

Lightbox

Oviya Seeniraj, Bhavya Ranjan, Saul Diaz

1 Device Functionality

Our project functions as a customizable lightbox. This was done by building a circuit that allows one to interact with the LED Matrix we have through a PS2 keyboard or through the serial port of Arduino Idle, which all runs through an Arduino Mega. With the help of libraries that allows us to interact with the matrix and the PS2 keyboard. The PS2 library allows us to receive user input from the keyboard and print it onto the LED matrix. While the LED Matrix library allows us to code commands which activate upon typing a keyword on the keyboard. For example, typing in a color such as “red” or “blue” before typing in what you want to display allows you to display your message in that color. There are other special commands that are activated by typing in other special keywords. Such as “circles” which display circles onto the LED Matrix and “flashing text” which allows your message to appear multiple times in different colors.

Systems Functional Diagram:



2 Hardware Components

The following hardware components we used:

1. A 64 x 32 with a 3mm pitch LED Matrix from Adafruit (\$45):
<https://www.adafruit.com/product/2279>
2. Arduino Mega (\$49):
https://www.amazon.com/gp/product/B0046AMGW0/ref=ppx_yo_dt_b_asin_title_o04_s01?ie=UTF8&psc=1
3. A PS2 keyboard (\$19):
https://www.amazon.com/dp/B097ZLWS7S?psc=1&ref=ppx_yo2ov_dt_b_product_detail

4. 5V, 10A power supply (\$20):
<https://www.amazon.com/SHNITPWR-Converter-Transformer-5-5x2-5mm-Controller/dp/B07Q26YG61/>
5. Femal PS2 Keybaord Connector (\$10):
https://www.amazon.com/gp/product/B08GKVKC82/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&th=1

images from Adafruit, Arduino, Arduino Project Hub, and Amazon



3 Design Timeline

Week 5: Purchase components, draw schematics and run them by TAs.

Week 6: Built the circuit for the LED Matrix. We tested it and believed that it was working fine.

Week 7: Started troubleshooting the circuit for the LED Matrix because it was not functioning the way it was supposed to. We thought we were able to find the solution to our LED Matrix problem.

Week 8: Tried the solution from the last week, which was fixing the power supply connection but had no success. We were still working on getting the LED Matrix to work properly. But at least we had the circuit for the PS2 keyboard.

Week 9: Finally solved the problem with the LED Matrix, it was an issue with the jumper cable that came with the matrix. We got it to work and connected it to the PS2 Keyboard. This was the week we got started writing out the code we needed to get both the keyboard and matrix to work together.

Week 10: Final troubleshooting and preparing for the science fair.

4 Documentation

These are the following libraries we used:

1. PS/2 Keyboard Input library
(<https://www.arduino.cc/reference/en/libraries/ps2keyboard/>)
2. RGB Matrix Panel Library
(<https://www.arduino.cc/reference/en/libraries/rgb-matrix-panel/>)

5 Software Design

Pseudocode plan:

1. Include the two libraries and std namespace for strings
2. Define pins
3. Declare keyboard, matrix, input string, and any other variables that need to be available to everything in scope
4. In setup, begin serial and keyboard
5. In loop, have two ifs to consistently check for text being entered in the keyboard or serial monitor
 - a. For both, text will be printed by calling the print function
 - b. For keyboard, enter -> call print, delete -> code delete for prev char
6. In printMatrix, initialize the matrix and execute the special graphic functions
 - a. For text printing, call the helper function for centering and anti-split-in-middle of words
7. Create helper functions as needed for blinking text (when too long or blink is typed), flashing text, circles, etc.

