

SerIO User Guide

2010.04.06

Overview

The SerIO is a serial interface platform for creating physical computing projects controlled from a computer. The board gives access to 14 digital input and output pins, six analog output pins and eight analog input pins through a serial interface. This allows the creation of physical computing projects without having to modify any code on the hardware platform. Projects that involve the use of analog sensors, button, LEDs, buzzers, motors and many more electronic components can be controlled from any environment that has access to a serial port; some examples or controlling environments are Visual Basic, Python, LabView and Processing, but it's certainly not limited to those applications!



Powering the Device

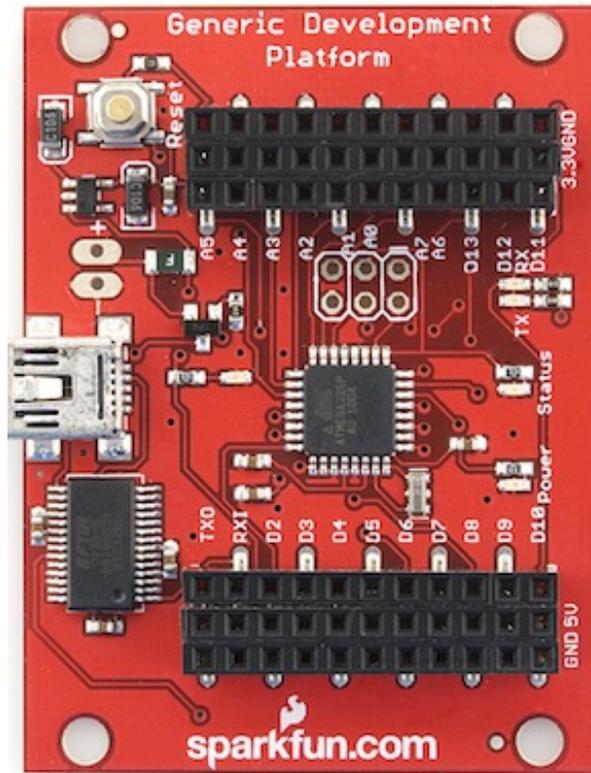
The SerIO is powered from a miniUSB cable. While the USB cable provides 5 Volts, the board is run from 3.3 Volts. It is important to remember this when interfacing to the board as inputs to the SerIO (analog or digital) should not exceed 3.3 Volts.

The USB cable is also used to provide the serial link to the SerIO board.

Adding Peripheral Hardware to the SerIO

The SerIO has two blocks of headers which are used to add peripheral hardware. Each block consists of one Ground column, one Power column, and 11 individual I/O pins. The function and name for each pin is described on the SerIO label. It's important to closely observe the labels on the SerIO; notice the power column on one side of the device is a 5 Volt power rail while on the other side of the device there is a 3.3 Volt power rail.

There are 14 digital I/O pins on the SerIO. The digital I/O pins are labeled on the SerIO device with the 'D' prefix (D0-D13). A digital I/O is a pin that can be configured as an input or an output. A digital input is an input that determines if a pin is high (1) or low(0).



The SerIO board has 6 analog output pins, also known as PWM pins. The analog output pins are labeled D3, D5, D6, D10 and D11. These labels are colored black on the SerIO label instead of white. These pins can be configured as digital inputs, digital outputs or analog outputs. Notice that only the pins that are labeled in black text can be used as analog outputs. An analog output pin can be used to generate an equivalent voltage on a pin by assigning the pin a value from 0 (0 Volts) to 255 (3.3 Volts).

SerIO also has 8 analog input pins. The analog input pins are labeled on the SerIO device the the 'A' prefix (A0-A7). An analog input is used to determine the voltage level of a pin. For example an analog distance sensor can be connected to an analog input pin; by reading the voltage level of the analog input pin the distance from the sensor to an object can be determined. Reading an analog input pin will return a value between 0 (0 Volts) and 1023 (3.3 Volts).

Using the Serial Command Interface

The versatility of the SerIO comes from the Serial Command Interface. The Interface allows the manipulation of the digital and analog pins by sending text commands from a computer through the USB cable. The command interface allows physical computing projects to be created without writing 'firmware,' which means that the SerIO never has to be reprogrammed. The boards inputs and outputs can all be controlled from a computer; so if you already know a programming language like Visual Basic, Python, Labview, Processing, Java, etc. you can use this knowledge to create a physical computing project rather than having to learn a new programming language.

There are 6 types of commands supported by the SerIO: Configure Pin, Read Digital Pin, Read Analog Pin, Read All

SerIO User Guide

2010.04.06

Pins, Write to Digital Pin and Echo Toggle. After a command is entered the SerIO board will send an appropriate response. All commands are ended with a carriage return character (\r) or by pressing the enter key. By default, the command entered by the user will be echoed back to the computer; this can be disabled with the Echo Toggle command. Disabling the echo makes handling and parsing command responses easier and faster; this is highly recommended!

