

Mechatronics 2 Addon Module

This brief guide will give an overview of the features of the mecharonics 2 add-on module. It will also give details on how to control the stepper motor.

The module interfaces with ports A, B and C of the PIC. The functions for each pin are given in the table below. Pins which are not listed are not connected. It also uses the CPLD to control the stepper motor and route signal to peripheral modules. The CPLD must be programmed with the appropriate JED file.

Pin	Function
<u>PortA</u>	
0	Short Range IR
1	Long Range IR
3	+Vref (@ 4.09v)
<u>PortB</u>	
All pins	Pass through to header
<u>PortC</u>	
0	CPLD_CS (0)
1	CPLD_CS (1)
2	SM_STEP
3	SPI_CLK
4	SPI_SDI
5	SPI_SDO
6	USART_TX
7	USART_RX

The module draws power from the Create through the D25 cable. The DSX kit should not be powered by a plug pack when connected to the Create. The module also communicates with the Create through the D25 cable using the USART protocol. You will need to set up the USART to communicate at 57600 baud and refer to the open interface manual for the appropriate commands. You will also need to disable serial communications in Kirra to avoid communication conflicts.

Communication with the stepper motor controller, eeprom and radio frequency controller are accomplished using the SPI communication module. The SPI module should sample input data in the middle of the output data pulse, should transmit on a transition from active to idle, should have a low level idle state and be triggered from the internal instruction cycles. The SPI signals are routed to the appropriate module inside the CPLD according to the CPLD_CS (1,0) bits.

CPLD_CS [1,0]	Module
00	No Module selected
01	RF Module Selected
10	EEPROM Module Selected
11	Stepper Module Selected

When connected to either the EEPROM or the RF modules the SPI lines are routed within the CPLD to the appropriate pins on the target device. Refer to the appropriate datasheet for details on communication protocols. The stepper motor controller is implemented within the CPLD and is controlled as follows.

The stepper motor module is usually controlled by writing a configuration byte using the SPI interface followed by repeated pulsing of the SM_STEP pin each time a step is desired. If you want to change directions you must write a new control byte to the module. The function of each bit in the control byte is detailed in the table below.

SM_Control_Byte	Function
0	1 = Enable, 0 = Disable
1	1 = Clockwise, 0 = Counterclockwise
2	1 = Half Step, 0 = Full Step
3	Clock Select 0
4	Clock Select 1
5-7	Don't Care

The stepper motor module can receive a step pulse from a number of sources selected by the state of the clock select bits (4,3).

Clock Select (1,0)	Clock Source
00	No Clock
01	SM_STEP (PIC)
10	CLK5 (Kirra clock 1)
11	CLK6 (Kirra clock 2)

If the reset button on the CPLD is pressed the shift registers and control bytes of the Stepper Motor module are cleared and the state machine is reset to energise winding A1.

The wiring for the stepper motor is as follows

Screw Terminal	Colour
Y1	Black
Y2	Orange
Y3	Brown
Y4	Yellow
VDC	Red