Waveshare RGB-Matrix-P5-64x32 Guide

by Quanzhao Liang, Takuto Sakaemura, Tongyue Zhao
Last update: Apr 12, 2025

Guide has been tested on

BeagleY-AI (Target):	Debian 12.8
PC OS(host):	Debian 12.8

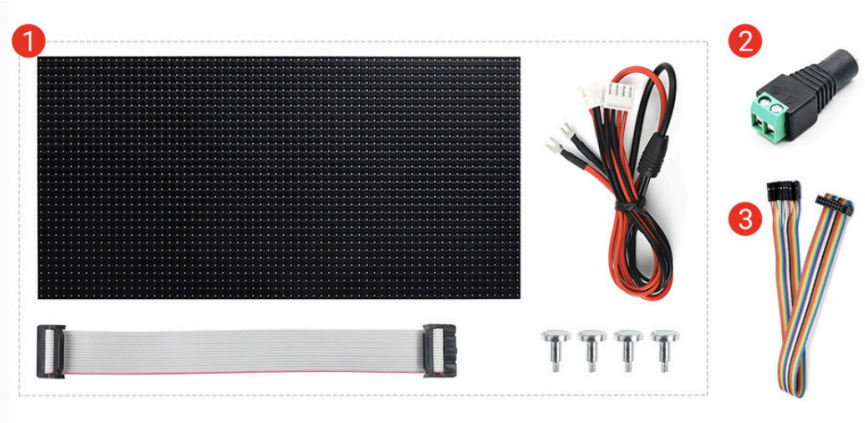
Table of Contents

1. Device Intro.....2
2. Wiring.....3
3. Driving the Matrix.....4
4. Troubleshooting…….....5
5. References……..5

1. Device Intro

RGB-Matrix-P5-64x32 is a Full-Color LED Matrix Panel with onboard 2048 RGB LEDs and 5mm pitch. The package usually includes the following contents:

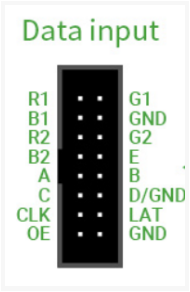
1. RGB-Matrix-P5-64x32 LED matrix x1
2. Power supply terminal adapter x1
3. 16P wire ~30cm x1



You will need an extra power supply for the LED matrix. For this project, we are using a **5V/4A adapter**.

The LED matrix has 16 pins for the control signals:

R1, G1, B1: Color data for the top half of the matrix
R2, G2, B2: Color data for the bottom half of the matrix
A, B, C, D: Selects one of the 8 rows in the matrix to display
CLK: Clock signal used for shifting color data into the matrix's internal shift registers.
LAT: When high, data in the shift registers is copied to the LED output registers.
OE: Active low output enable. Set this high to turn off all the LEDs temporarily.
GND x3: Pins need to connect to ground.



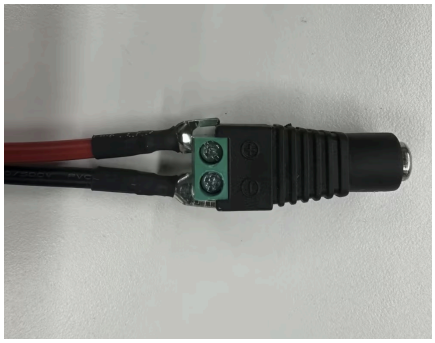
There are also pins for data output, allowing you to connect multiple matrices into one large panel.

2. Wiring

The LED matrix has 13 control pins and 3 ground pins. In our project, we removed our Zen Hat to obtain all the GPIO pins on BeagleY-AI since we don't need any hardware for the Zen Hat. If you want to do it with a Zen Hat, a GPIO expander may be an option.