Interface Commands

Command:	Configure Pin
Description:	Configures a digital pin as an input or output.
Parameters:	Pin number, Input(0) or Output(1)
Syntax:	C,pin_number,input/output\r
Response:	Pin: pin_number Input/Output: new pin configuration *Returns 'Invalid Command' if invalid command syntax is entered.
Example:	Configure Pin 3 as Input. Command – C,3,0\r (\r is equivalent to pressing 'enter'). Response - Pin: 3 Input/Output: 0\r Configure Pin 10 as Output. Command – C,10,1\r Response – Pin: 10 Input/Output: 1\r

Command:	Read Digital Pin
Description:	Reads the value of an input pin (D0-D13). The pin must be configured as an input before using this command.
Parameters:	Digital Pin Number
Syntax:	D, digital pin number\r
Response:	Pin: pin_number Value: read_value(0 or 1) *Returns 'Invalid Command' if invalid command syntax is entered.
Example:	Read pin D9. Command - D,9\r Response - Pin: 9 Value: 0\r (Assuming the value of pin 9 is Low) Read pin D4. D, 4\r Pin: 4 Value: 1\r (Assuming the value of pin 4 is High)

SerIO User Guide

2010.04.06

Command:	Read Analog Pin
Description:	Reads the value of an analog input pin (A0-A7)
Parameters:	Analog Pin Number
Syntax:	A, analog pin number\r
Response:	Pin: analog pin number Value: ADC Count** *Returns 'Invalid Command' if invalid command syntax is entered. **See ADC Count Conversion Note for more information on the ADC count value.
Example:	Read pin A3. Command - A,3\r Response - Pin: 3 Value: 489\r

Command:	Read All Input Pins
Description:	Reads all of the digital or analog pins currently configured as inputs
Parameters:	Analog (A) or Digital (D)
Syntax:	R,input type\r
Response:	Pin: lowest input pin number Value: associated value Pin: highest input pin number Value: associated value *Returns 'Invalid Command' if invalid command syntax is entered.
Example:	Read all digital input pins . Command – R,D\r Response - Pin: 3 Value: 0\r Pin: 4 Value: 0\r Pin: 7 Value: 1\r Pin: 10 Value: 1\r Pin: 11 Value: 0\r (Assuming digital pins 3, 4, 7, 10 and 11 are configured as inputs and the rest of the digital pins are configured as outputs. The values correspond to the value on the pin.)

Command:	Write to Pin
Description:	Writes a high or low value to a digital pin. Also writes an analog value to a pin if it is an analog output (PWM) pin.
Parameters:	Pin Number, Value (0-255) Note: A digital output can only be assigned a value of 0 or 1; if a value higher than 1 is assigned it will be ignored and set to 1. Only analog output pins can have a value greater than 1.

SerIO User Guide

2010.04.06

	*See ADC Conversion Note for information on how to set the value of an analog output.
Syntax:	W,digital pin number, value\r
Response:	Pin: digital pin number Value: assigned value *Returns 'Invalid Command' if invalid command syntax is entered.
Example:	Set digital pin 8 low (0). Command – W,8,0\r Response – Pin: 8 Value: 0\r Set Analog Output pin 11 to 100. Command – W,11,100\r Response – Pin:11 Value: 100\r

Command:	Echo Toggle
Description:	Controls if the input command will be echoed back. By default this setting is enabled. If you don't wish to see the command being sent to SerIO disable this setting
Parameters:	On(1) or Off(0)
Syntax:	E, setting
Response:	None *Returns 'Invalid Command' if invalid command syntax is entered.
Example:	Turn off the command echo. Command – E,0\r Turn on the command echo. Command – E,1\r

ADC Conversion Note

When an analog reading is made with the SerIO the response is returned as a number between 0 and 255. This number corresponds to a voltage on the pin that was read. When the SerIO reads a voltage, the voltage is sent to a module called an analog to digital converter. This module takes the voltage, which is in the range of 0 Volts to 3.3 Volts, and converts it to a number that can be manipulated in software. This number is called the ADC Count. The ADC count, though, usually needs to be converted back into a voltage in order to derive the meaning of the count. To convert the ADC count to a voltage, use this equation:

$$\text{Voltage in millivolts} = \frac{\text{ADC Count} * 3300}{1024}$$

This formula will find the number of millivolts on the pin. There are 1000 millivolts(mv) in a Volt(V).

Likewise when writing to analog output pin the value for the pin must be supplied in a digital format; the ADC count must be given to the SerIO rather than a voltage. To find the ADC count for a desired voltage use this formula:

SerIO User Guide

2010.04.06

ADC Count = Voltage in millivolts * 255
3300

This will result in the number to be used in the Write Analog Voltage command to set a desired voltage. The maximum voltage that can be produced is 3.3V or 3300 mV; the highest valid ADC count that can be sent to the command is 255.