Adafruit 16x32 LED Guide For BeagleBone

By: Jasper Wong

<u>Introduction</u>

This guide is written for students in CMPT 433 to help with wiring and executing the code to drive the Adafruit 16x32 LED matrix. Included with this guide is sample code for the LED matrix written by Janet Mardjuki and updated by Jasper Wong to be compatible with the BeagleBoard.org Debian Bullseye Minimal Image 2022-11-01 on the BeagleBone Green. Although components were tested on a BeagleBone Green, this guide should be compatible for running the LED Matrix on a BeagleBone Black as the pin outs are identical.

<u>Wiring</u>

Before wiring the LED Matrix to the BeagleBone, make sure the system is turned off and unplugged.

Included below is the orientation of the board with the orientation of the pins:

G1	R1
GND	B1
G2	R2
GND	B2
B	A
D	C
LAT	CLK
GND	OE

The pins correspond with the holes on the other side of the ribbon cable facing the camera.

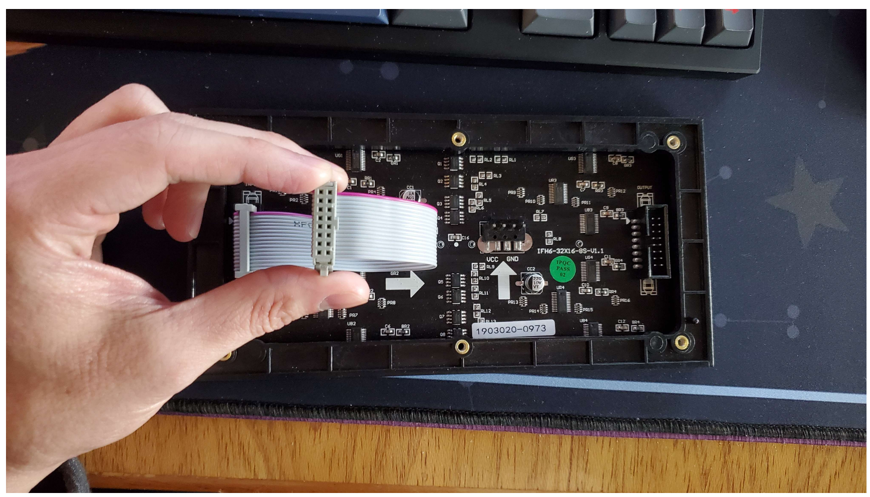

Figure 1: LED Matrix Pin Out Configuration

The arrows serve to help with orientation, with one facing right and the other facing up. There will be two ports where you can plug a provided ribbon, but only the port on the right is for input. Below is a table for where the pins will be wired to on the BeagleBone (see Figure 2):

These will be the last 12 pins on the corner of **P8**	R1 [GPIO_8] – Pin 35	G1 [GPIO_80] – Pin 36
	B1 [GPIO_78] – Pin 37	G2 [GPIO_79] – Pin 38
	R2 [GPIO_76] – Pin 39	B [GPIO_77] – Pin 40
	B2 [GPIO_74] – Pin 41	LAT [GPIO_75] – Pin 42
	A [GPIO_72] – Pin 43	CLK [GPIO_73] – Pin 44
	C [GPIO_70] – Pin 45	OE [GPIO_71] – Pin 46
These can be connected to any of the ground pins of **P9**	GND - Pin 43 (P9)	GND - Pin 44 (P9)
	GND - Pin 45 (P9)	D - Pin 46 (P9)

Where those pins are mapped to GPIO on Linux (same mapping as in the GPIO Guide):

65 possible digital I/Os

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 2: BeagleBone Pin Functionality Table

Functionality of the pins themselves is explained in the table below:

Label	Signal	Description
R1	Red1	Red colour signal for the top half of the matrix (x-index:0-7)
B1	Blue1	Blue colour signal for the top half of the matrix (x-index:0-7)
G1	Green1	Green colour signal for the top half of the matrix (x-index:0-7)
R2	Red2	Red colour signal for the top half of the matrix (x-index:8-15)
B2	Blue2	Blue colour signal for the top half of the matrix (x-index:8-15)
G2	Green2	Green colour signal for the top half of the matrix (x-index:8-15)
A	Row A	MSB of the row bits for the row selection
B	Row B	Middle bit of the row bits for the row selection
C	Row C	LSB of the row bits (for the 16x32 Matrix) for the row selection
D	Row D	LSB of the row bits (for anything bigger than 16x32 Matrix) for the row selection
CLK	Clock	Ends of each bit of the data (each pixel)
LAT	Latch	Ends of row of data
OE	Output Enable	LED on/off (enable/disable), for row transition
GND	Ground	Signal to ground

