Adafruit 16x32 LED Guide for BeagleBone

By Kai Chu, Chang Yun, Abdinajib Idle, Fatemeh Kianne

Preface

This is a guide for CMPT 433 students on using the Adafruit 16X32 LED matrix.

For context, components were used on a BeagleBone Red, but will also work for Green and Black Beaglebones as the GPIO pin layouts are the same.

Setting up the LED matrix

The original RGB Matrix guide can be found here and my guide will heavily reference its work.

There are two main components to the LED matrix, the power and the connector pins.

For the connector pins, ensure the power is turned off before connecting.

There are two sockets on the back of the RGB matrix, labeled INPUT and OUTPUT.
For our purposes, we will only be handling with the INPUT socket.

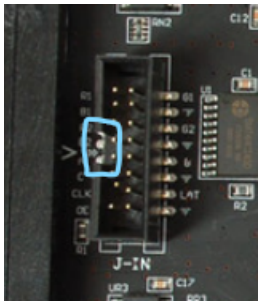


Figure 1

To match the pins correctly, ensure the notch is facing from the left-hand side as shown above.

If your RGB Matrix panel came with Jumper wires, the corresponding pin outputs might be oriented differently due to the wires. You can see a flipped configuration example in this previous guide.

Socket Information

There are multiple different layouts for different panel sizes, but for a 16x32 panel, it will use the corresponding layout. Each pin provides some sort of feedback and is listed in the table below.

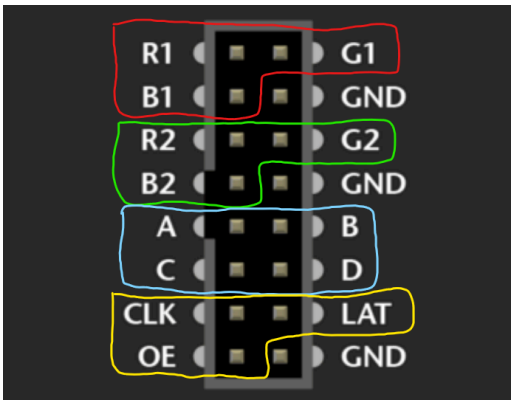


Figure 2

LABEL	PURPOSE
R1 (Red 1)	Handles R data for the top half of the display.
G1 (Green 1)	Handles G data for the top half of the display
B1(Blue 1)	Handles B data for the top half of the display
GND (Ground)	Connect the pin to the ground
R2 (Red 2)	Handles R data for the bottom half of the display.
G2 (Green 2)	Handles G data for the bottom half of the display.
B2 (Blue 2)	Handles B data for the bottom half of the display.
A	Selects which two rows of the display are currently lit
B	Selects which two rows of the display are currently lit
C	Selects which two rows of the display are currently lit
D	Not used in 32x16 Matrix panels
CLK	Arrival of each bit of data
LAT	Marks the end of a row of data
OE	Switches LED off when transitioning from one row to the next.

You will need to connect every pin to your beagle bone. Depending on your available GPIO pins, you can connect this however you want, but you must match **GND pins to DGND** and **every other pin must connect to a GPIO number.**

For simplicity and consistency, it is recommended you connect them with the same GPIO pins following this guide.

The way most students have set this up in the past is by having all 4 ground pins be connected within P9 and all GPIO pins connected within P8.

*Note that pin D is connected to the ground because it is not used for 32x16 matrix panels.

Detailed connections can be shown here with GPIO numbers listed for corresponding Beaglebone pins.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

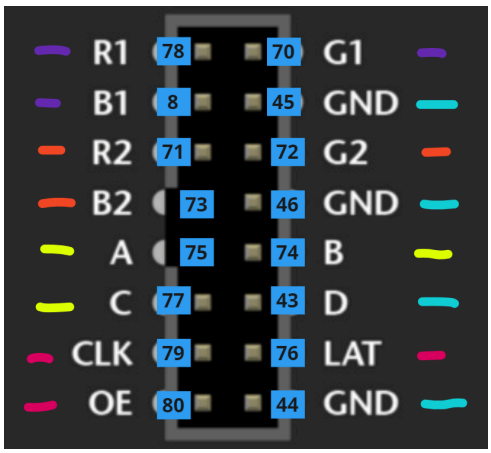


Figure 3

Final connections should look like this,

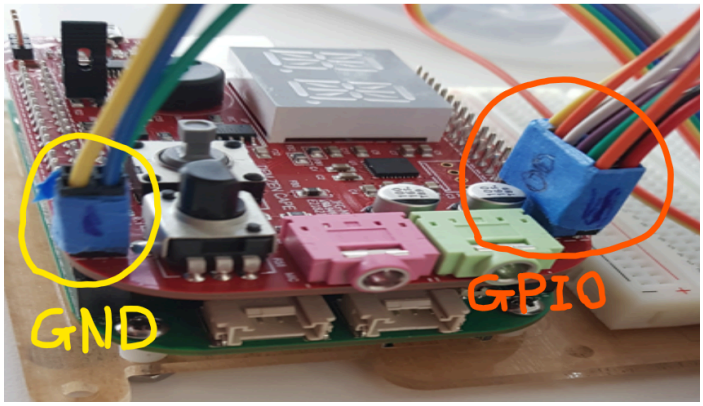


Figure 4

For the power, RGB Matrix should be using Molex-style headers which have connections that are pre-connected with a power plug. All you need to do is connect matching pins (Usually already done for you) and plug power into an electrical outlet.

