error with three gain terms: the Proportional term, the Integral term and the Derivative term. The Acceleration Feedforward term is also added to achieve greater system control.

4.1 Servo Tuning – Adjustment of Gain Parameters

Servo tuning is used to optimize the servo system's overall performance and reduce system response time. Servo tuning allows the servo motor to execute host control commands more precisely in order to maximize its system potential. Therefore, it is highly recommended that the gain parameters be optimized before actual system operation.



The PID loop compares the intended motor position to the actual motor position as reported by the encoder. The difference is called error. The PID loop acts on this error with these three gain terms: Global gain (KP), Integrator Gain (KI), Derivative gain (KD). In addition to the PID loop control, the SV200 series drives add a number of extra terms to enable greater system control. These additional terms include: position loop gain (KF), Damping gain (KV), Inertia feed forward gain (KK), Follow Factor (KL), Derivative filter gain (KE), and PID filter (KC).

In general, for systems having stiff mechanical transmissions, increasing the servo gain parameters will improve response time. On the other hand, for systems having more compliant mechanical transmissions, increasing servo gain parameters will potentially cause system vibrations and reduce system response time.

4.1.1 Gain Parameter Introduction

Global gain (KP):
This parameter is the primary gain term for minimizing the position error. It defines the system stiffness. Larger KP values means higher stiffness and faster response times. However, if gain values are too high, vibration can result. Values ranging from 6000 to 16000 are commonly used. In general, use default parameter values when possible.

Position loop gain (KF):
This parameter is also used for minimizing the position error. Increasing KF will increase stiffness and reduce settling time. However, increasing this gain term too much may cause system vibration.

Derivative gain (KD):
This parameter is used to damp low speed oscillations and increase system smoothness.

Integrator gain (KI):
This parameter minimizes (or may even eliminate) position errors especially when motor is holding position.

Damping gain (KV):
KV minimizes the velocity error and reduces vibration in position control mode.

Inertia Feedforward Constant (KK):
KK improves acceleration control by compensating for the load inertia.

Follow Factor (KL):
Higher values will reduce system noise and eliminate overshoot, but will reduce the system's dynamic following performance. Lower values will raise system stiffness, but may cause system noise.

Derivative Filter Gain (KE):
The differential control parameters filter frequency. This filter is a simple one-pole, low-pass filter intended for attenuating high frequency oscillations. This value is a constant that must be calculated from the desired roll off frequency.

PID Filter gain (KC):
The servo control overall filter frequency. This filter is a simple one-pole, low-pass filter intended for attenuating high frequency oscillations. The value is a constant that must be calculated from the desired roll off frequency.
Among all the parameters, changes for KP, KE, and KC are NOT recommended after system configuration. Therefore, parameter tuning is based more on KF, KD, KV, KI, KL and KK.

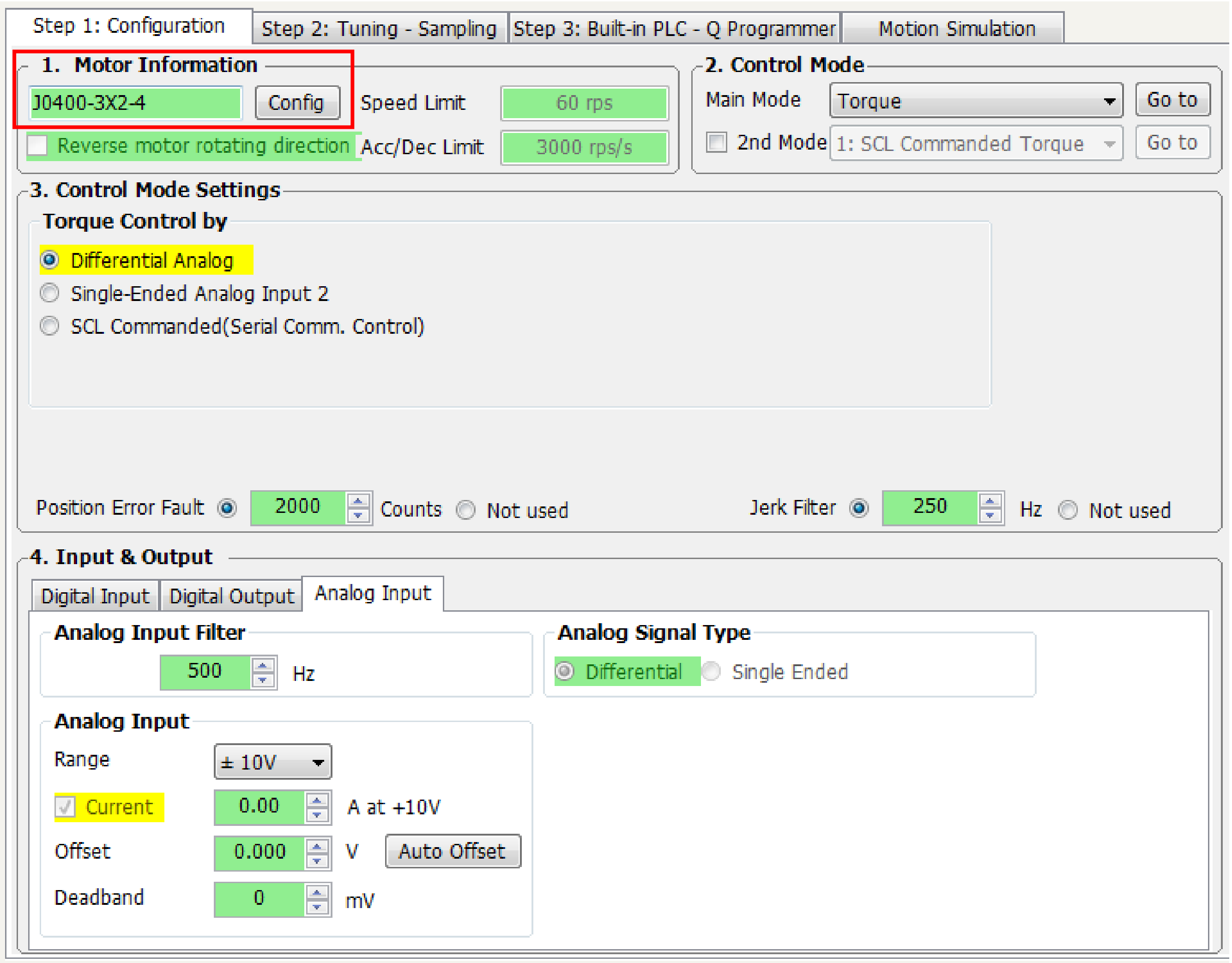
4.2 Auto-Tuning

SV200 servo systems can achieve real time response to the dynamic feedback of the load and optimize tuning parameters automatically. The auto-tuning function can save time and simplify the debugging process. Auto-tuning can be completed using the SVX Servo Suite software in only a few minutes.

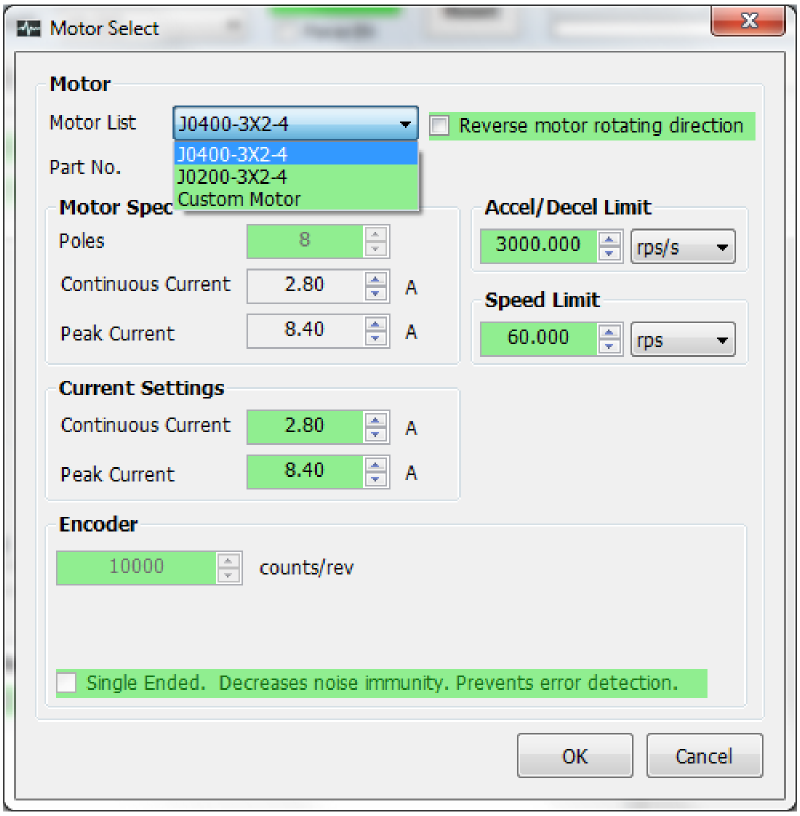
NOTE: Auto-Tuning must operate with the load installed.

4.2.1 Step 1: Select Motor

Before using the auto-tuning, make sure the motor configuration is correct.
On the SVX Servo Suite “**Configuration**” panel in the “**Motor Information**” section, click on “**Config**” (shown below)



In the pop-up menu, click on the drop-down motor list to choose the correct motor number and then click “**OK**”.



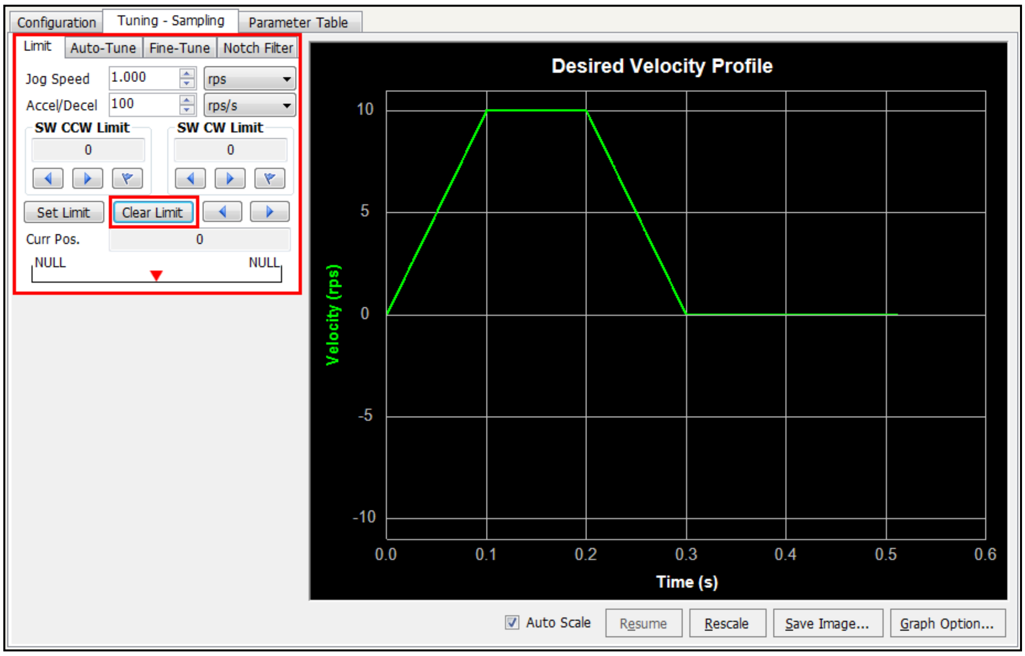
NOTE: Refer to the SV200 Series AC Servo User Manual, Chapter 2.3 Servo Motor Model Introduction for motor identification details.

4.2.2 Step 2: Setting the Software Position Limits

The Software Position Limit function uses encoder counts to set “soft” limits at user-defined locations that can then be used during the tuning process. These position limits ensure that the motor will ONLY rotate between the CCW and CW limits, which will help to prevent accidental system damage. This is especially useful when the motor is coupled to a linear actuator, for instance.

NOTE: The software Position Limits will ONLY be effective during current power-up operation and will not be saved to non-volatile memory for use at the next drive power up. Therefore, DO NOT rely on these software limits during actual system operation. Refer to LP and LM commands in SVX SVX Servo Suite’s built-in Q Programmer help for more details.

On the “**Tuning- Sampling**” panel, select the “**Limit**” tab to setup software position limits. If software position limits are not required, then click “**Clear Limit**” and go to the next step for the “**Auto-Tune**” function description.



4.2.3 Setup Software Position Limits

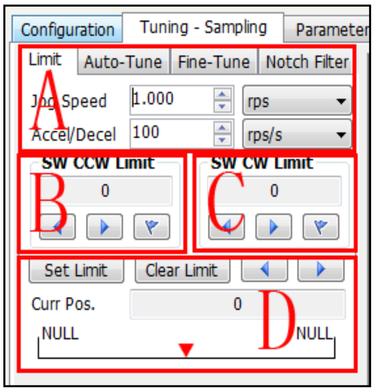
Here are the basic steps to set the soft limits, also shown below:

A. Before rotating the motor and setting limits, first set the desired Jog Speed, and Accel/Decel rate.

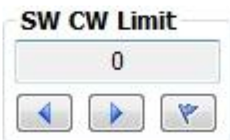
B. Set CCW limit – move to desired position with arrow buttons, then click the flag button

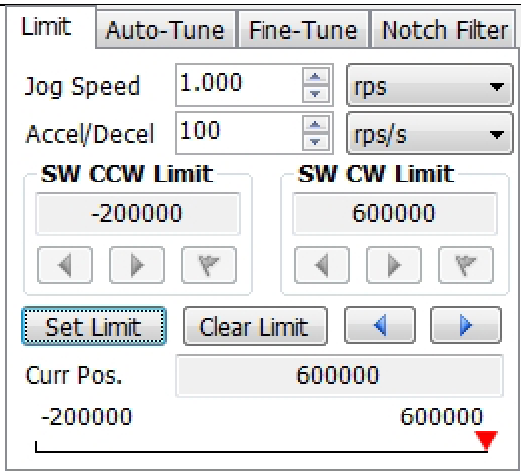
C. Set CW limit – move to desired position with arrow buttons, then click the flag button

D. Confirm or Cancel position limits set in step B and C with the Set Limit or Clear Limit buttons



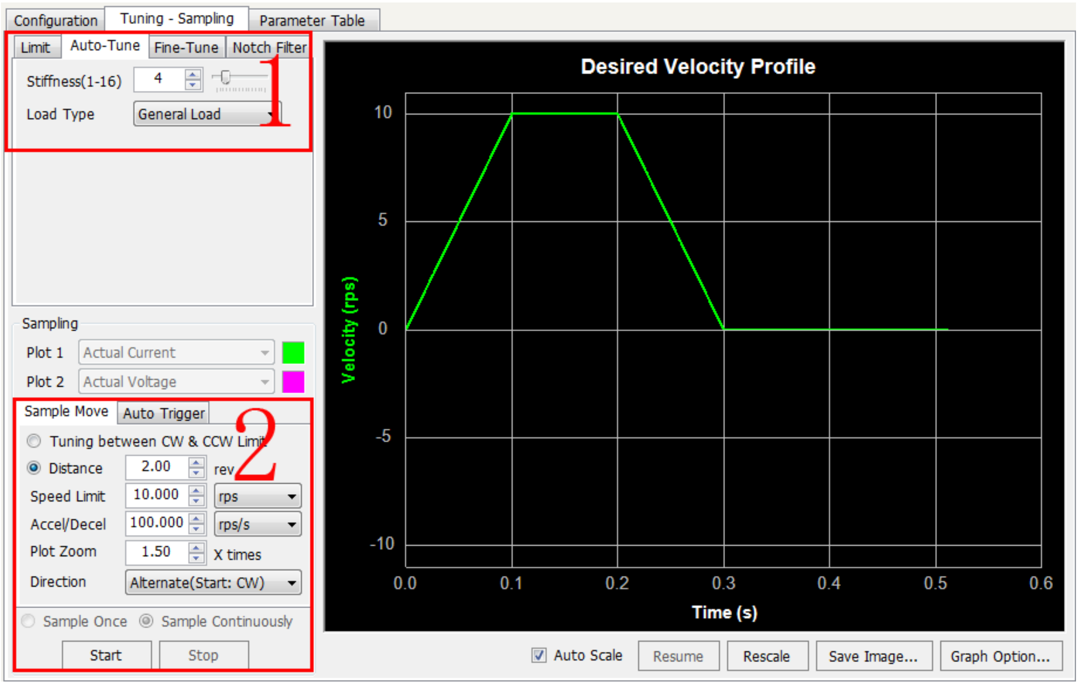
Detailed Steps for Software Position Limit

Step	Operation	Software
1	Make sure Servo is Enabled Click  or  to rotate motor in CCW or CW direction When target position reached, click  to accept and store position	
2	Same process as above	

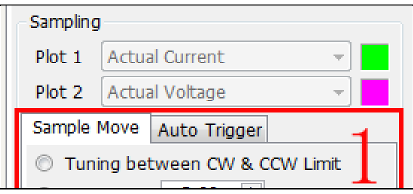
3	Confirm position limits Click on  **NOTE: CW limit must be larger than CCW limit.**	
4	Setting complete	

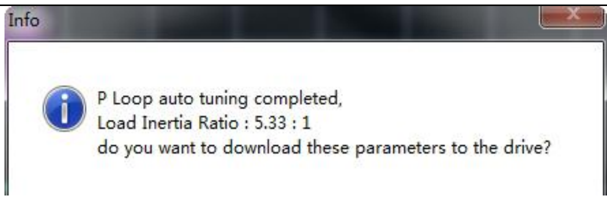
4.2.4 Step 3 Auto-Tuning Function

From the "**Auto-Tune**" tab, follow these steps to configure and run auto-tuning:



Operation steps:

1	Set Stiffness and Load type	
2	Set Auto-Tuning Distance, Speed target, and Accel/Decel **NOTE:** 1) **If software position limit is set, select "Tuning Between CW and**	

	CCW Limit 2) **If no limit is required, select “Distance” (ensure software position limits have been cleared)**	
3	Click **Start** to start the auto-tuning function	
4	When Auto-Tuning is complete, download parameters to the drive	

NOTE: During the tuning process, motor or load vibrations may occur. This is normal and the system will correct itself.

For customized performance requirements, use fine tuning functions.

4.3 Fine tuning

Depending on the mechanical system characteristics and the servo motor used, the following parameters are available and may need to be adjusted to improve system performance:

- **Global gain (KP)**
- **Position loop gain (KF)**
- **Derivative gain (KD)**
- **Damping gain (KV)**
- **Integrator Gain (KI)**
- **Inertia feed forward gain constant (KK)**
- **Derivative filter gain (KE)**
- **PID filters (KC)**

This step should be completed only after the **Auto-Tune** function has been done and if improvements are needed for the tuning. A sample move can be defined and run once for each click of the Start button or continuously to facilitate real-time dynamic tuning (i.e. adjustment of gains and filter settings while the motor is moving).

Among the parameters listed above, changes to Global gain (**KP**), Derivative filter gain (**KE**) and PID filter (**KC**) are NOT recommended after the system has been configured with the **Auto-Tune** function. Therefore, parameter adjustments during the fine tuning phase should be limited to Position loop gain (**KF**), Derivative gain (**KD**), Damping gain (**KV**), Integrator Gain (**KI**), Inertia feed forward gain constant (**KK**). See details below.

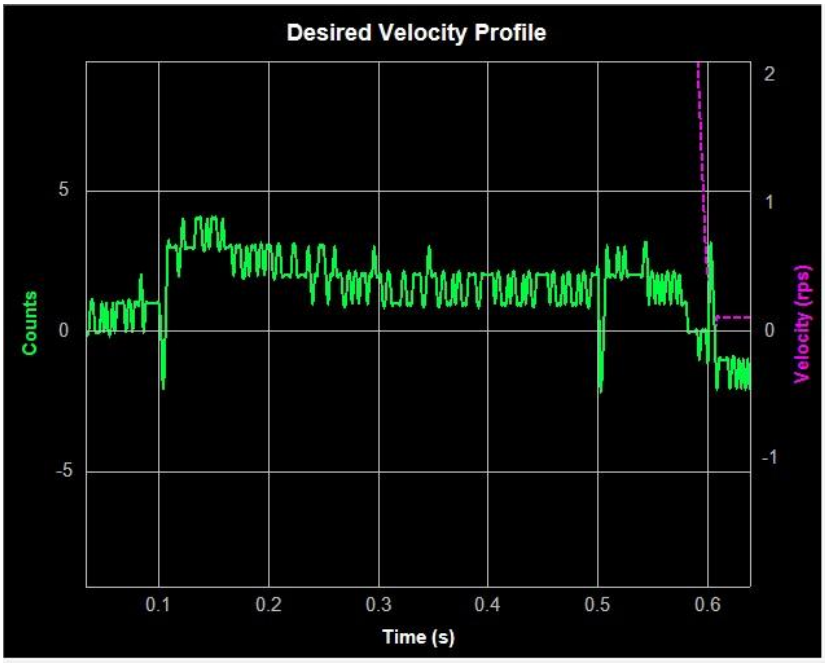
However, if you experience mechanical resonance or hear high-pitched squealing noises, you can lower the PID Output Filter below the natural frequency of your system so that the PID output does not excite the resonance.

If you have a large inertial load, you’ll probably find that you (or the auto-tuner) need to set the gain parameters high, especially PP and KI, to get good response. Then you will want to increase the damping to prevent ringing. Now the system is likely to be so tight that if you have a springy, all metal coupling it may “buzz” or “squawk”. Reducing the frequency of the derivative filter can remove this objectionable sound.