Actual code:

```
#include <PS2Keyboard.h>
#include <RGBmatrixPanel.h>
using namespace std;

const int DataPin = 4;
const int IRQpin = 3;
#define CLK 11 // USE THIS ON ARDUINO MEGA
#define OE 9
#define LAT 10
#define A A0
#define B A1
#define C A2
#define D A3

PS2Keyboard keyboard;
RGBmatrixPanel matrix(A, B, C, D, CLK, LAT, OE, false, 64);
String str = "";

int enters = 0;
int lineLeft = 10;
int calls;

void setup() {
    keyboard.begin(DataPin, IRQpin);
    Serial.begin(9600);
    Serial.println("Keyboard Test:");
}

void loop() {

    if (Serial.available()) {
        Serial.println("back at serial");
        delay(100);
        while (Serial.available() > 0) {
            if (calls > 0)
            {
                enters++;
                lineLeft = 10;
            }
            printMatrix(Serial.readString());
        }
        if (keyboard.available())
        {
            Serial.println("back at keyboard");
            // read the next key
            char c = keyboard.read();
            if (c == PS2_DELETE)
            {
                String strCopy = "";
                for (int i = 0; i < str.length() - 1; i++)
                {
                    strCopy = strCopy + str[i];
                }
                str = strCopy;
            }
            else if (c == PS2_ENTER)
            {
                printMatrix(str);
            }
            else
            {
                Serial.print(c);
                str = str + c;
            }
        }
    }

    void printMatrix(String str)
    {
        calls++;
        matrix.begin();
        matrix.setTextColor(matrix.Color333(7, 7, 7));

        if (str.length() > 44 || str.substring(0, 5) == "blink")
        {
            if (str.substring(0, 5) == "blink")
            {
                str = str.substring(5, str.length());
            }
            blinkWords(str);
        }
        else if (str == "circles")
        {
            circles();
            str = "";
        }
        else if (str == "x")
        {
            matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 7, 0));
            delay(300);
            matrix.drawRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(7, 7, 0));
            delay(300);
        }
    }
}
```

```

matrix.drawLine(0, 0, matrix.width() - 1, matrix.height() - 1, matrix.Color333(7, 0, 0));
matrix.drawLine(matrix.width() - 1, 0, 0, matrix.height() - 1, matrix.Color333(7, 0, 0));
delay(500);
str = "";
}
else if (str == "flashing")
{
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(7, 0, 0));
    delay(300);

    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(7, 7, 0));
    delay(300);
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 7, 0));
    delay(300);
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 0, 7));
    delay(300);
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(7, 0, 7));
    delay(300);
    str = "";
}
else if (str.substring(0, 3) == "red")
{
    matrix.setTextColor(matrix.Color333(7, 0, 0));
    wrapWords(str.substring(3, str.length()));
    str = "";
}
else if (str.substring(0, 6) == "yellow")
{
    matrix.setTextColor(matrix.Color333(7, 4, 0));
    wrapWords(str.substring(6, str.length()));
    str = "";
}
else if (str.substring(0, 5) == "green")
{
    matrix.setTextColor(matrix.Color333(0, 7, 0));
    wrapWords(str.substring(5, str.length()));
    str = "";
}
else if (str.substring(0, 4) == "blue")
{
    matrix.setTextColor(matrix.Color333(0, 0, 7));
    wrapWords(str.substring(4, str.length()));
    str = "";
}
else if (str.substring(0, 6) == "purple" || str.substring(0, 6) == "violet")
{
    matrix.setTextColor(matrix.Color333(7, 0, 7));
    wrapWords(str.substring(6, str.length()));
    str = "";
}
else if (str.substring(0, 5) == "clear")
{
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 0, 0));
    str = "";
    enters = 0;
    calls = 0;
    int lineLeft = 10;
    matrix.setCursor(0, 0);
}
else if (str.substring(0, 9) == "flashtext")
{
    flashText(str.substring(9, str.length()));
}
else
{
    wrapWords(str);
    str = "";
}
Serial.println("printMatrix complete");
return;
}

void wrapWords(String str)
{
    String word = "";
    str = str + " ";
    String thisLine = "";
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] == ' ' && word.length() < lineLeft)
        {
            Serial.println(word);
            thisLine = thisLine + word;
            if (i != str.length() - 1)
            {
                thisLine = thisLine + " ";
            }
            matrix.fillRect(0, enters * 8, matrix.width(), 8, matrix.Color333(0, 0, 0));
            matrix.setCursor((63 - thisLine.length() * 5.5) / 2.0, enters * 8);
            matrix.print(thisLine);
            lineLeft = lineLeft - word.length() - 1;
            word = "";
        }
        else if (str[i] == ' ')
        {

