

Sample Programs for R256 and R356 controllers

CONTENTS

<i>Sample Program 1 (Simple Motor Movement. Set Run Current, Hold Current, Step Size, Velocity and Acceleration)</i>	3
<i>Sample Program 2 (Using Loops)</i>	3
<i>Sample Program 3 (Store Program to Run on Power Up. Use Halt Command for Push Button Input)</i>	3
<i>Sample Program 4 (Set Current, Wait for Switch 2 to Close, Home to Opto)</i>	4
<i>Sample Program 5 (Using Multiple Program Locations)</i>	5
<i>Sample Program 6 (A Move to Absolute Position)</i>	5
<i>Sample Program 7 (Storing a String in Location 2 for Later Execution)</i>	5
<i>Sample Program 8 (Nested Loops)</i>	5
<i>Sample Program 9 (Skip and Branch Instructions)</i>	6
<i>Sample Program 10 (Monitor 4 Switches and Execute 4 Different Programs Depending on Which Input is Pushed)</i>	6
<i>Sample Program 11 (Move 100 Steps Forward on Each Rising Edge of Switch 2)</i>	7
<i>Sample Program 12 (Coordinated Motion with Axes Doing Same Motion)</i>	7
<i>Sample Program 13 (Coordinated Motion with Axes Doing Different Motion)</i>	7
<i>Miscellaneous Advanced Commands</i>	8
<i>Pulse Jog and Opto Limit Modes</i>	8

PLEASE READ BEFORE USING

DISCLAIMER

The information provided in this document is believed to be reliable. However, no responsibility is assumed for any possible inaccuracies or omissions. Specifications are subject to change without notice.

Lin Engineering reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. Lin Engineering does not assume any liability arising out of the application or use of any product, circuit, or instructions described herein; neither does it convey any license under its patent rights, nor the rights of others.

There are known issues with R256/R356 controllers involving the Halt command (i.e., H01) when stored in memory location zero. Upon power up, the remaining command string after the Halt command might be executed if the user types in a new command. If memory location zero is not being used, the user is advised to always clear everything in memory by typing `/1?9`. Otherwise, the user may terminate the remaining command string in the buffer by issuing a `/1T`.

In this document,

References to R256 can also apply to SilverPak 17C, 17CE

References to R356 can also apply to SilverPak 23C, 23CE

Sample program 1, in addition to simple motor movement, demonstrates setting the running current, holding current, step size, velocity, and acceleration. You may refer to sample program 1, for these settings as they are not listed in many of the other programs samples.

Please verify that the running current is set at, or below the motors rated current before sending command strings to the motor or controller. Failing to do so can damage the motor and the controller.

Remember, to terminate a running command at any time, type, **"/1T"** and send it to the controller.

SAMPLE PROGRAM 1 (SIMPLE MOTOR MOVEMENT. SET RUN CURRENT, HOLD CURRENT, STEP SIZE, VELOCITY AND ACCELERATION)

/1m30h10j4V1000L300P1600R

/ Start character for a command string
1 Controller address **1**
m30 Running current **30%** of maximum current (R256 = 0.6 Amps, R356 = 0.9 Amps) **Verify current does not exceed motor rating**
h10 Holding current **10%** of maximum current
j4 ¼ stepping micro step size (1.8° motor = 800 steps/Rev)
V1000 Velocity set to **1000** microsteps per second (1.25 RPS for 1.8° motor 1/4 step)
L300 Acceleration set to **300** microsteps per second²
P1600 Move motor 1600 steps in **Positive** direction (2 revolutions for 1.8° motor)
R Run command string

SAMPLE PROGRAM 2 (USING LOOPS)

/1gP1000M500D1000M500G5R

/ Start character for a command string
1 Controller address **1**
g Begin a repeat loop (Anything between 'g' and 'G' will repeat)
P1000 Move motor **1000** steps in **Positive** direction
M500 Wait for **500** Milliseconds
D1000 Move motor **1000** steps in **negative** direction
M500 Wait for **500** Milliseconds
G5 Exit loop after **5** iterations
R Run command string

SAMPLE PROGRAM 3 (STORE PROGRAM TO RUN ON POWER UP. USE HALT COMMAND FOR PUSH BUTTON INPUT)

/1s0gH01A100H01A0G0R

/ Start character for a command string
1 Controller address **1**
s0 store to program **0** (Program 0 will run at power up)
g Begin a repeat loop
H01 Halt commands until a low '0' is seen on input 1 (Push button is pressed)
A100 Move motor to **Absolute** position **100**
H01 Halt commands until a low '0' is seen on input 1 (Push button is pressed)
A0 Move motor to **Absolute** position **0**
G0 Repeat infinitely
R Run command string

To execute program, type **/1e0R**. Or, powering down and powering back up will execute this program string. Only program location 0 will start upon power up.

