Buttonpad Controller USB User Guide

2009.01.26

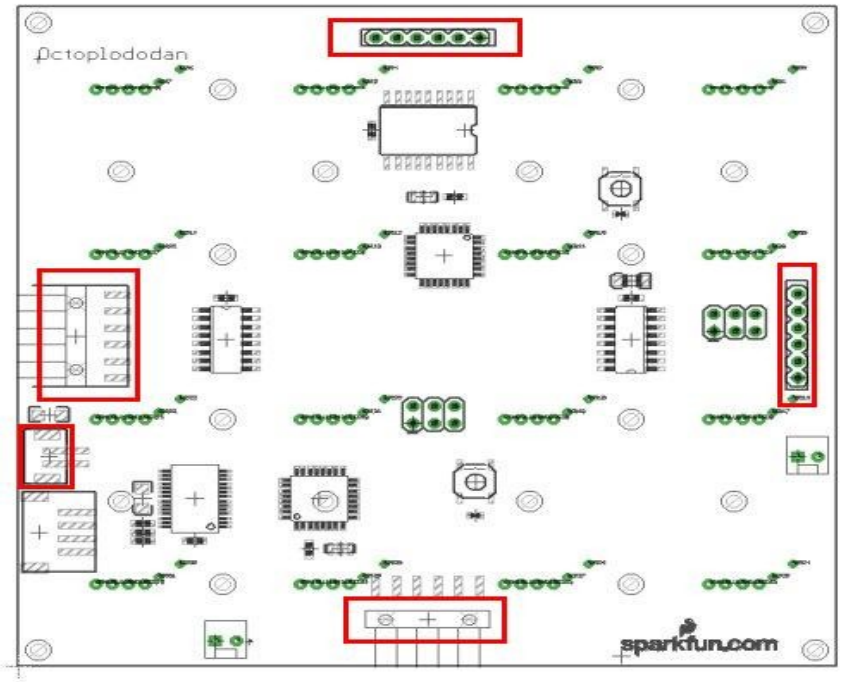
Overview

The Button Pad Controller USB was designed to control an entire multi-board system of Button Pad Controllers using an easy to use API through a USB connection. The board is 4”x4” and has 16 tri-color LEDs and 16 corresponding button pads (i.e. The button pads surround each LED). The USB Button Pad Controller is designed to communicate with up to 9 other Button Pad Controller SPI boards. The default firmware uses a 512 bit color scheme for each LED. Each board comes configured to work as a standalone unit and must be reconfigured for multiple board systems.

Features

- Runs on 5V with maximum current draw of 290 mA
- 16 Tri-Color LEDs
- 16 Button Pads (which correspond to the LED locations)
- Vertical JST Power Connector
- 2 Output connectors for controlling multiple Button Pad Controller-SPI Boards
- Serial API over USB for controlling LED colors and retrieving button status
- 6-pin programming headers for reprogramming on-board microcontrollers
- 4-pin JST connector for USB-JST cable

Figure 1: Buttonpad Controller USB PCB with power connectors highlighted



Buttonpad Controller USB User Guide

2009.01.26

Powering the Button Pad Controller USB

The Button Pad Controller must be powered with a regulated 5V supply. There is a JST connector located on the board for powering the device, however you may also use any of the four external connectors to power the device.

*Figure 2: Current Rating is for a single board. Ratings for multiple boards will be linear (i.e. If 'N' boards are connected in the system, the maximum current draw will be N*Max Current mA.). If your power supply is not capable of providing enough power for the required load, the boards may malfunction.*

Parameter	Min.	Recommended	Max	Unit
Voltage	4.50	5.00	5.25	V
Current	-	-	290	MA

There is no on-board voltage regulation so it is recommended to use a known good voltage supply, and to ensure that there won't be voltage fluctuations on the power line. Make sure to properly polarize your power connection according to the indicators on the board before turning the device on.

