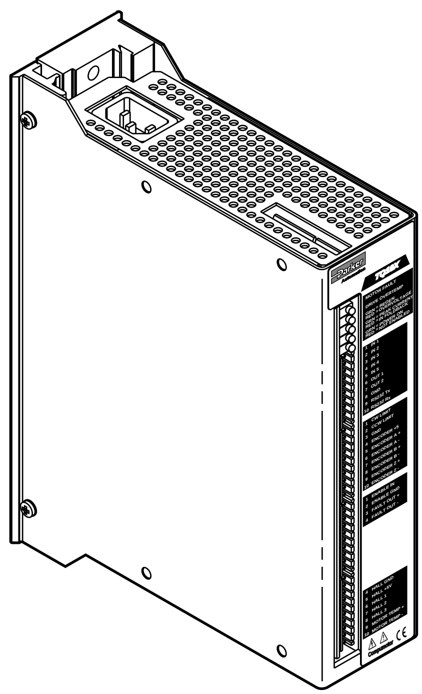
Compumotor

TQ10X

Servo Controller / Drive

User Guide



Compumotor Division
Parker Hannifin Corporation
p/n 88-015174-02 B June 1999

Parker

IMPORTANT

User Information

To ensure that the equipment described in this user guide, as well as all the equipment connected to and used with it, operates satisfactorily and safely, all applicable local and national codes that apply to installing and operating the equipment must be followed. Since codes can vary geographically and can change with time, it is the user's responsibility to identify and comply with the applicable standards and codes. **WARNING:** ***Failure to comply with applicable codes and standards can result in damage to equipment and/or serious injury to personnel.***

Personnel who are to install and operate the equipment should study this user guide and all referenced documentation prior to installation and/or operation of the equipment.

In no event will the provider of the equipment be liable for any incidental, consequential, or special damages of any kind or nature whatsoever, including but not limited to lost profits arising from or in any way connected with the use of this user guide or the equipment.

Technical Assistance ⇨ *Contact your local automation technology center (ATC) or distributor, **or ...***

North America and Asia:
Compumotor Division of Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070
Fax: (707) 584-3793
FaxBack: (800) 936-6939
BBS: (707) 584-4059
e-mail: tech_help@cmotor.com

Europe *(non-German speaking)***:**
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Telephone: 0202-690911
Fax: 0202-600820

Germany, Austria, Switzerland:
HAUSER Elektronik GmbH
Postfach: 77607-1720
Robert-Bosch-Str. 22 • D-77656 Offenburg
Telephone: (0781) 509-0
Fax: (0781) 509-176

Parker
Automation

Compumotor

USER GUIDE CHANGE SUMMARY

TQ10X Servo Controller/Drive User Guide

Revision B
June 1999

The following is a summary of the primary technical changes to this document since the previous version was released. This document, p/n 88-015115-02 **B** , supersedes 88-015174-02 A.

RS485 Communication Option

New – RS485 is now available as a custom option. Contact Custom Products Department for details. This feature is only available on new controllers, identified by firmware 92-016637-01 rev A and higher.

Communication Error Checking (pg 91 an pg 102)

New – SSE and % are two new commands added to support communication error checking. These commands are only available on new controllers, identified by firmware 93-016637-01 rev A and higher.

Updated Speed Torque Curves for SM Motors (pg 40)

Change – Peak motor current values were reduced.

Revision A
October 1997

Motor Pole Compensation Range (pg 11)

Change – The range has been increased by the addition of a third DIP switch on the bottom of the product. This allows improved current-loop performance with a wider range of motor electrical pole frequencies, including all Parker Neometric 70mm and some of the 92mm motors. If the new switch, SW3-3 is placed in the OFF (up) position, settings are backward-compatible with earlier product.

Reset Input Functionality (pg 27)

Change – Asserting the RESET input will now reset controller, as well as drive faults. This is accomplished by cycling +5 volt power to controller. The effect of asserting the RESET input is identical with cycling power, except that power to the encoder and hall sensors is not interrupted by RESET.

Enable Input Low-State Threshold (pg 25)

Change – Threshold has been increased by approximately one volt for improved noise immunity in applications where this signal originates at some distance from the drive.

Technical Information on SM & NeoMetric Series Added (pg 35 – 38)

Addition – Complete technical information is now included on SM and NeoMetric Series motors.

New Back Cover Illustration

Change – The illustration on the back cover of this User Guide has been updated to incorporate the changes listed in this change summary.

Product Type: TQ10X Servo Drive

The above product is in compliance with the requirements of directives

- **72/23/EEC Low Voltage Directive**
- **93/68/EEC CE Marking Directive**

The TQ10X Drive, when installed according to the procedures in the main body of this User Guide, may not necessarily comply with the Low Voltage Directive (LVD) of the European Community. To install the TQ10X Drive so that it complies with LVD, Appendix B, you must follow the additional procedures described in *LVD Installation Instructions*. If you do not follow these instructions, the protection of the product may be impaired.

The TQ10X Series of Drive is sold as a complex component to professional assemblers. As a component, it is not required to be compliant with Electromagnetic Compatibility Directive 89/336/EEC. However, information is offered in Compumotor's *EMC Installation Guide* on how to install this drive in a manner most likely to minimize the effects of drive emissions and to maximize the immunity of drives from externally generated interference.

Table of Contents

CHAPTER ONE

Introduction

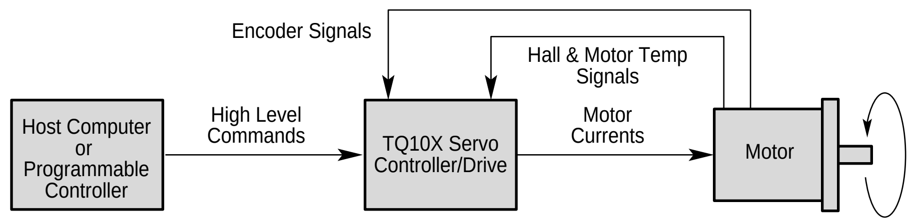
IN THIS CHAPTER

- TQ10X Servo Controller/Drive – Description & Block Diagram

TQ10X Controller/Drive—Description & Block Diagram

The TQ10X Servo Controller/Drive is a servo drive designed to run three phase brushless DC servo motors equipped with Hall effect sensors, using trapazoidal commutation. It can also operate brushed DC servo motors.

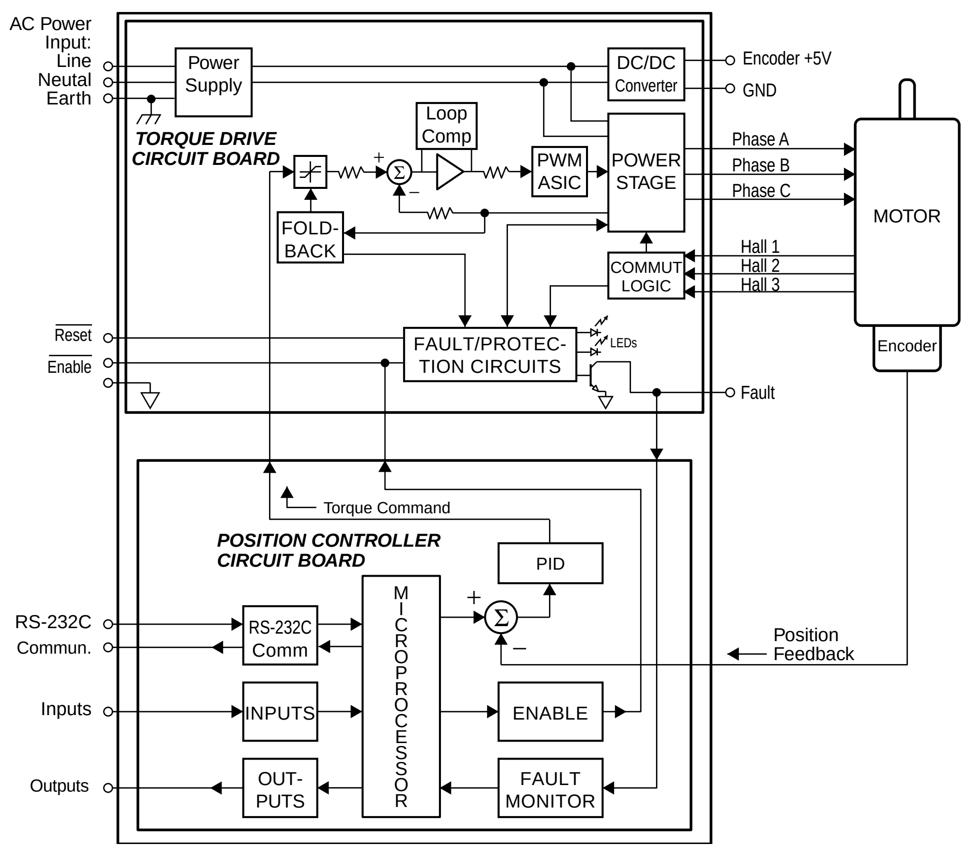
The block diagram for a typical system is shown below.



Block Diagram – System with TQ10X Controller/Drive

The host computer or programmable controller may or may not be necessary, depending upon the application's requirements.

The next drawing shows a block diagram of the drive itself.



Block Diagram – TQ10X Controller/Drive

Inside the drive, there are two circuit boards—a position controller board, and a torque drive board. The position controller receives information about actual position from an encoder.-

Inputs, outputs, and RS-232C communications also enter the position controller board, where they interface with a microprocessor. The microprocessor can enable or disable the torque board. It also generates a position command. The position command is compared with actual position at a summing node; the result passes through a digital-to-analog converter, where it becomes an analog voltage signal.

The voltage signal goes through a PID position control loop, and then on to the torque board.

The analog voltage is a *torque command* that represents commanded current. It can range from -10VDC to +10VDC. For each volt of command input, the TQ10X will produce 1.0 A of output current to the motor.

On the torque board, the torque command goes into one of the inputs of a summing node. A feedback signal representing actual motor current goes into the other input. When actual current is subtracted from commanded current at the summing node, the difference is *current error*.

The resulting error signal goes through an error amplifier whose output controls a pulse width modulation (PWM) circuit. If actual current is too low, the PWM circuit will send longer pulses to the drive's power stage. These pulses keep the stage turned on longer, which results in more motor current. If actual current is too high, the PWM circuit sends shorter pulses, resulting in less motor current.

Other Features

Dip Switches

The TQ10X Drive has 12 DIP switches on the top and three DIP switches on the bottom. You can set these switches to configure the drive for your particular application.

Inputs and Outputs

All input and output signal connections are made on the front panel of the drive, through removable screw terminal connectors.

Cooling Options

The drive has a heatplate design. You can mount the drive without a heatsink, if your mounting surface provides sufficient heatsinking capability.

An external heatsink/fan unit is available from Compumotor as an option.

The drive is open on the top and bottom. You can purchase optional covers from Compumotor. If you install the covers, you may also need to install the heatsink/ fan unit, to help keep the drive within its temperature limits.

You can purchase a TQ10X drive from Compumotor that has the heatsink/fan unit and the covers installed at the factory. The part number for this drive is TQ10X-EHS. (The suffix -EHS is an acronym for *Enclosure/Heat Sink*.)

Compumotor SM and NeoMetric Series Brushless Servo Motors

Compumotor manufactures SM and NeoMetric Series servo motors; you can use these motors with the TQ10X Drive. Each motor is equipped with Hall effect sensors, an encoder, and a thermostat. The motor cables are color coded, which makes connecting the motor to the drive a straightforward procedure.

Compumotor Family of Products

The TQ10X Drive is completely compatible with Compumotor's broad range of single-axis and multi-axis motion control products.

CHAPTER TWO

Installation

IN THIS CHAPTER

- Product Ship Kit List
- Installation Procedure

What You Should have *(ship kit)*

If you ordered a TQ10X, you should have

Part	**Part Number**
TQ10X Drive	TQ10X
includes 10-pin plug (four included)	43-013891-01
includes 7-pin plug (one included)	43-013575-01
Power Cable – 6 feet (1.8 m) in length	44-000054-01
Thermally Conductive Strip	58-014871-01
TQ10 Torque Servo Drive User Guide	88-015115-02

Accessories

External Heatsink/Fan Unit	TQ-HS3
Top & Bottom Covers (to enclose drive)	TQ-ENCL

NOTE: If you ordered a TQ10X-EHS, the External Fan/Heatsink Unit and the Top & Bottom Covers are installed at the factory.

SM Motor Information

SM with 500 Line Encoder	SM with 1000 Line Encoder	SM Motor Cables
SM161AD-N10N	SM161AE-N10N	23TQ Cable-10
SM162AD-N10N	SM162AE-N10N	(1 set/10’ cables)
SM161AD-NTQN	SM161AE-NTQN	23TQ Cable-25
SM162AD-NTQN	SM162AE-NTQN	(1 set/25’ cables)
SM231AD-NTQN	SM231AE-NTQN	Above cables are for
SM232AD-NTQN	SM232AE-NTQN	Size 16 and 23 motors
SM232BD-NTQN	SM232BE-NTQN	
SM233AD-NTQN	SM233AE-NTQN	
SM233BD-NTQN	SM233BE-NTQN	

NeoMetric Motor Information

NeoMetric with 500 Line Encoder	NeoMetric with 1000 Line Encoder	NeoMetric Motor Cables
N0701DD-NTQN	N0701DE-NTQN	70TQ Cable-10
N0701FD-NTQN	N0701FE-NTQN	(1 set/10’ cables)
N0702ED-NTQN	N0702EE-NTQN	70TQ Cable-25
N0702FD-NTQN	N0702FE-NTQN	(1 set/25’ cables)
N0703FD-NTQN	N0703FE-NTQN	92MS Cable-10
N0703GD-NTQN	N0703GE-NTQN	(1 set/10’ cables)
N0704FD-NTQN	N0704FE-NTQN	92MS Cable-25
N0704GD-NTQN	N0704GE-NTQN	(1 set/25’ cables)
	N0921FE-NMSN	
	N0921GE-NMSN	
	N0922GE-NMSN	

Precautions

The TQ10X Drive has an open-frame style of construction. The top and bottom of the sheet metal enclosure is open, and internal components are exposed. Hazardous voltages are present inside the drive and on many of its terminals. Therefore, observe the following precautions:

- Do not reach inside the drive
- Do not probe inside the drive
- Do not touch the drive's motor terminals while power is applied to the drive
- Do not touch **MOTOR TEMP+** or **MOTOR TEMP-** while power is applied to the drive
- Configure the drive's DIP switches for your application before you apply power to the drive

Installation Overview

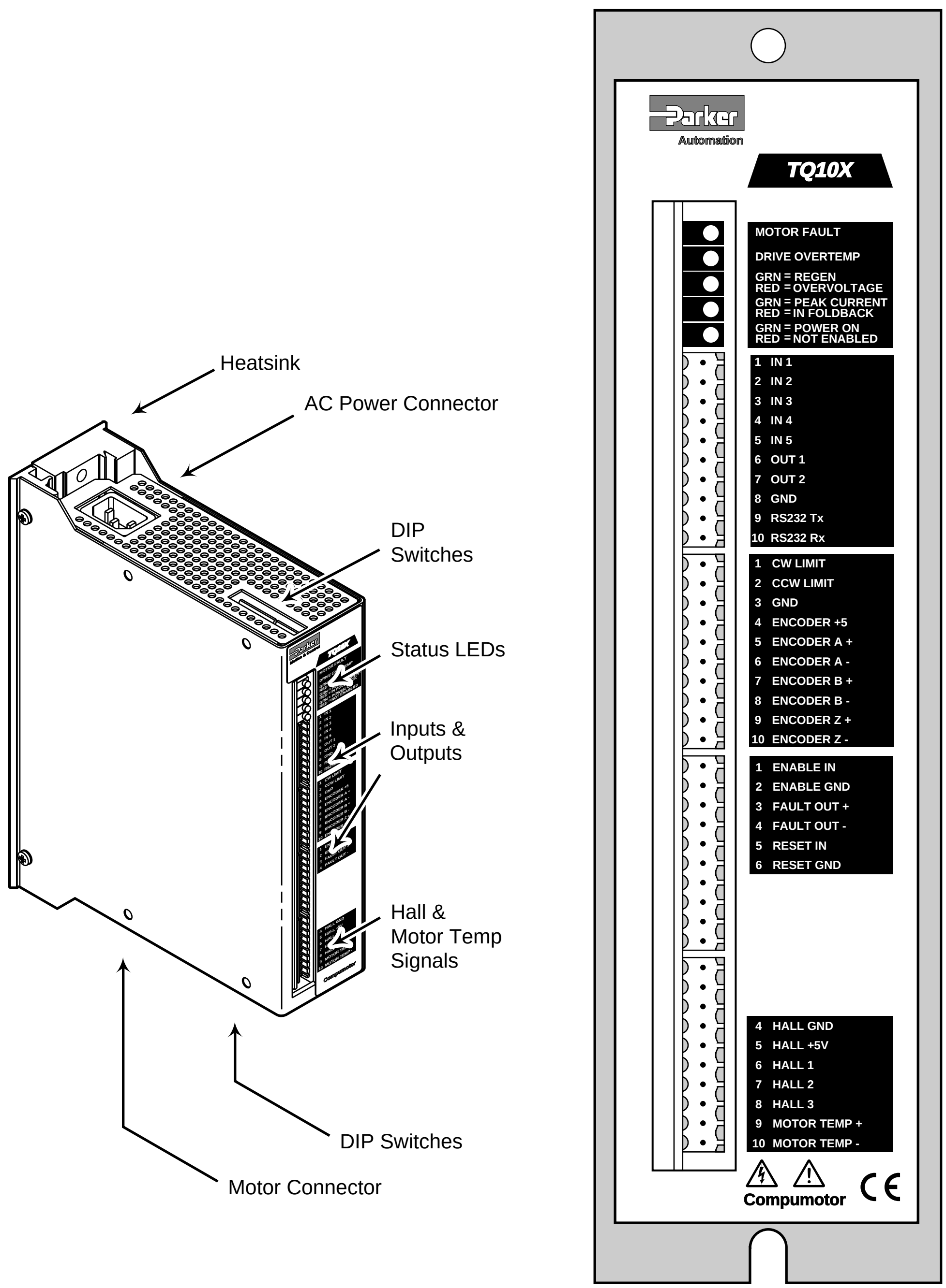
The order of topics in the installation procedure is:

- Quick Test
- DIP Switch Configuration
- Drive Mounting & Heatsinking
- Motor Mounting
- Connecting the Motor to the Drive
- Connecting AC Power
- Testing the System Installation
- Connecting the Motor to the Load

Installation Procedure

Topics in this chapter are arranged to lead you through the installation process in a step-by-step manner. Complete each step before proceeding to the next.

The next drawing shows locations and names of the various connectors, switches, and components that you will encounter during the installation procedure.



TQ10X Servo Controller/Drive

TQ10X – Component Locations

Quick Test

The steps in the following installation procedure will lead you through the permanent installation process. However, if you wish to familiarize yourself with the drive before you install it, you can perform a bench top *quick test*. To do so, complete the following sections, in the order given below:

Set configuration DIP switches (on top of drive)
Set peak current at twice the motor's continuous rated current, or less
Set compensation DIP switches (on bottom of drive)
Connect RS-232C signals
Connect the Motor to the Drive
Connect AC Power
Test your system

During your permanent installation, complete the other sections in this chapter—drive mounting, motor mounting, and connecting the load—and review *Chapter 4 Tuning*.

Installation

The following procedures will lead you through the steps required to permanently install your TQ10X Drive and motor.

1. Set DIP Switches (Top of Drive)

Configure the TQ10X Drive's DIP switches for your motor and application. Two 6-position DIP switches—Switch 1 (SW1) and Switch 2 (SW2)—are located on top of the drive. The table below summarizes their settings. A three-position DIP switch—Switch 3 (SW3)—is located on the bottom of the drive. See the section after this for instructions on setting SW3.

Default Settings

The factory default position is *off* for all switches. You must set the DIP switches appropriately for your application.

Peak Current

Set DIP switches SW1-#1 — SW1-#3 for the *peak current* that you want your drive to produce. If you use a high performance motor (peak current rating greater than three times the continuous current rating), such as Compumotor's SM or NeoMetric Series servo motors, see the caution note below.

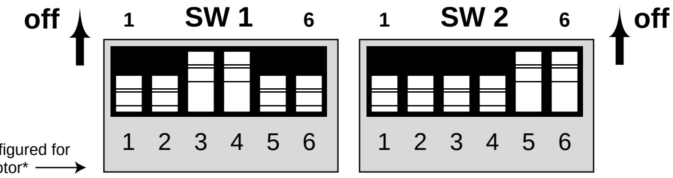
CAUTION

Peak current settings for initial drive tuning with SM and NeoMetric Motors: Set peak current at twice the motor's continuous rated current, or less. Otherwise, motor damage due to excessive heating may result from high peak currents and improper tuning values. See *Chapter 4 ,Tuning* for a procedure to iteratively raise peak current during tuning.

Time at Peak Current

Set DIP switches SW1-#4 — SW1-#6 to control the length of time the drive can produce peak current, before it goes into current foldback.

TQ10X DIP SWITCH SETTINGS



Shown Configured for SM161A Motor*

PEAK CURRENT

(amps)	1	2	3
0	off	off	off
1.5	on	off	off
3.0	off	on	off
4.4	on	on	off
6.0	off	off	on
7.4	on	off	on
8.9	off	on	on
10.0	on	on	on

TIME AT PEAK

(seconds)	4	5	6
1.0	on	on	on
1.2	on	on	off
1.4	on	off	on
1.6	on	off	off
3.3	off	on	on
5.0	off	on	off
10.0	off	off	on

LOOP GAIN

4	5	6	setting number
off	off	off	0
on	off	off	1
off	on	off	2
on	on	off	3
off	off	on	4
on	off	on	5
off	on	on	6
on	on	on	7

Less Gain (use with lower inductance motors) ↑

More Gain (use with higher inductance motors) ↓

FOLDBACK

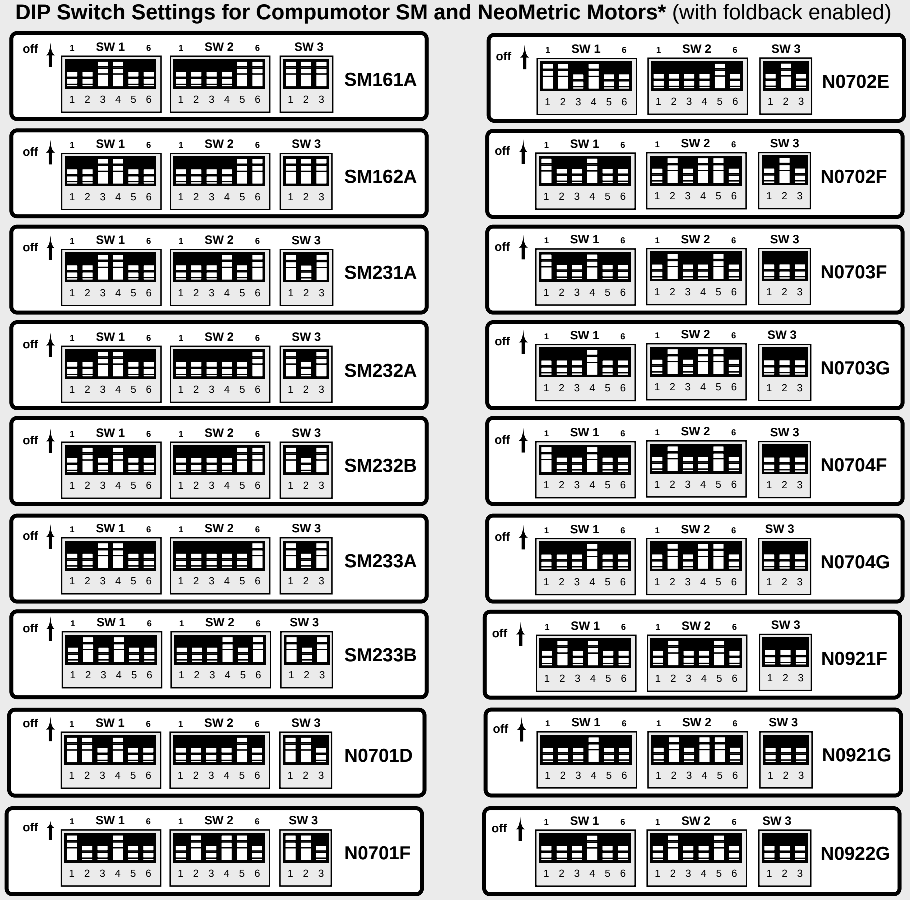
3	
off	Foldback Disabled
on	Foldback Enabled

2	
off	High Threshold
on	Low Threshold

FOLDBACK FAULT

1	
on	Fault on Foldback
off	No Fault on Foldback

DIP Switch Settings for Compumotor SM and NeoMetric Motors* (with foldback enabled)



*Switches shown configured for initial tuning, with peak current approx. twice motor's continous current rating. See Tuning for procedure to raise current. For SW 3 settings, see next page.

Foldback Fault Enable

If DIP switch SW2–#1 is in the *off* position, the drive can go in and out of current foldback without causing a fault condition. If DIP switch SW2-#1 is in the *on* position, then going into foldback will cause a latched fault condition; this setting can aid in troubleshooting your system.

Foldback Threshold

DIP switch SW2-2 sets the current threshold for the foldback circuit. During initial tuning, this switch should be ON (low threshold) for all SM motors, and for NeoMetric motors with "D" and "E" windings. After tuning is completed, this switch should be ON for "A" winding SM motors. It should be OFF (high threshold) for "B" winding motors, NeoMetric (70mm and 92mm) motors, and for most motors with continuous current ratings above 4 amps.

Foldback Enable

DIP switch SW2-3 enables foldback. When this switch is ON (the default) foldback is enabled.

Loop Gain

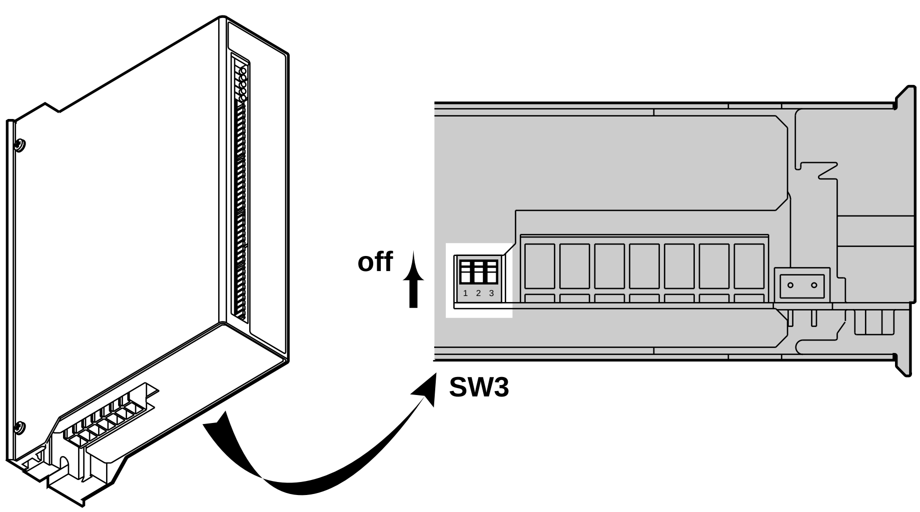
Set DIP switches SW2-#4 — SW2-#6 to control the gain of the drive's internal current control loop. Properly setting this switch will match the drive to your motor's parameters (inductance and resistance). The suggested settings provide wide current-loop bandwidth. In some applications, lower bandwidth may allow easier position-loop tuning. To reduce bandwidth, set the loop gain DIP switch for lower gain by one or two settings. *Do no set loop gain higher than the suggested setting.*

DIP Switch Settings for SM and NeoMetric Motors

DIP switch settings for Compumotor SM and NeoMetric motors are shown above. If you use a non-Compumotor motor, see the *Appendix* at the end of this user guide for information about setting DIP switches for your motor.

2. Set DIP Switches (Bottom of Drive)

Switch 3 (SW3) is a three-position DIP switch located on the bottom of the drive, near the motor connector.

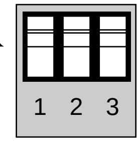


DIP Switch 3 – Location

Set it to control the drive's motor pole compensation, based upon your motor's electrical time constant. The following table shows switch settings for Compumotor SM and NeoMetric Series servo motors.

TQ10X DIP SWITCH #3*

*Located on Bottom of Drive



MOTOR POLE COMPENSATION

	1	2	3
SM161A, SM162A	off	off	off
Reserved	on	off	off
SM231A, SM232A+B, SM233A+B	off	on	off
Reserved	on	on	off
N0701D, N0701F	off	off	on
N0702E, N0702F	on	off	on
Reserved	off	on	on
N0703F, N0703G, N0704F, N0704G, N0921F, N0921G, N0922G	on	on	on

If you use a non-Compumotor motor, see the *Appendix* at the end of this user guide for information about setting DIP switches for your motor.

Offset Potentiometer – Do Not Adjust

Located next to DIP SW3 is a small potentiometer that controls the drive's offset. It was adjusted at the factory, and requires no further adjustment.

WARNING

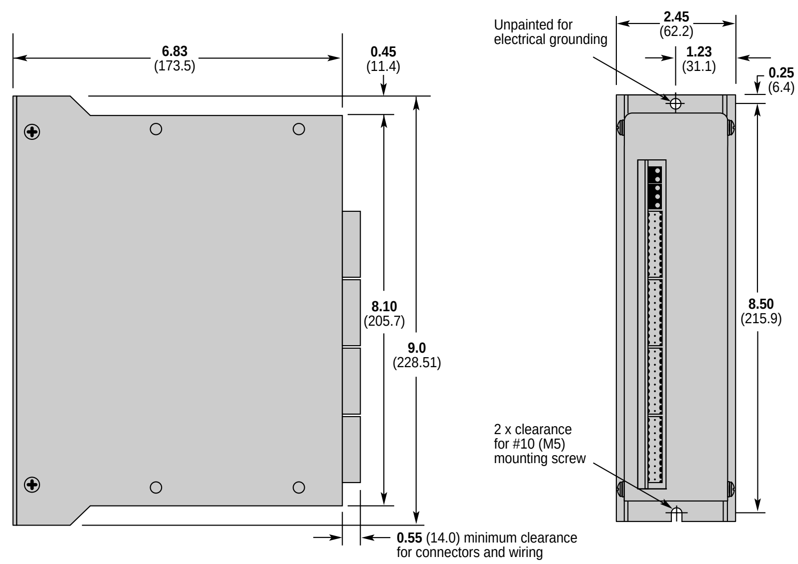
Do not adjust the offset potentiometer. Lethal voltages are present inside the drive. Adjusting the potentiometer with AC power applied can be hazardous to personnel.

3. Mount the Drive

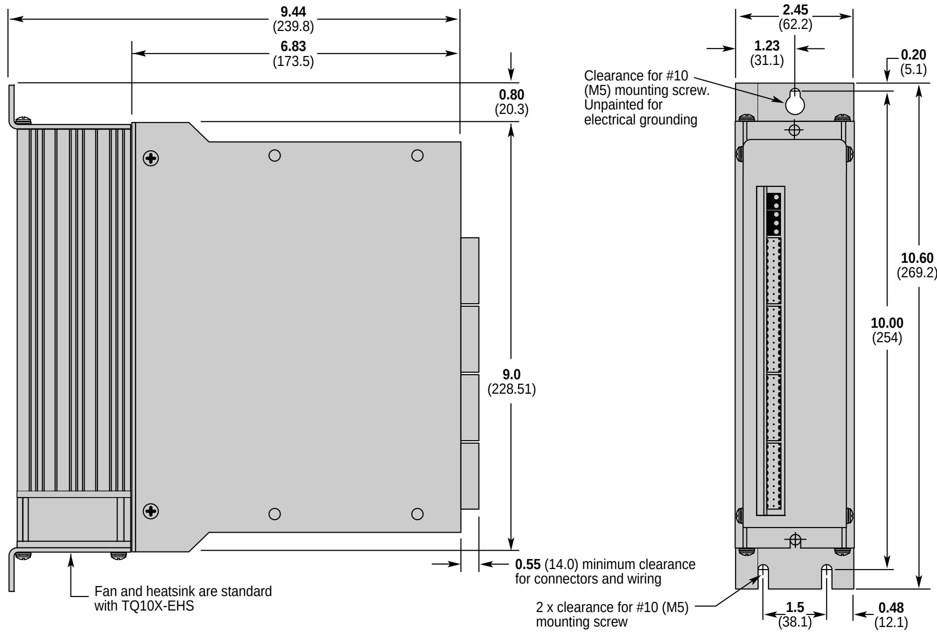
The TQ10X has an "open frame" enclosure—sheet metal encloses the front and sides, but the top and bottom are open. A TQ10X-EHS drive has top and bottom covers, and an attached heatsink and fan. You can purchase the heatsink or covers separately, and install them on your drive.

Dimensions for each version of the drive are shown below.

Drive Dimensions



Dimensions — TQ10X Drive



Dimensions — TQ10X With Heatsink Attached

Environmental Considerations

TEMPERATURE SPECIFICATIONS

Maximum Ambient Temperature:	50°C	(122°F)	
Minimum Ambient Temperature:	0°C	(32°F)	
Maximum Temperature of Mounting Surface:	45°C	(113°F)	(for non -EHS version)

HUMIDITY Keep the relative humidity below 95%, non-condensing.

