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

RGB Matrix – Serial Backpack User Guide

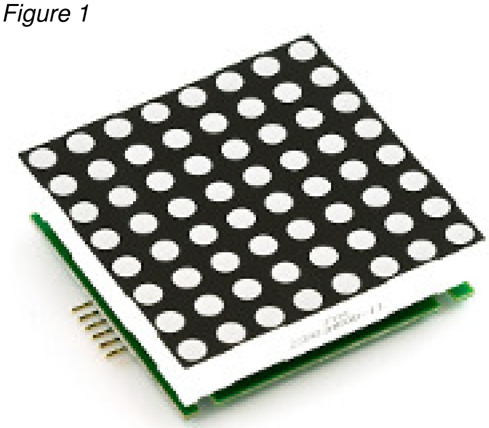
2009.04.20

Overview

The new RGB Matrix – Serial Backpack controller from SparkFun offers an easy way to control the tri-color common cathode 8x8 LED matrices. The entire controller fits behind the LED matrix so that even larger arrays of matrices may be created; external connectors are oriented to allow daisy-chaining of multiple controller boards. The boards accept a special software command to be re-configured for daisy-chained systems. Each backpack is based around an AVR microcontroller and contains all the necessary circuitry to drive the LED matrix.

By default the controller runs a simple frame buffer program that listens for image data and displays in constantly on the LEDs. Data is sent to the backpack using an SPI interface. The microcontroller may also be reprogrammed to give the LED matrix more individual intelligence.

Figure 1



Features

- Runs on 5V
- 64 Tri-Color LEDs
- Input and Output Connectors for daisy-chaining multiple matrices
- 6-pin programming headers for reprogramming the AVR

Powering the RGB Matrix

The controller must be powered with a regulated 5V supply. Power can be supplied either at the SPI Input connector or the SPI output connector as they each share the same power bus. If you are daisy-chaining multiple controllers it is recommended to supply power at the Input connector of the first board in your system, and use the Output connector to supply power to the next board in the system.

RGB Matrix – Serial Backpack User Guide

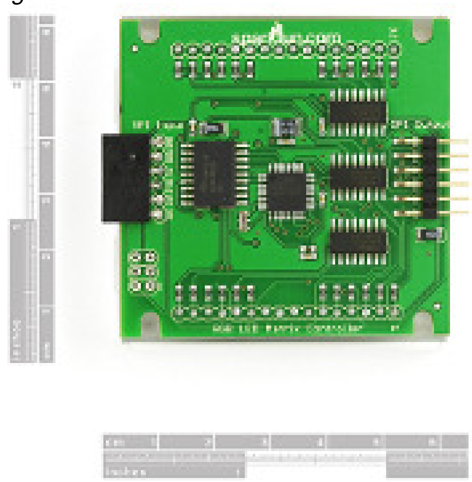
2009.04.20

Figure 2

Parameter	Min.	Recommended	Max	Unit
Voltage	4.50	5.00	5.25	V
Current	-	120(typical)	275	mA

**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.*

Figure 3



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.

SPI Interface to the Default Firmware

The RGB matrix backpacks' default program communicates via standard SPI protocol. Data in to the device must be provided via the MOSI pin. SPI Clock must be provided vial the SCLK pin. The device will return data on the MISO pin. All input is ignored while the CS pin is high (5V); and data is copied into the frame buffer while the CS pin is low (0V).

The device maintains a single 64 byte buffer which represents each position in the matrix. When CS is asserted (low) the device begins reading data from the SPI input and writing it sequentially to the 64 byte buffer. Simultaneously the device will output the old buffer data on the MISO line. Hence, to display an image on the matrix a set of 64 bytes must be sequentially transferred to the backpack while keeping the CS pin low (this process is slightly different for a daisy-chained system).

By default, the backpack recognizes up to 255 individual colors. The 64 bytes transferred to the backpack represent the desired color of each LED. The first 3 bits of each byte represent the Red brightness level for that LED; the second 3 bits represent the Green brightness level while the last 2 bits represent the Blue brightness level. Below is a table which illustrates how to construct your color value.