API

The API for the Button Pad Controller USB provides the user with an easy way to communicate with the Button Pad Controller from an external device, specifically using a USB cable from a computer. By connecting a USB cable from a computer to the Button Pad Controller USB you will be able to use a set of API commands to change the color of the entire board, a single LED or retrieve the status of the buttons. If you decide to connect more than one button pad controller to your system, you will also change the system settings telling the board how many controllers are connected using the API.

To use the API, you will need to add a USB connector to the Button Pad Controller USB. SparkFun has a JST to USBa cable that is perfect for this application, but you can build your own cable as well. The device needs to be used with a Type B USB receptacle. Located next to the power connector is a horizontal 4 pin JST connector, this connector is for the USB receptacle. If you decide to build your own USB cable to interface with the Button Pad Controller, make sure to follow the pin-out indicated on the PCB.

After plugging in the USB cable, API commands can be sent serially to the Button Pad Controller. The simplest way to do this is through a Virtual Com Port on your computer using a terminal program. Because there is a serial gateway though, writing your own programs to communicate with the board is very easy.

API Commands

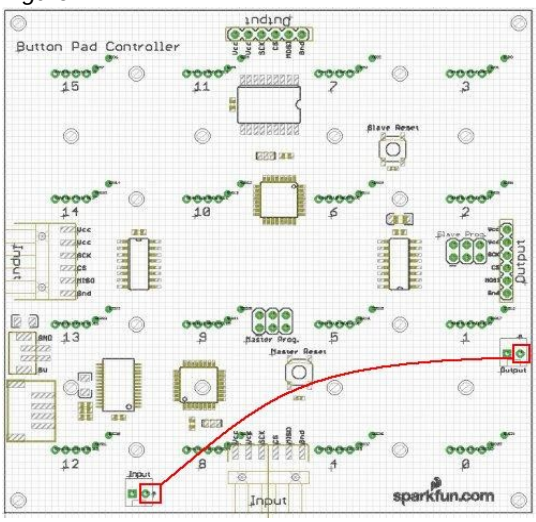
Currently there are four supported API commands on the Button Pad Controller USB. The commands can be used to set the color of the entire board, set the color of an individual LED, retrieve the status of the buttons, or change the number of boards being addressed by the master. All of the commands use the following format for the command structure:

Figure 3

Start Character($)	Command ID	Command Parameters (Variable Length)	End Character('\r')

Figure 4

Figure 4



After sending an API command to the Button Pad Controller USB, the board will reply with the status of the buttons. This pertains to all API commands except the ones that alter system parameters (i.e. The Set Number of Boards command). In order to receive the correct button status you'll need to provide a feedback wire from the last board in the system to the first board. You can read more about this in the “Connecting Multiple Boards” section below. If you are only using the Button Pad Controller USB in a 1 board configuration you'll simply need to connect a wire from the “Output” pin to the “Input” pin as indicated in the image above.

The button response is formatted as follows:

Figure 5

Start Character($)	Number of Buttons Pressed	Button Indices (Variable Length)	End Character('\r')

There are two fields of importance in the button response. The first parameter returned after the start character indicates how many buttons are being pressed. If no button are being pressed, this value will be 0. The maximum value of this field is 16 * Number of Boards in the system.

Following the “Number of Buttons Pressed” field are the button coordinates. There are two coordinates for each pressed button: the board number and the position number. The board number indicates which board the button is being pressed on, while the position number indicates which specific button on the respective board is being pressed. For more information about what the Board Number is, read the 'Set LED Color' command description below. Also, a sample button response is explained in the 'Get Buttons' command description.

Command: Set LED Color(0x00)

Board Number:
This parameter is used to indicate which board you want to address, this is generally only significant in systems where more than one board is connected. If your system only has the main board (the one the USB is plugged into), than you should set this value to 0.

Note: The addressing of the LED and the Board number is 0 based, so the first board or LED would be 0, the second

is 1, the third is 2 and so on...

