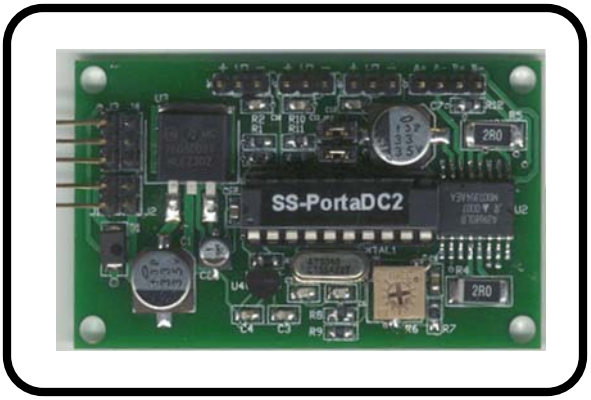
SS-DC268 Manual

A Super Stepper Command Bus Architecture Product

Avayan Electronics

Dual DC Motor Controller – Super Stepper Architecture



Dual DC Motor Controller (SS-DC268):
Controls 2 DC Motors with up to 650 mA of current each.
Each motor can be enabled or disabled independently.
Each motor can be braked independently.
Command both directions to run indefinitely.
Command both directions to run on a time basis.
Command both directions to run on a number of encoder pulses basis.
Sensor input per motor to control features such as home and pulse accumulation.

The SS-DC268 controller is a small brush DC motor controller. The board works with 12V and will supply motor currents of up to 650 mA per phase.

The board is pretty flexible and contain features such as direction control, motor enabling, disabling and braking, motor homing motor shaft encoder pulse accumulation. All of these features are available for both Motor A and Motor B. One extra feature available to Motor A alone is the ability to measure shaft encoder period.

A potentiometer controls the maximum amount of current being supplied to the motor winding. The Vref Test Hole allows easy access to the Reference Voltage that needs to be tweaked in order to obtain proper current supply.

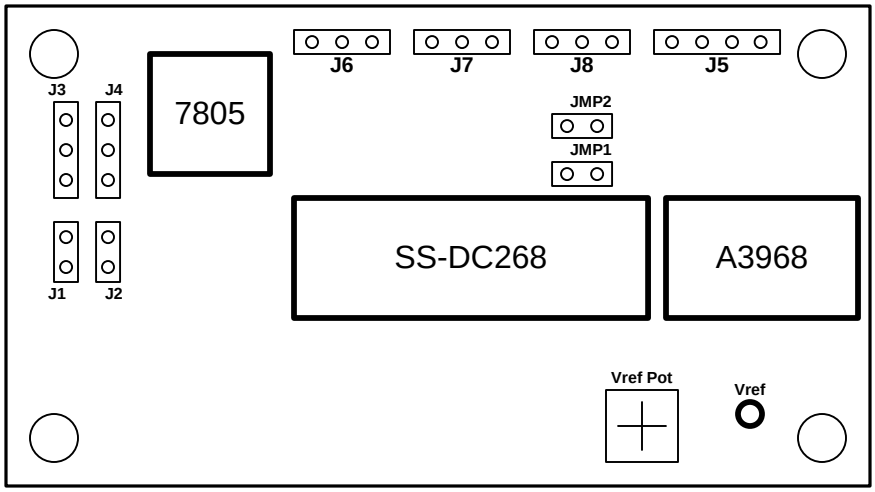
The module receives command through the SSB (Super Stepper Bus). Two jumpers, JMP1 and JMP2, configure the controller UART for either 8 or 9 bit communication and a different range of BAUD Rates and addresses.

Commands received through the bus are executed immediately. Available commands for the SS-DC268 are shown on Table 1.

