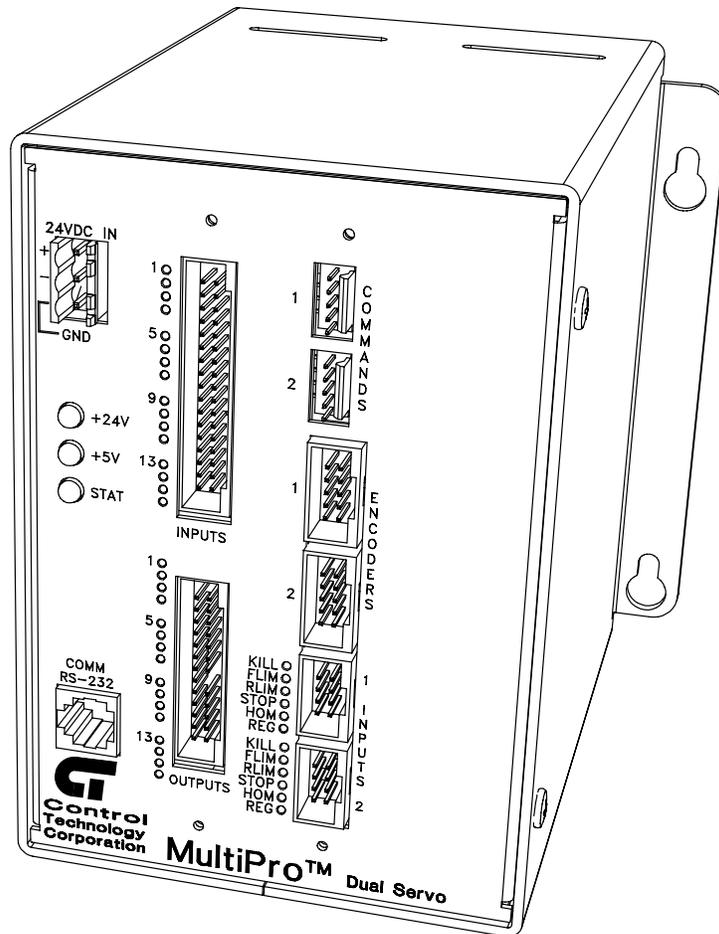


# Model 2648 MultiPro™ 16 In / 16 Out Dual Servo Installation Guide



Doc. No. 2648IG  
Revision E  
November 2001

---

The information in this document is subject to change without notice. The software described in this document is provided under license agreement and may be used or copied only in accordance with the terms of the license agreement.

The information, drawings, and illustrations contained herein are the property of Control Technology Corporation. No part of this manual may be reproduced or distributed by any means, electronic or mechanical, for any purpose other than the purchaser's personal use, without the express written consent of Control Technology Corporation.

The following are trademarks of Control Technology Corporation:

- Quickstep
- CTC Monitor
- CTC Utilities

Windows is a trademark of Microsoft Corporation.

# Contents

---

- Notes to Readers ..... 1**
- Related Documents ..... 2
- Formatting Conventions ..... 2
- Contacting Control Technology Corporation ..... 3
- Your Comments ..... 3

**1 Getting Started**

- Dimensions and Mounting Considerations ..... 7
- 2648 Description ..... 8
- Connectors and Pinout Diagrams ..... 9
- Specifications ..... 12
- Power Connections ..... 16
  - Connecting DC Power ..... 16
  - The Importance of Proper Grounding ..... 16
- Status Light Description ..... 17
  - Software Fault ..... 17
  - Hardware Fault ..... 17
- Connecting Digital Inputs ..... 18
  - Activating a Digital Input ..... 18
  - Using Solid-State Sensors ..... 18
- Connecting Digital Outputs ..... 20
  - Using Open-Collector Outputs ..... 20
  - Connecting Multiple Devices ..... 21
  - Connecting to a Second External Supply ..... 21
- Connecting Dedicated Inputs ..... 24
  - Using Solid-State Sensors ..... 24
- Computer - Controller Communications ..... 25
  - RS-232 Communications ..... 25
  - RS-232 Connections (Single Port) ..... 25
  - RS-232 Connections (Dual Ports) ..... 26
  - Connecting to a D Connector ..... 27

## 2 Application Notes

Using the Real-Time Clock .....	31
Examples .....	31
Programming a Servo .....	33
Setting Up Servo Motor Operating Parameters .....	33
Using Servo Filters .....	35
PID Filter .....	35
PAV Filter .....	35
Direct Mode .....	36
Sample Servo Motor Tuning Program .....	36
Setting Acceleration and Deceleration Values .....	38
Searching for Home .....	38
Specifying the Homing Direction .....	39
Turning a Servo .....	39
Stopping the Servo .....	40
Monitoring and Changing Other Servo Parameters .....	40
Monitoring Dedicated Inputs .....	40
Setting Up Electronic Following .....	42
Dual Servo Axis-to-Axis Following .....	42
Configuring Electronic Following .....	43
Ending Electronic Following .....	43
Reading Current Position and Velocity .....	44
Specifying Encoder Following .....	44
Interconnecting Multiple Followers With the Same Leader .....	45
Setting Up Registration for a Servo .....	46
Designating a Predefined Registration Window .....	46
Using Registration to Change the End Position of a Move .....	47
Overshooting the End Position .....	49
Registration during Deceleration .....	50
Guidelines and Rules for Setting up Registration .....	51
Servo Hardware Considerations .....	52
Dedicated Inputs .....	52
Enabling Servo Drives .....	53
Sample Quickstep Programs .....	55
Example 1 — Absolute Move of One Servo Motor .....	55
Example 2 — Relative Move of One Servo Motor .....	55
Example 3 — Velocity Move of One Servo Motor .....	56
Example 4 — Changing the Velocity of a Servo Motor During Motion .....	56
Example 5 — Ratio Axis to Leader Encoder .....	57
Example 6 — Velocity Move of Two Servo Motors .....	57
Example 7 — Ratio of Two Servo Axes Follower to Leader .....	58
Example 8 — Absolute Move of Two Servo Motors .....	59
Example 9 — Staggering the Motion of Two Servo Motors .....	59

### 3 Special Registers

Special Purpose Registers for Servos .....	63
Group Access Registers .....	63
Leader On-Start Feature .....	63
Axis Status and FeedForward Parameters .....	63
Individual Access Registers .....	64
Axis Status and FeedForward Parameters for Axis #1 .....	64
Registration Feature for Axis #1 .....	65
Axis Following Feature for Axis #1 .....	66
Special Registers – Additional Features for Axis #1 .....	67

This page is intentionally left blank.

# Notes to Readers

---

The *Model 2648 Installation Guide* provides the following information:

- Dimensions and Mounting Considerations -- mounting dimensions and precautions on mounting the MultiPro.
- Description and Connection Diagrams -- an overview of the MultiPro's basic functions; pinout diagrams for all connectors.
- Specifications -- general, servo, and digital I/O specifications; hardware and firmware revisions for the 2648 controller.
- Power Connections -- connecting power to the MultiPro.
- Status Lights -- how the status light functions.
- I/O Connections -- connecting digital inputs and outputs; connecting dedicated inputs.
- Controller Communications -- describes the controller's RS-232 port and its function.
- Application Notes -- how to use the real-time clock; programming a servo, electronic following and registration setup, hardware considerations, and sample Quickstep programs.
- Special Purpose Registers -- how to use the special purpose registers.

## Related Documents

The following documents contain additional information:

- For information on Quickstep, refer to the *Quickstep™ Language and Programming Guide* or the *Quickstep™ User Guide*.
- For information on the registers in your controller, refer to the *Register Reference Guide* (available at [www.ctc-control.com](http://www.ctc-control.com)).
- For information on Microsoft Windows or your PC, refer to the manuals provided by the vendor.

## Formatting Conventions

The following conventions are used in this book:

<b>ALL CAPS BOLDFACE</b>	Identifies DOS, Windows, and installation program names.
<b>Boldface</b>	Indicates information you must enter, an action you must perform, or a selection you can make on a dialog box or menu.
<i>Italics</i>	Indicates a word requiring an appropriate substitution. For example, replace <i>filename</i> with an actual file name.
Text_Connected_With_Underlines	Indicates symbolic names used in Quickstep programs. Step Names are ALL_CAPITALS. Other symbolic names can be Initial_Capitals or lower_case.
SMALL CAPS	Identifies the name of Quickstep instructions in text.
Courier font	Identifies step names, comments, output changes, and Quickstep instructions appearing in the Quickstep editor.
Art Code 	Identifies the file name of a particular graphic image.

## Contacting Control Technology Corporation

Control Technology Corporation is located in Massachusetts. Our business hours are 8:30 AM to 5:00 PM. EST (Eastern Standard Time).

Contact Method	Address or Number
<b>E-Mail:</b>	
Technical Support:	help@ctc-control.com
Technical Publications:	techpubs@ctc-control.com
<b>Website:</b>	www.ctc-control.com
<b>Telephone:</b>	508.435.9595 and 800.282.5008
<b>FAX:</b>	508.435.2373
<b>Mail:</b>	Control Technology Corporation 25 South Street Hopkinton, MA 01748

## Your Comments

Suggestions and comments about this or any other Control Tech document can be e-mailed to the Technical Publications Group at techpubs@ctc-control.com.

This page is intentionally left blank.

# Getting Started

---

Dimensions and Mounting Considerations .....	7
2648 Description .....	8
Connectors and Pinout Diagrams .....	9
Specifications .....	12
Power Connections .....	16
Status Light Description .....	17
Connecting Digital Inputs .....	18
Connecting Digital Outputs .....	20
Connecting Dedicated Inputs .....	24
Computer - Controller Communications .....	25

This page is intentionally left blank.

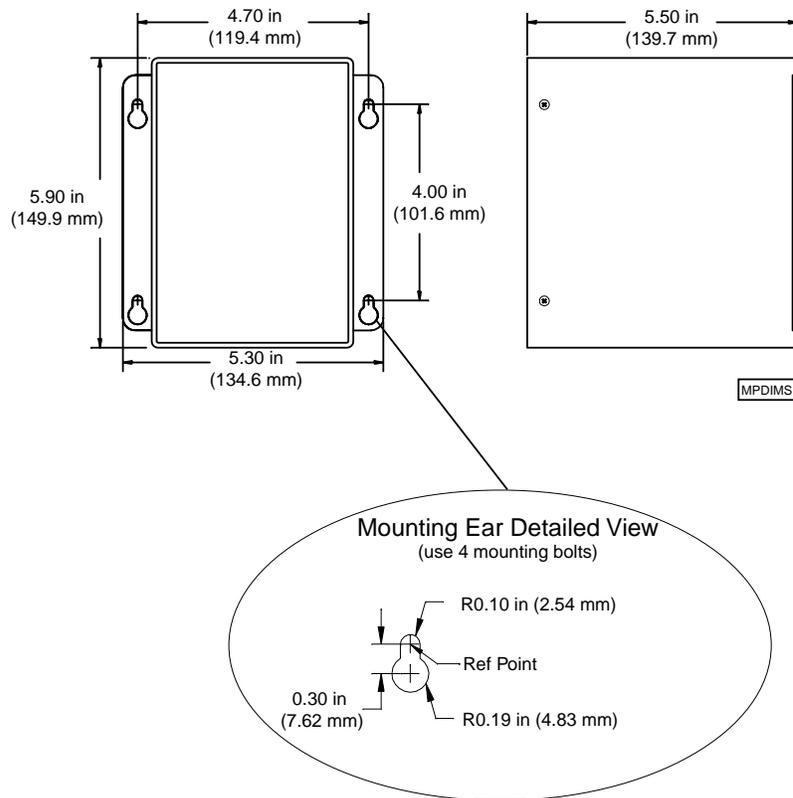
## Dimensions and Mounting Considerations

All MultiPro controllers have mounting ears that allow easy mounting to flat surfaces such as a NEMA-rated electrical enclosure. Figure 1–1 provides mounting ear details and MultiPro dimensions.

Select a mounting location that protects against the environmental hazards listed below:

- Avoid flying metal chips that may result from installation or subsequent machine construction. You should also avoid conductive dusts, liquids, or condensing humidity. If any of these conditions exist, mount the MultiPro in a NEMA 4 or NEMA 12 rated enclosure.
- Do not mount the MultiPro in an environment that requires explosion proof practices.
- Avoid mounting locations that are in close proximity to devices that produce electromagnetic interference (EMI) or radio frequency interference (RFI). Devices such as motor starters, relays, large power transformers, and ultrasonic welding apparatus fall into this category.

