sparkfun ELECTRONICS

WEBSITE: sparkfun.com
6175 LONGBOW DRIVE, SUITE 200
BOULDER, COLORADO USA
ZIP CODE: 80301
[303] 284.0979 [GENERAL] P
443.0048 F

Buttonpad Controller SPI User Guide

2009.02.26

Overview

The Button Pad Controller SPI was designed to compliment the Button Pad Controller USB; though it can also be used as a standalone device. The board is 4 x4   and has 16 tri-color LEDs and 16 corresponding button pads (i.e. The button pad surrounds the LED). The boards communicate via an SPI bus, and up to 10 Button Pad Controllers can be connected in a system (including one Button Pad Controller USB). The default firmware uses a 24 bit color scheme. 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 (Capable of up to 24 bit color)
- 16 Button Pads (which correspond to the LED locations)
- 2 Output headers for controlling multiple Button Pad Controller-SPI Boards (includes power bus)
- 2 Input headers for receiving commands from other Button Pad Controllers (includes power bus)
- 6-pin programming header for reprogramming on-board microcontroller
- Microcontroller runs at 20 MHz

Powering the Button Pad Controller USB

The Button Pad Controller must be powered with a regulated 5V supply. The board can be powered from any of the 6 pin headers located on the sides of the board (either the input or output headers). Be sure to connect your 5V supply to the VCC pin of the header, and the ground of your supply to the GND pin. There are two pins on each header for VCC, only one needs to be connected to provide power to the board.

*Figure 1: *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.

Using the Button Pad Controller SPI

This section will discuss how to send and receive data from the Button Pad Controller SPI. If you're using a multi-board configuration and utilizing a Button Pad Controller USB board for control, you can skip this section as the USB board takes care of the communication. This section only concerns those who are using their own board to control the Button Pad Controller SPI.

After providing power to the Button Pad Controller SPI, the device will go through it's power up sequence. Essentially the power up sequence runs a test to make sure all of the LEDs are outputting correctly, and to notify the user of what configuration the device is in. Once powered, the board will display all red for about 300 ms, then all green and finally all blue. After cycling through all three colors, a single LED will light up as white. The lit LED corresponds to the the current board configuration of the unit. So, if the '0' LED is white the board is configured as a standalone system (one board only, no externally connected boards). If the '1' LED is white the board is configured for a two board system. We'll go over how to configure the board for multiple documents later in the user guide.

After the Button Pad Controller SPI has gone through the power up sequence it is ready to receive data. The board receives data on it's SPI bus. The Button Pad Controller SPI devices are each slaves, however their data lines are switched. Since the board is powered by 5V, the signals should also be sent with 5V logic levels.

Figure 2

Signal	Function
SCK	Input (Max 50 KHz)
CS	Input
MISO	Input
MOSI	Output

In order to successfully communicate with the Button Pad Controller SPI a complete 'set' of data must be sent. A 'set' includes 128 * N bits of Red data, 128 * N bits of Green data, 128 * N bits of Blue data, and 128 * N extra clock cycles to allow the board to transfer it's current button status. Each subset of data in the set is considered a 'Frame' of data. So the data set consists of a Red Frame, a Green Frame, a Blue Frame and a Button Frame. 'N' is the number of boards in the system, so in the standalone configuration N is one. In a standalone configuration, it takes 512 clock cycles to complete a data set. In a multiple board configuration, the system works by passing along all of the data it receives until it realizes that the correct amount of information has been passed and it can display it's own data.

Writing to the Button Pad Controller SPI

The first three frames of data are the color frames. The color frames contain the desired shade of color to be set for each LED. A color frame consists of 8 bits of color for each LED, which totals 128 bits for each frame since there are 16 LEDs. The frame starts with the color data for the first LED, or LED 0, and ends with the last LED. The color byte should be sent out with the least significant bit first.

Figure 3

Red Frame			Green Frame			Blue Frame			Button Frame		
Led[0]	...	Led[15]	Led[0]	...	Led[15]	Led[0]	...	Led[15]	Button [0]	...	Button [15]

In order to pass the set of data, an external device must be used to control the SCK, CS and MISO signals. The Button Pad Controller SPI will begin listening for a data set once the CS input has been brought high. The CS signal should remain high for the duration of the data set, at which point it should be brought low for a minimum of 400 µs before sending the next set of data. SCK is an active low signal, so initially the clock should be set high (HINT: It's best if the SCK signal is set high before setting the CS pin high). The data on the MISO line should be set while the clock is low. When the SCK signal goes high the Button Pad Controller locks in the data currently on the MISO line. In order to provide for the lowest bit-error we recommend you don't exceed and SCK frequency of 50 Khz, though the system may still work beyond this frequency.

