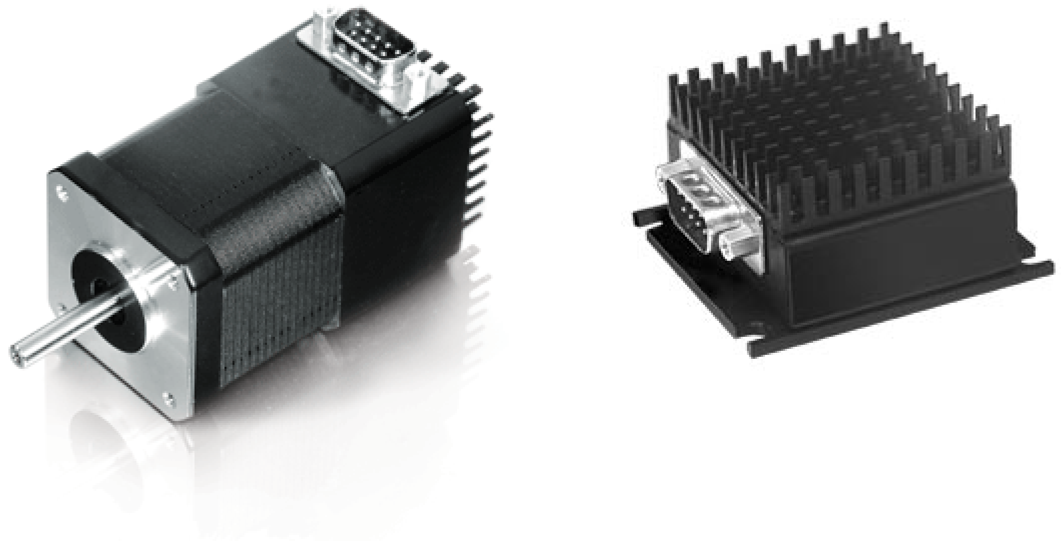
Commands List for

Silverpak 17C Integrated motor + controls
R256 controller/driver



Version 1.14

Thank you for purchasing the R256 Controller with Microstepping Driver or the Silverpak 17C integrated motor and controller/driver. This product is warranted to be free of manufacturing defects for one year from the date of purchase.

Technical Support for Lin Engineering, a distributor for RMS Technologies
By Telephone: 408-919-0200
(Mon.-Fri., 8:00 a.m.-5:00 p.m.)
On the Web: www.linengineering.com
Our technical support group is glad to work with you in answering your questions. If you cannot find the solution to your particular application, or, if for any reason you need additional technical assistance, please call technical support at **408-919-0200**.

PLEASE READ BEFORE USING

Before you begin, ensure there is a suitable DC Power Supply. **Do not disconnect the DB-9 cable while power is still being applied to the controller.** This will damage the board. Under any circumstances, do not exceed +40 VDC.

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.

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

There are known issues 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*.

Special Symbols

⚠ **Indicates a WARNING and that this information could prevent injury, loss of property, or even death (in extreme cases).**

R256-Silverpak Commands User Manual

Product:	R256 and Silverpak 17C/CE
Version:	1.14
Date:	10/29/2009

Version History		
Version	**Date**	**Description of Changes**
1.00		New User Manual
1.01		Typographical errors
1.02		Added more explanation of commands
1.03		Added the error codes that the board outputs
1.04		Added "Understanding Response of the unit"
1.05		Typographical errors
1.06		Typographical errors
1.07	9/29/2006	Typographical errors
1.08	2/2/2007	Standardization of all user manuals
1.09	2/6/2007	Updated errors on L command, Homing Example
1.10	4/13/2007	Velocity range, explanation of commands more in detail
1.11	07/17/2007	Updated response codes
1.12	10/26/2007	Updated Z command max value
1.13	8/21/2009	Updated special mode, n, description and examples.
1.14	10/29/2009	Added full stepping

TABLE OF CONTENTS

DT Protocol syntax:

The DT Protocol allows the unit to be commanded over a simple serial port.