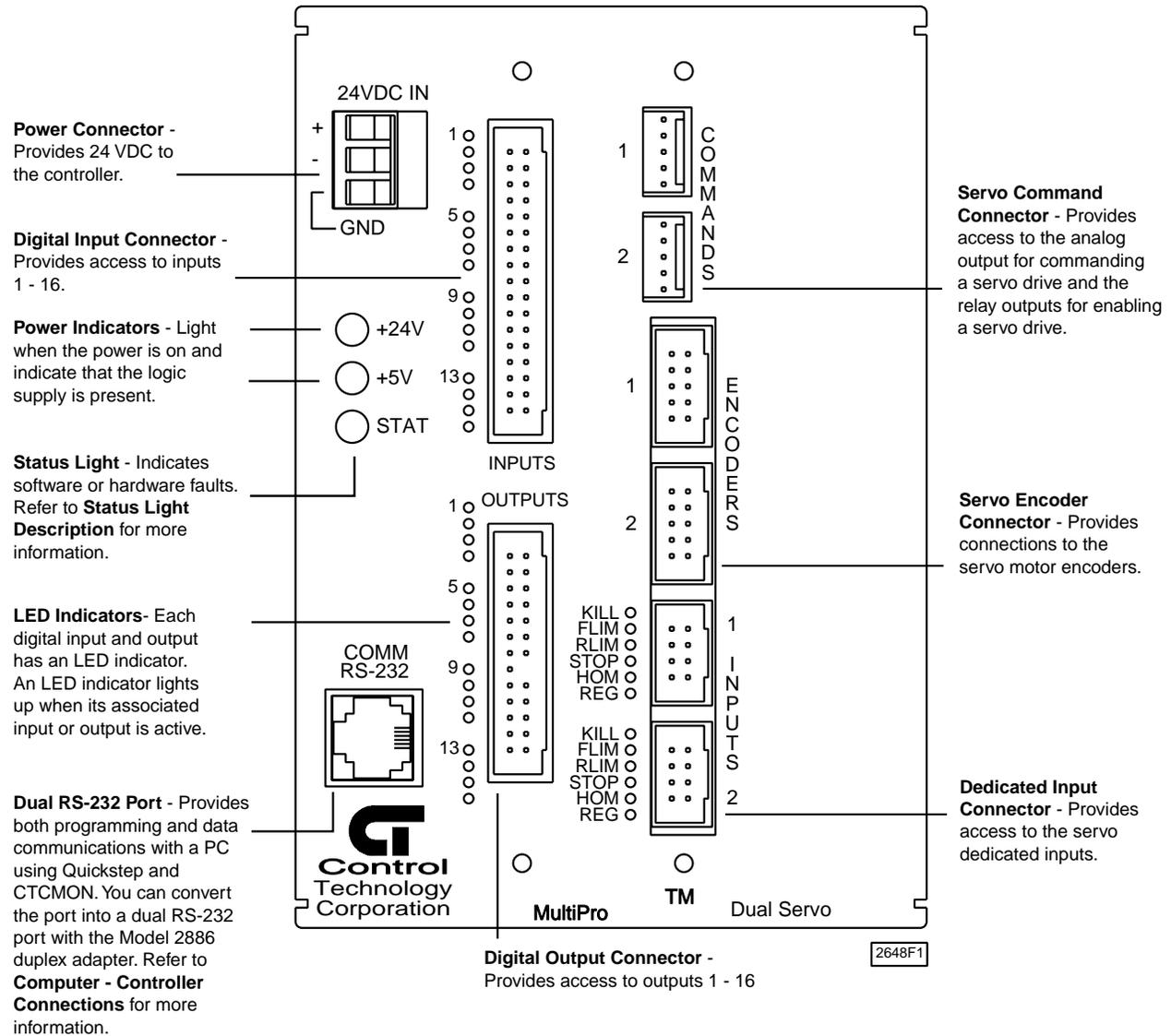
Figure 1–1. MultiPro Dimensions and Mounting Ear Details



# 2648 Description

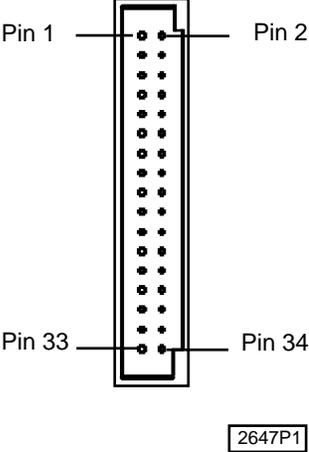
Figure 1-1 shows the 2648's faceplate and describes its different features.

Figure 1-1. 2648 Faceplate and Features

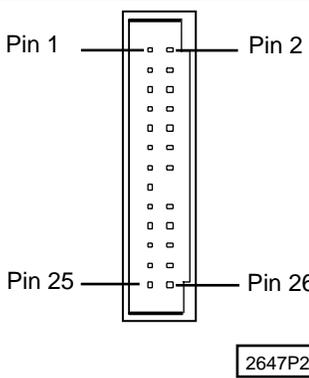


# Connectors and Pinout Diagrams

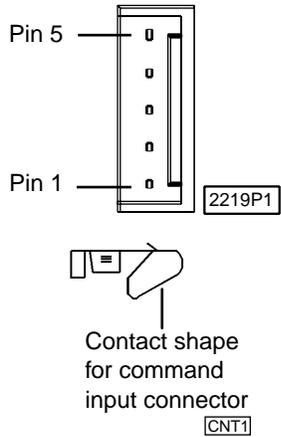
**Table 1–1. Connection Diagram - Digital Input Connector**

Digital Input Connector	Pin #	Signal	Pin #	Signal
	1	Input 1	2	Return
	3	Input 2	4	Return
	5	Input 3	6	Return
	7	Input 4	8	Return
	9	Input 5	10	Return
	11	Input 6	12	Return
	13	Input 7	14	Return
	15	Input 8	16	Return
	17	Input 9	18	Return
	19	Input 10	20	Return
	21	Input 11	22	Return
	23	Input 12	24	Return
	25	Input 13	26	Return
	27	Input 14	28	Return
	29	Input 15	30	Return
	31	Input 16	32	Return
33	+24 VDC	34	Return	

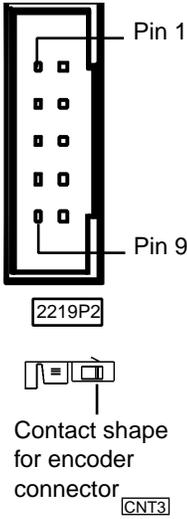
**Table 1–2. Connection Diagram - Digital Output Connector**

Digital Output Connector	Pin #	Signal	Pin #	Signal
	1	Output 1	2	Output 14
	3	Output 2	4	Output 15
	5	Output 3	6	Output 16
	7	Output 4	8	Return
	9	Output 5	10	Return
	11	Output 6	12	Return
	13	Output 7	14	Return
	15	Output 8	16	No pin
	17	Output 9	18	+24 VDC
	19	Output 10	20	+24 VDC
	21	Output 11	22	+24 VDC
	23	Output 12	24	+24 VDC
	25	Output 13	26	NC

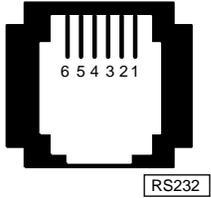
**Table 1–3. Connection Diagram - Command Connector**

Command Connector	Pin #	Signal
 <p>Pin 5</p> <p>Pin 1</p> <p>2219P1</p> <p>Contact shape for command input connector</p> <p>CNT1</p>	5	Analog Command Output
	4	Analog Command Return
	3	Drive Kill Relay (N.O.)
	2	Drive Kill Relay (Common)
	1	Shield

**Table 1–4. Connection Diagram - Encoder Connector**

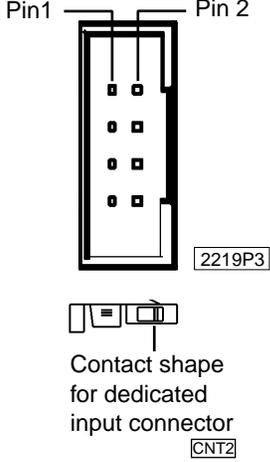
Encoder Connector	Pin #	Signal	Pin #	Signal
 <p>Pin 1</p> <p>Pin 9</p> <p>2219P2</p> <p>Contact shape for encoder connector</p> <p>CNT3</p>	1	Channel A (+)	6	5 Volt Return
	2	Channel A (-)	7	+5 VDC (for encoder)
	3	NC	8	Channel B (+)
	4	NC	9	Channel B (-)
	5	Index (-)	10	Index (+)

**Table 1–5. Connection Diagram - RS-232 Connector**

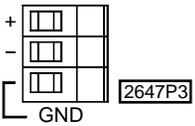
RS-232 Connector <sup>1</sup>	Pin #	Signal
	1	TxD Outbound 2
	2	TxD Outbound 1
	3	Common
	4	Common
	5	RxD Inbound 1
	6	RxD Inbound 2

1. The Model 2886 duplex adapter provides convenient access to both RS-232 ports. Refer to the *RS-232 Connections (Dual Ports)* section for more information.

**Table 1–6. Connection Diagram - Dedicated Input Connector**

Dedicated Input Connector	Pin #	Signal
	1	Start
	2	Kill
	3	Forward Limit
	4	Reverse Limit
	5	Home
	6	Registration Input
	7	+24 VDC
	8	24 V Return

**Table 1–7. Connection Diagram - Power Connector**

Power Connector	Pin #	Signal
	1	+24 VDC
	2	+24 V Return
	3	Ground

# Specifications

**Table 1–8. General Specifications**

Description	Min.	Typical	Max.	Units
<b>Absolute Maximum Ratings</b>				
Ambient Temperature				
Operating	0		+50	°C
Storage	-20		+80	°C
<b>Controller Characteristics</b>				
Voltage range	22.0	24.0	27.0	VDC
Current requirements at 24 VDC		.4	.6	A
User memory capacity (11 year lithium-cell RAM)		32K		Bytes
<i>The main CPU is an Hitachi SH2 processor running at 24.576 MHz.</i>				
<b>Controller Performance Specifications <sup>1</sup></b>				
Controller CPU clock speed		24.576		MHz
Servo CPU clock speed		16		MHz
Sense input, jump to new step, change output		0.5		ms
Perform multiplication (between volatile registers)		0.3		ms
Change servo profile		1.0		ms
Time delay duration, 10 ms programmed		10.08		ms
Time delay duration, 1 s programmed		1.001		s
Internal count rate		750		Hz
<b>Communications Characteristics</b>				
RS-232 transmitters		± 5	± 12	VDC
RS-232 receivers	± 5		± 12	VDC
<b>Controller Resource Summary</b>				
Multi-tasking (tasks)			56	
Volatile registers (32-bit)			490	
Non-volatile registers (32-bit)			500	
Data table elements (16-bit, non-volatile)			8000	
Input-linkable counters			16	
Flags			32	
Program steps			1280	
<ol style="list-style-type: none"> <li>Performance specifications shown are with one task running. RS-232 communications may degrade the count by up to 10%.</li> <li>Specifications are at 25°C unless otherwise specified.</li> </ol>				

**Table 1–9. Servo Specifications**

Description	Min.	Typical	Max.	Units
<b>Absolute Maximum Ratings</b>				
Command load resistance	2			kΩ
Encoder input voltage	0.0		+5.0	VDC
Encoder (+5 V) supply output current (total for each pair of axes)			500	mA
<b>Electrical Specifications</b>				
Command outputs				
Nominal voltage range	-10		+10.0	VDC
Differential encoder inputs				
Nominal input range	0.0		+5.0	VDC
Open circuit voltage ( $I_i = 0$ mA)		5.0	5.38	VDC
Logic low current ( $V_i = 0$ V)		1.1	1.2	mA
Axis control inputs (except registration)				
Off voltage ( $I_i = 0$ mA) <sup>4</sup>		24.0	26.4	VDC
On current ( $V_i = 0$ V)		2.12		mA
Threshold				
Low-to-high		14.0		VDC
High-to-low		12.5		VDC
Registration auxiliary input				
Off voltage ( $I_i = 0$ mA)		24.0	26.4	VDC
On current ( $V_i = 0$ V)		2.28		mA
Threshold				
Low-to-high		5.1		VDC
High-to-low		4.9		VDC
<b>Performance Specifications</b> <sup>5</sup>				
Maximum velocity setting	1		4,000,000	steps/s
Resolution of maximum velocity setting		1		steps/s
Acceleration and deceleration settings	1		130,000,000	steps/s <sup>2</sup>
Resolution of acceleration and deceleration settings		1		steps/s <sup>2</sup>

- Specifications are at 25°C unless otherwise specified.
- PID parameters are programmed as relative values in the range of 0-255. Acceleration ( $A_{ff}$ ) and Feedforward Velocity ( $V_{ff}$ ) range from 0-32,767.
- The ratio range for both axis following and ratio control is  $\pm 1:32,767$  minimum and  $\pm 32,767:1$  maximum. Depending on the application, high ratios may result in instability.
- This value is dependent on the controller's auxiliary supply voltage (24V is typical).
- The term *step* refers to one edge transition on either encoder input for that axis.

**Table 1–9. Servo Specifications (Continued)**

Description	Min.	Typical	Max.	Units
Position range (absolute mode)	-2,147,483,648		2,147,483,647	steps
Relative motion command range	-2,147,483,648		2,147,483,647	steps
Position registration accuracy		± 1		count

- Specifications are at 25°C unless otherwise specified.
- PID parameters are programmed as relative values in the range of 0-255. Acceleration ( $A_{ff}$ ) and Feedforward Velocity ( $V_{ff}$ ) range from 0-32,767.
- The ratio range for both axis following and ratio control is ± 1:32,767 minimum and ± 32,767:1 maximum. Depending on the application, high ratios may result in instability.
- This value is dependent on the controller's auxiliary supply voltage (24V is typical).
- The term *step* refers to one edge transition on either encoder input for that axis.

**Table 1–10. Digital I/O Specifications**

Description	Min.	Typical	Max.	Units
<b>Absolute Maximum Ratings</b>				
Applied input voltage <sup>1</sup>	0		27.0	VDC
Applied output voltage <sup>2</sup>	0		24.0	VDC
Output current				
Single output			500	mA DC
Total limit			5	A
<b>Operating Characteristics</b>				
Output on voltage ( $I_o = 500$ mA)		0.8	1.8	VDC
Output off leakage (applied voltage = 24V) <sup>3</sup>		0.01	0.75	µA
Input off voltage ( $I_i = 0$ mA)		24.0	26.4	VDC
Input on current ( $V_i = 0$ V)		-2.10	-2.85	mA
Input on current threshold ( $V_i = 8$ V typical)		-1.0	-1.85	mA
Input off current (typical leakage current allowable)			-250	µA

- Under normal operation, no external input voltage is applied. Inputs should be externally switched to the input common.
- An on-board protection diode returns to +24 V from each output.
- In the off state, unconnected outputs are internally pulled to +5 V through a diode and an LED indicator.
- Specifications are at 25°C unless otherwise specified.