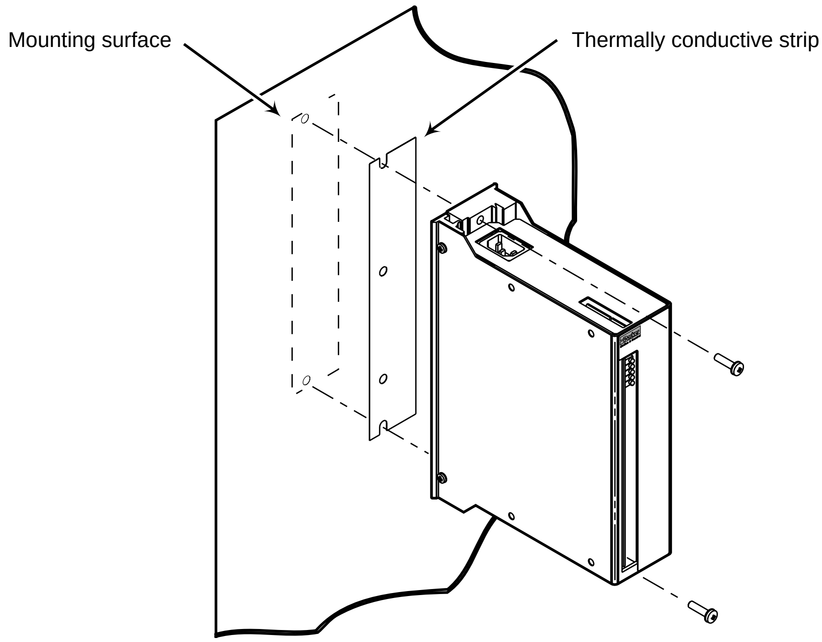
LIQUIDS Do not allow liquids or fluids to come into contact with the TQ10X Drive or its cables.

AIRBORNE CONTAMINANTS Particulate contaminants, especially electrically conductive material such as metal shavings or grinding dust, can damage the TQ10X Drive and motor. Do not allow contaminants to come into contact with the drive or motor.

Mounting the Drive to a Heat Sinking Surface

Move profiles and loads affect the amount of heat dissipated by the TQ10X Drive. If yours is a low power application with moderate ambient temperature, the drive may not need a large heatsink. The mounting surface may be adequate as a heatsink, provided it has sufficient mass and surface area.

The mounting plate at the rear of the drive is a *heatplate*—it is a thermal pathway through which the drive can dissipate its excess heat. Mount the drive to a suitable heat sinking surface. A thermally conductive strip is provided with the drive. When you mount the drive, install the strip between the drive's heatplate and the mounting surface, as the next drawing shows. Do not fold or wrinkle the strip.



Mounting with Thermally Conductive Strip

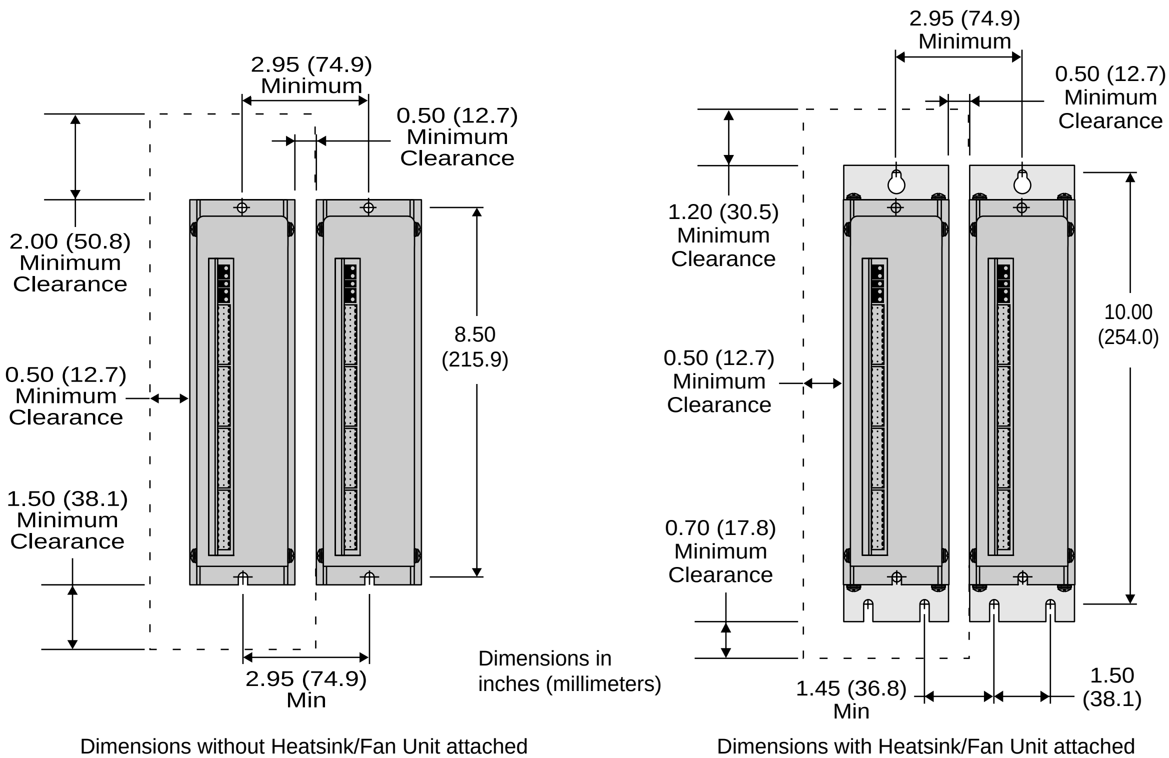
While the drive is operating, ensure that the temperature of the mounting surface is no higher than 45°C (113°F), and that the temperature of the ambient air is no higher than 50°C (122°F).

Do not mount equipment that produces substantial heat below the drive. Avoid mounting heat sensitive equipment directly above the drive.

Panel Layout

High power applications may require a heatsink. A heatsink/fan unit for the TQ10X Drive is available from Compumotor (part number TQ-HS3). If you purchased a TQ10X-EHS, the heatsink/fan unit was installed at the factory.

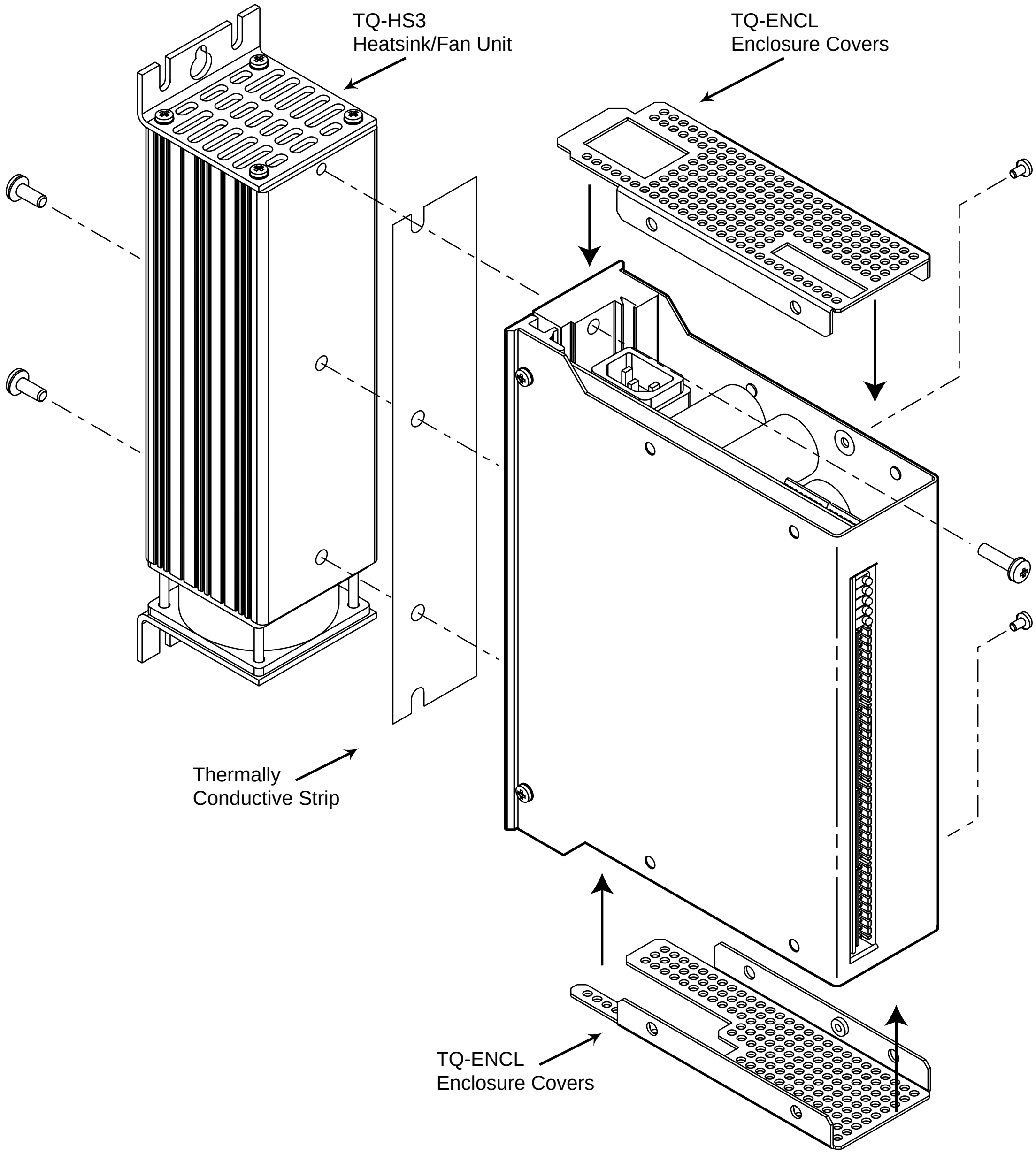
The next drawing shows minimum spacing and clearance requirements you should follow when you mount TQ10X Drives. Dimensions are shown with and without heatsink/fan units attached to the drive.



Panel Layout Dimensions

Attaching Heatsink/Fan Unit and Enclosure Covers

The next drawing shows how to attach Compumotor's optional TQ-HS3 Heatsink/ Fan Unit to your TQ10X Drive.

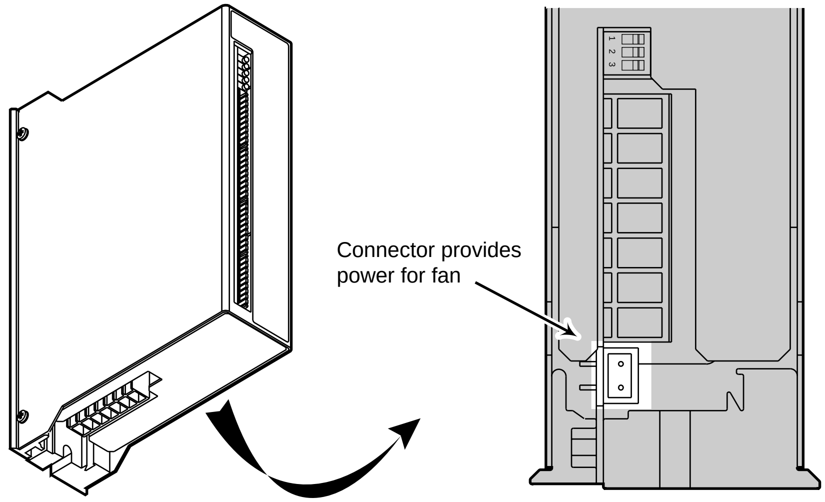


Heatsink/Fan Unit and Covers – Attaching to Drive

Perforated sheet metal covers that enclose the top and bottom of the drive are available from Compumotor (part number TQ-ENCL). The drawing above shows how to attach these optional parts to your TQ10X Drive.

Because the covers will reduce convection available for cooling internal components, drive temperature may increase. You may need to reduce ambient temperature or provide forced air cooling to cool the drive. You may also need to install an external regeneration resistor (power dump), if regenerated energy causes drive overheating with the covers installed.

The heatsink/fan unit has a cable with connector attached. Plug the connector into the TQ10X's fan power connector.

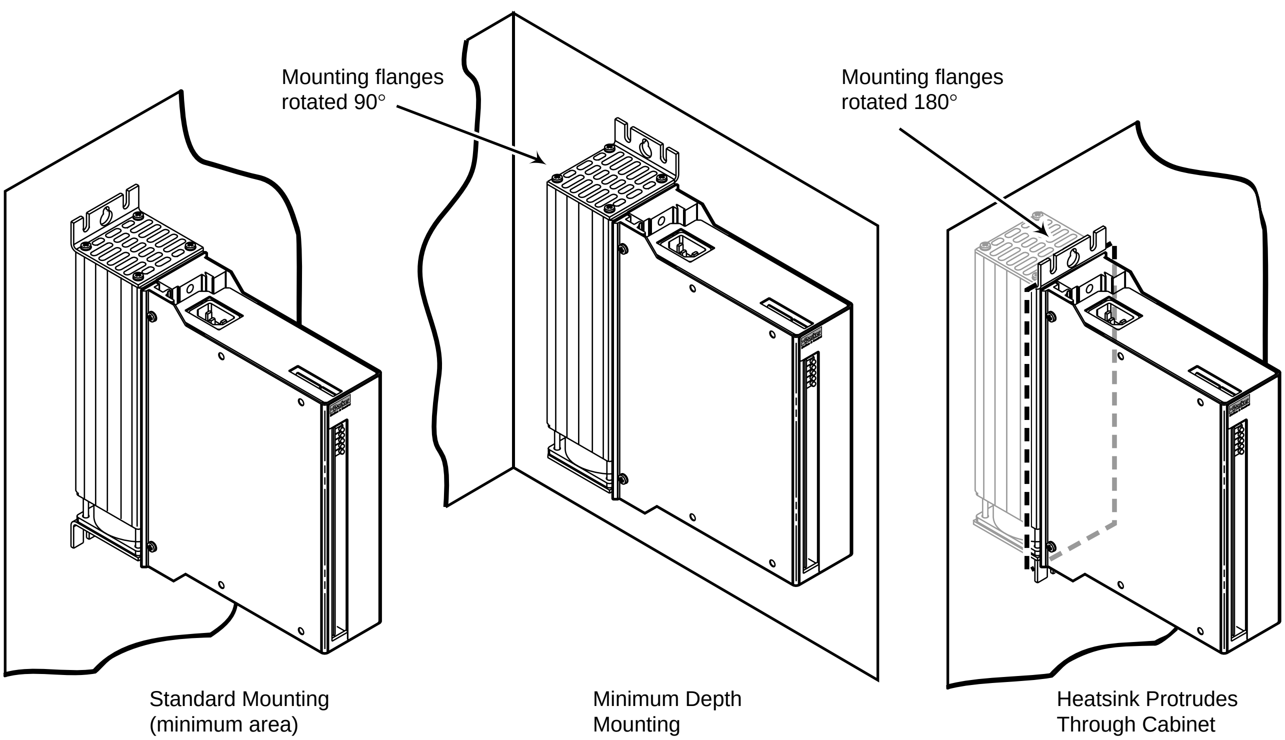


Fan Power Connector

Mounting Options with Heatsink/Fan Unit

If your drive has a TQ-HS3 Heatsink/Fan Unit attached, you have several options for mounting the drive, as the next drawing shows.

You can mount the drive in a "minimum area" configuration, as shown on the left, below. Or, you can mount the drive in a "minimum depth" configuration by rotating the mounting flanges 90° (located at the top and bottom of the heatsink), as shown in the center. If you rotate the mounting flanges 180°, as shown on the right, you can mount the drive with the heatsink protruding through an opening in your equipment cabinet. This configuration may help to remove heat from the interior of your cabinet.



Mounting Options

4. Mount the Motor

The following guidelines present important points about motor mounting and its effect on performance.

For mechanical drawings of SM and NeoMetric Series servo motors, see *Chapter ③—Specifications.*

WARNING
Improper motor mounting can limit system performance and jeopardize safety of personnel.

Servo motors should be mounted by bolting the motor's face flange to a suitable support. Foot mount or cradle configurations are not recommended because the motor's torque is not evenly distributed around the motor case. Any radial load on the motor shaft is multiplied by a much longer lever arm when a foot mount is used rather than a face flange.

Servo motors used with the TQ10X can produce large torques and high accelerations. These forces can shear shafts and mounting hardware if the mounting is not adequate. High accelerations can produce shocks and vibrations that require much heavier hardware than would be expected for static loads of the same magnitude.

Under certain move profiles, the motor can produce low-frequency vibrations in the mounting structure. If harmonic resonances are induced by the move profiles you are using, these vibrations can cause metal fatigue in structural members. A mechanical engineer should check the machine design to ensure that the mounting structure is adequate.

CAUTION
Modifying or machining the motor shaft will void the motor warranty. Call Compumotor's Application Engineers (800-358-9070) about shaft modifications as a custom product.

Motor Heatsinking

Performance of a servo motor is limited by the amount of current that can flow in the motor's coils without causing the motor to overheat. Most of the heat in a brushless servo motor is dissipated in the stator—the outer shell of the motor. Performance specifications usually state the maximum allowable winding or case temperature. Exceeding this temperature can permanently damage the motor. The maximum case temperature for Compumotor SM and NeoMetric motors is 70°C (158°F).

If yours is a demanding application, your motor may become quite hot. The primary pathway through which you can remove the heat is through the motor's mounting flange. Therefore, mount the motor with its flange in contact with a suitable heatsink.

CAUTION
In temporary "bench top" setups, often used for prototyping or demonstrations, motors are very vulnerable to overheating if they are not mounted to a heatsink. Limit peak current to the motor's continuous current rating if you operate your motor without a heatsink.

Specifications for Compumotor SM and NeoMetric Series servo motors apply when the motor is mounted to a ten inch by ten inch aluminum plate, 1⁄4 inch thick. To get rated performance in your application, you must mount the motor to a heatsink of at least the same thermal capability. Mounting the motor to a smaller heatsink may result in decreased performance and a shorter service life. Conversely, mounting the motor to a larger heatsink can result in enhanced performance.

5. Connect the Motor to the Drive

The TQ10X Drive works with three-phase brushless motors equipped with Hall effect sensors. The typical motor has a permanent-magnet rotor with four poles (two pole pairs). Higher pole-count motors may also be used.

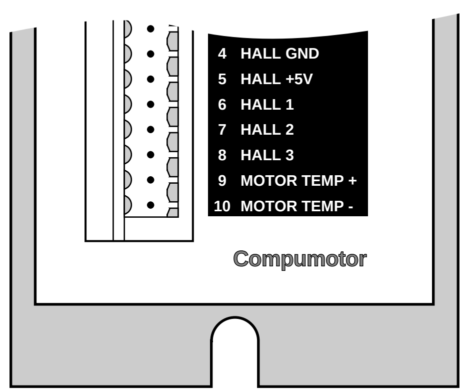
In the following installation procedure, we assume you are using a Compumotor SM or NeoMetric Series Motor with your TQ10X Drive. If you are using a non-Compumotor motor, consult the *Appendix* at the end of this user guide for information you may need during the following installation steps.

Connect Hall Effect Wires

The TQ10X Drive is designed to be used with motors that have single-ended, open collector Hall outputs. Internally, the drive pulls these signals up to +15V.

For best performance and reliability, the drive should be used with motors having no more than six electrical degrees of commutation error in either direction. Motors with greater commutation error may cause increased torque ripple and motor heating, reduced average torque, and greater stresses on the drive output stage.

Connect your motor's Hall effect wires to the 10-pin screw terminal on the front of the TQ10X. Each terminal is labeled with the name of the wire you should connect.



Hall Effect and Motor Temperature Connections – Front of Drive

14 AWG (2.5 mm^2) is the maximum wire size that can fit in the connector.

Connect Motor Thermostat Wires

Connect your motor's thermostat wires to **MOTOR TEMP+** and **MOTOR TEMP-**.

If your motor does not have a thermostat, short **MOTOR TEMP+** and **MOTOR TEMP-** together by connecting an insulated jumper wire between them. The drive will experience a motor fault if neither a thermostat nor a jumper wire is attached to the **MOTOR TEMP** terminals.

WARNING

Hazardous voltages are present on **MOTOR TEMP+** and **MOTOR TEMP–** when the drive is powered up. Use insulated wires for connections. Protect personnel from contacting these terminals or any attached wires. Do not short these terminals to earth ground.

The TQ10X’s motor overtemperature fault can, in many cases, protect the motor against overheating. Through its **MOTOR TEMP+** and **MOTOR TEMP-** terminals, the drive checks for electrical continuity provided by a normally-closed thermostat mounted on the motor. If the motor overheats and the thermostat opens, the loss of continuity triggers protection circuitry in the TQ10X. It will turn off power output to the motor, and illuminate the LED labeled **MOTOR FAULT**.

This is a latched fault. Wait for the motor to cool, then cycle power to resume operations. A motor overtemperature fault indicates improper motor sizing, or improper installation of your application.

This circuit may not protect the motor in every possible application. It works best in cases where the temperature rise occurs slowly over a long period of time. In this situation, the thermal sensor and motor windings will be at the same temperature. When the windings and sensor reach the sensor’s threshold temperature, the sensor can trigger the overtemperature circuit.

In cases where the temperature rise is caused by continuous peak current flowing—such as a mechanical jam— the winding temperature may rise much more quickly than the sensor temperature does. In this situation, the windings may be damaged from overheating *before* the sensor can trigger the overtemperature circuit.

Be careful not to overheat your motor during system tuning. Allow the motor to cool for several seconds between test moves. Instability while tuning can cause rapid motor heating and possible motor damage. A thermal switch may not protect the motor if the motor heats up too quickly. Do not allow oscillations to persist longer than a few seconds. Allow the motor to cool for several minutes after each incident of instability.

Connect Motor Phase Wires

Connect your motor’s phase wires and ground wire to the removable 7-pin **MOTOR** connector located on the bottom of the drive. The next drawing shows the location of each terminal.

WARNING

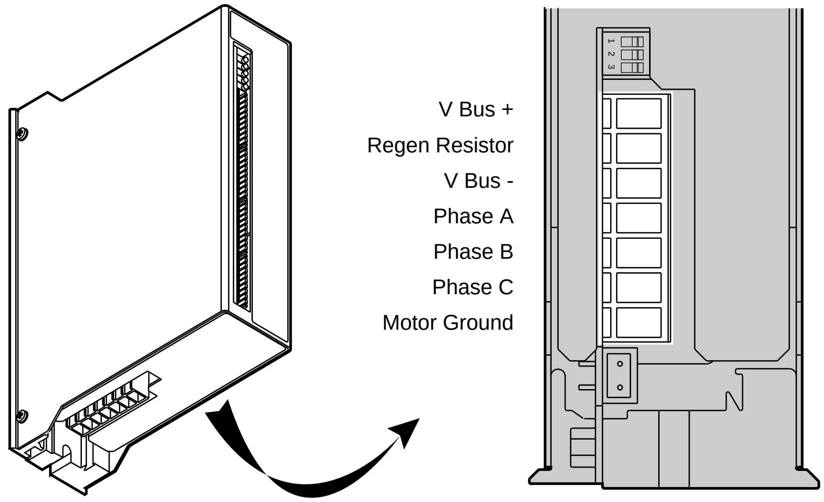
Potentially hazardous voltages are present on motor connector terminals when power is applied to the drive.

Ensure the connector is aligned correctly when inserted—and not misaligned by one position.

The terminal labeled **MOTOR GROUND** is connected internally to the **EARTH** terminal on the drive’s AC power connector.

WARNING

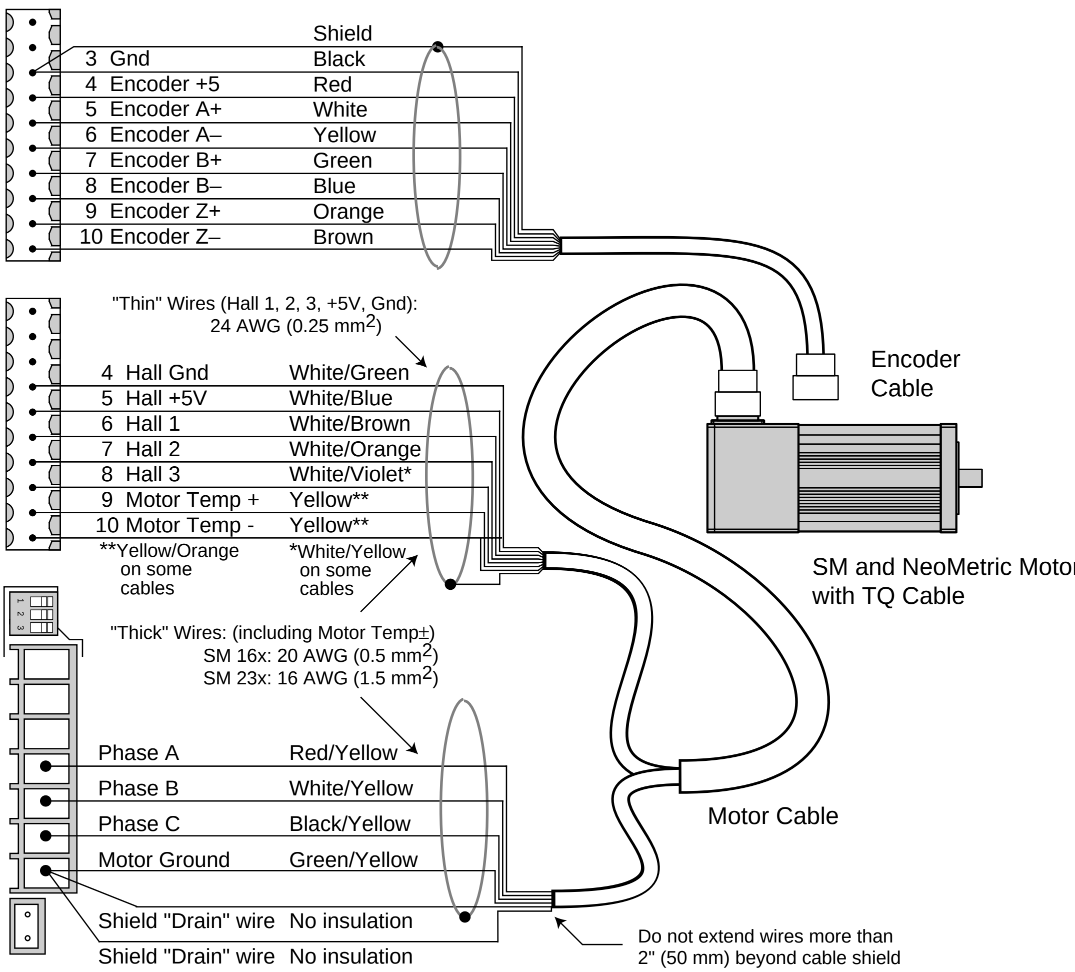
DO NOT OMIT the **Motor Ground** connection. Internal failure of motor insulation can place the motor frame at deadly potential if it is not properly grounded.



Motor Connector

Connecting Compumotor SM or NeoMetric Series Motors

To connect a Compumotor SM or NeoMetric Series servo motor to the TQ10X, follow the color code shown in the next drawing.



Motor Cable Connections for SM or NeoMetric Series Motors

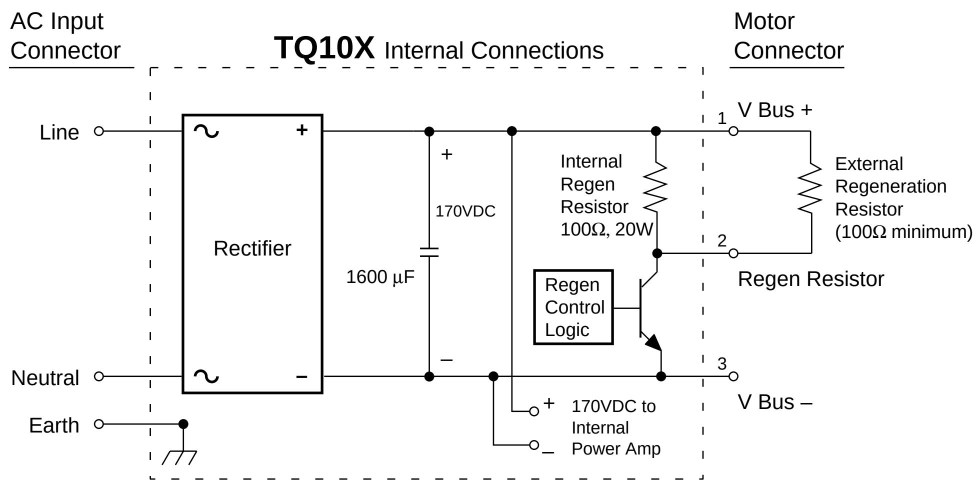
Inside the motor cable, there are two sets of wires. One contains Hall effect and motor thermostat wires; the other contains motor phase wires. Each set of wires has its own shield and shield drain wire. As shown in the drawing, you should connect both drain wires and the green/yellow ground wire to the **MOTOR GROUND** connector.

Optional – Connect an External Regeneration Resistor

The TQ10X Drive can dissipate regenerated energy in its internal *regeneration resistor*. If your system regenerates more energy than the internal resistor can dissipate, you can connect an external resistor between two terminals called V Bus+ and Regen Resistor, located on the motor connector. The external resistor doubles the TQ10X's dissipation capabilities.

WARNING
Potentially hazardous voltages are present on **V Bus+** and **Regen Resistor** terminals when power is applied to the drive.

The next drawing shows the internal regeneration resistor, terminals for an external regeneration resistor, and the DC power bus.



Regeneration Circuit

The TQ10X's regeneration circuit works automatically—there are no adjustments to make. The circuit monitors the voltage on the power bus. If regenerated energy from the motor causes the bus voltage to rise above a threshold value, the circuit closes a switch, thus connecting the regeneration resistor between the positive and negative sides of the power bus, **V Bus+** and **V Bus–**. The energy is then dissipated in the resistor—its power capacity is 1KW for one second, or 10 watts on a continuous basis. During the regeneration event, the bicolor LED labeled **REGEN/ OVERVOLTAGE** will be illuminated *green*.

The TQ10X also has an *overvoltage* circuit; it protects the drive from excessive regeneration. If the motor regenerates more energy than the internal resistor can dissipate, voltage on the power bus will rise and trigger an overvoltage fault. The drive will shut down power output to the motor. The bicolor LED labeled **REGEN/ OVERVOLTAGE** will be illuminated *red*. This is a latched fault—cycle power to resume operations.

If excessive regeneration causes overvoltage faults in your system, you can install an external regeneration resistor. Ensure that the external resistor is adequately mounted and cooled. Excessive heating can cause the resistor to fail.

CAUTION
Adequately cool the external resistor. Forced air cooling may be required. Maintain resistor temperature below its rated temperature limit.

The internal resistor is a 100Ω, 10% non-inductive resistor. It is intended to dissipate no more than 10 watts. (For safety margins, a 20W resistor was designed into the circuit.) For an external resistor, we suggest:

- Manufacturer Name: Dale
- Manufacturer Part Number: NHL-55-16N-100Ω (5% – 20% is suitable)

This resistor has "faston" mounting tabs, and can be mounted with two screws. You can order this resistor, with two 18 inch (457 mm) cables, as a kit from Compumotor. The part number is:

TQ-REGEN-KIT

Or, you may use an equivalently rated 100Ω 55W *non-inductive* resistor for your external resistor.

To connect the external resistor, follow these steps:

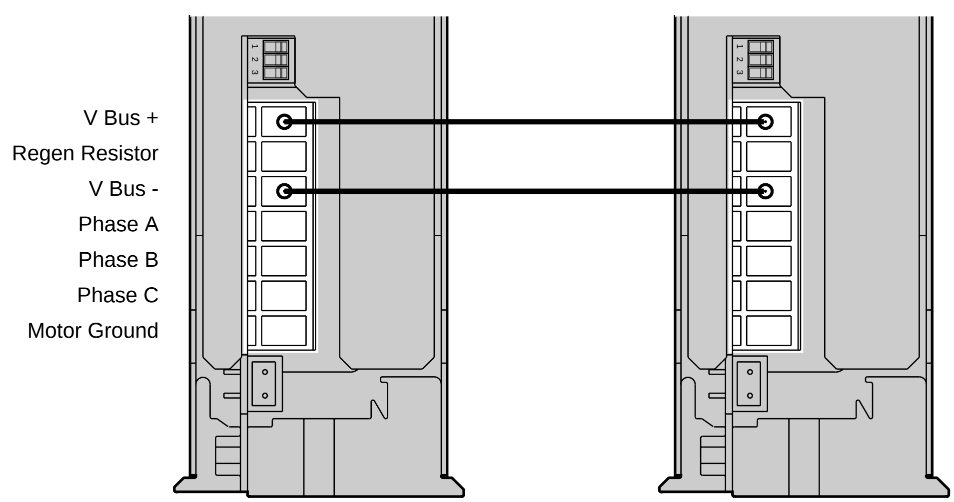
① Connect the resistor's two terminals to **V BUS+** and **REGEN RESISTOR**, located on the motor connector.
② Keep wires as short as possible, and twist them together.

The circuit will automatically dissipate half the regenerated energy in the external resistor. Total dissipation with the external resistor installed is 2KW peak, 20W continuous.

Optional – Sharing the Power Bus

In some applications with multiple drives, one or more drives may continuously receive regenerated power from their loads. For example, in a tensioning application, two TQ10Xs apply tension (opposite torques) to a single moving load. In this situation, one TQ10X could receive substantial regenerated power from its motor.

In such applications, you can connect the power buses for the TQ10Xs in parallel, through the **V BUS+** and **V BUS-** terminals, located on the motor connector, as shown below. (See the previous drawing, *Regeneration Circuit*, for an internal schematic.)



Sharing the Power Bus

With the buses connected in parallel, the regenerated power from one TQ10X is dissipated by the power consumption of the other.

In standard multi-axis applications, where regeneration occurs when one axis decelerates, the energy can be used by other axes if the power bus is shared. This improves system efficiency by using energy that would otherwise be wasted.

Connecting Motors from Other Vendors

Before connecting a motor from another vendor, you must determine which motor phase wires correspond to Phase A, Phase B, and Phase C inputs on the TQ10X. Similarly, you must determine which Hall effect wires correspond to Hall 1, Hall 2, and Hall 3.

Connect each wire to its appropriate terminal on the TQ10X. Ensure that the Hall effect sensors accurately transmit information about rotor position, and that motor current is commutated to the correct motor phases. See the *Appendix* at the end of this user guide for more information about using a motor from a vendor other than Compumotor.

Connecting a Brushed DC Servo Motor

You can use the TQ10X as a drive for brushed DC servo motors. To do this, follow these steps:

① Connect drive terminals **HALL 1** and **HALL 2** to **HALL GND**

② Make no connections to drive terminal **HALL 3**

③ Connect the drive's **PHASE A** to your motor's positive input.

