

Simple Step Command Set and Communications Protocol

Command Syntax:

Commands to the Simple Step driver and controller are single alpha characters normally followed by a numeric value. The case sensitive alpha character represents “what to do” and the numeric value represents “how much to do it”.

Values for desired velocities, accelerations, positions, stepping resolutions, and currents can all be set. You can also query the program what position it is currently at and what velocity it currently possesses. Creating loops within the strings are easily possible, and storing these strings for later execution can also be done. These strings are stored on the EEPROM, enabling the Simple Step to power up into a mode of your choice. This is how it can act with no computer attached.

The commands are simply typed into a Terminal Program such as “HyperTerminal”, no special software is required. The Simple Step can even be commanded from a serial enabled PDA. Please refer to the Simple Step user guide for setting up Hyperterminal on your PC.

Command Set:

Command (Case Sensitive)	Operand	Description
Z	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.
z	0-max*	Sets current position to be position specified without moving motor
f	0 or 1	Flag polarity. Sets polarity of home sensor, default value is 0.
A	0-max*	Move Motor to Absolute position

Simple Step Command Set and Communications Protocol

Command (Case Sensitive)	Operand	Description
P	0-max*	Move Motor relative number of steps in positive direction. A value of zero will cause an endless forwards 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.
D	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. Then use this command.) 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.
F	0, 1	Reverses the positive direction to be negative. The P command will move the motor in the negative direction, and the D command will move the motor in the positive direction. Default is 0.
v	50-900	In Position Mode, this sets the Start Speed of the Motor in half steps per second.
V	5-5800	In Position Mode, this sets the Top Speed of the Motor in half steps per second.
V	5-2500	In Velocity Mode, you can change the Top Speed "on the fly". This is allowed when Top Speed < Start Speed (pps).
c	50-900	In Position Mode, this sets the stop speed of the motor in half steps per second.
L	1-20	In Position Mode, this sets the Acceleration factor (acceleration = L*7500 steps/sec ²)
N	0-1	Sets the micro step mode to on or off. Turning Micro step mode on allows the motor to be positioned in micro steps.

Simple Step Command Set and Communications Protocol

Command (Case Sensitive)	Operand	Description
R		Run the command string that is currently in the execution buffer
X		Repeat the current command string
g		Beginning of a repeat loop
G	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.
T		Terminate current command
M	0-30000	Delay for "M" milliseconds
H	Blank	Halt current command string and wait until condition specified. Wait for switch 2 closure
	01	Wait for low on input 1 (Switch 1)
	11	Wait for high on input 1 (Switch 1)
	02	Wait for low on input 2 (Switch 2)
	12	Wait for high on input 2 (Switch 2)
	03	Wait for low on input 3 (Opto 1)
	13	Wait for high on input 3 (Opto 1)
	04	Wait for low on input 4 (Opto 2)
	14	Wait for high on input 4 (Opto 2)
		Halted operation can also be resumed by typing /1R
J	0-3	On/Off Driver. It's a two bit Binary value: 3=11=Both Drivers On, 2=10=Driver2 on, Driver1 off, etc.
m	0-100	Sets "Fast Move" Current on a scale of 0 to 100% of the max current. When $V > v$, $m100 = 100\%$ of 1.25A for the 4118S series.
l	0-100	Sets "Slow Move" Current on a scale of 0 to 100% of the max current. Use this when $V < v$.
h	0-50	Sets the Hold Current on a scale of 0 to 50% of the max current.

Simple Step Command Set and Communications Protocol

s	0-15	Stores a program. Program 0 is executed on power up (Total of 25 commands max per string)
e	0-15	Executes the Stored Programs 0-15
j	2, 4, 8, 16, 32, 64	Adjusts the resolution in micro-steps per step. Resolution depends on model.
o	0-250	Allows user to correct any unevenness in microstep size
S	01 11 02 12 03 13 04 14	Skip next instruction if low on input 1 (Switch no. 1) Skip next instruction if hi on input 1 (Switch no. 1) Skip next instruction if low on input 2 (Switch no. 2) Skip next instruction if hi on input 2 (Switch no. 2) Skip next instruction if low on input 3 (Switch no. 3) Skip next instruction if hi on input 3 (Switch no. 3) Skip next instruction if low on input 4 (Switch no. 4) Skip next instruction if hi on input 4 (Switch no. 4)
?	0	Simple Step 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 = Input 0 Bit 1 = Input 1 Bit 2 = Opto 0 Bit 3 = Opto 1
?	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 Simple Step: 0 = No Error 1 = Initialization error 2 = Bad Command 3 = Operand out of range
		*10 ⁹ (32 Bit)

Simple Step Command Set and Communications Protocol

Communications Protocol:

The protocol is based on the Cavro serial communications protocol. The Simple Step defaults to communicate in the DT protocol mode.

DT Protocol syntax:

The DT Protocol allows the unit to be commanded over a simple serial port. HyperTerminal works well for this. The syntax for the DT protocol is as follows:

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

*To access drivers 10-16, use these commands:

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.

Simple Step Command Set and Communications Protocol

Example #1 (A Move to Absolute Position, delay, then a move to home)

/1A10000M500A0R

/	Start character
1	Device address
A10000	Motor turns to Absolute Position 10000
M500	Wait for 500 Milliseconds
A0	Motor will turn to Absolute Position 0
R	Run the command that it has received (Press Enter)

Example #2 (Moves with waits in a loop)

/1gA10000M500A0M500G10R

/	Start character
1	Device address
g	The start of a repeat loop
A10000	Motor turns to Absolute Position 10000
M500	Wait for 500 Milliseconds
A0	Motor will turn to Absolute Position 0
M500	Wait for 500 Milliseconds
G10	End of the repeat loop, which will repeat for 10 times
R	Run the command that it has received (Press Enter)

Simple Step Command Set and Communications Protocol

Example #3 (Program Storage and Results)

/1s2A10000M500A0M500G10R

The program outlined in the prior example is stored as Program 2.