Figure 3: Matrix Pin Functionality Table and Explanation

Included in the kit is a power cable terminating in Molex (connected through a barrel plug), which is needed to supply sufficient power to the LED Matrix. There will be two Molex connectors, where one can be left dangling and not connected. Wiring is as shown below:

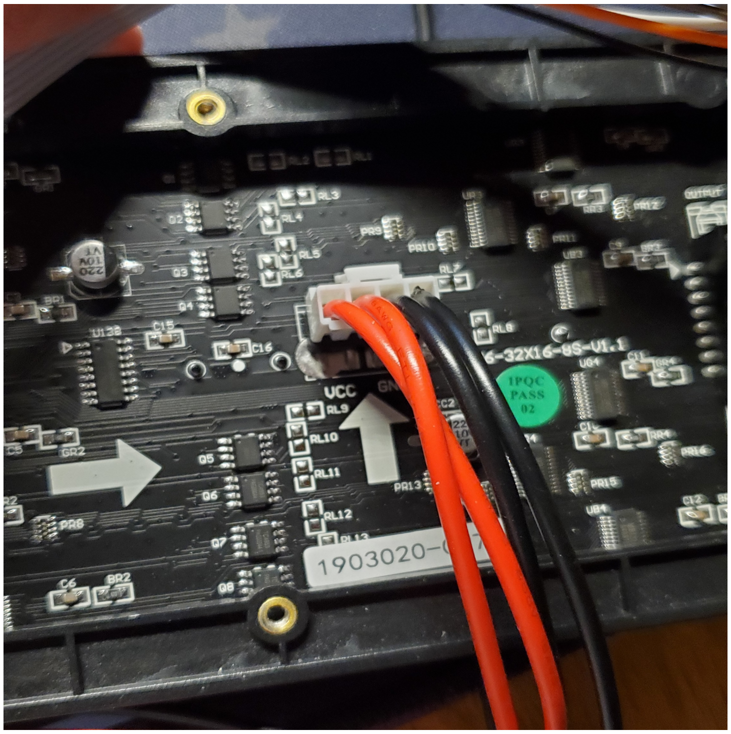

Figure 4: Power Cable to the LED Matrix

Wiring of jumper cables can be done either directly to the BeagleBone or through the Zen Cape pass-through pins.