④ Connect the drive's **PHASE C** to your motor's negative input.

Under these conditions (**HALL 1** and **HALL 2** connected to ground; **HALL 3** unconnected), the drive's internal logic determines that a brushed motor is connected. The drive will send DC current out of Phase A, through the motor, and back into the drive through Phase C. The amount and polarity of the current will be determined by the command input signal.

Shielded Motor Cables

Keep electrical noise from interfering with the signals that the Hall effect sensors send to the drive. Position the motor as close as possible to the drive. If you need to connect a long cable between the drive and motor, we recommend you use a shielded cable for the Hall wires (Hall 1, Hall 2, Hall 3, +5V, Hall Gnd). Run the power wires (phase A, B, and C) separately from the Hall wires.

Motor Grounding

For safety reasons, the motor case should be grounded. Often, the motor can be grounded through the equipment to which it is mounted. This requires a good electrical connection between the motor's mounting flange and the equipment, and that the equipment be connected to ground. Check with the National Electrical Code (NEC) and your local electrical code to ensure you use proper grounding methods.

Proper grounding can also reduce electrical noise.

GROUND THE MOTOR CASE!

The motor case must be grounded, to reduce electrical noise. An ungrounded motor can cause electrical noise problems throughout the system, particularly in encoder wiring and circuitry. This noise may cause the encoder to output erroneous information, such as missing encoder pulses. To avoid electrical noise problems, ground the motor case.

6. Connect Inputs and Outputs

This section describes how to connect inputs and outputs to the TQ10X.

CAUTION

I/O is not OPTO isolated. For greater noise immunity, we recommend the use of optical isolation modules. For added noise immunity, this controller has a digital filter; each input must be true for three successive clock cycles before recognizing a given state.

Enable Input

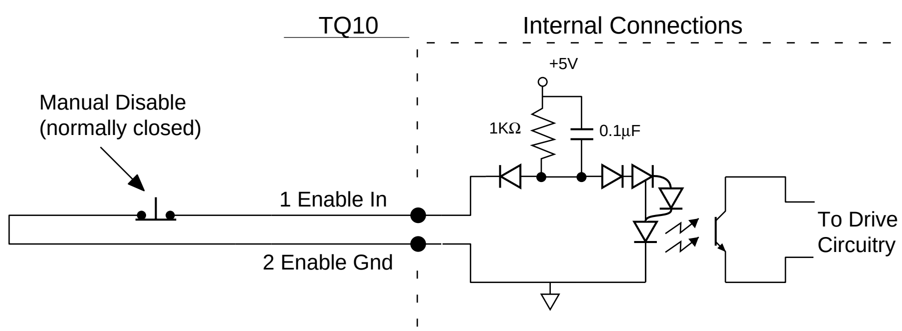
To enable the drive, you must connect the enable input to ground. If you break the connection to ground while the drive is on, the TQ10X's fault circuitry will activate, with the following results:

- The drive will shut down power output to the motor.
- The motor will freewheel (it may not stop immediately).
- The bicolor LED labeled **GRN = POWER ON**, **RED = NOT ENABLED** will be illuminated red.
- The fault output will become active (no current will flow through it).

To re-enable the drive, establish the connection between enable and ground, and cycle power.

In most applications, you can permanently wire the enable input to ground. This input is internally pulled up to +5V. If your equipment requires, you can connect the input to an external voltage as high as +24V. You can also connect it to a dry contact closure to ground.

If you need to disable the drive in an emergency, use the enable input. Connect a *manual disable* switch to the enable input, as the next drawing shows. The switch is normally closed. When it is opened, the drive will be disabled. The load can freewheel—therefore, you should use a brake to stop the motor immediately in applications where a freewheeling motor can cause injury or damage.



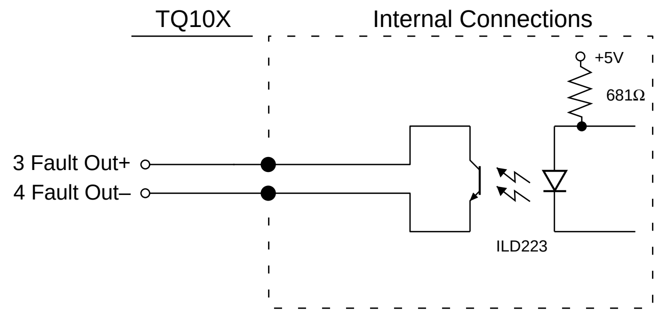
Enable Input Connected to a Switch

WARNING

Do not use the ENABLE INPUT by itself as an emergency stop. The motor can freewheel when the drive is disabled and may not stop immediately. Use a mechanical brake or some other method to stop the motor quickly.

Fault Output (Optional)

When the TQ10X is operating normally, its fault output's internal transistor is in the "on" state, and conducts current. If the TQ10X detects a fault, it turns off the transistor, and current stops flowing.



Fault Output

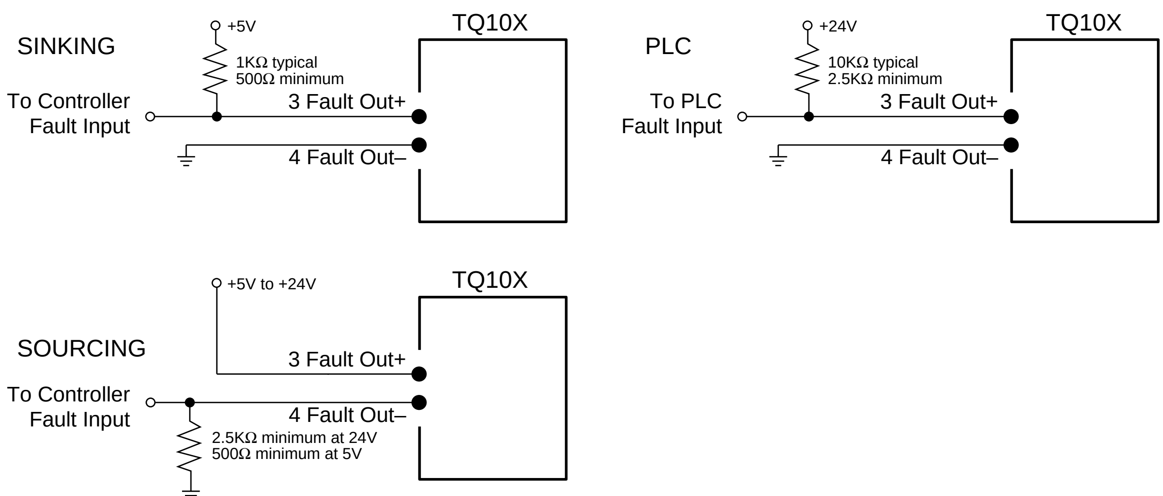
Specifications	
Maximum Applied Voltage	50 V
Maximum Current	10 mA
Active Level	No Fault: Transistor *on*, current flows
	Fault: Transistor *off*, no current flows

The following conditions will activate the fault output.

Fault Condition	
Drive Not Enabled	
Over Temperature	*latched*
Overvoltage	*latched*
Undervoltage	*latched*
Excess Position Error	*latched*
Short Circuit	*latched*
Power Supply Fault	*latched*
Foldback	*foldback causes fault if DIP SW2, Position#1 is set to ON*

Latched means you must cycle power before the drive will operate again.

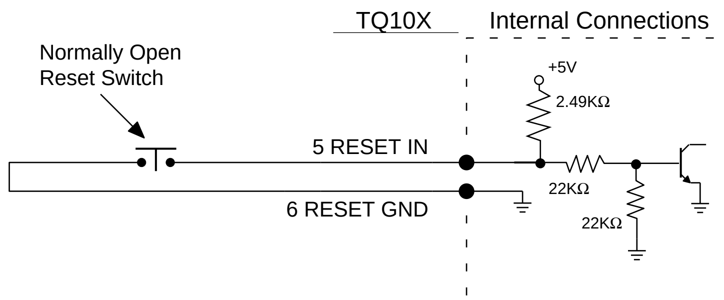
You can use the TQ10X's fault output as a signal to a PLC or other equipment that a fault has occurred. The following drawing shows several ways to connect external devices to the TQ10X's fault output.



Fault Output – Typical Applications.

Reset Input (Optional)

You can use the reset input to reset the drive and the controller. The effect of a reset is identical to cycling power. The reset input is internally pulled up to +5V. If your equipment requires, you can connect the input to an external voltage as high as +24V. You can also connect it to a dry contact closure to ground.

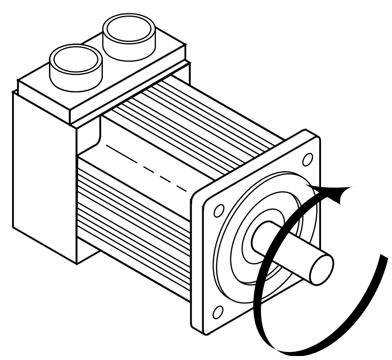


Reset Input

Specifications	
Maximum Applied Voltage	30 V
Maximum Current	5 mA
Active Level	Low to reset

Rotation Direction

In the following sections, we will refer to clockwise and counterclockwise directions. As the next drawing shows, shaft rotation is defined as the direction the shaft rotates, as viewed from the mounting flange end of the motor.



Clockwise Shaft Rotation

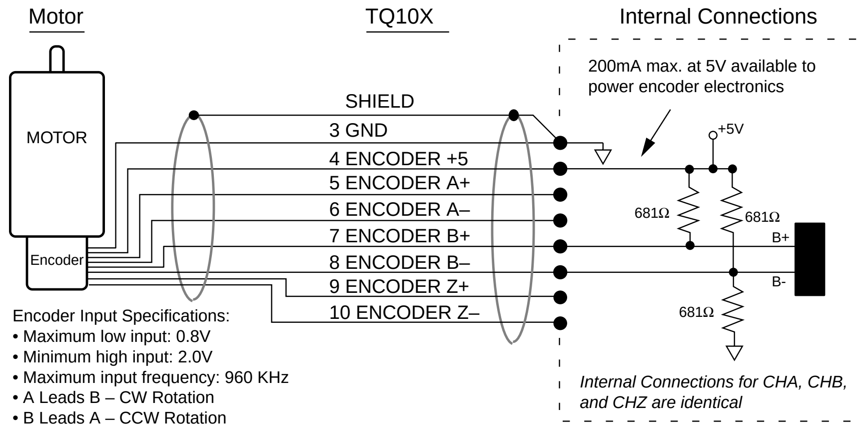
Maximum Voltage Level on I/O Signal Inputs

CAUTION

Do not apply more than 5V maximum to inputs described in the following sections.

Encoder Input Connections

The TQ10X has six dedicated inputs for use with a differential incremental encoder. These inputs provide position information for the servo loop.



Encoder Input

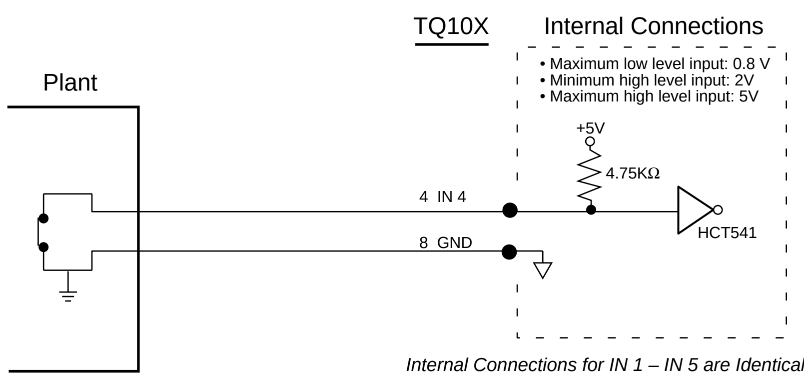
The color code for encoder wires was shown earlier for Compumotor SM and NeoMetric Series servo motors. Connect the cable's shield wire to the encoder ground terminal (pin 3).

CAUTION

If you do not connect an encoder Z-channel output to the TQ10X, then you must ground the Z+ input on the TQ10X. To do this, connect a jumper wire between the Z+ input and the nearest available ground terminal (labeled GND).

General Purpose Inputs (IN 1 – IN 5)

The TQ10X has five general purpose inputs. Each of these inputs may be configured to match the application needs. The figure represents a typical configuration of one of these inputs.



General Purpose Input Connected to a Switch

The **IN** command is used to configure the inputs to the following functions:

TRIGGER INPUT – The TQ10X can dedicate up to five Trigger inputs. These inputs are pulled up internally. These inputs are used with the Trigger (**TR**) command to control the TQ10X's trigger function. Minimum pulse width is 1 ms.

HOME POSITION INPUT – The TQ10X can dedicate up to one Home input. The Home input allows you to establish a home reference position. This input is not active during power-up. Refer to the Go Home (**GH**) command for more information on setting up and using this function. Minimum pulse width is 1 ms.

SEQUENCE SELECT INPUT – The TQ10X can dedicate up to three Sequence Select inputs that allow you to control seven different sequences. Sequences are executed remotely by using one of the following logic patterns in conjunction with the **XP** command.

Sequence #	**Ø**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
SEQ Input #1	Ø	1	Ø	1	Ø	1	Ø	1
SEQ Input #2	Ø	Ø	1	1	Ø	Ø	1	1
SEQ Input #3	Ø	Ø	Ø	Ø	1	1	1	1

Ø = low, pulled to ground

1 = high, 5VDC

STOP or KILL INPUT – The TQ10X can dedicate up to one Stop and one Kill input. The active state is high. The Stop or Kill input is identical in function to the effect of the **S** or **K** command respectively.

GO INPUT – The TQ10X can dedicate up to one Go input. The active state is high. The Go input is identical in function to the effect of the GO (**G**) command.

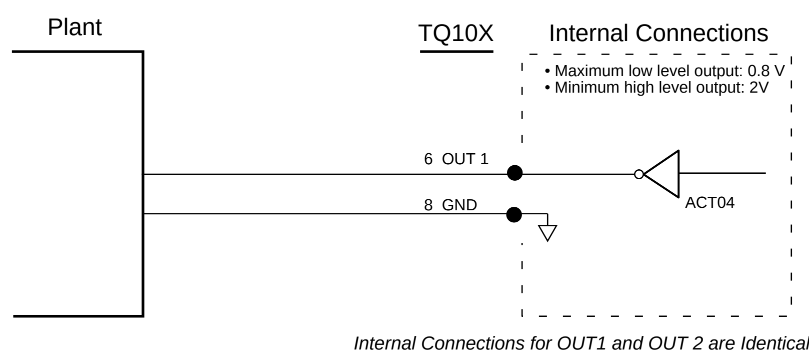
NOTE

Unless configured otherwise (**SSH** command), the controller will dump the commands following the IN command in the buffer. Please pay special attention to the state of the inputs before entering the IN command.

Output #1 (OUT 1) and Output #2 (OUT 2)

The TQ10X has two dedicated programmable +5 volt outputs. They may be used to signal peripheral devices upon the start or completion of a move. The default state for Outputs #1 and #2 is logic low. Refer to the Output (**O**) command for information on using these outputs.

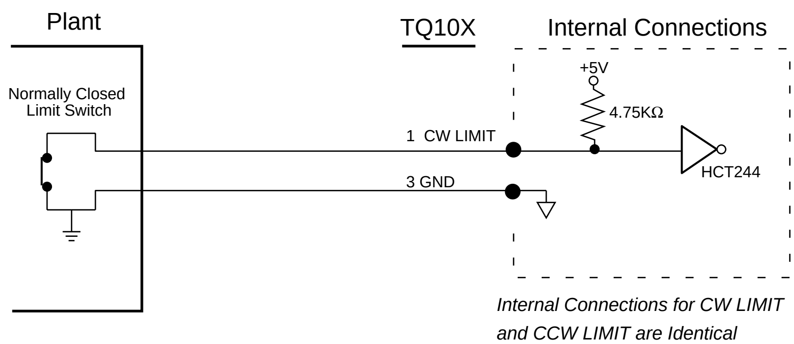
The next drawing shows the schematic for one of the outputs.



General Purpose Outputs

CW LIMIT & CCW LIMIT Inputs

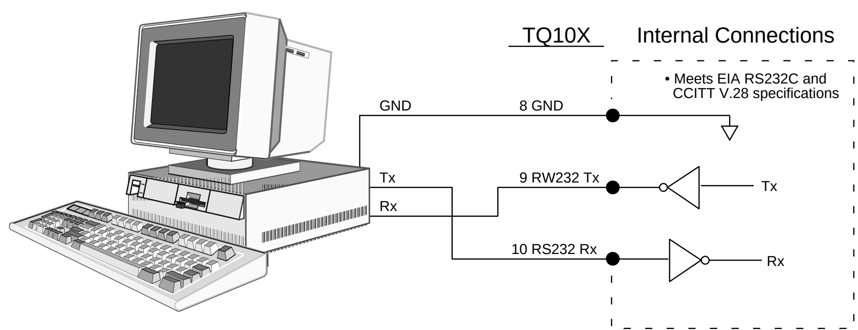
The TQ10X has two dedicated hardware end-of-travel limits (**CW LIMIT** and **CCW LIMIT**). When you power up the TQ10X, these inputs are enabled (high). To test the TQ10X without connecting the CCW and CW limits, you must disable the limits with the **LD3** command. You can use the Limit Switch Status Report (**RA**) and Input Status (**IS**) commands to monitor the limits' status. The following figure represents a typical configuration of these inputs. Minimum pulse width is 1 ms.



Hardware Limit Switch Inputs

Connecting RS232-C Communications

The TQ10X uses RS-232C as its communication medium. The controller does not support handshaking. A typical three-wire (Rx, Tx, and Signal Ground) configuration is used. The figure represents a typical RS-232C configuration.

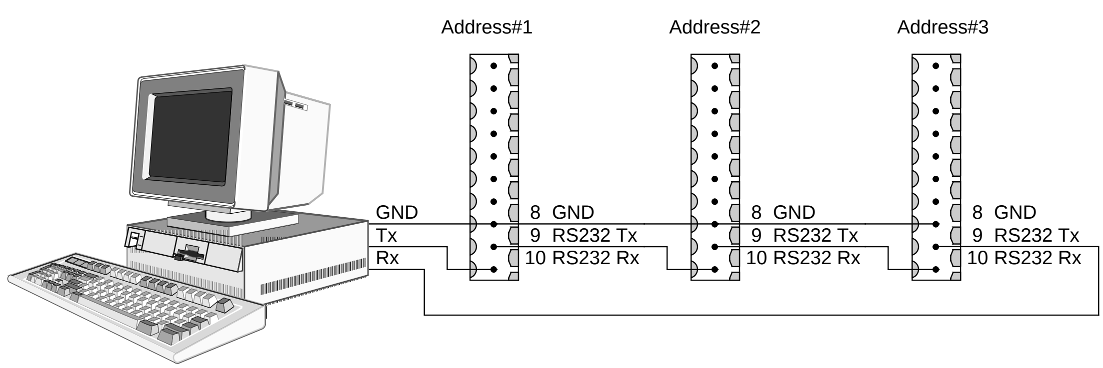


RS232-C Connections

Daisy Chaining

You may daisy chain up to 255 TQ10Xs. Individual drive addresses are set with the **#** (Address Numbering) command. When daisy chained, the units may be addressed individually or simultaneously. You should establish a unique device address for each TQ10X.

Refer to the figure below for TQ10X daisy chain wiring.



RS232-C Daisy Chain of Three TQ10Xs

Commands prefixed with a device address control only the unit specified. Commands without a device address control all units on the daisy chain. The general rule is: *Any command that causes the drive to transmit information from the RS-232C port (such as a status or report command), must be prefixed with a device address*. This prevents daisy chained units from all transmitting at the same time.

Attach device identifiers to the front of the command. The Go (**G**) command instructs all units on the daisy chain to go, while **1G** tells only unit 1 to go.

When you use a single communications port to control more than one TQ10X, all units in a daisy chain receive and echo the same commands. Each device executes these commands, unless this command is preceded with an address that differs from the units on the daisy chain. This becomes critical if you instruct any TQ10X to transmit information. To prevent all of the units on the line from responding to a command, you must precede the command with the device address of the designated unit.

7. Connect AC Power

At this point in your installation procedure, you should have mounted your drive and motor, and connected motor cables and encoder cables to the drive.

The TQ10X does not have an *on/off* switch. When you plug the power cord into the drive, the system will turn on. Therefore, before you apply power to the TQ10X, verify the following:

- Motor should be properly secured
- Load should not yet be connected to motor shaft
- Motor cable should be connected to drive
- Drive should be properly mounted
- Encoder cable should be connected to drive
- Encoder cable should not be located close to motor cable

Apply Power

Apply power to the TQ10X by plugging one end of the molded power cord into the drive's AC Power connector. The cord is 6 feet (1.8 m) long. Plug the other end of the power cord into an AC power source that meets the following specifications:

Specifications – AC Power Input

Input Power:	120VAC	nominal
	95VAC	minimum
	132VAC	maximum
	50 – 60 Hz	
Fuses:	No user serviceable fuses	
Grounding:	You must provide a proper AC power ground	

WARNING

The motor case and drive are grounded through the AC power connector ground pin. You must provide a proper AC power ground for safety purposes.

Peak Power Ratings

The amount of power the TQ10X draws from your AC power source depends upon the motor you use and upon your specific application. For high power applications, we suggest you use a dedicated 20 amp service for the TQ10X. Even applications that use low *average* power may require 20 amp service if the *peak* power is high. Inadequate AC power can cause various problems that are difficult to diagnose.

8. Test Your System

Except for connecting the motor to the load, system installation should be complete at this point. Perform the test procedure below to verify that your system is functioning properly.

In the test procedure, you will command single revolution moves in the clockwise and counterclockwise direction. If your mechanics do not permit such moves, choose a move that allows you to easily verify correct system response.

CAUTION

If you have an SM or NeoMetric Motor, use the drive's DIP switches to set the peak current at twice the motor's continuous current rating, or less. Motor damage due to excessive heating may result from the combination of high peak current and improper tuning values.

Test Procedure

① Apply 120VAC power. The bicolor LED labeled GRN = POWER ON should be illuminated green.

② Command a slow move of one revolution in the clockwise direction. Verify that the motor turns as commanded. (If you have not connected hardware limit switches, send the command LD3 to disable limits, before you send a GO command.)

③ Command a slow move of one revolution in the counterclockwise direction. Verify that the motor turns as commanded.

④ Test any of the optional inputs and outputs that you have connected.

You may need to tune your system before you can obtain motion from the motor. See *Chapter 4 Tuning* for tuning instructions.

Successful completion of this procedure will verify that your motor and encoder are correctly connected to the TQ10X, and that the drive is functioning properly.

If the test was unsuccessful, observe the LEDs on the front panel of the TQ10X while you try the test procedure—they may indicate the cause of the problem. (*Chapter 6 Troubleshooting* has a complete description of LED functions.) Review earlier sections of this user guide, verify that you have completed each step, and try this test procedure again.

If the test is still unsuccessful, proceed to *Chapter 6 Troubleshooting* for problem identification and solution procedures.

9. Connect the Motor to the Load – Couplers

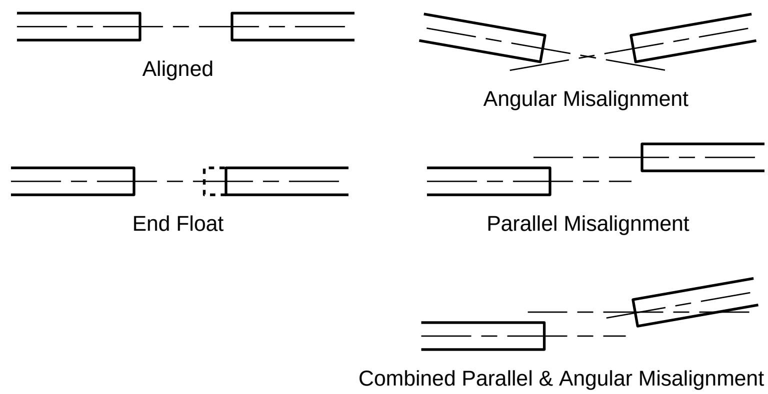
Your mechanical system should be as stiff as possible. Because of the high torques and accelerations of servo systems, the ideal coupling between a motor and load would be completely rigid. Rigid couplings require perfect alignment, however, which can be difficult or impossible to achieve. In real systems, some misalignment is inevitable. Therefore, a certain amount of flexibility may be required in the system. Too much flexibility can cause resonance problems, however. These conflicting requirements are summarized below.

- Maximum Stiffness (in the mechanical system)
- Flexibility (to accommodate misalignments)
- Minimum Resonance (to avoid oscillations)

The best design may be a compromise between these requirements.

Misalignment & Couplers

Align the motor shaft and load as accurately as possible. In most applications, some misalignment is unavoidable, due to tolerance variations in components. However, excessive misalignment may degrade your system’s performance. The three misalignment conditions, which can exist in any combination, are illustrated and described in the next drawing.



Misalignment Conditions

- Angular Misalignment: The center lines of two shafts intersect at an angle other than zero degrees.
- Parallel Misalignment: The offset of two mating shaft center lines, although the center lines remain parallel to each other.
- End Float: A change in the relative distance between the ends of two shafts.

The type of misalignment in your system will affect your choice of coupler.

Single-Flex Coupling

Use a single-flex coupling when you have angular misalignment only. Because a single-flex coupling is like a hinge, one and only one of the shafts must be free to move in the radial direction without constraint. ***Do not use a double-flex coupling in this situation***: it will allow too much freedom and the shaft will rotate eccentrically, which will cause large vibrations and catastrophic failure. ***Do not use a single-flex coupling with a parallel misalignment***: this will bend the shafts, causing excessive bearing loads and premature failure.

Double-Flex Coupling

Use a double-flex coupling whenever two shafts are joined with parallel misalignment, or a combination of angular and parallel misalignment (the most common situation).

Single-flex and double-flex couplings may or may not accept end play, depending on their design.

Rigid Coupling

Rigid couplings are generally not recommended, because they cannot compensate for *any* misalignment. They should be used only if the motor or load is on some form of floating mounts that allow for alignment compensation. Rigid couplings can also be used when the load is supported entirely by the motor’s bearings. A small mirror connected to a motor shaft is an example of such an application.

Coupling Manufacturers

HUCO	ROCOM CORP.	HELI-CAL
70 Mitchell Blvd, Suite 201	5957 Engineer Drive	P.O. Box1460
San Rafael, CA 94903	Huntington Beach, CA 92649	Santa Maria, CA 93456
(415) 492-0278	(714) 891-9922	(805) 928-3851

Resonance Issues

A coupler that is too flexible may cause a motor to overshoot its commanded position. When the encoder sends a position feedback signal, the controller will command a correction move in the opposite direction. If the resonant frequency of the system is too low (too flexible), the motor may overshoot again and again. In extreme cases, the system could become an oscillator.

To solve resonance problems, increase the mechanical stiffness of the system to raise the resonant frequency so that it no longer causes a problem.

If you use a servo as a direct replacement for a step motor, you may need to modify your mechanical coupling system to reduce resonance. For example, we recommend using a bellows-style coupler with servo motors, rather than the helical-style coupler that is often used with step motors. Helical couplers are often too flexible, with resonant frequencies that can cause problems. Bellows couplers are stiffer, and perform better in servo systems.

What's Next

This completes the system installation procedure. Proceed to *Chapter 4 ,Tuning* for instructions about tuning the TQ10X Drive.

CHAPTER THREE

Specifications

IN THIS CHAPTER

- Drive Specifications
- SM and NeoMetric Motor Specifications
- SM and NeoMetric Motor Speed/Torque Curves
- SM and NeoMetric Motor Dimensions
- SM and NeoMetric Encoder Specifications
- SM and NeoMetric Motor Part Numbering System

Specifications – TQ10X Servo Controller/Drive

Parameter	Value
Power Input	
Voltage	95-132 VAC single phase
Frequency	50/60 Hz
Power Output—Motor	
Peak Current	10 A (approx 2 sec max duration at 45°C ambient temperature)
Continuous Current	5 A with TQ-HS3 or mounted to heatsink at or below 45°C
Voltage	170 VDC nominal
Peak Power	830 W (1.1 hp) @ 170 V supply voltage
Continuous Power	420 W (0.56 hp)
Switching Frequency	20 kHz
Bandwidth	2 kHz typical (dependent on motor)
Transconductance	1 volt = 1.0 amp
Commutation	120° hall effect sensors for six-state commutation method or brushed DC motor
Short-Circuit Protected	Yes
Power Output—Hall Effect Sensors	
Voltage	+5 VDC ±0.5 VDC
Current	50 mA (max)
Short-Circuit Protected	Yes
Power Output—To Encoder	
Voltage	+5 VDC
Current	200 mA max each output
Hall Inputs	
Low State	Ø-2V
High State	Internal 3 k pull-up resistor to +15 V (open-collector Hall outputs should be used)
Input Frequency	Ø–2 kHz max
Inputs	
Programmable Inputs	5 user-defined, TTL signal levels: low = 0.8 V; high = 2–5V
End-of-travel limits	CW/CCW, 0–5 V, TTL signal levels: low = 0.8 V; high = 2–5V
RS-232C	3-wire (Rx, Tx, GND) connections, RS485 optional
Outputs	
Programmable outputs	2 user-defined, TTL signal levels: low = 0.8 V; high = 2–5V
Fault Output–Isolated	24 V max voltage; 10 mA max current
LED Name	Indication
-Enable (Bicolor)	Green = enabled; Red = power on, not enabled
-Drive temp	Red = fault (drive overtemp, etc.)
-Motor fault	Red = fault (short circuit, motor overtemp, etc.)
-Peak current/Foldback	Green = current is near peak (over~75%)
(Bicolor)	Red = in foldback (peak current time exceeded)
-Regen (Bicolor)	Green = regen active; Red = overvoltage fault
Performance	
Position Range	±1,073,741,823
Velocity Range	0.01 to 200 rps
Acceleration Range	0.01 to 9999 rps2
Velocity Accuracy	±0.02% of max rate
Velocity Repeatability	±0.02% of set rate
Resolution	400–65,532 encoder counts/rev
Digital Servo Loop	
Update Time	266 µs
Output	12-bit DAC
Servo Tuning	Via RS-232C
Tuning Parameters	PID with digital filter
Protective Circuits	
Short Circuit	Turns off outputs to motor; latched
Overtemperature	55°C ±5°C trip temperature; latched
Undervoltage	80 V min
Memory	
2K of BBRAM	For program, RS232 address, tuning, and setup storage
Physical	
All connection points	10 pin screw terminal, removable
Environment	
Drive Temperature ambient	0-50°C (32-122°F)
Humidity	0-95% non-condensing
Storage	-40-85°C (-40-185°F)

Motor Specifications – Compumotor SM Series Servo Motors

	Symbol	Units	SM161A	SM162A	SM231A	SM232A	SM232B	SM233A	SM233B
Continuous Stall Torque[1]	T_{CS}	oz-in	24	44	46	92	92	140	140
		N-m	0.17	0.31	0.32	0.65	0.65	0.99	0.99
Continuous Stall Current[1]	I_{CS}	amps-rms	2.1	2.1	2.1	2.1	4.1	2.1	4.1
Rated Speed[2]	ω_T	rpm	7,500	7,500	7,500	4,500	7,500	2,800	7,500
		rps	125	125	125	75	125	47	125
Peak Torque[1]	T_{pk}	oz-in	72	132	138	276	276	420	420
		N-m	0.51	0.83	0.97	1.95	1.95	2.97	2.97
Peak Current, rms[1]	I_{pk}	amperes	6.3	6.3	6.3	6.3	12.3	6.3	12.3
Torque @ Rated Speed[1]	T_C	oz-in	20	41	38	78	84	134	114
		N-m	0.14	0.29	0.27	0.55	0.59	0.95	0.81
Rated Power-Output Shaft[1]	Po	Watts	110	222	205	260	362	277	505
Resistance[4]	Ω	ohms	4.53	6.50	5.22	7.50	2.01	9.65	2.58
Inductance[3]	L	millihenries	0.808	1.39	1.64	2.9	0.78	4.08	1.06
Thermal Resistance[1]	R_{th}	°C/watt	2.75	2.00	2.23	1.58	1.50	1.25	1.26
Thermal Time Constant	T_{th}	minutes	30	30	30	35	35	40	40
Electrical Time Constant	T_{te}	milliseconds	0.178	0.21	0.31	0.39	0.39	0.42	0.41
Mechanical Time Constant	T_m	milliseconds	9.2	5.0	13.7	8.6	8.8	5.4	7.0
Rotor Inertia	J	lb-in-sec^2	0.000094	0.00016	0.00046	0.00082	0.00082	0.00117	0.00117
		oz-in^2 (mass)	0.58	0.99	2.84	5.07	5.07	7.23	7.23
		kg-m^2 x 10^{-6}	10.62	18.07	51.97	92.65	92.65	132.19	132.19
Weight	#	pounds (kg)	1.1 (0.5)	1.6 (0.7)	2.6 (1.2)	3.5 (1.6)	3.5 (1.6)	4.4 (2.0)	4.4 (2.0)

Data listed is for SM motors alone. Drive specifications may change some values.

[1] @25°C ambient with 10 x 10 x 0.25 in. mounting plate, 90°C winding temperature. For 40°C ambient operation, reduce values by 12%.

[2] With 500 ppr encoders. For 1,000 ppr encoders, derate to 6000 rpm.