However if you have connected extra boards to the Button Pad Controller USB than you will have to choose which board has the LED you want to light up. The Master board (the one the USB is plugged into) will always be the last board in the system. So for example, if there are two boards in the system the board with the USB plugged into it would be board number 1, and the other board would be board number 0.

LED Number:
The LED number dictates which LED on the specified board will be set. There are 16 LED's on the board, and they are addressed using hexadecimal values. This means that the first LED is address 0, and the 16th LED is address F. You can see how the LED's are addressed in the graphic below.

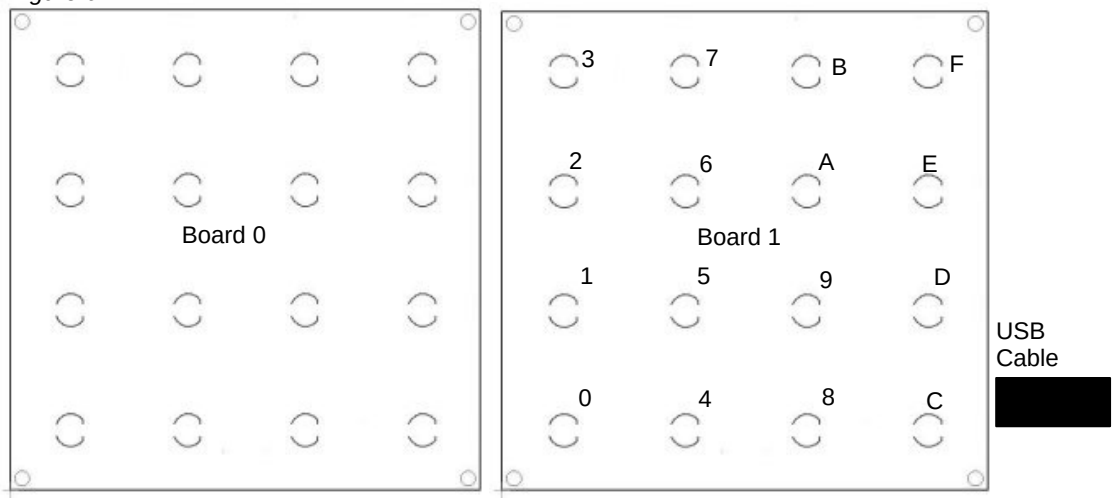
Red Value:
The Red value is a hex number (between 00 and FF) that represents the Red brightness that will be set for your corresponding LED.

Green Value:
The Green value is a hex number (between 00 and FF) that represents the Green brightness that will be set for your corresponding LED.

Blue Value:
The Red value is a hex number (between 00 and FF) that represents the Red brightness that will be set for your corresponding LED.

Mode 2 logs ADC measurements according to which are selected as active at whatever frequency is specified ("Frequency = 100" in this case).

Figure 6



Command Structure:

Figure 7

Start Character	Command ID	Command Payload					End Character
$	00	Board Number	LED Number	Red Value	Blue Value	Green Value	'\r' (Enter)

Example Command:
$00010A030508\r

Results:
The example code specifies sends the Set LED command, and addresses LED 'A' on board 1. So the tenth LED on board one would change it's color so that the red brightness would be 03, the green brightness would be 05 and the blue brightness would be 08.

Note: Remember, all of the API commands also send a response that indicates the button status of the entire system.

Command: Set Board Color(0x01)

The 'Set Board Color' command is used to set the color of the entire Button Pad Controller. There are three parameters associated with this command: red value, green value and blue value.

Red Value:
The Red value is a hex number (between 00 and FF) that represents the Red brightness that will be set for your board.

Green Value:
The Green value is a hex number (between 00 and FF) that represents the Green brightness that will be set for your board.

Blue Value:
The Red value is a hex number (between 00 and FF) that represents the Red brightness that will be set for your board.