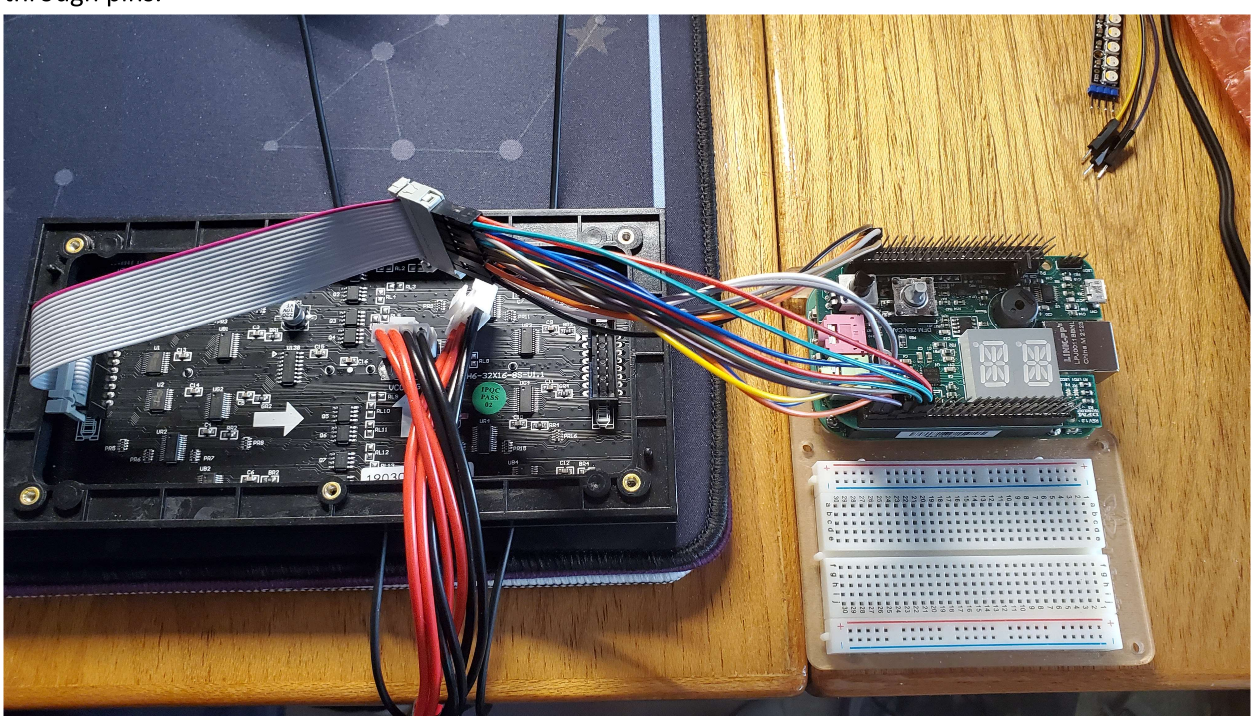

Figure 5: Final Wiring Configuration

Sample Code

Included with this guide is updated sample code from previous guides. In the case that the updated sample code is unavailable, a previous version of the code can be found at https://github.com/MontrealI/BeagleBone-Green-Adafruit-16x32-LED-Matrix-Sample-Code/blob/master/test_ledMatrix.c. The export process of pins seems to have changed with the newer Debian image used in the course, and so the code will segmentation fault without displaying any lights. To fix, replace the `exportAndOut` function (starting on line 70) from the code linked above with the code in the following page:

```
// Taken from page 7 of assignment 1 description
static void sleepForMs(long long delayInMs)
{
    const long long NS_PER_MS = 1000 * 1000;
    const long long NS_PER_SECOND = 1000000000;
    long long delayNs = delayInMs * NS_PER_MS;
    int seconds = delayNs / NS_PER_SECOND;
    int nanoseconds = delayNs % NS_PER_SECOND;
    struct timespec reqDelay = {seconds, nanoseconds};
    nanosleep(&reqDelay, (struct timespec *)NULL);
}

/**
 * exportAndOut
 * Export a pin (if not already exported) and set the direction to output
 * @params
 *  int pinNum: the pin number to be exported and set for output
 */
static void exportAndOut(int pinNumber)
{
    char fileNameBuffer[1024];
    sprintf(fileNameBuffer, "/sys/class/gpio/gpio%d/direction", pinNumber);
    FILE *gpioDirP = fopen(fileNameBuffer, "w");

    // Check for whether the pin is exported
    if (gpioDirP == NULL) {
        // If the pointer is NULL, the pin has not been exported, so export it
        FILE *pFile = fopen("/sys/class/gpio/export" , "w");
        if (pFile == NULL) {
            printf("ERROR: Unable to open export file.\n");
            exit(1);
        }
        fprintf(pFile, "%d", pinNumber);
        fclose(pFile);
        sleepForMs(330);

        // Pin should now be exported, so reassign the gpioDirP file pointer
        gpioDirP = fopen(fileNameBuffer, "w"); // okay to be reassigned as it was
NULL
    }

    // Now set the direction to out
    fprintf(gpioDirP, "out");
    fclose(gpioDirP);
}
```

Change of displayed color is done through `ledMatrix_setPixel(int x, int y, int colour)`, where the variable of note is color. Since the board in the sample code is being driven through GPIO, and not through PWM, there are only seven colors available to be displayed, represented by three bits where one bit corresponds to red, another to green, and the last to blue.

