



# Dual Stepper Motion Module Applications Guide

<b>Stepper</b>	T	<b>DI 24 VDC</b>	T
<b>Stepper Axis</b>		<b>8 Digital Input +24 VDC Sourcing</b>	

## Contents

Programming a Stepper.....	5
Setting Up Stepper Motor Operating Parameters .....	5
Setting Acceleration and Deceleration Values .....	6
Searching for Home.....	7
Specifying the Homing Direction .....	7
Turning a Stepper .....	7
Stopping the Stepper.....	8
Stepper Hardware Considerations.....	9
Virtual Inputs.....	9

Doc. No. 5150AG  
Revision A  
September 2002

Copyright © 2002 Control Technology Corporation  
All Rights Reserved.  
Printed in USA

---

Sample Quickstep Programs .....	11
Example 1 — Absolute Move of One Stepper Motor .....	11
Example 2 — Relative Move of One Stepper Motor .....	11
Example 3 — Velocity Move of One Stepper Motor.....	12
Example 4 — Changing the Velocity of a Stepper Motor During Motion .....	12
Example 5 — Velocity Move of Two Stepper Motors .....	13
Example 6 — Absolute Move of Two Stepper Motors .....	14
Example 7 — Staggering the Motion of Two Stepper Motors .....	15

---

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.



## Programming a Stepper

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

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

---

### Note

The TT module is programmed as if it were a servo module. This enables the TT module to take advantage of Quickstep's expanded servo instruction set. TT motor axes are defined as servos in the Quickstep Symbol Browser.

The registers mentioned in this section only apply to Axis 1. Refer to *Special Registers* in the *Dual Axis Stepper Motion Module Installation Guide* for more information on other special registers.

---

## Setting Up Stepper Motor Operating Parameters

The 5100 must have a set of operating parameters before it can turn a stepper motor. You must specify these parameters with the `PROFILE SERVO` instruction. The stepper 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.
- **Holding Mode** — Must be specified but has no effect on stepper operation.

---

## 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 motor to take on new settings during a motor motion, is possible. To re-profile the motor, 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 motor 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 motor is stopped or is turning at maximum speed. You can view the status of the motor by checking the appropriate special registers. For example, check register number 14301 for the current status of the first motor.
- 

## 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 axis_1 servo at position maxspeed=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>.

---

### 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 motor 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 motor stops and the position is set to zero.

The 5100 supports a highly accurate method of finding the home position. In addition to providing direct support for a two-stage homing routine, the 5100 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 5100 to accept the index signal.

The motor travels in a counterclockwise (default) direction unless otherwise specified. You can change this default setting with a special purpose register. The homing sequence is as follows:

1. When the 5100 executes a `SEARCH AND ZERO SERVO` instruction, the motor 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 motor stops at the profiled deceleration rate.
3. The motor 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 motor 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 Stepper

There are three modes of turning the stepper:

1. **Absolute Positioning** — In this mode, the 5100's stepper 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 axis_1 to 50000
```

causes the motor to position itself 50,000 steps from home. The motor 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 axis_1 cw 12340 steps
```

turns the motor 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 axis_1 cw
```

starts the motor turning clockwise at its current maximum speed and acceleration. The motor continues to turn until the 5100 issues a `STOP SERVO` instruction or until a Limit or Stop input is activated.

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

The 5100's stepper board tracks the position of the motor 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 motor is wired according to the manufacturer's recommendations and that the logical sense of the direction output of the 5100's stepper board agrees with the logical sense expected by the motor's drive.

---

## Stopping the Stepper

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

- `STOP (SOFT) SERVO` causes the motor to stop at the deceleration rate specified in the last profile instruction.
- `STOP (HARD) SERVO` causes the 5100's stepper board to try to stop the motor instantly. However, because of the momentum, the motor may not stop instantly.

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



# Stepper Hardware Considerations

## Virtual Inputs

---

**Note**

You cannot use the digital inputs on the stepper module as virtual inputs unless they are among the first 16 digital inputs in the 5100 controller.

---

You can assign the following functions to any of the 5100 controller's first 16 general purpose inputs:

- **Start Input:** Begins motion from a turn "on start" Quickstep instruction.
- **Reset Input:** Uninitializes the motor axis and sets the command output to 0V.
- **Home Input:** Used during a search and zero instruction.
- **Forward Limit:** Prevents motion in the forward (+) direction; hard stops motion if activated during a forward move.
- **Reverse Limit:** Prevents motion in the reverse (-) direction; hard stops motion if activated during a reverse move.

These inputs are assigned using the special purpose registers in Table 1.

**Table 1. Motor Functions and General Purpose Inputs**

	<b>Axis 1</b>	<b>Axis 2</b>	<b>Axis 3</b>	<b>Axis 4</b>
<b>Start</b>	15160	15170	15180	15190
<b>Reset</b>	15161	15171	15181	15191
<b>Home</b>	15162	15172	15182	15192
<b>Fwd Lim</b>	15163	15173	15183	15193
<b>Rev Lim</b>	15164	15174	15184	15194

To reverse the polarity of any inputs, use register 17002 for axis 1, 17012 for axis 2, 17022 for axis 3, and 17032 for axis 4 as shown in Table 2.

---

**Note**

These registers use a bit mask.

---

**Table 2. Reversing the Polarity of Inputs**

<b>Bit</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>Input</b>	None	Home	Start	Reset	Rev Lim	Fwd Lim	Index	None
<b>Value</b>	1	2	4	8	16	32	64	128

To reverse the polarity on selected inputs, add the values for the appropriate inputs together and store to that register. For example to change the polarity of the Start and Reverse Limit Inputs only, add the appropriate values ( $4 + 16 = 20$ ) together and store the result to the register for the desired axis. So the Quickstep instruction

Store 20 to Register\_17002

changes the polarity of the Start and Reverse Limit inputs for axis 1.

## Sample Quickstep Programs

This section contains sample Quickstep programs for different stepper applications.

---

### Note

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

---

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

This example shows a stepper motor moving 100,000 steps from its home position. The monitor `axis_1:stopped` instruction causes the 5100'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 axis_1 servo at position maxspeed=50000
      accel=100000
      turn axis_1 to 100000
      monitor axis_1:stopped goto next
```

