

Project Write-up

Group Member:

- JiaMing Wu (301278354)
- Zihao Wu (301280709)
- Xiaolu He (301288416)
- Yiwei Ma (301278307)

System Explanation:

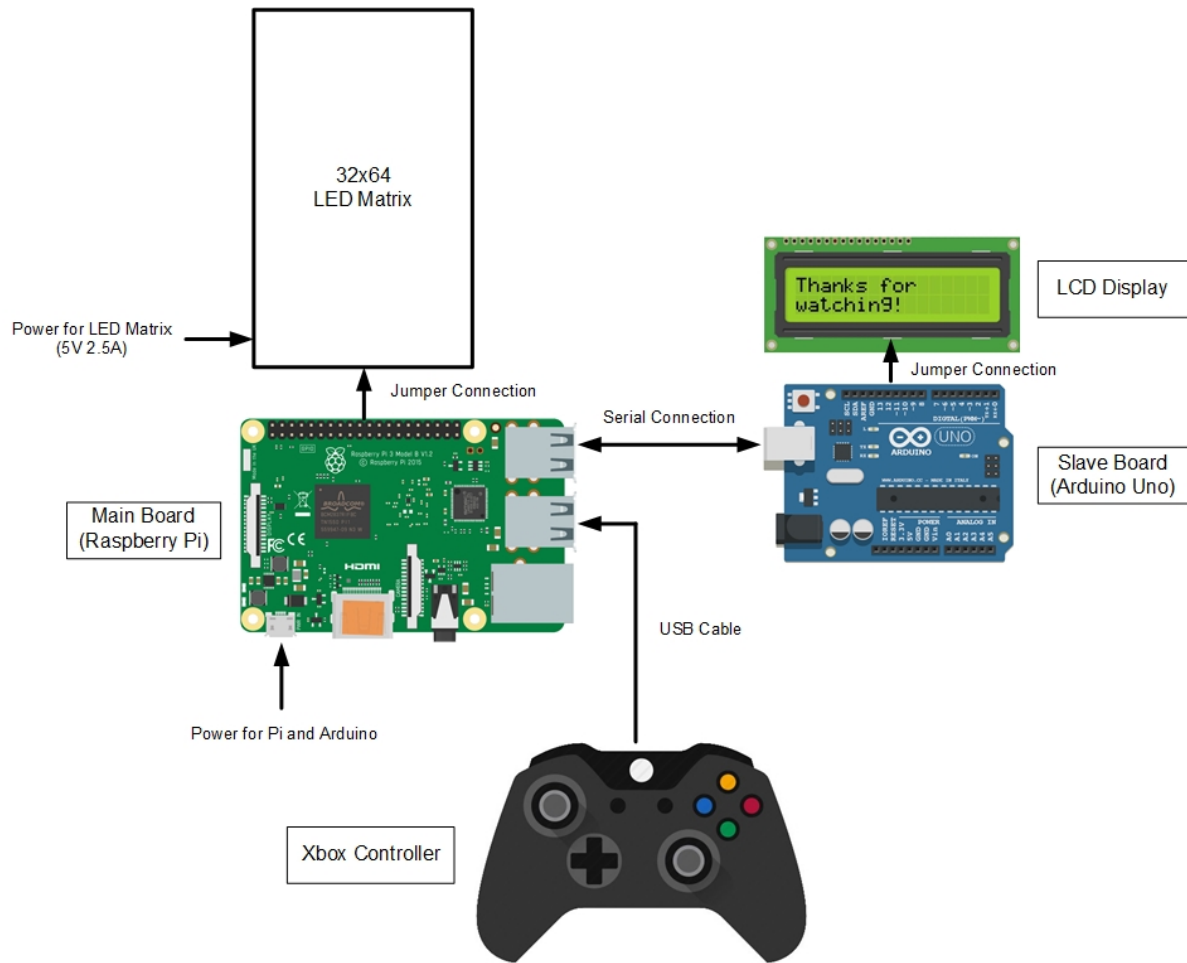


Figure 1. Project Frame

Our project is to create a 2D space shooter game similar to Space Invader (Figure 1. above). The game display is a 64 x 32 RGB LED matrix (Figure 2. below), and players can use Xbox controller to control movements. The LED matrix can be driven properly on full brightness with a 5V 2.5A power source. As shown in figure.3, we jumpwire the GPIO pins on the Raspberry Pi to the ribbon cable connected to the LED matrix. Moreover, we implemented the serial text display to display the final score (see Figure 3. below). To improve responsiveness, we implement two threads in total. Using one thread to monitor the Xbox controller inputs, filter and debounce