Super Stepper DC268							
Opcode	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Description	# of Bytes
0	Hm,En/Ds,Dir,Mtr	-	-	-	-	En/Dis Motor	5
1	Dir,Mtr	Steps Hi	Steps Lo	-	-	Move Steps	4
2	Dir,Mtr	Time Hi	Time Lo	-	-	Move Time	3
3	Mtr Sel	-	-	-	-	Brake Motor	3
4	Mtr Sel	-	-	-	-	Tacometry Clear	1
5	-	-	-	-	-	EXPANDED	1
6	Mem Sel	Address	Data	-	-	Write	4
7	Mem Sel	Address	-	-	-	Read	3

Table 1

SS-DC268 Layout and signals:



J1 and J2 Power Connector. Connect a power source from 9V to 15V. J2 is a bypass to another Porta Board piggy backed on top.

Pin1: Positive Power Source
Pin2: Ground

J3 and J4 Serial Communications Connector: Connect the TTL level serial communication signal as obtained from a PC through an RS-232 driver or any microcontroller UART output.

Pin1: Module Receive (Rx)
Pin2: Module Transmit (Tx)
Pin3: Ground

J5 Dual DC Motor Output Connector. Please observe polarity for proper rotation.

Pin1: Motor A+ terminal
Pin2: Motor A- terminal
Pin3: Motor B+ terminal
Pin4: Motor B- terminal

J6 and J7 Motor A (J6) and Motor B (J7) Input Sensor Connector. Connect a shaft encoder optical sensor if the motor is to move on a number of steps basis. Connect an optical sensor or switch if the motor is to be homed.

Pin1: Vcc
Pin2: Input Signal
Pin3: Ground

J8 Home Sensor Input. Connect a shaft encoder optical sensor to determine period from one pulse to the next. when the home position has been reached.

Pin1: Vcc
Pin2: Input Signal
Pin3: Ground

NOTE: Inputs J6 to J8 signal are asserted low. If using a switch, please connect with respect to ground. The input signal has a pull up to Vcc.

SS-DC268 Layout and signals (Cont.):

JMP1 MOD1 Jumper. Selects from UART hardwired parameters or EEPROM stored parameters.
- Open: BAUD Rate is 2400 and SS-ADDRESS is 0.
- Closed: BAUD RATE is stored on EEPROM address 0 and SS-ADDRESS is stored on EEPROM address 1.

JMP2 MOD0 Jumper. Selects from 8 bit or 9 bit communications.
- Open: 8 bit communications selected.
- Closed: 9 bit communications selected.

Vref Test Hole. Measure the Reference Voltage. The user must configure the potentiometer position while looking at this test hole and following the Maximum Current Setting Equation:

Max Current = Vref/8

Commands:

Enable/Disable Motor A/B: (Opcode 0) This command will enable or disable either motor A or motor B to move in any direction either indefinitely or until a home sensor trips. The first byte parameter contains four configuration bits which make this command so flexible.

SS ADDRESS	0	X	X	X	X	H	E	D	M

Figure 1

M Bit (Motor Select) This bit specifies to which motor the command is intended. 1 = Motor A, 0 = Motor B
D Bit (Direction) This bit specifies which direction will the selected motor move. 1 = clockwise, 0 = counterclockwise
E bit (Enabled) This bit specifies whether the motor will be enabled or disabled. 1 = enabled, 0 = disabled
H bit (Home) This bit specifies whether the motor will move until its home sensor trips. 1 = homing, 0 = not homing

H	E	D	M	Description
0	0	0	0	Motor B is disabled
0	0	0	1	Motor A is disabled
0	0	1	0	Motor B is disabled
0	0	1	1	Motor A is disabled
0	1	0	0	Motor B is enabled CCW
0	1	0	1	Motor A is enabled CCW
0	1	1	0	Motor B is enabled CW
0	1	1	1	Motor A is enabled CW
1	0	0	0	Motor B is disabled
1	0	0	1	Motor A is disabled
1	0	1	0	Motor B is disabled
1	0	1	1	Motor A is disabled
1	1	0	0	Motor B is enabled CCW until it reaches home
1	1	0	1	Motor A is enabled CCW until it reaches home
1	1	1	0	Motor B is enabled CW until it reaches home
1	1	1	1	Motor A is enabled CW until it reaches home

Table 2

Move Motor A/B for an X number of shaft encoder pulses: (Opcode 1) This command starts to move Motor A or Motor B in any direction until a certain number of shaft encoder pulses have been accounted for. The command utilizes the respective motor input sensor (J6 for motor A and J7 for motor B). First parameter byte contains bit to select motor and direction. Consecutive two parameter bytes contain a word with the number of pulses to count. This command always enables the motor.