[3] ±30%, line-to-line, inductance bridge measurement method @ 1 kHz

[4] ±10%, line-to-line, at 25°C

Motor Specifications – Compumotor NeoMetric Series Servo Motors (70mm)

Parameter	Symbol	Units	N0701D N0341D	N0701F N0341F	N0702E N0342E	N0702F N0342F	N0703F N0343F	N0703G N0343G	N0704F N0343F	N0704G N0344G
Stall Torque Continuous[1,4]	T_{CS}	lb-in	5.8	5.8	10.6	10.6	16.2	16.2	20.0	20.0
		N-m	0.65	0.65	1.19	1.19	1.83	1.83	2.26	2.26
Stall Current Continuous[1,2]	I_{CS}	amps-rms	2.92	4.56	3.36	4.67	4.59	6.38	4.73	6.57
Rated Speed	ω_r	rpm	4850	7500	3025	4380	2825	4050	2365	3390
		rps	80.8	125	50.4	73	47.1	67.5	39.4	56.5
Peak Torque[1]	T_{pk}	lb-in	17.3	17.3	32.0	32.0	48.6	48.6	60.0	60.0
		N-m	1.95	1.95	3.62	3.62	5.49	5.49	6.78	6.78
Peak Current, rms[1,6]	I_{pk}	amperes	8.8	13.7	10.0	14.0	13.7	19.1	14.2	19.7
Torque @ Rated Speed[1]	T_C	lb-in	4.6	4.6	8.6	8.6	13.0	13.0	15.8	15.8
		N-m	0.52	0.52	0.97	0.97	1.47	1.47	1.79	1.79
Rated Power-Output Shaft[1]	Po	watts	265	415	309	447	435	624	442	634
Voltage Constant[3,4]	K_b	volts/radian/sec	0.221	0.14	0.353	0.253	0.392	0.282	0.468	0.338
Voltage Constant[3,4]	K_e	volts/KRPM	23.11	14.67	36.97	26.52	40.99	29.54	49.02	35.36
Torque Constant[3,4]	K_t	lb-in/amp rms	1.95	1.24	3.12	2.24	3.46	2.50	4.14	2.99
Resistance[3]	R	ohms	5.52	2.27	5.22	2.7	3.36	1.74	3.47	1.80
Inductance[5]	L	millihenries	12.98	5.23	15.86	8.16	12.13	6.30	14.50	7.55
Thermal Resistance[1]	R_{th}	°C/watt	1.44	1.44	1.15	1.15	0.96	0.96	0.87	0.87
Motor Constant	K_m	lb-in/√watt	0.83	0.83	1.37	1.36	1.89	1.89	2.23	2.23
Viscous Damping	B	lb-in/Krpm	0.044	0.044	0.05	0.05	0.0563	0.0563	0.0625	0.0625
Torque-Static Friction	T_f	oz. in.	1.4	1.4	2.1	2.1	2.8	2.8	3.5	3.5
Thermal Time Constant	τ_{th}	minutes	45	45	45	45	45	45	45	45
Electrical Time Constant	τ_e	milliseconds	2.35	2.35	3.03	3.03	3.61	3.61	4.19	4.19
Mechanical Time Constant	τ_m	milliseconds	1.3	1.3	0.77	0.77	0.54	0.54	0.52	0.52
Rotor Inertia	J	lb-in-sec^2	0.000106	0.000106	0.000173	0.000173	0.000240	0.000240	0.000307	0.000307
Weight	#	pounds	3.54	3.54	4.53	4.53	6.04	6.04	7.28	7.28
Winding Class			H	H	H	H	H	H	H	H

1 @25°C ambient with 10 x 10 x 0.25 in. mounting plate, 90°C encoder temperature.
2 RMS current, line-to-line, six state commutation.
3 +/- 10% line-to-line.
4 Peak value.
5 +/- 30% line-to-line, inductance bridge measurement @ 1KHz.
6 Peak current for 2 seconds maximum with initial winding temperature of 40°C.
All specifications are subject to engineering change.

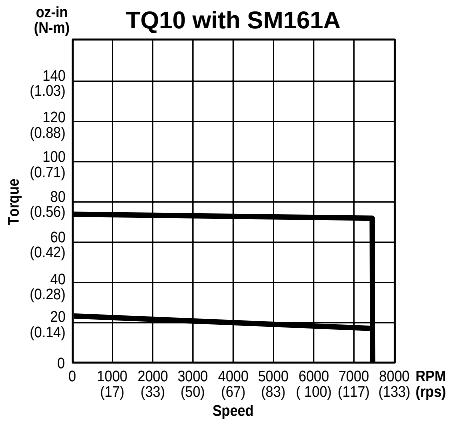
Motor Specifications – Compumotor NeoMetric Series Servo Motors (92mm)

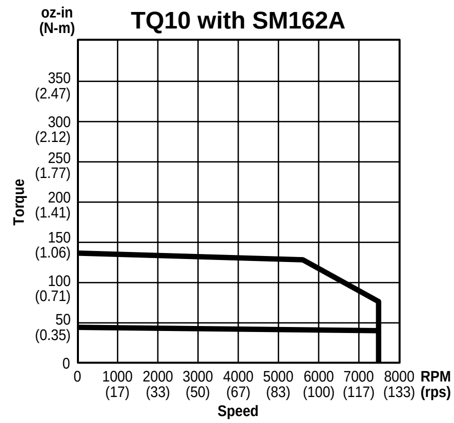
Parameter	Symbol	Units	N0921F	N0921G	N0922G
Stall Torque Continuous[1,4]	T_{CS}	lb-in	16	16	31
		N-m	1.81	1.81	3.50
Stall Current Continuous[1,2]	I_{CS}	amps-rms	4.16	5.76	6.17
Rated Speed	ω_r	rpm	2500	3575	1975
		rps	41.7	59.6	32.9
Peak Torque[1]	T_{pk}	lb-in	48	48	93
		N-m	5.42	5.42	10.51
Peak Current, rms[1,6]	I_{pk}	amperes	12.5	17.3	18.5
Torque @ Rated Speed[1]	T_C	lb-in	12.9	12.9	24.6
		N-m	1.46	1.46	2.78
Rated Power-Output Shaft[1]	Po	watts	383	548	578
Voltage Constant[3,4]	K_b	volts/radian/sec	0.427	0.309	0.556
Voltage Constant[3,4]	K_e	volts/KRPM	44.66	32.37	58.18
Torque Constant[3,4]	K_t	lb-in/amp rms	3.84	2.78	5.0
Resistance[3]	R	ohms	3.72	1.94	2.32
Inductance[5]	L	millihenries	17.11	8.99	14.72
Thermal Resistance[1]	R_{th}	°C/watt	1.06	1.06	0.77
Motor Constant	K_m	lb-in/√watt	1.96	2.00	3.29
Viscous Damping	B	lb-in/Krpm	0.075	0.075	0.0875
Torque-Static Friction	T_f	oz. in.	4	4	6
Thermal Time Constant	τ_{th}	minutes	60	60	60
Electrical Time Constant	τ_e	milliseconds	4.6	4.6	6.4
Mechanical Time Constant	τ_m	milliseconds	1.13	1.13	0.64
Rotor Inertia	J	lb-in-sec²	0.000363	0.000363	0.000623
Weight	#	pounds	8.1	8.1	11.7
Winding Class			H	H	H

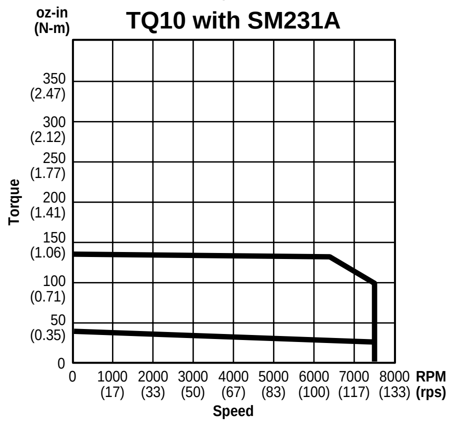
[1] @25°C ambient with 10 x 10 x 0.25 in. mounting plate, 90°C encoder temperature.
[2] RMS current, line-to-line, six state commutation.
[3] +/- 10% line-to-line.
[4] Peak value.
[5] +/- 30% line-to-line, inductance bridge measurement @ 1KHz.
[6] Peak current for 2 seconds maximum with initial winding temperature of 40°C.
All specifications are subject to engineering change.

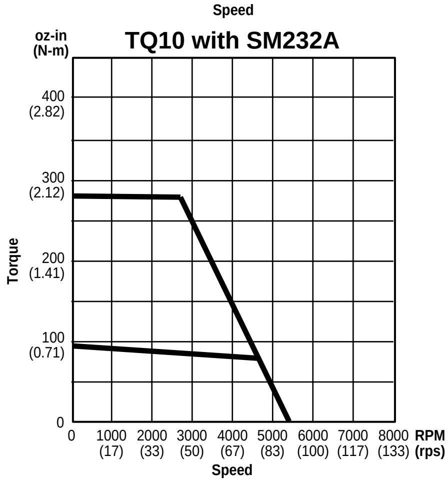
Speed/Torque Curves

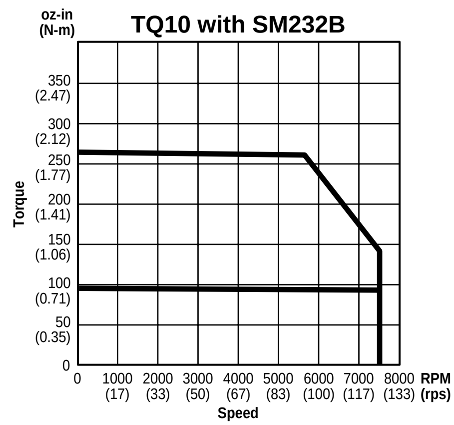
NOTE: Curves are based on 120VAC line, nominal motor K_T and K_C. Actual values may vary ±10%. Line voltage directly limits maximum speed.

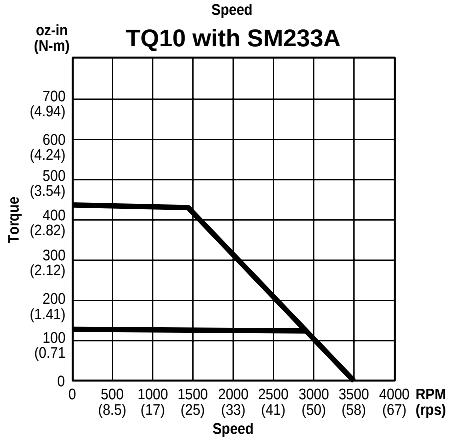


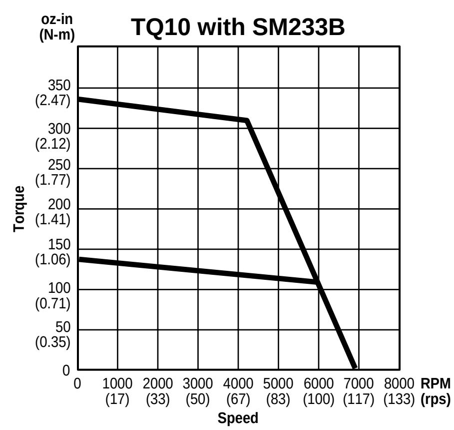






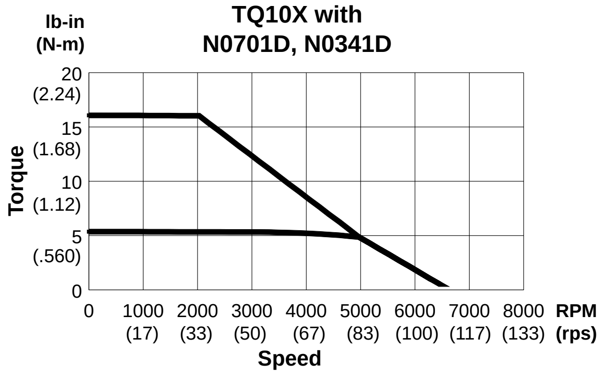


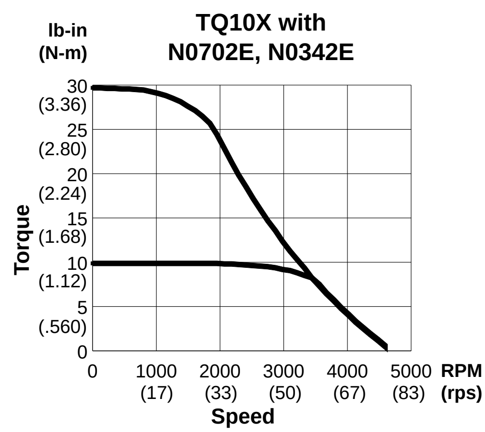


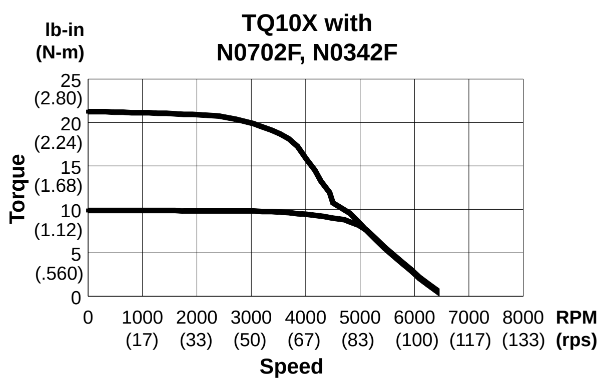


Speed/Torque Curves

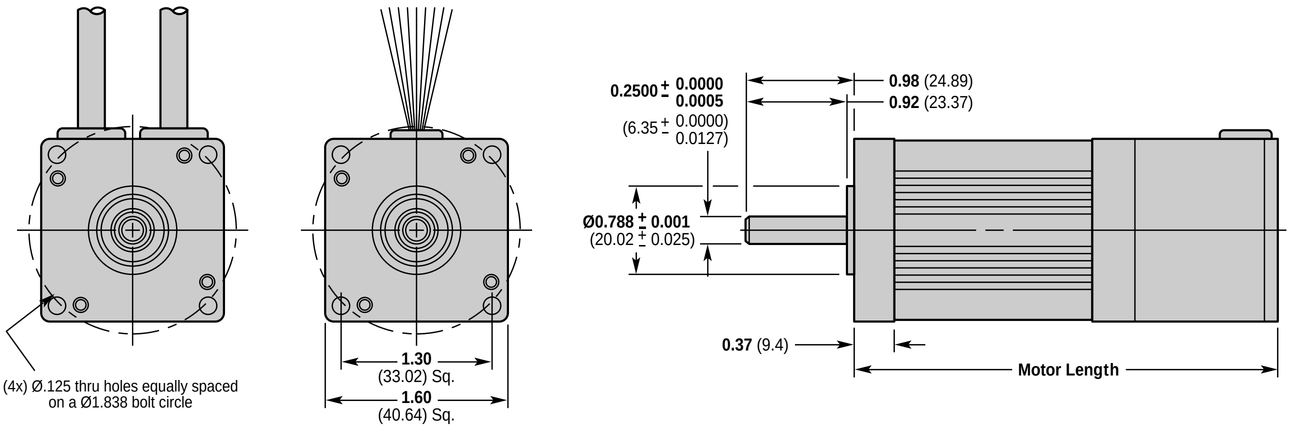
NOTE: Curves are based on 120VAC line, nominal motor K_T and K_C. Actual values may vary ±10%. Line voltage directly limits maximum speed.







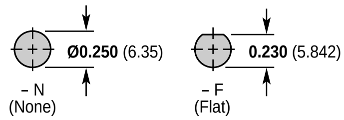
Motor Dimensions – Compumotor SM Series, Size 16



Cable Options	
Part #	Description
– FL	18" Flying Leads
– 10	10 ft. Cable

Longer lengths available
Consult Compumotor for information

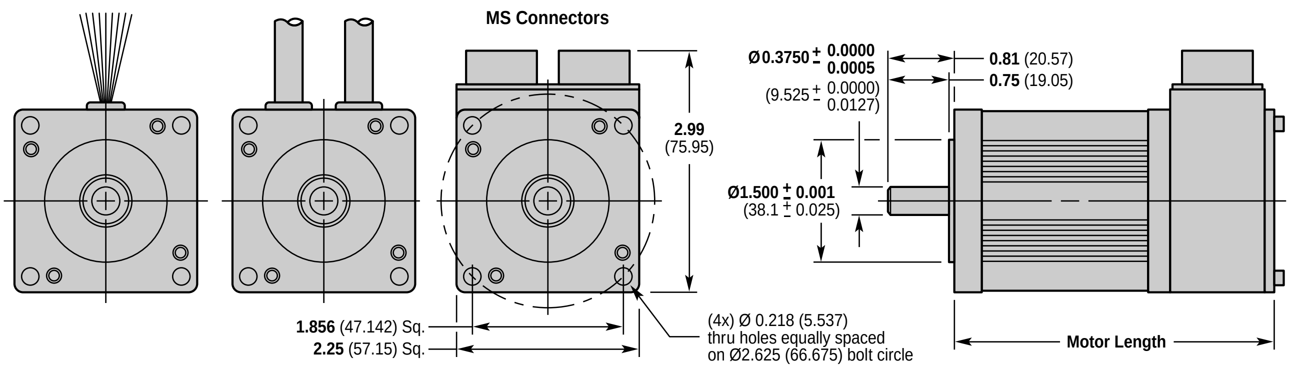
Shaft Options



Dimensions in inches
(millimeters)

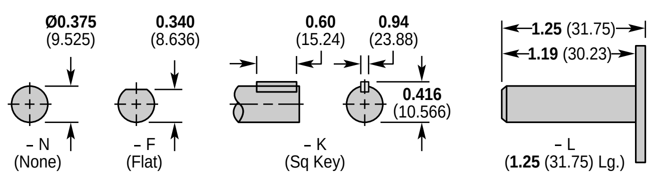
Motor Sizes	
Motor Length	Model
4.79 (121.66)	162 Motor
3.79 (96.27)	161 Motor

Motor Dimensions – Compumotor SM Series, Size 23



Dimensions in inches
(millimeters)

Shaft Options

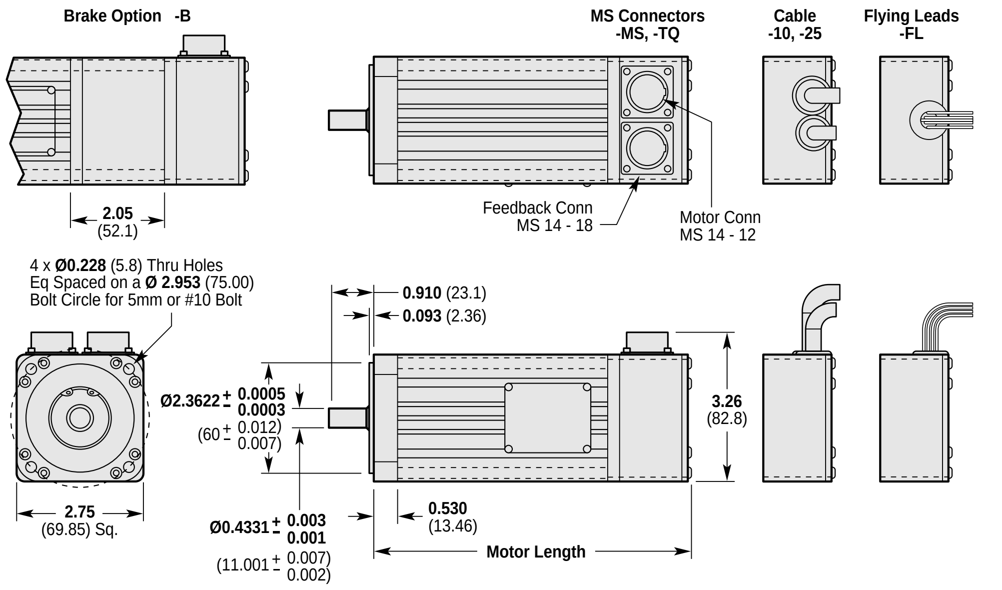


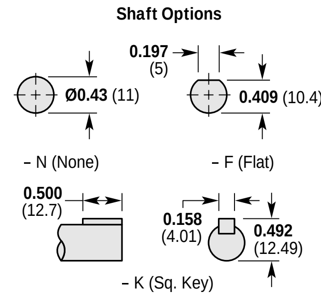
Cable Options	
Part #	Description
– FL	18" Flying Leads
– 10	10 ft. Cable

Longer lengths available
Consult Compumotor for information

Motor Sizes	
Motor Length	Model
5.98 (151.89)	233 Motor
4.98 (126.49)	232 Motor
3.98 (101.09)	231 Motor

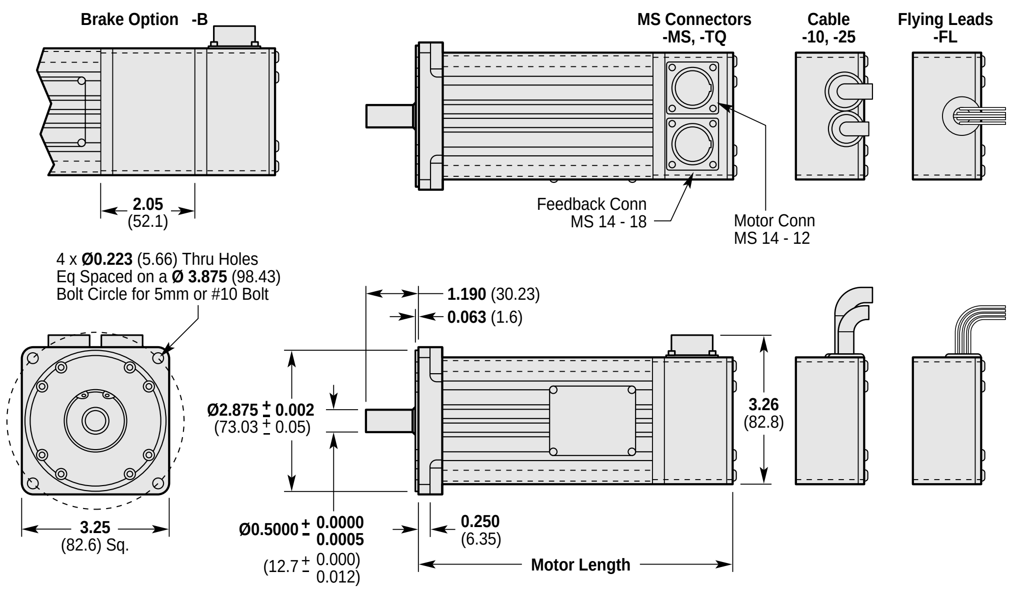
Motor Dimensions – Compumotor NeoMetric Series, Size 70

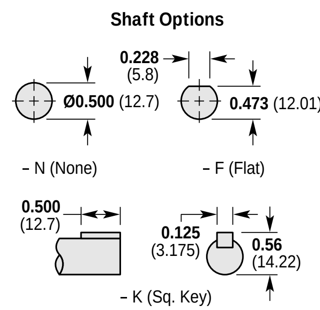




Motor Sizes	
Motor Length	Model
10.00 (254.00)	70-4 Brake
9.00 (228.60)	70-3 Brake
8.00 (203.20)	70-2 Brake
7.00 (177.80)	70-1 Brake
7.94 (201.68)	70-4
6.94 (176.28)	70-3
5.94 (150.88)	70-2
4.94 (125.48)	70-1

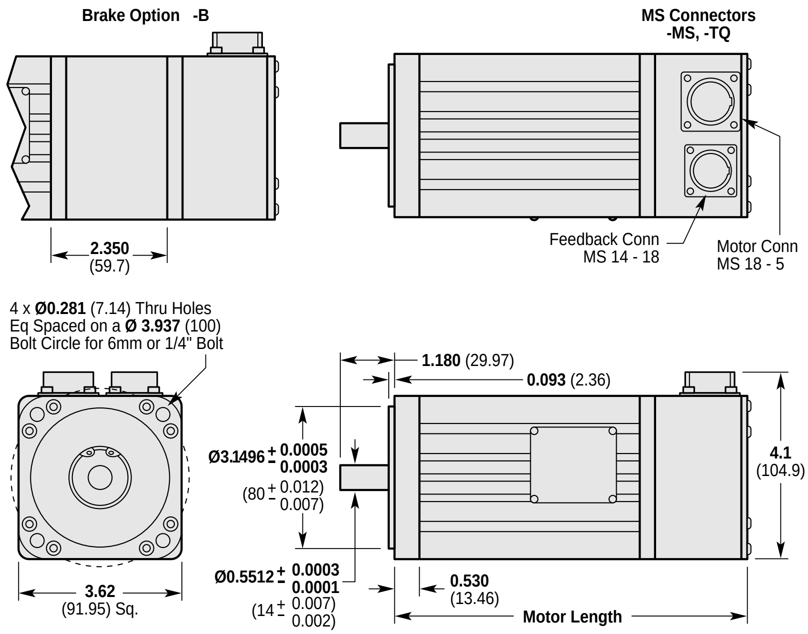
Motor Dimensions – Compumotor NeoMetric Series, Size 34

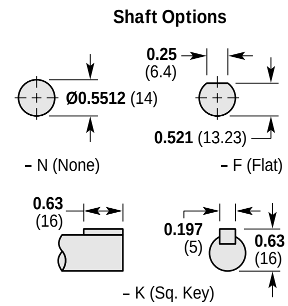




Motor Sizes	
Motor Length	Model
10.00 (254.00)	34-4 Brake
9.00 (228.60)	34-3 Brake
8.00 (203.20)	34-2 Brake
7.00 (177.80)	34-1 Brake
7.94 (201.68)	34-4
6.94 (176.28)	34-3
5.94 (150.88)	34-2
4.94 (125.48)	34-1

Motor Dimensions – Compumotor NeoMetric Series, Size 92





Motor Sizes

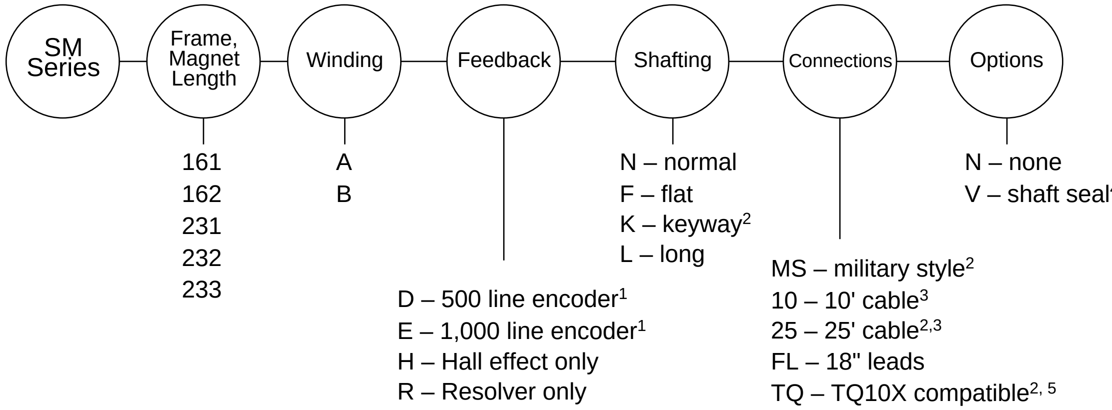
Motor Length	Model
13.50 (342.90)	92-4 Brake
12.00 (304.80)	92-3 Brake
10.50 (266.70)	92-2 Brake
9.00 (228.60)	92-1 Brake
11.13 (282.70)	92-4
9.63 (244.60)	92-3
8.13 (206.50)	92-2
6.63 (168.40)	92-1

Encoder Specifications – Compumotor SM & NeoMetric Series Motors

Parameter	Value
Performance	
Accuracy	±2 min of arc
Electrical	
Input Power	5VDC ± 5% @ 135 mA
Operating Frequency	100 kHz maximum (pre-quadrature)
Output Device	26LS31
Sink/Source, nominal	20 mA
Suggested User Interface	26LS32

Motor Part Numbering System

The diagram below illustrates the part numbering system for Compumotor SM Series servo motors

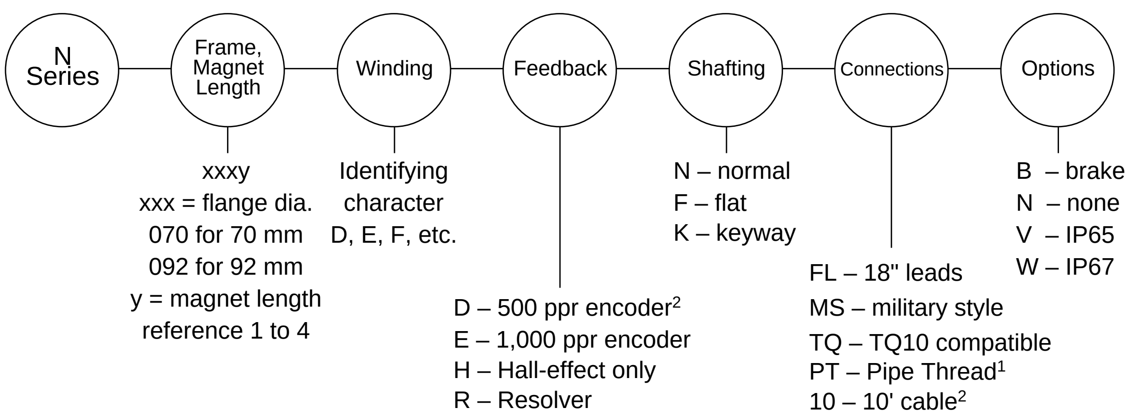


[1] includes Hall effect sensors
[2] not available on size 16 motors
[3] cable is hardwired to motor
[4] size 23 with MS or TQ connectors–IP65
[5] MS recommended for TQ10X; TQ can be used also.

Example: SM-231AE-NMSN = 23 Frame, 1" magnet, A winding, 1,000 line Encoder, Normal shaft, MS cable, No options

Motor Part Numbering System— Compumotor NeoMetric Series Servo Motor

The diagram below illustrates the part numbering system for Compumotor NeoMetric Series servo motors.



[1] 92 mm motors only
[2] 70 mm motors only

Example: N0702EE-KMSN

CHAPTER FOUR

Tuning

IN THIS CHAPTER

Tuning and Performance

You must tune the parameters on the TQ10X's *Proportional Integral Derivative* (PID) filter for optimum system performance. A properly tuned system will exhibit smooth motor rotation, accurate tracking, and fast settling time.

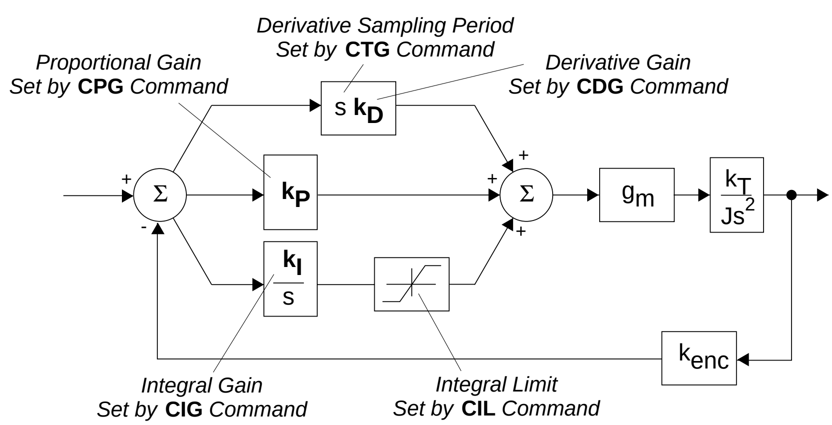
All tuning is performed via RS232-C communications.

PID Tuning

The TQ10X generates a move profile based upon the user supplied acceleration, velocity, and distance commands (**A**, **V**, and **D**). At each servo sampling period (every 266 microseconds), the TQ10X calculates the position the motor should reach as it follows the move profile. This is called *commanded position*, and is one of two inputs to a summing node. Position information from the encoder, which is called *actual position*, is the other input to the summing node. During a typical move, actual position will differ from commanded position by at least a few encoder counts. When actual position is subtracted from commanded position at the summing node, an error signal is produced. The error signal is the input to the PID filter.

The position specified by the distance command (**D**) is called the *target position*. During the move, commanded position is not the same as target position. The commanded position is incremented each sampling period. When it finally matches the target position, the move is over.

The servo block diagram is shown below.



As the figure shows, you can adjust five different parameters to tune the PID filter. The relevant commands are:

CPG	Configure Proportional Gain
CDG	Configure Derivative Gain
CTG	Configure Derivative Sampling Period
CIG	Configure Integral Gain
CIL	Configure Integral Limit

To tune the system, you will iteratively increase **CDG**, **CTG** and **CPG** to their optimum values. If necessary, you will also increase **CIG** and **CIL**.

In general, you will set **CDG** and **CPG** as high as necessary, and **CIG** as low as possible. Trade-offs between response time, stability, and final position error will dictate the values you select. For loads that vary during operations, you can download new parameters by using buffered versions of the five tuning commands (**BCPG, BCDG, BCTG, BCIG, BCIL**).

WARNING

During servo tuning, the system can undergo accidental and violent movement due to improper gain settings and programming errors. Please use extreme caution while prototyping.

Each tuning parameter is described in the following sections.

CPG – Proportional Gain

Proportional gain provides a torque that is directly proportional to the *magnitude* of the error signal. Proportional gain is similar to a spring—the larger the error, the larger the restoring force. It determines the stiffness of the system and affects the following error. High proportional gain gives a stiff, responsive system, but can result in overshoot and oscillation. Damping—provided by derivative gain—can reduce this overshoot and oscillation.

CDG – Derivative Gain; CTG – Derivative Sampling Period

Derivative gain provides a torque that is directly proportional to the *rate of change* of the error signal. The previous error is subtracted from the present error each sampling period. The difference represents the error's instantaneous rate of change, or *derivative*. The difference is multiplied by the value set by the **CDG** command, and the product contributes to the motor control output.

Derivative gain opposes rapid changes in velocity. It will dampen the resonance effects of proportional gain. With higher derivative gain, you can use higher proportional gain.

You can use the **CTG** command to make the derivative sampling period longer than the system's sampling period. The system sampling period—266 µsec—is the period between updates of position error, and cannot be changed. The derivative sampling period is an integer multiple of the system sampling period. It can range from 266 µsec to 68 msec, in increments of 266µsec (for example: CTG0 = 266 µs, CTG1 = 532 µs, CTG2 = 798 µs, etc.).

With a longer derivative sampling period, more time elapses between derivative error measurements. The difference between previous and present error is still multiplied by the CDG value. The product contributes to the motor control output every *system* sampling period, but is only updated every *derivative* sampling period. This gives a more constant derivative term and improves stability. Low velocity systems in particular can benefit from a longer sampling period.