**Table 1–11. Hardware / Firmware Revision Levels**

Model Numbers	Hardware Revision Level	Firmware Revision Level <sup>1 2</sup>
All	0	2.18
<p>1. You can confirm firmware revision levels by doing a register read in Quickstep's monitor program. Use register 13003 to confirm the firmware revision in a MultiPro controller.</p> <p>2. Firmware revision levels are not equivalent to standard decimal numbers. For example, firmware revision level 2.18 translates to:</p> <p style="padding-left: 40px;">Major Revision Level 2 Minor Revision Level 18</p> <p>If this value changes to 2.20, it translates to:</p> <p style="padding-left: 40px;">Major Revision Level 2 Minor Revision Level 20 (not revision level 2)</p>		

## Power Connections

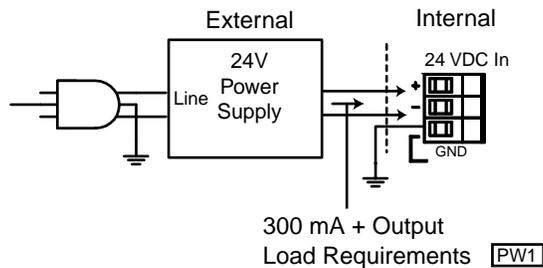
This section describes how to connect power to the MultiPro.

### Connecting DC Power

The MultiPro requires an externally supplied voltage of +24 VDC for proper operation (Figure 1–2). Power is connected through the connector located on top of the MultiPro.

+24 VDC supplies power to all the I/O circuitry and also to a DC/DC converter. This conversion circuit creates an isolated +5 VDC supply that provides power to the MultiPro's logic circuitry.

Figure 1–2. 24 Volt Power Supply



### The Importance of Proper Grounding

The MultiPro's ground should follow a direct, low-impedance path to the plant's power source that is not shared by any machinery that injects a large amount of electrical noise onto the line.



#### Note

For more information on noise protection, refer to CTC Technical Note No. 26, *Reducing Noise Susceptibility*. This document is available in the Customer Support area of our Web site at [www.ctc-control.com](http://www.ctc-control.com).

---

## Status Light Description

The status light (STAT) is located on the MultiPro's front left panel and is used to indicate a software or hardware fault. These faults are described below.



### Note

After the MultiPro is turned ON, the light is steady and red during the first second of operation.

---

## Software Fault

Quickstep programs may produce software faults that are indicated by a periodic flashing light. These faults occur when the MultiPro is unable to execute because an application problem exists within the Quickstep program. You can view the fault type by viewing the program status in Quickstep's monitoring utility, CTCMon. Once a software fault occurs, the MultiPro is idle and all resources that can be set (outputs, etc.) remain in the state they were in before the fault occurred.



### Note

You can program register 13009 to turn off a specific output when a software fault occurs. Refer to the *Quick Reference Register Guide* at [www.ctc-control.com](http://www.ctc-control.com) for more information.

---

## Hardware Fault

Hardware faults are displayed as a steady red light and indicate that the internal watchdog timer has disabled the MultiPro's CPU. When this occurs, the MultiPro's outputs are also disabled.

Try clearing a hardware fault by cycling the power and/or downloading the same Quickstep program. If the fault doesn't clear, your MultiPro controller may require repair. Contact our Technical Support representative for assistance.

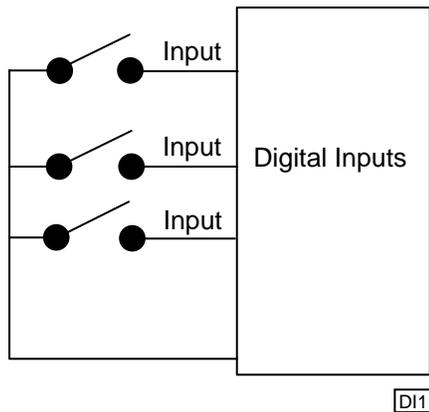
## Connecting Digital Inputs

This section describes the digital inputs and how to connect devices to these inputs.

### Activating a Digital Input

The Model 2648 has 16 digital inputs that you can activate with a switch closure (Figure 1–3) to **Return**, which is the common for the controller's 24 V supply. Each input is opto-isolated from the MultiPro's logic circuitry and is internally self-powered by the 24 V supply through a current-limiting resistor.

Figure 1–3. Digital Input Activation



The MultiPro senses when an input is pulled down to **Return** by a switch closure. A `MONITOR` instruction or any other programmed instruction referring to a general purpose input can use this information.

### Using Solid-State Sensors

You can connect many types of electronic sensors such as three-wire Hall-effect sensors, proximity sensors, and phototransistors to the inputs without any additional circuitry. These devices must have sinking type open-collector outputs (NPN) and must be able to withstand at least +24 V on their output terminals when they are in the OFF state. The sensor must also be able to sink the required input current (i.e. - 2.1 mA) when ON.



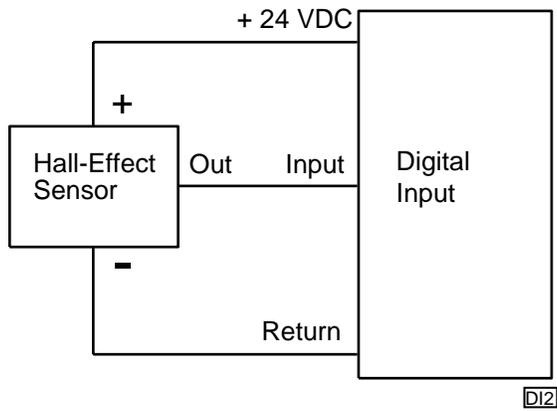
#### Note

Do not use two-wire, solid-state sensors.

---

Electronic sensors have internal circuitry that generally requires an external power source. Figure 1-4 shows how to connect a solid-state sensor.

Figure 1-4. Connecting a Solid-State Sensor to a Digital Input



## Connecting Digital Outputs

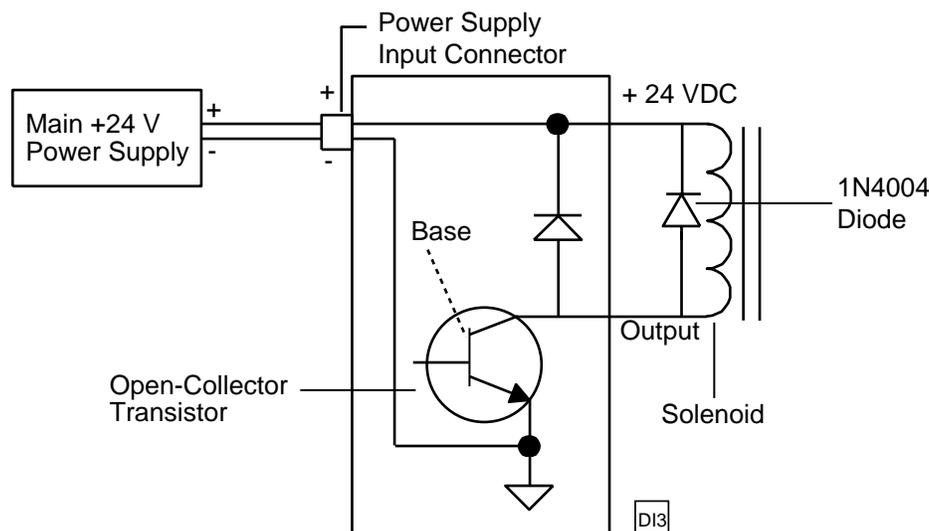
This section describes the digital outputs and how to connect devices to these inputs.

### Using Open-Collector Outputs

The Model 2648 has 16 outputs for driving external loads such as solenoid valves, indicators, solid-state relays and other low-power DC loads. These outputs are configured as open-collector transistors that can switch loads up to 0.5 Amps DC. Open-collector outputs are transistors whose collector terminal is left unconnected to allow greater flexibility in its use.

An open-collector output, which is shown in Figure 1–5, performs roughly the same function as a switch contact with one side of the switch connected to ground. When the output is turned OFF, no current can flow through the transistor. This is equivalent to an open switch contact because the device being controlled is turned OFF.

Figure 1–5. Open-Collector Output



When the output is turned on, current flows through the transistor, which is equivalent to a closed switch contact. The controlled device turns on in response to the flow of current.

Connect a device to an open-collector output by connecting one of the device's terminals to the output and the other terminal to the positive side of the power supply. If the device is polarized, connect its negative [-] terminal to the output.



#### Note

Control Tech recommends that you place a suppression diode across inductive loads. Use a 1N4004 diode or its equivalent. Locate the diode as close to the load as possible. Refer to Figure 1–5 for more information.



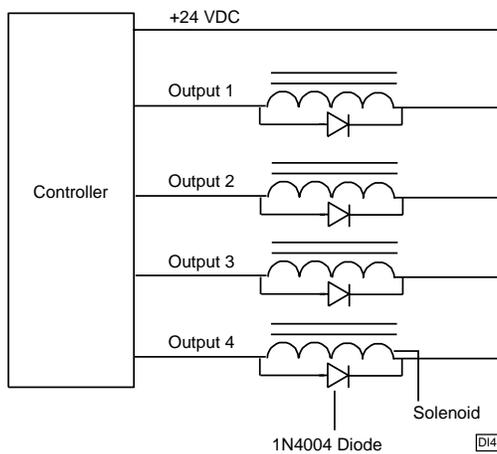
**Caution**

Do not exceed the rated current of the power supply in use. When you calculate your system's current requirements, you need to consider the maximum number of output devices that will be turned on simultaneously. Include 0.45 A for the MultiPro in your calculation.

**Connecting Multiple Devices**

It is possible to supply power to multiple devices from the same power source. One lead of each device is attached to an independent output and the other lead is connected to the positive [+] terminal of the power source. Figure 1–6 shows four solenoid valves that are controlled by Outputs 1-4. Power is supplied to each output from the MultiPro's power supply.

**Figure 1–6. Multiple Device Connection Diagram**



**Connecting to a Second External Supply**

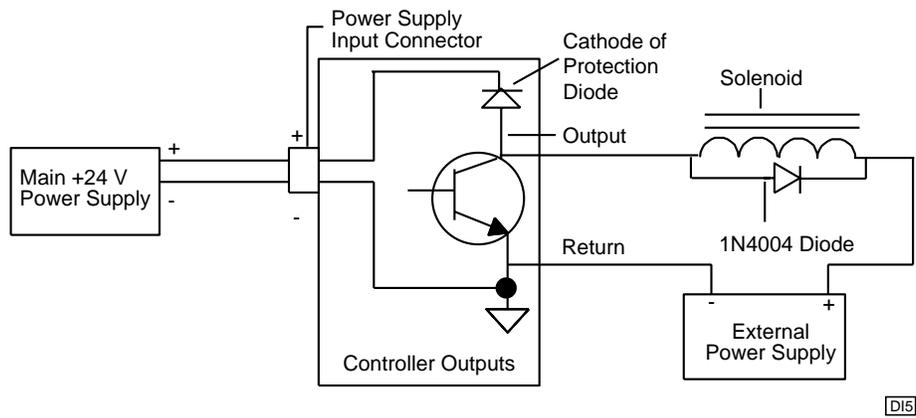
Multiple devices can also derive their power source from a secondary external power supply. Some devices use this external supply while others use the controller's power supply. If you use this option, connect each device to the positive [+] terminal of the appropriate power supply. Figure 1–7 illustrates this setup.



**Note**

If you decide to use an external power supply, do not tie the positive terminals of the two supplies together by direct means or indirect means.

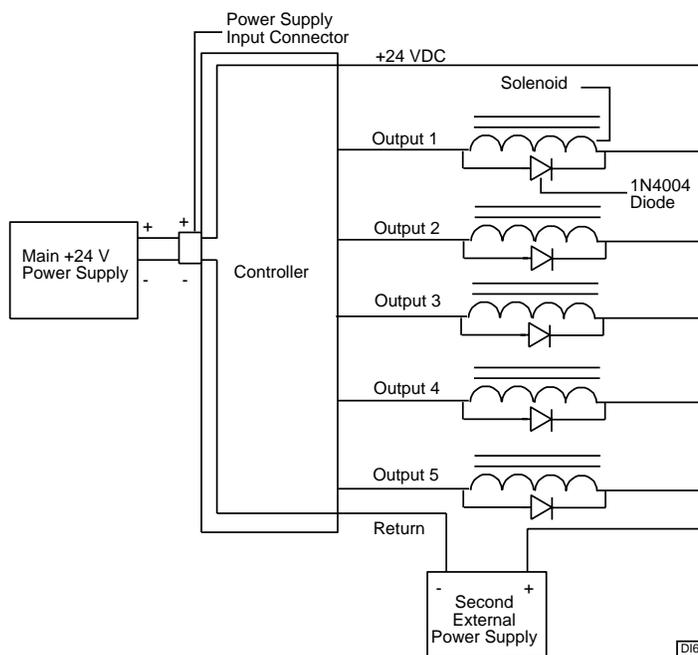
Figure 1–7. External Power Supply Connection Diagram



**Caution**

Each output has a protection diode with its cathode connected to the +24 VDC power supply through the input connector. This diode prevents damage to the output when it is connected to an inductive load. If you use an external supply as shown in Figure 1–7, a current path exists between the two supplies through the devices being controlled. Under normal circumstances, this practice is acceptable. However, some power supplies offer low impedance with respect to the power supply return when you turn them OFF. For example, in Figure 1–7, the main supply is turned OFF and the external supply is ON. Current from the external supply can energize the device connected to the output and turn it ON. To prevent this situation, make sure that both supplies are turned ON and OFF at the same time. Refer to Figure 1–8 for an alternative way to connect an external power supply.

Figure 1–8. Alternative Wiring Diagram for an External Power Supply





**Note**

Do not use an external power supply with an output voltage that exceeds the voltage rating of the MultiPro's outputs.

