Modern Robotics, Inc
Core Device Interface Module

Version 1.1   April 20, 2015

1 Contents

2 Document Control

Revision History			
Version	Date	Description	By
1.0	November 5, 2014	Initial document	Modern Robotics
1.1	April 20, 2015	Minor updates	Modern Robotics

Specifications subject to change without notice.

Modern Robotics, Inc
13335 SW 124th St
Miami, FL 33186

Phone: (786) 393-6886
Email: support@modernroboticsinc.com
Web: www.modernroboticsinc.com

3 Overview

The Modern Robotics Core Interface Device has 26 ports for connecting sensors and devices that can be read from a host via USB and. The Core Device Interface, or CDI, provides 8 digital I/O ports, 8 analog inputs, 2 analog outputs, 2 PWM outputs and 6 I2C bus ports. The CDI is powered from the 5v available from the USB connection.

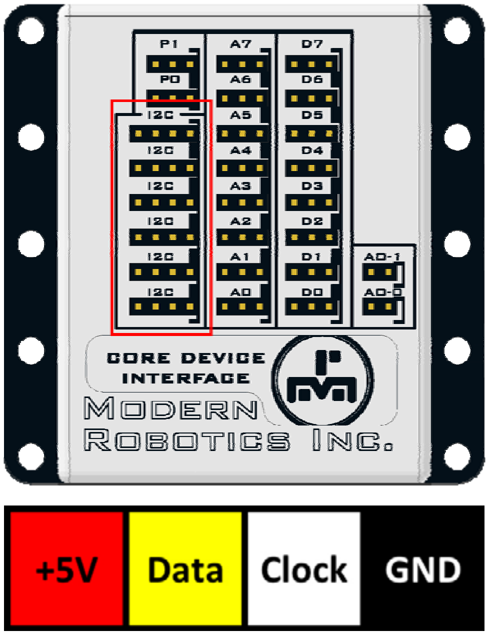
To access ports on the CDI, requests are written to the memory map and results are read from the memory map. The structure allows the entire memory map to be read or written in one operation and registers that are *read only* are protected by the firmware so writes to these registers will be ignored.

The functions available will depend on the programming environment implementation and the host system.

4 Ports

The CDI has an array of ports for connection of different sensors and devices. These are arranges in groups based on the function. Ground is pin 1 on the CDI and signified by the black shading adjacent to the pin. Note, the connector block color coding matches the Modern Robotics sensor connection wires and pigtails.

4.1 I2C



The six I2C ports are on a common I2C bus so each I2C device must have a different I2C address. For Modern Robotics I2C sensors, the address can be selected to avoid addressing conflicts.

The I2C read functionality is defined by a data structure used to initialize I2C reads.

Mode	I2C addr	Mem addr	Mem len	B0 – B26	Flag
byte	byte	byte	byte	bytes	byte

The *Mode* byte controls the overall functionality of the channel.

4.1.1 Mode byte

D7	D6	D5	D4	D3	D2	D1	D0
R/W	-	-	-	-	-	-	-

The *R/W* bit is used to control what kind of I2C transaction to perform. If *R/W* bit is set, the transaction will be read. If *R/W* bit is clear, the transaction will be a write.

4.1.2 I2C addr

The *I2C addr* byte contains the I2C hardware address of the device to be accessed. This address byte is transmitted as the first byte of an I2C transaction. This byte must be an even 8 bit address, ie; bit 0 = 0. **Note:** Some documentation refers to I2C addresses as a 7 bit address and if so this refers to the upper 7 bits of the address b1 – b7. The CDI treats the address as an 8 bit value and the firmware will automatically adjust the address to be read or write as needed based on the function selected. Addresses expressed as 7 bit values must be provided as an 8 bit value in the form of 7bit address + 0 in the low order bit

4.1.3 Mem addr

The *Mem addr* byte determines what memory map address byte is transmitted following the I2C addr byte during an I2C transaction.

4.1.4 Mem len

The *Mem len* byte determines how many bytes will be transferred to/from the device starting from the *Mem addr*. This value must be between 1 – 27.

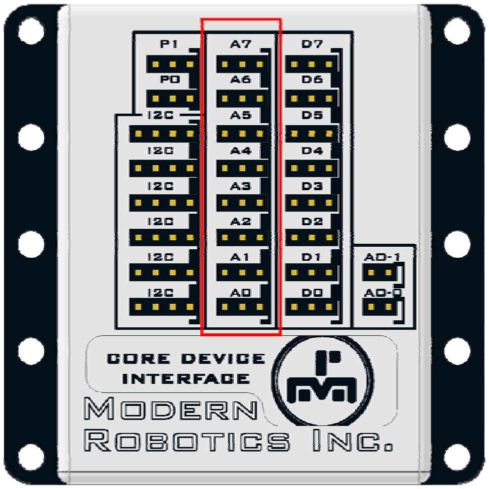
4.1.5 B0 – B26

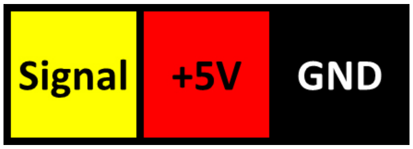
The *B0 – B26* bytes are the buffer from which data is transmitted, or into which received data is placed.