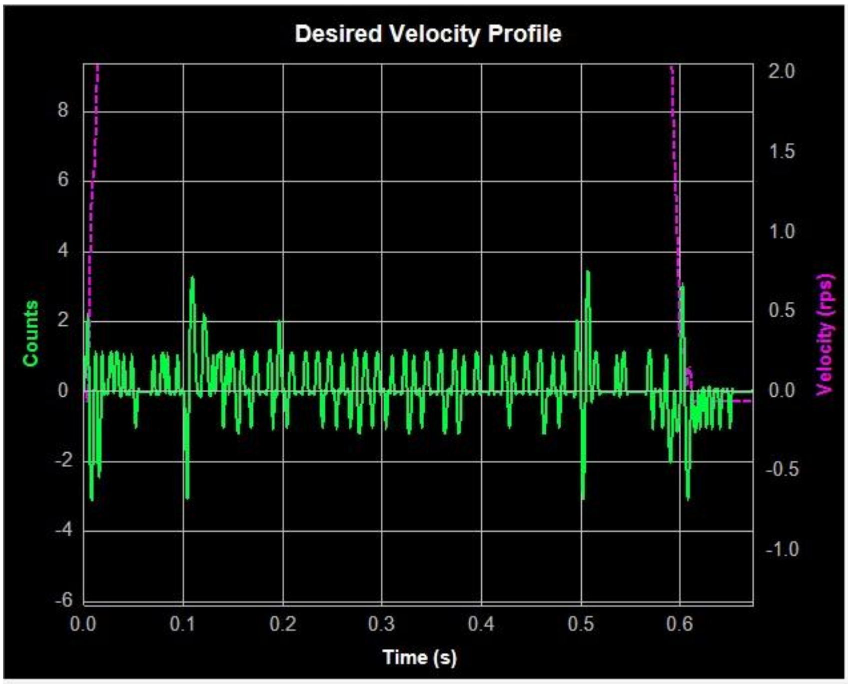
4.3.1 Position loop gain (KF)

This parameter is the primary gain term for minimizing the position error. Increasing KF will increase stiffness and reduce settling time. However, it might cause vibration if increased too much. This is simplest part of the PID loop; the drive will apply current to the motor in direct proportion to the error. Because the current controls the torque output from the motor, increasing this gain will increase the magnitude of torque in direct proportion to the position error. Here's an example: if the motor were standing still, and you suddenly turned the shaft by hand, you'd want the drive to increase the motor current so that it goes back into position. The further you disturb the motor from its target position, the more the torque will increase.

As shown below, if KF is small, position error will be high at all times (during acceleration, constant velocity, and deceleration).



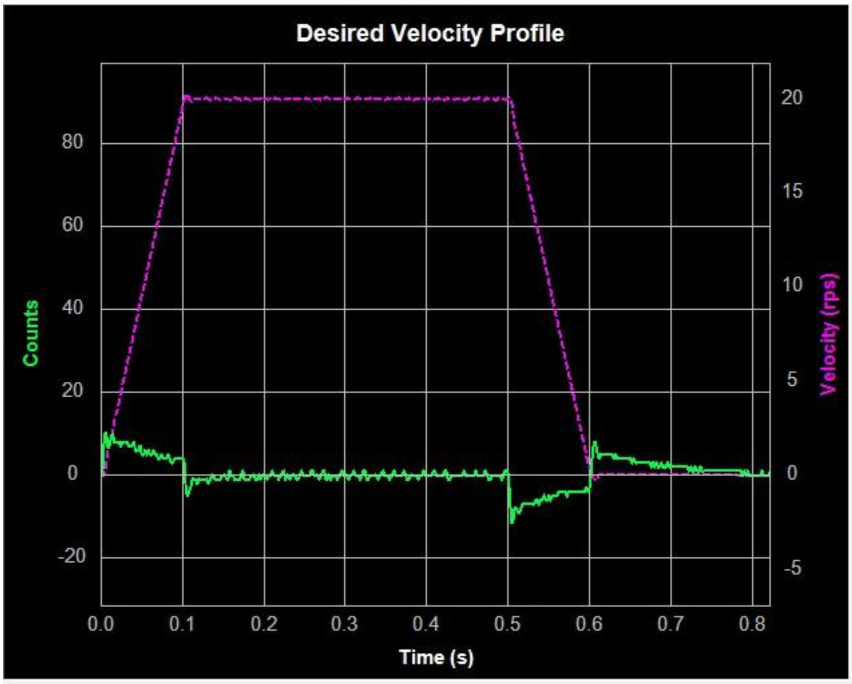
As shown below, if the KF value is set appropriately, the position error during acceleration and deceleration will settle very quickly, and position error of ±1 count can be achieved during constant velocity as well as when the motor comes to rest at its target position.



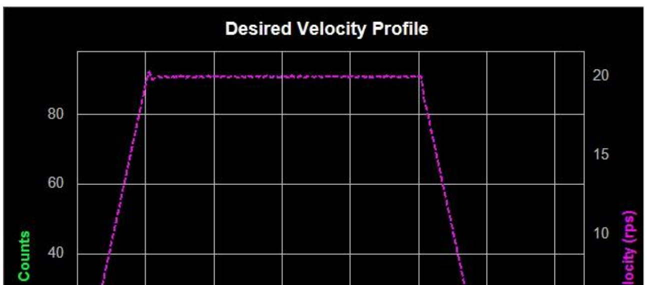
4.3.2 Integrator Gain (KI)

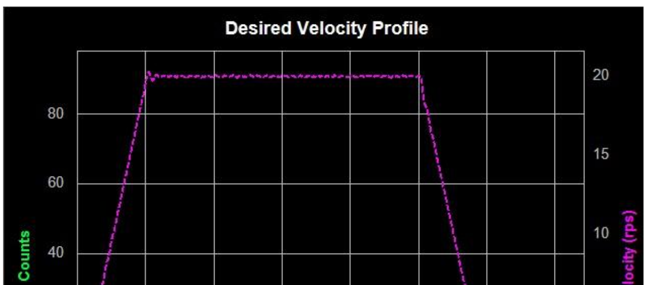
The position loop gain (KF) alone will often not be enough to give the best performance in terms of minimizing the position error and may require a long time settling time. In these cases, the Integral gain (KI) will keep adding up that error and continue to increase the torque until the motor truly returns to the target position.

As the next chart shows, when KI is small, the system will require excessive time for position errors to settle out and during acceleration, deceleration, and when the motor stops.

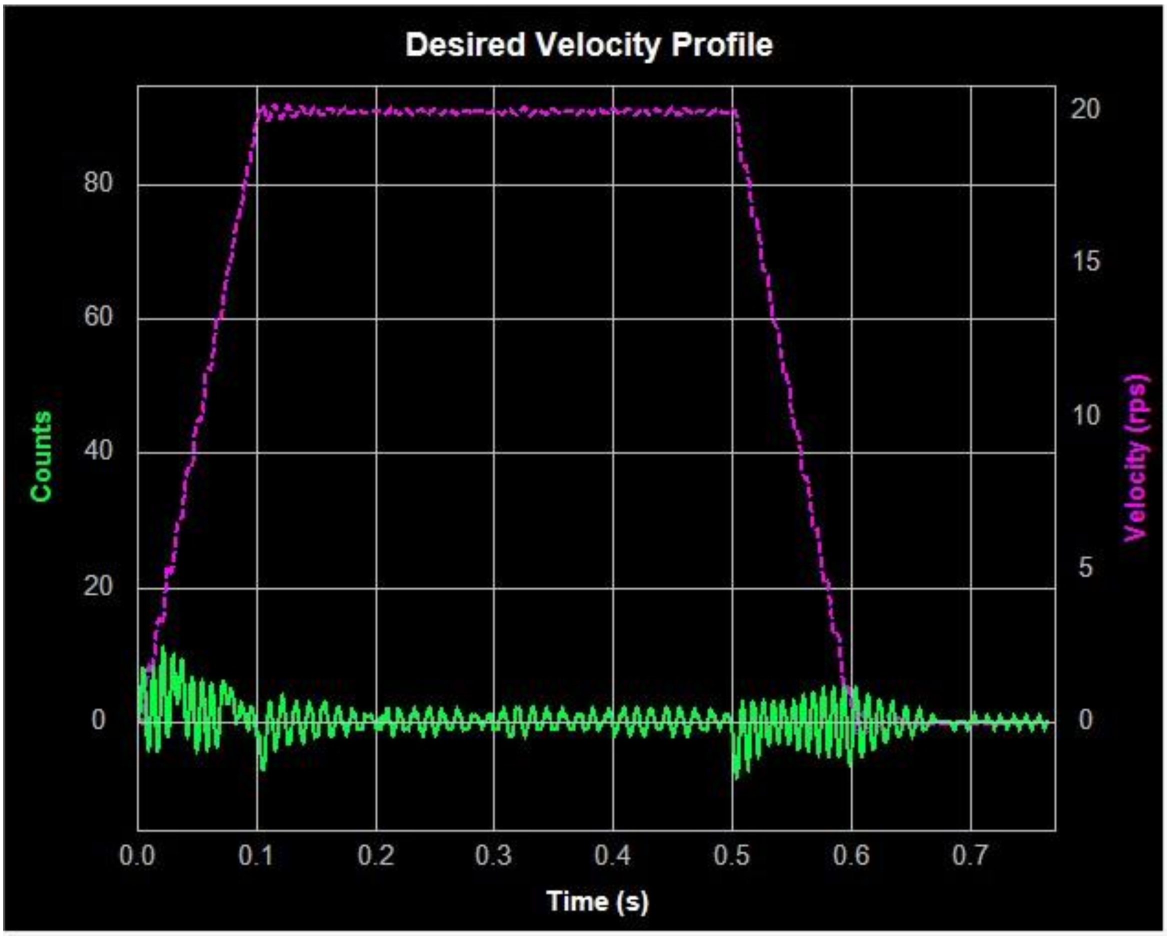


As the next charts show, increasing KI can improve system response time and reduce position error and settling time during motor acceleration, deceleration, and when the motor comes to rest.





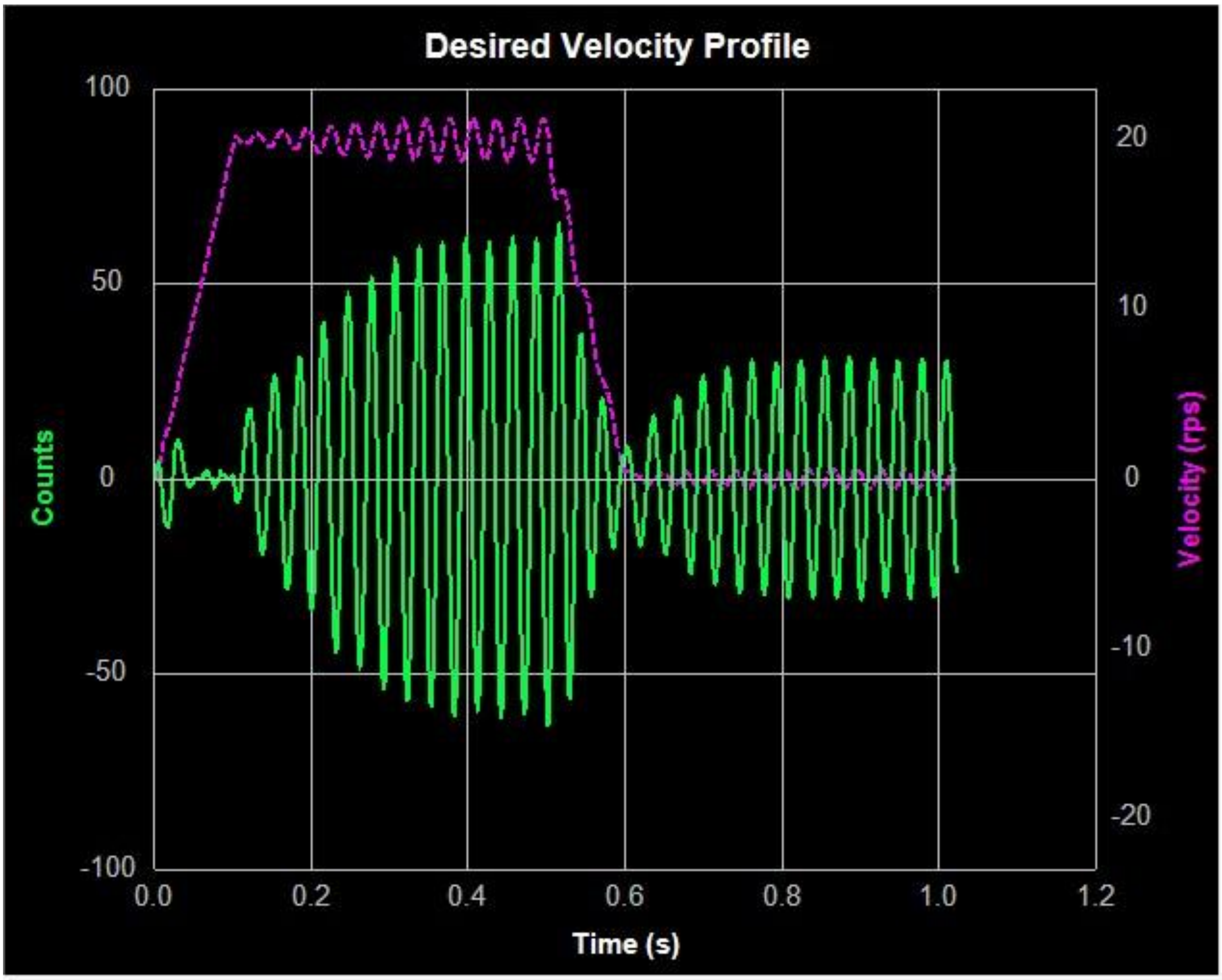
As seen below, if KI is too large, the whole servo system will vibrate and make noise. This, in turn, will increase the position error and may greatly extend settling time due to system oscillations around the target position.



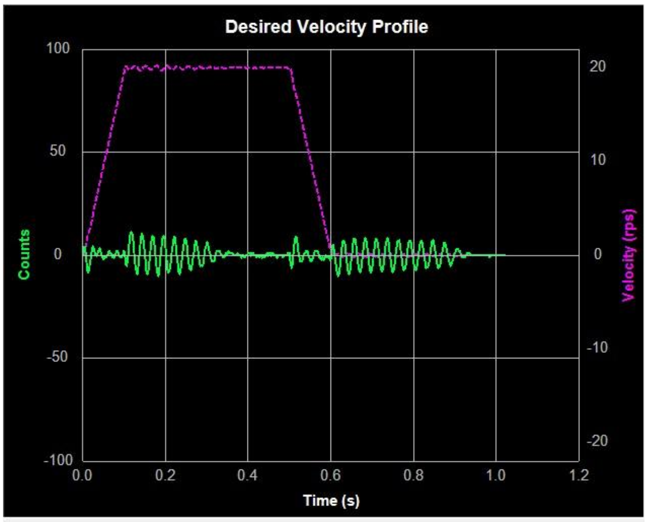
4.3.3 Damping gain (KV)

As the motor load inertia increases, the servo system will require higher damping gain (KV) to reduce position errors during constant speed and when the motor stops.

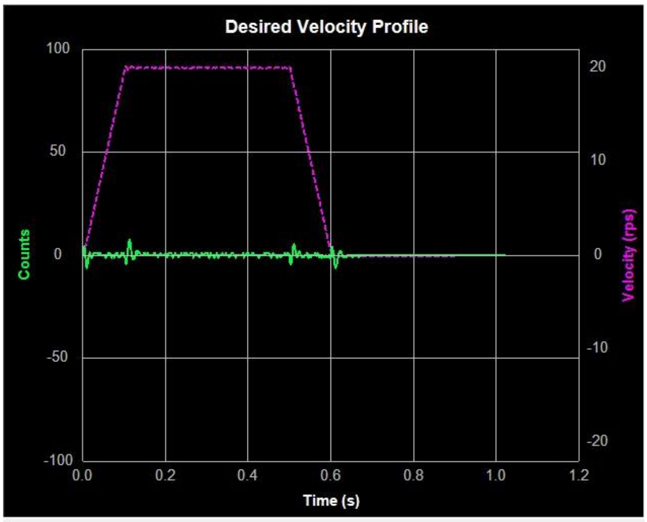
When KV is too small, this low damping value will cause large position error fluctuations while the motor is running at constant velocity and while stopped, holding position. As the next chart shows, an increasing amount of position error occurs during constant velocity and when stopped. These oscillations seen on the graph will result in motor and system vibration, as well as audible noise.



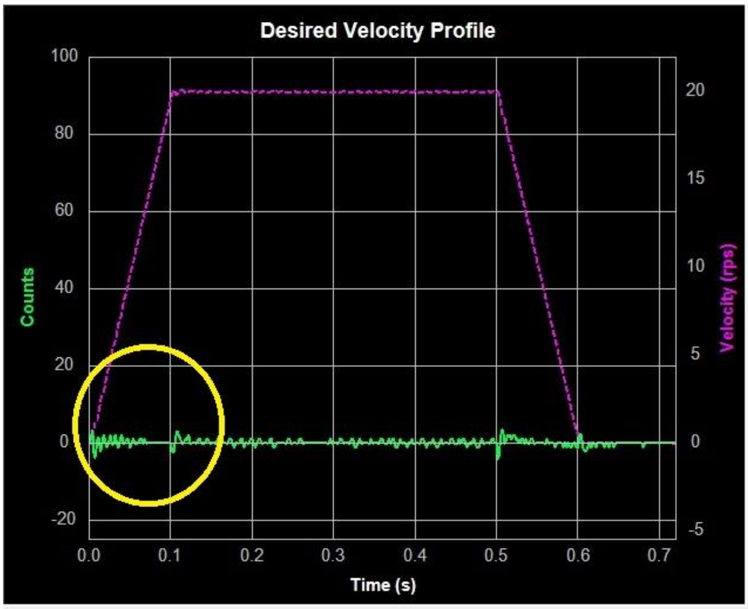
As seen below show, the position error is reduced as KV increases.



KV= 10000



KV = 16000

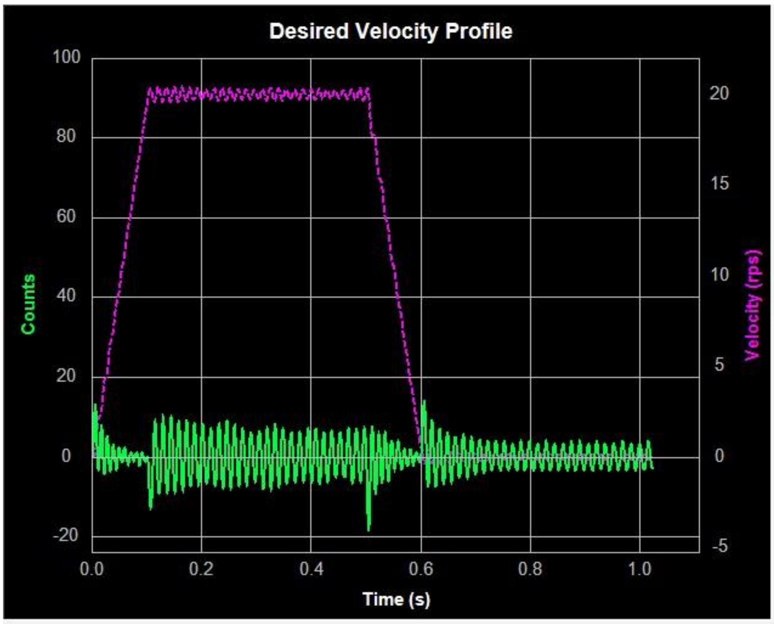


KV = 32000 (too large)

4.3.4 Derivative gain (KD)

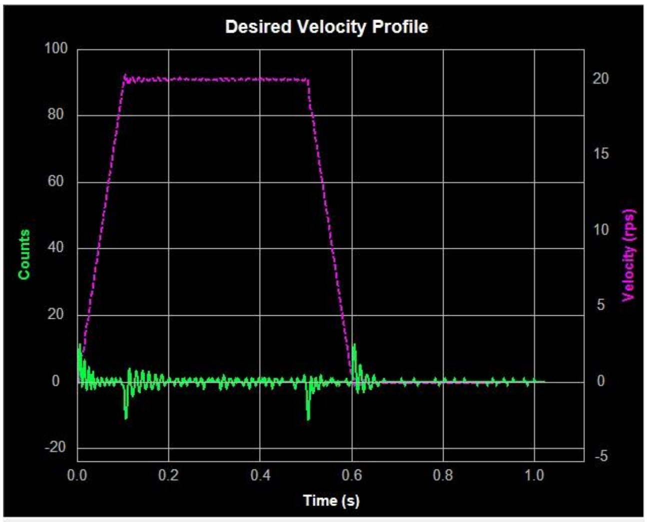
A simple PI controller without Derivative gain (KD) would cause the motor to overreact to small errors, creating ever larger errors and, ultimately becoming unstable. If you knew what the motor was going to do before it did it, this behavior could be prevented. When pulling a car into a garage, for example, most people do not wait until the car is fully into the garage before stepping on the brakes. Instead, most people slow down as they see the distance between them and their objective get smaller.

A motor drive can control a motor better if it examines the rate of change of the position error and includes that in its torque calculation. So, as the position error decreases, the torque commanded to the motor can be reduced with the appropriate KD setting. In the example shown below, when KD is small, the system does not settle quickly after changes in the move profile. Instead, the response indicates that the motor is oscillating around the target position that is being defined throughout the move profile.

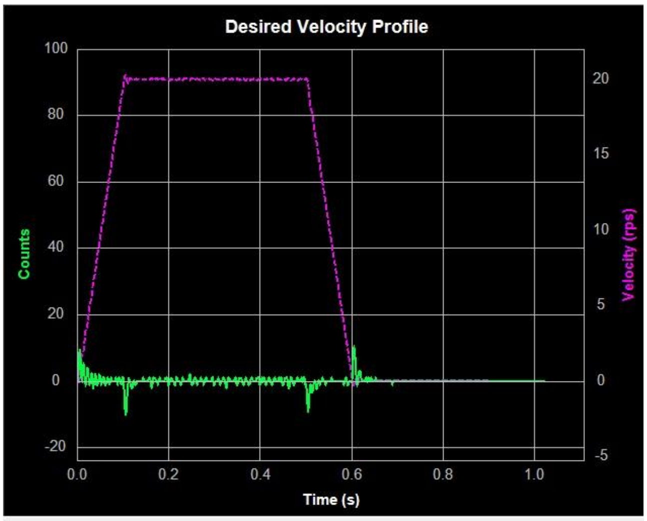


KD = 3000 (too small)

As KD increases, the system takes less time to settle as shown below.