---



**Caution**

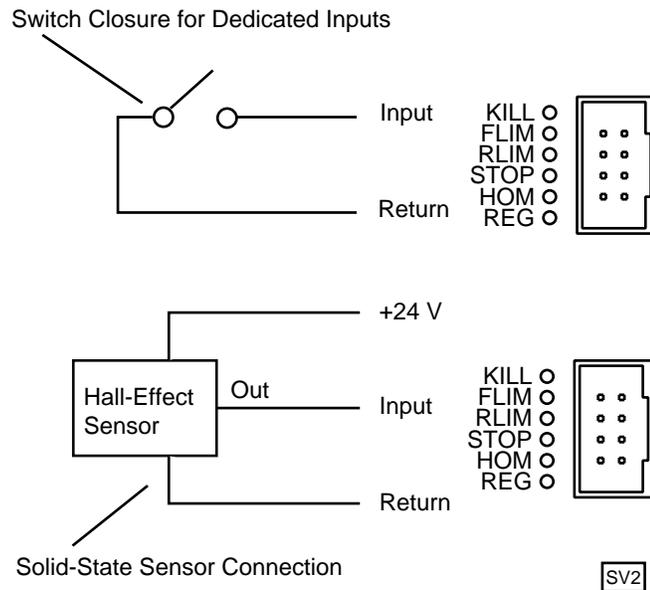
Do not connect the positive [+] terminals of the power supplies together! Damage to one of the supplies may result. Figure 1–8 shows the connection between the external supply's negative terminal and the output connector's return terminal. This provides a complete path for the current traveling through the device being controlled.

---

## Connecting Dedicated Inputs

The MultiPro has six dedicated inputs that you can activate with a switch closure to **Return**, which is the common for the MultiPro's 24 V supply. Each input is opto-isolated from the MultiPro's logic circuitry and is internally self-powered by the 24 V supply through a current-limiting resistor. Figure 1–9 shows the switch closure for the dedicated inputs.

**Figure 1–9. Dedicated Input Switch Closure**



### Using Solid-State Sensors

You can connect many types of electronic sensors such as three-wire Hall-effect sensors, proximity sensors, and phototransistors to the inputs without any additional circuitry. These devices must have sinking type open-collector outputs (NPN) and must be able to withstand at least +24 V on their output terminals when they are in the OFF state. The sensor must also be able to sink the required input current (i.e. - 2.1 mA) when ON.



#### Note

Do not use two-wire, solid-state sensors.

---

If the sensor requires a voltage that equals the MultiPro's built-in auxiliary voltage (24 V), the MultiPro can supply this voltage directly without requiring an additional external power supply.

# Computer - Controller Communications

The MultiPro's RS-232 port provides a way to download Quickstep programs and also supports data communications. You can add a second RS-232 port with an optional duplex adapter. Refer to *RS-232 Connections (Dual Ports)* for more information.

## RS-232 Communications

The MultiPro's RS-232 port allows the following activities:

- **Direct communications between a PC and the MultiPro's RS-232 port** - This feature enables you to directly interact with all the MultiPro's resources such as registers, inputs, outputs, and flags without modifying the MultiPro's program.
- **Monitoring** - You can monitor a controller's activity through an RS-232 port with CTCMON.
- **Host configuration** - The MultiPro is configurable as a host that can support communications with other external peripherals such as operator interface terminals, bar-code readers, printers, and other controllers. Refer to *Technote No. 30, ASCII Message Transmitting with CTC Controllers*, which is available in the Support section of our web site at [www.ctc-control.com](http://www.ctc-control.com).



### Note

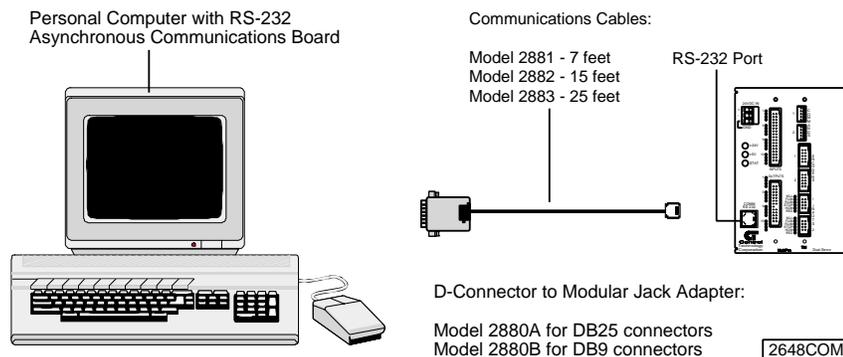
For more information on data communications and the DLL functions required to communicate with the MultiPro, refer to the *CTC 32-Bit Data Communications Functions Reference Guide*, which is available in the Support area of our web site at [www.ctc-control.com](http://www.ctc-control.com).

## RS-232 Connections (Single Port)

Connect to the RS-232 port through the modular jack (labeled COMM) on the MultiPro's front panel. This jack carries the receive and transmit signals and two commons (ground). Refer to Table 1-5 on page 11 for connection information on this jack.

Standard Control Technology cables are available for connecting to this jack (Figure 1-10). As an alternative, many commonly available telephone cables may be substituted.

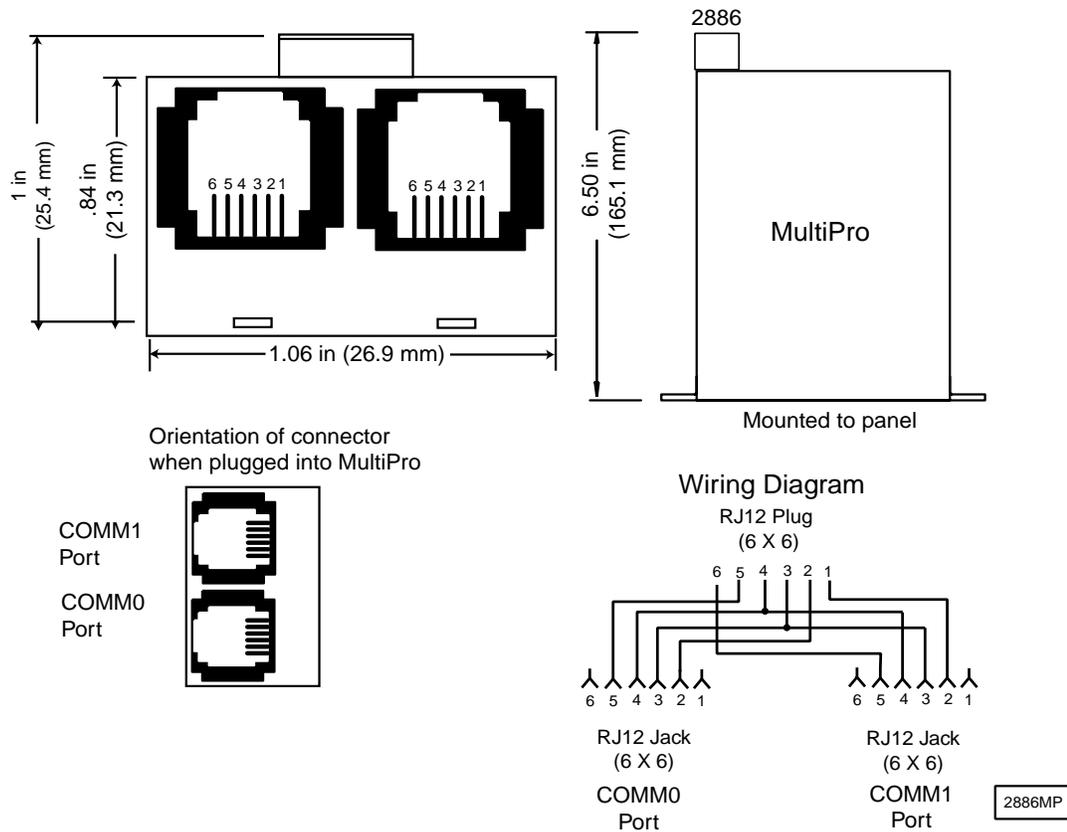
**Figure 1-10. Communication Cables and Connectors**



## RS-232 Connections (Dual Ports)

The MultiPro can operate with dual RS-232 ports by adding the Model 2886 duplex adapter. Connect through one of the adapter's modular jacks (labeled COMM1 and COMM0). These jacks carry receive and transmit signals for both ports as well as two commons (ground). Refer to Table 1-5 on page 11 for connection information on these jacks. Figure 1-11 shows the adapter's dimensions, its wiring diagram, and how it interfaces with the MultiPro.

Figure 1-11. Model 2886 Adapter



## Connecting to a D Connector

RS-232 ports on computers are usually configured through 25-pin (DB25) or 9-pin (DB9) D-type connectors. Most PC manufacturers use standard wiring on these connector types. Control Technology has adapters available that connect directly to a male DB25 (Model 2880A) or DB9 (Model 2880B) connector. These adapters have a modular jack that is wired for compatibility with the COMM port. To ensure full compatibility with these adapters, you should wire the computer's communications port as a DTE (Data Terminating Equipment) device.

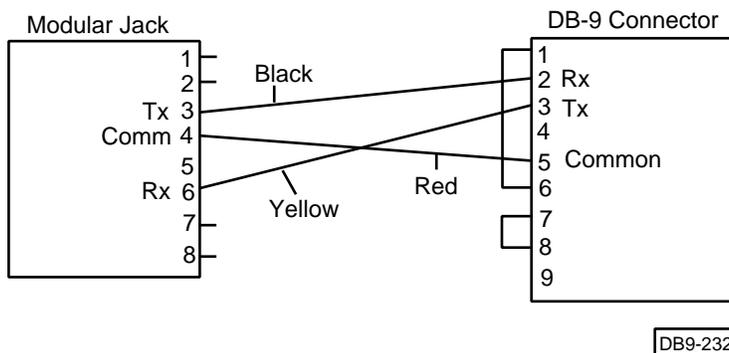


### Note

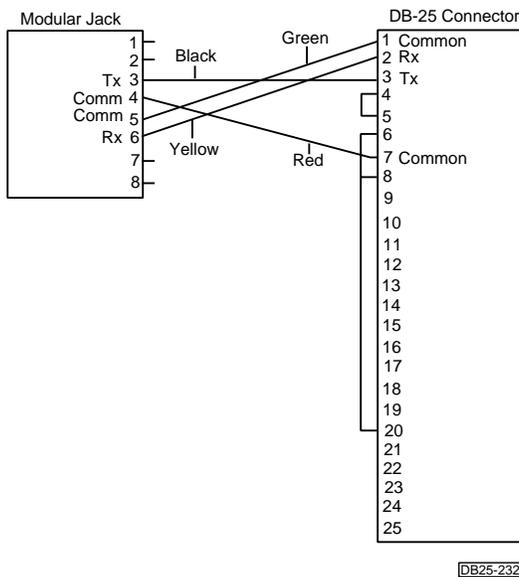
Do not connect the MultiPro to a telephone line.

Figures 1–12 and 1–13 show computer-controller connections using an RS-232 connection and DB25 and DB9 connectors.

**Figure 1–12. DB9 Connections**



**Figure 1–13. DB25 Connections**



This page is intentionally left blank.

# Application Notes

---

Using the Real-Time Clock .....	31
Programming a Servo .....	33
Setting Up Electronic Following .....	42
Setting Up Registration for a Servo .....	46
Servo Hardware Considerations .....	52
Sample Quickstep Programs .....	55

This page is intentionally left blank.

## Using the Real-Time Clock

The MultiPro includes a real-time clock (RTC) feature. The clock is preset at the factory to Eastern Standard Time. The registers listed below are used to read the current time or set the clock to local time:

- **Register 13014** - seconds
- **Register 13015** - minutes
- **Register 13016** - hours (24 hour clock)
- **Register 13017** - day of month
- **Register 13018** - month of year (1-12)
- **Register 13019** - year (two fields)
- **Register 13020** - day of week (1-7, where Sunday=1)



### Note

All registers are read/write.

## Examples

In this example, when the RTC reaches five minutes past the hour, the program goes to the next step.

```
[10] Clock_Test
    ;;;
    _____
    <No Change in Digital Outputs>
    _____
    if Minutes = 5 goto next
```

In this example, the program goes to the next step at 12:59:59 p.m. on December 31, 1999.

```
[20] Clock_Test
    _____
    <No Change in Digital Outputs>
    _____
    if Seconds <> 59 goto CLOCK_TEST
    if Minutes <> 59 goto CLOCK_TEST
    if Hours <> 23 goto CLOCK_TEST
    if Day <>31 goto CLOCK_TEST
    if Month <>12 goto CLOCK_TEST
    if Year <> 99 goto CLOCK_TEST
    goto HAPPY_MILLENIUM
```

In this example, when a specific event occurs it activates an input and the controller moves to the next step.

```
[50] Time_Stamp
-----
<No Change in Digital Outputs>
-----
monitor Event_Trap goto next
[51] Save_Data
-----
<No Change in Digital Outputs>
-----
store Seconds to Saved_Seconds
store Minutes to Saved_Minutes
store Hours to Saved_Hours
store Day to Saved_Day
store Month to Saved_Month
store Year to Saved_Year
```

## Programming a Servo

The following Quickstep instructions are used to program a servo motor:

- Profile Servo
- Turn Servo
- Monitor Servo
- Zero Servo
- Search and Zero Servo
- Stop Servo
- If Servo
- Store Servo



---

### Note

The registers mentioned in this chapter only apply to Axis 1. Refer to *Chapter 3, Special Registers*, for more information on other special registers.

---

## Setting Up Servo Motor Operating Parameters

The MultiPro must have a set of operating parameters before it can turn a servo motor. You must specify these parameters with the `PROFILE SERVO` instruction. The servo motor operating parameters are as follows:

- **Max Speed** — Establishes the maximum speed of the motor.
- **Accel Rate** — Specifies the acceleration rate of the motor. The deceleration rate is the same as the acceleration rate. Refer to *Setting Acceleration and Deceleration Values* for information on setting a different deceleration rate.
- **P Parameter** — The P parameter is the system gain. It specifies the factor applied to the sensed position error to create a correction signal. The gain factor is highly dependent on the gain of any external amplifier that is used to drive the actuator. Possible values range from 1 to 255.
- **I Parameter** — The I (integral) factor is used to obtain increased accuracy at low frequencies. It integrates, or builds up, a corrective signal in response to a steady-state error. A greater I factor causes the filter to build up a corrective signal for even small amounts of error and greatly increases the terminal accuracy of each move. Possible values range from 0 to 255.
- **D Parameter** — The D (derivative) factor senses and responds to rapidly changing rates of error and is most useful in increasing the system response to varying loads and friction at high speeds. Possible values range from 0 to 255.