To **Terminate** this infinite loop, type **/1T**

SAMPLE PROGRAM 4 (SET CURRENT, WAIT FOR SWITCH 2 TO CLOSE, HOME TO OPTO)

/1s0m75h10gJ3M500J0M500G10HZ10000A1000A0R

/ Start character for a command string
1 Controller address **1**
s0 store to program **0** (Program 0 will run at power up)
m75 Running current 75% of maximum current
h10 Holding current 10% of maximum current
g Begin a repeat loop
J3 Turn on both drivers.
M500 Wait for **500** Milliseconds
J0 Turn off both drivers.
M500 Wait for **500** Milliseconds
G10 Exit loop after **10** iterations
H Halt commands until switch2 closure (Switch 2 automatically selected if no number)
Z10000 Home the stepper to Opto #1 (Sets a maximum of 10000 steps to find Opto#1)
A1000 Move motor to Absolute position **1000**
A0 Move motor to Absolute position **0**
R Run command string

Note: This program string will abort after the Z command if it does not find a flag.

SAMPLE PROGRAM 5 (USING MULTIPLE PROGRAM LOCATIONS)

This sample uses 3 separate command strings.

The first two, will be stored into memory locations 1 and 2 on the controller.

The third string, sets run current, holding current, micro step size, velocity, acceleration and calls string 1 if input 2 goes low.

This will loop continuously, until a momentary switch on input 2 (Pin 8 R256, Pin 5 R356) is pressed and released.

/1s1M300gD25S02G0e2R

/1s2M300gP25S02G0e1R

/1m30h10j8V1000L100gP25S02G0e1R

/1s1M300gD25S02G0e2R

/ Start character for a command string
1 Controller address **1**
s1 store to program **1**
M300 Wait for **300** Milliseconds (in this sample M300 allows switch bounce to settle)
g Begin a repeat loop
D25 Move motor **25** steps in negative direction
S02 Skip next instruction if input **2** (Pin 8 R256, Pin 5 R356) is low (in this sample switch is normally low)
G0 Repeat infinitely
e2 execute program stored in memory location **2**
R Run command string

/1s2M300gP25S02G0e1R

/ Start character for a command string
1 Controller address **1**
s2 store to program **2**
M300 Wait for **300** Milliseconds (in this sample M300 allows switch bounce to settle)
g Begin a repeat loop
P25 Move motor **25** steps in positive direction
S02 Skip next instruction if input **2** (Pin 8 R256, Pin 5 R356) is low (in this sample switch is normally low)
G0 Repeat infinitely
e1 execute program stored in memory location **1**
R Run command string

To Terminate this set of infinite loops, type **/1T**

SAMPLE PROGRAM 6 (A MOVE TO ABSOLUTE POSITION)

/1A12345R

/ Start character for a command string
1 Controller address 1
A12345 Move motor to Absolute position 12345
R Run command string

SAMPLE PROGRAM 7 (STORING A STRING IN LOCATION 2 FOR LATER EXECUTION)

/1s2gA10000M500A0M500G10R

/1e2R (Will execute the previously stored program #2)

/1s2gA10000M500A0M500G10R

/ Start character for a command string
1 Controller address 1
s2 store to program 2
g Begin a repeat loop
A10000 Move motor to Absolute position 10000
M500 Wait for 500 Milliseconds.
A0 Move motor to Absolute position 0
M500 Wait for 500 Milliseconds
G10 Exit loop after 10 iterations
R Run command string

/1e2R

/ Start character for a command string
1 Controller address 1
e2 execute program stored in memory location 2
R Run command string

SAMPLE PROGRAM 8 (NESTED LOOPS)

/1gA10A1000gA10A100G10G100R

/ Start character for a command string
1 Controller address 1
g Begin a repeat loop
A10 Move motor to Absolute position 10
A1000 Move motor to Absolute position 1000
g Begin inner repeat loop
A10 Move motor to Absolute position 10
A1000 Move motor to Absolute position 100
G10 Exit inner loop after 10 iterations (End of inner Loop)
G100 Exit outer loop after 100 iterations (End of outer loop)
R Run command string

To Terminate the above loop type **/1T**

SAMPLE PROGRAM 9 (SKIP AND BRANCH INSTRUCTIONS)

Two “Program strings” are stored, one in string **0** and one in string **1**. The code will switch from one program to the other depending on the state of input 3. In the example given the code will cycle the motor between position A0 and A100 if input 3 is High and between A0 and A10 if input 3 is Low.

