

# SILVERPAK 17C 64x

## Commands

Version 1.00

Commands to the R164 Controller can be issued from a HyperTerminal connection. The syntax is intuitive and easy to follow.

### 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 “C” (motors 3 and 4) to “A” (Absolute Position) 5000.

## LIST OF COMMANDS FOR THE 17C

Command (Case Sensitive)	Operand	Description
<b>HOMING &amp; POSITIONING</b>		
<b>Z</b>	0-max*	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 the 'v' command.
<b>z</b>	0-max*	Sets current position without moving motor
<b>A</b>	0-max*	Move Motor to Absolute position
<b>f</b>	0 or 1	Flag polarity. Sets polarity of home sensor, default is 0.
<b>P</b>	0-max*	Move Motor relative number of steps in positive direction. A value of zero will cause an endless forward move at speed 'V'. By doing so, it enters into Velocity Mode. Any other finite number will set the mode to be in Position Mode.
<b>D</b>	0-max*	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.) A value of zero will cause an endless backwards move at speed 'V'. This will enter Velocity Mode. Any other finite number will set the mode to be in Position Mode.
<b>T</b>	-	Terminate current command
<b>F</b>	0, 1	Reverses the positive direction to be negative. The 'P' and 'D' command will reverse directions. Default is 0.
<b>VELOCITY &amp; ACCELERATION</b>		
<b>v</b>	200-2500	In Position Mode, this sets the Start Speed of the Motor in half steps per second. This also sets the 'Z' command speed.
<b>V</b>	100-10000	In Position Mode, this sets the Top Speed of the Motor in half steps per second.
<b>V</b>	100-4000	In Velocity Mode, changes the Top Speed "on the fly".
<b>c</b>	300-900	In Position Mode, this sets the stop speed of the motor in half steps per second.
<b>L</b>	1-20	In Position Mode, this sets the Acceleration factor (acceleration = $L * 7500 \text{ steps/sec}^2$ )
<b>SETTING CURRENT</b>		
<b>m</b>	0-100	Sets "Fast Move" Current on a scale of 0 to 100% of the max current, 1.5A. Default setting is m30.
<b>l</b>	0-100	Sets "Slow Move" Current on a scale of 0 to 100% of the max current. Use this when $V < v$ . Default is l30.
<b>h</b>	0-50	Sets the Hold Current on a scale of 0 to 50% of the max current. Default setting is h10.

Command (Case Sensitive)	Operand	Description
<b>LOOPING &amp; BRANCHING</b>		
<b>g</b>	-	Beginning of a repeat loop
<b>G</b>	0-30000	End of a repeat loop. Loops can be nested up to 4 levels. A value of 0 causes the loop to be infinite.
<b>M</b>	0-30000	Delay for "M" milliseconds
<b>H</b>	Blank	Halt current command string and wait until condition specified. † (See note on page 4) Wait for switch 2 closure
	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
<b>S</b>	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)
<b>PROGRAM STORAGE &amp; RECALL</b>		
<b>s</b>	0-15	Stores a program. Program 0 is executed on power up (Total of 25 commands max per string)
<b>e</b>	0-15	Executes the Stored Programs 0-15
<b>PROGRAM EXECUTION</b>		
<b>R</b>	-	Run the command string that is currently in the execution buffer
<b>X</b>	-	Repeat the current command string
<b>MICROSTEPPING</b>		
<b>j</b>	2, 4, 8, 16, 32, 64	Adjusts the resolution in micro-steps per step.
<b>o</b>	0-250	Allows user to correct any unevenness in microstep size
<b>ON/OFF DRIVERS</b>		
<b>J</b>	0-3	On/Off Driver. It's a two bit Binary value: 3=11=Both Drivers On, 2=10=Driver2 on, Driver1 off, etc.

Command (Case Sensitive)	Operand	Description
<b>QUERY COMMANDS</b>		
The following commands are queries and cannot be cascaded in strings or stored. They can be executed while other commands are still running.		
?	0	Returns the current motor position
?	1	Returns the current Start Velocity
?	2	Returns the current Slew Speed for Position mode
?	3	Returns the current Stop Speed
?	4	Returns the status of all four inputs, 0-15 representing a 4 bit binary pattern: Bit 0 = Output 1 (Pin 2) Bit 1 = Output 2 (Pin 8) Bit 2 = Input 3 (Pin7) Bit 3 = Input 4 (Pin 5)
?	5	Returns the current Velocity mode speed
?	6	Returns the current step size
?	7	Returns the current 'o' value
&	-	Returns the current Firmware revision and date
Q	-	Query current status of the R164: 0 = No Error 1 = Initialization error 2 = Bad Command 3 = Operand out of range
T	-	Terminate Current Command
		*10 <sup>9</sup> (32 Bit)

### Responses from R164 in HyperTerminal

Syntax	Hex value	Description
/0'□	0x60	Command is terminated
/0@□	0x40	Good command, command received
/0C□	0x43	Command out of range
/0b□	0x62	Bad Command
/0O□	0x4F	Overflow

† 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 */I?9*. Otherwise, the user may terminate the remaining command string in the buffer by issuing a */IT*.

### Example:

**/1gP1000D1000G10R** will move the 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:

**/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 its original position when pressing the button a second time. This will repeat infinitely

/	Always begin a program with the forward slash
1	Address of controller (Check with the Red Dial on the unit)
s0	Store to program 0 – defined as running upon power up
g	Beginning of loop (All commands within 'g' and 'G' will repeat)
H01	Halt commands until a low '0' is seen on input 1. Push button is pressed.
A100	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, type /1T to terminate)
R	Run this command

To execute the 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**

### 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 the 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 lower case ‘v’, i.e. **/1v4000Z5000R**.

## Understanding the Response

The R164 controller board sends commands to a Master Device, i.e. a PC, and is dedicated as Address 0. The Master Device looks for the response from the controller in a certain syntax that is partially transparent in HyperTerminal. /0 is the beginning of the syntax which the Master Device looks for.

Following the /0 characters are the status characters which is a compilation of 8 bits:

Bit 7	Reserved
Bit 6	Always set
Bit 5	Ready Bit – Set when controller is ready to accept commands
Bit 4	Reserved
Bit 3-0	Status code: 0 = No Error 1 = Initialization Error 2 = Bad Command 3 = Operand Out of Range 7 = Overload Error

## Example Responses to command /1?4

FFh	RS485 line turn around character
2Fh	ASCII '/' Start character. The DT Protocol uses '/' as the start
30h	ASCII 0. Master Device's Address
60h	Status character as explained above
31h	Two bytes are the actual answer in ASCII. This is an eleven (11) which represents the status of the four inputs: Bit 0 = Switch 1 Bit 1 = Switch 2 Bit 2 = Opto 1 Bit 3 = Opto 2 (eleven is 1011 in binary)
03h	This is the ETX or end of text character. It's the end of the answer string
0Dh	Carriage Return
0Ah	Line Feed