RGB Matrix – Serial Backpack User Guide

2009.04.20

Figure 4

Buffer Byte Representing and LED Color Value							
Red			Green			Blue	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 5

Color(Brightest Setting)	Byte Value
Black	0x00
Red	0xE0
Green	0x1C
Blue	0x03
Orange	0xFC
Magenta	0xE3
Teal	0x1F
White	0xFF

**These values correspond to the brightest levels of the given color. Different variations may be achieved by changing the brightness level of each individual color.*

Example of Sending a Frame of Data

To have an RGB matrix display “black, red, green blue” on the first four positions of the first row, and black everywhere else, this would be the process:

1.) Assert CS (bring it low)

2.) Delay 0.5ms

3.) Transfer 0x00 via SPI

4.) Transfer 0xE0 via SPI

5.) Transfer 0x1C via SPI

6.) Transfer 0x03 via SPI

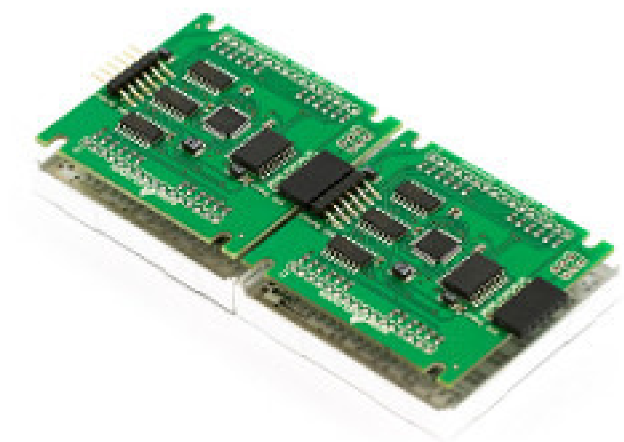
7.) Transfer 0x00 60 times via SPI

8.) De-Assert CS (bring it high)

Daisy-chaining RGB Matrix Controllers

The newest version of the RGB matrix controller has a new layout and a new version of firmware that allow the user to more easily daisy-chain multiple RGB matrix controllers. Daisy-chaining is when multiple controllers are connected together, and the data is passed from the first controller in the "chain" to the next connected controller; data is passed along until the data reaches the end of the chain.

Figure 6



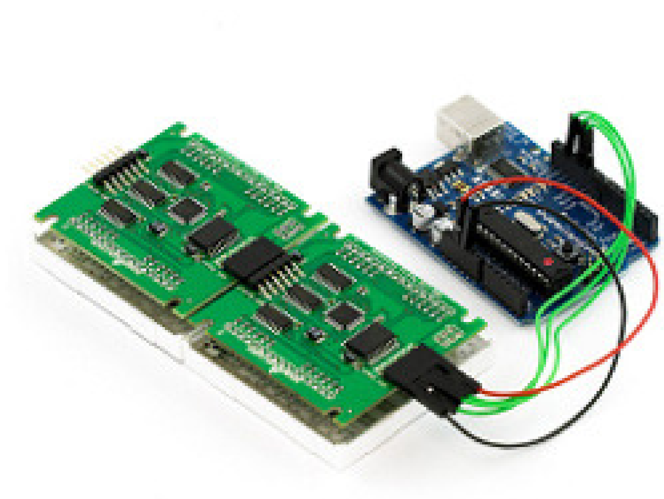
In order to daisy-chain the controllers each controller must be reconfigured for the size of the desired chain. By default the controllers come configured to operate in standalone mode, or a 1 controller chain. If there are two controllers to be daisy-chained together, each of them must be reconfigured for a 2 controller chain. To reconfigure the boards a special two byte command sequence must be sent via the SPI interface: the character '%' followed by the decimal number of the desired chain length. The controllers must be reconfigured individually (so they can't already be connected in the chain). The commands should be sent using the normal SPI protocol. Be sure to assert CS (low) before sending the two command bytes, then de-assert CS (high) when you are finished. Do not start sending the color values to the chain before de-asserting the CS pin. Also, it is recommended to limit the daisy-chain lengths to 8 boards.

When you are connecting the boards, make sure to connect the output of the 1st board in the system to the input of the 2nd board in the system. The connections from your Host system should be plugged into the Input connector of the first controller in the system. Power will be passed through the connectors to all the boards in the system. However, be sure that the power supply can provide enough current for the entire system.

Interfacing to a Daisy-chained System

If the controller boards have been reconfigured to operate in a daisy-chain there are several differences that need to be addressed to display an image on the system. Obviously now that there are more controllers connected there needs to be more data sent. Luckily sending the data to a daisy-chained system is very similar to sending data to a standalone controller. To display an image on a standalone controller all that's done is: assert CS(low), send a 64 byte buffer (frame), de-assert CS. To display an image on a daisy-chained system, this exact process must be repeated for each board in the system. The image won't be displayed until data is received for each controller in the daisy-chained system. A 10ms delay is recommended between de-asserting the CS pin and asserting it before the next frame. The first frame sent to the daisy-chained system will be displayed on the last board in the system, while the last frame sent to the system will be displayed on the first board in the system.

Figure 7



SPI Timing Recommendations:

A delay of 0.5ms is recommended between the assertion of CS and at the start of data transfer, as well as after the end of data transfer and the negation of CS. A 10ms delay is recommended between the CS pulses in a daisy-chained system. The SPI clock should not exceed 125 kHz.