Store string to location 0:

```
/1s0gA0A100S13e1G0R  
/  
1 Controller address 1  
s0 store to program 0 (Program 0 will run at power up)  
g Begin a repeat loop  
A0 Move motor to Absolute position 0  
A100 Move motor to Absolute position 100  
S13 Skip next instruction if 1 (hi) on input 3  
e1 execute program stored in memory location 1  
G0 Repeat infinitely  
R Run command string
```

Store string to location 1:

```
/1s1gA0A10S03e0G0R  
/  
1 Controller address 1  
s0 store to program 0 (Program 0 will run at power up)  
g Begin a repeat loop  
A0 Move motor to Absolute position 0  
A10 Move motor to Absolute position 100  
S03 Skip next instruction if 0 (low) on input 3  
e0 execute program stored in memory location 0  
G0 Repeat infinitely  
R Run command string
```

SAMPLE PROGRAM 10 (MONITOR 4 SWITCHES AND EXECUTE 4 DIFFERENT PROGRAMS DEPENDING ON WHICH INPUT IS PUSHED)

```
/1s0gS11e1S12e2S13e3S14e4G0R  
/1s1A100e0R  
/1s2A200e0R  
/1s3A300e0R  
/1s4A400e0R
```

Five program strings are stored. String 0 automatically executes on power up and loops around sampling the switches one by one, and skips the subsequent instruction if it is not pressed.

When Switch1 is pressed the stored string 1 is executed, which moves the stepper to position 100. Execution will then return to string 0, due to the e0 command at the end of each stored string. If the switch is still pressed it will jump back to string 1 again but, since it is already at that position there will be no visible motion.

To terminate the above loop type **/1T**

Note: The “e” command is a more of a “GOTO” than a “GOSUB” because execution does not automatically return to the original location.

SAMPLE PROGRAM 11 (MOVE 100 STEPS FORWARD ON EACH RISING EDGE OF SWITCH 2)

/1gH02H12P100G0R

This endless loop first waits for a 0 level on switch1 then waits for a "1" level on Input2. Then a relative move of 100 Steps is issued, and the program returns to the beginning to look for another rising edge.

To Terminate the above endless loop type **/1T**

SAMPLE PROGRAM 12 (COORDINATED MOTION WITH AXES DOING SAME MOTION)

/_A10000R

/_ (Slash then Underscore) Talk to ALL Motors (up to 15 motors).

A1000 Move motor to Absolute position **1000**

R Run command string (All motors will go to Absolute position 1000)

SAMPLE PROGRAM 13 (COORDINATED MOTION WITH AXES DOING DIFFERENT MOTION)

/1A10000

/2A200

/AR

/1A10000 Set up motor 1 command buffer to go to Absolute position **10000**.

/2A200 Set up motor 2 command buffer to go to Absolute position **200**.

/AR Execute current commands in Bank Address "A" which is motors 1 and 2.

(The "A" here is an Address of motors 1&2 because it comes after the slash and should not be confused with the "A" that means absolute position.) Both moves will start at the same time, and complete at a time determined by the Velocity set for each axis.

Skip command: will skip the command following it if the input is high or low.

Useful for executing different programs based on a high or low signal on an input. Used with looping in this sample.

/1gS02A10000A0G20R

Halt current command string and wait until condition specified.

This example will wait for switch two (2) to close (0) and then proceed to execute "P10000". Used with looping in this sample.

/1gH02P10000G20R

PULSE JOG AND OPTO LIMIT MODES

Enable Pulse Jog Mode

Will now use inputs 1 & 2 to move 10,000 CW or CCW

/1B10000n1R

Enable Opto Limit Mode

This will rotate the motor in the positive direction infinitely until it hits a switch, then it will rotate in the negative position infinitely until it hits the other switch. This will repeat continuously. (Use inputs 3 & 4)

/1n2R

/1n2gP0D0G0R

Enable Pulse Jog Mode and Opto Limit Mode

/1n3R (Uses all 4 inputs to combine the two modes)

Enable Continuous Jog Mode

/1n4R (Now use inputs 1 & 2. Pull to ground for movements, go high to stop motion)

Enable Opto Limit Mode and Continuous Job Mode

/1n6R (Uses all 4 inputs to combine the two modes)

Contact Lin Engineering Tech Support for R256, or R356 programing assistance at techsupport@linengineering.com