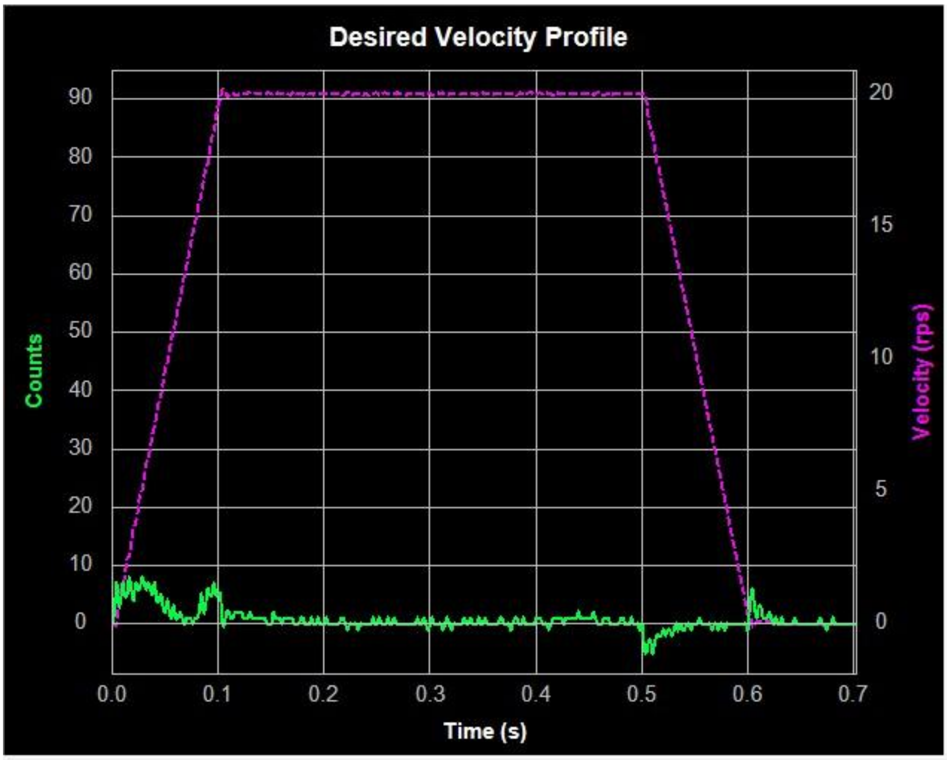


KD= 4000



KD= 7000

When KD is too large, however, the system will become highly sensitive to the commanded changes in motion, which can potentially cause unexpected system vibrations and noise as shown below.

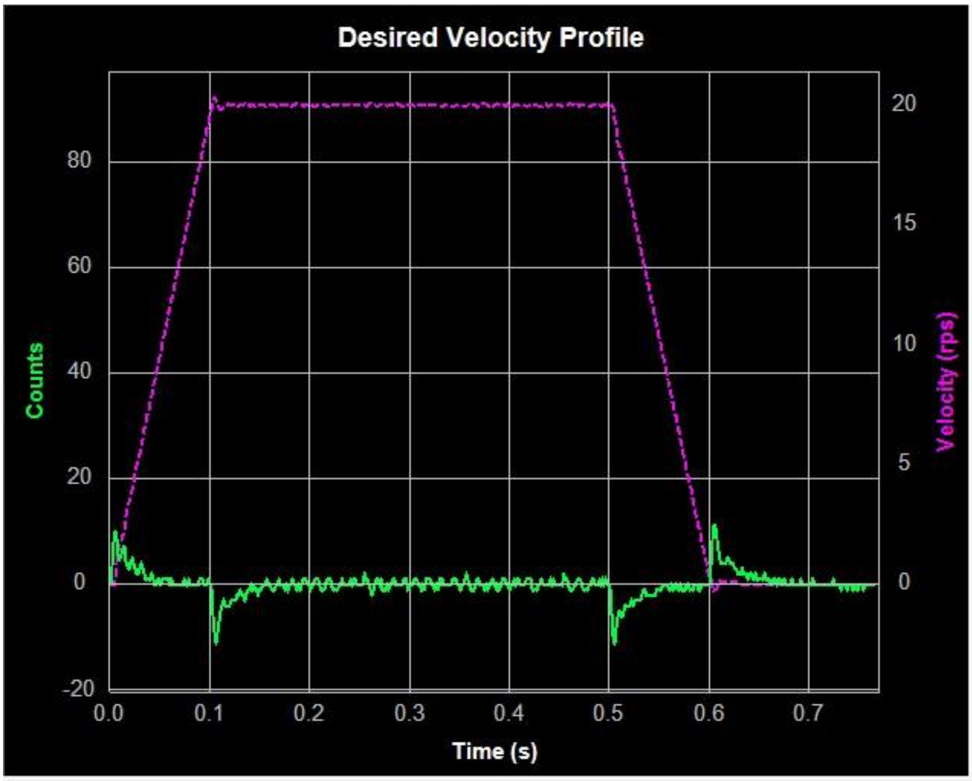


KD = 15000 (too large)

4.3.5 Inertia Feedforward Constant (KK)

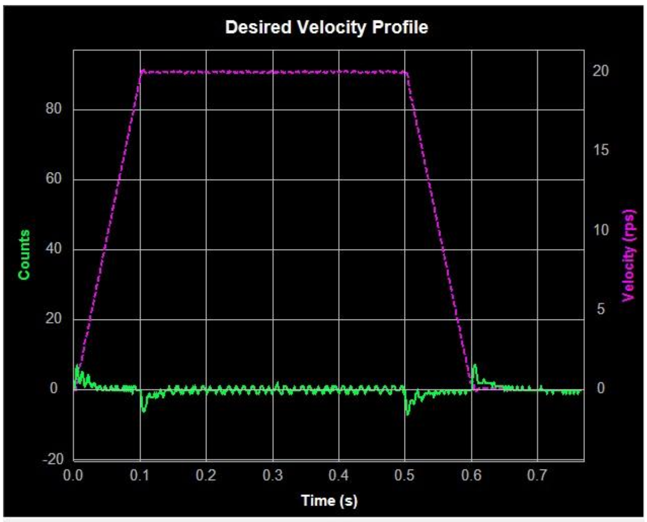
With larger loads typically comes larger load Inertia. These larger inertias can be more easily accelerated or decelerated by anticipating the control system needs. The Acceleration Feedforward gain term (KK) does this by adding an acceleration value to the control value, which reduces position error during acceleration and deceleration.

When KK is small, the feedforward constant will not be enough to effectively reduce position error. This will cause undesirable effects on the system's dynamic performance during the acceleration and deceleration. The result will be larger position error and longer settling time as shown below.

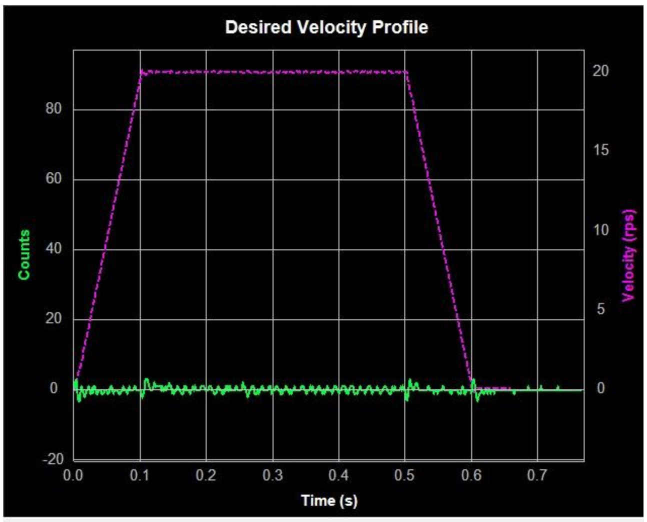


KK = 2000(too small)

As shown below, as KK increases, the system's dynamic performance improves. The position error during acceleration and deceleration is reduced significantly as a result.

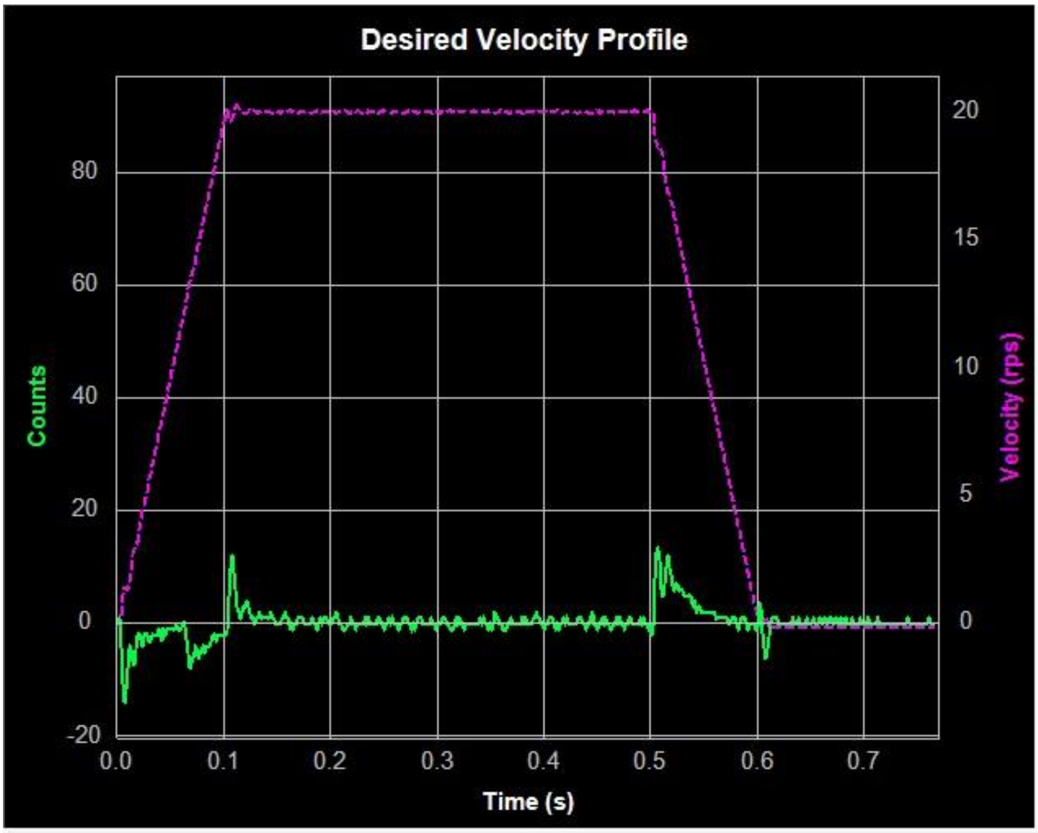


KK= 4000



KK = 11000

When the feedforward (KK) gain is too large, however, the opposite effect can be seen. This will also decrease system dynamic performance by increasing position error and system settling time, as shown below.

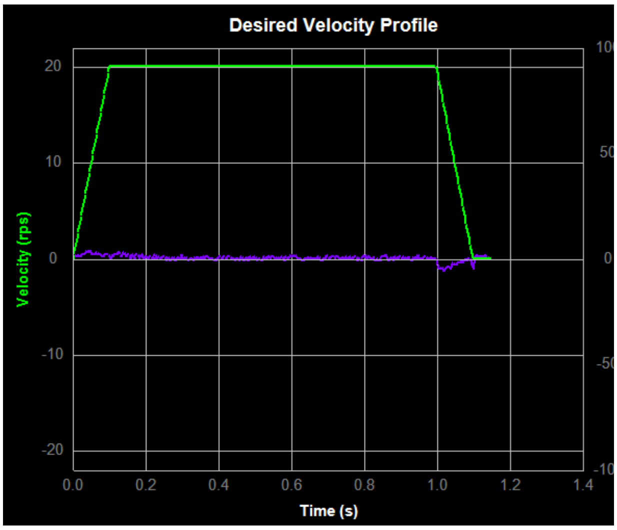


KK=19000 (too large)

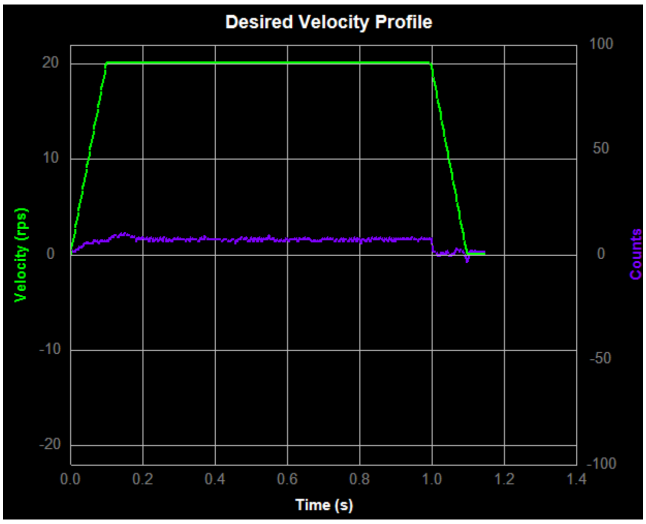
NOTE: When adjusting control loop gain values remember that the Feedforward Term (KK) has no effect when operating in the Position – Pulse & Direction Control Mode.

4.3.6 Follow Factor (KL)

A larger Follow Factor (KL) value will reduce system noise and eliminate overshoot, but will reduce the system's dynamic following performance. Lower values will increase system stiffness, but may cause system noise as shown below (Green = Actual Speed; Purple = Position error).



KL=0

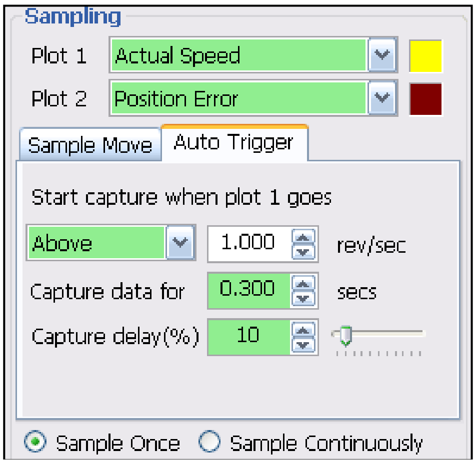


KL=200

4.4 Using Auto Trigger Sampling

In cases where an external controller is used to perform move profiles, such as in **Position Control Mode** using **Pulse & Direction** input, the **Auto Trigger** function will allow the **Sampling** tool to collect data and display the move profile.
This sampling technique is different in that it is not triggered by the start of a move profile as the drive cannot know when the move is actually started (remember the controller is external). Instead, the **Auto Trigger** function waits for a predefined set of conditions, or triggering event, before the move profile data is collected.

When using **Auto Trigger**, it's important to first select the conditions that will trigger the sampling. Begin by selecting the desired trigger value in the **Plot 1** list. This selection is what is monitored by the Auto Trigger; **Plot 2** will be displayed, but is not monitored for scope triggering purposes. See below.



In the Auto Trigger tab the displayed text will indicate the value to be used and the condition that will trigger the capture of the selected data plots. In the example above, the capture will begin when **Actual Speed** is **Above 1.000 rev/sec**, the capture will **Capture data** for **0.300 seconds** and there will be a **10% Capture delay** from the beginning of the capture to the trigger point. The **Capture delay** allows viewing of the data prior to the trigger point so that a more complete profile can be observed.

When changing **Plot 1** to other selections notice that the units for the capture trigger will change with it. For example, when selecting **Position Error** the capture will look at **Counts** for determining the trigger point.

Sample Once: when the **Start** button is clicked, the servo drive begins continuous collection of data. It will constantly check the data to see if the value meets the capture trigger conditions. At the same time SVX Servo Suite monitors the status of the servo drive to detect if the capture is complete.
When the capture is complete the data is displayed in the profile window.

Sample Continuously: when the **Start** button is clicked, the capture is repeated each time the trigger condition is met until the **Stop** button is clicked. During continuous sampling the tuning gains can be changed at any time and will be updated automatically. This allows for more dynamic adjustment of the gains, thereby speeding up the tuning process.

5 Step 3: Q Programmer

Q Programmer takes the SCL language to a new level by allowing users to create programs that can be stored in the drive's flash memory. The drive can be set to automatically execute programs at power up or it can wait for the programs to be started and stopped using streaming commands over RS-232, RS-485, Ethernet, Modbus, CANopen or EtherNet/IP connections. Q programs can also be triggered using the drive's built-in I/O.

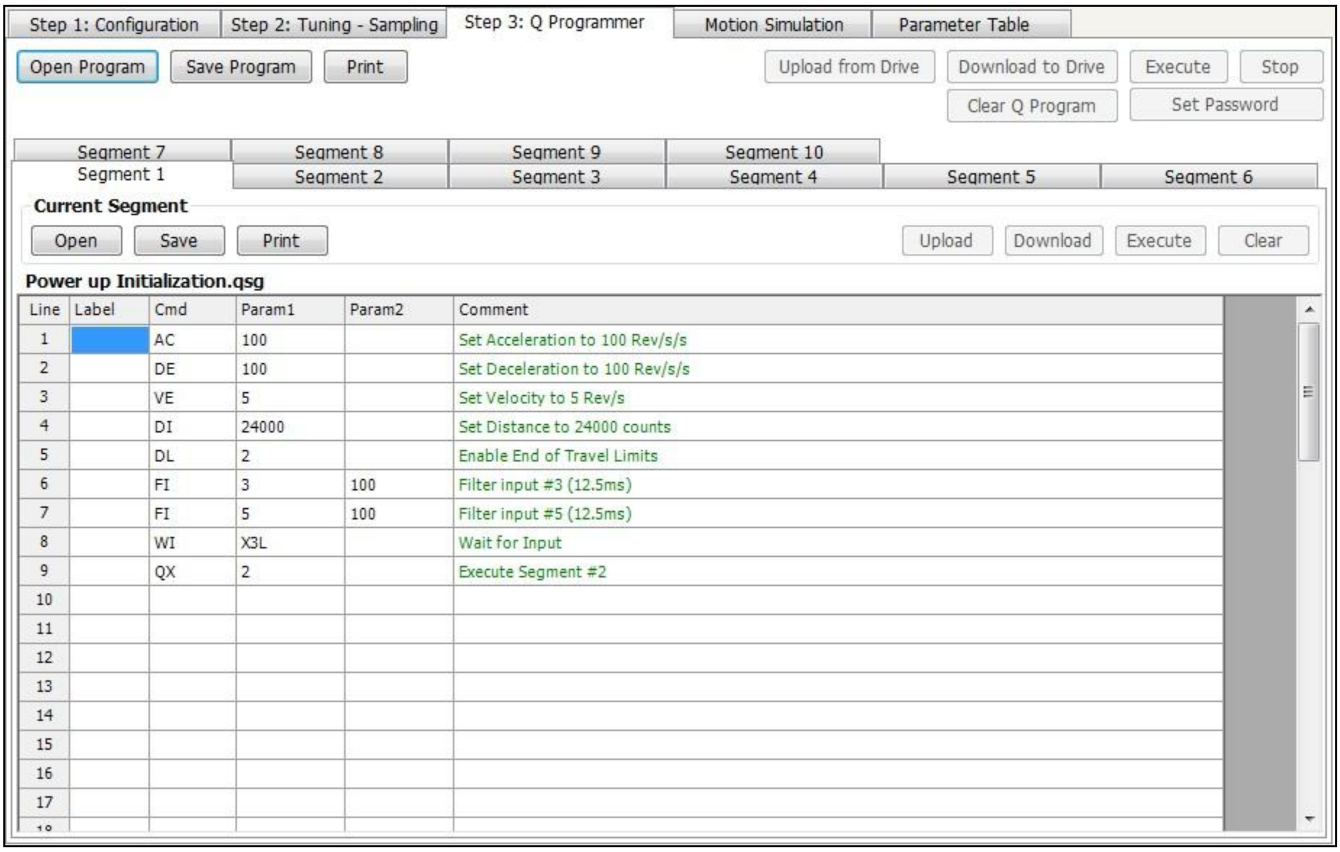
Q Programmer feature list:

- Single-axis motion control
- Stand alone or networked operation
- Multi-tasking
- Conditional processing and program flow control
- Math calculations
- Data register manipulation
- User interaction via touch screen HMI's

For the SV200 Series servo drives, a single Q program can have 10 individual segments with each segment containing a maximum of 62 lines.

5.1 Q Programmer Tab

The Q Programmer page shown here contains all of the control features needed for programming, testing, and setting up Q segments and complete programs to run within a programmable SV200 drive.



Q Programmer interface built into SVX Servo Suite

Control features at top of Q Programmer tab:

Open Q program: Open Q program file from your computer disk

Save Q program: Save Q program file to your computer disk

Print: Print current Q program

Upload from Drive: Upload Q program from the drive
Download to Drive: Download current Q program to the drive
Clear Q Program: Clear current Q program
Execute: Execute current Q program (starts at segment 1)
Stop: Stop the current running Q program
Set Password: Set Q program password. Password can be set to prevent the Q program from being uploaded from the drive. Incorrect password entry will not allow for the program to be uploaded from the drive. To reset your password if forgotten, enter the default password "1234"; doing this will erase the stored Q program from the drive's memory.
Auto Execute Q program at power up: checking this box and downloading the Q program will cause the drive to automatically execute segment 1 of the Q program at power up.

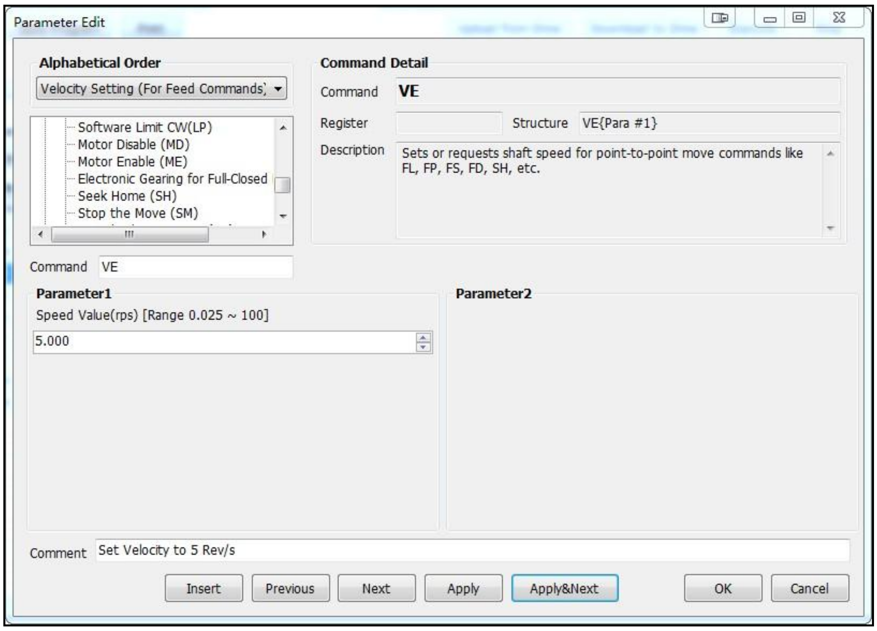
5.2 Current Segment

There are 10 Q segments available for use within the Q program. The "Current Segment" page is used to edit the segment that is currently being viewed.
Control features in the "Current Segment" section:

Open: Open Q segment file from your computer disk
Save: Save Q segment file from your computer disk
Print: Print current Q segment
Upload: Upload Q segment from the drive.
Download: Download Q segment from the drive.
Execute: Execute current Q segment. Use **Stop** button in upper right corner of Q Programmer tab to stop current Q segment.

5.3 Command Editing

Click on any cell in the Cmd column located in the Q programming area and then click on the button that appears on the right side of the cell to open the Parameter Edit panel.
The Parameter Edit panel will pop up as follows:



The Command list is shown on the left hand side of the Parameter Edit panel. Searching for commands in alphabetical order may be done by clicking on the drop-down list above the tree. Typing a valid command directly in to the "command" box will find and display the command details on the right side of the panel.