Start Character	Address	Commands	Run	End of a string
/	1-9*	Command strings	R	<CR>

*To Access Drivers 10 – 16 use the following:

Driver #	**Command**	
A	:	(colon)
B	;	(semi colon)
C	<	(less than)
D	=	(equals)
E	>	(greater than)
F	?	(question mark)
0	@	(at sign)

Running two or more motors together:

Motors 1 and 2:	"A"	
Motors 3 and 4:	"C"	
Motors 5 and 6:	"E"	
Motors 7 and 8:	"G"	
Motors 9 and 10:	"I"	
Motors 11 and 12:	"K"	
Motors 13 and 14:	"M"	
Motors 15 and 16:	"O"	
Motors 1, 2, 3 and 4:	"Q"	
Motors 5, 6, 7 and 8:	"U"	
Motors 9, 10, 11 and 12:	"Y"	
Motors 13, 14, 15 and 16:	"]"	(close bracket)
For all motors:	"_"	(underscore)

Example: /CA5000R will move motors 3 and 4 to Absolute Position 5000.
Note: When using the multi-addressing, no response will be returned

Default Values:

Function (command)	**Description**
Running Current (m)	30% of 2.0 Amps
Holding Current (h)	10% of the run current
Step Resolution (j)	256x
Top Velocity (V)	305175 pps (microsteps/sec)
Acceleration (L)	L = 1000; 6103500 µsteps/sec^2
Position	0
Microstep smoothness (o)	1500
Outputs (J)	Both are turned off, J0
Baud Rate	9600 bps

List of Commands

Command (Case Sensitive)	Operand	Example	Description
HOMING & POSITIONING			
Z	0-2^{31}	**/1Z10000R**	Initialize the Motor. Motor will turn towards 0 until the home opto sensor is interrupted. If already interrupted it will back out of the opto and come back in until re-interrupted. Current motor position is set to zero. Speed of homing is set by V. In the example, the motor will take 10000 steps to find the home sensor. If sensor is still not found after 10000 steps, it will stop motion.
z	0-2^{31}	**/1z65536R**	Sets current position without moving motor
A	0-2^{31}	**/1A10000R**	Move Motor to Absolute position. i.e. moves to the 10,000th step. Issuing A10000 again will NOT move the motor because it is already at that location.
f	0 or 1	**/1f1R**	Flag polarity. Sets polarity of direction of home sensor, default is 0.
P	0-2^{31}	**/1P10000R**	Move Motor relative number of steps in positive direction. A "P0" command rotates motor infinitely, which enters into Velocity Mode. Any other finite number will set the mode to be in Position Mode.
D	0-2^{31}	**/1D10000R**	Move Motor relative number of steps in negative direction (Note: Motor will not run in the negative direction if the position is at 0. You can use the 'z' command to set the 0 position to be further away in the negative direction. OR you can use the F command to reverse direction of rotation.) A "D0" command rotates motor infinitely, which enters into Velocity Mode. Any other finite number will set the mode to be in Position Mode.
B	0-2^{31}	**/1B1000R**	Sets the distance for pulse jog mode (see n command)
T		**/1TR**	Terminate current command
F	0, 1	**/1F1R**	Reverses the positive direction to be negative. The P and D command will reverse directions. Default is 0.
	2^{31}		2^{31} = 2,147,483,648
VELOCITY & ACCELERATION			
V	0-2^{31}	**/1V2000R**	In Position Mode, this sets the Top Speed of the Motor in **µsteps/sec.** During velocity mode, speed can be changed on the fly (during rotation). (Max speed is 2GHz)
L	0-65000	**/1L5000R**	This sets the Acceleration factor **µsteps/sec²** = (L Value) x (6103.5). **i.e. /1L1R takes 16.384 Seconds to get to a speed of V=100000 (µsteps/sec)** Default is L=1000, default speed V = 305175 µsteps/sec, so default acceleration = 6103500 µsteps/sec^2. It should take 0.05 seconds to get to top speed.