Because of stability considerations, however, the derivative sampling period should be no longer than one tenth of the system mechanical time constant. This means many systems must have low values of **CTG**.

CIG – Integral Gain; CIL – Integral Limit

Integral gain provides a torque that is directly proportional to the sum, over time, of the error values—the *integral* of the error. The controller reads the error value every sampling period, and adds it to the sum of all previous error values. The sum is multiplied by the value set by the **CIG** command (Integral * CIG), resulting in the *integral term* which contributes to the motor control output every system sampling period.

Integral gain can remove steady state errors that are due to gravity or a constant static torque. Integral gain can also correct velocity lag that can occur in a constant velocity system.

If error persists during a move, the sum of the error values may be quite high at the end of the move. In this case, the torque commanded by the integral gain can also be very high, and cause overshoot. This effect is called *integral windup*. Integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term **(Integral * CIG)**.

Tuning the System

In the procedure given below, you will systematically vary the tuning parameters until you achieve a move that meets your requirements for accuracy and response time.

Special Considerations when Tuning with SM or NeoMetric Motors

If you use a high performance motor (peak current rating greater than three times the continuous current rating), be careful not to overheat the motor while tuning your system. If you accidentally choose tuning gains that cause motor instability, excess motor current can quickly overheat and damage the motor—even before the thermostat can trigger the motor overtemperatu re circuit.

CAUTION

For initial tuning with an SM or NeoMetric motor, set peak current DIP switches at twice the motor's continuous rated current, or less. Otherwise, high peak currents during instability may cause motor overheating and damage.

To avoid damage, we recommend a tuning procedure that may differ from methods you have used before. Instability sometimes *does occur* during tuning; to avoid the damaging currents that instability can cause, reduce the peak current *before* you begin the tuning process. Then, as you refine your tuning values, you can gradually increase peak current. These steps are included in the tuning procedure described below.

Preparing the System for Tuning

Before applying power and tuning the drive, complete the following steps.

Setup Procedure:

① **Heatsink your motor:** This is especially important in temporary "bench top" procedures. SM and NeoMetric Motors dissipate excess heat through their faceplate; to ensure proper motor cooling, the faceplate must be mounted to a heatsink.

② **Reduce peak current:** Using the drive's DIP switches, set the peak current at a level that is less than twice the motor's continuous current rating.

- 4.4 amps for SM motors with **-A** windings
- 7.4 amps for SM motors with **-B** windings
- 6.0 amps for N0701D and N0702E motors
- 8.9 amps for N0702F motor

This helps protect SM and NeoMetric motors from overheating when you begin tuning the drive.

③ **Prepare to Disable the Drive Quickly:** Wire a normally closed switch between the **ENABLE IN** input and ground. Use this switch to quickly and reliably disable the drive if the system becomes unstable during tuning.

Avoid Instability

After you change current settings, adjust a tuning parameter, or command a move, closely monitor the motor. If there is any sign of uncontrolled instability, use the switch described in *Step 3* above to quickly disable the drive, and avoid motor damage.

Tuning Procedure

You will achieve best results by making a short, repetitive test move, using high acceleration and high velocity. Because you will start with reduced peak current, the motor may not be able to make the move called for in your application. After you initially tune the system with the short test move, you can try your actual application move and choose final tuning values.

① **Issue a RETURN TO FACTORY SETTINGS command (RFS)**

The RFS command will reset the gains to their default values (CDG240, CTG0, CPG16, CIG2, CIL2, CPE4000)

② **Decrease CIG**

Set the integral gain to zero (**CIG0**).

③ **Command a Short, Repetitive Test Move**

Program the drive to continuously execute a short, repetitive test move. Depending upon your system's mechanics, the move can be back and forth, or in a single direction. Program a five second delay between each move. Use a high acceleration, such as A5000 (or the maximum you expect to accelerate the load); and use a high velocity, such as V100 (or the maximum velocity you expect to move the load). The motor may not reach the commanded velocity during the short move; this does not matter, and will not affect your tuning procedure. An example program is shown below.

Command	Description
CITØ	Sets “In Position” time to Ø (very short)
CEW1ØØØ	Sets “In Position Error Window” to 1ØØØ (very wide)
LD3	Disable limits (use only if appropriate for your system)
MN	Sets positioning mode to preset
A5ØØØ	Sets acceleration to 5ØØØ revs/sec^2 (very high value)
V1ØØ	Sets velocity to 1ØØ revs/sec (very high value)
D4000	Sets distance to one rev (for a 1ØØØ line encoder).
LØ	Loop continuously
G	Executes the move (GO)
T5	Wait 5 seconds at end of move
N	Ends the loop

If the mechanics of your system do not permit the moves described above, modify your test move accordingly.

④ **Observe the Results of the Move**

Closely observe the motor shaft and evaluate motor performance at the end of the move, as indicated by response time, end of position overshoot, following error, and so on. If the move is successful and your motion requirements are satisfied, you do not need to adjust tuning parameters or increase peak current any further—you may proceed to *Step 7*.

Prepare to disable the drive quickly if the motor becomes unstable!

⑤ **Adjust Tuning Parameters**

Vary the tuning parameters to improve motor performance and to achieve *satisfactory* motion—only tune for absolute maximum performance if the application requires it. Typical responses and suggested adjustments are:

Sluggish Motion – Increase proportional gain (**CPG**) or decrease derivative gain (**CDG**) to make the motor respond more quickly. Use caution when you increase **CPG**—too much will cause the system to oscillate and become unstable. If you increase **CPG** and your system becomes unstable, disable the drive immediately.

Oscillatory or Erratic Motion – Increase derivative gain (**CDG**) or decrease proportional gain (**CPG**) to help damp out oscillatory motion. Too much **CDG** will cause a sluggish or overdamped response; excessive **CDG** may cause the system to become unstable. If you increase **CDG** and your system becomes unstable, disable the drive immediately. You can try increasing the derivative sampling period (**CTG**) to control instability. For most applications, **CTG** will not need to be higher than 4.

Steady State Errors – See *Step 9* for information about controlling steady state errors by adjusting integral gain and integral limit parameters. Because integral gain reduces stability, it should only be adjusted after you determine settings for **CPG, CDG,** and **CTG**.

If you still need to improve performance after you adjust **CPG, CDG,** and **CTG**, proceed to *Step 6*.

⑥ Increase Peak Current

Using the DIP switches, increase peak current to the next level. Do not exceed three times the motor's continuous current rating. Closely monitor the motor immediately after you increase peak current. Be prepared to disable the drive if the system shows any signs of instability.

You can change the DIP switches with power applied—the drive will immediately sense the new current setting. For safety reasons, however, we recommend removing power from the drive before changing DIP switches.

⑦ Repeat Steps 4 – 6

With higher peak current, you can evaluate system performance and readjust tuning parameters if necessary. Continue repeating Steps 4 – 6 until you achieve satisfactory performance.

⑧ Choose Final Tuning Values

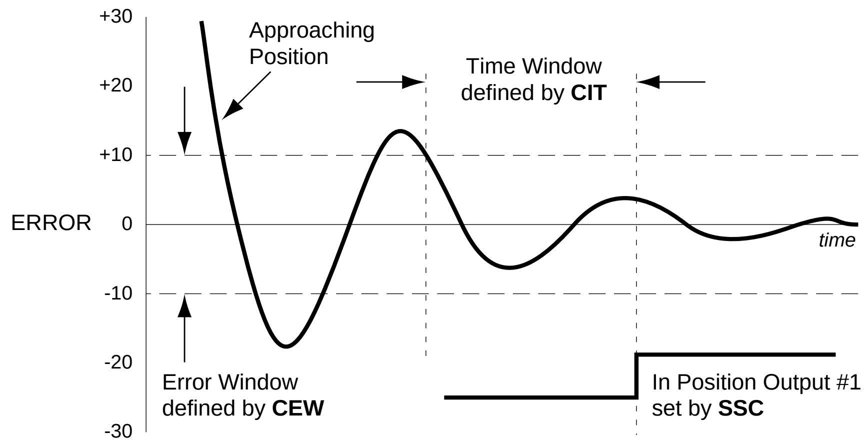
Program your applications's actual move, and select final tuning values based upon the performance you observe.

⑨ Determine Integral Gain (CIG) and Integral Limit (CIL) Values

Increasing integral gain **(CIG)** will reduce position errors at the end of the move and velocity errors during the constant velocity portion of the move. However, high values for **CIG** can cause the system to become unstable. If you increase **CIG** and your system becomes unstable, disable the drive immediately. In general, you should set **CIG** to the *lowest* value that will correct following errors and static position errors, but not increase overshoot or settling time. In a system without static torque loading, a **CIG** of zero may be appropriate.

Configuring an In Position Window

You can define an In Position Window, and use it to indicate that the preceding move is done. Two commands—**CEW** and **CIT**—determine the height and width of the window. A third command—**SSC**—can turn on output #1 when the In Position criteria are met.



As the drawing shows, **CEW** defines the position error window at the end of a move. **CIT** specifies the length of time the motor must be within the error window. The motor is In Position when three conditions are satisfied:

① The controller algorithm is finished (no input position command)

② Position error is less than that specified by the **CEW** command

③ Condition ② above has been true for the length of time specified by the **CIT** command

If **SSC** has been set to 1, output #1 will turn on when these three conditions have been met. You can use output #1 to trigger external hardware from the In Position condition. The output will stay on until the next move command is issued, such as **GO** or **GO HOME**.

(*Note:* If the motor is held (mechanically, or against an end stop), and **CPE** is greater than **CEW**, the motor may become "trapped" between **CPE** and **CEW**: it will not execute the next move. In this rare situation, two things are happening: 1.) **CPE** is not violated, and therefore no position error fault occurs; 2.) in position criteria are not met.

If you were to execute a **1R**, the response would be ***B**, which means the drive is "busy" waiting for the move to be over. Why doesn't the drive force the motor to finish the move? The motor is somehow held. To correct this situation, try touching the motor; this may complete the move, and the drive may execute the next move. Or, execute a **DPA** to read actual position, and verify that the move is not complete. You can also execute a **KILL** to reset the positions, and then do the next series of moves.)

CHAPTER FIVE

Software Commands

IN THIS CHAPTER

- Descriptions of each command for the TQ10X Controller/ Drive
- Descriptions of the function, range, default, and sample use of each command for the TQ10X Controller/Drive.

Command Descriptions

① A—Acceleration *(Example Command)*

② Command Type: Motion
③ Syntax: <a>An
④ Range: n = 0.01-999.99
⑤ Attributes: Buffered, Savable in Sequence
⑥ Valid Software Version: A
⑦ Units: revs/sec^2
⑧ Default Value: A = 100
⑨ See Also: D, G, MR, V
⑩ Response to aA is *An

① ***Command Mnemonic***
The beginning of each command entry contains the command's mnemonic value and full name.

② ***Command Type***
Set-Up—Set-up commands define application conditions. These commands establish the output data's format from the controller.

Motion—Motion commands affect motor motion, such as acceleration, velocity, distance, go home, stop, direction, mode, etc.

Programming—Programming commands affect programming and program flow for trigger, output, all sequence commands, time delays, pause and continue, enable and disable, loop and end-loop, line feed, carriage return, and backspace.

Status—Status commands respond (report back) with data. These commands instruct the system to send data out from the serial port for host computer use.

③ ***Syntax***
The proper syntax for the command is shown here. The specific parameters associated with the command are also shown. If any of these parameters are shown in brackets, such as <a>, they are optional. The parameters are described below.

a—An *a* indicates that a device address must accompany the command. Only the device specified by this parameter will receive and execute the command. Valid addresses are 1-255.

n—An *n* represents an integer. An integer may be used to specify a variety of values (acceleration, velocity, etc.).

s—An *s* indicates that a sign character, either positive or negative (+ or -), is required.

x—An *x* represents any character or string of characters.

④ ***Range***
This is the range of valid values that you can specify for n (or any other parameter specified).

⑤ ***Attributes***
This first attribute indicates if the command is *immediate* or *buffered.* The system executes immediate commands as soon as it receives them. Buffered commands are executed in the order that they are received with other buffered commands. Buffered commands can be stored in a sequence.

The second attribute explains how you can save the command.

- Savable in Sequence
- Never Saved
- Automatically Saved

Savable in Sequence commands are saved when they are defined in a sequence (see **XT** command). *Savable in Sequence* commands can be stored in system memory (nonvolatile) and retained when power is removed from the system. A command that is *Never Saved* is executed without being saved into the system's permanent memory . *Automatically Saved* commands are automatically saved into memory upon execution.

⑥ ***Valid Software Version***
This field contains the current revision of the software in which the command resides at the time this user guide was released.

⑦ ***Units***
This field describes what unit of measurement the parameter in the command syntax represents.

⑧ ***Default Value***
The default setting for the command is shown in this box. A command will perform its function with the default setting if you do not provide a value.

⑨ ***See Also***
Commands that are related or similar to the command described are listed here.

⑩ ***Response***
A sample status command and system response are shown. When the command has no response, this field is not shown.

A—Acceleration

- ❑ Command Type: Motion
- ❑ Syntax: <a>An
- ❑ Range: n = 0.01-9999.99
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: revs/sec^2
- ❑ Default Value: A = 100
- ❑ See Also: D, G, MR, V
- ❑ Response to aA is *An

The Acceleration command specifies the rotary acceleration rate to be used for the next Go (**G**) command. The acceleration remains set until you change it. You do not need to reissue this command for subsequent Go (**G**) commands. Accelerations outside the valid range cause the acceleration to remain at the previous valid **A** setting.

If the Acceleration command is entered with only a device address (**1A**), the controller will respond with the current acceleration value. If a move is commanded without specifying an acceleration rate, the previously commanded acceleration rate will be used. Acceleration cannot be changed on the fly. The minimum acceleration is

Min Accel = Encoder resolution (**ER** command) x .00465

Command	Description
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D2ØØØ	Sets distance to 2,000 encoder counts
G	Executes the move

B—Buffer Status

- ❏ Command Type: Status
- ❏ Syntax: aB
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aB is *B or *R
- ❏ See Also: BS

The buffer status command will report the status of the command buffer. If the command buffer is empty or less than 95% full, the controller will respond with a ***R**.

The command buffer is 512 bytes long. A ***B** response will be issued if less than 5% of the command buffer is free.

*R = More than 5% of the buffer is free

*B = Less than 5% of the buffer is free

This command is commonly used when a long series of commands will be loaded remotely via RS-232C interface. If the buffer size is exceeded, the extra commands will not be received by the controller until more than 5% of the command buffer is free.

Command	Response
1B	***B** (less than 5% of the command buffer is free)

BCDG—Buffered Configure Derivative Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>BCDGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 240
- ❏ Response to aBCDG is *CDGn
- ❏ See Also: BCIG, BCPG, BCTG, CIG, CPG, CTG

This buffered command is used for system tuning. This term represents the gain applied to the derivative of the position error—in other words, the rate at which the position error is changing. This gain produces a damping effect similar to velocity feedback.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
BCDG400	Set the derivative gain term to 400
1BCDG	Reports derivative gain term (*CDG400)

BCIG—Buffered Configure Integral Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>BCIGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 2
- ❏ Response to aBCIG is *CIGn
- ❏ See Also: BCDG, BCPG, BCIL BCTG, CDG, CPG, CTG, CIL

This buffered command is used for system tuning. This term represents the gain applied to the integral of the position error—the net accumulation of the position error over time. Thus integral gain will contribute when a position error is not being reduced over time, as may be caused by the effects of friction or gravity. This gain will improve overall accuracy but may increase settling time and, if excessive, may cause a low frequency oscillation around the commanded position.

Refer to *Chapter ④—Tuning* for more information.

Command	**Description**
BCIL40	Set the integral limit term to 40
BCIG10	Set the integral gain term to 10
1BCIG	Reports integral gain term (*CIG10)

BCIL—Buffered Configure Integral Limit

- Command Type: Set-up
- Syntax: <a>BCILn
- Range: n = 0-32,767
- Attributes: Buffered Savable in Sequence
- Valid Software Version: A
- Units: N/A
- Default Value: 2
- Response to aBCIL is *CILn
- See Also: BCDG, BCIG, BCPG, BCTG, CDG, CIG,CPG, CTG

This buffered command is used for system tuning. Since integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term **(Integral * CIG)**.

Refer to *Chapter ④—Tuning* for more information.

Command	**Description**
BCIG10	Set the integral gain term to 10
BCIL40	Set the integral limit to 40
1BCIL	Reports integral limit term (*CIL40)

BCPE—Buffered Configure Position Error

- Command Type: Set-up
- Syntax: <a>BCPEn
- Range: n = 0-32,767
- Attributes: Buffered Savable in Sequence
- Valid Software Version: A
- Units: N/A
- Default Value: 4000
- Response to aBCPE is *CPEn
- See Also: DPE, CPE

This buffered command defines the maximum allowable position or following error. If the actual position error ever exceeds the allowable position error, the TQ10X will generate a fault condition and shut down power output to the motor. If the maximum allowable position error is set to Ø, the function is disabled and no amount of position error will generate the fault condition.

Refer to *Chapter ④—Tuning* for more information.

Command	**Description**
BCPE400	Set the maximum allowable position error to 400 encoder counts
BCPE0	Disable fault generation due to position error
1BCPE	Reports maximum position error setting (*CPE0)

BCPG—Buffered Configure Proportional Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>BCPGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 16
- ❏ Response to aBCPG is *CPGn
- ❏ See Also: BCDG, BCIG, BCTG, CDG, CIG, CTG

This buffered command is used for system tuning. This term represents the gain applied directly to the position error. The proportional gain sets how actively the system will respond to position error. High proportional gain will give a stiff, responsive system, but may result in overshoot and oscillation.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
BCPG50	Set the proportional gain term to 50
1BCPG	Reports proportional gain term (*CPG50)

BCTG—Buffered Configure Derivative Sampling Period

- ❏ Command Type: Set-up
- ❏ Syntax: <a>BCTGn
- ❏ Range: n = 0-255
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: 266 µsec
- ❏ Default Value: 0
- ❏ Response to aBCTG is *CTGn
- ❏ See Also: BCDG, BCIG, BCPG, CDG, CIG, CPG

This buffered command is used for system tuning. Use **BCTG** to adjust the derivative sampling period. The *system* sampling period—266 µsec—is the period between updates of position error. The *derivative* sampling period is an integer multiple of the system sampling period. In general, a longer derivative sampling period gives a more constant derivative term and improves stability. Many systems require a low **BCTG** value to prevent oscillations, however. Therefore, start with a low value and increase it incrementally.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
BCTG0	Set the derivative sampling period to 266 µsec
BCTG1	Set the derivative sampling period to 532 µsec
BCTG2	Set the derivative sampling period to 798 µsec
BCTG3	Set the derivative sampling period to 1064 µsec
1BCTG	Reports derivative sampling period (*CTG3)

BS—Buffer Size Status

- ❏ Command Type: Status
- ❏ Syntax: aBS
- ❏ Range: N/A
- ❏ Attributes: Immediate,
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aBS is *n
- ❏ See Also: B

This command reports the number of bytes remaining in the command buffer. When entering long string commands, check the buffer status to be sure that there is enough room in the buffer. Otherwise, commands may be lost. Each character (including delimiters) uses one byte. The range for the response is 0 - 512 bytes.

Command	Response
1BS	***122** (122 bytes available in the buffer)

C—Continue

- ❏ Command Type: Motion
- ❏ Syntax: <a>C
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: PS, U

The Continue (**C**) command ends a pause state. It enables your controller to continue executing buffered commands. After you enter a Pause (**PS**) or the Pause and Wait for Continue (**U**) command, you can clear it with a Continue (**C**) command. This command is useful when you want to transmit a string of commands to the buffer before you actually execute them.

Command	Description
MC	Sets move to continuous mode
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
PS	Pauses system until controller receives **C** command
G	Accelerates the motor to 10 revs/sec
C	Continues executing commands in the buffer

The motor will not execute the **G** command until the **C** command is issued

CDG—Configure Derivative Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CDGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Immediate Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 240
- ❏ Response to aCDG is *CDGn
- ❏ See Also: CIG, CPG, CTG, RFS

This command is used for system tuning. This term represents the gain applied to the derivative of the position error, in other words, the rate at which the position error is changing. This gain produces a damping effect similar to velocity feedback.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CDG400	Set the derivative gain term to 400
1CDG	Reports derivative gain term (*CDG400)

CEW—Configure In Position Error Window

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CEWn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: encoder counts
- ❏ Default Value: 50
- ❏ Response to aCEW is *CEWn
- ❏ See Also: CIT, SSC

This command, together with the **CIT** command, can be used to configure an In Position window, which can be used to indicate that the preceding move has terminated.

The In Position condition is met when:

- The controller algorithm has finished (no input position command)
- The CEW condition is met (the position error is less than that specified by the **CEW** command).
- The above condition has been true for the length of time specified by the **CIT** command

The position error range, specified by *n* in **CEW*n***, is the maximum number of encoder counts allowed on either side of the desired position. For example, if *n* = 10, then the In Position window is 20 encoder counts wide.

Output 1 can be configured with the **SSC** command to show the state of the In Position detector. This allows the user to trigger external hardware from the In Position condition.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CEW10	Configure an In Position Error Window ±10 encoder counts either side of desired position
1CEW	Reports ± number of encoder counts (*CEW10)

CIG—Configure Integral Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CIGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Immediate
  Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 2
- ❏ Response to aCIG is *CIGn
- ❏ See Also: CIL, CDG, CPG, CTG, RFS

This command is used for system tuning. Integral gain provides a torque that is directly proportional to the sum, over time, of the error values—the *integral* of the error. The controller reads the error value every sampling period, and adds it to the sum of all previous error values. The sum is multiplied by the value set by the **CIG** command (Integral * CIG), resulting in the *integral term* which contributes to the motor control output every system sampling period. This gain will improve overall accuracy but may increase settling time and, if excessive, may cause a low frequency oscillation around the commanded position.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CIL40	Set the integral limit term to 40
CIG10	Set the integral gain term to 10
1CIG	Reports integral gain term (*CIG10)

CIL—Configure Integral Limit

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CILn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Immediate
  Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 2
- ❏ Response to aCIL is *CILn
- ❏ See Also: CIG,CDG, CPG, CTG, RFS

This command is used for system tuning. Since integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term **(Integral * CIG)**.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CIG10	Set the integral gain term to 10
CIL40	Set the integral limit to 40
1CIL	Reports integral limit term (*CIL40)

CIT—Configure In Position Time

- ❏ Command Type: Setup
- ❏ Syntax: <a>CITn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Milliseconds
- ❏ Default Value: 20
- ❏ Response to aCIT is *CITn
- ❏ See Also: CEW, SSC

This command is used to specify the time period that the servo is to be within the In Position window before the "In Position" signal is generated. The range is 0 to 32,767, and is the number of milliseconds to be used as the testing time frame.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
1SSC1	Set output 1 as "In Position"
1CIT30	Set "In Position" time to 30ms
1CEW20	Set allowable position error to ±20 encoder counts

CPE—Configure Position Error

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CPEn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Immediate
  Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 4000
- ❏ Response to aCPE is *CPEn
- ❏ See Also: DPE, RFS

This command defines the maximum allowable position or following error. If the actual position error ever exceeds the allowable position error, the TQ10X will generate a fault condition and shut down power output to the motor. If the maximum allowable position error is set to Ø, the function is disabled and no amount of position error will generate the fault condition.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CPE400	Set the maximum allowable position error to 400 encoder counts
CPE0	Disable fault generation due to position error
1CPE	Reports maximum position error setting (*CPE0)

CPG—Configure Proportional Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CPGn
- ❏ Range: n = 0-32,767
- ❏ Attributes: Immediate
  Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 16
- ❏ Response to aCPG is *CPGn
- ❏ See Also: CDG, CIG, CTG, RFS

This command is used for system tuning. This term represents the gain applied directly to the position error. The proportional gain sets how active the system will be to position error. High proportional gain will give a stiff, responsive system, but may result in overshoot and oscillation.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CPG50	Set the proportional gain term to 50
1CPG	Reports proportional gain term (*CPG50)

CR—Carriage Return

- ❏ Command Type: Programming
- ❏ Syntax: <a>CR
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: LF

The Carriage Return (**CR**) command determines when the controller has reached a particular point in the execution buffer. When the controller reaches this command in the buffer, it responds by issuing a carriage return (**ASCII 13**) over its interface back to the host computer or terminal. If you place the CR command after a Go (**G**) command, it indicates when a move is complete. If you place the CR command after a Trigger (**TR**) command, it indicates when the trigger condition is met.

You can use Carriage Return (**CR**) and Line Feed (**LF**) commands with the Quote (") command to display multiple-line messages via the RS-232C interface.

Command	Description
MN	Sets mode to preset mode
A5Ø	Sets acceleration to 50 revs/sec^2
V5	Sets Velocity to 5 revs/sec
D5ØØØ	Sets distance to 5,000 encoder counts
G	Executes the move (Go)
1CR	Sends a carriage return after move is completed

The motor moves 5,000 encoder counts. When the motor stops, the controller sends a carriage return over its interface.

CTG—Configure Derivative Sampling Period

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CTGn
- ❏ Range: n = 0-255
- ❏ Attributes: Immediate
  Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: 266 µsec
- ❏ Default Value: 0
- ❏ Response to aCTG is *CTGn
- ❏ See Also: CDG, CIG, CPG, RFS

This command is used for system tuning. Use **CTG** to adjust the derivative sampling period. The *system* sampling period—266 µsec—is the period between updates of position error. The *derivative* sampling period is an integer multiple of the system sampling period. In general, a longer derivative sampling period gives a more constant derivative term and improves stability. Many systems require a low **CTG** value to prevent oscillations, however. Therefore, start with a low value and increase it incrementally.

Refer to *Chapter ④—Tuning* for more information.

Command	Description
CTG0	Set the derivative sampling period to 266 µsec
CTG1	Set the derivative sampling period to 532 µsec
CTG2	Set the derivative sampling period to 798 µsec
CTG3	Set the derivative sampling period to 1064 µsec
1CTG	Reports derivative sampling period (*CTG3)

D—Distance

- ❏ Command Type: Motion
- ❏ Syntax: <a>Dn
- ❏ Range: n = ±1,073,741,823
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: encoder counts
- ❏ Default Value: 4000
- ❏ Response to aD is *Dn
- ❏ See Also: A, G, MN, MPA, MPI, V, H

The Distance (**D**) command defines either the number of encoder counts the motor will move or the absolute position it will seek after a Go (**G**) command is entered. In incremental mode (**MPI**), the value set with the Distance (**D**) command will be the distance (in encoder counts) the motor will travel on all subsequent Go (**G**) commands. In absolute mode (**MPA**), the distance moved by the motor will be the difference between the present motor position and the position (referenced to the zero position) set with the **D** command. In either mode, the direction is controlled by the direction (+ or -) that precedes the distance value. The **D** command has no effect on continuous moves (**MC**).

In Mode Normal (**MN**) the position may not exceed the maximum distance range of 1,073,741,823 encoder counts. If the motor approaches the absolute maximum (plus or minus), the controller will not execute any **GO** commands that would cause the distance to exceed the absolute maximum. To proceed further, use the **PZ** command to reset the absolute counter to zero, and then resume operations.

If **D** is entered with only a device address (**1D**), the controller will respond with the current distance value. If a move is commanded without specifying a distance, the previously commanded distance will be applied to the move.

Command	Description
MN	Sets controller to Normal mode
MPI	Sets controller to Incremental Position mode
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D4ØØØ	Sets distance to 4000 encoder counts
G	Executes the move

A servo motor with a 4000 count encoder will travel 1 rev (CW) after **G** is issued.

DPA—Display Position Actual

- ❏ Command Type: Status
- ❏ Syntax: aDPA
- ❏ Range: n = ±1,073,741,823
- ❏ Attributes: Immediate
  Device Specific, Never saved
- ❏ Valid Software Version: A
- ❏ Units: Encoder counts
- ❏ Default Value: NA
- ❏ Response to aDPA is *DPAn
- ❏ See Also: D, PZ

Single display of actual motor position as measured by the encoder. This command is functionally identical to the **PX** command. The response is the position in encoder counts as referenced to the last power reset or position zero command (**PZ**).

Command	Description
1DPA	Report the actual motor position of axis 1 (*+0004000000)

DPE—Display Position Error

- ❏ Command Type: Status
- ❏ Syntax: aDPE
- ❏ Range: n = ±2,147,483,646
- ❏ Attributes: Immediate
  Device Specific, Never saved
- ❏ Valid Software Version: A
- ❏ Units: Encoder counts
- ❏ Default Value: NA
- ❏ Response to aDPE is *DPEn
- ❏ See Also: D, DPA

Single display of position error. The response is the difference in encoder counts of the actual motor position and the commanded motor position. This information is used by the control algorithm to control the torque command to the motor. It is normal for the position error to be present during a move but large position errors are usually caused by inappropriate gain settings. Issuing a DPE command before the motor has settled into final position will also result in larger position errors.

Command	Description
1DPE	Report the position error of axis 1 (*+0000000005).

DVA—Display Velocity Actual

- ❏ Command Type: Status
- ❏ Syntax: aDVA
- ❏ Range: n = 1-20000
- ❏ Attributes: Immediate
  Device Specific, Never saved
- ❏ Valid Software Version: A
- ❏ Units: (rev/sec)*100
- ❏ Default Value: NA
- ❏ Response to aDVA is *DVAn
- ❏ See Also: DPA, V

Single display of actual motor velocity in revolutions per sec. There is an implied decimal point before the last two digits.

Command	Description
MC	Mode continuous
V2	Velocity of 2 rev/sec
G	Go
1DVA	Report the actual motor velocity of axis 1 (*+00200)

E—Enable Communications

- ❏ Command Type: Programming
- ❏ Syntax: <a>E
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: Enabled
- ❏ See Also: F

The Enable Communications (**E**) command allows the controller to accept commands over the serial communications interface. You can re-enable the communications interface with this command if you had previously disabled the RS-232C interface with the Disable Communications Interface (**F**) command. If several units are using the same communications interface, the **E** and **F** commands can help streamline programming.

Command	Description
F	Disables all units (axes) on the communications interface
1E	Enables serial interface on Device 1
4E	Enables serial interface on Device 4
A1Ø	Set acceleration to 10 revs/sec^2
V5	Set velocity to 5 revs/sec
D5ØØØ	Sets distance to 5000 encoder counts
G	Executes the move (Go—only axes 1 & 4 will move)

ER—Encoder Resolution

- ❏ Command Type: Set-up
- ❏ Syntax: <a>ERn
- ❏ Range: n = 400 - 65,532
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: n = encoder counts/rev
- ❏ Default Value: 4000
- ❏ Response to aER is *ERn
- ❏ See Also: CPE

The encoder resolution defines the number of encoder counts the controller will see per revolution of the motor. The number of lines on an encoder should be multiplied by 4 to arrive at the correct ER value per revolution of the motor. (In other words, one line of an encoder will produce 4 encoder counts due to quadrature detection)

Command	Description
ER4ØØØ	Sets encoder resolution to 4000 encoder counts per 1 motor revolution
1ER	Reports Encoder Resolution (*ER4000)

F—Disable Communications

- ❏ Command Type: Programming
- ❏ Syntax: <a>F
- ❏ Range: N/A
- ❏ Attributes: Immediate Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ See Also: E

The Disable Communications (**F**) command is useful when you are programming multiple units on a single interface. Axes that are not intended to process global commands should be disabled using device specific **F** commands. This allows you to program other units without specifying a device identifier on every command. If you do not disable other units in a daisy chain, uploading programs may cause other units on the daisy chain to perform uploaded commands.

Command	Description
1F	Disables the communications interface on unit #1
3F	Disables the communications interface on unit #3
G	All controllers (except 1 & 3) will execute a move

G—Go

- ❏ Command Type: Motion
- ❏ Syntax: <a>G
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ See Also: A, D, MC, MN, S, V

The Go (**G**) command instructs the motor to make a move using motion parameters that you have previously entered. You do not have to re-enter Acceleration (**A**), Velocity (**V**), Distance (**D**), or the current mode (**MN** or **MC**) commands with each **G** (if you do not need to change them).

In the Normal mode (**MN**), moves can be either incremental or absolute. In the Incremental Preset mode (**MPI**), a **G** will initiate the move distance you specified with the **D** command. A **G** command in the Absolute Preset mode (**MPA**) will not cause motion unless the distance (**D**) value is different from the present motor position (**PR**) .

In Continuous mode (**MC**), you only need to enter the Acceleration (**A**) and Velocity (**V**) commands prior to **G**. The system ignores the Distance (**D**) command in this mode.

No motor motion will occur until you enter **G** in either the Normal (**MN**) or Continuous (**MC**) modes. If motion does not occur with **G**, an activated end-of-travel limit switch may be on. Check the hard limit switches or use the limit disable command (**LD3**—see **RA** command also). The next buffered command will not be executed until after the move is completed.

Command	Description
MN	Sets Normal mode (preset)
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D2ØØØ	Sets distance to 2,000 encoder counts
G	Executes the move (Go)
A1	Sets acceleration to 1 rev/sec^2
G	Executes the move (Go)

Assuming the controller is in Incremental Preset mode, the motor turns 2,000 encoder counts and repeats the 2,000 count move using the new acceleration value of 1 rev/sec^2 (Total distance moved = 4,000 encoder counts).

GH—Go Home

- ❏ Command Type: Motion
- ❏ Syntax: <a>GHn
- ❏ Range: n = ± .01 - 200
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Revs/sec
- ❏ Default Value: n = 0
- ❏ See Also: OS, RC, V,IN

The Go Home (**GH**) command instructs the Controller to search for an absolute home position in the positive or negative (+ or -) direction. It defines home as the position where the home limit signal changes states. To use the GH command, one of the general purpose inputs (pins 1-5) must be configured as a home input (**IN** command).

Homing can be as simple as decelerating to a stop when the edge of the home limit is detected. By using the **OS** commands, the homing process can be tailored to meet the application needs.

OSB—*Back up to home* makes homing more repeatable by backing off the home switch and re-approaching at low speed. The final approach to home switch is always form the CW direction.

OSC—*Define active state of home* allows the use of a normally closed or normally open limit switch.