- **Holding Mode** — Specifies the status of the servo when stopped using one of the following parameters:
  - *Servo at position* — Once the servo reaches the desired position, the actuator will continuously seek this position. If the actuator is forced from its position, the MultiPro sends a correction signal and attempts to correct the perceived error.
  - *Deadband of \_\_ at position* — The servo senses position errors but does not correct them unless the error is out of the range of the deadband. This parameter is specified with encoder counts.
  - *Off at position* — Once the servo reaches its position, no further corrective action occurs. This allows manual adjustment or another external force to change the servo's position.



## Notes

1. The maximum speed is expressed in units of steps-per-second (steps/s). The programmed maximum speed has a resolution of 1 step/s. Acceleration and deceleration are expressed in units of steps-per-second-per-second (steps/s<sup>2</sup>) with a granularity of 1 step/s<sup>2</sup>.
  2. The `PROFILE SERVO` instruction must appear before the first `TURN SERVO` instruction in your Quickstep program. If it is not executed before the first `TURN SERVO` instruction, a software fault stating, "Servo not ready," results. Additional `PROFILE SERVO` instructions are only necessary when you want to change the motor's operating parameters.
  3. Re-profiling on-the-fly, which allows the servo to take on new settings during a motor motion, is possible. To re-profile the servo, program another `PROFILE SERVO` instruction with a new maximum speed or acceleration value. You do not have to re-specify a value that does not change.
  4. Adjustments to the ramping (acceleration and deceleration) parameters while the servo is accelerating or decelerating causes an instantaneous change in the ramp that may be undesirable. To avoid this, make changes to the ramping parameters when the servo is stopped or is turning at maximum speed. You can view the status of the servo by checking the appropriate special registers. For example, check register number 14301 for the current status of the first servo. Refer to *Chapter 3, Special Registers*, for more information.
-

## Using Servo Filters

A servo filter is a high speed calculation that continuously commands a servo system's output. The MultiPro offers a variety of filters that perform this function. The filter you choose depends on the type of servo drive used in your application. If the default filter (PID) is not used, you must set the filter register associated with each axis before the initial profile instruction.

### PID Filter

The MultiPro's default filter setting is a calculation called PID (Proportional, Integral, Derivative). It is generally used with drives configured for **Torque**, or **Current**, mode. In this case, the command output (0 to ±10 VDC) represents zero to full current of your servo drive's output. The polarity of the command output governs your servo's direction of travel.

The difference between the actual position of a servo and the intended position is called servo error. This error is represented by encoder counts. At a rate of 2,048 times per second, the MultiPro's servo board uses the following equation to command the servo:

$$\text{Servo Output} = (\text{position\_error} * \text{User\_Proportional}) + [(\text{position\_error} - \text{last\_position\_error}) * \text{User\_Differential}] + (\text{cumulative\_error} * \text{User\_Integral})$$

The result of this calculation is scaled into the span of the servo board's analog output in the form of a new command signal. The MultiPro's servo board then adds the servo error to the cumulative error and records the servo error in preparation for the next calculation.

### PAV Filter

The PAV (Proportional, Acceleration-Feedforward, Velocity-Feedforward) filter is selected by storing a value of 5 to the filter register (Register 17001) before the initial profile instruction. This filter is generally used with drives configured for velocity mode. The servo board's analog command output (0 to ±10 VDC) represents zero to full velocity of the servo drive's and motor's capabilities (or configuration). The polarity of the command output governs your motor's direction of travel.

The MultiPro's servo board uses the following calculation when you specify the PAV filter:

$$\text{Servo Output} = (\text{position\_error} * \text{User\_Proportional}) + (\text{change\_in\_velocity} * \text{User\_AccelFF}) + (\text{current\_velocity} * \text{User\_VelocityFF})$$

The final result of this calculation is scaled into the span of the servo board's analog output in the form of a new command signal.

In this mode, the MultiPro ignores the I gain and the D gain in the profile instruction. However, you must assign values to these parameters when you write your Quickstep program for the compiler and for proper program operation. CTC recommends that you set these values to 0. The Feedforward parameters are set with special-purpose registers 14501 and 14801. Refer to *Chapter 3, Special Registers*, for a description of these registers.

## Direct Mode

You can set each axis of the MultiPro into direct mode for applications where a servo loop is not desired but you wish to command a velocity output. Set a register to a value between 0 and 32767 to command a 0 to 10 VDC output. Register 14501, which is the Velocity-Feedforward register, is used for this purpose. Refer to *Chapter 3, Special Registers*, for a description of this register.

To configure a servo axis into direct mode, you must store a value of 1 for counterclockwise direction (negative command signal) or a value of 2 for clockwise direction (positive command signal) before profiling the axis. Refer to the description of register 17001 in *Chapter 3, Special Registers*, for information on specifying servo direction with direct mode.

You must program a complete profile instruction in your Quickstep program and it must be executed to activate this feature.

## Sample Servo Motor Tuning Program

The following program is a sample program for tuning a servo motor. It consists of two tasks: `Servo_Error` and `Run_Servo`. `Servo_Error` monitors the servo error. If the error exceeds the specified value, it turns off the servo driver's output and stops the servo. `Run_Servo` tunes the servo with the P, I, and D parameters. It turns the servo clockwise and counterclockwise and allows a technician to adjust the three tuning factors.

```
[1] New_Servo_Program
    ;;; This program tunes a servo for Torque mode operation.
    ;;; It consists of two tasks: Servo_Error and Run_Servo.
    ;;; If your servo drive must be enabled by turning on an
    ;;; output from the MultiPro, you must specify and turn
    ;;; on the output in this step.
    -----
    <OUT_1_ON>
    -----
    profile servo_1 servo at position maxspeed=reg_501 accel=reg_502 P=reg_503
    I=reg_504 D=reg_505
    zero servo_1
    monitor in_1A goto Next

[2] Start_Tasks
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    do (Run_Servo Servo_Error) goto New_Servo_Program

[3] Servo_Error
    ;;; This task monitors servo error and shuts down the
    ;;; drive if servo error is too great. For tuning
    ;;; purposes, we use an error of 4000 encoder counts.
    ;;; For a 500 line encoder, this equates to two
    ;;; revolutions. After the servo is tuned, you may wish
    ;;; to reduce this servo error if you include such a
    ;;; task in your program.
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    store servo_1:error to reg_515
    store servo_1:position to reg_516
    if servo_1:error > 4000 goto Stop_Servo
    if servo_1:error < -4000 goto Stop_Servo
    goto Servo_Error
```

```

[4] Stop_Servo
    ;; In this program, we assume your servo drive must be
    ;; enabled by turning on an output. This step stops the
    ;; servo by sending a hard stop command and by turning
    ;; off the output that enabled the drive.
    -----
    <OUT_1_OFF>
    -----
    stop (hard) servo_1
    cancel other tasks
    monitor servo_1:stopped goto New_Servo_Program

[5] Run_Servo
    ;; This step turns the servo clockwise. While the servo
    ;; is in motion, it can be tuned by programming the tuning
    ;; parameters to access registers. The program executes
    ;; a clockwise turn followed by a counterclockwise
    ;; return. The tuning process is as follows:
    ;; 1. Set the P parameter to 1, the I parameter to 0, and
    ;; the D parameter to 0.
    ;;
    ;; 2. Set switch 1 and watch/listen to the servo. It
    ;; should turn but it will be mushy.
    ;; 3. While it is turning, increase the D parameter in
    ;; increments of 10 up to a maximum of 255 until the
    ;; servo stabilizes.
    ;; 4. Increase the P parameter until the servo becomes
    ;; unstable, then reduce it until the servo becomes
    ;; stabilized.
    ;; 5. While monitoring the servo error, increase the I
    ;; parameter to minimize the servo error to the point
    ;; where the servo becomes unstable, then reduce it until
    ;; the servo stabilizes. Your servo is now tuned!
    ;;
    ;; NOTE: Please insure you have loaded the appropriate
    ;; registers with valid values before switch No.1
    ;; is set.
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    profile servo_1 maxspeed=reg_501 accel=reg_502 P=reg_503 I=reg_504 D=reg_505
    turn servo_1 to 4000
    monitor servo_1:stopped goto Next

[6] Delay_Step
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    delay 1 sec goto Next

[7] Reverse_Direction
    ;; This step reverses direction of the servo and returns
    ;; it to the starting position.
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    profile servo_1 maxspeed=reg_501 accel=reg_502 P=reg_503 I=reg_504 D=reg_505
    turn servo_1 to 0
    monitor servo_1:stopped goto Next

[10] Delay_2
    ;; After the delay, the program returns to the
    ;; Run_Servo step and starts the cycle over. Since
    ;; the Run_Servo step specifies the servo profile, we can
    ;; change the P,I, and D parameters to optimize motor
    ;; performance.
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    delay 1 sec goto Run_Servo
    
```

## Setting Acceleration and Deceleration Values

The `PROFILE SERVO` instruction acceleration parameter sets both the acceleration and deceleration values. If you want the acceleration and deceleration values to be different, use one of the group or individual access special purpose registers to set a different deceleration value. For example:

```
profile servo_1 max=50000 accel=100000  
store 20000 to reg_15006 (axis No. 1 deceleration register)
```

sets the acceleration equal to 100,000 steps/s<sup>2</sup> and the deceleration equal to 20,000 steps/s<sup>2</sup>. Refer to *Chapter 3, Special Registers*, for the appropriate register number for each axis.



### Note

If you specify a new acceleration rate, it overwrites the existing deceleration rate. Therefore, you must specify a new deceleration rate.

---

## Searching for Home

Each servo axis has a dedicated home input. This input is used in conjunction with the `SEARCH AND ZERO` instruction to set a home position for the axis. When home is sensed, the servo stops and the position is set to zero.

The MultiPro supports a highly accurate method of finding the home position. In addition to providing direct support for a two-stage homing routine, the MultiPro also makes use of the index signal available on many encoders to further increase the consistency of the home position. The encoder has a connector that provides an additional input for each axis on the MultiPro to accept the index signal.

The servo travels in a counterclockwise (default) direction unless otherwise specified. You can change this default setting with a special purpose register. Refer to *Chapter 3, Special Registers*, for more information. The homing sequence is as follows:

1. When the MultiPro executes a `SEARCH AND ZERO SERVO` instruction, the servo begins searching in a counterclockwise direction at the acceleration rate and maxspeed specified in the most recent `PROFILE` instruction.
2. When the home input closes (turns on), the servo stops at the profiled deceleration rate.
3. The servo then automatically begins searching in a clockwise direction at a fixed speed of 950 steps per second.

4. When the home input turns on again, the speed decreases to 192 steps per second.
5. When the home input opens (turns off), the servo hard stops.
6. If the encoder's index marker signal is connected to the index input on the module (this is automatically sensed), the servo begins searching in the counterclockwise direction at a speed of 192 steps per second.
7. When the index marker is sensed, the servo hard stops and the position is set to zero.

## Specifying the Homing Direction

You can reverse the direction of the homing motions described above by storing the number 1 to special purpose register 17003. You can restore the default setting mentioned above by storing 0 or -1 to the register.

## Turning a Servo

There are three modes of turning the servo:

1. **Absolute Positioning** – In this mode, the MultiPro's servo board always references the home (or zero) position in a turn instruction and moves a specified distance from the home position. For example, the following instruction

```
turn servo_1 to 50000
```

causes the servo to position itself 50,000 steps from home. The servo automatically turns in the correct direction to reach the new position.

2. **Relative Positioning** – In this mode, the direction of the turn (clockwise or counterclockwise) is specified in the turn instruction along with a defined number of steps to turn. For example, the following instruction

```
turn servo_1 cw 12340 steps
```

turns the servo 12,340 steps clockwise from its current position.

3. **Velocity Control** – In this mode, you establish a direction and begin continuous operation. The maximum speed and acceleration are based on the current profile instruction and can be changed. For example, the following instruction

```
turn servo_1 cw
```

starts the servo turning clockwise at its current maximum speed and acceleration. The servo continues to turn until the MultiPro issues a `STOP SERVO` instruction or until a Limit or Stop input is activated.

Once a servo is in motion, do not initiate another turn or zero instruction until the motion is complete or the “servo not ready” software fault occurs. Use the `MONITOR SERVO` instruction to check the current status (running/stopped) of the servo.

The MultiPro's servo board tracks the position of the servo with all three modes and allows you to use all three types of positioning and control in the same program.



## Note

Quickstep instructions specifying clockwise or counterclockwise operation assume that the servo is wired according to the manufacturer's recommendations and that the logical sense of the direction output of the MultiPro's servo board agrees with the logical sense expected by the servo's drive.

## Stopping the Servo

There are two instructions that terminate the motion of a servo already in motion:

- `STOP (SOFT) SERVO` causes the servo to stop at the deceleration rate specified in the last profile instruction.
- `STOP (HARD) SERVO` causes the MultiPro's servo board to try to stop the servo instantly. However, because of momentum, the servo may not stop instantly.

In either case, you should use a `MONITOR SERVO STOPPED` instruction before issuing another turn instruction.

## Monitoring and Changing Other Servo Parameters

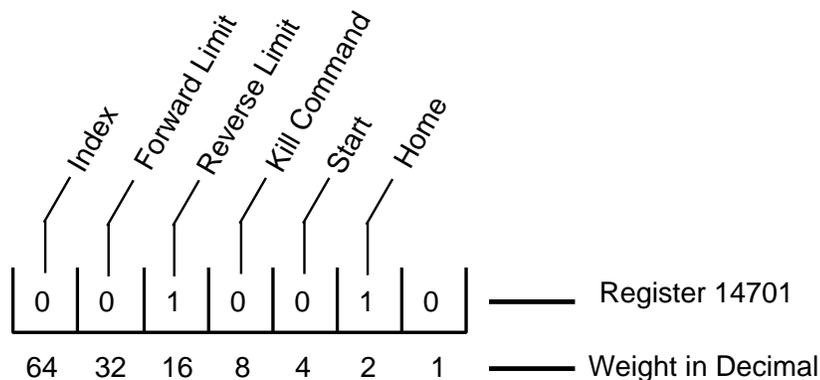
There are various special purpose registers that allow you to monitor and change the servo parameters. For more information, refer to *Chapter 3, Special Registers*.