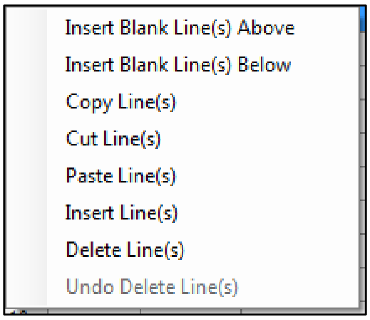
The command's parameters can be entered in the Parameter1 and Parameter2 fields on the Parameter Edit panel or typed directly into the Param1 and Param2 cells in the Q programming area. The number of parameters accepted is based on the command definition and its parameter entry requirements.

The following is a list of the buttons (and their meanings) that appear at the bottom of the Parameter Edit panel.

- **Insert:** Insert a blank line within the current Q segment.
- **Previous:** Moving up by one line within the current Q segment.
- **Next:** Moving down by one line within the current Q segment.
- **Apply:** Apply current command to the segment.
- **Apply&Next:** Apply current command and move to the next line.
- **OK:** Apply current command to the segment and quit.
- **Cancel:** Quit the command editing window without save the change.

The Comment cells in the Q programming area allow for descriptions to be typed in by the programmer to help document and explain the program for later reference. Keep in mind that these comments are not downloaded and stored in the drive's memory; they are available only in the version of the Q program (or segment) that has been saved as a file.

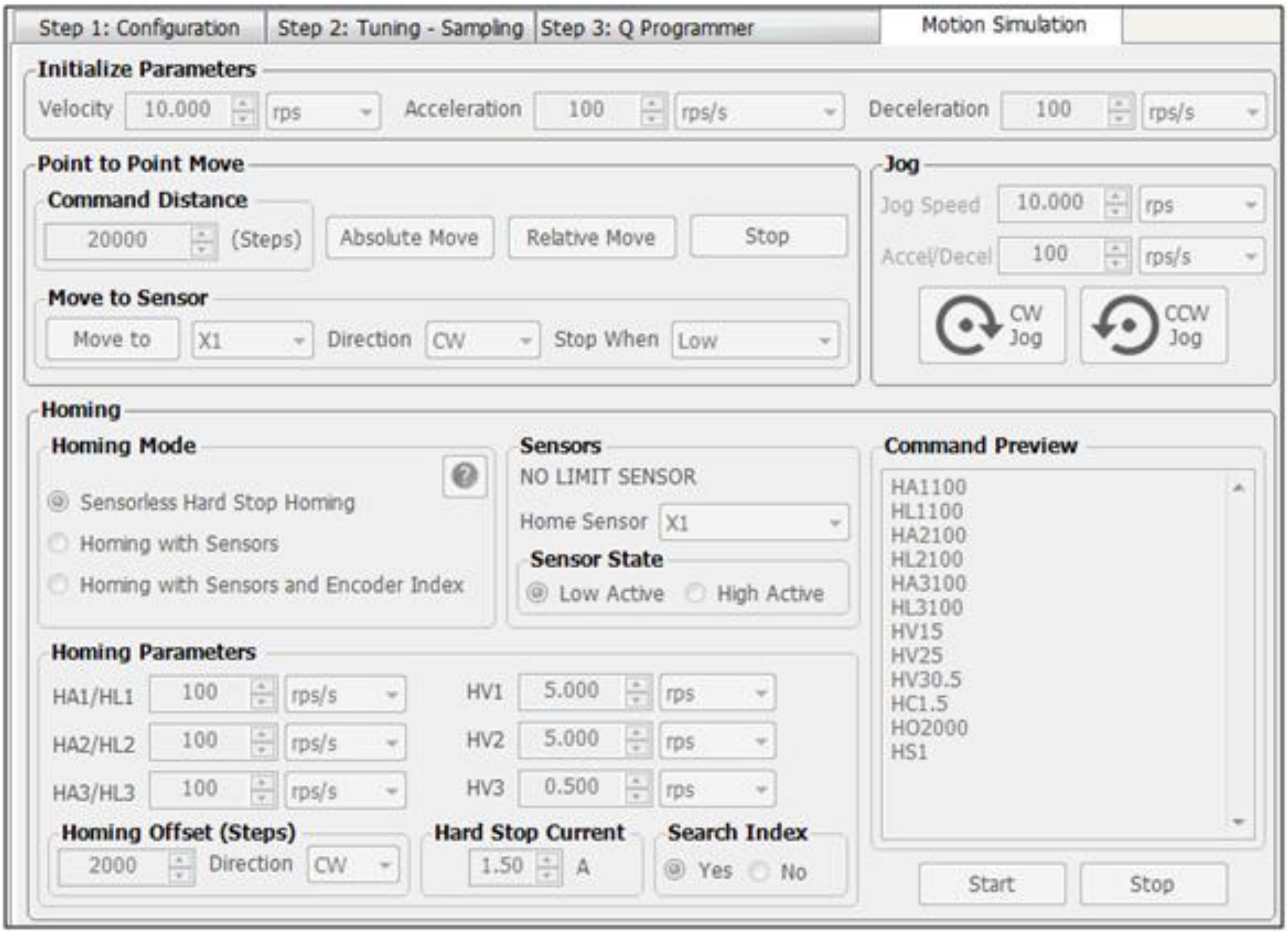
Right clicking on the line number within the Q program area will display the following function list:



These functions are useful when creating and editing a Q program.

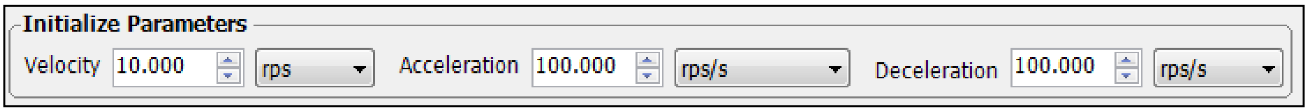
6 Motion Simulation

The Motion Simulation panel provides Point to Point Move, Jog and Homing simulation.



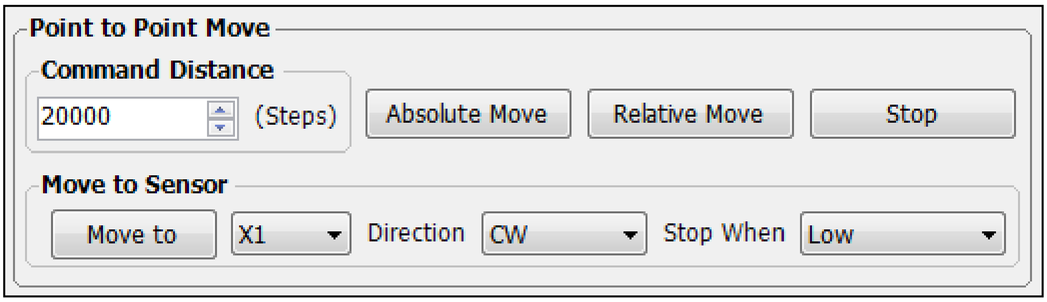
6.1 Initialize Parameters

Before running the simulation, initialize the motion parameters: velocity, acceleration and deceleration.



6.2 Point to Point Move

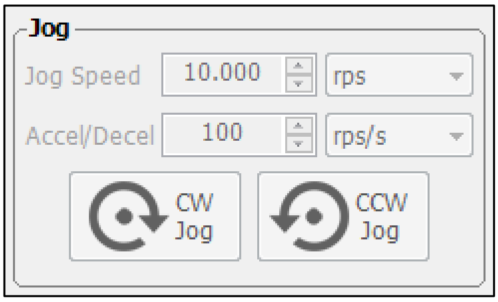
The Point to Point Move section allows the Command Distance to be set for simulation. Click the desired type of move button (either Absolute or Relative) to initiate motion. Click Stop to interrupt motion.



Absolute Move: Execute the absolute motion according to the Command Distance setting. The Absolute ZERO is the zero count of the motor encoder.
Relative Move: Execute the relative motion according to the Command Distance setting.
Move to Sensor: Click the "Move to" after setting the input number, direction of rotation, and input condition.

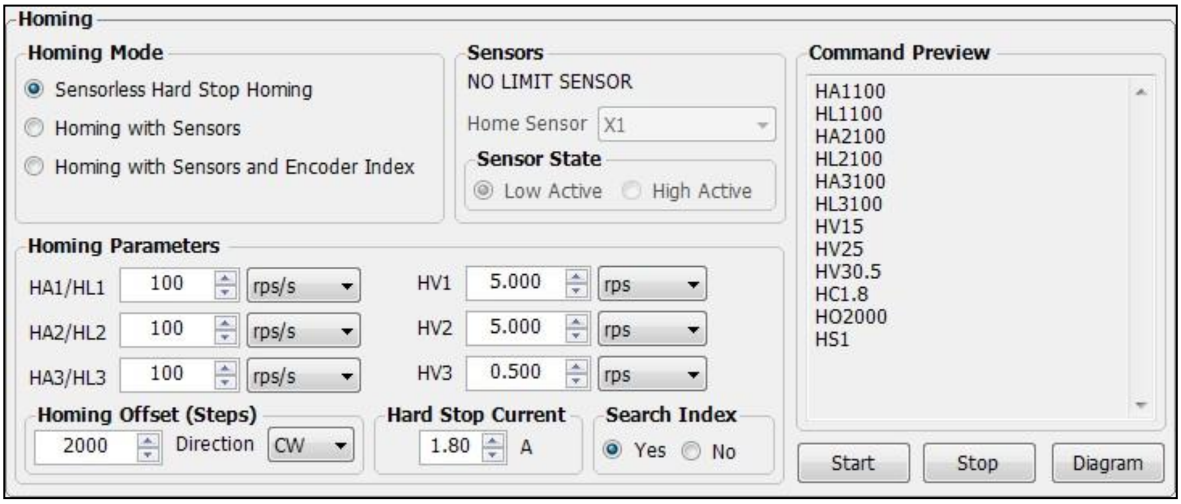
6.3 Jog

The Jog section allows for setting the Jog Speed and Jog acceleration/deceleration. To jog the motor, click and hold the CW Jog or CCW Jog button. The motor will stop jogging when the mouse button is released.



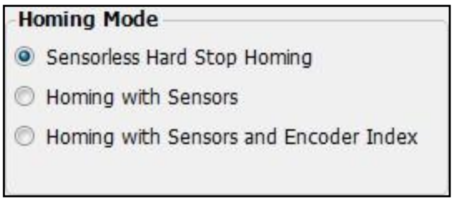
6.4 Homing

The Homing section allows for setting the homing mode, home sensor and state, search speed, acceleration/deceleration, offset, hard stop current, etc... Click "Start" to start simulation of the homing routine. Click the "Stop" button to interrupt and stop motion while the homing routine is in progress.



6.4.1 Homing Mode

There are three possible homing modes that can be simulated: Sensorless Hard Stop Homing, Homing with Sensors, Homing with Sensors and Encoder Index.

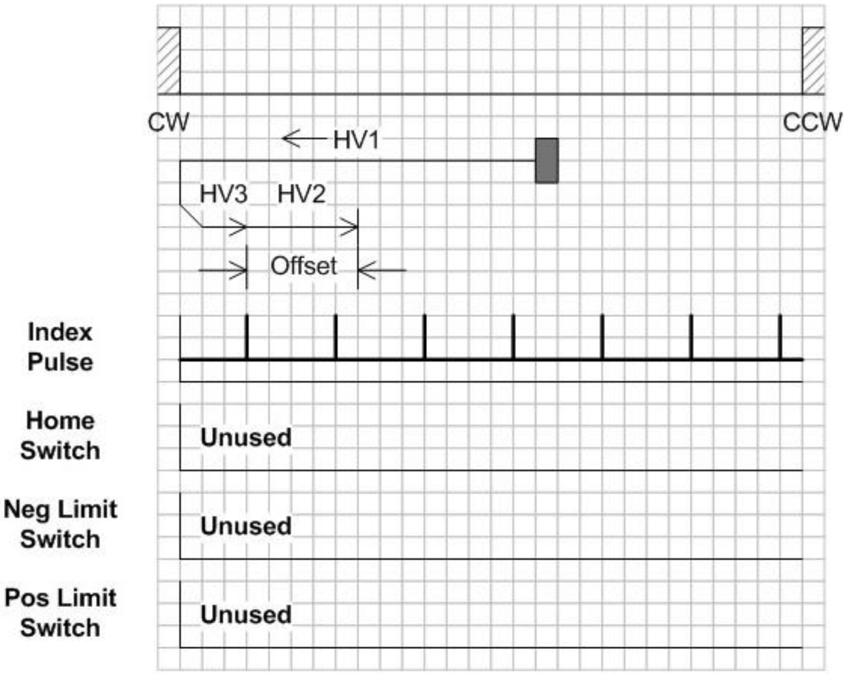


Click the "Diagram" button to view a graphical representation showing details for each Homing mode.

6.4.1.1 Sensorless Hard Stop Homing

Sensorless Hard Stop Homing means homing without any homing sensors. The load will home to a fixed mechanical end-stop with a preset current limit condition.

A diagram of this homing process is shown here:



Searching for the mechanical end-stop begins with HV1 speed. The start direction comes from the sign of the HO command ("-" is CCW, no sign is CW). The motor will stop when the actual current equals the preset Hard Stop Current (HC) as the actuator reaches the end of travel.

Search Index behavior:

If YES for Search Index is checked…

Then the motor runs in the opposite direction at the HV3 speed to the first encoder index. After that the motor moves to the Homing Offset (HO) position at the HV2 speed.

If NO for Search Index is checked…

Then the motor just moves to the Homing Offset (HO) position at the HV2 speed.

6.4.1.2 Homing with Sensors

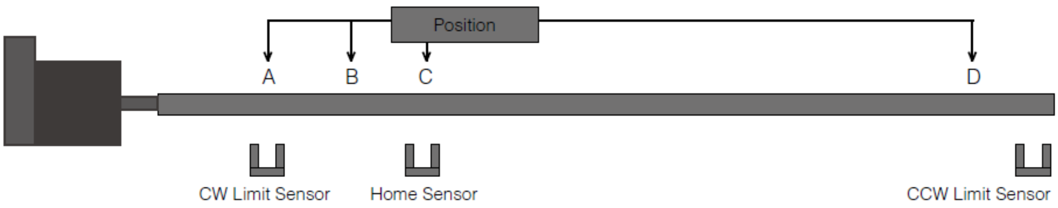
This option executes an Extended Homing routine, which requires the input number and condition to be specified for the home sensor. The speed is set by HV commands; there are three velocity setting for each step of the homing routine (see the detailed description below for HV1, HV2, and HV3).

Acceleration and deceleration are set using HA (Homing Accel) and HL (Homing Decel). The direction of travel when homing starts comes from the sign of the HO command ("-" is CCW, no sign is CW).

Here is a description for each command along with the expected motion.

HV1: Homing velocity while searching for Limit Sensor and Home sensor.
HV2: Homing velocity while moving the setting distance after (beyond) home sensor reached.
HV3: Homing velocity while returning back to the home sensor after offset distance move has been completed.
HO: Home offset distance to move after home sensor has been reached.

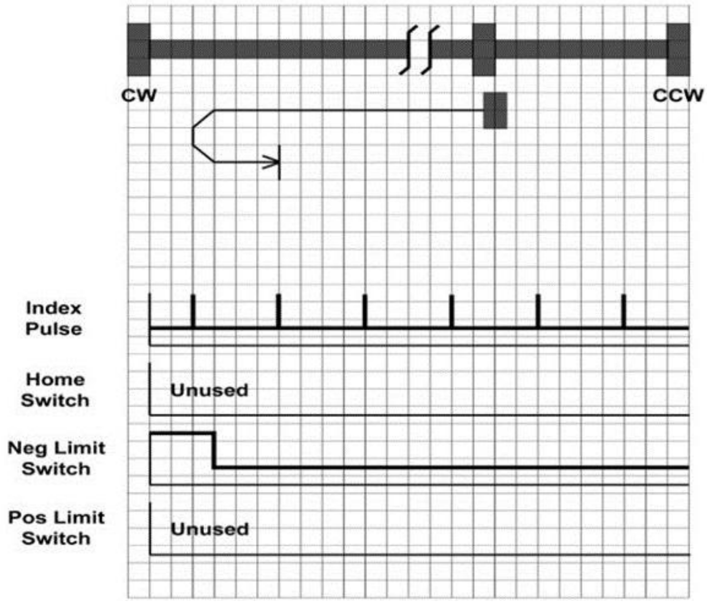
Here is an example of Extended Homing operation.
Conditions:
HO = 20000(no sign, CW direction), DL = 2 (Define Limit setting 2; limit input open when limit reached)
For End-of-Travel settings in the SVX Servo Suite software, see Digital Input settings on the Configuration tab.



A. Homing started when the motor is positioned at point A (CW Limit Sensor triggered)
   1) The motor searches for the home sensor at speed specified by HV1 value, with HA1/HL1 for acceleration/deceleration.
   2) Once the home sensor is reached, the offset move (HO distance) is executed at the HV2 speed and HA2/HL2 accel/decel beyond home sensor in CCW direction.
   3) Finally, the motor returns back to the home sensor at HV3 speed and HA3/HL3 accel/decel.
B. Homing is started when the motor is positioned at point B (stopped between CW Limit Sensor and Home Sensor)
   1) The motor moves in the CW direction to find CW limit sensor with HV1 speed and HA1/HL1 accel/ decel.
   2) The CW limit sensor triggered and motion stops.
   3) Then the steps described in the scenario above are followed.
C. Homing is started when the motor is positioned at point C (Home Sensor triggered)
   1) The motor moves to the distance specified by HO value with HV2 speed and HA2/HL2 accel/decel beyond the home sensor in CCW direction.
   2) Then, the motor returns back to the home sensor at HV3 speed and HA3/HL3 accel/decel.
D. Homing is started when the motor is positioned at point D (stopped between Home Sensor and CCW Limit Sensor)
   1) The motor moves to the home sensor at speed HV1 in CW direction.
   2) After the home sensor is triggered, the steps described in the scenario above are followed.

NOTE: If the HO value is negative, the motor will start in the CCW direction.

6.4.1.3 Homing with Sensors and Encoder Index



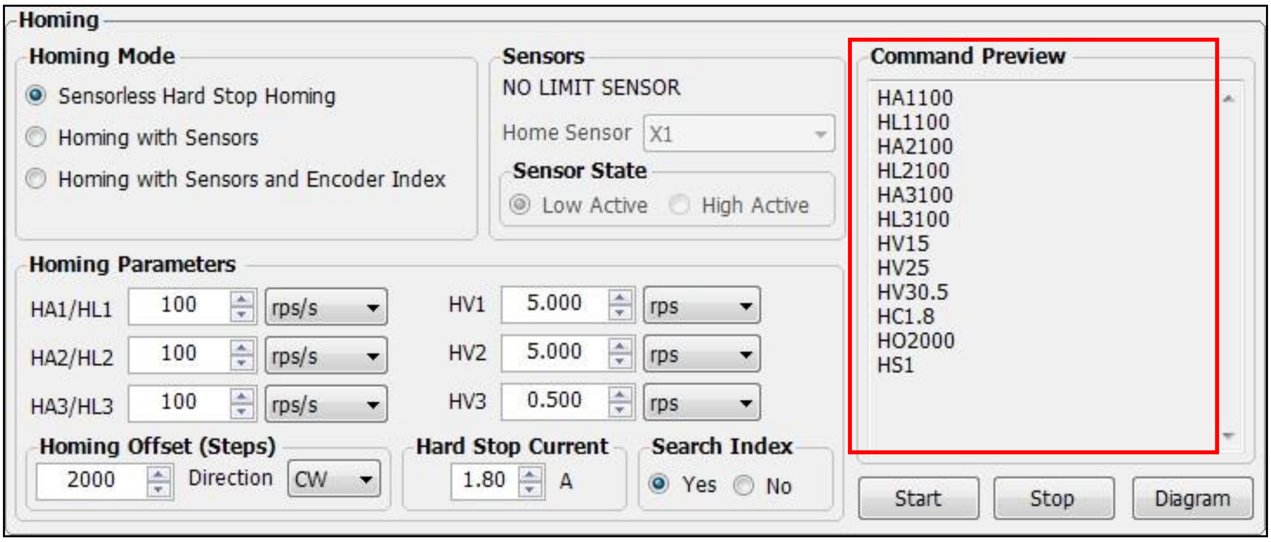
As shown above, the initial direction of movement shall be CW if the CW limit switch is inactive. The home position shall be at the first index pulse reached in the CCW direction after the CW limit switch becomes inactive after being triggered.

In this scenario, the motor will first move in CW direction and then stop when CW limit switch is triggered. Then, it moves in CCW direction until the first index pulse is reached after the CW limit switch transitions from active to inactive. Velocity, acceleration and deceleration are set by the VE, AC and DE respectively in the first move. Velocity, acceleration and deceleration are set by the VC, AC and DE commands respectively in the second move. The index is masked until the motor moves in the CCW direction and the CW limit has transitioned states.

The DL command or the end-of-travel input configuration sets the active signal state for the limit sensors (i.e. high level or low).