4.1.6 Flag byte

The *Flag* byte is used to activate an I2C transaction. The *Flag* byte defaults to 00H indicating that the write channel is idle. After the structure has been written, the final byte will be the *Flag* byte which should be written to FFH. The *Flag* byte will auto-clear to 00H when the transaction has completed. The flag byte can be conveniently monitored by checking the corresponding *Buffer flag status byte.*

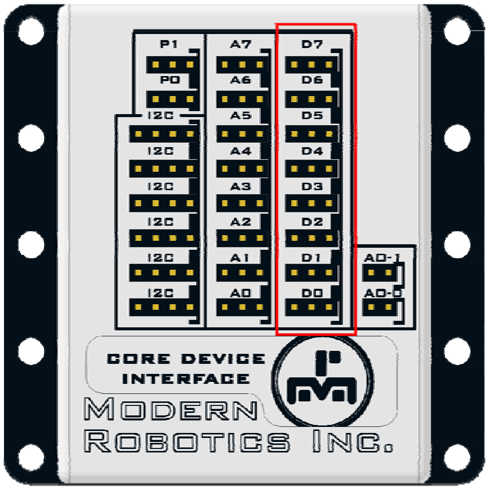
4.2 Analog Input

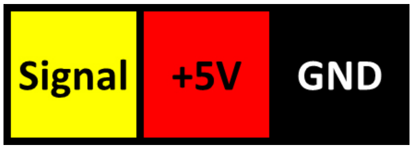




The 8 Analog Input ports have 3 pin headers. The analog signal values can be from 0v – 5v and the output is a 10 bit A/D, providing the program register with a value in the range of 0 – 1023. Note: If nothing is connected to an analog port the program register will contain a random value.

4.3 Digital Input/Output

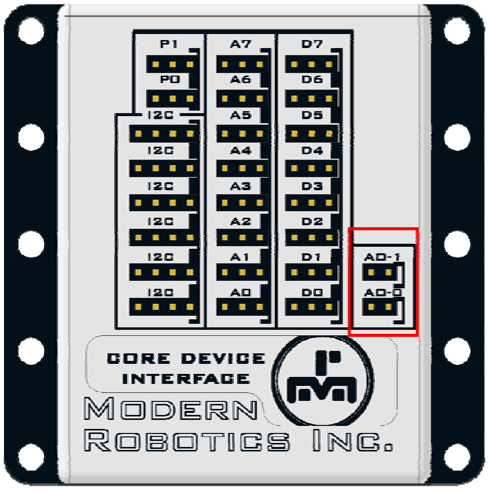


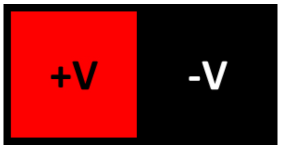


The 8 Digital Input/Output ports, *D7 – D0*, have 3 pin headers. Each port can be individually set as input or output. There are three registers that control the digital ports, *input state, I/O control* and *output set*.

The *D7 – D0 input state* field is a byte containing the current logic levels present on the D7 – D0 channel pins. If a particular pin is in output mode, the current output state will be reported.
The *D7 – D0 I/O control* field is a byte containing the required I/O state of the D7 – D0 channel pins. If a particular bit is set to one, the corresponding channel pin will be in output mode, else it will be in input mode.
The *D7 – D0 output set* field is a byte containing the required I/O output of the D7 – D0 channel pins. If the corresponding *D7 – D0 I/O control* field bit is set to one, the channel pin will be in output mode and will reflect the value of the corresponding *D7 – D0 output set* field bit.

4.4 Analog Output





The Analog Output ports are 2 pin connectors providing signal and ground. There are three values for each of the two Analog Output ports that control the mode and output voltage of each port. These are *output mode, output voltage* and *output frequency*.