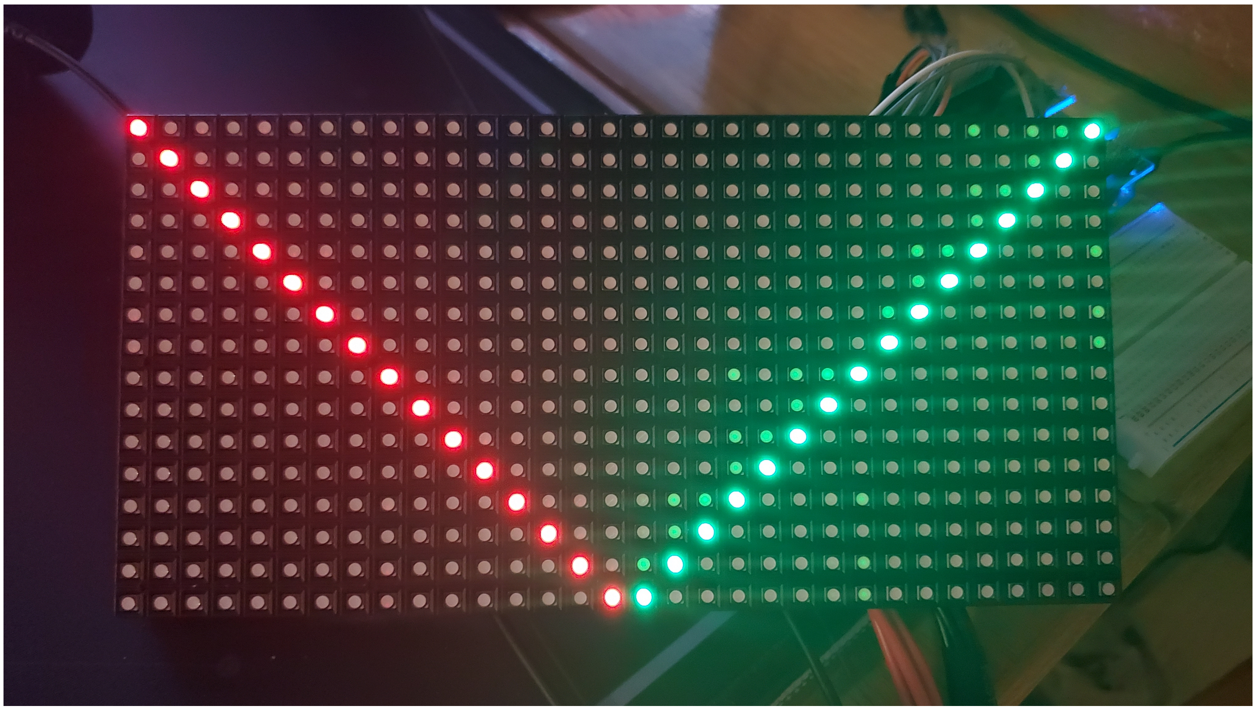

Figure 6: Expected Output of the Sample Code to the LED Matrix

Troubleshooting

- If the BeagleBone is not booting up, it is either one of two things:
    - The Molex power connector is not connected. In this case, unplug the BeagleBone, connect the Molex power cable to the LED matrix, and reconnect the BeagleBone.
    - Power is connected but the BeagleBone is not booting up. In this case, double check that the cables are connected correctly.
- It is normal for the LEDs to flicker randomly in the provided code. This is due to non-deterministic timing and nature of the operating system. If the display is flickering a lot and erratically, there are two possible options:
    - Option 1 (easier): If your program is running a lot of threads, try to schedule the thread running the LED using SCHED_FIFO to give it high priority. You will need to then adjust timings of the LED thread to ensure other threads are not starved of CPU run time.
    - Option 2 (harder): Another option is to run the LED Matrix off of the BeagleBone's PRU (Programmable Real-time Unit), which is a separate CPU running independently from Linux controlled through `remoteproc`. This should eliminate flickering completely, but is significantly harder to get up and running with shared memory (using `mmap`)

compared to thread scheduling. Refer to code provided in the link for PRU: https://beagleboard.org/static/prucookbook/#blocks_rgb1

- If parts of the LED matrix is not working, double check the connections to the BeagleBone.

References

- https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2019-student-howtos/Adafruit16x32LEDMatrix.pdf: 2019 Adafruit 16x32 LED Matrix Guide for the table in Figure 1 and 2
- https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2018-student-howtos/Adafruit16x32LEDMatrixGuide.pdf: 2018 Adafruit 16x32 LED Matrix Guide (Group A) for the table in Figure 3 explaining the functionalities in the pin and for the code from which the provided code is modified from (originally by Janet Mardjuki)