## Monitoring Dedicated Inputs

Group access register 14701 or individual access register 15007 can return a bit pattern that indicates if any of the dedicated inputs are active. A binary representation of the dedicated input number is stored in the register. Each input has a unique binary value.

Figure 2-1 shows that the home and reverse limit inputs are active. Register 14701 returns a value of 18 because the respective weights of the inputs are 2 and 16. To test any individual input, use the bitwise `AND` instruction to apply a mask to the register.

Figure 2-1. Register 14701 Bit Pattern



SV1

The following instruction applies a bit mask that tests to see if the Home input is active:

```
[1] TEST_FOR_HOME_AXIS1  
    ;; Home = 2
```

```
-----  
<NO CHANGE IN DIGITAL OUTPUTS>
```

```
-----  
store reg_14701 and 2 to reg_10  
if reg_10 = 2 goto FOUND_HOME  
goto TEST_FOR_HOME_AXIS1
```

The Index bit (64) is inverted. If you are not using the index marker on your encoder, then the MultiPro sets this bit continuously. If you are using the index marker, the MultiPro only sets this bit when the encoder position reaches the index position. This occurs only once per encoder revolution.

## Setting Up Electronic Following

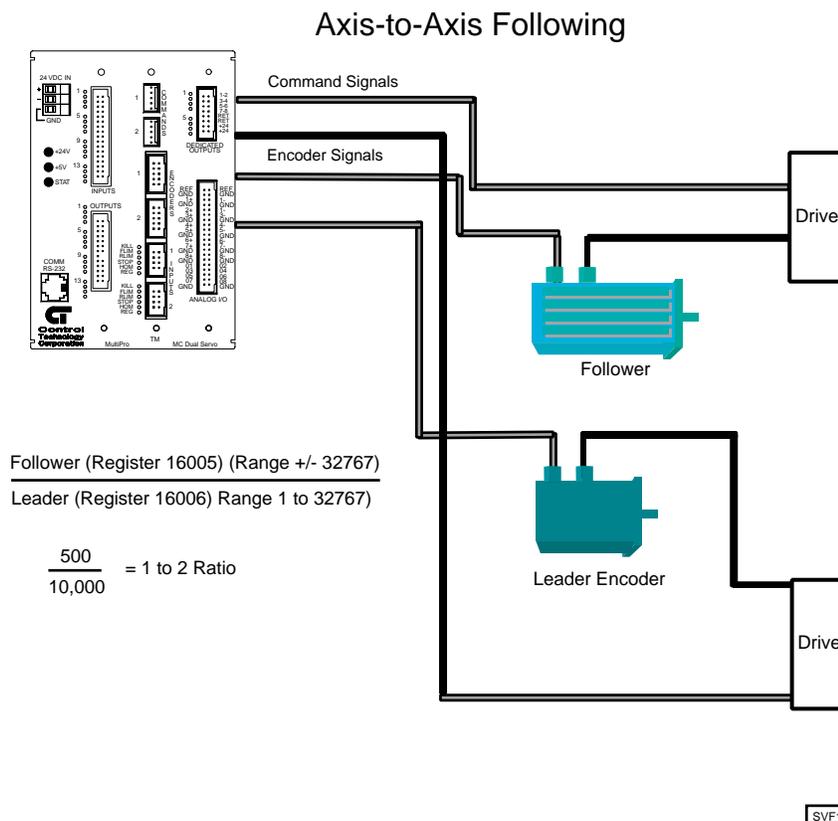
The MultiPro can perform a function called electronic following, or ratioing. In electronic following, one servo axis is commanded to match its motions to a leader or encoder based on a specific ratio. You can create this ratioing function with two special registers and two simple Quickstep instructions. You can even adjust the ratios on the fly within your Quickstep program.

### Dual Servo Axis-to-Axis Following

You can configure the MultiPro for axis-to-axis following (Figure 2–2). The first axis is always the follower and the second axis is always the leader. There are two types of following modes:

- **Trajectory Following** - This mode is the default mode. It creates a ratio mode that is based on the leader servo's theoretical (calculated) position, not its actual position. This mode causes the follower axis to be in phase with the leader axis, which results in a closer match that is based on the defined ratio.
- **Encoder Following** - This mode causes the follower to use the leader's encoder information to perform its ratio.

Figure 2–2. Axis-to-Axis Encoder Following



## Configuring Electronic Following

Configure an axis for following by setting two special purpose registers for the follower axis as if you were defining a fraction. The first register specifies the numerator and represents the follower axis. The second register specifies the denominator and represents the leader axis. You must decide how to describe the fraction since a 1/1 fraction and a 10,000/10,000 fraction both define a 1:1 ratio. However, to achieve better resolution in your application, you may want to use more decimal places in the fraction. For example, defining a fraction of 9,978/10,000 causes the follower to be geared slightly lower than the leader.



### Note

Special register 16005 specifies the numerator and register 16006 specifies the denominator. You should store the denominator value before the numerator value because storing the numerator activates electronic following.

You must define a complete `PROFILE SERVO` instruction with working tuning parameters for the follower axis before storing the values to the follower's special purpose registers for ratioing. Once the values are stored, the follower is engaged and begins following the leader. While it is engaged, the follower's status register (register 14301 or 15003) contains the number 10, which indicates that it is following its leader.

When you activate the follower axis and you are encoder following, the servo board automatically resets the leader's position to zero. You cannot reset the leader's position with a Quickstep instruction.



### Notes

1. The maximum range of values for the fraction is  $\pm 32767/32767$ . The sign of the numerator represents the direction the follower will travel with respect to the leader.
2. The servo board automatically accumulates and adjusts for fractional remainders to maintain synchronization between the follower and the leader.

## Ending Electronic Following

You can disengage the axis from following the leader by storing a 0 to the numerator. This causes the axis to decelerate to a stop at the profiled deceleration. You can also execute a `STOP SERVO` (soft) or (hard) instruction. If you have also programmed your servo for a registration move, a valid registration input with an offset value causes the follower axis to depart from the leader. The follower axis then begins the offset move and later comes to a stop.

## Reading Current Position and Velocity

Special register 16007 specifies the current leader position and special register 16008 specifies the current leader velocity registers. Leader position is expressed in encoder pulses and leader velocity is expressed as encoder pulse-per-second. These registers are read-only. You can use them to monitor real-time leader activities from within your Quickstep program. They are updated approximately every 250 ms.

The following examples show how to set up axis-to-axis and encoder following:

```
[11] AXIS_TO_AXIS_FOLLOWING
    ;;
    ;; Example Axis-to-Axis following program.
    ;;
    ;; Follower_Servo_1 is the follower axis.
    ;; Leader_Servo_2 is the leader axis.
    ;; In this example, the follower will follow
    ;; the leader at a 1:2 ratio
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    profile Follower_Servo_1 servo at position maxspeed=
      Max accel=Accel P=Pval I=Ival D=Dval
    profile Leader_Servo_2 servo at position maxspeed=
      Max accel=Accel P=Pval I=Ival D=Dval
    store 10000 to Denominator_r16006
    store 5000 to Numerator_r16005

[15] ENCODER_FOLLOWING
    ;;
    ;; Example encoder following program.
    ;;
    ;; Follower_Servo_1 is the follower axis.
    ;; In this example, the follower will follow
    ;; the leader at a 1:10 ratio
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    profile Follower_Servo_1 servo at position maxspeed=
      Max accel=Accel P=Pval I=Ival D=Dval
    store 10000 to Denominator_r16006
    store 1000 to Numerator_r16005
```

## Specifying Encoder Following

Store the number 128 plus the servo filter type code to the servo filter register (17001) to select the encoder following mode within axis-to-axis following. For example, the command

```
store 133 to register 17001
```

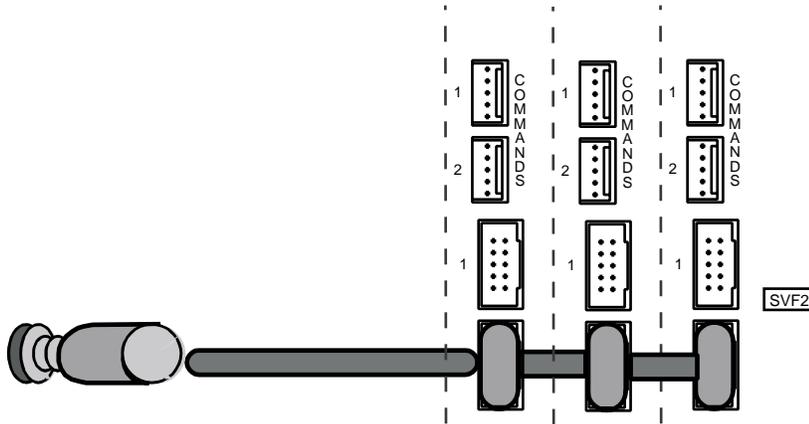
specifies the PAV filter mode for the first servo axis (5 + 128) and tells it to follow its leader in the encoder following mode.

## Interconnecting Multiple Followers With the Same Leader

In applications that require one leader with multiple followers, it is possible to daisy-chain the leader encoder signals across a maximum of three MultiPro encoder inputs (Figure 2–3). To daisy-chain more followers, you need an encoder expander board to distribute the encoder signals.

Figure 2–3. Multiple Followers

Partial view of MultiPro encoder sections interconnected as multiple followers



## Setting Up Registration for a Servo

The MultiPro is set up with a special registration input. The registration input works in conjunction with a series of special registers. If you have an application that requires automatic synchronization of a product from one cycle to another, the registration input and the special registers provide a method of recording real-time position information. Both axes have the ability to record the absolute position of the servo when the registration input is activated. The MultiPro's servo board also has the ability to change the current servo motion and adjust the end position of the move for reliable synchronization. Registration on the MultiPro is so accurate that the servo's absolute position is captured with a resolution of  $\pm 1$  encoder count (step) regardless of the servo's velocity.

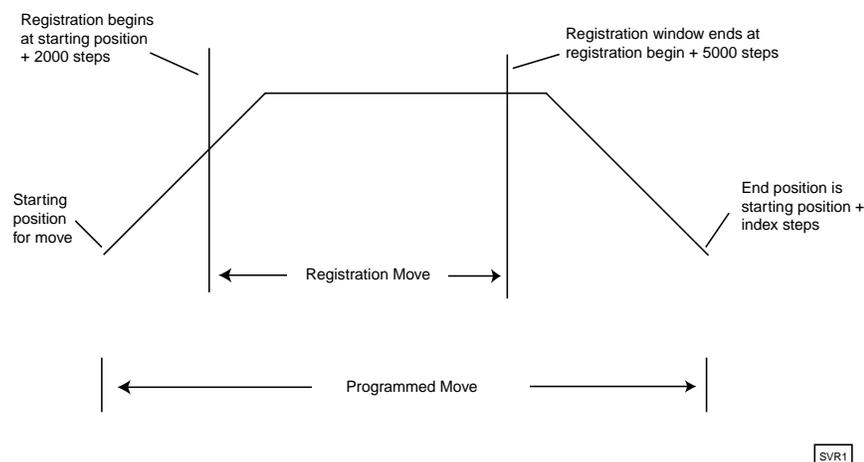
### Designating a Predefined Registration Window

The registration input is usually connected to some type of electronic sensor or photo eye. To make use of the registration feature, you need to define the window where the servo expects the electronic sensor to activate the registration input.

A predefined registration window (Figure 2–4) tells the servo where to look in its move for the sensor's input. The servo's position is only captured if the sensor triggered the registration input in this window. The MultiPro's servo board does not record the servo's position if the sensor triggers the registration input outside of this window. This prevents other events from triggering registration. You define the size and range of the registration window with the special registers set up for this purpose. In the following example, the servo is programmed to move a specific distance (labeled Index). The registration window for servo 1 is defined as 5000 steps long. The servo begins looking for a registration input when its absolute position is at 2000 steps and ends when its absolute position is at 7000 steps. To define this window, enter the following values in registers 16000 and 16001:

- Register 16000 is set to 2000, which means that the registration window begins at 2000 steps from the beginning of servo 1's move.
- Register 16001 is set to 5000, which means that after the servo travels another 5000 steps, the registration window ends.

**Figure 2–4. Predefined Registration Window**



If the sensor is triggered during the registration window, the MultiPro records the absolute position of the servo in register 16002. If the sensor is triggered outside of the registration window, the MultiPro does not record the servo's position.

When the MultiPro records the servo's absolute position, it also sets the value in register 16004 to 1. As long as the value in register 16004 is 1, the absolute position of the servo where registration occurred is locked into register 16002. The MultiPro does not change the value in register 16002 until the value in register 16004 is reset to zero by your Quickstep program. Resetting register 16004 re-arms registration for the next move. Refer to *Chapter 3, Special Registers* for a list of special registers.