Command (Case Sensitive)	Operand	Example	Description
SETTING CURRENT			
m	0-100	**/1m50R**	Sets the running current on a scale of 0 to 100% of the max current, 2.0A. Default setting is m30.
h	0-50	**/1h20R**	Sets the Hold Current on a scale of 0 to 50% of the max current, 2.0A. Default setting is h10.
LOOPING & BRANCHING			
g		**/1gP10G5R**	Beginning of a repeat loop
G	0-30000	**/1gP10G0R**	End of a repeat loop. Loops can be nested up to 4 levels. A value of 0 causes the loop to be infinite.
M	0-30000	**/1M2000R**	Delay for "M" milliseconds
H	01 11 02 12 03 13 04 14	**/1gH02P10000G20R**	Halt current command string and wait until condition specified. Example will wait for switch two (2) to close (0) 01 Wait for low on input 1 (Pin 2) 11 Wait for high on input 1 (Pin 2) 02 Wait for low on input 2 (Pin 8) 12 Wait for high on input 2 (Pin 8) 03 Wait for low on input 3 (Pin 7) 13 Wait for high on input 3 (Pin 7) 04 Wait for low on input 4 (Pin 5) 14 Wait for high on input 4 (Pin 5) • Halted operation can also be resumed by typing /1R • Only input 4 can stop motion of a P0 or D0 command
S	01 11 02 12 03 13 04 14	**/1gS02A10000A0G20R**	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. 01 Skip next instruction if low on input 1 (Pin 2) 11 Skip next instruction if hi on input 1 (Pin 2) 02 Skip next instruction if low on input 2 (Pin 8) 12 Skip next instruction if hi on input 2 (Pin 8) 03 Skip next instruction if low on input 3 (Pin 7) 13 Skip next instruction if hi on input 3 (Pin 7) 04 Skip next instruction if low on input 4 (Pin 5) 14 Skip next instruction if hi on input 4 (Pin 5)

Command (Case Sensitive)	Operand	Example	Description
n	0-4095	**/1n2R**	**Sets Modes – Interpret as combination of Binary Bits** **Bit0: /1n1R Enable Pulse Jog Mode**. Jog distance is given by "B" command. Velocity is given by "V" command . The Switch Inputs 1 and 2 become the Jog Inputs. Press input 1 to rotate CW for a given distance, B. Press input 2 to rotate CCW for a given distance, B. **Bit1: /1n2R Enable Limit**. The opto input #3, pin 7, becomes one limit switch, normally closed. Motor will only rotate using /1D0R command and while switch is low. Can only be combined with other commands if you jump in and out of the modes, for example: /1n2D0n0P10000R. **Bit2: /1n4R Enable Continuous Jog Mode**. Continuous run of motor while switch is depressed. Velocity is given by the "V" command. Input 1 will rotate CCW, Input 2 will rotate CW, when either is low. When inputs are high, no motion will occur.
PROGRAM STORAGE & RECALL			
s	0-15	**/1s1A10000A0R**	Stores a program. Program 0 is executed on power up (Total of 14 commands max per string when storing, but if not storing you can send 256 characters max).
e	0-15	**/1e1R**	Executes the Stored Programs 0-15. You can execute one program within a program. i.e. /1s0V500j2e1R, and /1s1gP1000M500G5e2R and /1s2f1Z100000R: This example will power on, execute speed of 500, 2x microstep and execute program #1: move 1000 steps and pause 0.5 seconds (5 times), then execute program 2 which will home the motor in the opposite direction.
PROGRAM EXECUTION			
R		**/1R**	Run the command string that is currently in the execution buffer – Always end commands with 'R'.
X			Repeat the current command string