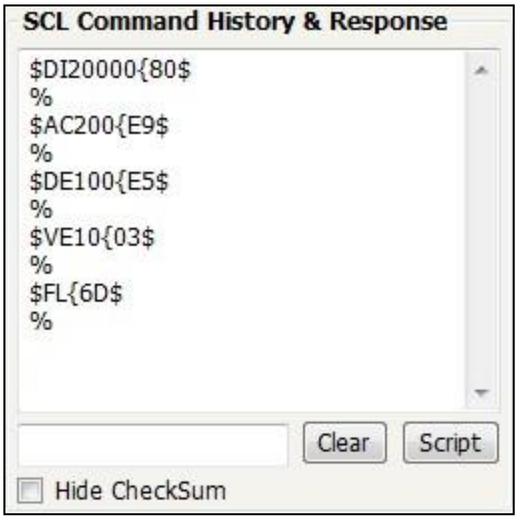
6.4.2 Command Preview

The Command Preview window shows all the SCL commands used for the homing mode selected. These commands

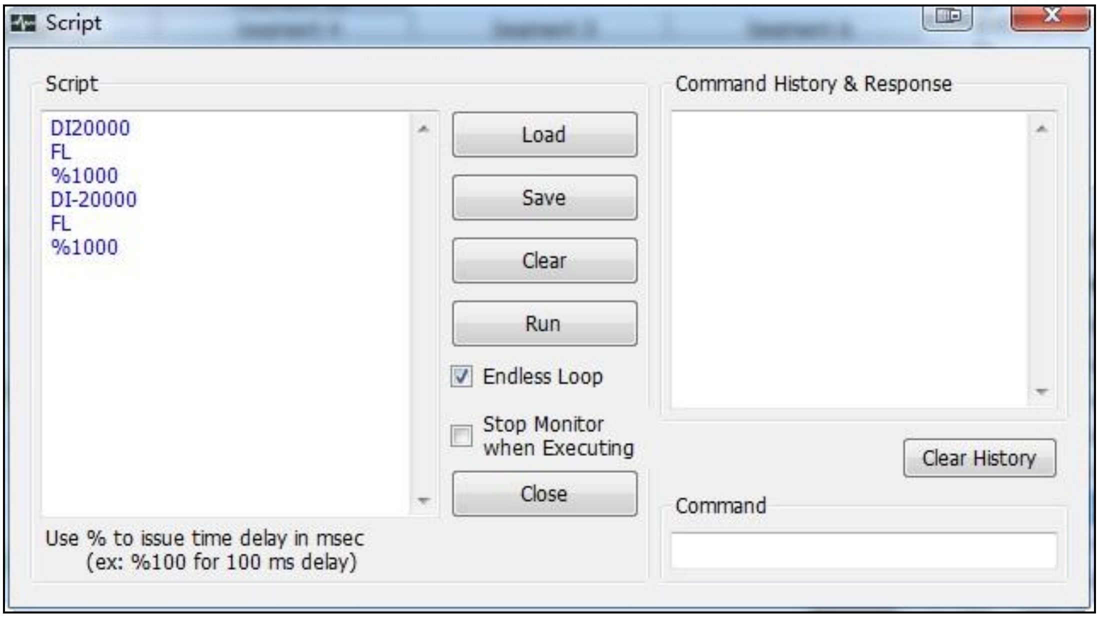


7 SCL Terminal

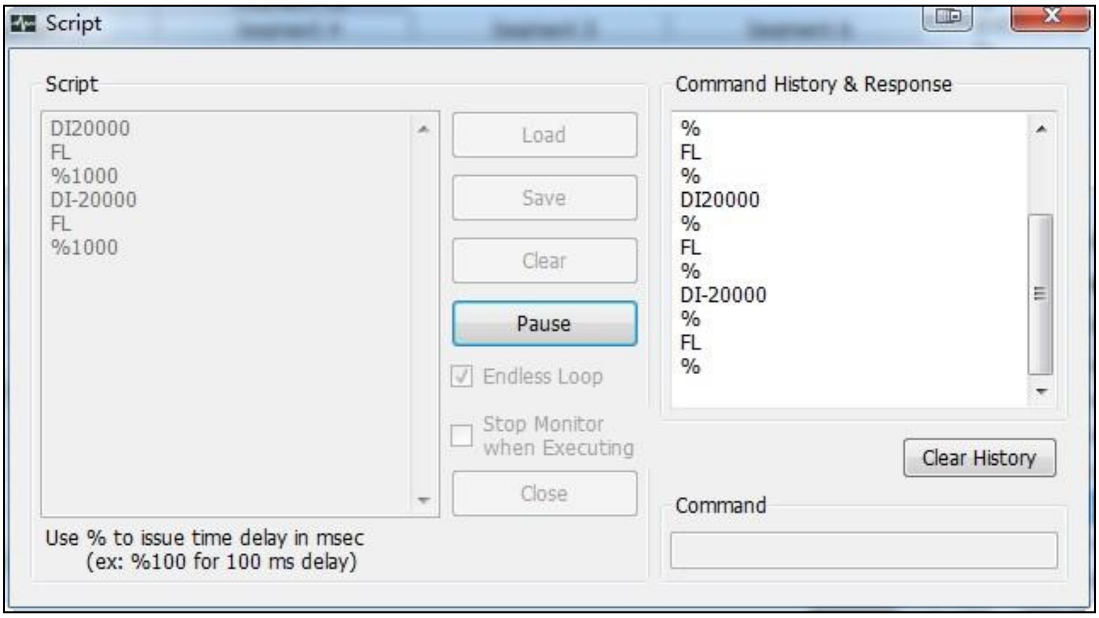
The SCL Terminal allows you to send SCL commands to the drive, regardless of the Operating Mode. The terminal is also useful as a commissioning tool, allowing you to test your drive using SCL commands without having to launch a separate application.



In the SCL Terminal window, there is a “Script” button, which will open the Script window shown below when clicked.



This panel can be used to edit, load, save, and run a SCL command script. For continuous looping of a script, check the “Endless Loop” box and click Run. Click Pause to stop the script.

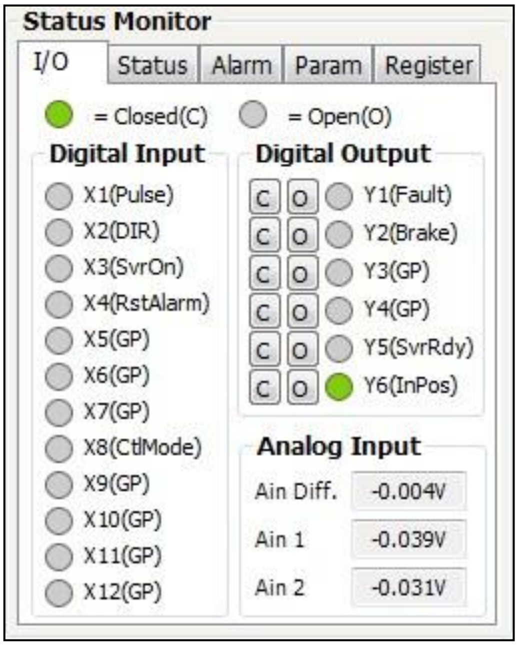


Note: Check the “Stop Monitor when Executing” box to suspend background status monitoring. This will make the script run more efficiently and reduce delay time.

8 Status Monitor

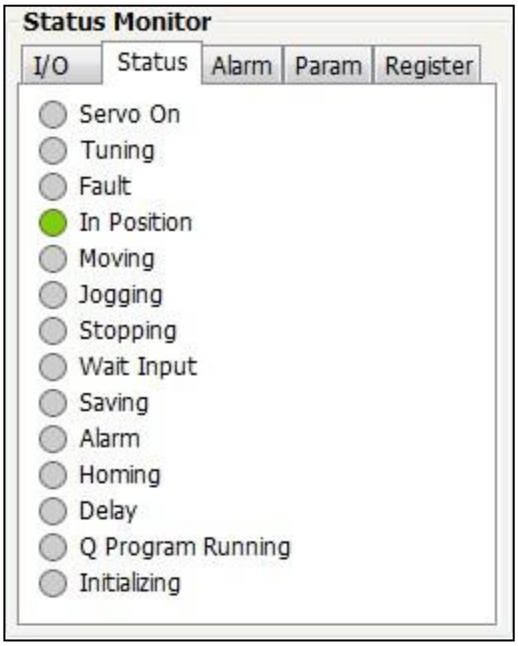
The Status Monitor displays I/O status, drive status, alarm status, parameters and registers.

8.1 I/O Monitor



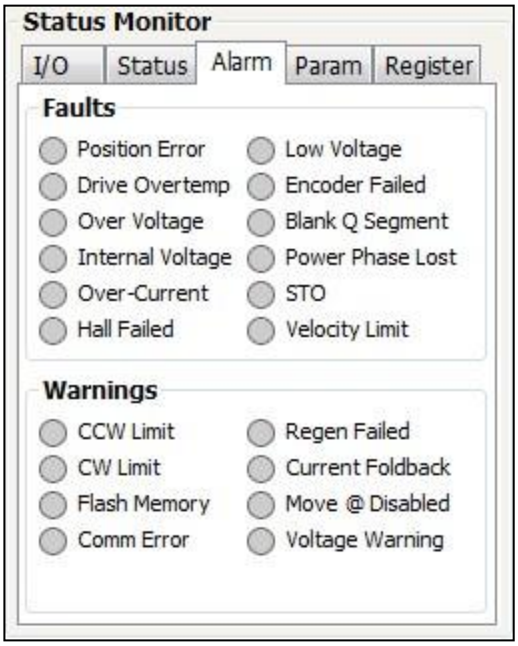
The I/O tab shows the status (open or closed) of digital inputs and outputs and the voltage at the analog input. You can also force an output on or off by clicking “C” for closed or “O” for open.

8.2 Drive Status Monitor



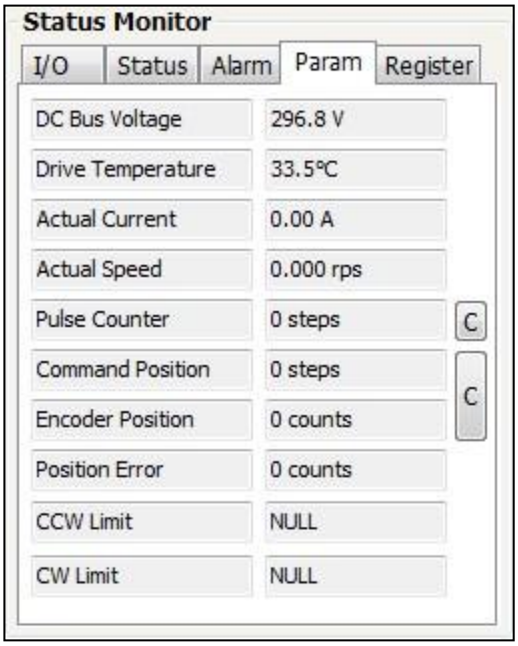
The Status tab shows the bits in the status code word. Each bit has a meaning that is conveyed by a virtual LED in the software. For example, if the Alarm LED is lit, there is an alarm present. If the Moving LED is lit, the motor is moving.

8.3 Alarm Monitor



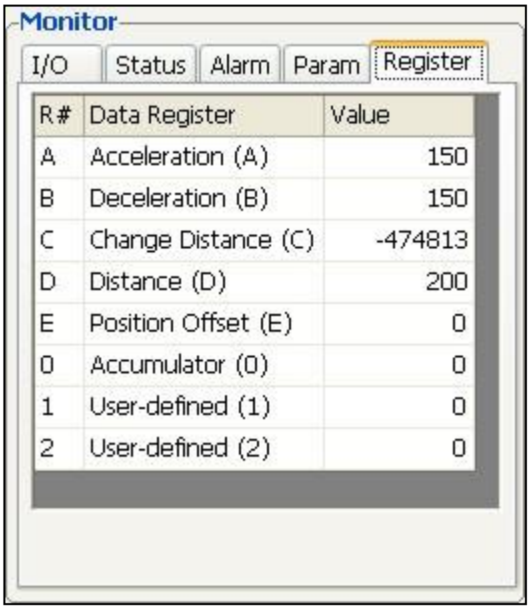
If your drive experiences an alarm, you can click on the Alarm tab to learn more. There are two categories of alarms: faults and warnings. Faults are indicated in red and warnings are indicated in yellow. Faults disable the motor, warnings do not.

8.4 Drive Parameter Monitor



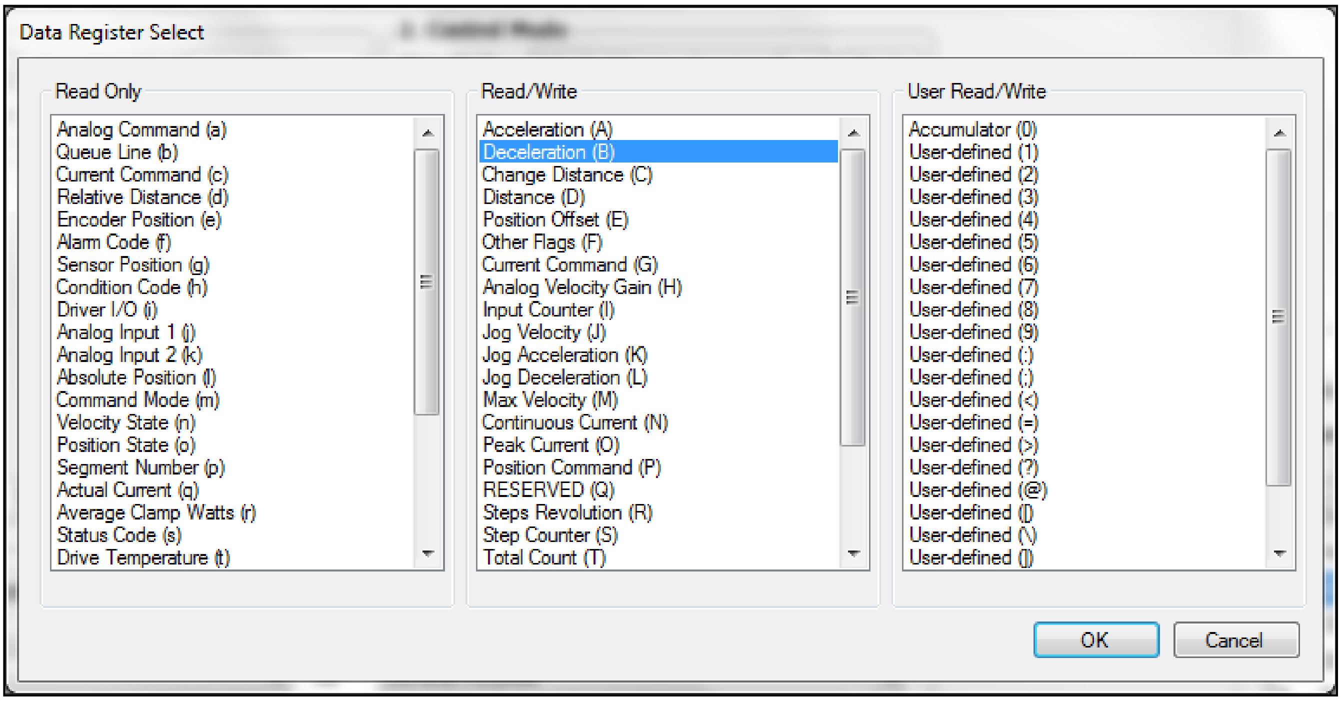
To observe drive parameters such as motor current or encoder position while testing a Q program, click the Param tab. The Pulse Counter, Command and Encoder Position can be reset to zero by clicking one of the buttons marked “C”.

8.5 Register Monitor



The Data Register Monitor tab let's you see your choice of eight data registers in real time. This is especially useful when manipulating registers within a Q program.

Select data register that are important to the program and application. To replace one of the registers with a different one, click on the register name, then click the "…" button that appears next to it. When the following Data Register Select dialog box appears, choose the desired register.



9 Appendix A: SCL Reference

The SCL serial command language gives users a simple way to control a motor drive via serial port. This eliminates the need for separate motion controllers to supply control signals, like pulse & direction or +/-10V, to your motor drive. It also provides an easy way to interface to a variety of other industrial devices like PLCs and HMIs, which frequently include serial ports for communicating with other devices.
For more details about SCL commands, please download latest *Host Command Reference* from our website. This document may be changed without notification to the customers.

9.1 Commands

There are two types of SCL commands: buffered and immediate. Buffered commands are loaded into and executed out of the drive's volatile command buffer, also known as the "queue". Immediate commands are not buffered: when received by the drive they are executed immediately.

9.1.1 Buffered Commands

After being loaded into the command buffer of a drive, buffered commands are executed one at a time. (See "Multi-tasking in Q Drives" below for an exception to this rule). If you send two buffered commands to the drive in succession, like an FL (Feed to Length) command followed by an SS (Send String) command, the SS command sits in the command buffer and waits to execute until the FL command is completed. The command buffer can be filled up with commands for sequential execution without the host controller needing to wait for a specific command to execute before sending the next command. Special buffer commands, like PS (Pause) and CT (Continue), enable the buffer to be loaded and to pause execution until the desired time.

Stored Programs in Q Drives

Stored Q programs, created with the *Q Programmer* software, are created using buffered commands.

Multi-tasking in Q Drives

Multi-tasking allows for an exception to the "one at a time" rule of buffered commands. The multi-tasking feature of a Q drive allows you to initiate a move command (FL, FP, CJ, FS, etc.) and proceed to execute other commands without waiting for the move command to finish.

9.1.2 Immediate Commands

Immediate commands are executed right away, running in parallel with a buffered command if necessary. For example, this allows you to check the remaining space in the buffer using the BS (Buffer Status) command, or the immediate status of digital inputs using the IS (Input Status) command, while the drive is processing other commands. Immediate commands are designed to access the drive at any time.
We recommend waiting for an appropriate Ack/Nack response from the drive before sending subsequent commands. This adds limited overhead but ensures that the drive has received and executed the current command, preventing many common communication errors. If the Ack/Nack functionality cannot be used in the application for any reason, the user should allow a 10ms delay between commands to allow the drive sufficient time to receive and act on the last command sent.
This approach allows a host controller to get information from the drive at a high rate, most often for checking drive status or motor position.

9.2 Using SCL Commands

The basic structure of a command packet from the host to the drive is always a text string followed by a carriage return (ASCII code 13). No line feed is required. The text string is always composed of the command itself, followed by any parameters used by the command. The carriage return denotes the end of transmission to the drive. Here is the basic syntax.

YXXAB<cr>
In the syntax above, “Y” symbolizes the drive’s RS-485 address, and is only required when using RS-485 networking. “XX” symbolizes the command itself, which is always composed of two capital letters. “A” symbolizes the first of two possible parameters, and “B” symbolizes the second. Parameters 1 and 2 vary in length, can be letters or numbers, and are often optional. The “<cr>” symbolizes the carriage return which terminates the command string. How the carriage return is generated in your application will depend on your host software. Once a drive receives the <cr> it will determine whether or not it understood the preceding characters as a valid command. If it did understand the command the drive will either execute or buffer the command. If Ack/ Nack is turned on (see PR command), the drive will also send an acknowledge character (Ack) back to the host. The Ack for an executed command is % (percent sign), and for a buffered command is * (asterisk).

It is always recommended that the user program wait for an ACK/NACK character before subsequent commands are sent. If the ACK/NACK functionality cannot be used in the application, a 10ms delay is recommended between non-motion commands.

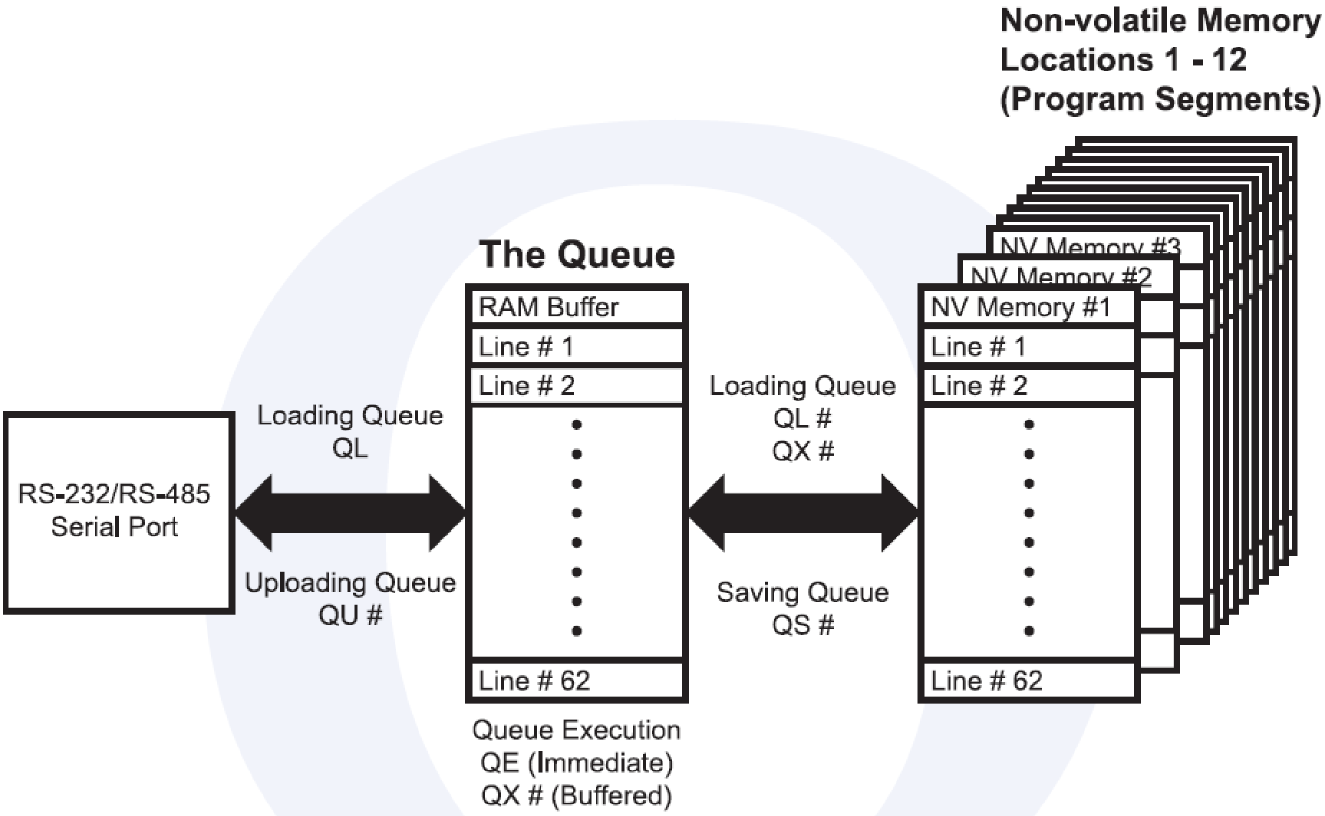
If the drive did not understand the command it will do nothing. If Ack/Nack is turned on a Nack will be sent, which is signified by a ? (question mark). The Nack is usually accompanied by a numerical code that indicates a particular error. To see a list of these errors see the PR command details in the Appendix.

Responses from the drive will be sent with a similar syntax to the associated SCL command.

YXX=A<cr>
In the syntax above, “Y” symbolizes the drive’s RS-485 address, and is only present when using RS-485 networking. “XX” symbolizes the command itself, which is always composed of two capital letters. “A” symbolizes the requested data, and may be presented in either decimal or hexadecimal format (see the IF command). The “<cr>” symbolizes the carriage return (ASCII code 13) which terminates the response string.

9.2.1 Commands in Q drives

Q drives have additional functionality because commands can also be composed into a stored program that the Q drive can run stand-alone. The syntax for commands stored in a Q program is the same as if the commands were being sent directly from the host, or “XXAB”. *Q Programmer™* is used to create stored Q programs and can be downloaded from www.applied-motion.com/products/software
The diagram below shows how commands sent from the host’s serial port interact with the volatile command buffer (AKA the Queue), and the drive’s non-volatile program memory storage. Loading and Uploading the Queue contents via the serial port are done with the QL and QU commands, respectively. Similarly, the Queue’s contents can be loaded from NV memory using the QL and QX commands, and can be saved to NV memory with the QS command. Finally, commands currently in the Queue can be executed with the QE or QX command.