The following program shows how to set up and use a predefined registration window:

```
[1] REGISTRATION_EXAMPLE
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    store 5000 to Reg_16001
    goto Next

[2] REGISTRATION_MOVE
    ;;
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    profile Servo_1 servo at position maxspeed=Reg_501 accel=
      Reg_502 P=Reg_503 I=Reg_504 D=Reg_505
    store Servo_1:position + 2000 to Reg_16000
    store 0 to Reg_16004
    turn Servo_1 ccw Index steps
    monitor Servo_1:stopped goto Next

[3] REGISTRATION_CHECK
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    if Reg_16004=1 goto GOT_REGISTRATION
    delay Reg_100 sec goto REGISTRATION_MOVE

[4] GOT_REGISTRATION
    ;;
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    store Reg_16002 to Reg_10
```

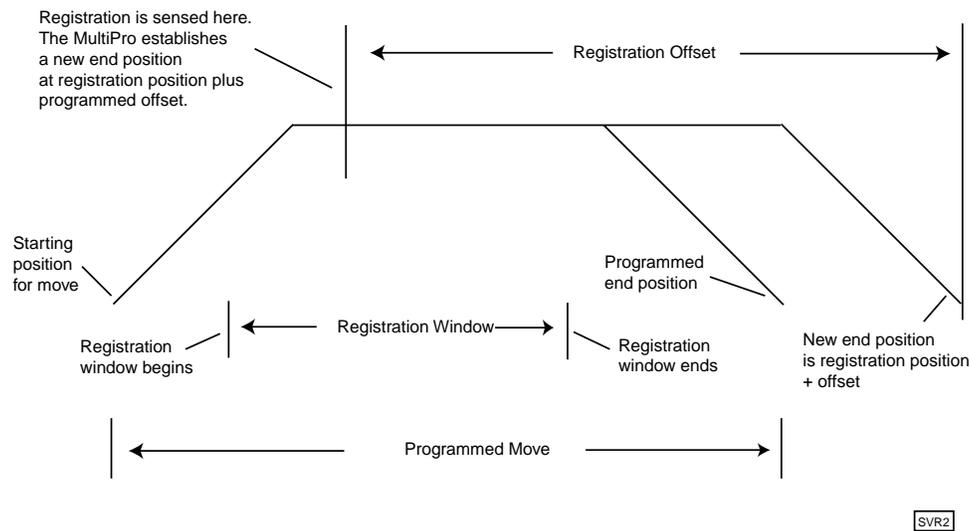
## Using Registration to Change the End Position of a Move

Some applications require registration to change the end target position of the servo's move while the servo is still in motion. The MultiPro allows you to program an offset position that you can add to the captured registration position. It uses this new position to redefine the stopping point for the current motion and overwrites the original programmed destination. This allows for precise correction to your motion based on when registration was sensed.

In the following example, if registration is sensed within the registration window, the servo defines a new end position by adding the number of steps defined in the offset register (register 16003) to the position where the sensor was triggered (Figure 2–5).

- Register 16000 is set to 2000, which means that the registration window begins at 2000 steps from the beginning of servo 1's move.
- Register 16001 is set to 5000, which means that after the servo travels another 5000 steps, the registration window ends.
- Register 16003 is set to 7500, which means that the new end position for the move is 7500 steps after the servo senses the registration input.

**Figure 2–5. Changing the End Position of a Move**



When the sensor is triggered during the registration window, the MultiPro operates as follows:

- Records the absolute position of the servo in register 16002 (for axis 1).
- Calculates a new end position for the servo by adding the position where the registration sensor was triggered (stored in register 16002) and the offset position in register 16003.
- Sets the value in register 16004 to 1. As long as the value in register 16004 is 1, the absolute position of the servo where registration occurred is locked into register 16002.

The MultiPro does not change the value in register 16002 until the value in register 16004 is reset to zero by your Quickstep program. Resetting register 16004 re-arms registration for the next move. During registration, the deceleration rate for the servo is always the rate you specified.

The following program shows how to set up and use a registration offset window:

```
[1] REGISTRATION_EXAMPLE
    ;;
    ;; Here, we program the registration window for 5000
    ;; steps and program the registration offset for 7500
    ;; steps.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    store 5000 to Reg_16001
    store 7500 to Reg_16003
    goto Next

[2] REGISTRATION_MOVE
    ;;
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile Servo_1 servo at position maxspeed=Reg_501 accel=
      Reg_502 P=Reg_503 I=Reg_504 D=Reg_505
    store Servo_1:position + 2000 to Reg_16000
    store 0 to Reg_16004
    turn Servo_1 ccw Index steps
    monitor Servo_1:stopped goto Next

[3] REGISTRATION_CHECK
    ;;
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    if Reg_16004=1 goto GOT_REGISTRATION
    delay Reg_100 min goto REGISTRATION_MOVE

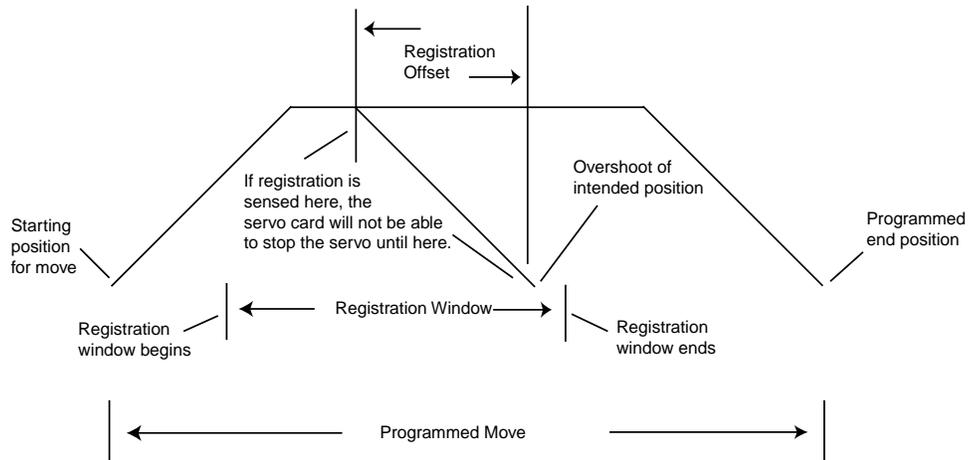
[4] GOT_REGISTRATION
    ;;
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    store Reg_16002 to Reg_10
```

## Overshooting the End Position

When you program a registration offset, the deceleration rate is always the rate you programmed. This makes it possible to write a Quickstep program so that the servo can overshoot the intended end position once the offset is taken into account. Figure 2–6 shows a case where the servo is unable to decelerate in time to stop at the new end position. Avoid this problem by using one of the following methods:

- Raise the deceleration rate so that the servo can reach the desired offset position.
- Lengthen the registration offset value.

Figure 2–6. Overshooting the End Position

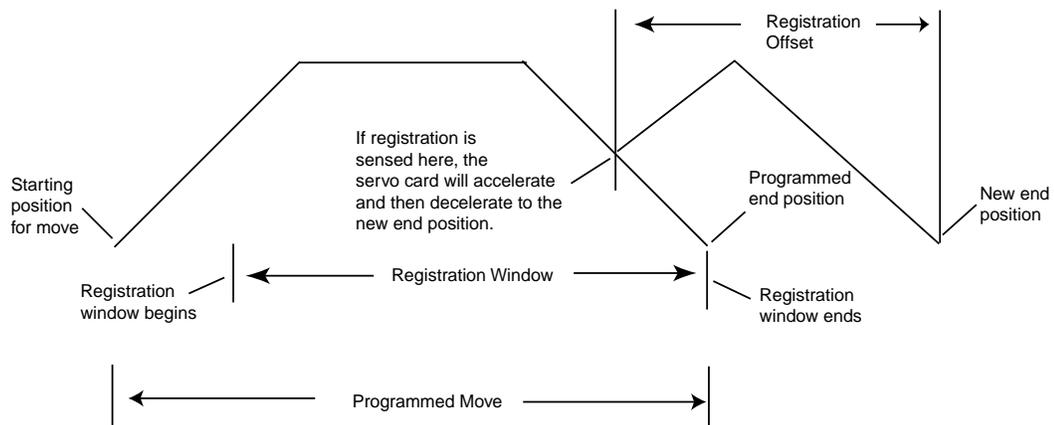


SVR3

## Registration during Deceleration

There are cases where the MultiPro senses registration during the deceleration portion of a programmed move (Figure 2–7). In the following example, the servo re-accelerates when it senses registration and then decelerates to a stop at the new end position. If there is sufficient distance available before the new end position is reached, the servo either accelerates to the maximum speed or accelerates until it is time to decelerate before coming to a stop at the new end position. This is an automatic function.

Figure 2–7. Registration during the Deceleration Portion of a Programmed Move



SVR4

## Guidelines and Rules for Setting up Registration

The following rules and guidelines make it easier to program your MultiPro for accurate registration:

- Make sure the registration offset value reflects the direction the servo is traveling. A positive value represents a clockwise direction and a negative value represents a counterclockwise direction. Failure to take the direction into account results in your servo becoming uninitialized at the point where registration is triggered.
- Inhibit the registration offset function by using a STORE instruction to set it to zero.
- The MultiPro senses registration when the state of your sensor changes to the opposite state.

## Servo Hardware Considerations

This section discusses the MultiPro's dedicated inputs and how to enable a servo drive.

### Dedicated Inputs

All of the MultiPro servo board's dedicated inputs are internally pulled up to 24 VDC and require a contact closure to 24 V Return to actuate. You can supply this closure with a mechanical switch or with certain types of solid state open-collector outputs. The dedicated inputs are as follows:

- **Forward Limit** – If the servo is moving clockwise, this input disables all clockwise movement and attempts to stop the servo instantaneously.
- **Reverse Limit** – If the servo is moving counterclockwise, this input disables all counterclockwise movement and attempts to stop the servo instantaneously.



#### Note

The MultiPro's servo board must have full closed loop control before Forward Limit and Reverse Limit can take effect. The forward and reverse limits do not stop a motor that has lost its encoder feedback.

---

- **Kill** – This input places the MultiPro's servo board in an uninitialized state that sets the command output to 0 VDC. The contact output for this axis opens and disables your servo drive. Refer to *Enabling Servo Drives* for more information on the Kill input.



#### Caution

In critical or dangerous applications, you should use external means to implement an E-STOP function.

---

- **Home** – This input is used to establish a home (zero) reference point for absolute positioning. Refer to *Searching for Home* for more information on this input.
- **Start** – Any motion may optionally be programmed to wait for this input. To use the Start dedicated input, you must program your `TURN SERVO` instruction with the `ON START` parameter. The MultiPro servo checks the On Start input every 488  $\mu$ s if the other axis is not turning or every 4 ms if it is turning. This improves the repeatability of the start motion. The motion does not begin until the Start switch is activated.

You can also re-direct the On-Start feature to occur when the leader servo reaches a user-defined position. To choose this option, store a 1 in register 13801 and store the absolute leader position (where you want the motion to start) in register 13901. You should then program your `TURN SERVO` instruction for the servo with the On Start

parameter. The motion commences when the leader’s position reaches the set point value in register 13901.

- **Registration** – This input accurately captures the servo position and can also alter the move. Refer to *Setting Up Registration For a Servo* for more information.



**Note**

If a servo has been stopped by any of the above stop inputs, the programmed instruction MONITOR SERVO STOPPED becomes true and your program proceeds to the specified step.

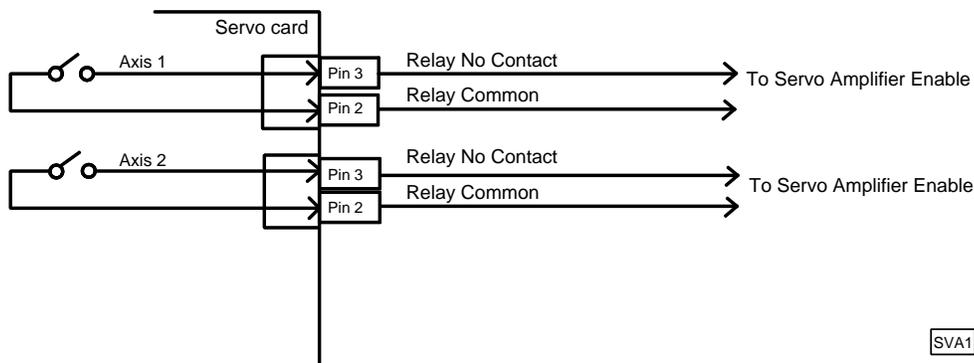
The dedicated inputs are not all active at the same time. The following list describes when the dedicated inputs are active and inactive.

- **Forward Limit input** – This input is active only when the servo is moving in the clockwise direction. This allows the servo to back off of a limit switch because you can still move counterclockwise. For example, if the Forward Limit input is connected to a limit switch for a linear table and the table hits the limit switch, the table is able to move counterclockwise and return to the home position.
- **Reverse Limit input** – This input is active only when the servo is moving in the counterclockwise direction.
- **Kill input** – This input is active at all times.
- **Start input** – This input is active when you have programmed a TURN SERVO instruction with an ON START parameter.
- **Registration** – This input is active only when the registration feature is active.

**Enabling Servo Drives**

Figure 2–8 shows how to connect the command output to a servo amplifier’s enable input. This is CTC’s recommended method for inhibiting an external servo drive.

**Figure 2–8. Connecting the Command Output to a Servo Amplifier**



The enable relay closes when the MultiPro executes a `PROFILE SERVO` instruction containing a `SERVO AT POSITION` parameter for the holding mode. The enable relay remains closed unless one of the following actions occur:

- The servo error exceeds  $\pm 32768$ .
- The kill dedicated input closes.
- The MultiPro executes a `PROFILE SERVO` instruction containing a `MOTOR OFF` parameter for the holding mode and stops the servo.



### Note

The servo axis becomes uninitialized when the servo error exceeds the limit or the kill input closes. In this case, you must execute a new complete `PROFILE SERVO` instruction before turning the servo.

---

## Sample Quickstep Programs

This section contains sample Quickstep programs for different servo applications.



### Note

Maxspeed units are in steps/s. Acceleration units are in steps/s<sup>2</sup>.