OSD—*Enable Z Channel for home* uses the Z channel of the encoder, in conjunction with a home switch, to determine the final home position. The Z channel is a more accurate home position than the edge of a switch.

OSH—*Reference edge of home switch* allows either edge of the home switch to be used as the final edge position.

The Controller will reverse direction if an end-of-travel limit is activated while searching for home. However, if a second end-of-travel limit is encountered in the new direction, the Go Home procedure will stop and the operation will be aborted. The Homing Status (**RC**) command will indicate if the homing operation was successful.

The Go Home command will use acceleration set by the A command. The Go Home velocity will not affect the standard velocity (**V**) value.

Command	Description
INE1	Configure input #1 as home input
OSB1	Back up the home switch
OSD1	Reference Z channel as final home
GH-2	The motor moves CCW at 2 revs/sec and looks for the Home Limit input to go active.

Since the motor is turning CCW, it will see the CW edge of the limit first. It will decelerate to a stop and turn at 0.1 rev/sec in the CW direction until it detects the Z channel.

^H—Delete

- ❏ Command Type: Programming
- ❏ Syntax: ^H
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None

This command allows you to delete the last character that you entered. The **^H** command will not prevent execution of an immediate command. A new character may be entered at that position to replace the existing character. (**^H** indicates that the Ctrl key is held down when the H key is pressed.) This command prompts the controller to backup one character in the command buffer, regardless of what appears on the terminal. On some terminals, the Ctrl and the left arrow (<—) keys produce the same character.

This command will *not* delete characters beyond the last delimiter issued. Pressing the delete key will not delete the previous character.

H—Set Direction

- ❑ Command Type: Programming
- ❑ Syntax: <a>H(s)
- ❑ Range: s = + or -
- ❑ Attributes: Buffered
  Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: +
- ❑ See Also: D

The Set Direction (**H**) command changes or defines the direction of the next move that the system will execute. This command does not affect moves already in progress.

H+ = Sets move to CW direction
H– = Sets move to CCW direction
H = Changes direction from the previous setting

In preset moves, a Distance (**D**) command entered after the **H** command overrides the direction set by the **H** command. In Continuous mode (**MC**), only the **H** command can set the direction of motion.

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
G	Executes the move (Go) in CW direction
H	Reverses direction
G	Executes the move (Go) in CCW direction
MC	Sets mode to continuous
H+	Sets direction to CW
G	Moves continuously in CW direction

IN—Set Input Function

- ❑ Command Type: Set-up
- ❑ Syntax: aINxn
- ❑ Range: n = 1 – 5, x = A – F
- ❑ Attributes: Buffered
  Savable in sequence
- ❑ Valid Software Version: A
- ❑ Units: NA
- ❑ Default Value: AAAAA
- ❑ Response to aIN is *xxxxx
- ❑ See Also: IS, TR, XP, #, K, GH

This command configures the function of each of the 5 general purpose inputs. You can configure each input to perform one of the following functions:

Function A—*Trigger Input*

Used with the **TR** command as a comparison input. The TR command defines active high, low, or “don't care.” Up to 5 inputs can be configured as trigger inputs.

Function B—*Sequence Select Input*

Executes predefined sequences from remote inputs based on the **XP** command. Active state (sequence selected) is high. Up to 3 inputs can be configured as sequence select inputs.

Function C—*Kill Input*

Immediately halts execution of the move. Same as kill (**K**) command. Active state (kill initiated) is high. Up to one input can be configured as a kill input.

Function D—*Stop Input*

Decelerates the motor to a stop using the value specified in the Acceleration (**A**) command (dumps the sequence or command buffer if configured by **SSHØ**). Same as Stop (**S**) command. Active state (stop initiated) is high. Up to one input can be configured as a stop input.

Function E—*Home Input*

Defines the motor home or origin position as executed by the **GH** command and configures with the Homing Function (**OS**) commands. **OSC** determines the active state of the input. Up to one input can be configured as a home input.

Function F—*Go Input*

Accelerates the motor to a velocity and distance specified in the Acceleration (**A**), Velocity (**V**), and Distance (**D**) commands. Same as Go (**G**) command. Active state (go initiated) is high. Up to one input can be configured as a go input.

Some of the functions (stop, kill, home, go) can only have one input configured to that function. If you try to configure another input to that function, the controller will not recognize the new function and revert back to the previous definition. For example, if input 1 is a kill function, and you want input 5 to be the kill function, you must first change input 1 to another function.

Function *B* (sequence select) can configure up to three inputs as sequence select inputs. If you try to configure more than three inputs, the controller will only recognize the first three. It will revert back to the previous definitions for the additional inputs.

Command	Description
1INA1	Configure input one as trigger input 1
1INB2	Configure input two as sequence select input 1
1INB3	Configure input three as sequence select input 2
1INC4	Configure input four as kill input
1INF5	Configure input five as go input
1IN	Reports input configuration (*ABBCF)

IS—Input Status

- ❑ Command Type: Status
- ❑ Syntax: aIS
- ❑ Range: N/A
- ❑ Attributes: Immediate Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: IN, LD, RSE
- ❑ Response to aIS is *nnnnnnnn

This command reports the status of all hardware inputs. The response is 8 ASCII digits (Ø or 1) corresponding to the following I/O bits:

1—IN1 (Ø = Low, 1 = High)
2—IN2 (Ø = Low, 1 = High)
3—IN3 (Ø = Low, 1 = High)
4—IN4 (Ø = Low, 1 = High)
5—IN5 (Ø = Low, 1 = High)
6—CW limit (Ø = Low, 1 = High)
7—CCW limit (Ø = Low, 1 = High)
8—Fault (Ø = Low, 1 = High)

This is not a software status. It will report the actual hardware status of the inputs. **IS** can help you troubleshoot an application, to verify that limit switches, trigger inputs and home switches work.

Command	Response
2IS	***ØØØ1ØØØ1** (The input status of device 2 is reported: I/O bits 1-3 and 5-7 are low (grounded), and I/O bits 4 (IN4), and 8 (Fault), are high)

K—Kill

- ❏ Command Type: Motion
- ❏ Syntax: <a>K
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: IN,S

This command causes commanded motion to cease immediately. There is *NO* deceleration of the motor. The motor will servo around the position where the kill was entered. The load could be driven past limit switches and cause damage to the mechanism and possibly to the operation. In addition to stopping the motor, the **K** command will terminate a loop, end a time delay, abort down-loading a sequence (**XD**), and clear the command buffer.

WARNING

The Kill (**K**) command is not an emergency stop. The motor is not disabled. Motion caused by instability or incorrect wiring will not be stopped. An emergency stop should cut power to the amplifier or interrupt the hardware enable input (pin 10), and mechanically prevent the motor from turning.

Command	Description
A5	Sets acceleration to 5 revs/sec^2
V2	Sets velocity to 2 revs/sec
MC	Sets mode to continuous
G	Executes the move (Go)
K	Stops the motor instantly

L—Loop

- ❏ Command Type: Programming
- ❏ Syntax: <a>Ln
- ❏ Range: n = 0 - 65,535
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: number of loops
- ❏ Default Value: None
- ❏ See Also: C, N, U, Y

When you combine the Loop (**L**) command with the End-of-Loop (**N**) command, all of the commands between **L** and **N** will be repeated the number of times indicated by n. If you enter **L** without a value specified for n, or with a Ø, subsequent commands will be repeated continuously. If you specify a value greater than 65,535, the loop will be repeated continuously.

The **N** command prompts the controller to proceed with further commands after the designated number of loops have been executed. The **Y** command stops loop execution after completing the current loop cycle. The Immediate Pause (**U**) command allows you to temporarily halt loop execution after completing the current loop cycle. You can use the Continue (**C**) command to resume loop execution.

Command	Description
L5	Loop 5 times
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D1ØØØØ	Sets distance to 10,000 encoder counts
G	Executes the move (Go)
N	End of loop

The commands in the loop will be executed 5 times.

LD—Limit Disable

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>LDn
- ❏ Range: n = 0 - 3
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: RA

The Limit Disable (**LD**) command allows you to enable/disable the end-of-travel limit switch protection. The **LDØ** condition does not allow the motor to turn without properly installing the limit inputs. If you want motion without wiring the limits, you must issue **LD3**. For machine and operator safety, hardware limits are highly recommended.

- Enable CCW and CW limits—***n = Ø (Default)***
- Disable CCW limit—n = 2
- Disable CCW and CW limits—n = 3

Command	Description
1LDØ	Enables CW and CCW limits. The motor will move only if the limit inputs are bypassed or connected to normally-closed limit switches.
1LD3	Allows you to make any move, regardless of the limit input state.

LF—Line Feed

- ❏ Command Type: Programming
- ❏ Syntax: <a>LF
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: CR

When you issue the Line Feed (**LF**) command, the system transmits a line feed character over the communications link. When the controller reaches this command in the buffer, it responds by issuing a line feed (ASCII 10) over its interface back to the host computer. If you place the **LF** command after a Go (**G**) command, it indicates when a move is complete. If you place the **LF** command after a Trigger (**TR**) command, it indicates when the trigger condition is met.

You can use the Carriage Return (**CR**) and **LF** commands with the Quote (**"**) command to display multiple-line messages via the RS-232C interface.

Command	Description
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D15ØØØ	Sets distance to 15,000 encoder counts
G	Executes the move (Go)
1LF	Transmits a line feed character over the communications interface after the move is completed

MC—Mode Continuous

- ❏ Command Type: Motion
- ❏ Syntax: <a>MC
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Inactive
- ❏ See Also: MN, T, TR, V

The Mode Continuous (**MC**) command causes subsequent moves to ignore any distance parameter and move continuously. You can clear the **MC** command with the Move Normal (**MN**) command.

The controller uses the previously defined Acceleration (**A**) and Velocity (**V**) commands to reach continuous velocity. Using the Time Delay (**T**), Trigger (**TR**), and Velocity (**V**) commands, you can achieve basic velocity profiling. After a new parameter is entered a Go (**G**) command is required. Acceleration (**A**) cannot be changed on the fly.

Command	Description
MC	Sets mode to continuous
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
G	Executes the move (Go)
T1Ø	Move at 5 revs/sec for 10 seconds
V7	Set velocity to 7 revs/sec
G	Change velocity to 7 revs/sec
T1Ø	Move at 7 revs/sec for 10 seconds
VØ	Set velocity to 0 rev/sec (stop)
G	Executes the **VØ** command

The motor turns at 5 revs/sec for 10 seconds, then moves at 7 revs/sec for 10 seconds before decelerating to a stop.

MN—Mode Normal

- ❏ Command Type: Motion
- ❏ Syntax: <a>MN
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Active
- ❏ See Also: A, D, G, MC, MPA, MPI

The Mode Normal (**MN**) command sets the positioning mode to preset. In Mode Normal, the motor will move the distance specified with the last distance (**D**) command. To define the complete move profile, you must define Acceleration (**A**), Velocity (**V**), and the Distance (**D**). The **MN** command is used to change the mode of operation from Mode Continuous (**MC**) back to normal or preset. To use the **MPA** or **MPI** command, you must be in Mode Normal (**MN**).

Command	Description
MN	Set positioning mode to preset
A5	Set acceleration to 5 revs/sec^2
V5	Set velocity to 5 revs/sec
D1ØØØ	Set distance to 1,000 encoder counts
G	Executes the move (Go)

Motor turns 1,000 encoder counts CW after the **G** command is issued.

MPA—Mode Position Absolute

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>MPA
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Inactive
- ❏ See Also: D, MN, MPI, PZ

This command sets the positioning mode to absolute. In this mode all move distances are referenced to absolute zero. In Mode Position Absolute (**MPA**), giving two consecutive Go (**G**) commands will cause the motor to move only once, since the motor will have achieved its desired absolute position at the end of the first move.

MPA is most useful in applications that require moves to specific locations while keeping track of the beginning position.

You can set the absolute counter to zero by cycling power or issuing a Position Zero (**PZ**) command. You must be in Normal mode (**MN**) to use this command. In continuous mode (**MC**), **MPA** is ignored.

Command	Description
MN	Sets Normal mode (preset)
PZ	Resets absolute counter to zero
MPA	Sets position mode absolute
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D4ØØØØ	Sets destination to absolute position 40,000
G	Motor will move to absolute position 40,000
D1ØØØØ	Sets destination to absolute position +10,000
G	Motor will move to absolute position +10,000

The motor will move 40,000 encoder counts in the CW direction (if starting from position Ø) and then move 30,000 encoder counts in the CCW direction to reach the absolute position 10,000.

MPI—Mode Position Incremental

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>MPI
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Active
- ❏ See Also: D, MN, MPA

This command sets the positioning mode to incremental. In incremental mode all move distances specified with the Distance (**D**) command will be referenced to the current position. Mode Position Incremental (**MPI**) is most useful in applications that require repetitive movements, such as feed to length applications.

You must be in normal mode (**MN**) to use this command. In continuous mode (**MC**), this command is ignored.

Command	Description
MN	Set positioning mode normal (preset)
MPI	Set positioning mode incremental
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D1ØØØØ	Sets distance of move to 10,000 encoder counts
G	Move 10,000 encoder counts CW
G	Move another 10,000 encoder counts CW

The motor moves 10,000 encoder counts CW after each **G** command (total move is 20,000 encoder counts).

N—End of Loop

- ❏ Command Type: Programming
- ❏ Syntax: <a>N
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: C, L, PS, U

This command marks the end of a loop. You must use this command in conjunction with the Loop (**L**) command. All buffered commands that you enter between the **L** and **N** commands are executed as many times as the number that you enter following the **L**

Command	Description
MN	Sets move to Normal mode
A5	Sets acceleration to 5 $revs/sec^2$
V5	Sets velocity to 5 revs/sec
D1ØØØØ	Sets move distance to 10,000 encoder counts
L5	Loops the following commands five times
G	Executes the move (Go)
N	Ends the loop

The move will be executed five times.

OFF—De-Energize Drive

- ❏ Command Type: Programming
- ❏ Syntax: <a>OFF
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: ST, ON

The OFF command immediately disables the drive. This command can be used to shut down the drive in an emergency. This command is functionally identical to the ST1 command.

Command	Description
OFF	De-energize the drive
1IS	*00000001 (fault bit active)

ON—Energize Drive

- ❏ Command Type: Programming
- ❏ Syntax: <a>OFF
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: ST, OFF

The ON command immediately re-enables the drive. This command is used to re-enable the drive after a commanded shutdown or after a fault condition such as excessive position error. This command is functionally identical to the ST0 command.

For details on the enable output see *Chapter ② Installation*.

Command	Description
ON	Energize the drive
1IS	*00000000 (fault bit inactive)

O—Output

- ❏ Command Type: Programming
- ❏ Syntax: <a>Onn
- ❏ Range: Ø, 1 or X (See Below)
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: on, off, or unchanged
- ❏ Default Value: ØØ
- ❏ See Also: OS, S, TR

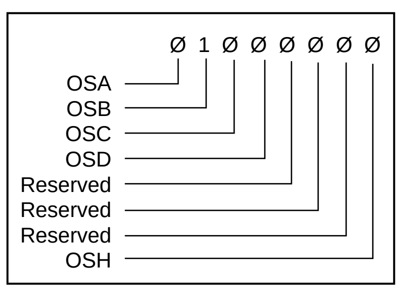
The Output (**O**) command turns the programmable output bits on and off. This is used for signaling remote controllers, turning on LEDs, or sounding whistles. The output can indicate that the motor is in position, about to begin its move, or is at constant velocity, etc.
n=1 = Turns output bits on
n=Ø = Turns output bits off
n=X = Leaves output bits unchanged

Command	Description
MN	Set to Mode Normal
A1Ø	Set acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D2ØØØØ	Set move distance to 20,000 encoder counts
OØ1	Set programmable output 1 off and output 2 on
G	Executes the move (Go)
OØØ	After the move ends, turn off both outputs

OS—Report Homing Function Set-Ups

- ❏ Command Type: Status
- ❏ Syntax: <a>OS
- ❏ Range: N/A
- ❏ Attributes: Buffered,
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: OS(A-H)
- ❏ Response to aOS is nnnnnnnn

This command results in a report of which software switches have been set by **OS** commands. The reply is eight digits. This command reports **OSA** through **OSH** Set-up status in binary format. The digit 1 represents ON (enabled), the digit Ø represents OFF (disabled). The default response is ***Ø1ØØØØØØ**.



OSA—Define Active State of End-of-Travel Limits

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSAn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: NA
- ❏ Default Value: Ø
- ❏ See Also: LD, OSC

OSAØ: Normally Closed Contacts
OSA1: Normally Open Contacts

This command sets the active state of the CW and CCW end-of-travel limit inputs. It enables you to use either normally closed or normally open switches.

Command	Description
OSA1	Sets active state for normally open limit switches
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

OSB—Back Up To Home

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSBn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: 1
- ❏ See Also: GH, OSC, OSD, OSH

OSBØ: Back up to home
OSB1: Back up to selected edge

This command is used to make homing more repeatable. With Back Up to Selected Home (**OSB**) command enabled, homing is a two step process. First it approaches the home switch at high speed (set by the GH command) until it sees the switch. Then it decelerates and returns to the switch at slow speed. Since it always makes its final approach from the same direction, the system will act differently whether the active edge of the switch is the first or second edge encountered.

If the selected edge for final home position is the first edge encountered the motor will decelerate to 0 velocity, when that edge is detected. The motor will then reverse direction and stop on the selected edge.

If the selected edge for the final home position is the second edge encountered the motor will travel until that edge is detected. The motor will decelerate to a 0 velocity. The controller will then position the motor 1/2 of a revolution on the outside of the selected edge. Finally the motor will creep at 0.1 rev/sec in the direction of the active home region, until home is detected.

With **OSB** disabled, the motor will decelerate to 0 velocity after encountering the active home region, and will be considered to be at home if the home limit input is still active. If the deceleration overshoots the active home region the motor will reverse direction and travel back to the active home region.

Command	Description
OSB1	Sets back up to home switch active
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

OSC—Define Active State of Home Switch

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSCn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: NA
- ❏ Default Value: Ø
- ❏ See Also: GH, OSB, OSD, OSH

OSCØ: Active state of home input is n = Ø (closed)
OSC1: Active state of home input is n=1 (open)

OSCØ requires that a normally open (high) switch be connected to the home limit input.
OSC1 requires that a normally closed (low) switch be connected to the home limit input.

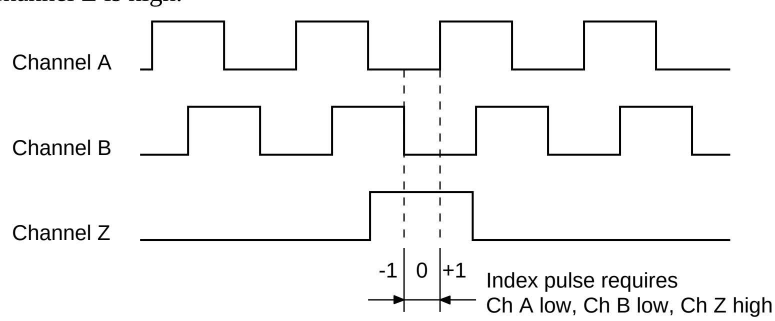
Command	Description
OSC1	Sets the active state of the home input to open

OSD—Enable Encoder Z Channel for Home

- ❏ Command Type: Set-up
- ❏ Syntax: <a>OSDn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: OSB,OSC,OSH,GH

OSDØ = Do not reference Z Channel during homing
OSD1 = Reference Z Channel during homing

The encoder Z channel is used (in conjunction with a load activated switch connected to the home limit) to determine the home position. The switch determines the home region, and the Z channel determines the exact and final home position inside the home region. As the next drawing shows, the final home position occurs when Channel A is low, Channel B is low, and Channel Z is high.



For OSD1 to be selected, OSB1 must also be selected.

Command	Description
OSD1	Recognizes Z channel as final home reference

OSH—Reference Edge of Home Switch

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSHn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: NA
- ❏ Default Value: Ø
- ❏ See Also: GH, OSB, OSC, OSD

OSHØ: Selects the CW side of the Home signal as the edge on which the final approach will stop
OSH1: Selects the CCW side of the home signal as the edge on which the final approach will stop

The CW edge of the Home switch is the first switch transition seen by the controller when traveling from the CW limit in the CCW direction. If n = 1, the CCW edge of the Home switch will be referenced as the Home position. The CCW edge of the Home switch is the first switch transition seen by the controller when traveling from the CCW limit in the CW direction. If n = Ø, the CW edge of the Home switch will be referenced as the Home position.

Command	Description
OSB1	Sets back up to home switch active
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

The home limit becomes active when the home limit input is closed. The controller recognizes the CCW edge of the switch as the home limit and backs up to that edge to complete the Go Home move.

PR—Absolute Position Report

- ❏ Command Type: Status
- ❏ Syntax: aPR
- ❏ Range: ±1,073,741,820
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Encoder counts
- ❏ Default Value: N/A
- ❏ See Also: D, MPA, MPI, MN, PZ, PX
- ❏ Response to aPR is *±nnnnnnnnnn

This command reports the commanded motor position relative to the power-up position. When a **D** command is issued, the distance is relative to the value of **PR**. The difference between the commanded position (**PR**) and the actual encoder position (**PX**) is the position error (**DPE**). The controller is always trying to minimize the position error. You can reset the encoder position counter to zero by using the position zero (**PZ**) command or reset (**Z**). After **PZ** the encoder position (**PX**) will be set to zero. If there was a position error before the **PZ** was issued, the value of **PR** will differ from the value of **PX** by the amount of the position error. Increasing integral gain (**CIG**) can help reduce the position error at rest thus insuring the value of **PR** equals the value of **PX**.

Command	Description
1PR	Commanded position report. (*+ØØØØØØ2ØØØ)
1PX	Encoder position report (*+ØØØØØØ2ØØ5)

The actual motor position is 5 encoder counts from the commanded position.

PS—Pause

- ❏ Command Type: Programming
- ❏ Syntax: <a>PS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: C, U

This command pauses execution of a command string or sequence until the controller receives a Continue (**C**) command. **PS** lets you enter a complete command string before running other commands. **PS** is also useful for interactive tests and synchronizing multiple controllers that have long command strings.

Command	Description
PS	Pauses execution of commands until the controller receives the Continue (C) command
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets move distance to 4,000 encoder counts
G	Executes the move (Go)
T2	Delays the move for 2 seconds
G	Executes the move (Go)
C	Continues Execution

When the controller receives the **C** command, the motor moves 4,000 encoder counts twice with a 2 second delay between moves.

PX—Report Absolute Encoder Position

- ❏ Command Type: Status
- ❏ Syntax: aPX
- ❏ Range: ±1,073,741,820
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Encoder counts
- ❏ Default Value: N/A
- ❏ See Also: PR, PZ
- ❏ Response to aPX *±nnnnnnnnn

This command reports the actual motor position as measured by the encoder. When a **D** command is issued, the distance is relative to the value of **PR** not the value of **PX**. The difference between the commanded position (**PR**) and the actual encoder position (**PX**) is the position error (**DPE**). The controller is always trying to minimize the position error. You can reset the encoder position counter to zero by using the position zero (**PZ**) command or reset (**Z**). After **PZ** the encoder position (**PX**) will be set to zero. If there was a position error before the **PZ** was issued, the value of **PR** will differ from the value of **PX** by the amount of the position error. Increasing integral gain (**CIG**) can help reduce the position error at rest thus insuring the value of **PR** equals the value of **PX**.

Command	Description
MN	Set to mode normal
PZ	Sets the absolute counter to zero
A1Ø	Sets acceleration to 10 rev/sec2
V5	Sets velocity to 5 rev/sec
D46ØØ	Sets move distance to 4,600 encoder counts
G	Executes the move (Go)
1PX	After the motor executes the move, the encoder position is reported: The response is *+0000004600.

PZ—Set Absolute Counter to Zero

- ❏ Command Type: Programming
- ❏ Syntax: <a>PZ
- ❏ Range: N/A
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: D, MN, PR, PX

This command sets the absolute encoder position counter to zero. If there was a position error before the **PZ** was issued, the new value of **PR** will differ from the value of **PX** by the amount of the position error. Absolute counter will also be set to zero when you cycle power or when you successfully execute a homing (**GH**) function.

Command	Description
MN	Enter position mode (mode normal)
MPA	Makes preset moves from absolute zero position
PZ	Sets absolute position counter to zero
A1Ø	Sets acceleration to 10 rev/sec^2
V5	Sets velocity to 5 rev/sec
D4ØØØØ	Sets move distance to 40000 encoder counts
G	Executes the move (Go)
1PX	Reports absolute encoder position (*+0000040000)
PZ	Sets the absolute counter to zero
1PX	Reports absolute encoder position (*+0000000000)

"—Quote

- ❏ Command Type: Programming
- ❏ Syntax: "x
- ❏ Range: x = up to 17 ASCII characters
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: CR, LF
- ❏ Response to "x is x

Up to 17 characters entered after the quotation marks (**"**) will be transmitted, exactly as they are entered, over the RS-232C link. A space entered by the space bar indicates the end of the command. A space is always sent after the last character in the string. This command is used during buffered moves or sequences to command other devices to move, or to send the message to a remote display. On a daisy chain of multiple units, if more than one unit is reporting a message at once, the messages will overlap and be garbled.

Command	Description
PS	Pause execution until Continue (C) is entered
A5	Set acceleration to 5 revs/sec^2
V5	Set velocity to 5 revs/sec
D2ØØØ	Set distance to 2,000 encoder counts
G	Executes the move (Go)
"MOVE_DONE	Transmits message
C	Continue move

After the move, the TQ10X will send the message MOVE_DONE via the RS-232C port

QØ—Exit Velocity Profiling Mode

- Command Type: Set-Up
- Syntax: <a>QØ
- Range: N/A
- Attributes: Immediate, Never Saved
- Valid Software Version: A
- Units: N/A
- Default Value: N/A
- See Also: Q1, RM

The **QØ** command exits the Velocity Profiling mode. The motor will stop when **QØ** is issued. Entering this command will cause the TQ10X to enter Normal mode (**MN**).

Q1—Enter Velocity Profiling Mode

- Command Type: Set-Up
- Syntax: <a>Q1
- Range: N/A
- Attributes: Immediate, Never Saved
- Valid Software Version: A
- Units: N/A
- Default Value: N/A
- See Also: QØ, RM

Q1 activates Velocity Profiling mode. Subsequent **RM** commands will immediately change motor velocity. **QØ** exits this mode.

Command	Description
ER2ØØØ	Set encoder resolution to 2000
Q1	Enter Velocity Streaming mode
RMØØØØ22ØC	Accelerate to 0.25 revs/sec^2
RMØØØØ4418	Accelerate to 0.5 revs/sec^2
RMØØØØ8831	Accelerate to 1 revs/sec^2
RMØØØ11Ø62	Accelerate to 2 revs/sec^2
RMØØØØ8831	Decelerate to 1 revs/sec^2
RMØØØØ4418	Decelerate to 0.5 revs/sec^2
RMØØØØ22ØC	Decelerate to 0.25 revs/sec^2
RMØØØØØØØØ	Decelerate to 0 revs/sec^2
QØ	Exit Velocity Streaming mode

R—Request Controller Status

- ❏ Command Type: Status
- ❏ Syntax: aR
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: RA, RB, RC, XSR, XSS
- ❏ Response to aR is *x

The Request Controller Status (**R**) command can be used to indicate the general status of the controller. Possible responses are:

Character	Definition
***R**	Ready
***S**	Ready, Attention Needed
***B**	Busy
***C**	Busy, Attention Needed

When the controller is not prepared to accept another command, the following conditions will cause a controller is busy (***B**) response:

* Performing a move
* Accelerating/decelerating during a continuous move
* A time delay is in progress. (T command)
* In RM mode
* Paused
* Waiting on a Trigger
* Going Home
* In Power-on sequence mode
* Running a sequence
* Executing a loop

The following conditions will cause an error (***S** or ***C**) response:

* Go home failed
* Limit has been encountered
* Sequence execution was unsuccessful
* Sequence memory checksum error
* Undervoltage
* Drive recently enabled

When the response indicates that attention is required, the **RA**, **RB**, **RC**, **XSR**, or **XSS** commands can provide details about the error.

It is not recommended that this command be used in tight polling loops that could result in microprocessor overload. Time delays can alleviate this problem.

This command is not intended to be used to determine if a move is complete. It should be used after a move is complete to determine if errors or faults exist. Use a buffered status request (**CR** or **LF**) command or a programmable output to indicate move completion.

Command	Response
R	***R** (Controller ready to accept a command, and no error conditions exist.)

RA—Limit Switch Status Request

- ❑ Command Type: Status
- ❑ Syntax: aRA
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: R, RB
- ❑ Response to aRA is *x

The **RA** command responds with the status of the end-of-travel limits during the last move as well as the present condition. This is done by responding with one of 12 characters representing the conditions listed below.

Response Character	***Last Move Terminated By CW Limit***	***—CCW Limit***	***Current Limit Status CW Limit***	***—CCW Limit***
*@	No	No	Off	Off
*A	Yes	No	Off	Off
*B	No	Yes	Off	Off
*D	No	No	On	Off
*E	Yes	No	On	Off
*F	No	Yes	On	Off
*H	No	No	Off	On
*I	Yes	No	Off	On
*J	No	Yes	Off	On
*L	No	No	On	On
*M	Yes	No	On	On
*N	No	Yes	On	On

The **RA** command is useful when the motor will not move in either or both directions. The report back will indicate if the last move was terminated by one or both end-of-travel limits. This command is not intended to be used to determine if a move is complete. It should be used after a move to determine if errors or faults exist. If you are hitting a limit switch, the Ready Status (**R**) will return a ***S**.

Command	**Response**
1RA	***@** (the last move was not terminated by a limit and no limits are currently active.)

RB—Loop, Pause, Shutdown, Trigger Status Request

- ❑ Command Type: Status
- ❑ Syntax: aRB
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A\
- ❑ See Also: L, PS, R, RA, ST, TR
- ❑ Response to aRB is *x

This command receives a response from ***@** to ***O**, as defined below. The four conditions for which status is indicated are as follows:

Loop Active: A loop is in progress.

Pause Active: Buffered commands waiting for a Continue (**C**).

Shutdown Active: The motor is shutdown by the **ST1** command.

Trigger Active: At least one trigger is active.

Response Character	***Loop Active***	***Pause Active***	***Shutdown Active***	***Trigger Active***
*@	No	No	No	No
*A	Yes	No	No	No
*B	No	Yes	No	No
*C	Yes	Yes	No	No
*D	No	No	Yes	No
*E	Yes	No	Yes	No
*H	No	No	No	Yes
*I	Yes	No	No	Yes
*J	No	Yes	No	Yes
*K	Yes	Yes	No	Yes
*L	No	No	Yes	Yes
*M	Yes	No	Yes	Yes
*N	No	Yes	Yes	Yes
*O	Yes	Yes	Yes	Yes

This command is not intended to be used to determine if a move is complete. It should be used after the move is complete to determine if errors or faults exist.

Command	Response
1RB	***A** (After issuing a 1RB command, the response came back as *A. This means that the controller is currently executing a loop.)

RC—Homing Status Request

- ❑ Command Type: Status
- ❑ Syntax: aRC
- ❑ Range: N/A
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ Response to aRC IS *x
- ❑ See Also: R, RA, RB, FS, GH

The RC command has the same response format of RA and RB. The condition for which status is indicated is:

Homing Function Failure:

In this condition, the controller has encountered both End-of-Travel limits or one of several possible Stop commands or conditions. Go Home motion was concluded, but not at Home.

Response Character	Go Home Successful?
*@	YES
*B	NO

Command	Description
1RC	*B Go home was unsuccessful.

RFS—Return Servo Gains to Factory Settings

- ❑ Command Type: Status
- ❑ Syntax: aRFS
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: RA, RB, RC, XSR, XSS
- ❑ Response to aRFS is *x

This command can be used to return all the servo gains to the factory defaults immediately. This command is useful when the current servo gain values are inappropriate and re-tuning is required. By returning to factory defaults the gains will be in proper relationship to one another allowing easy fine tuning of the system. The factory defaults are:

Proportional Gain	(**CPG**)	16
Integral Gain	(**CIG**)	2
Derivative Gain	(**CDG**)	240
Derivative Sampling Period	(**CTG**)	0
Position Error	(**CPE**)	4000
Integral Limit	(**CIL**)	2

Command	Description
1RFS	All servo gains returned to the factory defaults

RM—Rate Multiplier in Velocity Streaming

- ❑ Command Type: Motion
- ❑ Syntax: <a>RMn
- ❑ Range: n = Ø - FFFFFFFF
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: revs/sec
- ❑ Default Value: None
- ❑ See Also: D, H, QØ, Q1

The **RM** command sets an immediate velocity where n represents an 8-digit hexadecimal value. The value for n is determined with the following formula:

*(desired revs/sec) • (encoder resolution) * 17.432576* = **decimal # for velocity value to be rounded off to the closest whole number.**

The resulting decimal number must be converted to a hexadecimal number to obtain the value for n.

The velocity change is instant—there is no acceleration/deceleration ramp between velocities. A limit switch closure will stop movement in Velocity Profiling mode, but does not cause the TQ10X to exit Velocity Streaming mode. To recover from a limit stop in **RM** mode, **QØ** must be issued and the direction must be changed. Velocity Profiling mode is uni-directional. The last direction set either from a move or from a Distance (**D**) or Direction (**H**) command will be used. Bi-directional moves can be made in this mode by returning to velocity zero (Ø), turning **RM** mode off, changing the direction, and re-enabling **RM** mode. Exiting **RM** mode with **QØ** causes the TQ10X to enter Normal mode (**MN**).