The *Q Programmer* software automates many of the functions shown in the diagram above.

9.2.2 SCL Utility software

The *SCL Utility* is an excellent software application for familiarizing yourself with host commands. *SCL Utility* can be downloaded from www.applied-motion.com.

To send commands to your drive from *SCL Utility* simply type a command in the command line and press the ENTER key to send it. (Remember that all commands are capital letters so pressing the Caps Lock key first is a good tip). Pressing the ENTER key while in SCL Utility does two things: it terminates the command with a carriage return and automatically sends the entire string. Try the example sequence below. In this example, note that <ENTER> means press the ENTER key on your keyboard, which is the same as terminating the command with a carriage return.

IMPORTANT: We recommend practicing with SCL commands with no load attached to the motor shaft. You want the motor shaft to spin freely during startup to avoid damaging mechanical components in your system.

AC25<ENTER>	Set accel rate to 25 rev/sec/sec.
DE25<ENTER>	Set decel rate to 25 rev/sec/sec
VE5<ENTER>	Set velocity to 5 rev/sec
FL20000<ENTER>	Move the motor 20000 steps in the CW direction.

If your motor didn't move after sending the FL20000 check the LEDs on your drive to see if there is an error present. If so send the AR command (AR<ENTER>) to clear the alarm. If after clearing the alarm you see a solid green LED it means the drive is disabled. Enable the drive by sending the ME command (ME<ENTER>) and verify that you see a steady, flashing green LED. Then try the above sequence again.

Here is another sample sequence you can try.

JA10<ENTER> Set jog accel rate to 10 rev/sec/sec
JL10<ENTER> Set jog decel rate to 10 rev/sec/sec
JS1<ENTER> Set jog speed to 1 rev/sec
CJ<ENTER> Commence jogging
CS-1<ENTER> Change jog speed to 1 rev/sec in CCW direction
SJ<ENTER> Stop jogging

In the above sequence notice that the motor ramps to the new speed set by CS. This ramp is affected by the JA and JL commands. Try the same sequence above with different JA, JL, JS, and CS values to see how the motion of the motor shaft is affected.

9.3 Command Summary

This section contains a set of tables that list all of the SCL commands available with your drive. In each table there are a number of columns that give information about each command.

- “Command” shows the command’s two-letter Command Code.
- “Description” shows the name of each command.
- “NV” designates which commands are Non-volatile: that is, which commands are saved in non-volatile memory when the SA (Save) command is sent to the drive. Note that certain commands (PA, PB, PC, PI, and PM) save their parameter data to non-volatile memory immediately upon execution, and need not be followed by an SA command.
- “Write only” or “Read only” is checked when a command is not both Read/Write compatible.
- “Immediate” designates an immediate command (all other commands are buffered).
- “Compatibility” shows which drives use each of the commands.

The different categories for these tables - Motion, Servo, Configuration, I/O, Communications, Q Program, Register - are set up to aid you in finding particular commands quickly.

- “Motion” commands have to do with the actual shaft rotation of the step or servo motor.
- “Servo” commands cover servo tuning parameters, enabling / disabling the motor, and filter setup.
- “Configuration” commands pertain to setting up the drive and motor for your application, including tuning parameters for your servo drive, step resolution and anti-resonance parameters for your step motor drive, etc.
- “I/O” commands are used to control and configure the inputs and outputs of the drive.
- “Communications” commands have to do with the configuration of the drive’s serial ports.
- “Q Program” commands deal with programming functions when creating stored programs for your Q drive.
- “Register” commands deal with data registers. Many of these commands are only compatible with Q drives.

9.3.1 Motion Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
AC	Accel Rate	•				All drives
AM	Accel Max	•				All drives
CJ	Commence Jogging		•			All drives
DC	Distance for FC, FM, FO, FY	•				All drives
DE	Decel Rate	•				All drives
DI	Distance or Position	•				All drives
ED	Encoder Direction	•				Servos and steppers with encoder feedback

EF	Encoder Function	·				Servos and steppers with encoder feedback
EG	Electronic Gearing	·				All drives
EH	Extended Homing		·			All Step-Servo drives and M2 Servo drives
EI	Input Noise Filter	·				All drives
EP	Encoder Position					Servos and steppers with encoder feedback
FC	Feed to Length with Speed Change		·			All drives
FD	Feed to Double Sensor		·			All drives
FE	Follow Encoder		·			All drives
FH	Find Home		·			All Step-Servo drives and M2 Servo drives
FL	Feed to Length		·			All drives
FM	Feed to Sensor with Mask Dist.		·			All drives
FO	Feed to Length & Set Output		·			All drives
FP	Feed to Position		·			All drives
FS	Feed to Sensor		·			All drives
FY	Feed to Sensor with Safety Dist.		·			All drives
HA	Homing Acceleration	·				All Step-Servo drives and M2 Servo drives
HC	Hard Stop Current	·				All Step-Servo drives
HL	Homing Deceleration	·				All Step-Servo drives and M2 Servo drives
HO	Homing Offset	·				All Step-Servo drives and M2 Servo drives
HS	Hard Stop Homing		·			All Step-Servo drives
HV	Homing Velocity	·				All Step-Servo drives and M2 Servo drives
HW	Hand Wheel		·			All drives
JA	Jog Accel/Decel rate	·				All drives
JC	Velocity mode second speed	·				All drives
JD	Jog Disable		·			All drives
JE	Jog Enable		·			All drives
JL	Jog Decel rate	·				All drives
JM	Jog Mode	·				Al drives (see JM command)
JS	Jog Speed	·				All drives
MD	Motor Disable		·			All drives
ME	Motor Enable		·			All drives
MR	Micro step Resolution	·				Stepper drives only
PA	Power-up Accel Current	·				STM stepper drives only
SD	Set Direction	·				STM stepper drives with Flex I/O only
SH	Seek Home		·			All drives
SJ	Stop Jogging		·		·	All drives
SM	Stop the Move		·			Q drives only
SP	Set Absolute Position					All drives
ST	Stop Motion		·		·	All drives

VC	Velocity for Speed Change (FC)	·				All drives
VE	Velocity Setting (For Feed Commands)	·				All drives
VM	Velocity Max	·				All drives
WM	Wait on Move		·			Q drives only
WP	Wait on Position		·			Q drives only

9.3.2 Servo Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
CN	Second Control Mode	·				M2 servo drives only
CO	Node ID/ IP Address Series Number	·				M2 servo drives only
CP	Change Peak Current	·				Servo drives only
DD	Default Display Item of LEDs	·				M2 servo drives only
DS	Switching Electronic Gearing	·				M2 servo drives only
EN	Numerator of Electronic Gearing Ratio	·				M2 servo drives only
EP	Encoder Position					Servo drives only
EU	Denominator of Electronic Gearing Ratio	·				M2 servo drives only
FA	Function of the Single-ended Analog Input	·				M2 servo drives only
GC	Current Command	·			·	Servo drives only
GG	Controller Global Gain Selection	·				M2 servo drives only
IC	Immediate Current Command			·	·	Servo drives only
IE	Immediate Encoder Position			·	·	Servo drives only
IQ	Immediate Actual Current			·	·	Servo drives only
IX	Immediate Position Error			·	·	Servo drives only
JC	Eight Jog Velocities	·				M2 servo drives only
KC	Overall Servo Filter	·				Servo drives only
KD	Differential Constant	·				Servo drives only
KE	Differential Filter	·				Servo drives only
KF	Velocity Feedforward Constant	·				Servo drives only
KI	Integrator Constant	·				Servo drives only
KJ	Jerk Filter Frequency	·				SV7 Servo drives only
KK	Inertia Feedforward Constant	·				Servo drives only
KP	Proportional Constant	·				Servo drives only
KV	Velocity Feedback Constant	·				Servo drives only
MS	Control Mode Selection	·				M2 servo drives only
PF	Position Fault	·				Servo drives, drives with encoder feedback
PH	Inhibition of the pulse command	·				M2 servo drives only
PK	Parameter Lock	·				M2 servo drives only
PL	Position Limit	·				Servo drives only
PP	Power-Up Peak Current	·				Servo drives only
PV	Second Electronic Gearing	·				M2 servo drives only
TV	Torque Ripple	•				M2 servo drives only
VI	Velocity Integrator Constant	•				Servo drives only

VP	Velocity Mode Proportional Constant	•				Servo drives only
VR	Velocity Ripple	•				M2 servo drives only

9.3.3 Configuration Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
AL	Alarm Code			•	•	All drives
AR	Alarm Reset		•		•	All drives
BD	Brake Disengage Delay time	•				All drives
BE	Brake Engage Delay time	•				All drives
BS	Buffer Status			•	•	All drives
CA	Change Acceleration Current	•				STM stepper drives only
CC	Change Current	•				All drives
CD	Idle Current Delay	•				Stepper drives only
CF	Anti-resonance Filter Frequency	•				Stepper drives only
CG	Anti-resonance Filter Gain	•				Stepper drives only
CI	Change Idle Current	•				Stepper drives only
CM	Control mode	•				All drives
CP	Change peak current	•				Servo drives only
DA	Define Address	•				All drives
DL	Define Limits	•				All drives
DP	Dumping Power	•				SS drives only
DR	Data Register for Capture		•			Q servo drives only
ED	Encoder Direction	•				Servo drives, drives with encoder feedback
ER	Encoder or Resolution	•				Servo drives, drives with encoder feedback
HG	4th Harmonic Filter Gain	•				Stepper drives only
HP	4th Harmonic Filter Phase	•				Stepper drives only
IA	Immediate Analog			•	•	All drives
ID	immediate Distance			•	•	All drives
IE	Immediate Encoder			•	•	Servo drives, drives with encoder feedback
IF	Immediate Format	•			•	All drives
IQ	Immediate Current			•	•	Servo drives only
IP	Immediate Position			•	•	All drives
IT	Immediate Temperature			•	•	All drives
IU	Immediate Voltage			•	•	All drives
IV	Immediate Velocity			•	•	All drives
LP	Software Limit CW					All Step-Servo drives and M2 Servo drives
LM	Software Limit CCW					All Step-Servo drives and M2 Servo drives
LV	Low Voltage Threshold	•				All drives
MD	Motor Disable				•	All drives
ME	Motor Enable				•	All drives
MN	Model Number			•	•	All drives

MO	Motion Output	•				All drives
MR	Micro step Resolution	•				All drives (deprecated - see EG
MV	Model & Revision			•	•	All drives except Blu servos
OF	On Fault		•			Q drives only
OI	On Input		•			Q drives only
OP	Option Board	•		•	•	All drives
PA	Power-up Acceleration Current	•				
PC	Power up Current	•				All drives
PD	In Position Counts	•				All Step-Servo drives and M2 Servo drives
PE	In Position Timing	•				All Step-Servo drives and M2 Servo drives
PF	Position Fault	•				Servo drives, drives with encoder feedback
PI	Power up Idle Current	•				Stepper drives only
PL	In Position Limit	•				Servo drives only
PM	Power up Mode	•				All drives
PP	Power up peak current	•				Servo drives only
PW	Pass Word		•			Q drives only
RE	Restart / Reset		•		•	All drives
RL	Register Load				•	All drives
RS	Request Status			•	•	All drives
RV	Revision Level			•	•	All drives
SA	Save all NV Parameters		•			All drives
SC	Status Code			•	•	
SD	Set Direction	•				STM stepper drives with Flex I/O
SF	Step Filter Frequency	•				Stepper drives only
SI	Enable Input usage	•				All drives
SK	Stop & Kill		•		•	All drives
TT	Pulse Complete Timing	•				All Step-Servo drives and M2 Servo drives
ZC	Regen Resistor Continuous Wattage	•				BLuAC5 and STAC6 drives only
ZR	Regen Resistor Value	•				BLuAC5 and STAC6 drives only
ZT	Regen Resistor Peak Time	•				BLuAC5 and STAC6 drives only

9.3.4 I/O Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
AD	Analog Deadband	•				All stepper drives and SV servo drives
AF	Analog Filter	•				All drives
AG	Analog Velocity Gain	•				All stepper drives and SV servo drives
AI	Alarm Input usage	•				All drives
AN	Analog Torque Gain	•				All Step-Servo drives and M2 Servo drives
AO	Alarm Output usage	•				All drives
AP	Analog Position Gain	•				All drives

AS	Analog Scaling	•				All stepper drives and SV servo drives
AT	Analog Threshold	•				All drives
AV	Analog Offset	•				All drives
AZ	Analog Zero (Auto Zero)		•			All drives
BD	Brake Disengage Delay time	•				All drives
BE	Brake Engage Delay time	•				All drives
BO	Brake Output usage	•				All drives
DL	Define Limits	•				All drives
EI	Input Noise Filter	•				All drives
FI	Filter Input	•				All drives (Note: not NV on Blu servos)
FX	Filter Selected Inputs					Blu, STAC5, STAC6, SVAC3
IH	Immediate High Output		•		•	All drives
IL	Immediate Low Output		•		•	All drives
IO	Output Status				•	All drives
IS	Input Status request			•	•	All drives
MO	Motion Output	•				All drives
OI	On Input		•			Q drives only
SI	Enable Input usage	•				All drives
SO	Set Output		•			All drives
TI	Test Input		•			Q drives only
TO	Tach Output	•				TSM drives only
WI	Wait on Input		•			All drives

9.3.5 Communications Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
BR	Baud Rate	•				All drives
BS	Buffer Status				•	All drives
CE	Communications Error				•	All drives
IF	Immediate Format	•			•	All drives
PB	Power up Baud Rate	•				All drives
PR	Protocol	•				All drives
TD	Transmit Delay	•				All drives

9.3.6 Q Program Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
AX	Alarm Reset		•			All drives
MT	Multi-Tasking					Q drives only
NO	No Operation		•			Q drives only
OF	On Fault		•			Q drives only
OI	On Input		•			Q drives only
PS	Pause		•			All drives
QC	Queue Call		•			Q drives only
QD	Queue Delete		•			Q drives only

QE	Queue Execute		•		•	Q drives only
QG	Queue Goto		•			Q drives only
QJ	Queue Jump		•			Q drives only
QK	Queue Kill		•			Q drives only
QL	Queue Load		•		•	Q drives only
QR	Queue Repeat		•			Q drives only
QS	Queue Save		•		•	Q drives only
QU	Queue Upload			•	•	Q drives only
QX	Queue Load & Execute		•			Q drives only
SM	Stop Move		•			Q drives only
SS	Send String		•			All drives
TI	Test Input		•			Q drives only
WD	Wait Delay using Data Register		•			Q drives only
WI	Wait for Input		•			All drives
WM	Wait for Move to complete		•			Q drives only
WP	Wait for Position in complex move		•			Q drives only
WT	Wait Time		•			Q drives only

9.3.7 Register Commands

Command	Description	NV	write only	read only	Immediate	Compatibility
CR	Compare Register		•			Q drives only
DR	Data Register for Capture		•			Q drives only
RC	Register Counter		•			Q drives only
RD	Register Decrement		•			Q drives only
RI	Register Increment		•			Q drives only
RL	Register Load				•	Q drives only
RM	Register Move		•			Q drives only
RR	Register Read		•			Q drives only
RU	Register Upload		•		•	
RW	Register Write		•			Q drives only
RX	Register Load					Q drives only
R+	Register Addition		•			Q drives only
R-	Register Subtraction		•			Q drives only
R*	Register Multiplication		•			Q drives only
R/	Register Division		•			Q drives only
R&	Register Logical AND		•			Q drives only
R|	Register Logical OR		•			Q drives only
TR	Test Register		•			Q drives only
TS	Time Stamp read		•			Q drives only

10 Appendix B: *Q Programmer™* Reference

Q takes the SCL language to an entire new level by allowing users to create programs that can be stored in the drive’s flash memory. The drive can be set to automatically executed programs at power up or it can wait for the programs to be started and stopped using streaming commands over RS-232, RS-485, Ethernet, Modbus, CANopen or EtherNet/IP connections. Q programs can also be triggered using the drive's built-in I/O.

Q programs offer

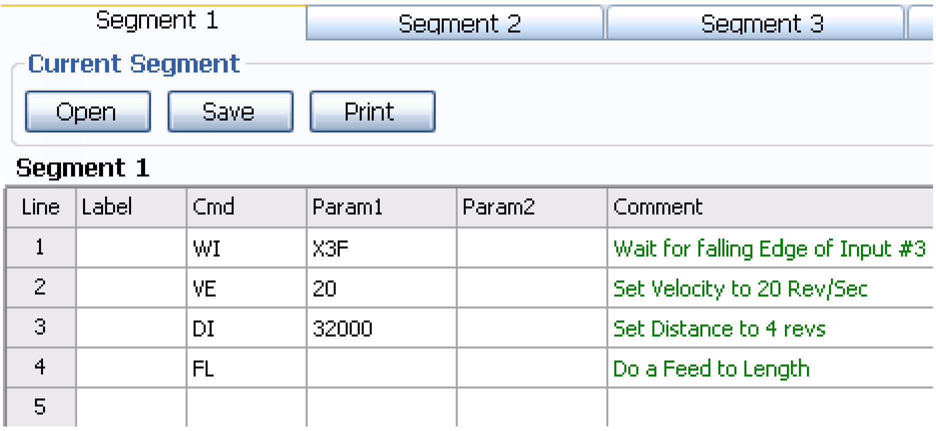
- Single-axis motion control
- Stand alone or networked operation
- Multi-tasking
- Conditional processing and program flow control
- Math calculations
- Data register manipulation
- User interaction via touch screen HMI's

10.1 Sample Command Sequences

What follows are sequences of commands that give examples of how to create motion and logic within a program. All of the commands in this section are buffered-type commands.

10.1.1 Feed to Length

The FL (Feed to Length) command is used for relative or incremental moves. When executed, the motor will move a fixed distance, using linear acceleration and deceleration ramps and a maximum velocity. These move parameters are set using the DI (Distance), AC (Acceleration), DE (Deceleration), and VE (Velocity) commands. The direction of the move is determined by the sign of the DI parameter. "DI32000" is 32000 counts in the CW direction, whereas "DI-32000" is 32000 counts in the CCW direction.

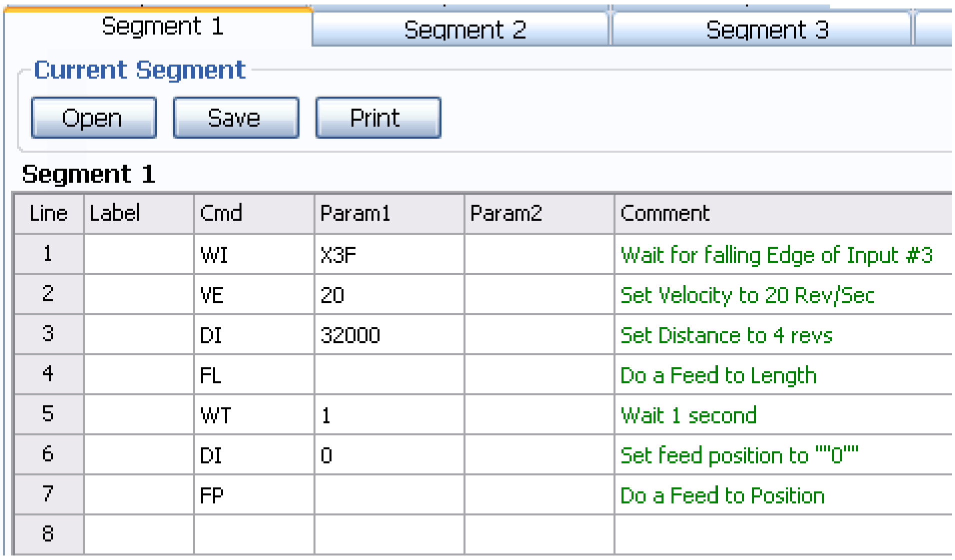


Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		VE	20		Set Velocity to 20 Rev/Sec
3		DI	32000		Set Distance to 4 revs
4		FL			Do a Feed to Length
5					

Here is a sample sequence showing a move of 80000 counts, with a velocity of 20 rps, and accel/decel rates of 500 rps/s. The FL command initiates the move. Also, the order of the commands is not significant, except that any changes to the move parameters must be done before the FL command.

10.1.2 Feed to Position

The FP (Feed to Position) command is used for absolute moves. When executed, the motor will move to a position, with linear acceleration and deceleration ramps and a maximum velocity, based on the internal motor position of the drive. The move parameters are set using the AC, DE, VE and DI commands. In the case of the FP command, the DI command sets the motor position, not the relative move distance.



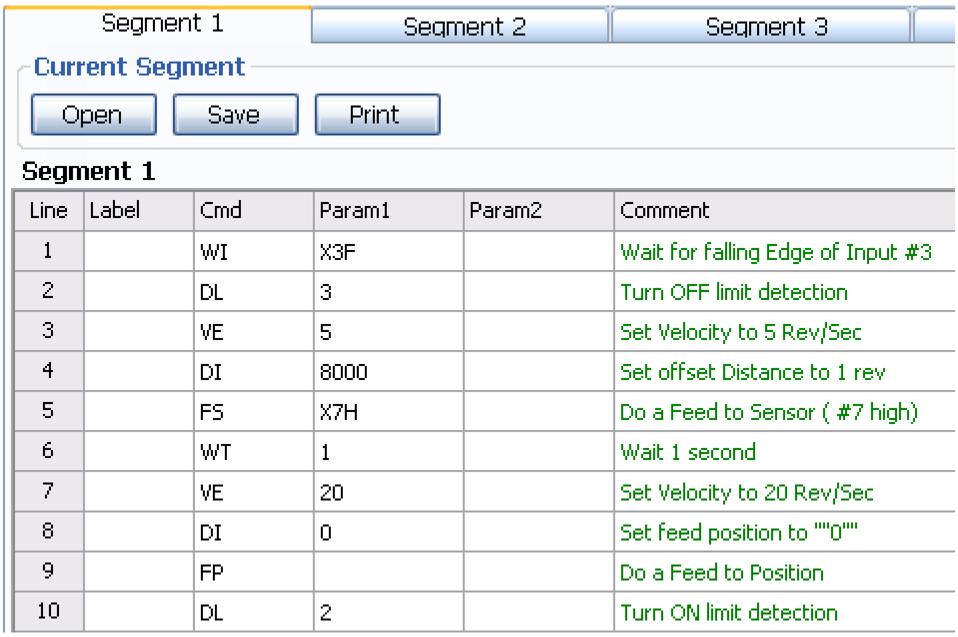
Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		VE	20		Set Velocity to 20 Rev/Sec
3		DI	32000		Set Distance to 4 revs
4		FL			Do a Feed to Length
5		WT	1		Wait 1 second
6		DI	0		Set feed position to ""0""
7		FP			Do a Feed to Position
8					

Here is a sample sequence showing a move to motor position 32000 counts (motor may move CW or CCW depending on the actual motor position before the start of the move), with a velocity of 20 rps and accel/ decel rates of 500 rps/s.