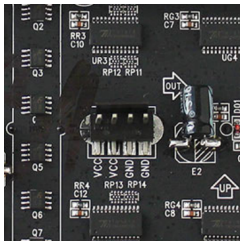


Figure 5

Booting

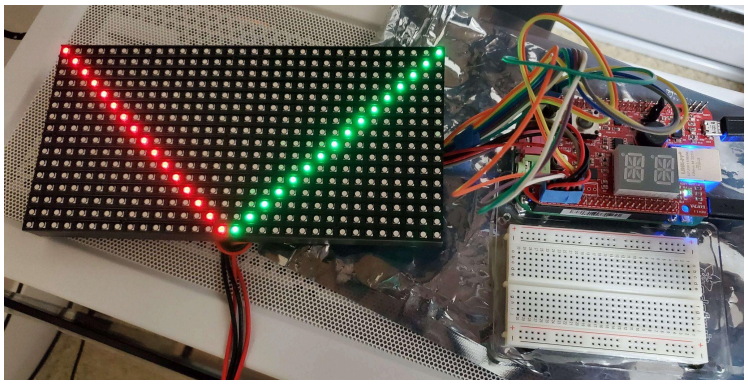
The order for starting up the Beaglebone is as follows,

1. Connect LED panel pins to the Beaglebone
2. Connect Beaglebone to the host device *(See troubleshooting guide for issues on this step)*
3. Power LED panel through power plug.

Testing

To test that the wires are properly connected, you can use the example code for setting up the LED matrix panel.
https://github.com/Montreal/BeagleBone-Green-Adafruit-16x32-LED-Matrix-Sample-Code/blob/master/test_ledMatrix.c

However, if you have been following this guide, you will need to reconfigure the #defined GPIO pin numbers for the **test_ledMatrix.c** file to match the new layout.



```
17  // GPIO PIN DEFINITION
18  #define RED_PIN_1 "78"
19  #define GREEN_PIN_1 "70"
20  #define BLUE_PIN_1 "8"
21
22  #define RED_PIN_2 "71"
23  #define GREEN_PIN_2 "72"
24  #define BLUE_PIN_2 "73"
25
26  #define A_PIN "75"
27  #define B_PIN "74"
28  #define C_PIN "77"
29
30  #define CLOCK_PIN "79" //Ar
31  #define LATCH_PIN "76" //Ma
32  #define OE_PIN "80"    //Sw
```

Your board should display a V shape if built correctly.

The provided test code will give you a very convenient way to update the LED panel.

It will store color values in a 16 by 32 matrix which represents each LED on the panel.

The available colors to change range from

Value	**Color**
0	None
1	Red
2	Green
3	Yellow
4	Blue
5	Purple
6	Light blue
7	Even lighter blue

Each LED can be set through the **ledMatrix_setPixel()** method.

Your program should also include a separate thread that solely runs the **ledMatrix_refresh()** method. See the troubleshooting section for more information.

Troubleshooting

Can't turn on Beaglebone with an LED panel.
If using a Beaglebone red, you may have issues turning on the Beaglebone while the pins are connected to the LED panel. If this is the case, perform booting with this sequence instead.

1. Connect Beaglebone to the host device
2. Connect LED panel pins to the Beaglebone
3. Power LED panel through power plug.

LED lights start flickering randomly
For the LED panel to display the lights, it needs to be constantly refreshing the information being displayed. If you are using the example code, this is done through the **ledMatrix_refresh()** method.

This command must be within its own separate thread and only executing the command to update the LED panel.

```
while(1) {
    ledMatrix_refresh();
}
```

If you are running too many threads, and this approach does not resolve the flickering issues, A harder solution is to run the LED matrix on the Beaglebones PRU.

I followed the steps correctly but the LED lights seem to be off from the example
Sometimes if the wires are not properly connected, they can distort the resulting display on the panel. Ensure wires are firmly connected and separated for consistent results.

References

Figures 1, 2, 3, and 5:
Example diagrams taken from the Adafruit guide, some slightly modified with drawings.

Figure 4:
An example image is taken from the student guide (pg. 5), *16x32LEDPanel Guide*