Command (Case Sensitive)	Operand	Example	Description
MICROSTEPPING			
j	1, 2, 4, 8, 16, 32, 64, 128, 256	**/1j256R**	Adjusts the resolution in micro-steps per step. Available resolutions: full step, 2x, 4x, 8x, 16x, 32x, 64x, 128x, 256x.
o (default 1500)	1400-1650		Allows user to correct any unevenness in microstep size. Adjusts audible noise and should be executed while motor is running. Should only be adjusted in small increments and should not exceed 1650 and should not be set below 1400.
ON/OFF DRIVERS (OUTPUTS)			
J	0-3		On/Off Driver. Turns on or off the two outputs. (I/O's are bidirectional) It's a two bit Binary value: 3=11=Both Drivers On, 2=10=Driver2 on, Driver1 off, etc. (Drivers output 3VDC max)
QUERY COMMANDS			
The following commands are queries and cannot be cascaded in strings or stored. They can be executed while other commands are still running.			
?	0	**/1?0**	Returns the current motor position
?	1	**/1?1**	Returns the current Start Velocity
?	2	**/1?2**	Returns the current Slew Speed for Position mode
?	3	**/1?3**	Returns the current Stop Speed
?	4	**/1?4**	Returns the status of all four inputs, 0-15 representing a 4 bit binary pattern: Bit 0 = Input 1 (Pin 2) Bit 1 = Input 2 (Pin 8) Bit 2 = Input 3 (Pin7) Bit 3 = Input 4 (Pin 5)
?	5	**/1?5**	Returns the current Velocity mode speed
?	6	**/1?6**	Returns the current step size
?	7	**/1?7**	Returns the current 'o' value
?	9	**/1?9**	Erases all commands/program stored in EPROM except for the any settings such as current, microstepping, velocity, acceleration and any other settings
$		**/1$**	Recalls current command executed. To see what currently stored in a specific program, run the program and issue the /1$ example: /1e2R /1$
&		**/1&**	Returns the current Firmware revision and date
Q		**/1Q**	Query current status of the controller: 0 = No Error 1 = Initialization error 2 = Bad Command 3 = Operand out of range
T		**/1T**	Terminate current commands
BAUD CONTROL			
	b **9600** **19200** **38400**	**/1b19200 R**	**Adjustable baud rate** This command will usually be stored as program zero and execute on power up. Default baud rate is 9600.

Responses from Controller in HyperTerminal

The SilverpakC respond to commands by sending messages addressed to the Master Device (in most cases is your PC). It always assumes it has an address of zero (0). The master device should parse the communications on the bus continuously for responses starting with /0. (It is not recommended, for example, to look for the next character coming back after issuing a command because glitches on the bus when the bus reverses direction can sometimes be interpreted as characters). After the /0 the next is the "status character" which is a collection of 8 bits.

These bits are:

Bit 7	Reserved
Bit 6	Always set
Bit 5	Ready Bit – it is set when the unit is ready to accept a command
Bit 4	Reserved

Bit 3, 2, 1, 0 represent the error codes:

0	‘	No error
1	A	Initialization error
2	B	Bad command (illegal command was sent)
3	C	Bad operand (out of range operand value)
4		N/A
5	E	Communication error (internal communication error)
6		N/A
7	G	Not initialized (controller was not initialized before attempting a move)
8		N/A
9	I	Overload error (system could not keep up with commanded position)
10		N/A
11	K	Move not allowed
12		N/A
13		N/A
14		N/A
15	O	Command overflow (unit was already executing a command when another command was received)

Example of initialization error response:
The Upper nibble only takes on values of 4 or 6 in Hex. An initialization error has a response of "1" in the lower nibble. Therefore the response is 41 or 61 in Hex, which corresponds to the ASCII characters of upper case "A" and lower case "a", depending on if the devices I busy or not, respectively.

Example of invalid command response:
The Upper nibble only takes on values of 4 or 6 in Hex. An invalid command has a response of "2" in the lower nibble. Therefore the response is 42 or 62 in Hex, which corresponds to the ASCII characters of upper case "B" or lower case "b", depending on if the device is busy or not, respectively.

Example of Operand Out of Range response:
The Upper nibble only takes on values of 4 or 6 in Hex. An invalid command has a response of "3" in the lower nibble. Therefore the response is 43 or 63 in Hex, which corresponds to the ASCII characters of upper case "C" or lower case "c", depending on if the device is busy or not, respectively.