Buttonpad Controller SPI User Guide

2009.02.26

Figure 4

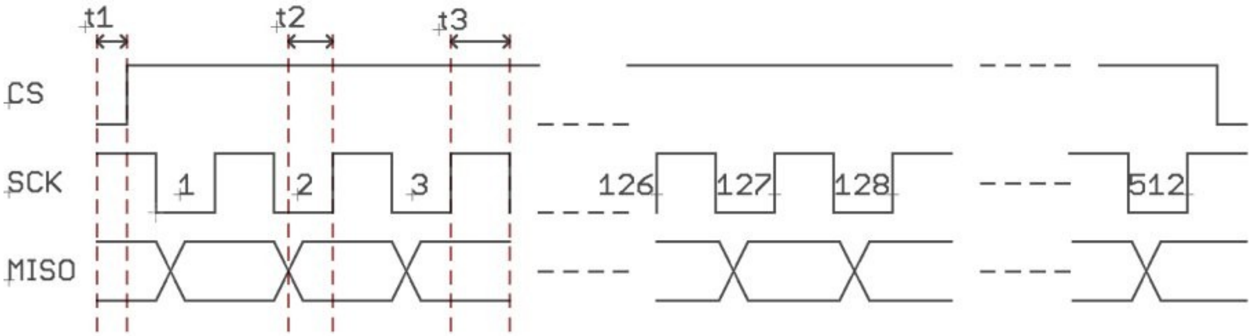


Figure 4 shows how the external device should control the signals to send a set of data (remember, a data set consists of 4 frames of data). Notice how the data on the MISO line is only changed while the SCK is low. This is also true for the MOSI line; which means if you are reading data from the Button Pad Controller SPI you should do so only while the SCK line is high. The data will not be valid while the SCK line is low.

Figure 5

	Time	Function
t1	10us	Chip select to clock start
t2	5us	Data setup time
t3	10us	Data latch time

Reading from the Button Pad Controller SPI

After the three color frames are sent, an extra frame must be sent for the button values. It doesn't matter what data is sent on the MISO line in this frame; it's only important that the Button Pad Controller receives 128 more clock cycles on the SCK line. The extra clock cycles allow the Button Pad Controller to shift out the status of the current buttons onto the MOSI line. The data on the MOSI line should be read while the SCK signal is high; when the SCK line is low the data is changing and is therefor invalid.

The MOSI line will shift out as much data as is shifted in; however in a standalone system the first three frames of data output on the MOSI line are meaningless. The last frame, the button frame, is important though. The last 16 bytes of data shifted out contain the status of the buttons on the Button Pad Controller. Each data byte in the Button Frame represents one button.

The first byte in the Button Frame corresponds to button zero, while the last byte in the frame corresponds to button 16. The value of each byte will either by 0x00 or 0xFF. If the value is 0x00, then the corresponding button is being pressed; if the value is 0xFF then the button is *not* being pressed.

After the Button Pad Controller SPI receives an entire data set, the board will display the color designated by the color frames. Then the board is ready for another set of data to be sent again.

Buttonpad Controller SPI User Guide

2009.02.26

Using the Button Pad Controller in a Multi-Board Configuration

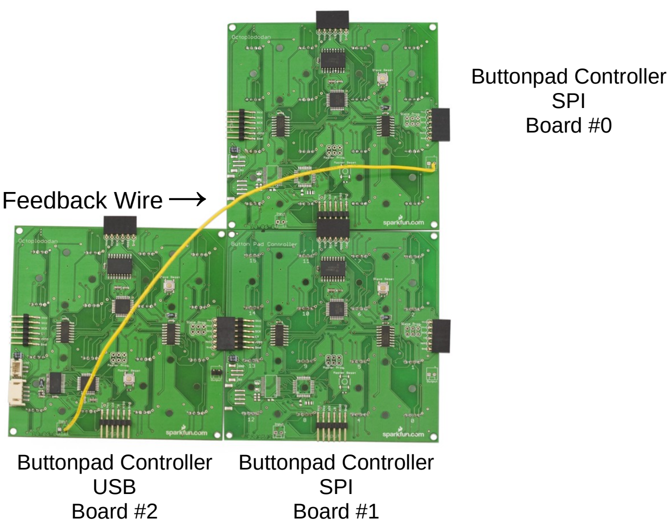
Up to 10 Button Pad Controllers can be connected to a single system. While it is recommended to use 1 USB Button Pad Controller in the system to control the color output and button inputs, you can also control the entire system on your own by sending commands to an SPI Button Pad Controller. If you're going to use a multi-board system, however, *each board* in the system must be re-configured to reflect the total number of boards in the system.