Another command to keep in mind when using absolute moves is the SP (Set Position) command. This command allows you to zero the motor position at any time, by entering “SP0”, or to set the motor position to another value. The parameter in the SP command is encoder counts. For example with a 2000 line encoder on the motor, an “SP5000” command would set the current motor position to 2.5 revolutions CW from the zero position.

10.1.3 Feed to Sensor

The FS (Feed to Sensor) command causes the motor to move at a fixed velocity until an input changes state. When the designated input changes state the motor decelerates to a stop. The parameters of the move are set by the AC, DE, VE and DI commands. In an FS command, the DI command sets both the distance in which the motor should stop after the input changes state and the direction of the move. Parameters for the FS command are the input number (0-7) and the input state the drive should look for: H (high), L (low), R (rising edge), or F (falling edge).

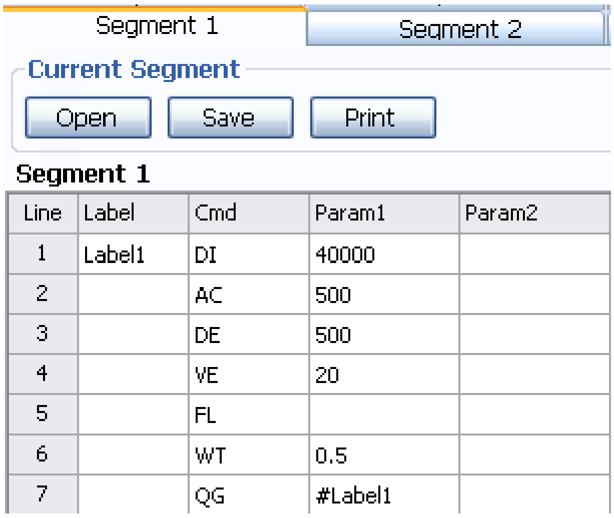


Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		DL	3		Turn OFF limit detection
3		VE	5		Set Velocity to 5 Rev/Sec
4		DI	8000		Set offset Distance to 1 rev
5		FS	X7H		Do a Feed to Sensor (#7 high)
6		WT	1		Wait 1 second
7		VE	20		Set Velocity to 20 Rev/Sec
8		DI	0		Set feed position to ""0""
9		FP			Do a Feed to Position
10		DL	2		Turn ON limit detection

Above is an example where the motor will move in the clockwise direction, starting off with an acceleration rate of 500 rps/s and a maximum speed of 5 rps, until drive input X7 goes high, at which point the drive will use the distance set in the DI command (8000 counts) and the deceleration rate set in the DE command (500 rps/s) to bring the motor to a stop.

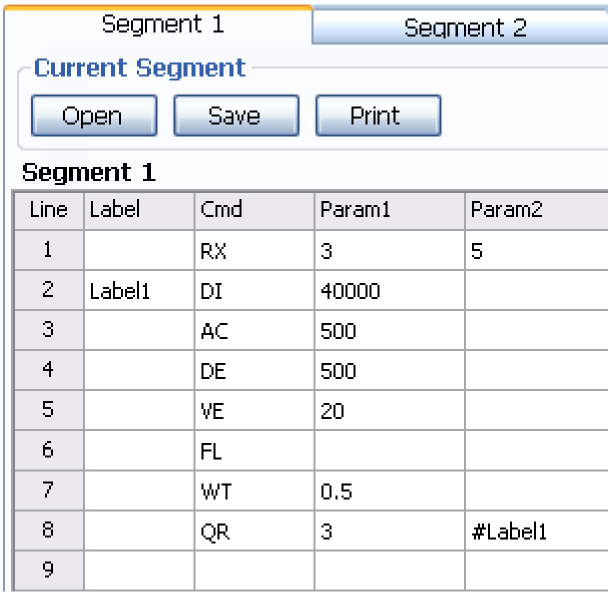
10.1.4 Looping

There are two ways to accomplish looping, or repeat loops, within a pro-gram. The first method accomplishes an infinite loop and uses the QG (Queue Goto) command. The parameter for this command is a line number in the segment, and whenever the sequence gets to the QG command the segment will jump to the designated line.



Line	Label	Cmd	Param1	Param2
1	Label1	DI	40000	
2		AC	500	
3		DE	500	
4		VE	20	
5		FL		
6		WT	0.5	
7		QG	#Label1	

In the example to above, the sequence contains an FL command, with related parameter commands ahead of it (AC, DE, DI, VE). After the FL command is a WT (Wait Time) command with a time of 0.5 seconds, and then a QG command that points to line 1. This sequence will loop forever now, with the segment always starting at line one after it executes the QG command.

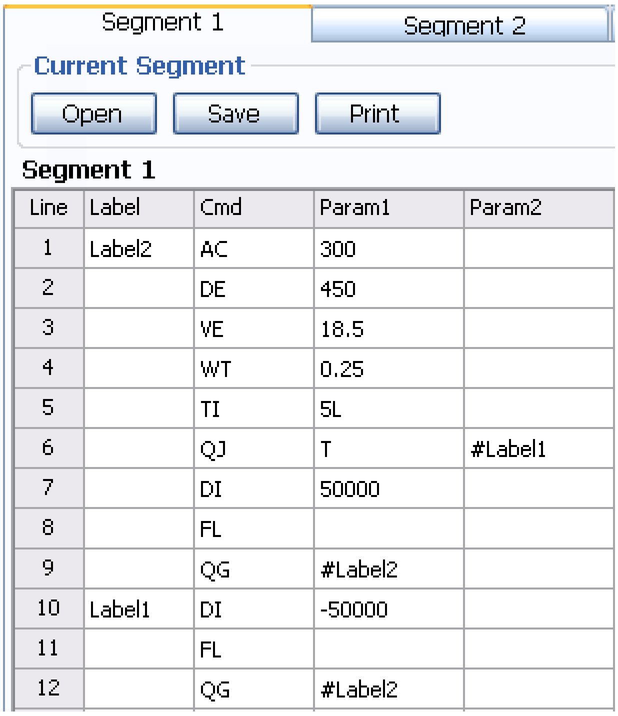


Line	Label	Cmd	Param1	Param2
1		RX	3	5
2	Label1	DI	40000	
3		AC	500	
4		DE	500	
5		VE	20	
6		FL		
7		WT	0.5	
8		QR	3	#Label1
9				

The second method for looping utilizes the QR (Queue Repeat) command. It works by jumping to a given segment line for the number of times indicated in a user-defined data register. Any user-defined data register will work. In the example to the right, the QG command from the previous example has been replaced with the QR command, and parameters have been added. In this sequence the segment will jump to line 2 for the number of times indicated in register 3. Notice on line 1 of the segment that data register 3 has been loaded (using the RX command) with the value 5. Therefore, the FL command in this example (as well as the DI, AC, DE, VE and WT commands) will repeat five times.

10.1.5 Branching

Branching in a program is done using the QJ (Queue Jump) command. Branching is different than looping in that a branch (or jump) is done based on a tested condition. The QJ command will always work in conjunction with one other command: TI (Test Input), TR (Test Register), or CR (Compare Register).



Line	Label	Cmd	Param1	Param2
1	Label2	AC	300	
2		DE	450	
3		VE	18.5	
4		WT	0.25	
5		TI	5L	
6		QJ	T	#Label1
7		DI	50000	
8		FL		
9		QG	#Label2	
10	Label1	DI	-50000	
11		FL		
12		QG	#Label2	

Let's say we have an application with two possible moves. We always want to make a CW move, unless input X5 is low in which case we want to make a CCW move. In this example we set all of the move parameters except distance at the top of the segment. We set accel to 300 rps/s, decel to 450 rps/s, and velocity to 18.5 rps. There is a WT (Wait Time) of 0.25 seconds so that we may have a noticeable delay between moves. Then, we test input X5 to see if it's low using the TI (Test Input) command. If it is true (i.e. input X5 is low), we branch (using QJ) to line 10, set the distance to -50000 counts and make a CCW move. Otherwise the program proceeds to line 7, sets the distance to 50000 counts and makes the CW move. To keep from doing the CCW move right after the CW move, and to repeat the segment forever QG commands are placed after each FL command.

10.1.6 Calling

Calling is a way to use sub-routines in your Q program. The QC (Queue Call) command allows us to exit a segment, execute another segment, and then return to the original segment to the line right after the "call". Subroutines are useful when we have a sequence of commands that is used often in a program. Rather than repeatedly program these commands into our segment(s), we place the frequently-used sequence in its own segment, and then call that segment whenever we need it.

Segment 1 | Segment 2

Current Segment

Open | Save | Print

Segment 1

Line	Label	Cmd	Param1	Param2
1	Label1	AC	300	
2		DE	450	
3		VE	18.5	
4		DI	40000	
5		FL		
6		QC	2	
7		VE	1	
8		DI	4000	
9		FL		
10		QC	2	
11		QG	#Label1	

In this example we are making two distinct moves (FL), one fast move and one slow move. After each move we'd like to turn 2 outputs on and off. To accomplish this using the QC command, we must program two segments. In this example, segment 1 is the primary (or calling) segment, and in it we program the two distinct FL commands. We are using the same accel and decel rates for the two moves, but the velocities and distances change. After each move we'd like to set outputs Y1 and Y2 on then off, and rather than entering the necessary commands to do this after each FL command in segment 1, we place the commands in segment 2 and then use the QC command to call it.

Segment 1 | Segment 2

Current Segment

Open | Save | Print

Segment 2

Line	Label	Cmd	Param1	Param2
1		SO	1L	
2		WT	0.25	
3		SO	2L	
4		WT	0.25	
5		SO	2H	
6		WT	0.25	
7		SO	1H	
8		QC	1	

In segment 2 we place the desired SO (Set Output) commands that turn output Y1 on, then output Y2 on, then output Y2 off and finally output Y1 off. Notice we've also placed WT (Wait Time) commands of 0.25 seconds between each SO command to make the changing output states more noticeable. When segment 1 reaches its first QC command (with the parameter "2" indicating a call to segment 2), the program will call segment 2 as a subroutine to execute its sequence of commands. Notice at the end of segment 2 we've placed a QC command with no parameter. That's a "return", which tells the drive to return to the calling segment and execute the line right after the original Queue Call.

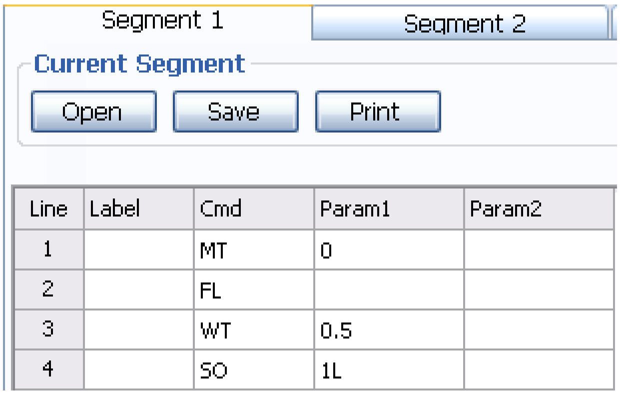
Upon returning to segment 1, the drive completes the second move, calls segment 2 again, then returns to segment 1 once more, and then starts the process over by looping back to line 1 (“QG1”).

10.1.7 Multi-tasking

The multi-tasking feature of Q drives allows you to initiate a move command (FL, FP, CJ, FS, etc.) and proceed to execute other commands without waiting for the move command to finish. Without multi-tasking (or more accurately with multi-tasking turned off), a Q drive always executes commands in succession by waiting for the completion of a particular command before moving on to the next command. In the case of move commands, this means waiting for the move to finish before executing subsequent commands. For example, if you have an FL command (Feed to Length - incremental move) followed by an SO command (Set Output), the drive will wait to finish the motor move before setting the drive’s digital output.

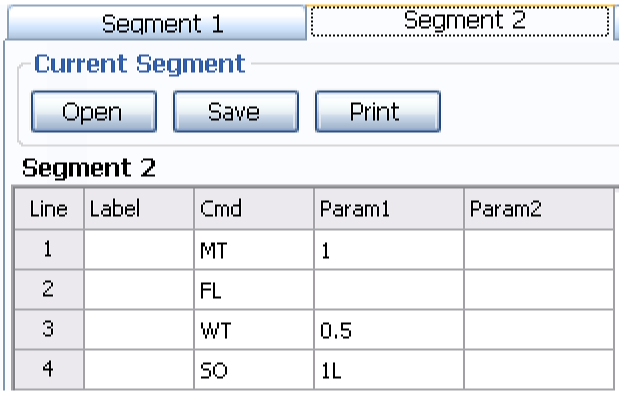
With multi-tasking turned on, a Q drive initiates a move command and then immediately proceeds to execute subsequent commands. For example, doing the same FL and SO commands as above, but this time with multi-tasking turned on, the drive will initiate the move command and immediately proceed to execute the set output command without waiting for the move command to finish. Multi-tasking is turned on and off with the MT command. “MT1” turns multi-tasking on, and “MT0” turns it off.

To illustrate the use of the MT command some more, here are a couple of sample command sequences.



Line	Label	Cmd	Param1	Param2
1		MT	0	
2		FL		
3		WT	0.5	
4		SO	1L	

In the above command sequence to the right, notice that multi-tasking is turned off, “MT0”. When this sequence is executed by a drive, the FL (Feed to Length) incremental move will complete before the drive waits 0.5 seconds (WT0.50) and then sets output 1 low (SOY1L).



Line	Label	Cmd	Param1	Param2
1		MT	1	
2		FL		
3		WT	0.5	
4		SO	1L	

In the above command sequence to the right, notice that multi-tasking is turned on, “MT1”. When this sequence is executed by the drive, the drive will not wait for the FL command to complete before executing the WT and SO commands. In other words, the drive will initiate the FL command, then wait 0.50 seconds, and then set output 1 low. If the last distance set by the DI command is sufficiently long, the drive’s output 1 will be set low before the FL command has completed.

This example is actually quite basic, even though it illustrates the function of multi-tasking well. If you try these sequences with your drive, make sure the last DI command is sufficiently large enough to see a noticeable difference in when the drive sets the output.

NOTE: Because it is physically impossible for a motor to make two moves at the same time, move commands are always blocked even with Multi-tasking turned on. For example, if you have Multi-tasking turned on and the program has two move commands in a row, the drive will wait to execute the second move command until the first move command is finished.

11 Appendix C: CANopen Reference

11.1 CANopen Communication

CANopen is a communication protocol and device profile specification for embedded systems used in automation. In terms of the OSI model, CANopen implements the layers above and including the network layer. The CANopen standard consists of an addressing scheme, several small communication protocols and an application layer defined by a device profile. The communication protocols have support for network management, device monitoring and communication between nodes, including a simple transport layer for message segmentation/desegmentation. The lower level protocol implementing the data link and physical layers is usually Controller Area Network (CAN)

The basic CANopen device and communication profiles are given in the CiA 301 specification released by CAN in Automation. [1] Profiles for more specialized devices are built on top of this basic profile, and are specified in numerous other standards released by CAN in Automation, such as CiA 401[2] for I/O-modules and CiA 402[3] for motion control.

11.2 Why CANopen?

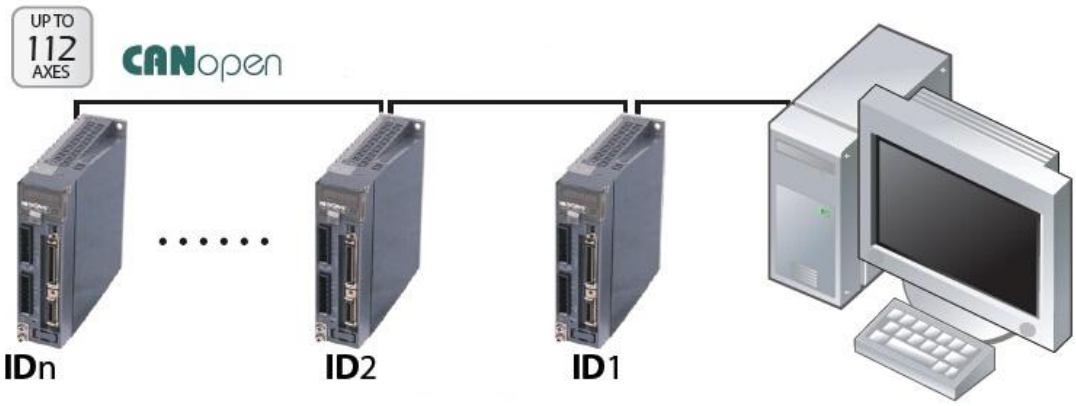
Multi-axis Control

Up to 127 axis can be supported via CANopen, and the maximum communication baud rate is up to 1Mbps.

A further advantage with CAN is the Multi-Master Capability. This means that each user on the bus has the same access rights. The access authorization alone controls the users among one another via the priority of the communication objects and their identifiers (arbitration). This allows direct communication between the individual users without a time-consuming "detour" over a central master.

Easy Wiring

A shielded twisted pair cable is be used as the bus cable. Less cable will cause less error, reduce the wiring cost, labor cost, whilst maintaining availability and minimizing cost.



11.3 CANopen Example Programs

11.3.1 Profile Position Mode

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Operation Enabled
**** Set to Profile Position Mode ****
$0603 $8 $2F $60 $60 $00 $01 $00 $00 $00 ‘Set to Profile Position Mode
**** Set Motion Parameters ****
$0603 $8 $23 $81 $60 $00 $F0 $00 $00 $00 ‘Set Profile Velocity to 1 rps
$0603 $8 $23 $83 $60 $00 $58 $02 $00 $00 ‘Set Acceleration to 100 rps/s
$0603 $8 $23 $84 $60 $00 $58 $02 $00 $00 ‘Set Deceleration to 100 rps/s
Single Move Absolute
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Clear New Set Point Bit
Single Move Relative
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Stopping between Moves
$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Continuous Motion
$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $02 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $5F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $02 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Immediate Change in Motion
$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $7F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $6F $02 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $7F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $6F $02 $00 $00 ‘Clear New Set Point Bit

11.3.2 Profile Velocity Mode

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 'Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 'Switched on
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 'Operation Enabled; Motion Halted
**** Set to Profile Velocity Mode ****
$0603 $8 $2F $60 $60 $00 $03 $00 $00 $00 'Set to Profile Velocity Mode
**** Set Motion Parameters ****
$0603 $8 $23 $FF $60 $00 $F0 $00 $00 $00 'Set Target Velocity to 1 rps
$0603 $8 $23 $83 $60 $00 $58 $02 $00 $00 'Set Acceleration to 100 rps/s
$0603 $8 $23 $84 $60 $00 $58 $02 $00 $00 'Set Deceleration to 100 rps/s
**** Start/Stop Motion ****
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 'Motion Starts
$0603 $8 $23 $FF $60 $00 $60 $09 $00 $00 'Change Target Velocity to 10 rps
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 'Motion Halts

11.3.3 Homing Mode

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 'Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 'Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 'Operation Enabled
**** Set to Homing Mode ****
$0603 $8 $2F $60 $60 $00 $06 $00 $00 $00 'Set to Homing Mode
$0603 $8 $2F $98 $60 $00 $13 $00 $00 $00 'Set Homing Method to 19
**** Set Motion Parameters ****
$0603 $8 $23 $9A $60 $00 $58 $02 $00 $00 'Set Homing Acceleration to 100rps/s
$0603 $8 $23 $99 $60 $01 $F0 $00 $00 $00 'Set Homing Velocity (Search for Switch) to 1rps
$0603 $8 $23 $99 $60 $02 $78 $00 $00 $00 'Set Index Velocity (Search for Index or Zero) to 0.5rps
$0603 $8 $23 $7C $60 $00 $40 $9C $00 $00 'Set Homing Offset to 40000 Steps
$0603 $8 $2F $01 $70 $00 $03 $00 $00 $00 'Set Homing Switch to Input 3
**** Start/Stop Homing ****
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 'Homing Starts
$0603 $8 $2B $40 $60 $00 $1F $01 $00 $00 'Homing Stops

11.3.4 Normal Q Mode

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 'Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 'Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 'Operation Enabled
**** Set to Normal Q Mode ****
$0603 $8 $2F $60 $60 $00 $FF $00 $00 $00 'Set to Normal Q Mode
$0603 $8 $2F $07 $70 $00 $01 $00 $00 $00 'Set Q Segment Number to 1
**** Start/Stop Q Program ****
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 'Q Program Starts
$0603 $8 $2B $40 $60 $00 $1F $01 $00 $00 'Q Program Halts

11.3.5 Sync Q Mode

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 'Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 'Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 'Operation Enabled
**** Set to Sync Q Mode ****
$0603 $8 $2F $60 $60 $00 $FE $00 $00 $00 'Set to Sync Q Mode
$0603 $8 $2F $07 $70 $00 $01 $00 $00 $00 'Set Q Segment Number to 1
$0603 $8 $23 $05 $10 $00 $80 $00 $00 $00 'Set Sync Pulse to 0x80
**** Start/Stop Q Program ****
$80 $0 'Q Program Starts
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 'Q Program Halts

11.3.6 PDO Mapping

****Mapping TPDO2 ****
$0000 $2 $80 $03 'Return back to "PreOperation" Mode
$0603 $8 $23 $01 $18 $01 $80 $02 $00 $80 'Turn off the TPDO2
$0603 $8 $2F $01 $1A $00 $00 $00 $00 $00 'Set Number of Mapped objects to zero
$0603 $8 $23 $01 $1A $01 $10 $00 $41 $61 'Map object1 (0x6041) to TPDO2 subindex1.
$0603 $8 $23 $01 $1A $02 $20 $00 $0A $70 'Map object2 (0x700A) to TPDO2 subindex2.
$0603 $8 $2F $01 $1A $00 $02 $00 $00 $00 'Set Number of total Mapped objects to two
$0603 $8 $23 $01 $18 $01 $80 $02 $00 $00 'Turn on the TPDO2

11.4 Downloads

CANopen EDS files and the CANopen User Manual can be downloaded from the product page of your drive. You can find your drive's product page by visiting the SV200 Series Page and selecting it from the list.

12 Appendix D: Modbus/RTU Reference

The Q versions of Applied Motion servo drives support the Modbus/RTU communication protocol. Since Modbus is a master/slave protocol, that means one node is a master and the others are slave nodes. Our drives are always used as Modbus slaves. The host is typically a PLC, motion controller or HMI. Each device intended to communicate using Modbus is given a unique address. Only the node assigned as the Master may initiate a command.

A Modbus command contains the Modbus address of the device for which it is intended. Only the intended device will act on the command, even though other devices might receive it (an exception is specific broadcast able commands sent to node 0 which are acted on but not acknowledged). All Modbus commands contain checksum information, to allow the recipient to detect transmission errors. The basic Modbus commands can instruct an RTU to change the value in one of its registers, control or read an I/O port, and command the device to send back one or more values contained in its registers.