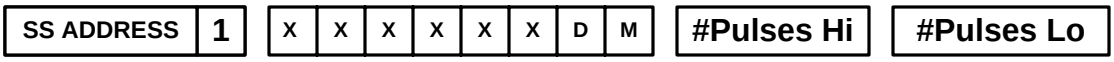


Figure 3

M Bit (Motor Select) This bit specifies to which motor the command is intended. 1 = Motor A, 0 = Motor B
D Bit (Direction) This bit specifies which direction will the selected motor move. 1 = clockwise, 0 = counterclockwise

D	M	Description
0	0	Motor B moves CCW until #Pulses have been accounted for
0	1	Motor A moves CCW until #Pulses have been accounted for
1	0	Motor B moves CW until #Pulses have been accounted for
1	1	Motor A moves CW until #Pulses have been accounted for

Table 3

Move Motor A/B for an amount of time: (Opcode 2) This command starts to move Motor A or Motor B in any direction until a certain number of milliseconds have elapsed. Maximum amount of time to have the motor enabled is 65.535 seconds. First parameter byte contains bits to select motor and direction. Consecutive two parameter bytes contain a word with the number of milliseconds that will be counted towards disabling the motor. This command always enables the motor.

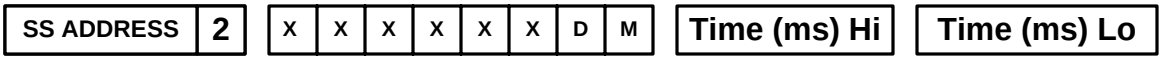


Figure 4

M Bit (Motor Select) This bit specifies to which motor the command is intended. 1 = Motor A, 0 = Motor B
D Bit (Direction) This bit specifies which direction will the selected motor move. 1 = clockwise, 0 = counterclockwise

D	M	Description
0	0	Motor B moves CCW until Time in ms elapses
0	1	Motor A moves CCW until Time in ms elapses
1	0	Motor B moves CW until Time in ms elapses
1	1	Motor A moves CW until Time in ms elapses

Table 4

Brake Motor A/B: (Opcode 3) This command brakes Motor A or Motor B stopping it very fast. The first byte parameter is used to determine which motor will be braked.

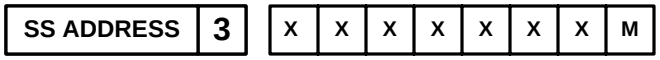


Figure 4

M Bit (Motor Select) This bit specifies to which motor the command is intended. 1 = Motor A, 0 = Motor B

Clear Motor A/B Tachometry: (Opcode 4) This command clears the respective tachometry register found on the RAM. The first byte parameter is used to determine which motor tachometry register will be cleared. Motor A Tachometry Register is found on address 0x00 while Motor B Tachometry Register is found on address 0x02.

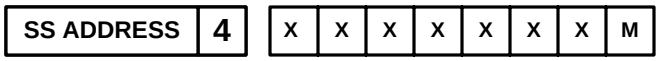


Figure 4

M Bit (Motor Select) This bit specifies to which motor the command is intended. 1 = Motor A, 0 = Motor B

Write RAM/EEPROM: (Opcode 6) Writes to internal RAM or EEPROM memory. The MEM SEL parameter specifies which type of memory will be accessed. Logic Low (0) accesses RAM while Logic High (1) accesses EEPROM. All writes are done to byte memory spaces. Writing to a word is not supported, so user must do two consecutive word writes to write to a word memory allocation, if needed.