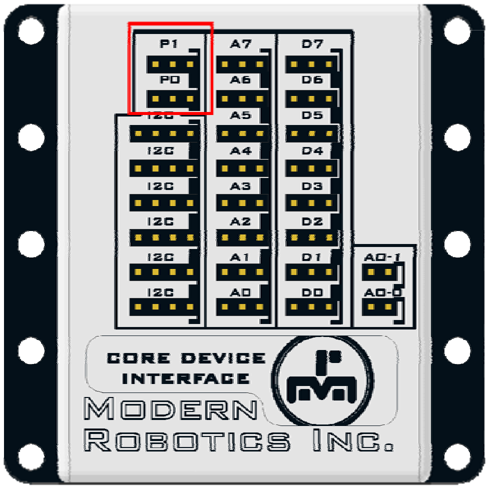
The output mode can be set to one of the following;

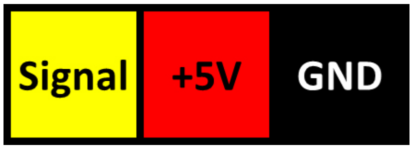
Mode	Channel operation	Voltage range
0	Voltage output	-4v – +4v
1	Sine wave output	0 – ±4v (8v p-p)
2	Square wave output	0 – ±4v (8v p-p)
3	Triangle wave output	0 – ±4v (8v p-p)

The *output voltage* field will set the channel output voltage in the range -4 to +4 volts in *Voltage output* mode (mode 0) based on the value that can range from -1023 to 1023. If the mode is in waveform mode (1,2 or 3), this field controls the peak to peak voltage in the range 0 – 8 volts based on the value being 0

to 1023. If the voltage value is set outside of the allowable range, it will be modified to force the value in range.
The *output frequency* field will set the channel output frequency in the range 1 – 5,000Hz based on the value being in the range 1 to 5,000 if the *Mode* is waveform (1, 2 or 3). If *Mode 0* is selected, this field will be over-written to 0.

4.5 PWM Output





The PWM Output ports, P0 and P1 output a Pulse Width Modulated (PWM) signal. Each port is individually controllable and the output time (on time) and the period can both be set.

The *P0/P1 output on time* field sets the pulse width for the channel output in units of 1μS. Setting a value which is greater than the *P0/P1 output period* will result in the output being permanently set to logic 1.
The *P0/P1 output period* field sets the pulse repetition period for the channel output in units of 1μS. If the PWM feature is being used to generate pulses for a standard R/C style servo, the output period should be set to 20,000 and the output on time should be set within the range 750 – 2,250 for most servos. Setting a pulse width outside these limits may damage the servo.

4.6 LED Set

The CDI has two built in LEDs, one red and one blue that can be controlled by the user. The *LED set* field, 0x17 in the memory map is a byte can be set as follows to turn on the LEDs.

0x01 – turn on the red LED
0x02 – turn on the blue LED

5 Memory map.

Address	Type	Contents	R/W
00H	byte	Device version number	r/o
01H	byte	Manufacturer code	r/o
02H	byte	Device id.	r/o
03H	byte	Buffer flag status byte	r/o
04, 05H	word	A0 ADC value	r/o
06, 07H	word	A1 ADC value	r/o
08, 09H	word	A2 ADC value	r/o
0A, 0BH	word	A3 ADC value	r/o
0C, 0DH	word	A4 ADC value	r/o
0E, 0FH	word	A5 ADC value	r/o
10, 11H	word	A6 ADC value	r/o
12, 13H	word	A7 ADC value	r/o
14H	byte	D7 – D0 input state	r/o
15H	byte	D7 – D0 I/O control	r/w
16H	byte	D7 – D0 output set	r/w
17H	byte	LED set	r/w
18, 19H	word	V0 output voltage	r/w
1A, 1BH	word	V0 output frequency	r/w
1CH	byte	V0 output mode	r/w
1DH	byte	not used	r/o
1E, 1FH	word	V1 output voltage	r/w
20, 21H	word	V1 output frequency	r/w
22H	byte	V1 output mode	r/w
23H	byte	not used	r/o
24, 25H	word	P0 output on time	r/w
26, 27H	word	P0 output period	r/w
28, 29H	word	P1 output on time	r/w
2A, 2BH	word	P2 output period	r/w
2C – 2FH	bytes	Reserved	r/o

30 – 4FH	struct	I2C port I2C0 buffer	r/w
50 – 6FH	struct	I2C port I2C1 buffer	r/w
70 – 8FH	struct	I2C port I2C2 buffer	r/w
90 – AFH	struct	I2C port I2C3 buffer	r/w
B0 – CFH	struct	I2C port I2C4 buffer	r/w
D0 – EFH	struct	I2C port I2C5 buffer	r/w

r/o – read only. r/w – read/write

The *Device version number* field will report a revision number as two 4 bit nibbles, the high nibble being the major version number and the lower nibble being the revision level.

The *Manufacturer Code* field will contain 4DH.

The *Device id.* field will contain 41H.

The *Buffer flag status byte* field returns six bits reflecting the status of the six buffer flags.

D7	D6	D5	D4	D3	D2	D1	D0
-	-	Buffer 5 flag	Buffer 4 flag	Buffer 3 flag	Buffer 2 flag	Buffer 1 flag	Buffer 0 flag

The buffer flag bits reflect the values contained within the port buffer flag bytes. If the flag byte is 0, the corresponding bit will be set to 0. If the flag byte is non zero, the corresponding bit will be set to 1.