ACTIVITY 7 - Arduino 7: Keypad

Jason Harvey Lorenzo, March 23, 2024
ENGG 122.02, Department of Electronics, Computer, and Communications Engineering
School of Science and Engineering, Ateneo de Manila University
Quezon City, 1108, Philippines
jason.lorenzo@obf.ateneo.edu

***Abstract*—This paper focuses on integrating a 4x4 matrix keypad, a tactile input device featuring 16 keys organized in a grid layout, each representing a unique input. Visual feedback for keypad inputs is provided through an RGB LED controlled by three Pulse Width Modulation (PWM) pins. The LED illuminates according to predefined RGB values assigned to each keypad input. The implementation of this system is facilitated by Arduino hardware. The functionality of the system is further explained through a detailed program listing and an illustration of the Arduino setup.**

***Index Terms*—Arduino, matrix keypad, RGB LED**

I. Introduction

This activity centers around integrating a 4x4 matrix keypad, which is a tactile input device featuring 16 keys organized in a grid layout. Each key corresponds to a unique input and is typically marked with alphanumeric characters or symbols. The keypad employs a matrix configuration, linking each row and column to a microcontroller or control circuit. Upon pressing a key, it establishes a connection between a specific row and column, enabling the microcontroller to identify the pressed key based on the intersection of these elements. To provide visual feedback for keypad inputs, an RGB LED controlled by three Pulse Width Modulation (PWM) pins will be utilized. The RGB LED will illuminate according to the colors specified in Table I for the corresponding inputs.

TABLE I
LED COLOR FOR EACH KEYPAD INPUT

Keypad Input	LED Color (RGB Values)
1	255,0,0
2	255,128,0
3	255,255,0
A	128,255,0
4	0,255,0
5	0,255,128
6	0,255,255
B	0,128,255
7	0,0,255
8	127,0,255
9	255,0,255
C	255,0,127
*	102,15,15
0	15,102,15
#	15,15,102
D	255,255,255

II. Program Listing

A. Source Code

```
int RGBvalues[4][4][3]={
{{255,0,0}, {255,128,0}, {255,255,0},
  {128,255,0}},
{{0,255,0}, {0,255,128}, {0,255,255},
  {0,128,255}},
{{0,0,255}, {127,0,255}, {255,0,255},
  {255,0,127}},
{{102,15,15}, {15,102,15}, {15,15,102},
  {255,255,255}}};

void setup() {
  for(int i=2;i<=5;i++)
    pinMode(i, OUTPUT);
  for(int i=14;i<=17;i++)
    pinMode(i, INPUT_PULLUP);
  for(int i=9;i<=11; i++)
    pinMode(i, OUTPUT);
}

void color(int row, int column){
  for (int i=9; i<=11; i++){
    analogWrite(i,
      RGBvalues[row][column][i-9]);
  }
}

void loop() {
  for (int i=2; i<=5; i++){
    digitalWrite(i, HIGH);
  }
  for (int i=2; i<=5; i++){
    digitalWrite(i, LOW);
    if(!digitalRead(14)){
      color(i-2,0);
      break;
      }
    else if (!digitalRead(15)){
      color(i-2, 1);
      break;
      }
    else if (!digitalRead(16)){
      color(i-2, 2);
      break;
      }
    else if (!digitalRead(17)){
```

```
        color(i-2, 3);
        break;
        }
    }
}
```

B. How It Works

Initially, RGBvalues array was declared. This array holds the RGB values for each of the 16 keys on the keypad as identified in Table I. It's a 3D array where the first dimension represents the row, the second dimension represents the column, and the third dimension represents the Red, Green, and Blue values respectively.

In the setup function, the pins are configured accordingly: pins 2 to 5 are set as output pins to control the rows of the keypad, pins 14 to 17 are set as input pins with pull-up resistors to read the columns of the keypad, and pins 9 to 11 are set as output pins to control the RGB LED's red, green, and blue channels, respectively.

The color function takes two parameters, row and column, indicating the position of the key pressed on the keypad. It then sets the analog output of pins 9 to 11 (corresponding to Red, Green, and Blue) based on the RGB values stored in the RGBvalues array for that specific key.

The loop function continuously scans the keypad for key presses. It first activates each row of the keypad by setting pins 2 to 5 HIGH (which is the inactive state in this case). Then, it checks each column (pins 14 to 17) to see if a key is pressed (represented by a LOW digital reading). If a key is pressed, it determines the row and column of the pressed key and calls the color function to set the RGB LED to the corresponding color based on the RGBvalues array. Finally, it breaks out of the loop to prevent checking the remaining rows once a key press is detected.

III. Arduino Setup

The Arduino setup depicted in Fig. 1 illustrates an Arduino UNO connected to a laptop for both power and program uploading via a USB cord. The RGB LED is linked to PWM outputs, as specified in the setup function, alongside the ground connection. The initial four inputs of the 4x4 matrix (representing rows) are connected to pins 2 to 5 as outputs. Simultaneously, pins 14 to 17 (representing columns) are configured as input pull-up pins.

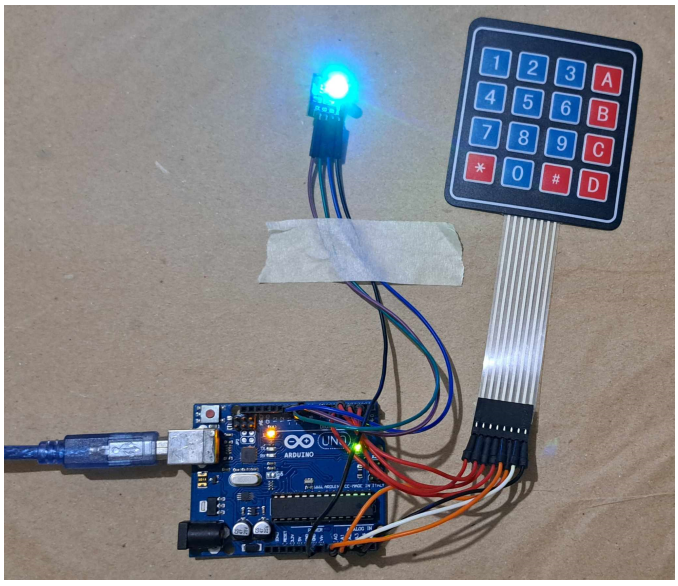

Fig. 1. Arduino setup with 4x4 matrix keypad and RGB LED with key '6' pressed