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

This example shows a motor moving clockwise 100,000 steps from its current position. The monitor `axis_1:stopped` instruction causes the 5100's program to remain in this step until the motor completes the move.

```
[1]  ONE_AXIS-RELATIVE_MOVE
      ;;; This program will commence a relative move on axis
      ;;; one based on the parameters in the profile
      ;;; instruction.
      _____
      <NO CHANGE IN DIGITAL OUTPUTS>
      _____
      profile axis_1 servo at position maxspeed=50000
      accel=100000
      turn axis_1 cw 100000 steps
      monitor axis_1:stopped goto next
```

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

This example shows a 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 axis_1:stopped` instruction causes the 5100'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 axis_1 servo at position maxspeed=50000 accel=100000
    turn axis_1 cw
    monitor axis_1:stopped goto next
```

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

This sample program positions a 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 stepper position where the program moves to the next step.

```
[1] COMPLEX_PROFILE
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile axis_1 servo at position maxspeed=10000 accel=200000
    turn axis_1 to 500000
    if axis_1:position >= 50000 goto next

[2] SECOND_PROFILE
    ;;; Re-profile the motor for a new velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile axis_1 servo at position maxspeed=20000
    if axis_1:position >= 70000 goto next

[3] THIRD_PROFILE
    ;;; Re-profile the motor for a new velocity.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    profile axis_1 servo at position maxspeed=50000
    if axis_1:position >= 110000 goto next
```

```
[4]  FOURTH_PROFILE
      ;;; Re-profile the motor for a new velocity.
      _____
      <NO CHANGE IN DIGITAL OUTPUTS>
      _____
      profile axis_1 servo at position maxspeed=100000
      if axis_1:position >= 300000 goto next

[5]  FIFTH_PROFILE
      ;;; Re-profile the motor for a new velocity.
      _____
      <NO CHANGE IN DIGITAL OUTPUTS>
      _____
      profile axis_1 servo at position maxspeed=80000
      if axis_1:position >= 420000 goto next

[6]  SIXTH_PROFILE
      ;;; Re-profile the motor for the final velocity and wait
      ;;; for the move to complete.
      _____
      <NO CHANGE IN DIGITAL OUTPUTS>
      _____
      profile axis_1 servo at position maxspeed=30000
      monitor axis_1:stopped goto PROFILE_COMPLETE
```

### Example 5 — Velocity Move of Two Stepper Motors

This example shows two stepper 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 `axis_1:stopped axis_2:stopped`) instruction causes the 5100'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 *Virtual Inputs* for more 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 axis_1 servo at position maxspeed=50000
    accel=100000
    profile axis_2 servo at position maxspeed=25000
    accel=50000
    turn axis_1 cw
    turn axis_2 cw
    monitor (and axis_1:stopped axis_2:stopped)
    goto next
```

### Example 6 — Absolute Move of Two Stepper Motors

This example shows two stepper 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 axis_1:stopped axis_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 *Virtual Inputs* for more information on using the `ON START` parameter and the Start dedicated input.

```
[1] TWO_AXIS_ABSOLUTE_MODE
    ;;; 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 axis_1 servo at position maxspeed=50000
    accel=100000
    profile axis_2 servo at position maxspeed=25000
    accel=50000
    turn axis_1 to 100000
    turn axis_2 to 50000
    monitor (and axis_1:stopped axis_2:stopped) goto next
```

### Example 7 — Staggering the Motion of Two Stepper Motors

This example shows two stepper 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 axis_1:stoppped axis_2:stopped)
```

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

```
[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 axis_1 servo at position maxspeed=50000
    accel=100000
    profile axis_2 servo at position maxspeed=25000
    accel=50000
    turn axis_1 to 100000
    if axis_1:position >= 50000 goto next
[2] TRIGGER_SECOND_AXIS
    ;;; Turn the second axis and wait for both axes to be
    ;;; complete before moving on to the next part of the
    ;;; program.
    _____
    <NO CHANGE IN DIGITAL OUTPUTS>
    _____
    turn axis_2 to 25000
    monitor (and axis_1:stopped axis_2:stopped) goto next
```