### Example 1 — Absolute Move of One Servo Motor

This example shows a servo motor moving 100,000 steps from its home position. The `monitor servo_1:stopped` instruction causes the MultiPro's program to remain in this step until the motor completes the move.

```
[1] ONE_AXIS_ABSOLUTE_MOVE
    ;;; This program will commence an absolute move on axis
    ;;; one based on the parameters in the profile
    ;;; instruction.

    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____

    profile servo_1 maxspeed=50000 accel=100000
    turn servo_1 to 100000
    monitor servo_1:stopped goto next
```

### Example 2 — Relative Move of One Servo Motor

This example shows a servo motor moving clockwise 100,000 steps from its current position. The `monitor servo_1:stopped` instruction causes the MultiPro's program to remain in this step until the motor completes the move.

```
[1] ONE_AXIS_RELATIVE_MOVE
    ;;; This program will commence an relative move on axis
    ;;; one based on the parameters in the profile
    ;;; instruction.

    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____

    profile servo_1 maxspeed=50000 accel=100000
    turn servo_1 cw 100000 steps
    monitor servo_1:stopped goto next
```

### Example 3 — Velocity Move of One Servo Motor

This example shows a servo motor moving clockwise from its current position. The motor turns until it receives a `STOP SERVO` instruction or until a stop input is activated. The `monitor servo_1:stopped` instruction causes the MultiPro's program to remain in this step until the motor completes the move.

```
[1] ONE_AXIS_VELOCITY_MOVE
    ;; This program will commence a velocity move on axis one
    ;; based on the parameters in the profile instruction.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=50000 accel=100000
    turn servo_1 cw
    monitor servo_1:stopped goto next
```

### Example 4 — Changing the Velocity of a Servo Motor During Motion

This sample program positions a servo motor and generates various velocity profiles throughout the move. After the initial parameters are set, the motor motion is started. When the position reaches 50,000 steps, the program continues to the next step. Each subsequent step changes the velocity and specifies the servo position where the program moves to the next step.

```
[1] COMPLEX_PROFILE
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=10000 accel=200000
    turn servo_1 to 500000
    if servo_1:position >= 50000 goto next

[2] SECOND_PROFILE
    ;; Re-profile the motor for the a new velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=20000
    if servo_1:position >= 70000 goto next

[3] THIRD_PROFILE
    ;; Re-profile the servo for a new velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=50000
    if servo_1:position >= 110000 goto next

[4] FOURTH_PROFILE
    ;; Re-profile the servo for a new velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=100000
    if servo_1:position >= 300000 goto next

[5] FIFTH_PROFILE
    ;; Re-profile the servo for the next velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=80000
    if servo_1:position >= 420000 goto next

[6] SIXTH_PROFILE
    ;; Re-profile the servo for the final velocity and wait
    ;; for the move to complete.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=30000
    monitor servo_1:stopped goto PROFILE_COMPLETE
```

## Example 5 — Ratio Axis to Leader Encoder

Register 16005 contains the ratio numerator and register 16006 contains the ratio denominator for the follower axis.

```
[1] RATIO_AXIS_TO LEADER_ENCODER
    ;;; This example sets up a ratio of Axis 1 to Axis 2.
    ;;; Axis 1 is the follower axis and Axis 2 is the
    ;;; leader axis. This step sets up the initial
    ;;; parameters for the servo motion and programs a ratio
    ;;; for Axis 1. The ratio is 2 to 1. From this point on,
    ;;; Axis 1 follows Axis 2 positions and velocity
    ;;; activities using the specified ratio.

    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____

    profile servo_1 servo at position maxspeed
    store 10000 to reg_16006
    store 5000 to reg_16005
    goto next
```

## Example 6 — Velocity Move of Two Servo Motors

This example shows two servo motors moving clockwise from their current positions. The motors will turn until they receive a STOP SERVO instruction or until a stop input is activated. The monitor (and servo\_1:stopped servo\_2:stopped) instruction causes the MultiPro's program to remain in this step until both motors complete their moves. You can re-profile either motor at any time to establish a new velocity.

If you want to start two axes simultaneously, you can program the TURN SERVO instructions using the ON START parameter. The motion of each motor will begin once the start input located on each axis is triggered. This will start all motion within one millisecond. Refer to *Dedicated Inputs* for information on using the ON START parameter and the start dedicated input.

```
[1] TWO_AXIS_VELOCITY_MOVE
    ;;; This program will commence a velocity move on two motor
    ;;; axes based on the parameters in the profile instructions.

    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____

    profile servo_1 maxspeed=50000 accel=100000
    profile servo_2 maxspeed=25000 accel=50000
    turn servo_1 cw
    turn servo_2 cw
    monitor (and servo_1:stopped servo_2:stopped)
    goto next
```

## Example 7 — Ratio of Two Servo Axes Follower to Leader

```
[1] RATIO_TWO_AXES
    ;;; This example sets up a ratio between two servo motors.
    ;;; Servo axis one (servo_1) is the follower and servo axis
    ;;; two (servo_2) is the leader.
    ;;;
    ;;; This step sets up the initial parameters for the servo
    ;;; motion and programs a ratio for servo_1. The ratio is 2
    ;;; to 1. From this point on, servo_1 follows all leader
    ;;; positions and velocity activities using the ratio.
    ;;;
    ;;; When programming a follower to leader ratio on the servo
    ;;; board, the first servo axis must be the follower axis and
    ;;; the second axis must be the leader axis.
    ;;;

<NO CHANGE IN DIGITAL OUTPUTS>

profile servo_1 servo at position maxspeed=reg_501
  accel=reg_502 P=reg_503 I=reg_504 D=reg_505
profile servo_2 servo at position maxspeedmaxspeed=reg_501
  accel=reg_502 P=reg_503 I=reg_504 D=reg_505
store 10000 to reg_16006
store 5000 to reg_16005
goto next

[2] LEADER_MOVE_FORWARD
    ;;; This step moves servo_2 (the leader) clockwise. The fol-
    ;;; lower (servo_1) follows axis two at the ratio.

<NO CHANGE IN DIGITAL OUTPUTS>

turn servo_2 cw reg_506 steps
monitor servo_2:stopped goto next

[3] TIME_DELAY

<NO CHANGE IN DIGITAL OUTPUTS>

delay 500 ms goto next

[4] MASTER_MOVE_REVERSE

<NO CHANGE IN DIGITAL OUTPUTS>

turn servo_2 ccw reg_506 steps
monitor servo_2:stopped goto RATIO_TWO_AXES
```

## Example 8 — Absolute Move of Two Servo Motors

This example shows two servo motors. The motor connected to the first axis moves 100,000 steps from its home position and the motor connected to the second axis moves 50,000 steps from its home position. The instruction

```
monitor (and servo_1:stopped servo_2:stopped)
```

causes the controller's program to remain in this step until both motors complete their moves.

You can re-profile either motor at any time to establish a new velocity.

If you want to start two or more axes simultaneously, you can program the `TURN SERVO` instructions with the `ON START` parameter. Both motors begin moving once the start input located on each axis is triggered. This action starts all motions within one millisecond. Refer to *Dedicated Inputs* for more information on using the `ON START` parameter and the Start dedicated input.

```
[1] TWO_AXIS_ABSOLUTE_MOVE
    ;; This program will commence an absolute move on two motor
    ;; axes based on the parameters in the profile instructions.
    ;;
    ;; Maxspeed units are in steps per second.
    ;; Acceleration units are in steps per second
    ;; per second.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=50000 accel=100000
    profile servo_2 maxspeed=25000 accel=50000
    turn servo_1 to 100000
    turn servo_2 to 50000
    monitor (and servo_1:stopped servo_2:stopped) goto next
```

## Example 9 — Staggering the Motion of Two Servo Motors

This example shows two servo motors. The motor connected to the first axis moves 100,000 steps from its home position and the motor connected to the second axis moves 25,000 steps from its home position.

The first motor is put in motion. When the first motor's position reaches half the travel distance, the program moves on to the next step and starts the motion of the second motor. The instruction

```
monitor (and servo_1:stopped servo_2:stopped)
```

causes the controller's program to remain in the second step until both motors complete their moves.

Any ratio between multiple axes may be achieved by applying the following formula:

- $Velocity = Acceleration * Time$

In this example, we are running a 2:1 ratio between the two axes. These motions will ramp up and down simultaneously.

```
[1] TWO_AXIS_STAGGERED_MOVE
    ;; This program will commence an absolute move on two motor
    ;; axes based on the parameters in the profile instructions.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile servo_1 maxspeed=50000 accel=100000
    profile servo_2 maxspeed=25000 accel=50000
    turn servo_1 to 100000
    if servo_1:position >= 50000 goto next

[2] TRIGGER_SECOND_AXIS
    ;; Turn the second axis and wait for both axes to be
    ;; complete before moving on the next part of the program.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    turn servo_2 to 25000
    monitor (and servo_1:stopped servo_2:stopped) goto next
```

# Special Registers

---

Group Access Registers .....	63
Individual Access Registers .....	64

This page is intentionally left blank.

## Special Purpose Registers for Servos

The special registers are the same as the ones in the 2600/2700 series controllers and are set up for 2 axes of motion. For the MultiPro, use the first two registers. For example, register 14001 contains the current position of servo axis 1 and register 14002 contains the current position of servo axis 2.



### Note

R/W indicates that the MultiPro can read and write to the register.

## Group Access Registers

Group access special purpose registers display the same parameters for both axes.

### Leader On-Start Feature

Registers 13801 – 13802	R/W	Leader on-start enable 0 = disabled 1 = enabled
Registers 13901 – 13902	R/W	Leader position setpoint for triggering armed axis

### Axis Status and FeedForward Parameters

Registers 14001 – 14002	R/W	Actual position
Registers 14101 – 14102	Read-only	Position error
Registers 14201 – 14202	Read-only	Theoretical velocity
Registers 14301 – 14302	Read-only	Status
		0 = Uninitialized
		1 = Stopped
		2 = Waiting
		3 = Accelerating
		4 = At speed
		5 = Deceleration speed
		6 = Decelerating to stop
		7 = Commence soft stop
		8 = Commence registration move
		9 = Searching for home
		10 = Following (ratioid from leader)
		12 = Command accepted
		128-255 = Errors
Registers 14401 – 14402	Read-only	Cumulative (integrated) position error
Registers 14501 – 14502	R/W	Velocity feedforward constant. Normal values range from 0 – 65535.
Registers 14601 – 14602	R/W	Deceleration rate. Normal values range from 1 – 130,000,000 pulses/s <sup>2</sup> .

Registers 14701 – 14702	Read-only	Monitors dedicated (auxiliary) inputs using a bit map. The default is normally open inputs.  Bit 0 = Not used Bit 1 = Home Bit 2 = Start Bit 3 = Kill command Bit 4 = Reverse limit Bit 5 = Forward limit Bit 6 = Index Bit 7 = Not used
Registers 14801 – 14802	R/W	Acceleration feedforward constant. Normal values range from 0 – 65535.

### Individual Access Registers

Individual access special purpose registers display all the parameters for a single axis.

#### Axis Status and FeedForward Parameters for Axis #1

Register 15000	R/W	Actual position
Register 15001	Read-only	Position error
Register 15002	Read-only	Theoretical velocity
Register 15003	Read-only	Status  0 = Uninitialized 1 = Stopped 2 = Waiting 3 = Accelerating 4 = At speed 5 = Deceleration speed 6 = Decelerating to stop 7 = Commence soft stop 8 = Commence registration move 9 = Searching for home 10 = Following (ratio from leader) 12 = Command accepted 128-255 = Errors
Register 15004	Read-only	Cumulative (integrated) position error
Register 15005	R/W	Velocity feedforward constant. Normal values range from 0 – 65535.
Register 15006	R/W	Deceleration rate. Normal values range from 1 – 130,000,000 pulses/s <sup>2</sup> .

Register 15007	Read-only	Monitors dedicated (auxiliary) inputs using a bit map. The default is normally open inputs.  Bit 0 = Not used Bit 1 = Home Bit 2 = Start Bit 3 = Kill command Bit 4 = Reverse limit Bit 5 = Forward limit Bit 6 = Index Bit 7 = Not used
Register 15008	R/W	Acceleration feedforward constant. Normal values range from 0 – 65535.



**Note**

Axis #2 functions the same as Axis #1 for the axis status and feedforward parameters. Registers 15010 – 15018 are used for Axis #2.

**Registration Feature for Axis #1**

Register 16000	R/W	Specifies the position where the registration window begins. Absolute position is the number of steps from the servo's home position. Normal values range from -2,147,483,648 to 2,147,483,647.
Register 16001	R/W	Specifies the position where the registration window ends. Relative position is the number of steps from the beginning of the registration window. Normal values range from -2,147,483,648 to 2,147,483,647.
Register 16002	Read-only	Indicates the position where registration occurred. Absolute position is the number of steps from the servo's home position. Normal values range from -2,147,483,648 to 2,147,483,647.
Register 16003	R/W	Specifies an offset to be added to the location where registration occurred. Relative position is the number of steps from the registration position. Normal values range from -2,147,483,648 to 2,147,483,647.
Register 16004	R/W	Indicates whether registration occurred. 1 indicates that registration has occurred. 0 indicates that the servo is ready for a registration move.



**Note**

Axis #2 functions the same as Axis #1 for the registration feature. Registers 16010 – 16014 are used for Axis #2.

---

**Axis Following Feature for Axis #1**

Register 16005	R/W	Ratio numerator. Normal values range from +32,767 to -32,767.
Register 16006	R/W	Ratio denominator. Normal values range from 1 to 32,767.
Register 16007	Read-only	Leader position. Normal values range from -2,147,483,648 to 2,147,483,647.
Register 16008	Read-only	Leader velocity. Normal values range from +4,000,000 to -4,000,000 steps/s.

---



**Note**

Axis #2 functions the same as Axis #1 for the axis following feature. Registers 16015 – 16018 are used for Axis #2.

---

**Special Registers – Additional Features for Axis #1**

Register 17000	Read-only	Firmware version number.
Register 17001	R/W	Servo filter selection using a bit map.  0 = Default (PID) 1 = Direct CCW 2 = Direct CW 3 = PID 5 = PAV 6 = Deceleration speed 7 = Virtual master 128 = Encoder following mode
Register 17002	R/W	Reverses input polarity so that the default is normally closed; uses a bit map.  Bit 0 = Not used Bit 1 = Home Bit 2 = Start Bit 3 = Kill command; cannot be changed; remains open. Bit 4 = Reverse limit Bit 5 = Forward limit Bit 6 = Index; cannot be changed; remains open. Bit 7 = Not used
Register 17003	R/W	Direction of home 0 = CCW (Default) +1 = CW -1 = CCW



**Notes**

1. Axis #2 functions the same as Axis #1 for the additional features. Registers 17010 – 17013 are used for Axis #2.
  2. You must set register 17001 before the initial profile instruction.
-

This page is intentionally left blank.