Setting up the Hardware for a Multi-Board System

One important limitation in a multi-board system is that each Button Pad Controller can only output to one other Button Pad Controller, and it can only receive data from one Button Pad Controller. Each board has a 6 pin header on each of the four sides of the board; two of these headers are designated as outputs while the other two are designated as input. Make sure you only connect inputs to outputs and vice versa; connecting two inputs or outputs together may damage the microcontroller. The headers have been arranged to allow for different configurations of boards to be connected; don't think you have to connect the boards in any specific shape.

The illustration below shows one possible configuration of a three board system. In this system there is a USB Button Pad Controller followed by two SPI Button Pad Controllers. Any time you are using a USB Button Pad Controller in a system, it must be the first board in the configuration. This means that it is only used as an output, and no board are connected to the input header of the USB board.

Figure 6



The red line illustrates the feedback wire. This must be added by the user, and is required for button status feedback. To add this wire to your system, simply solder a wire from the OUTPUT 2 pin header of the last board in your system, to the INPUT 2 pin header of the first board in your system. The last board of your system is the one that receives an input from another board, but doesn't send an output; while the first board in the system is that one that outputs data to another board but doesn't receive an input. If a feedback wire is not connected to the system you will not receive the button status of the system.

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.

Buttonpad Controller SPI User Guide

2009.02.26

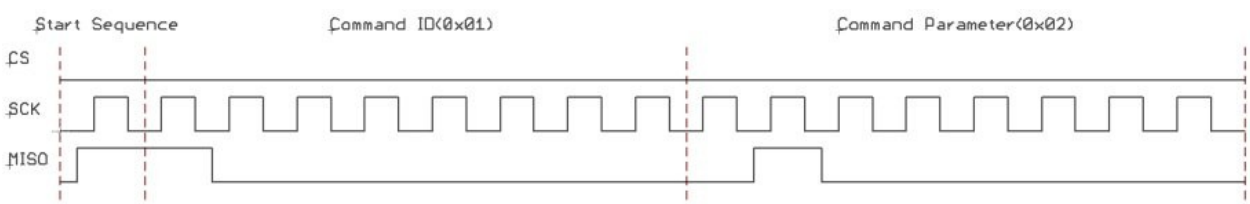
Configuring the Firmware for a Multi-Board Configuration

As has been mentioned before, if you are using your Button Pad Controllers in a multi-board configuration each board must be re-configured to operate in the new system. For a USB Button Pad Controller the configuration can be changed via an API command through the USB cable; however things are a little more difficult for the SPI Button Pad Controller.

To reconfigure the SPI board for multi-board systems a special sequence must be sent over the SPI bus. The sequence consists of three parts. The first part tells the Button Pad Controller to enter command mode, the second part indicates what parameter is going to be changed (Command ID), and the last part of the sequence is the value for the parameter(Command Parameter). Currently the only parameter that can be changed in command mode is the 'Number of Boards' parameter which tells the device how many boards will be in the system, though more system parameters may be added later. The Command ID for changing the 'Number of Boards' parameter is 0x01. The command parameter for this command can be anything between 0x01 and 0x0A.

In order to enter command mode in the Button Pad Controller SPI the following signals must be sent to the controller: while keeping the CS line held low bring the MISO line from low to high. Once you've brought MISO high, pulse the SCK line by bringing it high and then low again. After this series is sent, the Button Pad Controller is in command mode. Once command mode is entered the device waits for a Command ID(one byte) and a Command Parameter (one byte). The board will automatically exit command mode after these two bytes have been received. Figure 6 shows how you would configure a Button Pad Controller SPI for a two board system.

Figure 7



Communicating with a Multi-Board System

Once you've configured each board in your system for the proper number of boards you can go ahead and connect the system together in your desired arrangement. After providing power to the system, each board should go through it's power up sequence. After the power up sequence has been completed, you should double check that the boards have been configured correctly by verifying the position of the white LED. As was mentioned earlier, the location of the white LED indicates the configuration of the Button Pad Controller; in our two board system LED 1 should be white on both of the boards.