Command	Response
ER2ØØØ	Set encoder resolution to 2000
Q1	Enter Velocity Streaming mode
RMØØØØ22ØC	Accelerate to 0.25 revs/sec^2
RMØØØØ4418	Accelerate to 0.5 revs/sec^2
RMØØØØ8831	Accelerate to 1 revs/sec^2
RMØØØ11Ø62	Accelerate to 2 revs/sec^2
RMØØØØ8831	Decelerate to 1 revs/sec^2
RMØØØØ4418	Decelerate to 0.5 revs/sec^2
RMØØØØ22ØC	Decelerate to 0.25 revs/sec^2
RMØØØØØØØØ	Decelerate to 0 revs/sec^2
QØ	Exit Velocity Streaming mode

RSE—Report Servo Errors

- ❑ Command Type: Status
- ❑ Syntax: aRSE
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: R,RA, RB, RC
- ❑ Response to aRSE is *x

The Report Servo Errors (**RSE**) command can be used to indicate the general status of the controller. Possible responses are:

Character	Definition
***0**	No errors
***2**	Excessive position error
***4**	Drive fault
***6**	Commanded shutdown
***8**	Undervoltage, or drive was recently enabled

During a fault condition, the RSE command can be used to interrogate the controller for the reason of the fault. The following can cause a latched fault condition—the red *NOT ENABLED* LED will be illuminated, and the fault output will be active:

-Drive commanded shutdown via the **OFF** or **ST1** commands

-Position error exceeded the value set in the **CPE** command
-Drive fault (overvoltage, overtemperature, etc.)

The next two conditions are not latched, and do not illuminate the red LED. They will cause a ***8** response to **RSE**, until the next move command is issued.
-Undervoltage
-Drive was recently enabled

Command	Response
1RSE	***0** (Controller ready to accept a command, and no error conditions exist.)

RV—Revision Level

- ❏ Command Type: Status
- ❏ Syntax: aRV
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: D, H, QØ, Q1
- ❏ Response to aRV is nn-nnnnnn-nnnn

The Revision (**RV**) command responds with the software part number and its revision level. The response is in the form shown below:
92-nnnn-nn**<xn***>[cr]
(**part number**, ***revision level***)

The part number identifies which product the software is written for, as well as any special features that the software may include. The revision level identifies when the software was written. You may want to record this information in your own records for future use. This type of information is useful when you consult Parker Compumotor's Applications Department.

Command	Response
1RV	**92-Ø16637-Ø1A**

The product is identified by *92-Ø16637-Ø1A, and the revision level is identified by A.

S—Stop

- ❏ Command Type: Motion
- ❏ Syntax: <a>S
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: A, K, QØ, SSH, ST

This command decelerates the motor to a stop using the last defined Acceleration (**A**) command. This command clears the command buffer (at the end of a move, if one is in progress). The Sequence Definition (**XD**) command is aborted and a time delay is terminated. If **SSH1** is set the controller will stop the current move but it will not clear the command buffer.

Command	Description
MC	Sets move in continuous mode
A1	Sets acceleration to 1 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
G	Executes the move (Go)
S	Stops motor (motor comes to a stop at a deceleration rate of 1 revs/sec^2)

The **S** command is not buffered. As soon as the controller receives the **S** command, it stops motion.

SN—Scan

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SNn
- ❏ Range: 1 - 1000
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: n = mS
- ❏ Default Value: 50
- ❏ See Also: XP

The Scan (**SN**) command allows you to define the *debounce time* (in milliseconds) for external sequence selection inputs. The debounce time is the amount of time that the sequence inputs must remain constant for a proper reading from a remote controller, such as a programmable logic controller (PLC). If you are using a PLC you should change the debounce time to match the *on time* of the PLC outputs.

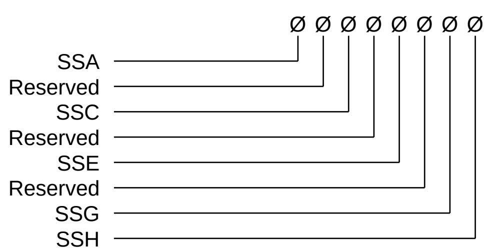
This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the TQ10X may respond to an electrical glitch. If you issue the Scan command with only a device address (**1SN**), the controller will respond with the current debounce time (***SNn**).

Command	Description
SN1Ø	Sets scan time of sequence select inputs to 10 ms

SS—Software Switch Function Status

- ❏ Command Type: Status
- ❏ Syntax: aSS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: SSA, SSC, SSG, SSH
- ❏ Response to aSS is *nnnnnnnn

This command reports the status of the **SS** commands. From left to right, the 8-character response corresponds to **SSA** through **SSH**.



SSA—RS-232C Echo Control

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSAn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable In Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also:

This command turns the RS-232C echo (transmission of characters received from the remote device by the TQ10X) on and off.

SSAØ = Echo on
SSA1 = Echo off

In the Echo On (**SSAØ**) mode, characters that are received by the controller are echoed automatically. In the Echo Off (**SSA1**) mode, characters are not echoed from the TQ10X. This command is useful if your computer cannot handle echoes. In a daisy chain, you must have the echo on (**SSAØ**) to allow controllers further down the chain to receive commands. *Status commands do not echo the command sent, but transmit the requested status report.*

Command	Description
SSA1	Turns echo off (Characters sent to the controller are not echoed back to the host.)

SSC—Output #1 on In Position

- ❏ Command Type: Set-up
- ❏ Syntax: aSSCn
- ❏ Range: n = 0 or 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: CEW, CIT
- ❏ Response to aSSC is *SSCn

With **SSC** set to 1, output 1 will turn on when the motor is within the In Position window for the specified time defined by the **CEW** and **CIT** commands

b = 0: Normal

b = 1: Output 1 is configured as an "In Position" output

Command	Response
SSC1	Set output 1 as an In Position output
1SSC	Reports output 1 configuration (*SSC1)

SSE—Enable/Disable Communication Error Checking

- ❏ Command Type: Communication interface
- ❏ Syntax: <a>SSEn
- ❏ Range: 0 (disable), 1 (enable)
- ❏ Attributes: Buffered Savable In Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: %

The command setting determines whether or not each byte received at the controller is checked for communication errors. SSE1 enables error checking for all bytes received at the controller, and SSE0 disables error checking. See the % command for the types of errors detected.

SSG—Clear/Save the Command Buffer on Limit

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSGn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable In Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: LD

SSGØ = Clears command buffer on limit
SSG1 = Saves command buffer on limit

In most cases, it is desirable that upon activating an end-of-travel limit input all motion should cease until the problem causing the over-travel is rectified. This will be assured if all commands pending execution in the command buffer are cleared when hitting a limit. This is the case if **SSGØ** is specified. If **SSG1** is specified and a limit is activated, the current move is aborted, but the remaining commands in the buffer continue to be executed.

Command	Description
SSG1	Saves buffer on limit
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
G	Executes the move (Go)
O11	Turn on outputs 1 and 2

If a limit switch is encountered while executing the move, outputs 1 and 2 will still go on.

SSH—Clear/Save Command Buffer on Stop

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSHn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable In Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: S

SSHØ = Clears command buffer on stop
SSH1 = Saves command buffer on stop

In Normal Operation (**SSHØ**) the Stop (**S**) command or a dedicated stop input will cause any commands in the command buffer to be cleared. If you select the Save Command Buffer On Stop (**SSH1**) command, a Stop (**S**) command will only stop execution of a move in progress. It will not stop execution of any commands that remain in the buffer. However, it will terminate a loop in the current pass.

Command	Description
SSHØ	Clears buffer on stop
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
L5Ø	Loops 50 times
G	Executes the move (Go)
T.5	Pauses the motor 500 ms
N	Ends Loop
S	Stops motion

When **S** is issued, the controller will clear the buffer and stop the move.

ST—Shutdown

- ❏ Command Type: Programming
- ❏ Syntax: <a>STn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See also: OFF, ON

The ST1 command immediately disables the drive. This command can be used to shut down the drive in an emergency. This command is functionally identical to the OFF command.

The ST0 command immediately re-enables the drive. This command is used to re-enable the drive after a commanded shutdown or after a fault condition such as excessive position error. This command is functionally identical to the ON command.

For details on the enable output see *Chapter ② Installation*.

Command	Description
ST1	Disable the drive
ST0	Enable the drive

T—Time Delay

- ❏ Command Type: Programming
- ❏ Syntax: <a>Tn
- ❏ Range: n = 0.01 - 99999.99
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: seconds
- ❏ Default Value: None

The Time (**T**) command causes the controller to wait the number of seconds that you specify before it executes the next command in the buffer. This command is useful whenever you need to delay the motor's actions or when you wish to move the motor in continuous velocity for preset time.

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
T1Ø	Pauses motor movement 10 seconds
G	Executes the move (Go)
T5	Pauses the motor for 5 seconds after the move
G	Executes the move (Go)

TR—Wait For Trigger

- ❏ Command Type: Programming
- ❏ Syntax: <a>TRnnnnn
- ❏ Range: n = Ø, 1, or X
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: None
- ❏ See Also: IN

This command allows you to specify a trigger configuration to be matched before continuing execution of the move, where *nnnnn* corresponds to triggers 1, 2, 3, 4 and 5 respectively. The possible values for *n* are as follows:

n = 1 Wait for the trigger input to be high (opened)

n = Ø Wait for the trigger input to be low (grounded)

n = X Ignore the trigger input

The lowest numbered input will be the first trigger. For example, if input 3,4 and 5 are configured with the IN command as triggers, they will be trigger 1,2 and 3 respectively.

Command	Description
IN3A	Configure input 3 as trigger input 1
IN5A	Configure input 5 as trigger input 2
TR1Ø	Wait for input 1 to be opened and input 2 to be grounded before going on to the next command
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
G	Executes the move (Go)

Motion will not occur until trigger conditions are true.

U—Pause and Wait for Continue

- ❏ Command Type: Programming
- ❏ Syntax: <a>U
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: C, PS

This command causes the indexer to complete the move in progress, then wait until it receives a Continue (**C**) to resume processing. Since the buffer is saved, the controller continues to execute the program (at the point where it was interrupted). The controller continues processing when it receives the **C** command. This command is typically used to stop a machine while it is unattended.

Command	Description
MN	Sets move to Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
LØ	Loops indefinitely
D46ØØ	Sets distance to 4,600 encoder counts
G	Executes the move (G)
T1Ø	Waits 10 seconds after the move
N	Ends loop
U	Halts execution until the controller receives the Continue command **(C)**

This command string pauses when the **U** command is entered. A **C** command resumes execution where it was paused. In this example, the loop stops at the end of a move, and resumes when the controller receives the **C** command. In reaction to the **T1Ø** command in the loop, there may be a 10 second delay before motion resumes after the **C** is executed, depending on when the **U** command is completed.

V—Velocity

- ❏ Command Type: Motion
- ❏ Syntax: <a>Vn
- ❏ Range: n = 0.01 - 200.00
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: revs/sec
- ❏ Default Value: 1
- ❏ See Also: A, D, G, GH, MR

The **V** command defines the maximum speed at which the motor will run when given the Go (**G**) command. The maximum encoder frequency the controller can accept is 960 kHz. In preset Mode Normal (**MN**), the maximum velocity may be limited when the resulting move profile is triangular. In Mode Continuous (**MC**), when a Go (**G**) command is issued the controller moves to the next command in the buffer.

Once you define the velocity, that value will be valid until you define another velocity, cycle DC power, or issue a **Z** (Reset) command.

*If the value specified for the **V** command is not valid, the TQ10X ignores that value and defaults to the value specified in the last **V** command. If **V** is issued with only a device address (**1V**), the controller will respond with the current velocity value (***Vn**).*

Command	Description
MC	Sets move to continuous
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
G	Go (Begin motion)

XC—Sequence Checksum

- ❏ Command Type: Status
- ❏ Syntax: aXC
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ See Also: XD, XE

XC computes the BBRAM checksum. After the unit is programmed, the response can be used for system error checking. The three-decimal response (ØØØ - 255) is followed by a [cr]. The response does not indicate the number of bytes programmed. This response is designed to be used for comparison. As long as the TQ10X is not re-programmed, the checksum response should always be the same.

Command	Response
1XC	*149

XD—Sequence Definition

- ❏ Command Type: Programming
- ❏ Syntax: <a>XDn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: None
- ❏ See Also: XE, XR, XRP, XT

This command begins sequence definition. All commands between the **XD** command and Sequence Termination (**XT**) command are defined as a sequence. The sequences will automatically be defined when **XT** is issued. If a sequence you are trying to define already exists, you must erase that sequence before defining it using the Erase Sequence (**XE**) command. A sequence cannot be longer than 256 characters. Immediate commands cannot be entered into a sequence.

Command	Description
XE1	Erases sequence #1
XD1	Defines sequence #1
MN	Sets to Normal mode
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets acceleration to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 encoder counts
G	Executes the move (Go)
XT	Ends definition of Sequence #1
XR1	Executes Sequence #1

XE—Sequence Erase

- ❑ Command Type: Programming
- ❑ Syntax: <a>XEn
- ❑ Range: n = 1 - 7
- ❑ Attributes: Buffered, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: Sequences
- ❑ Default Value: None
- ❑ See Also: XD, XR, XRP, XT

This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue the command. *Compumotor recommends that you delete a sequence before re-defining it.*

Command	Description
XE1	Deletes Sequence 1
XD1	Defines Sequence 1

XP—Set Power-up Sequence Mode

- ❑ Command Type: Set-Up
- ❑ Syntax: <a>XPn
- ❑ Range: n = 0 - 9
- ❑ Attributes: Buffered, Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: Sequences
- ❑ Default Value: 0
- ❑ See Also: IN, XQ, XSP, XSR

This co mmand executes a single sequence or multiple sequences on power-up. If *n* = 1-7, the sequence whose value = *n* will be executed on power up. Control will then be passed to the RS-232C interface.

If *n* = 8, the sequence whose number appears on the sequence select inputs (configured with the **IN** command) will be executed on power-up. Control will then be passed to the RS-232C interface.

If *n* = 9, the sequence whose number appears on the Sequence Select inputs (configured with the **IN** command) will be executed on power-up. When the first sequence is finished in **XP9** mode, the TQ10X will scan the Sequence Select inputs again and execute the next sequence. This cycle will continue until a Stop (**S**) or Kill (**K**) command is issued, a limit is encountered, an enable input is asserted, or the unit is powered down. The possible settings for this command are:

n = Ø: No sequence is executed on power-up
n =1-7: Sequence 1 - 7 is executed on power-up
n = 8: Sequence select inputs are read (single run) on power-up
n = 9: Sequence select inputs are read (continuous run) on power-up

In **XP9** mode, you can use the **XQ1 command** to stop the TQ10X from selecting the next sequence until all the sequence select inputs are first opened.

Command	Description
XP1	Executes Sequence #1 on power-up
XE1	Erases Sequence #1
XD1	Defines Sequence #1
LD3	Disables CW & CCW limits
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
G	Executes the move (Go)
XT	Ends definition of Sequence #1
Z	Resets the controller

The motor moves 4,000 encoder counts during power-up (with -M2 option only) or reset (Z).

XQ—Sequence Interrupted Run Mode

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>XQn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: Ø
- ❏ See Also: XP

n = 1: Interrupted Run mode is set (on)
n = Ø: Interrupted Run mode is reset (off)

This command can be used only when the TQ10X is stand-alone power-up sequencing in **XP9** mode. In **XP9** mode, if **XQ1** is executed, the TQ10X will not accept a sequence select input until all sequence select inputs are OFF (closed). After all lines have simultaneously been brought to a low state (OFF), the controller will then read the sequence select lines and execute the sequence whose number appears there. This paused mode will continue until an **XQØ** command is executed. You may use **S** or **K** command to stop sequence execution. **XQ1** must be the first command entered in the sequence.

Command	Description
XE1	Erases sequence #1
XD1	Defines sequence #1
XQ1	Turns Interrupted Run mode on
LD3	Disables CW & CCW limits
XT	Ends Sequence #1
XP9	Sets power-up sequences as sequence select inputs
Z	Resets the controller to start sequence scanning

If you execute Sequence #1 during power up by setting the sequence select inputs (configured with the **IN** command) inputs properly, Interrupted Run mode will be set.

XR—Run a Sequence

- ❏ Command Type: Programming
- ❏ Syntax: <a>XRn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequence
- ❏ Default Value: 0
- ❏ See Also: XD, XE, XRP, XT

This command loads a pre-defined sequence into the command buffer (clears the buffer first) and executes these commands as a normal set of commands. **XR** automatically recalls the sequence from BBRAM. Before executing the specified sequence, the TQ10X is put into the power-on default state with respect to the following conditions:

- The values for distance, velocity, and acceleration are zeroed.
- Move mode is Normal; position mode is Relative (incremental).
- The direction is set to CW.

XR can be used within one sequence to start execution of another sequence; however, all commands in the first sequence following **XR** will be ignored (in this respect an XR acts like a GOTO not a GOSUB). An **XR** command placed within a loop will be ignored.

Command	Description
XE1	Erases sequence #1
XD1	Defines sequence #1
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets acceleration to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 encoder counts
G	Executes the move (Go)
XT	Ends Sequence #1 definition
XR1	Executes Sequence #1

Sequence #1 is defined (XD1) and executed (**XR1**).

XRP—Sequence Run With Pause

- ❏ Command Type: Programming
- ❏ Syntax: <a>XRPn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequence
- ❏ Default Value: 0
- ❏ See Also: XD, XE, XR, XT

This command is identical to the Sequence Run (**XR**) command, except that it automatically generates a pause condition. You must clear this condition with the Continue (**C**) command before the controller executes the command buffer. The pause condition is invoked only if the sequence is valid. This allows you to execute a sequence without the delay of buffering that sequence.

Command	Description
XE5	Erases Sequence #5
XD5	Defines Sequence #5
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 encoder counts
G	Executes the move (Go)
XT	Ends definition of Sequence #5
XRP5	Runs Sequence #5 with a pause
C	Indexer executes Sequence #5

Upon issuing XRP5, Sequence #5 is entered in the command buffer, but is not executed. Issue a C command to execute Sequence #5.

XSD—Sequence Status Definition

- ❏ Command Type: Programming
- ❏ Syntax: aXSD
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XD, XE, XT
- ❏ Response to aXSD is *n

This command reports the status of the previous sequence definition (**XD**...**XT**). The response is 0 - 2. The valid values and descriptions of possible responses are shown below:

n = Ø: Download O.K.

n = 1: A sequence already exists with the number you have specified.

n = 2: Out of memory. The sequence buffer is full.

XSD verifies that the last sequence definition was successful.

Command	Response
1XSD	***1** (A sequence already exists as sequence 1)

XSP—Sequence Status Power-up

- ❏ Command Type: Status
- ❏ Syntax: aXSP
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XP, XQ, XSR
- ❏ Response to aXSP is *n

The Sequence Status Power-up (**XSP**) determines which, if any, sequence will be executed on power-up. After setting a power-up sequence using the Sequence Power-up (**XP**) command, you can check to make sure that proper sequence will be executed on power-up with **XSP**. The command reports which sequence the system will execute during power-up. The range of sequences is Ø - 9.

Command	Description
1XSP	***3** (Indicates that sequence #3. If it exists, will be executed upon power-up or reset.)

XSR—Sequence Status Run

- ❏ Command Type: Status
- ❏ Syntax: aXSR
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XR, XRP
- ❏ Response to aXSR is *n

This command allows you to check whether or not the last sequence issued was executed successfully without hitting limits, Stop (**S**), or Kill (**K**). The valid values and responses are shown below.

- **Ø** = Last sequence was successful
- **1** = In a loop
- **2** = Valid sequence
- **3** = Erased
- **4** = Bad checksum
- **5** = Running
- **6** = Killed, stopped, sequence never estimated

Command	Response
1XSR	***Ø** (Sequence ran O.K.)

XSS—Sequence Status

- ❏ Command Type: Status
- ❏ Syntax: aXSSn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: None
- ❏ See Also: XD, XE, XT
- ❏ Response to aXSSn is *x

XSS reports whether the sequence specified by n (representing one of the sequences 1 - 7) is empty, has bad checksum, or is OK.

Ø = Empty
1 = Bad Checksum
3 = O.K.

XSS verifies the existence of sequences and if that portion of memory has been corrupted.

Command	Response
1XSS1	***Ø** (Sequence #1 of device 1 is not defined.)

XT—Sequence Termination

- ❏ Command Type: Programming
- ❏ Syntax: <a>XT
- ❏ Range: N/A
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XD, XE, XR, XRP

XT is a sequence terminator. This command flags the end of the sequence currently being defined. Sequence definition is not complete until this command is issued. Properly defined sequences are saved into BBRAM automatically by issuing this command.

NOTE: In your communication program, use sufficient time delays after downloading a sequence before you send more commands to the TQ10X. In particular, after sending the **XT** command, wait at least 12.5 msec before sending a **Z** command.

Command	Description
XE1	Erases Sequence #1
XD1	Defines Sequence #1
MN	Sets to Normal mode
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
G	Executes the move (Go)
XT	Ends sequence definition

XU—Upload Sequence

- ❏ Command Type: Status
- ❏ Syntax: aXUn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: N/A
- ❏ See Also: F, XD, XE, XT
- ❏ Response to aXUn is contents of sequence n

This command sends the contents of sequence n to the host computer via the RS-232C interface, terminated by a carriage return [cr]. The contents of that sequence will appear on the computer CRT. All command delimiters in the sequence will be shown as spaces (2ØH). Any device identifiers that were included in the original sequence will also be eliminated (they are not stored in the sequence).

When using a daisy-chain, **XU** must be used cautiously as the contents of the sequence will go to all controllers in the loop between the controller that is uploading and the host. The **F** command may be used to turn off communication on units you are not uploading from.

Command	Description
2F	Turns off communication to unit #2
3F	Turns off communication to unit #3
1XU1	Uploads sequence #1 from unit #1

Y—Stop Loop

- ❏ Command Type: Programming
- ❏ Syntax: <a>Y
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: L, N

The Stop Loop (**Y**) command takes you out of a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the controller processes reach the last command of the current loop. At that time, the controller executes the command that follows the End Loop (**N**) command. You cannot restart the command loop unless you enter the entire command structure, including the Loop (**L**) and End Loop (**N**) commands.

Command	Description
L	Loops indefinitely
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
T2	Waits 2 seconds
G	Executes the move (Go)
N	Ends loop
Y	Stops loop. The loop requires the motor to move 4,000 encoder counts CW and then wait for 2 seconds. The loop terminates at the end of the loop cycle it is executing when it receives the **Y** command.

Z—Reset

- ❏ Command Type: Programming
- ❏ Syntax: <a>Z
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: K, S

The Reset (**Z**) command is equivalent to cycling DC power to the controller. This command returns all internal settings to their power-up values. It clears the command buffer. Like the Kill (**K**) command, the **Z** command immediately stops output pulses to the motor.

When you use the **Z** command, the controller is busy for 1,000 ms and ignores all commands. This command sets all position counters to zero and returns all values except the **XP** command to factory defaults.

Command	Description
1Z	Resets controller with address 1

#—Address Numbering

- ❏ Command Type: Set-up
- ❏ Syntax: <a>#n
- ❏ Range: n=1-255
- ❏ Attributes: Immediate, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: Address number
- ❏ Default Value: 1
- ❏ See Also: E,F

This command sets the individual unit address for each TQ10X, allowing addresses up to 255. Upon receipt of the command, the TQ10X will assign itself the address in the command and will pass on the daisy chain the address *plus one*, thus enabling automatic addressing of all units on the daisy chain. The address may also be set individually if preferred.

#1 - Automatic addressing of all units
Response - #(number of units plus one)

If the unit addresses exceed 255 , then the response will be #?. A <CR> or LF must be used with this command.

Command	Description
#1	#5 (for a daisy chain of 4 units, the units will assign themselves addresses 1 through 4 and return #5 as confirmation).

%—Reset Communication

- ❑ Command Type: Set-Up
- ❑ Syntax: %
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: E,F,SSE

When a communication error is detected, all external commands are ignored by echoing an **&** for each byte received from the host. This command is used to re-establish communication, and to identify the cause of the communication error.

In a daisy-chained environment, units located downstream from the unit detecting a communication error, will also disable external command processing. Units upstream in a daisy chain, are not affected.

(Note: Error detection will only occur if **SSE1** is enabled. Detection of a communication error has no effect on internal command processing, or sequence execution. A communication error will not stop motion.)

Possible responses are:

Character	Definition
*****	No errors
***0**	Unit upstream (daisy chained)
***1**	Overrun, data received to fast
***2**	Framing error

Command	Response
%	*2 (Either host or controller has lost synchronization.)

(Note: For daisy chained environments, the response values are in reverse order)

%	*0*0*0*1***** (First 5 Units report no error, 6th unit detected an overrun error, and the last 3 units turned communication off because of unit 6)

Summary of Commands

A—Acceleration
B—Buffer Status
BCDG—Buffered Configure Derivative Gain
BCIG—Buffered Configure Integral Gain
BCIL—Buffered Configure Integral Limit
BCPE—Buffered Configure Position Error
BCPG—Buffered Configure Proportional Gain
BCTG—Buffered Configure Derivative Sampling Period
BS—Buffer Size Status
C—Continue
CDG—Configure Derivative Gain
CEW—Configure In Position Error Window
CIG—Configure Integral Gain
CIL—Configure Integral Limit
CIT—Configure In Position Time
CPE—Configure Maximum Position Error
CPG—Configure Proportional Gain
CR—Carriage Return
CTG—Configure Filter Time Constant
D—Distance
DPA—Display Position Actual
DPE—Display Position Error
DVA—Display Velocity Actual
E—Enable Communications
ER—Encoder Resolution
F—Disable Communications
G—Go
GH—Go Home
^H—Delete
H—Set Direction
IN—Set Input Functions
IS —Input Status
K—Kill
L—Loop
LD—Limit Disable
LF—Line Feed
MC—Mode Continuous
MN—Mode Normal
MPA—Mode Position Absolute
MPI—Mode Position Incremental
N—End of Loop
O—Output
OFF—Servo Disable
ON—Servo Enable
OS—Report Homing Function Set-Ups
OSA—Define Active State of End-of-Travel Limits
OSB—Back Up To Home
OSC—Define Active State of Home Switch
OSD—Enable Encoder Z Channel for Home
OSH—Reference Edge of Home Switch
PR—Position Report
PS—Pause
PX—Report Absolute Encoder Position
PZ—Set Absolute Counter to Zero
“—Quote
Q1—Enter Velocity Profiling Mode
QØ—Exit Velocity Profiling Mode
R—Request Controller Status
RA—Limit Switch Status Report
RB—Loop, Pause, Shutdown, Trigger Status Report
RC—Homing Status Report
RFS—Return Servo Gains to Factory Settings
RM—Rate Multiplier in Velocity Streaming Mode
RSE—Report Servo Erros
RV—Revision Level
S—Stop
SN—Scan
SS—Software Switch Function Status
SSA—RS-232C Echo Control
SSC—Output #1 on In Position
SSE—Enable/Disable Communication Error Checking
SSG—Clear/Save the Command Buffer on Limit
SSH—Clear/Save Command Buffer on Stop
ST—Shutdown
T—Time Delay
TR—Wait For Trigger
U—Pause and Wait for Continue
V—Velocity
XC—Sequence Checksum
XD—Sequence Definition
XE—Sequence Erase
XP—Set Power-up Sequence Mode
XQ—Sequence Interrupted Run Mode
XR—Run a Sequence
XRP—Sequence Run With Pause
XSD—Sequence Status Definition
XSP—Sequence Status Power-up
XSR—Sequence Status Run
XSS—Sequence Status
XT—Sequence Termination
XU—Upload Sequence
Y—Stop Loop
Z—Reset
#—Address Numbering
%—Reset Communication

CHAPTER SIX

Troubleshooting

IN THIS CHAPTER

- Diagnostic LEDs
- RS-232C Problems
- Software Debugging Tips
- Encoder Problems
- Homing Diagrams
- Non-Drive Related Problems
- Protective Circuits
- Product Return Procedure

Troubleshooting Basics

When your system does not function properly (or as you expect it to operate), the first thing that you must do is identify and isolate the problem. When you have accomplished this, you can effectively begin to resolve the problem.

The first step is to isolate each system component and ensure that each component functions properly when it is run independently. You may have to dismantle your system and put it back together piece by piece to detect the problem. If you have additional units available, you may want to exchange them with existing components in your system to help identify the source of the problem.

Determine if the problem is mechanical, electrical, or software-related. Can you repeat or recreate the problem? Random events may appear to be related, but they are not necessarily contributing factors to your problem.

You may be experiencing more than one problem. You must isolate and solve one problem at a time. Log (document) all testing and problem isolation procedures. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

Once you have isolated a problem, take the necessary steps to resolve it. Refer to the problem solutions contained in this chapter. If the problem persists, contact your local technical support resource.

Diagnostic LEDs

The TQ10X Drive has five LEDs on its front panel. The following summary of LED functions may help you isolate problems.

LED Name	Color	Function
MOTOR FAULT	Red	Indicates short circuit in motor or cabling; or, Indicates motor overtemperature; or, Indicates Hall miswiring, or damaged Hall sensor; or, Indicates foldback with fault-on-foldback wet.
DRIVE OVERTEMP	Red	Indicates drive has exceeded temperature limit
REGEN/	Green	Illuminates green during regen event;
OVERVOLTAGE	Red	Illuminates red if regen causes overvoltage
PEAK CURRENT/	Green	Illuminates green during peak current output;
IN FOLDBACK	Red	Illuminates red while drive is in foldback
POWER ON/	Green	Illuminates green when AC power is applied;
NOT ENABLED	Red	Illuminates red when AC power is applied but drive is not enabled

Reducing Electrical Noise

For detailed information on reducing electrical noise in your system, refer to the current Compumotor Catalog.

RS-232C Problems

Use the following procedure to troubleshoot communication problems that you may have with the TQ10X.

1. Be sure the host computer's transmit (Tx) wire is wired to the TQ10X's receive (Rx) connection, and the host computer's receive (Rx) wire is wired to the TQ10X's transmit (Tx) connection. Switch the receive and transmit wires on either the host or TQ10X if the problem persists.
2. Confirm that the host and peripheral are configured for the same baud rate, 8 data bits, 1 stop bit, and no parity.
3. Use signal ground as a reference, not earth ground.
4. Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to connect the cable shield to earth ground at one end only.
5. To test the terminal or terminal emulation software and the RS-232C cable for proper three-wire communication, unhook the TQ10X and enter a character. You should not receive an echoed character. If you do, you are in half duplex mode; you should switch to full duplex mode. Connect the host's transmit and receive lines together and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin outs.
6. (Note: This only applies to firmware 92-016637-01 Rev A and higher.) If the controller echoes back **&** for each byte sent to the controller, a data communication error has occurred. To re-establish communication, see the **%** command.
7. (Note: This only applies to firmware 92-016637-01 Rev A and higher.) To extend cable length and/or improve noise immunity, RS-485 option is available as a custom product request. Contact Custom Products department for details.

Software Debugging Tips

This section offers helpful tips for debugging programs and understanding errors and fault conditions. The TQ10X has several tools that you can use to debug a problem in the system. The software tools are listed below:

R—Report Status
RA—Limit Switch Status Report
RB—Loop, Pause, Shutdown, Trigger Status report
RSE—Report Servo Errors
IS—Input Status Report
BS—Buffer Status Report
B—Buffer Status Report

The troubleshooting table on the next page also offers possible causes for typical symptoms.

Encoder Problems

Since the TQ10X relies on feedback information, the encoder connections are critical for the unit to operate properly. If you suspect the controller is not receiving good position data, use the following procedure to verify.

1. Disable the TQ10X (**ST1** or **OFF**)
2. Enter a **PZ** command
3. Rotate the motor clockwise by hand approximately one revolution
4. Enter a **PX**. It should read approximately 4000 (for a 1000 line encoder)
5. Rotate counterclockwise one revolution.
6. Enter **PX** command. It should read approximately 0.

If your TQ10X did not respond with similar values, it is not receiving encoder information. Either the encoder is mis-wired or in need of repair.

Homing Diagrams

The following diagrams are examples of the many possible homing set-ups. Your parameters may vary and the results may vary slightly depending on your settings.

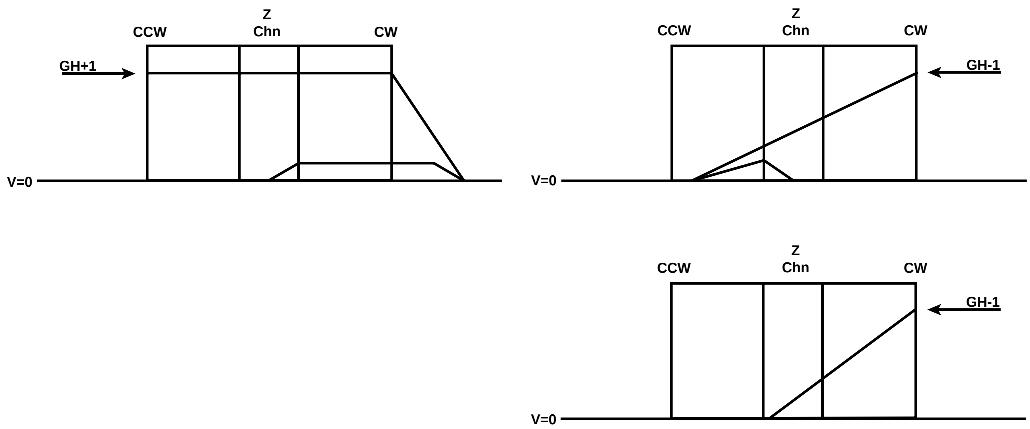
The CW side of the home pulse is the side closest to the CW limit. The CCW side of the home pulse is the side closest to the CCW limit.

The long pulse diagrams are indicative of situations where the motor decelerates while remaining inside the home pulse width due to the rapid homing deceleration or a very wide home pulse. The short pulse diagrams are indicative of situations where the motor decelerates through the home pulse width due to slow deceleration or a very narrow pulse width.