Command Structure:

Figure 8

Start Character	Command ID	Command Payload			End Character
$	01	Red Value	Blue Value	Green Value	'\r' (Enter)

Example Command:
$01050701\r

Results:
The example code specifies sends the Set Board Command, and sets the Red brightness for the board to 05, the green to 07 and the blue to 01.

Note: Remember, all of the API commands also send a response that indicates the button status of the entire system.

Command: Get Buttons(0x02)

The 'Get Buttons' command is used to retrieve the status of the buttons without changing the color of any of the LEDs. There are no parameters for this command.

Command Structure:

Figure 9

Start Character	Command ID	Command Payload	End Character
$	02	None	'\r' (Enter)

Example Command:$02\r

Results:
$010005\r
If the 'Get Button' command was sent to the board, this would be the response assuming button number 5 on board number 0 was being pressed.

Command: Set Number of Boards(0x03)

The 'Set Number of Boards' command changes a system parameter in the microcontrollers located on the Button Pad Controller USB. The purpose of setting this parameter is to tell the board how many button pads are connected in the system. By default the Button Pad Controller USB is set up to run a single board. If you connect any extra Button Pad Controllers without changing the Number of Boards parameter you will notice that the the other API commands do not work as described. This is because the main controller does not 'know' that there are extra boards connected.

In order to change the number of boards, disconnect the extra boards from the system and then send the command.

Command Structure:

Figure 10

Start Character	Command ID	Command Payload	End Character
$	03	Number of Boards	'\r' (Enter)

Example Command:
$0302/r

Results:
The Button Pad Controller USB is now configured to run 2 boards instead of 1. This command will not return the status of the buttons!

Note: If you connect multiple boards to the system, each individual board must be configured for the proper number

of boards. Read the section “Connecting Multiple Boards” for more information on this subject.

Connecting Multiple Boards

If you've purchased additional Button Pad Controller SPI boards along with your Button Pad Controller USB board, you can connect them to create a larger button pad system. Up to 9 additional Button Pad Controller SPI boards can be connected for a total of 10 button pads in the system. There can only be 1 Button Pad Controller USB in the system, and it must always be the first board (meaning the USB board only outputs data, there can't be any boards connected to the 'input' of the USB board); all the other boards must be Button Pad Controller SPI boards. There are several precautions that must be made in order for this type of system to work though.

Each board, whether it is USB or SPI, comes configured to work in a 1 board system. To get multiple boards to work together in one system, each board must be reconfigured to work in an 'N' board system (where 'N' is the total number of boards that will be connected in the system). To reconfigure the USB board, simply use the 'Set Number of Boards' API command, however you must ensure that no additional boards are connected to the USB board while sending the command. For instructions on reconfiguring the SPI boards refer to the user guide for the Button Pad Controller SPI.

Once all of your boards have been configured properly you'll just need to connect them and provide a feedback wire for the button status. It should be fairly obvious how the button pads need to be connected, but we'll go over them just in case. On the edge of each board you'll notice a footprint for an external connector. Two of these footprints are designated as outputs, while the other two are designated as inputs. A board can only output data to one other board, so you can not connect a board to each 'output' connector.

Once the boards are connected, you'll need to provide a feedback wire from the last board connected in the system, back to the Button Pad Controller USB. Near one of the 'output' sides of the board, you'll see a 2pin header that reads “Output,” you'll also see a 2pin header near an input connector that reads “Input.” Along with the header, you'll see a small dot on the silkscreen near one of the pins, this is the feedback pin. You'll need to provide a wire from the “Output” pin on the last board (Button Pad Controller SPI) to the “Input” pin on the first board(Button Pad Controller USB). If you don't care about getting the button status from the system, you may leave the feedback wire off.

When connecting additional boards to a system, be cautious of how you are affecting the overall current consumption of the system. With each additional board the maximum potential current consumption increases by 290 mA; if your power supply is not capable of providing enough power the system will not operate properly.

For further explanation, please reference Figure 11:

Figure 11