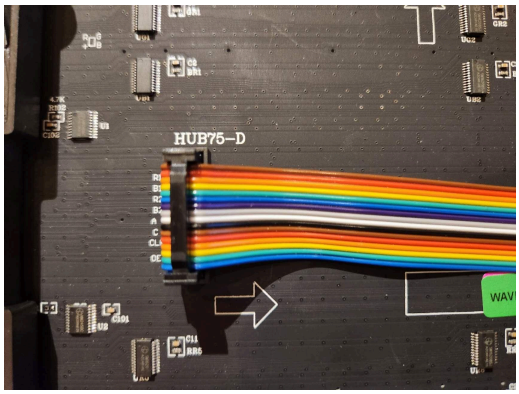
Power Supply Connection

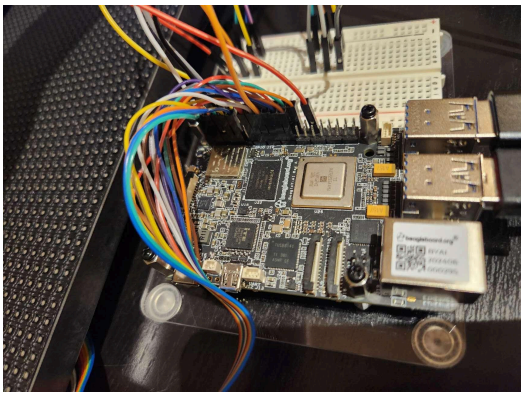
The red wire connects to "+" and the black wire connects to "-".



Main Control Wiring

Make sure to connect the wire to the GPIO pins on your target in the correct order. From top to bottom, the wires connecting are R1, G1, B1, GND, R2, G2, B2, GND, A, B, C, D, CLK, LAT, OE, GND.





You can refer to the BeagleY-AI pinout to help with connecting the wires to the GPIO pins on your device. (https://pinout.beagleboard.io/pinout/)

3. Driving the Matrix

The display of the LED matrix is multiplex, which means only one row on the top half and one row on the bottom half illuminated at a time. To display an image or animation, the entire LED panel needs to turn the rows on and off fast enough to display continuous frames without noticeable flickering.

The process used to refresh the display is the following:

1. Select the rows want to display by configuring the pins A, B, C, and D.
2. Set the OE high to disable output while shifting data.
3. For each column for the corresponding row, set the color data. Pulse the CLK pin to shift the data in.
4. Latch the column data for the row by pulsing the LAT pin.
5. Set OE low to display the frame.

The following sample code using the gpiod library demonstrates the process of refreshing the display. You can start by drawing simple shapes to test it.

```c
void led_matrix_update(void)
{

    for (int row = 0; row < MATRIX_HEIGHT / 2; row++)
    {
        // Set row address (A,B,C,D)
        gpiod_line_set_value(line_a, (row >> 0) & 1);
        gpiod_line_set_value(line_b, (row >> 1) & 1);
        gpiod_line_set_value(line_c, (row >> 2) & 1);
        gpiod_line_set_value(line_d, (row >> 3) & 1);

        // Disable output while shifting data
        gpiod_line_set_value(line_oe, 1); // OE high (disable)

        // Shift data for this row
        for (int col = 0; col < MATRIX_WIDTH; col++)
        {
            // Top half data
            RGB_Color top_pixel = frame_buffer[row][col];
            // Bottom half data
            RGB_Color bottom_pixel = frame_buffer[row + MATRIX_HEIGHT / 2][col];

            // Set color data
            gpiod_line_set_value(line_r1, top_pixel.r > 0);
            gpiod_line_set_value(line_g1, top_pixel.g > 0);
            gpiod_line_set_value(line_b1, top_pixel.b > 0);

            gpiod_line_set_value(line_r2, bottom_pixel.r > 0);
            gpiod_line_set_value(line_g2, bottom_pixel.g > 0);
            gpiod_line_set_value(line_b2, bottom_pixel.b > 0);

            gpiod_line_set_value(line_clk, 1);
            gpiod_line_set_value(line_clk, 0);
        }

        // Latch the data
        gpiod_line_set_value(line_lat, 1);
        gpiod_line_set_value(line_lat, 0);

        // Enable output
        gpiod_line_set_value(line_oe, 0); // OE low (enable)
        gpiod_line_set_value(line_oe, 1); // OE high (disbale)
    }
}
```

4. Troubleshooting

- If you see a “device busy” message when executing the code, it’s likely due to the overlay we added for our assignment. To fix this, disable the overlay:
  (byai)$ nano /boot/firmware/extlinux/extlinux.conf
  Comment out the line with a ‘#’.
- If you notice noticeable flickering on your display, make sure you haven’t added any pulse delay when triggering the pins.
- A 5V/4A power supply is recommended, otherwise, the matrix would not work properly.
- When initializing the GPIO pins using the gpiod library, make sure the chip and line numbers correctly correspond to their respective GPIO pins. You can find them out by:
  (byai)$ gpiofind gpio#

5. References

https://www.waveshare.com/rgb-matrix-p5-64x32.htm
https://bikerglen.com/projects/lighting/led-panel-1up/#Driving_the_Panel
https://www.bigmessowires.com/2018/05/24/64-x-32-led-matrix-programming/