inputs before signaling the game thread. The other thread is responsible to update game states, display image on LED matrix and send current score to Arduino.



Figure 2. 64*32 LED Matrix

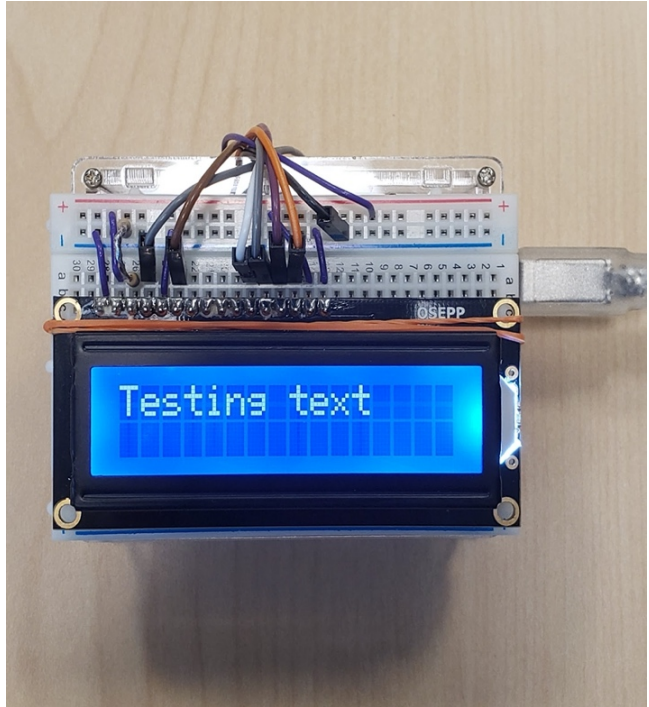


Figure 3. Arduino LCD Display

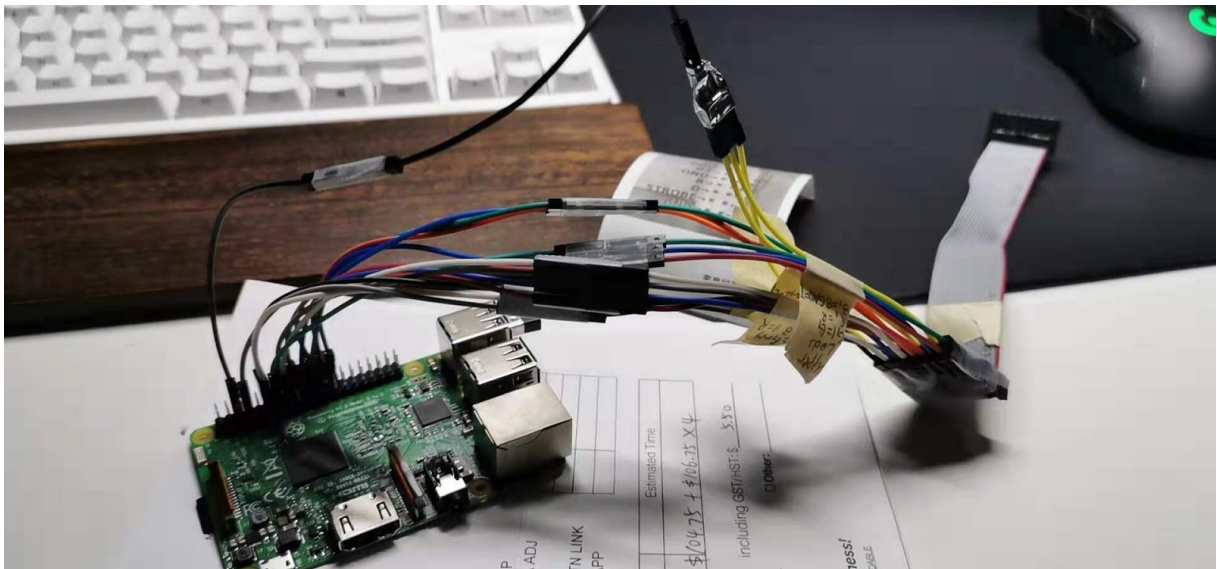


figure.4 Raspberry pi's jump wire

Remaining Challenge:

- During the demo, we found out that the memory does not free up properly before each restart. It caused a stack overflow, and program crashed unexpectedly.
- Some players complained about the lack of progressive difficulty. The difficulty should increase as the player progresses.
- Flickering may occur on a specific row, and it might be caused by a loose connection or interference.
- Adapted brightness adjustment according to ambient brightness.
- Duo play support
- Gaming sounds are not implemented, because we found that there are GPIO pin conflicts in gaming sound effects and LED Matrix. So, we could use serial connection to offload audio playback task to another slave board.

Feature table:

Description	Host/Target	Comp	Code	Notes
LED matrix driver	Host (Pi)	5/5	C/C++	Using well implemented Raspberry Pi library by hzeller on Github . We started out writing driver by ourselves, but we encounter flickering issue. We end up using double buffers and v-sync function in hzeller's library. Dev time: 11 days
Xbox controller	Host (Pi)	5/5	C	Self-implemented, using linux joystick library Dev time: 2 days
Serial port between Pi and Arduino	Host/Slave	5/5	C	Iterated from sample code Github . Did some error fixes base on our make environment Dev time: 2 days
Arduino LCD display	Slave(Arduino)	5/5	C/C++	Sample code from Arduino tutorial with few modifications Dev time: 1 days

Joystick	Host (BBG)	4/5	C	Modified from assignment one. We have implemented the joystick for early iterations of the game. This module is discarded after we had controller support. Dev time: 1 days
Game Logic	Host(BBG/Pi)	4/5	C	Although the logic has been tested well, it can still be improved in collision detection and gameplay design. During the demo, the program accidentally shut down, because of stack overflow. We spend almost half of the dev time on this module to polish gameplay. Dev time: 8 days

Extra Hardware:

1. Adafruit 64*32 RGB LED Matrix
2. Raspberry Pi
3. Arduino
4. LCD Display
5. Bread board
6. Jump wires
7. Resistors
8. Soldering iron
9. Power supplies
10. Xbox controller

Extra Software Used:

1. Arduino IDE
2. [rpi-rgb-led-matrix](#) library
3. [Pixel Art](#)
4. VS Code - Live Share extension

References:

- *Hzeller. (2019, October 7). `hzeller/rpi-rgb-led-matrix`. Retrieved October 2019, from <https://github.com/hzeller/rpi-rgb-led-matrix>.*
- *Industries, A. (n.d.). 64x32 RGB LED Matrix - 6mm pitch. Retrieved October 2019, from <https://www.adafruit.com/product/2276>.*
- *Burgess, P. (n.d.). 32x16 and 32x32 RGB LED Matrix. Retrieved October 2019, from <https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/new-wiring>.*
- *HelloWorld. (n.d.). Retrieved November 2019, from <https://www.arduino.cc/en/Tutorial/HelloWorld>.*
- *Xanthium-Enterprises. (n.d.). `xanthium-enterprises/Serial-Port-Programming-on-Linux`. Retrieved November 2019, from [https://github.com/xanthium-enterprises/Serial-Port-Programming-on-Linux/blob/master/USB2SERIAL_Write/Transmitter \(PC Side\)/SerialPort_write.c](https://github.com/xanthium-enterprises/Serial-Port-Programming-on-Linux/blob/master/USB2SERIAL_Write/Transmitter(PC%20Side)/SerialPort_write.c).*
- *Demo Link: <https://drive.google.com/open?id=12EcEZppbxfB3WhS9oTiQqW1d3CG7j7AC>*