Example of Overload Error Response:
The Upper nibble only takes on values of 4 or 6 in Hex. An invalid command has a response of “7” in the lower nibble. Therefore the response is 47 or 67 in Hex, which corresponds to the ASCII characters of upper case “I” or lower case “i”, depending on if the device is busy or not, respectively.

Understanding the Response

Example Response to the command /1?4

FF	RS485 line turn around character. It’s transmitted at beginning of a message
2F	ASCII “/” Start character. The DT protocol uses the ‘/’ for a start character
30	ASCII “0”, this is the address of the recipient for the message.
60	This is the status character, here, 60 is “ ‘ “, no error
31	
31	These two bytes are the actual answer in ASCII. It will indicate the status of the 4 inputs in the form of 4 bits:

Bit 0 – switch 1
Bit 1 – switch 2
Bit 2 – opto 1
Bit 3 – opto 2

03	This is the ETX or end of text character. It is at the end of the answer string
0D	This is the carriage return character
0A	This is a line feed

A program that receives these responses must continuously parse for /0 and take the response from the bytes that follow /0. The first Character that comes back may be corrupted due to line turn around transients, and should not be used as a “timing mark”.

Example #1:

/1gP1000D1000G10R will move motor 1000 steps counterclockwise, then 1000 steps clockwise, in a loop for 10 times.

/	Always begin a program with the forward slash
1	Address of controller (Check with the Red Dial on the unit)
g	Beginning of loop (All commands within ‘g’ and ‘G’ will repeat)
P1000	Move counterclockwise 1000 steps
D1000	Move clockwise 1000 steps
G10	End loop, repeat 10 times (G0 will repeat infinitely, type /1T to terminate)
R	Run this command

Example #2:

/1s0gH01A100H01A0G0R will store a program to memory, and run upon power up. This program will move 100 steps (90° for a 1.8° step motor) when you press a push button. And it will return to it’s original position when pressing the button a second time. This will repeat infinitely.

/	Always begin programs with a forward slash
1	Address of Controller. Check on the Red Dial of the unit
s0	Store to program 0 – defined as running upon power up
g	Beginning of loop. Anything between ‘g’ and ‘G’ will repeat
H01	Halt commands until a low ‘0’ is seen on input 1. Push button is pressed.
A100	Then move 100 steps (absolute position)
H01	Halt again until a low ‘0’ is seen on input 1. Push button is pressed.
A0	Move back to Position 0.
G0	End loop. Repeat infinitely.
R	Run commands

To execute program, type **/1e0R**. Or, power down and power up. Only program 0 will start upon power up. To terminate out of this infinite loop, type **/1T**.

Example #3:

Enable Pulse Jog Mode
/1B10000V51200n1R
Set jog distance B, speed of V, then enable the mode, use inputs 1 & 2 to go CW and CCW by 10,000 steps.
Enable Opto Limit Mode
/g1n2D0n0M500P1000G0R
Enable the mode, rotate until you hit the switch (until switch goes high), then it will get out of the special modes (n0), wait 0.5 seconds, and move 1000 steps.
Enable Continuous Jog Mode
/1n4R (Now use inputs 1 & 2. Pull to ground for movements, go high to stop motion)

Homing Sensor

The “Z” command is used to initialize the motor to a generally known amount of steps (a maximum of 10000 steps + 400 default steps). When issued, i.e. /1Z5000R, the motor will turn towards zero at a maximum step of 5400 until the home opto sensor is interrupted. If issued a /1Z0R, motor will only move 400 steps to find opto sensor.

If the sensor is already interrupted, and /1Z5000R was issued, the motor will move in the opposite direction until the sensor is un-cut again. At this time, the motor moves towards home in the same way described above. When sensor is cut, motor stops motion and current position is reset to zero. Speed is set by upper case V, i.e. /1V4000Z5000R.

The Z command is used in conjunction with Pins 7 and 9. An appropriate optical sensor must be attached to Pins 7 and 9 in order for the homing command to work properly. The Z command allows the motor to rotate until Pin 7, Input 3, goes from low to high.