After the system has gone through the power up sequence the system is ready for data. If you understand how to send and receive data on a standalone system it's not too hard to jump to the Multi-Board communication. In order to a picture to a standalone Button Pad Controller the board must receive a complete data set. Likewise in order to send a picture to a multi-board system, the system must receive 'N' complete data sets, where 'N' is the number of boards in the system. Also remember that each data set must be separated with a CS pulse low for 400 µs (see Figure 9). Since one data set consists of four separate frames and each frame has 64 bytes, a system would need 'N' * 512 clock cycles and corresponding data bits to produce an image on the system. The system will not update it's display until it receives enough data for the entire system; so once you start sending data you shouldn't stop until you've sent a complete image.

The order of the data for a multi-board system is the same as for a standalone system. When sending an image to a

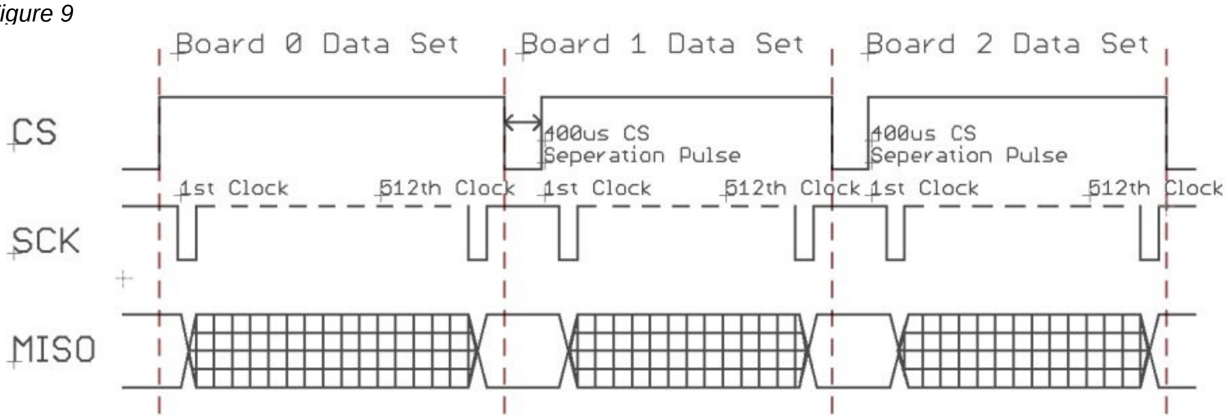
WEBSITE: sparkfun.com
6175 LONGBOW DRIVE, SUITE 200
BOULDER. COLORADO USA
ZIP CODE: 80301
[303] 284.0979 [GENERAL] P
443.0048 F

system you'll send the data one board at a time, starting with the data for the last board in the system (Board 0). For instance, lets assume the multi-board system demonstrated in the Setting up the Hardware for a Multi-Board System section of this document. You would start by sending four frames of data for Board 0 (Red Frame, Green Frame, Blue Frame, Button Frame), then you would send the data set for Board 1, and finally finish with the data for Board 2. Figure 9 shows how an external device would send a complete data set for a 3 board system.

Figure 8

Board 0				Board 1				Board 2			
Red Frame	Green Frame	Blue Frame	Button Frame	Red Frame	Green Frame	Blue Frame	Button Frame	Red Frame	Green Frame	Blue Frame	Button Frame

Figure 9



Receiving data from a multi-board system is also very similar to receiving data from a standalone system. Like in a standalone system, the Button Pad Controllers in a multi-board system will each output as much data as is sent to them ('N' * 512 bits or N data sets). The output data from the system is sent along the feedback wire. If a Button Pad Controller USB is being used to control the system, the feedback wire should be sent to the USB board; however if you're using your own board to control the system you should use the feedback wire as an input to your controller.

The feedback will come in data sets that are identical to the input data sets, and remember that the data gets put onto the feedback wire while you are sending data to the system. The first three frames or each data set on the feedback wire will be the color frames. The data in the color frames will reflect the image currently on the system (not the image currently being sent to the system). After the color frames comes the Button Frame. The button frame will reflect the status of the button being pressed on each board in the system. Receiving a byte in the button frame with a value 0x00 means a button is being pressed while a value of 0xFF means the button is not being pressed. Read the section Reading from the Button Pad Controller SPI to learn about how the button are represented in the frame. The feedback data sets will come in the same order that they are sent. The first data set will be for Board 0, then Board 1 all the way up to Board (N-1).