/1e2R<CR>

This will execute the previously stored program #2. (Note: Program 0 is always executed on power up. If we used 0 instead of 2 in this example, then this program would execute automatically on power up.)

Example #4 (Set Current, Wait for Switch Push Button, Home to Opto)

/1s0J0m75h10H11ZJ3A5000A0R

/1s0	Store the following string as program 0 for driver #1
J0	Turn off both on/off drivers (i.e. for an LED)
m75	Set move current to 75% of max.
h10	Set hold current to 10% of max.
H11	Wait (Halt) for a switch closure on input 1, pin 2
Z	Home the stepper to opto #1
J3	Turn on both on/off drivers (i.e. for an LED)
A5000	Move to position 5000
A0	Move to position 0
R	Run (Press Enter)

Simple Step Command Set and Communications Protocol

Example #5 (Skip / Branch Instruction)

```
/1s0gA0A100S13e1G0R  
/1s1gA0A10S03e0G0R
```

Two “Subroutines” are stored in string0 and string1 and the code switches from one subroutine to the other depending on the state of input3. In the example given, the code will cycle the motor between position A0 and A100 if input3 is High and between A0 and A10 if input3 is Low.

Stored string0:

```
/1s0      Store the following string as program 0 for driver #1  
g        Begin loop  
A0       Move motor to the absolute position 0  
A100     Move motor to the absolute position 100  
S13     Skip next instruction if 1 (hi) on input 3  
e1      Jump to string1  
G0      Run this loop infinitely  
R       Run command  
          (Press Enter)
```

Stored string1:

```
/1s1     Store the following string as program 1 for driver #1  
g       Begin loop  
A0      Move motor to the absolute position 0  
A10     Move motor to the absolute position 10  
S03     Skip next instruction if 0 (low) on input 3  
e0      Jump to string0  
G0      Run this loop infinitely  
R       Run command  
          (Press Enter)
```

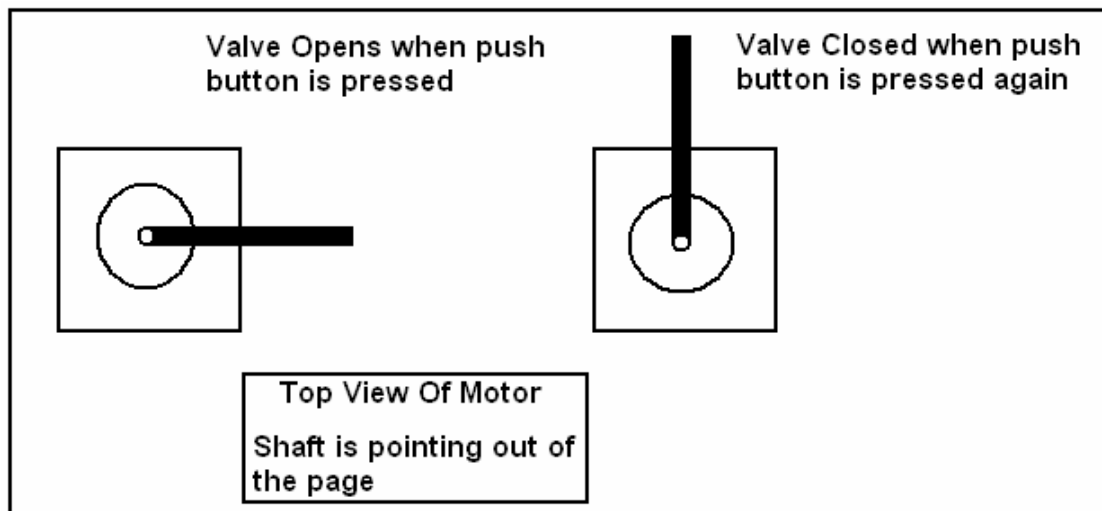
Note: Input 1 corresponds to Pin 2
 Input 2 corresponds to Pin 8
 Input 3 corresponds to Pin 7
 Input 4 corresponds to Pin 5

Simple Step Command Set and Communications Protocol

Example #6: (Open and Close a Valve with One Push Button)

/1s0j2gH12P100H12D100G0R

/1s0 Store the following string as program 0 for driver #1
j2 Step resolution set to half stepping
g Begin loop
H12 Wait for a switch closure (push button) on input 2, pin 8
P100 Move in the positive direction by 100 steps (90 degrees)
H12 Wait for another switch closure (push button) on input 2, pin 8
D100 Move in the negative direction by 100 steps
G0 Loop continuously
R Run
(Press Enter)



Simple Step Command Set and Communications Protocol

Example #7: (Push Button to Run Motor, Opto Sensor stops the Motor)

`/!s0gH11P10000H11Z10000G0R`

`/!s0` Store the following string as program 0 for driver #1
`g` Begin loop
`H11` Wait for a Halt Switch Closure (push button) on pin 2
`P10000` Move motor in the Positive Direction by 10000 steps
`H11` Wait for another Halt Switch Closure (push button) on pin 2
`Z10000` Move back by 10000 steps, or until a sensor is interrupted
`G0` Loop continuously
`R` Run Program
(Press Enter)