```

```

matrix.println();
thisLine = "";
lineLeft = 10;
enters++;
if (enters > 3)
{
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 0, 0));
    blinkWords(str);
}
Serial.println(word);
thisLine = thisLine + word;
if (i != str.length() - 1)
{
    thisLine = thisLine + " ";
}
matrix.fillRect(0, enters * 8, matrix.width(), 8, matrix.Color333(0, 0, 0));
matrix.setCursor((63 - thisLine.length() * 5.5) / 2.0, enters * 8);
matrix.print(thisLine);
lineLeft = lineLeft - word.length() - 1;
word = "";
}
else
{
    word = word + str[i];
}
}
}

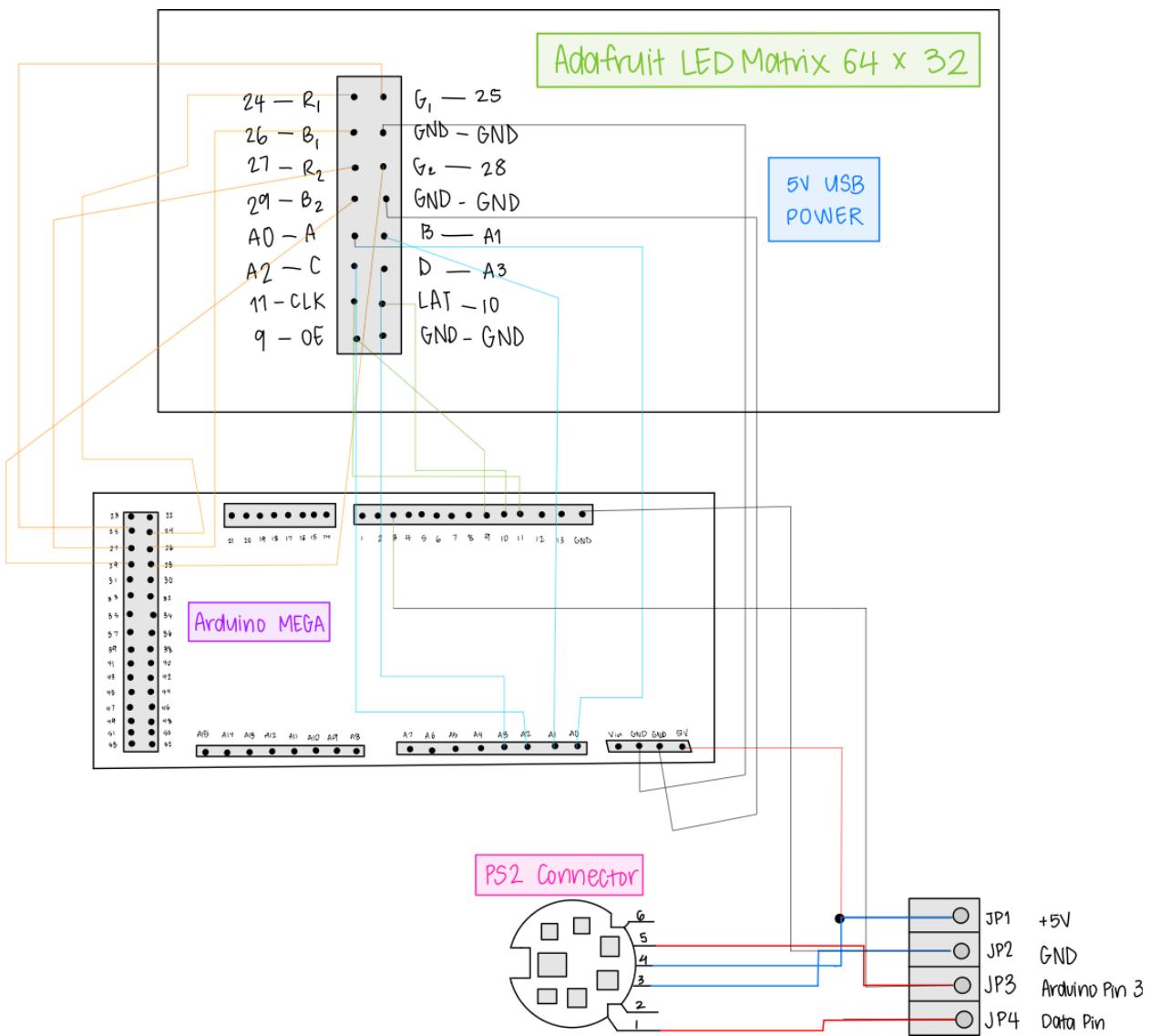
void flashText(String str)
{
    for (int i = 0; i < 5; i++)
    {
        printMatrix("red" + str);
        delay(300);
        printMatrix("yellow" + str);
        delay(300);
        printMatrix("green" + str);
        delay(300);
        printMatrix("blue" + str);
        delay(300);
        printMatrix("violet" + str);
        delay(300);
    }
}

void blinkWords(String str)
{
    String word = "";
    str = str + " ";
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] == ' ')
        {
            Serial.println(word);
            //matrix.setCursor(8, 12);
            enters = 1.5;
            wrapWords(word);
            delay(1000);
            matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(0, 0, 0));
            word = "";
        }
        else
        {
            word = word + str[i];
        }
    }
}

void circles()
{
    for (int i = 0; i < 5; i++)
    {
        matrix.drawCircle(10, 10, 10, matrix.Color333(0, 0, 7));
        delay(500);
        matrix.fillCircle(40, 21, 10, matrix.Color333(7, 0, 7));
        delay(500);
        matrix.drawCircle(10, 10, 10, matrix.Color333(7, 0, 7));
        delay(500);
        matrix.fillCircle(40, 21, 10, matrix.Color333(0, 0, 7));
        delay(500);
    }
}
}

```

6 Circuit Schematic



7 Circuit Prototype