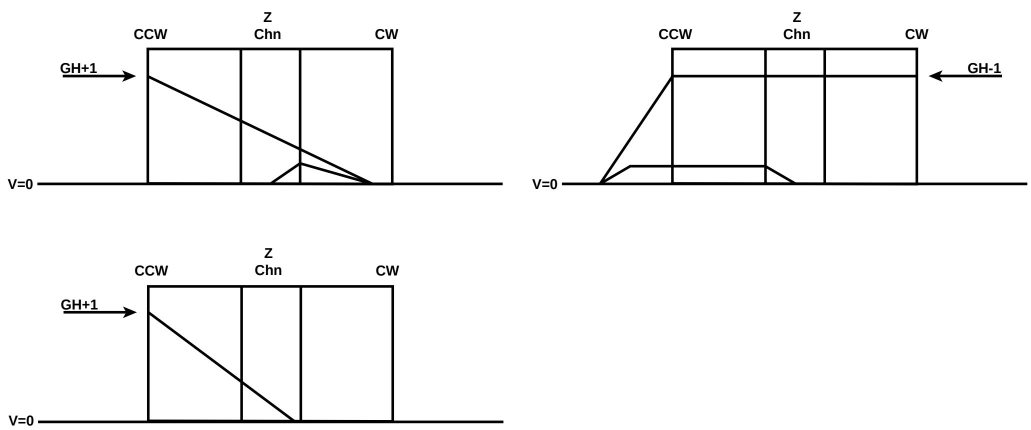
If an end-of-travel limit is hit during the initial homing, refer to the homing diagram for the opposite direction of travel.

The diagrams are drawn as a general guide. Velocity levels and slopes are drawn to indicate the general move profile the motor will make during the go home move. The vertical axis is velocity and the horizontal axis the position in relation to the home input transitions. Some lines are drawn as closely as possible together to indicate identical velocities, yet remain discernible.

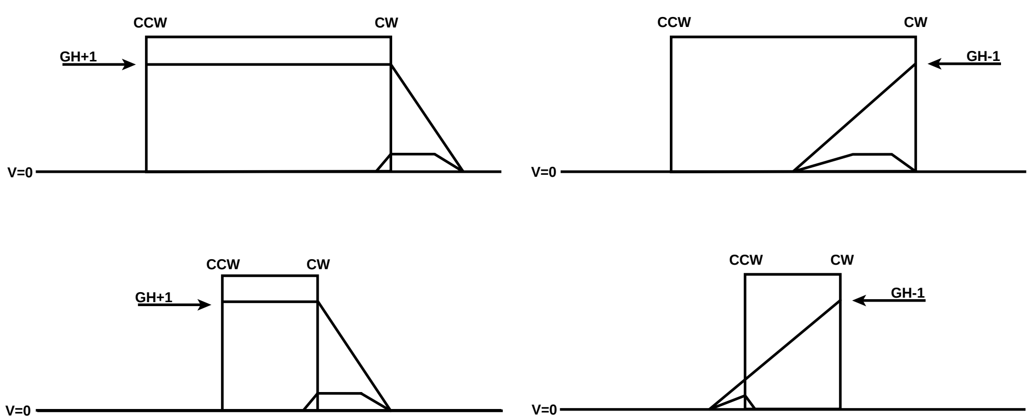
OSB1
OSD1
OSHØ



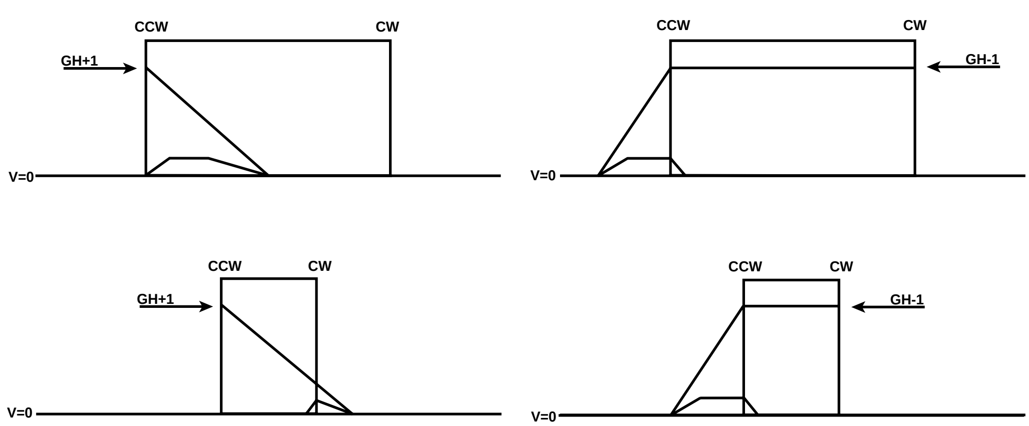
OSB1
OSD1
OSH1



OSB1
OSDØ
OSHØ



OSB1
OSDØ
OSH1



Troubleshooting Table

Symptoms	Possible Causes	Solutions	Useful Commands
No motion	Limits active	Check hard wiring Disable limits	R,RA,LD
	Exceeded maximum distance range	Reset absolute counter with PZ command	PZ, D, MPI, PR
	Position error	CPE command too small Check encoder wiring	R,RSE,CPE,DPE
	Already executing a command	Check status commands	R,RB
	Servo gains too small	Increase gains	CPG,CDG,CIL, CIG
	Stop, kill, trigger inputs active	Check wiring to inputs Check input setup	R,RB,IN,IS
	Loop, pause active	Check status commands	R,RB
	Incorrect unit address	Verify unit address Re-issue unit address	R,#
	In absolute mode (already at position)	Issue new or different absolute distance Issue MPI command	R,MPI,MPA
	Unit shutdown	Enable controller	ST, ON R
No communications	RS-232 miswired	Check wiring for RS-232	
	Wrong RS232 settings	9600 Baud, 8 data bits, 1 stop bit, no parity	
	F command enabled	Issue an E command	E,F
Motor runs away	Encoder miswired	Check encoder wiring Verify power to encoder	
	TQ10X not receiving encoder pulses	With drive disabled turn shaft clockwise (front of motor) PX should read positive counts	ST0,OFF,ST1, ON,DPA,PX
"Not Enabled" LED Illuminated	Position error	Increase allowable position error Decrease move parameters	CPE,DPE, RSE A,V
	Unit disabled	Enable unit	ST0,ON
Unit not responding to commands	Defined a sequence and didn't close with an XT	Issue an XT at the end of a sequence	XD,XT
	Communications disabled	Enable communications	E,F

Non-Drive Related Problems

When the drive is powered up, enabled, and operating properly:

- The **POWER ON** LED is illuminated green
- No LEDs are illuminated red
- The fault output is LOW

These conditions indicate that the TQ10X is probably not the source of the problem. The next table summarizes other possible sources of problems.

TROUBLESHOOTING TABLE	
Possible Source of Problem	***SOLUTION***
MOTOR	Check for motor problems. Check motor coils for continuity, shorts, proper resistance. Check Hall and Phase wiring.
MECHANICAL SYSTEM	Check for jams, binds, increased friction, etc. Uncouple motor from load to test motor separately.
WIRING	Check motor wiring: phases, Hall effects. Check I/O wiring, especially enable.
OVERHEATING	Verify that drive's heatplate has good thermal contact with heatsink. Check mounting screws. Provide sufficient ventilation.
AC POWER	Verify AC power mains supply delivers enough power during entire move without undervoltage, especially during acceleration.
MOVE PROBLEMS	Check speed/torque limitations. Check for excessive friction, regeneration, problems with gravity, transient undervoltage, etc.
ELECTRICAL NOISE	Check for problems caused by electrical noise. Consult the Compumotor sales guide for possible solutions. Check grounds.

Problems During Move

Some problems occur transiently during a move, or do not affect the LEDs. Others may be due to wiring mistakes, or failure of other components in the system (encoder, motor, etc.). The sections below will help you identify such problems.

Speed/Torque Limitations

Make sure that you are not commanding a move that requires the motor to go faster than it can, or use more torque than it can produce. Check the motor's speed/torque curve for your operating conditions.

Acceleration

Some problems during acceleration can be caused by an undervoltage on the AC power line (this can be a transient event), an unrealistic move profile, or too much load inertia.

Deceleration

Some problems during deceleration can be caused by excessive regeneration, resulting in an overvoltage fault.

Excessive Friction

Too much friction in your system might cause move problems. Excessive friction can cause trouble when mechanical components in a system age. As friction increases, problems may occur in a system that had previously been working well.

Mechanical Problems

Check for binds, jams, increased friction, or other problems in the mechanical system. If a system was working properly, but then suddenly develops new problems, check for changes in the mechanical system that could be causing the problems—increased friction, lack of grease, worn bearings, etc.

Protective Circuits

The TQ10X Drive has several protective circuits, some of which can indicate fault conditions by illuminating one of the LEDs.

Drive Overtemperature Protection

To protect against damage from high temperatures, the TQ10X Drive has an internal temperature sensor. If the output stage overheats, the red **DRIVE OVERTEMP** LED will illuminate, and the drive will shut down. This is a latched fault. To restart the drive, first allow it to cool, then cycle power. You can prevent overtemperature faults by heatsinking the drive properly (or adding the -HS3 Heatsink/Fan Unit option), and maintaining ambient temperature at or below 50°C (122°F).

Short Circuit Protection

The TQ10X Drive has short circuit protection. The drive monitors current in its motor output terminals. When the drive detects a short circuit in the motor or motor cabling, it illuminates the **MOTOR FAULT** LED, and stops producing motor current. This is a latched condition. To restart the drive, first fix the short in the motor or cable, then cycle power.

Regeneration

The TQ10X Drive has an internal regeneration resistor. If the motor *regenerates*—produces excess energy during deceleration—the drive will automatically dissipate the excess energy in its regeneration resistor. The power capacity of the resistor is 1KW for one second, or 10 watts on a continuous basis. If the motor regenerates more energy than can be dissipated in the regeneration resistor, the resulting voltage rise on the drive's motor output terminals can cause an overvoltage fault (see below).

If excessive regeneration repeatedly causes overvoltage faults, you may need to alter your move profile, change application conditions, or install an external regeneration resistor (see *Chapter 2—Installation* for instructions).

Vertical applications require careful sizing, and should use a mechanical brake to aid in deceleration.

Overvoltage

The TQ10X Drive monitors the voltage on the motor output terminals. If the voltage rises above a threshold level, the drive will shut down power output to the motor, and will illuminate *red* the bicolor LED labeled **RED = OVERVOLTAGE**. Excessive regeneration is the primary cause of an overvoltage fault.

This is a latched fault. Cycle power to resume operations.

Undervoltage

The TQ10X Drive monitors the voltage on its AC power input terminals. If the voltage falls below 80VAC, the drive will shut down and activate its fault output. When you apply power to the drive, the power-up event is indistinguishable from an undervoltage; therefore, the undervoltage fault is not latched. Furthermore, an undervoltage fault will unlatch any other faults that may be present at the time of the undervoltage.

There is no LED indicator for an undervoltage fault.

Foldback

A mechanical jam in a servo system can cause the motor to overheat. In contrast to a stepper motor, which does not run hotter when jammed, a servo will apply full current (for full torque) while it attempts to move as commanded. Usually, this current will be *much* higher than the motor can withstand continuously. If it persists indefinitely, it may damage the motor's windings.

To help protect the motor from overheating, the TQ10X has a *current foldback* circuit. If high motor current persists for too long, the circuit can reduce the current to a lower level that decreases the rate of motor heating; it can also activate a fault condition.

Three of the DIP switches on top of the drive allow you to set the maximum current the drive will produce ("**PEAK CURRENT LIMIT**") These switches should be set to a value no greater than three times the continuous current rating (Ics) of the motor used. This setting is used by the foldback circuitry to help protect the motor.

Six of the DIP switches configure the foldback circuitry. Three of them set the time that "high" current will be permitted before the foldback circuitry takes action. Settings range from 1 to 10 seconds. One switch defines what constitutes "high" current. This is set according to Ics. One switch will cause a latched fault to be generated when the selected time at high current is exceeded, otherwise current is reduced ("folded back") to 40% of the peak current limit setting. One switch allows foldback to be disabled entirely. (see *Chapter 2—Installation* and *Chapter 4—Tuning* (Tuning procedure) for instructions on setting the dip switches.

How Foldback Works

When actual current produced by the drive exceeds the threshold (relative to the peak current limit) set with SW2-2, the foldback circuit illuminates the **PEAK CURRENT** LED green, and starts a timer. If the current remains above this threshold for longer than the time set with the time-at-peak dipswitches, the foldback circuit activates, and the **IN FOLDBACK** LED illuminates red. The circuit can generate: A) A latched fault, if so configured, or B) reduce current to 40% of the peak current limit setting.

In case A (with **SW2-1 ON**), the fault will remain until power is cycled, or the RESET input is asserted. This fault condition will be indicated by a red **MOTOR FAULT** LED, and an active **FAULT OUT** signal. The **IN FOLDBACK** LED may go out when the **MOTOR FAULT** comes on, depending on the load. When faulted, the motor current will be reduced to zero.

In case B (with **SW2-1 OFF**), the current reduction will persist until the command input goes below the actual current, when the circuit will once again allow the full current permitted by the peak current limit setting. While foldback is limiting the current, the **IN FOLDBACK** LED will be red. When the command is reduced, the LED will go out. The **FAULT OUT** will not go active, since current foldback is not considered a fault.

Oscillating currents, such as are seen when a system goes unstable during tuning, will activate the foldback circuit if the current gets high enough, and the frequency of oscillation is greater than about 20 Hz. Systems in this power range typically have resonant frequencies much higher than this. The default settings suggested for each motor will result in a latched fault after about 3.3 seconds above threshold. This time can be selected with SW1-4,5,6. The sequence of events is: Current exceeds threshold, illuminating the **PEAK CURRENT** LED (green). If this continues for longer than 3.3 seconds, the **IN FOLDBACK** LED will illuminate red, and current will be reduced.

If SW2-1 is ON, then the **MOTOR FAULT** LED (red) will illuminate. The **IN FOLDBACK** LED may go out when the **MOTOR FAULT** LED turns on, depending on the load. Current will be reduced to zero and the **FAULT OUT** will go active. This setting is recommended for nearly all applications, since it causes a motor fault to be indicated. In most applications, allowing the current to be reduced by foldback (without causing a fault) will result in the controller faulting from excess position error, masking the source of the problem, or in the production of defective product. The latched drive fault will lead the troubleshooter to the source of the problem more directly. A mechanical jam, or increased friction in the mechanical system are the common causes.

An exception to this would be an application where stopping the axis will lead to unacceptable consequences, and continued motion, even at a reduced rate, is preferable. An example of this would be a conveyor drive running product through an oven which is hot enough to melt the product eventually.

The default settings cause the timer to start when the current is approximately twice the motor's continuous rating (2 times Ics). The thermal switches in Parker servo motors will protect a properly mounted motor in its specified ambient temperature, up to at least twice Ics, at which point foldback takes over. If the peak current limit is set to three times Ics as recommended, foldback will reduce current to a level below twice Ics, reducing the rate of motor heating enough to give the thermal switch time to react, and protect the motor.

Note that foldback is not a substitute for proper sizing of the application, and does not enforce a lengthy cooling-off period. As soon as the command goes below the actual current, the circuit starts over. Applications with insufficient dwell time to allow the motor to cool between periods of peak current will not be successful.

Foldback is intended to be a secondary safety net. Primary protection of the motor during application problems should be provided by end-of-travel-limit switches, and by proper setting of position-error faults in the controller. These will provide quicker, more direct detection of end-stop or mechanical jamming problems in the application than foldback can. If these systems fail, foldback can generate a fault (which stops all motor current) or reduce the rate of motor heating to enable the thermal switch to operate.

Hall Miswiring

The drive will produce a motor fault and illuminate the red **MOTOR FAULT** LED if it detects an improper Hall state on its Hall effect inputs (all **HIGH** or all **LOW**), or if no motor is connected to the drive. A damaged Hall sensor may also activate the motor fault protection circuit.

To diagnose Hall miswiring or a damaged Hall sensor:

① **Remove AC power**

② **Disconnect I/O:** Disconnect all front panel inputs and outputs, except for enable input, motor temp±, and Hall signals (Hall 1 – 3, Hall +5, Hall Gnd).

③ **Apply AC power**: If the **MOTOR FAULT** LED is still illuminated, then one or more Hall sensors in the motor may be damaged.

Motor Overtemperature Protection

The TQ10X Drive has a circuit that can protect the motor against overheating. Through its **MOTOR TEMP+** and **MOTOR TEMP–** terminals, the drive checks for electrical continuity provided by a normally-closed thermostat mounted on the motor. If the motor overheats and the thermostat opens, the loss of continuity activates protection circuitry in the TQ10X—it turns off power output to the motor, and illuminates the **MOTOR FAULT** LED.

This is a latched fault. Wait for the motor to cool, then cycle power to resume operations.

A motor overtemperature fault may be an indication that your motor is not sized properly for your application. It may also indicate that your motor is not installed properly (poor heatsinking, for example).

A thermostat cannot protect the motor in every situation! Bursts of peak current produce bursts of heat that are quickly absorbed in the motor windings, but slowly dissipated through the motor case and mounting flange. If the bursts are short and infrequent, winding temperature and thermostat temperature will be similar, and the thermostat can protect the motor. If full current flows continuously in the motor, however, the windings can exceed their rated temperature and be damaged, even before the thermostat temperature rises enough to trigger a motor fault.

Technical Support

If you cannot solve your system problems using this user guide, contact your local Automation Technology Center (ATC) or distributor for assistance. If you need to talk to our in-house application engineers, contact Parker Compumotor’s Applications Department at (800) 358-9070, from 6:00 AM to 5:00 PM Pacific time.

Product Return Procedure

If you must return your TQ10X Drive for repairs, use the following steps:

① Get the serial number and the model number of the defective unit, and a purchase order number to cover repair costs in the event the unit is determined to be out of warranty.

② Before you return the unit, have someone from your organization with a technical understanding of the TQ10X Drive and its application include answers to the following questions:

- What is the extent of the failure/reason for return?
- How long did the unit operate?
- Did any other items fail at the same time?
- What was happening when the unit failed (e.g., installing the unit, cycling power, starting other equipment, etc.)?
- How was the unit configured (in detail)?
- What, if any, cables were modified and how?
- With what equipment is the unit interfaced?
- What was the application?
- What was the system environment (temperature, enclosure, spacing, unit orientation, contaminants, etc.)?
- What upgrades, if any, are required (hardware, cables, user guide)?

③ In the USA, call your Automation Technology Center (ATC) for a Return Material Authorization (RMA) number. Returned products cannot be accepted without an RMA number. If you cannot obtain an RMA number from your ATC, call Parker Compumotor's Customer Service Department at (800) 722-2282.

Ship the unit to: Parker Hannifin Corporation
Compumotor Division
5500 Business Park Drive, Suite D
Rohnert Park, CA 94928
Attn: RMA # xxxxxxx

④ In Europe, call Parker Digiplan for a GRA (Goods Returned Authorization) number. Returned products cannot be accepted without a GRA number. The phone number for Parker Digiplan Repair Department is 0202-690911. The phone number for Parker Digiplan Service/Applications Department is 0202-699000.

Ship the unit to: Parker Digiplan Ltd.,
21, Balena Close,
Poole, Dorset,
England. BH17 7DX

⑤ Elsewhere: Contact the distributor who supplied the equipment.

APPENDIX A

Using Non-Compumotor Motors

IN THIS APPENDIX

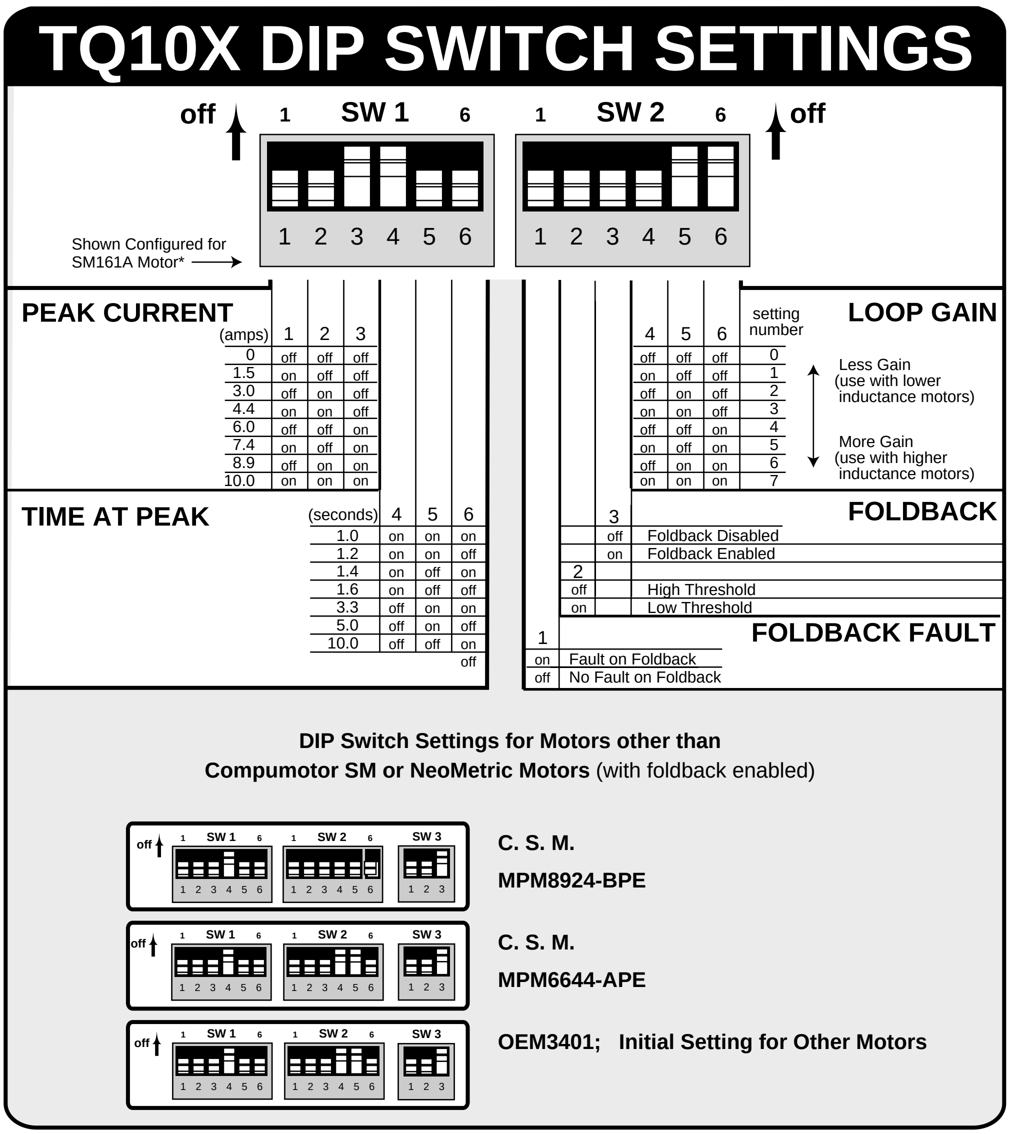
- Configuring DIP Switches
- Connecting Motor Phase Wires and Hall Effect Wires

Using Motors other than Compumotor SM or N Series Motors

Test all motors carefully. Verify that the motor temperature in your application is within the system limitations. *The motor manufacturer's maximum allowable motor case temperature must not be exceeded.* You should test the motor over a 2-to-3 hour period. Motors tend to have a long thermal time constant, but can still overheat, which results in motor damage.

Configuring the TQ10X Drive's Dip Switches

Set the TQ10X's 12 DIP switches located on top of the drive, and 2 DIP switches located on the bottom of the drive. The following drawing shows DIP switch settings for selected motors.



DIP Switch Settings – Non-Compumotor SM Motors

If you are unsure of which settings to use for your motor, start with the setting shown above for the OEM3401 motor. If the motor gets hot when it is stopped, reduce the loop gain setting (DIP SW2 — #4, #5, #6).

The DIP switch settings shown above will enable foldback—after approximately three seconds at full current, the drive will fault due to foldback. These settings will help protect your motor while you prototype your system. To disable the foldback fault, to adjust time until foldback, or to disable foldback entirely, adjust the DIP switches.

If you have further questions, *first consult your motor vendor* to obtain a full and complete motor specification sheet. Consult your Automation Technology Center (ATC) if you have questions regarding the use of a non-Compumotor motor with Compumotor equipment. If you still need further information, call Compumotor's Application Engineering Department at 800-358-9070.

Connecting Motor Phase Wires and Hall Effect Wires

If you use a motor from another vendor, obtain information from the motor’s manufacturer about its phase wire color code, sequence of Hall states, commutation scheme, etc. Use the information below for guidance on how to connect your motor's phase and Hall wires to the TQ10X.

Improper Wiring Can Result in Poor Performance

Assume that you arbitrarily connect your motor’s three Hall wires to the TQ10X’s Hall inputs. For any particular Hall wiring pattern, there are six different ways you can connect wires to Phase A, Phase B, and Phase C.

Of these six possible phase wiring combinations, only one will work properly. Three will not work at all. The other two deserve particular attention: if the motor is wired in one of these two configurations, the motor will turn, but its performance will be severely impaired.

How can you tell if your motor is wired improperly? If it is in one of the two poor-performance configurations, its torque will be much lower than the torque level of a properly wired motor. Also, torque ripple will be very pronounced as the motor turns.

The best way to determine whether or not your motor is wired correctly is to find the three wiring configurations that enable the motor to turn. Compare the motor’s torque in each configuration. The configuration with the most torque will be the proper configuration.

Trial and Error Method

WARNING

Motor shaft rotation may be opposite than you expect. Do not connect a load to the shaft until you first determine the direction of shaft rotation.

You can use a trial and error method to connect your motor to the TQ10X. Follow these steps:

① Arbitrarily assign numbers to your motor's three Hall output wires, and connect them to **HALL 1**, **HALL 2**, and **HALL 3** on the TQ10X.

② Connect **HALL +5V** and **HALL GND**.

③ Arbitrarily assign letters (A, B, C) to your motor's phase wires, and connect them to Phase A, Phase B, and Phase C on the TQ10X.

④ If the motor turns, find the best phase wiring configuration:

- Move each phase wire over one position (A B C ➔ C A B). Compare torque and torque ripple.
- Move each phase wire one position further (C A B ➔ B C A). Compare torque and torque ripple.
- Use the wiring configuration that gives highest torque and lowest torque ripple.

⑤ If the motor does not turn, exchange two of the phase wires. The motor should now turn. Go to *Step ④*, compare the three wiring configurations that make the motor turn, and use the proper one.

⑥ If your motor turns in the opposite direction than you want, you can reverse it using one of several methods.

- Reverse the appropriate encoder connections.
- Exchange two Hall input wires, then follow steps 2 through 5 above.

APPENDIX B

LVD Installation Instructions

IN THIS APPENDIX

- LVD Installation Instructions

LVD Installation Instructions

For more information about the Low Voltage Directive (LVD), see 73/23/EEC and 93/68/EEC, published by the European Economic Community (EEC).

Environmental Conditions

Pollution Degree

The TQ10X Drive is designed for pollution degree 2.

Installation Category

The TQ10X Drive is designed for installation category II.

Electrical

Connecting and Disconnecting Power Mains

The TQ10X Drive's protective earth connection is provided through its make first/break last earth terminal on the power mains connector. You must reliably earth the TQ10X Drive's protective earth connection.

Attach or remove the TQ10X Drive's power plug only while input power is OFF.

Using an Isolation Transformer

The TQ10X Drive's mains voltage is limited to 120 VAC nominal. If your mains voltage is higher, use an isolation transformer located between the power mains and the TQ10X Drive. Your isolation transformer should be insulated to ~2300V rms.

Do not interrupt the protective earth conductor between the source mains and the isolation transformer's secondary. The core of the isolation transformer and the drive's protective conductor terminal must *both* be connected to the main's protective earth conductor.

CAUTION

Do not use an autotransformer.

Line Fuses

Line fuses need to be added to protect the transformer and associated wiring. If the live wire cannot be readily identified, fuse both phase conductors. The value of fuse required is given by:

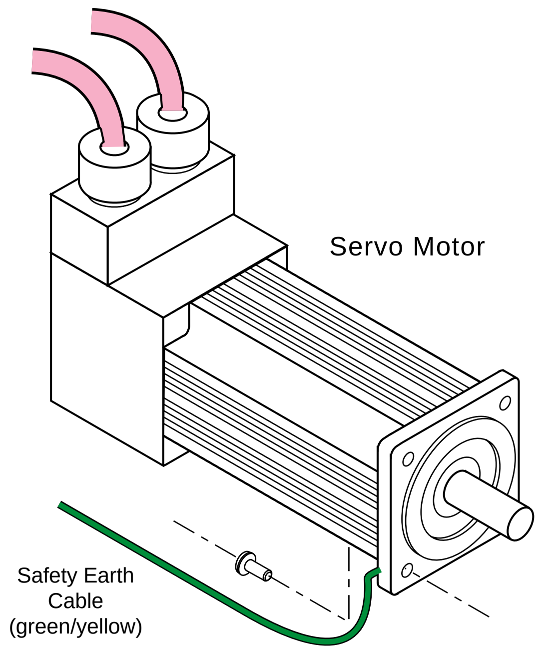
(1.5 x VA)/(supply volts) [amps]

Fuse types should be anti-surge HBC.

Providing a Protective Earth Connection for Motors

You must provide a connection from the motor to a reliable protective earth contact point. This connection provides a protective earth for the motor, and is *in addition* to the earth connection provided by the drain wire in the motor's power cable. The motor's protective earth connection is important for safety reasons, and *must not be omitted*.

Make connections according to the following instructions and diagram:



Providing Protective Earth Connection for Motor

① Use a spade lug in combination with a star washer and mounting bolt to make good contact with the bare metal surface of the motor's mounting flange.

② Use a green and yellow striped wire to make the connection between the motor and earth. Wire gauge must be no thinner than the current carrying wire in the motor's power cable.

③ Resistance between the motor and earth must be no greater than 0.1 ohm. Use thicker gauge wire if the resistance is too high.

Mechanical

Installing in an Enclosure

The TQ10X Drive must be installed within an enclosure. The enclosure's interior must not be accessible to the machine operator. The enclosure should be opened only by skilled or trained service personnel.

Servicing the TQ10 Drive

Changing Firmware

Only skilled or trained personnel should change firmware.

Do Not Replace Fuses

The TQ10X Drive has no fuses designed to be replaced by the user. Fuse failure indicates that other components have also failed. Fuses and other components should only be replaced by Compumotor or its designated repair facilities.

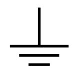
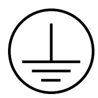
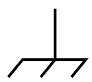
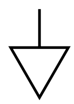
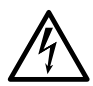
Thermal Safety

The Motor May Be Hot

The motor may reach high temperatures during normal operations, and may remain hot after power is removed.

Table of Graphic Symbols and Warnings

The following symbols may appear in this User Guide, and may be affixed to the products discussed in this user guide.

Symbol	Description
	Earth Terminal
	Protective Conductor Terminal
	Frame or ChassisTerminal
	Equipotentiality
	Caution, Risk of Electric Shock
	Caution, Refer to Accompanying Text
	Hot Surface

Index

Symbols

A

B

C

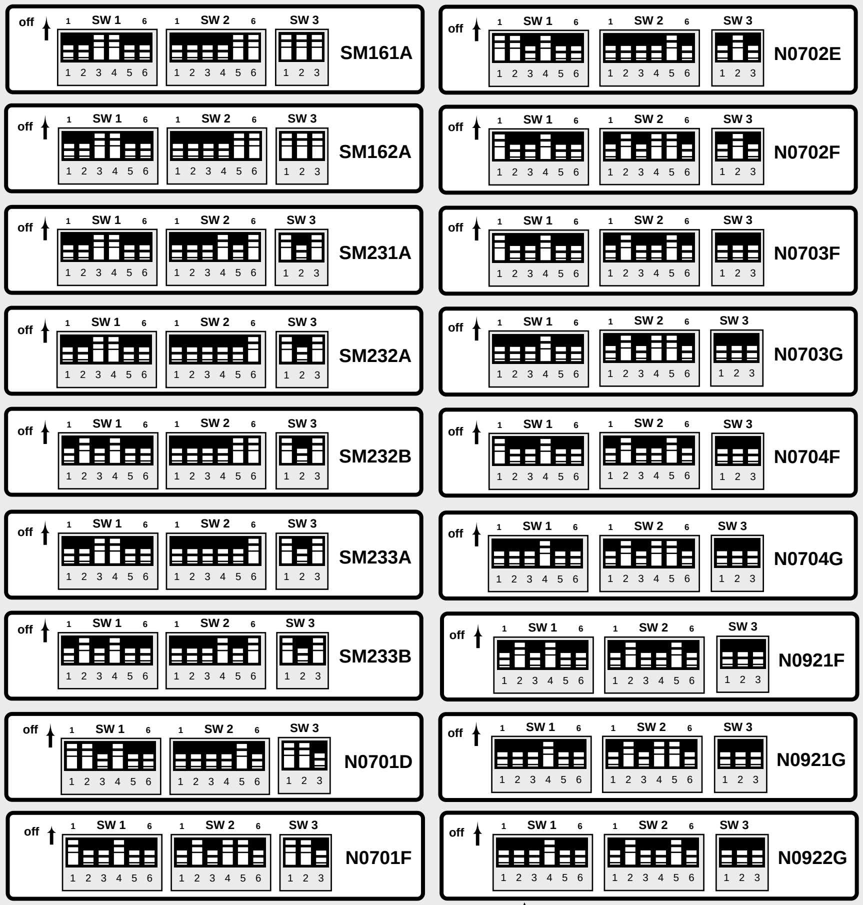
D

TQ10X Drive

Compumotor

*Switches shown at the right are configured for initial tuning, with peak current approx. twice motor's continous current rating. See Tuning for procedure to raise current.

DIP Switch Settings for Compumotor SM and NeoMetric Motors* (with foldback enabled)



TQ10X Status LEDs *see page 104*

MOTOR FAULT	Red indicates short circuit in motor cabling, or indicates motor overtemperature.
DRIVE OVERTEMP	Red indicates drive has exceeded temperature limit.
REGEN/ OVERVOLTAGE	Illuminates green during regen event; illumates red if regen causes overvoltage.
PEAK CURRENT/ IN FOLDBACK	Illluminates green during peak current output; illuminates red while drive is in foldback.
POWER ON/ NOT ENABLED	Illuminates green when AC power is applied; illuminates red when AC power is applied but drive is not enabled.