12.1 Communication Address

In the network system, each drive requires a unique drive address. Only the drive with the matching address will responded to the host command. In Modbus network, address "0" is the broadcast address. It cannot be used for individual drive's address. Modbus RTU/ASCII can set drive address from 1 to 31.

12.2 Data Encoding

Big-endian: The most significant byte (MSB) value is stored at the memory location with the lowest address; the next byte value in significance is stored at the following memory location and so on. This is akin to Left-to-Right reading in hexadecimal order.
For example: To store a 32bit data 0x12345678 into register address 40031 and 40032. 0x1234 will be defined as MSB, and 0x5678 as LSB. With big-endian system

Register 40031 = 0x1234

Register 40032 = 0x5678

When transfer 0x12345678, the first word will be 0x1234, and the second word will be 0x5678
Little-endian: The most significant byte (MSB) value is stored at the memory location with the highest address; the next byte value in significance is stored at the following memory location and so on. This is akin to Left-to-Right reading in hexadecimal order.
For example: To store a 32bit data 0x12345678 into register address 40031 and 40032. 0x5678 will be defined as MSB, and 0x1234 as LSB. With little-endian system

Register 40031 = 0x5678

Register 40032 = 0x1234

When transfer 0x12345678, the first words will be 0x5678, and the second words will be 0x1234
PR defines data transfer type.

12.3 Communication Baud Rate & Protocol

SV200 series servo drives have fixed communication data framing: 8 data bits, no parity checking, one stop bit.

BR and **PB** define the communication baud rate.
In serial communication, the change of baud rate will NOT take effect immediately, ONLY on next power up of the drive.
1 = 9600bps
2 = 19200bps
3 = 38400bps
4 = 57600bps
5 = 115200bps

12.4 Function Code

Applied Motion drives currently support following Modbus function codes:
1) 0x03: Read holding registers
2) 0x04: Read input registers
3) 0x06: Write single registers
4) 0x10: Write multiple registers

12.4.1 Function Code 0X03, Reading Multiple Holding Registers

If we want to read encoder's actual position command to drive Node ID 1, the data address for encoder's actual position is register 40005. If the register value is in decimal numbers it will be 250000, and the transfer method is P-75 (PR) = 5, for big-endian transfer.

Communication details are:

Command Message（Master）		Response Message（slave）

Function	Data	Bytes		Function	Data	Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	03H	1		Function Code	03H	1
Starting Data Address (Register 40005)	00H(High) 04H(Low)	2		Number of Data （In Byte）	04	1
Number of Data words	00(High) 02(Low)	2		Content of Starting Data Address 40005	00H(High) 26H(Low)	2
CRC Check Low	85	1		Content of second Data Address 40006	25H(High) A0(Low)	2
CRC Check High	CA	1		CRC Check Low	01H	1
				CRC Check High	10H	1

Host Sending: 01 03 00 04 00 02 85 CA
Drive Reply: 01 03 04 00 26 25 A0 01 10

If error is occurred, drive reply format: 01 83 XX CRC_L CRC_H
Where

XX = 01 : Function code 03 unsupported

XX = 02 : Incorrect reading on driving address or numbers

XX = 03 : Reading register address out of range

XX = 04 : Reading failure

12.4.2 Function Code 0x06, Writing Single Register

If we want to set motor speed to 12.5 rps on drive node ID 11, the corresponding address is register 40030. The write in data value for the register will be 12.5 x 240 = 3000. In hexadecimal number, it is 12CH.

Communication details are:

Command Message（Master）				Response Message（slave）		
Function	Data	Number of Bytes		Function	Data	Number of Bytes
Slave Address	0BH	1		Slave Address	0BH	1
Function Code	06H	1		Function Code	06H	1
Starting Data Address (Register 40030)	00H(High) 1DH(Low)	2		Starting Data Address (Register 40030)	00H(High) 1DH(Low)	2
Content of Data	01(High) 2C(Low)	2		Content of Data	01(High) 2C(Low)	2
CRC Check Low	19	1		CRC Check Low	19	1
CRC Check High	2B	1		CRC Check High	2B	1

Host Sending: 0B 06 00 1D 01 2C 19 2B

Drive Reply: 0B 06 00 1D 01 2C 19 2B

If error is occurred, drive reply format: 01 86 XX CRC_L CRC_H

Where

XX = 01 ： Function code 06 unsupported

XX = 02 ： Incorrect writing on driving address or number

XX = 03 ： Writing register address out of range

XX = 04 ： Writing failure

12.4.3 Function Code 0X10, Writing Multiple Registers

If we writing target distance 30000 (7530h) into drive NODE-ID 10, the corresponding register address will be 40031.

Communication Details are:

Command Message（Master）		
Function	Data	Bytes
Slave Address	0AH	1
Function Code	10H	1
Starting Data Address (Register 40031)	00H(High) 1EH(Low)	2
Number of Data words	00H(High) 02H(Low)	2
Number of Data bytes	04H	1
Content of first Data address	00(High) 00(Low)	2
Content of second Data address	75H(High) 30H(Low)	2
CRC Check Low	70	1
CRC Check High	8F	1

Response Message（slave）		
Function	Data	Bytes
Slave Address	0AH	1
Function Code	10H	1
Starting Data Address (Register 40031)	00H(High) 1EH(Low)	2
Number of Data（In word）	00H(High) 02H(Low)	2
CRC Check Low	20	1
CRC Check High	B5	1

Host Sending： 0A 10 00 1E 00 02 04 00 75 30 70 8F

Drive Reply： 0A 10 00 1E 00 02 20 B5

If error has occurred, the drive will reply in this format: 01 90 XX CRC_L CRC_H

Where

XX = 01 ： Function code 10 unsupported

XX = 02 ： Incorrect reading on driving address or number

XX = 03 ： Reading register address out of range

XX = 04 ： Reading failure

12.5 Modbus/RTU Data Frame

Modbus RTU is a master and slave communication system. The CRC checking code includes from drive's address bits to data bits. This standard data framing is as follows:

Address	Function Code	Data	CRC

Based on data transfer status, there can be two types of response code:

Normal Modbus response:

Response function code = request function code

Modbus error response:

Response function code = request function code + 0x80

12.6 Modbus Register Table

Modbus Register Table				
Register	**Access**	**Data Type**	**SCL Command**	**Map Register**
40001	Read	SHORT	Alarm Code (AL)	f
40002	Read	SHORT	Status Code (SC)	s
40003	Read	SHORT	Immediate Expanded Inputs (IS)	y
40004	Read	SHORT	Driver Board Inputs (ISX)	i
40005..6	Read	LONG	Encoder Position (IE, EP)	e
40007..8	Read	LONG	Immediate Absolute Position	l
40009..10	Write	LONG	Absolute Position Command	P
40011	Read	SHORT	Immediate Actual Velocity (IV0)	v
40012	Read	SHORT	Immediate Target Velocity (IV1)	w
40013	Read	SHORT	Immediate Drive Temperature (IT)	t
40014	Read	SHORT	Immediate Bus Voltage (IU)	u
40015..16	Read	LONG	Immediate Position Error (IX)	x
40017	Read	SHORT	Immediate Analog Input Value (IA)	a
40018	Read	SHORT	Q Program Line Number	b
40019	Read	SHORT	Immediate Current Command (IC)	c
40020..21	Read	LONG	Relative Distance (ID)	d
40022..23	Read	LONG	Sensor Position	g
40024	Read	SHORT	Condition Code	h
40025	Read	SHORT	Analog Input 1 (IA1)	j
40026	Read	SHORT	Analog Input 2 (IA2)	k
40027	Read	SHORT	Command Mode (CM)	m
40028	R/W	SHORT	Point-to-Point Acceleration (AC)	A
40029	R/W	SHORT	Point-to-Point Deceleration (DE)	B
40030	R/W	SHORT	Velocity (VE)	V
40031..32	R/W	LONG	Point-to-Point Distance (DI)	D
40033..34	R/W	LONG	Change Distance (DC)	C
40035	R/W	SHORT	Change Velocity (VC)	U

40036	Read	SHORT	Velocity Move State	n
40037	Read	SHORT	Point-to-Point Move State	o
40038	Read	SHORT	Q Program Segment Number	p
40039	Read	SHORT	Average Clamp Power (regen)	r
40040	Read	SHORT	Phase Error	z
40041..42	R/W	LONG	Position Offset	E
40043	R/W	SHORT	Miscellaneous Flags	F
40044	R/W	SHORT	Current Command (GC)	G
40045..46	R/W	LONG	Input Counter	I
40047	R/W	SHORT	Jog Accel (JA)	
40048	R/W	SHORT	Jog Decel (JL)	
40049	R/W	SHORT	Jog Velocity (JS)	J
40050	R/W	SHORT	Accel/Decel Current (CA)	
40051	R/W	SHORT	Running Current (CC)	N
40052	R/W	SHORT	Idle Current (CI)	
40053	R/W	SHORT	Steps per Revolution	R
40054	R/W	SHORT	Pulse Counter	S
40055	R/W	SHORT	Time Stamp	W
40056	R/W	SHORT	Analog Position Gain (AP)	X
40057	R/W	SHORT	Analog Threshold (AT)	Y
40058	R/W	SHORT	Analog Offset (AV	Z
40059..60	R/W	LONG	Accumulator	0
40061..62	R/W	LONG	User Defined	1
40063..64	R/W	LONG	User Defined	2
40065..66	R/W	LONG	User Defined	3
40067..68	R/W	LONG	User Defined	4
40069..70	R/W	LONG	User Defined	5
40071..72	R/W	LONG	User Defined	6
40073..74	R/W	LONG	User Defined	7
40075..76	R/W	LONG	User Defined	8
40077..78	R/W	LONG	User Defined	9
40079..80	R/W	LONG	User Defined	:
40081..82	R/W	LONG	User Defined	;
40083..84	R/W	LONG	User Defined	<
40085..86	R/W	LONG	User Defined	=
40087..88	R/W	LONG	User Defined	>
40089..90	R/W	LONG	User Defined	?
40091..92	R/W	LONG	User Defined	@
40093..94	R/W	LONG	User Defined	[
40095..96	R/W	LONG	User Defined	\
40097..98	R/W	LONG	User Defined	]

40099..100	R/W	LONG	User Defined	^
40101..102	R/W	LONG	User Defined	_
400103..104	R/W	LONG	User Defined	`
40105	R/W	SHORT	Brake Release Delay	
40106	R/W	SHORT	Brake Engage Delay	
40107	R/W	SHORT	Idle Current Delay	
40108	R/W	SHORT	Hyperbolic Smoothing Gain	
40109	R/W	SHORT	Hyperbolic Smoothing Phase	
40110	R/W	SHORT	Analog Filter Gain	
40111..124			(reserved)	
40125	R/W	SHORT	Command Opcode	
40126	R/W	SHORT	Parameter 1	
40127	R/W	SHORT	Parameter 2	
40128	R/W	SHORT	Parameter 3	
40129	R/W	SHORT	Parameter 4	
40130	R/W	SHORT	Parameter 5	

12.7 Command Register

Register 40125 is defined as the command register. When a command opcode is written to this register, the drive will execute the corresponding command.

1) SCL Command Encoding Table

SCL Command Encoding Table							
Function	SCL	Opcode	Parameter1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Alarm Reset	AX	0xBA	×	×	×	×	×
Start Jogging	CJ	0x96	×	×	×	×	×
Stop Jogging	SJ	0xD8	×	×	×	×	×
Encoder Function	EF	0xD6	0,1,2 or 6	×	×	×	×
Encoder Position	EP	0x98	Position	×	×	×	×
Feed to Double Sensor	FD	0x69	I/O Point 1	Condition 1	I/O Point 2	Condition 2	×
Follow Encoder	FE	0xCC	I/O Point	Condition	×	×	×
Feed to Length	FL	0x66	×	×	×	×	×
Feed to Sensor with Mask Distance	FM	0x6A	I/O Point	Condition	×	×	×
Feed and Set Output	FO	0x68	I/O Point	Condition	×	×	×
Feed to Position	FP	0x67	×	×	×	×	×
Feed to Sensor	FS	0x6B	I/O Point	Condition	×	×	×
Feed to Sensor with Safety Distance	FY	0x6C	I/O Point	Condition	×	×	×
Jog Disable	JD	0xA3	×	×	×	×	×
Jog Enable	JE	0xA2	×	×	×	×	×

Motor Disable	MD	0x9E	×	×	×	×	×
Motor Enable	ME	0x9F	×	×	×	×	×
Seek Home	SH	0x6E	I/O Point	Condition	×	×	×
Set Position	SP	0xA5	Position	×	×	×	×
Filter Input	FI	0xC0	I/O Point	Filter Time	×	×	×
Filter Select Inputs	FX	0xD3	×	×	×	×	×
Step Filter Freq	SF	0x06	Freq	×	×	×	×
Analog Deadband	AD	0xD2	0.001 V	×	×	×	×
Alarm Reset Input	AI	0x46	'1', '2' or '3'	I/O Point	×	×	×
Alarm Output	AO	0x47	'1', '2' or '3'	I/O Point	×	×	×
Analog Scaling	AS	0xD1	×	×	×	×	×
Define Limits	DL	0x42	'1', '2' or '3'	×	×	×	×
Set Output	SO	0x8B	I/O Point	Condition	×	×	×
Wait for Input	WI	0x70	×	×	×	×	×
Queue Load & Execute	QX	0x78	1..12	×	×	×	×
Wait Time	WT	0x6F	0.01 sec	×	×	×	×
Stop Move, Kill Buffer, Quick Decel	SK	0xE1	×	×	×	×	×
Stop Move, Kill Buffer, Normal Decel	SKD	0xE2	×	×	×	×	×

For more detailed command functions description, please refer to the **Host Command Reference manual.**

2) Digital I/O Function Selection and I/O Status

Character	hex code	
'0'	0x30	Index of encode
'1'	0x31	input 1 or output 1
'2'	0x32	input 2 or output 2
'3'	0x33	input 3 or output 3
'4'	0x34	input 4 or output 4
'L'	0x4C	low state (closed)
'H'	0x48	high state (open)
'R'	0x52	rising edge
'F'	0x46	falling edge

12.8 Modbus/RTU Applications

12.8.1 Position Control

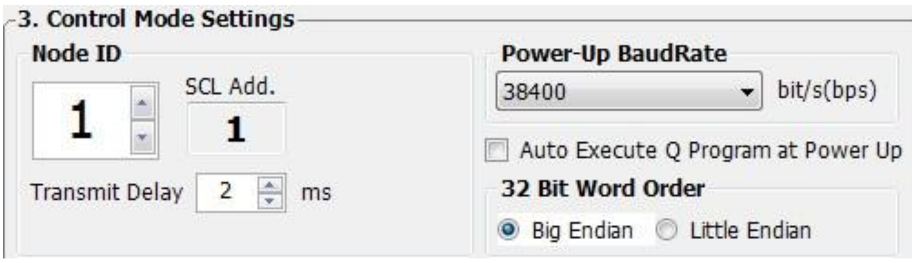
1. Target Profile Planning

SCL command	Target Value	Unit	Register Address	Dec (in Hex)	Description
AC Acceleration	100	rps/s	40028	600(258h)	The unit for register 40028 is 1/6 rps2, when target acceleration is 100rps/s, the value will be 600
DE Deceleration	200	rps/s	40029	1200(4B0h)	The unit for register 40029 is 1/6 rps2, when target acceleration is 200rps/s, the value will be 1200
VE Velocity	10	rps	40030	2400(960h)	The unit for register 40030 is 1/240 rps. When target velocity is 10rps, the value will be 2400.
DI Distance	20000	counts	40031~40032	20000(4E20h)	The target distance will be 20000 counts

2. Drive Setting

Parameter	Function
P-75 (PR) = 5	big-endian data transfer
P-77 (BR) = 3	communication baud rate 38400bps
P-78 (DA) = 1	Communication address 1
P-14 (PM) = 8	power up mode as Modbus/RTU

Use the *SVX Servo Suite* software for configuration:



3. Moving the Motor

Step 1: Set acceleration, deceleration and velocity

Set acceleration register 40028 = 258h, deceleration register 40029 = 4B0h, velocity register 40030 = 960h, and target position 40031~40032 = 4E20h.

Host Sending: 01 10 00 1B 00 05 0A 02 58 04 B0 09 60 00 00 4E 20 24 3B

Rive Respond: 01 10 00 1B 00 05 70 0D

Command Message（Master）				Response Message（slave）		
Function	Data	Number of Bytes		Function	Data	Number of Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	10H	1		Function Code	10H	1
Starting Data Address (Register 40028)	00H(High) 1BH(Low)	2		Starting Data Address (Register 40028)	00H(High) 1BH(Low)	2
Number of Data words	00H(High) 05H(Low)	2		Number of Data words	00H(High) 05H(Low)	2
Number of Data bytes	0AH	1		CRC Check Low	70	1
Content of first Data address 40028	02(High) 58(Low)	2		CRC Check High	0D	1

Content of second Data address 40029	04H(High) B0H(Low)	2		
Content of third Data address 40030	09H(High) 60H(Low)	2		
Content of fourth Data address 40031	00H(High) 00H(Low)	2		
Content of fifth Data address 40032	4EH(High) 20H(Low)	2		
CRC Check Low	24	1		
CRC Check High	3B	1		

Step 2: Send the Point to Point Motion Command

From the SCL Command Encoding Table we find the opcode for a point to point position move (FL), which is 0x66. Writing the value 0x66 to register 40125 will command the drive to execute a point to point (FL) move.

SCL Command Encoding Table							
Function	SCL	Opcode	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Feed to Length	FL	0x66	×	×	×	×	×

Host Sending: 01 06 00 7C 00 66 C8 38

Drive Reply: 01 06 00 7C 00 66 C8 38

Listed As Below:

Command Message（Master）				Response Message（slave）		
Function	Data	Bytes		Function	Data	Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	06H	1		Function Code	06H	1
Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2		Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2
Content of Data	00(High) 66(Low)	2		Content of Data	00(High) 66(Low)	2
CRC Check Low	C8	1		CRC Check Low	C8	1
CRC Check High	38	1		CRC Check High	38	1

12.8.2 Velocity Mode

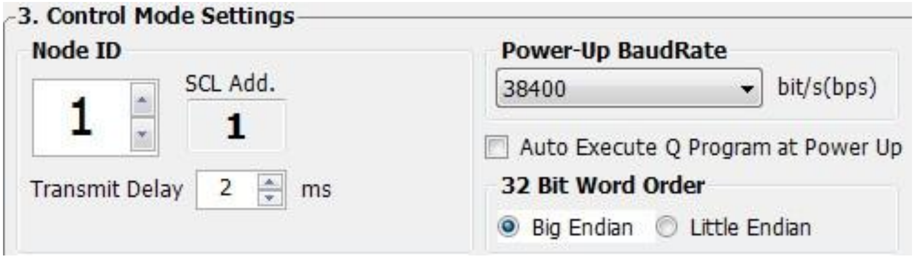
1. Velocity Mode Parameters

SCL Command	Target Value	Unit	Register Address	Write Value Dec(Hex)	Description
JA (Jog Acceleration)	100	rps/s	40047	600(258h)	The unit for register 40028 is rev/sec/sec x 6. Example: when target acceleration is 100 rev/s/s, the value will be 600.
JL (Jog Deceleration)	200	rps/s	40048	1200(4B0h)	The unit for register 40029 is rev/sec/sec x 6. Example: when target deceleration is 200 rev/s/s, the value will be 1200
JS (Jog Speed)	10	rps	40049	2400(960)	The unit for register 40049 is rev/sec x 240. Example: when target velocity is 10 rev/sec, the value will be 2400

2. Drive Setting

Parameter	Function
P-75 (PR) = 5	Big-endian data transfer
P-77 (BR) = 3	communication baud rate 38400bps
P-78 (DA) = 1	Communication address 1
P-14 (PM) = 8	Power up mode as Modbus/RTU

Use *SVX Servo Suite* software for configuration:



Step 1: Set acceleration, deceleration and velocity

Set velocity mode acceleration register 40047 to 258h, deceleration register 40048 to 4B0h, and velocity register 40049 to 960h.

Host Sending: 01 10 00 2E 00 03 06 02 58 04 B0 09 60 A0 9F

Drive Reply: 01 10 00 2E 00 03 E0 01

Command Message（Master）				Response Message（slave）		
Function	Data	Bytes		Function	Data	Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	10H	1		Function Code	10H	1
Starting Data Address (Register 40047)	00H(High) 2EH(Low)	2		Starting Data Address (Register 40047)	00H(High) 2EH(Low)	2
Number of Data words	00H(High) 03H(Low)	2		Number of Data words	00H(High) 03H(Low)	2
Number of Data bytes	06H	1		CRC Check Low	E0	1
Content of first Data address 40047	02(High) 58(Low)	2		CRC Check High	01	1
Content of second Data address 40048	04H(High) B0H(Low)	2				
Content of third Data address 40049	09H(High) 60H(Low)	2				
CRC Check Low	A0	1				
CRC Check High	9F	1				

Step 2: Command for Executing Point to Point Motion

Command Opcode describes register 40125's control code. From the SCL code list shows that for JOG mode, it requires to write data 0x96 to register 40125 to start, and sending 0xD8 to register 40125 to stop.

SCL Command Encoding Table							
Function	SCL	Opcode	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Start Jogging	CJ	0x96	×	×	×	×	×
Stop Jogging	SJ	0xD8	×	×	×	×	×

Start

Host Sending: 01 06 00 7C 00 96 C8 7C

Drive Reply: 01 06 00 7C 00 96 C8 7C

Stop

Host Sending: 01 06 00 7C 00 D8 48 48

Drive Reply: 01 06 00 7C 00 D8 48 48

Start Message

Command Message（Master）				Response Message（slave）		
Function	Data	Bytes		Function	Data	Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	06H	1		Function Code	06H	1
Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2		Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2
Content of Data	00(High) 96(Low)	2		Content of Data	00(High) 96(Low)	2
CRC Check Low	C8	1		CRC Check Low	C8	1
CRC Check High	7C	1		CRC Check High	7C	1

Stop Message

Command Message（Master）				Response Message（slave）		
Function	Data	Bytes		Function	Data	Bytes
Slave Address	01H	1		Slave Address	01H	1
Function Code	06H	1		Function Code	06H	1
Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2		Starting Data Address (Register 40125)	00H(High) 7CH(Low)	2
Content of Data	00(High) D8(Low)	2		Content of Data	00(High) D8(Low)	2
CRC Check Low	48	1		CRC Check Low	48	1
CRC Check High	48	1		CRC Check High	48	1