ECE Center for Intelligent Systems Banner Display Interface Documentation

Performance

Each pixel draws up to 0.06 Amps. The maximum power per panel needed is 128 * 0.06 = 7.68 Amps. This only occurs if every pixel is on full white. To limit going over the maximum power for three displays, we will limit using all pixels at a time. It is recommended to at least have half the power which we do with our current power supply.

Driving Pins

The total number of pins needed for the matrix color is 6 GPIOs. 3 of these pins will control the red, green, and blue LEDs of the top half of the board, while the other 3 will control the red, green, and blue LEDs of the bottom half of the board. There will be a total of 3 control pins where one GPIO pin will control if all LEDs are lit, and the other two will be the high speed clock and data latching for clocking the RGB data to the matrix. For the matrix address pins, we would need a total of 4 where each pin is used for multiplex circuitry. Total GPIOs here is 13.

All other pins will be used to drive the RGB matrix through the HAT. Adafruit RGB Matrix HAT + RTC is considered being used as it can provide safe power to the system, has a real time clock, and is compatible with HUB75 type RGB Matrices. This HAT will use most or all pins for the microcontroller. The HAT was designed specifically to work with a Raspberry Pi and it uses all 40 GPIO pins with a 2x20 socket header. Raspberry Pi boards we are considering include the Zero, Zero W, or Zero WH.

If we do not use a RGB Matrix Hat, we will just need 13 GPIO pins per panel and design our own PCB to connect electrical components needed.

Software Compatibility

There are three Arduino libraries that we can model our code after, for whichever microcontroller we choose.

1. RGB Matrix Panel library
2. Adafruit GFX library
3. Adafruit BusIO

These libraries contain highly useful, simplified methods to draw RGB pixels onto the display, such as:

```
matrix.drawPixel(x, y, matrix.Color333(r, g, b));
```

We could set our text size to a specific number of pixels high using something similar to:

```
matrix.setTextSize(1); // size 1 == 8 pixels high
```

We could also easily change the colors of each letter we print, in addition to specifying a font if we decide to include more than one font using something similar to:

```
matrix.setTextColor(matrix.Color333(7,0,0));
matrix.print('3');
```

Based on preexisting software, we know that the interface needs to be connected via the rows and columns of the matrix display. The pins will be mapped via the code, likely within a cpp and h file, similar to what's in the Adafruit BusIO library.

Revised Version of Doc:

Reference Link: https://bikerglen.com/projects/lighting/led-panel-1up/

- ➢ We will need to drive data to the panels by row.
    - Drive all three panels, physically, simultaneously.
    - Drive two rows per panel at a time (the R0, G0, B0 signals are for rows 0, 1, 2, …, 15, and R1, G1, B1 are for rows 16, 17, …, 31).
- ➢ Full screen cycle rate of 240 Hz
    - Line rate is 16 x 240 - 3.84KHz
    - Pixel rate is 64 x 3.84 = 246KHz
    - 100 machine cycles to manage output of a pixel
        - (have to fetch data for each of 18 data lines (3 colors x 2 rows/panel x 3 panels)
        - format for GPIO
        - toggle clock lines
- ➢ Instruction Rate of at least 24 MHz
    - PSoc 5LP boards current run at 24MHz (can be set to run as high as 64MHz)
- ➢ Needs for an MCU
    - At least 24 MHz clock rate
    - More than 6KBytes of RAM
    - Support 256 shades/color
        - 3 Bytes/pixel and or/ 18KBytes
    - 25 GPIO Pins
        - 18 data lines

        - Pins for SCLK, LATCH, A0, A1, A2, A3 and BLANK

- ➢ Memory Map for 3 panels
    - 64 x 32 x 3 planes x 3 bits = 18,432 bits
    - 1 RGB pixel value per byte
        - 6K bytes to store the entire display's image
    - 1-bit mode (ON/OFF) for each pixel giving us 8 colors:
        - Red (R)
        - Amber (R+G)
        - Green (G)
        - Aqua (G+B)
        - Blue (B)
        - Purple (R+B)
        - Black (all off)
        - White (R+G+B)

Links for connecting power supply

1. https://www.ebay.co.uk/itm/DC-12V-Power-Supply-5A-10A-12-5A-15A-20A-30A-Adapter-Waterproof-IP67-Pigtail-lot/201355039705
2. https://www.amazon.com/3-prong-pigtail/s?k=3+prong+pigtail
3. https://www.youtube.com/watch?v=7YGRS1zqcA8
4. Bergen 3 prong pigtail
5. Spades
6. https://forum.digikey.com/t/psoc-uart-example/13231

7. https://community.infineon.com/t5/Knowledge-Base-Articles/Programming-PSoC-3-PSoC-4-and-PSoC-5LP-MCUs-using-PPCLI-KBA232937/ta-p/288563
8. https://bikerglen.com/projects/lighting/led-panel-1up/#Required_Software
9.

	Requirements	Raspberry Pi Zero WH	Raspberry Pico WH	PSoC 5 LP	Arduino Mega 2560	BeagleBone Black
RAM	**>> 6 KB**	512 MB	264 KB	64 KB	18 KB	512 MB
Pins	**25**	40	26	62	54	92
Price	**< $100**	$15	$6	$92.50	$48.40	$52.50
CPU Clock Rate	**24 MHz**	1 GHz	133 MHz	80 MHz	16 MHz	1 GHz
CPU Bit Width	**12 bit**	32 bit	12 bit	32 bit	8 bit	16 bit
Ease of driving GPIO	**high**	high	high	high	high	high
Library support	**N/A**	low	low	low	high	medium
Familiarity	**N/A**	low	low	high	low	very low

https://www.digikey.com/en/products/detail/beagleboard-by-seeed-studio/102110420/12719590