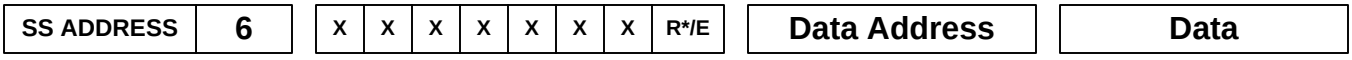


Figure 5

Read RAM/EEPROM: (Opcode 7) Reads from internal RAM or EEPROM memory. The MEM SEL parameter specifies which type of memory will be read and if the read request will return a word or a byte. Only reads to RAM can return a word. EEPROM reads always returns a byte.

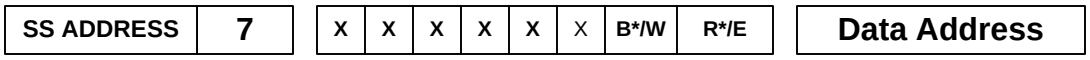


Figure 6

B*/W	R*/E	Description
0	0	Read a RAM Byte from ADDRESS
0	1	Read an EEPROM Byte from ADDRESS
1	0	Read a RAM Word (16bit) from ADDRESS
1	1	Read an EEPROM Byte from ADDRESS

Tachometry Features:

Each motor contain a respective Sensor Input which can be utilized for homing purposes or pulse accumulation. J6 is the Motor A sensor input connector. J7 is the same input but for Motor B. By connecting an optical, magnetic or any other sensor, the SS-DC268 can count shaft encoder pulses. These pulses in turn correlate with position and distance. Although this application is not as accurate or precise as other implementations with more sophistication, the SS-DC268 offers the pulse accumulation to those users that want to know a little bit more.

The sensor inputs are always counting up. It is recommended that the user clears the tachometry registers before enabling the respective motor. Opcode 5 has been coded to make it easier to clear these RAM locations. Writing to RAM is as effective as submitting the command, but then two writes are necessary.

NOTE: Because J6 and J7 is a shared resource, the user must select whether the input will be used to home the motor or count pulses.

Period Measurement:

Sometimes it is useful to determine how fast a motor is moving. To do this, a shaft encoder and a period counter are used, The SS-DC268 has a third sensor input which works just this way. It measures distance in time from one rising edge to the next. Hence, if the motor is moving fast, the number will be small while if it is moving slow, the number will be larger. If the motor is not moving at all, the number will remain static.

BAUD Rate Selection:

The SS-DC268 board communicates through the Super Stepper Serial Communication Protocol. Since the board has been implemented with a 4MHz crystal, certain BAUD Rates can be obtained. Programming the EEPROM with one of the values found on Table 3 will configure the UART to operate with the selected BAUD Rate after the device comes out of power on reset next time. This feature is of course available if the JMP1 is closed.

Baud Rate	EEPROM
2400	103
4800	51
9600	25
14400	16
19200	12
28800	8
38400	6
57600	3
76800	2
115200	1

Table 3

EEPROM Memory Map:

The user must refer to this section whenever the EEPROM is to be programmed. Programming the EEPROM memory on the device may result on non proper functionality.

EPROM Addre	Contents
0x00	SS BAUD Rate
0x01	SS Address
0x02 to 0xFF	User defined

Table 4

RAM Memory Map:

The user must refer to this section whenever the RAM is to be read or written. Modifying RAM memory on the device could result on non proper functionality.

RAM Address	Contents
0x00	Motor A Tachometry Register Hi
0x01	Motor A Tachometry Register Lo
0x02	Motor A Tachometry Register Hi
0x03	Motor A Tachometry Register Lo
0x04	Motor A Shaft Encoder Period Hi
0x05	Motor A Shaft Encoder Period Lo
0x06 to 0x17	User Defined
0x18 to 0x1F	Do not Use

Table 5