EMBEDDED SYSTEM AND DESIGN

EXPERIMENT NO. 1

SYMPHONY OF LIGHTS

Name: Navarro, Rod Geryk C.

Course/Section: CPE160P-4/A1

Group No.:

Date of Performance: 08/22/2024
Date of Submission: 08/23/2024

Cyrel O. Manlises, Ph.D.
Instructor

DISCUSSION

First of all, the first thing that I did is to think of how the LED lights should be positioned in the breadboard because the first time I saw the title of the experiment the first thing that I think about is to really mimic the actual symphony of light in Hong Kong. I am really interested in the experiment that it pushes me to somehow get the vibe of the real symphony of lights in Hong Kong. In the end I've used 4 color LED lights and positioned strategically to create a dancing illusion of lights. After positioning the LEDs, I create a circuit that will allow to trigger an input using a button that I've acquired in the kit, this button will then act as a switch to activate the various LED patterns/ sequence programmed in the board. In setting the buttons I maximized the number of inputs in the pins of the board, all of my buttons have their own input ports in the Arduino. This concept is from the YouTube video I've watched in the supplementary videos available in this experiment. I've connected 4 buttons in their designated ports A3 for the button1, A2 for the button2, A1 for the button3, and the reset button to the reset input pin. When I started the activity, I don't know how to set the reset button, however in the end I manage to successfully connect it with its purpose to reset the sequence even when it is running. I found out that it is the same with the setup of my other buttons but in the reset port with its 5v connection is also connected in the ground.

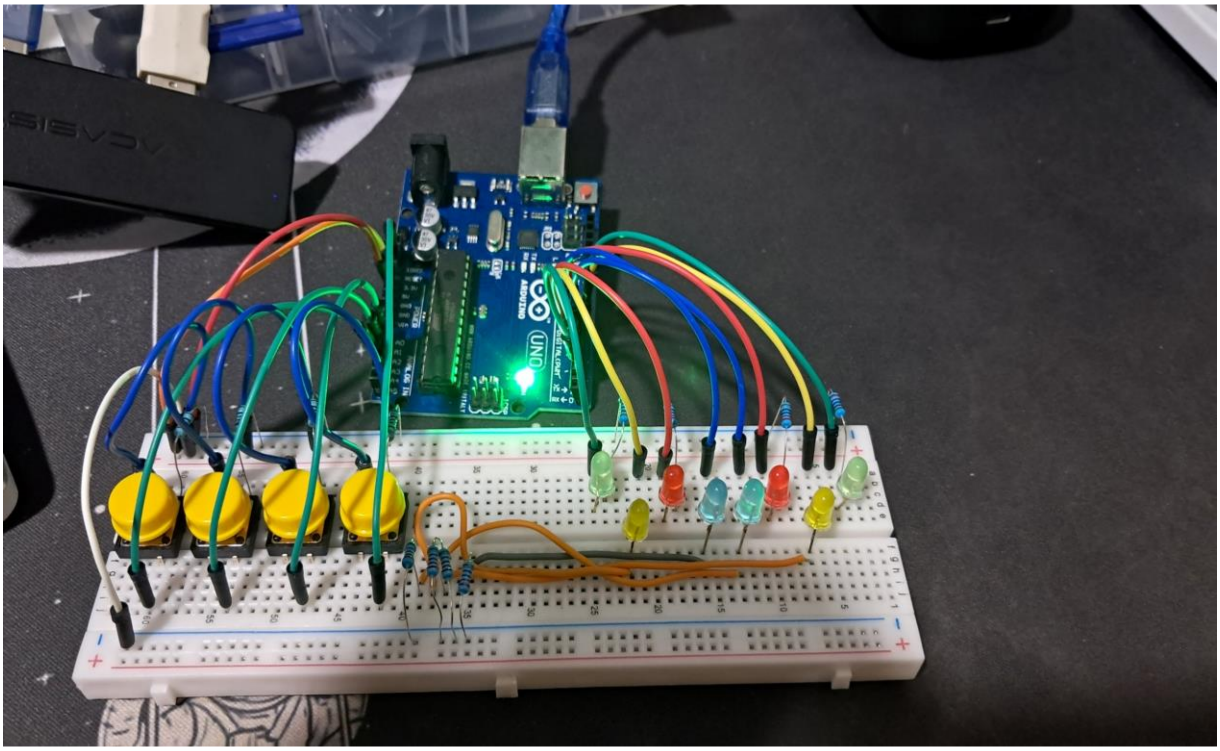
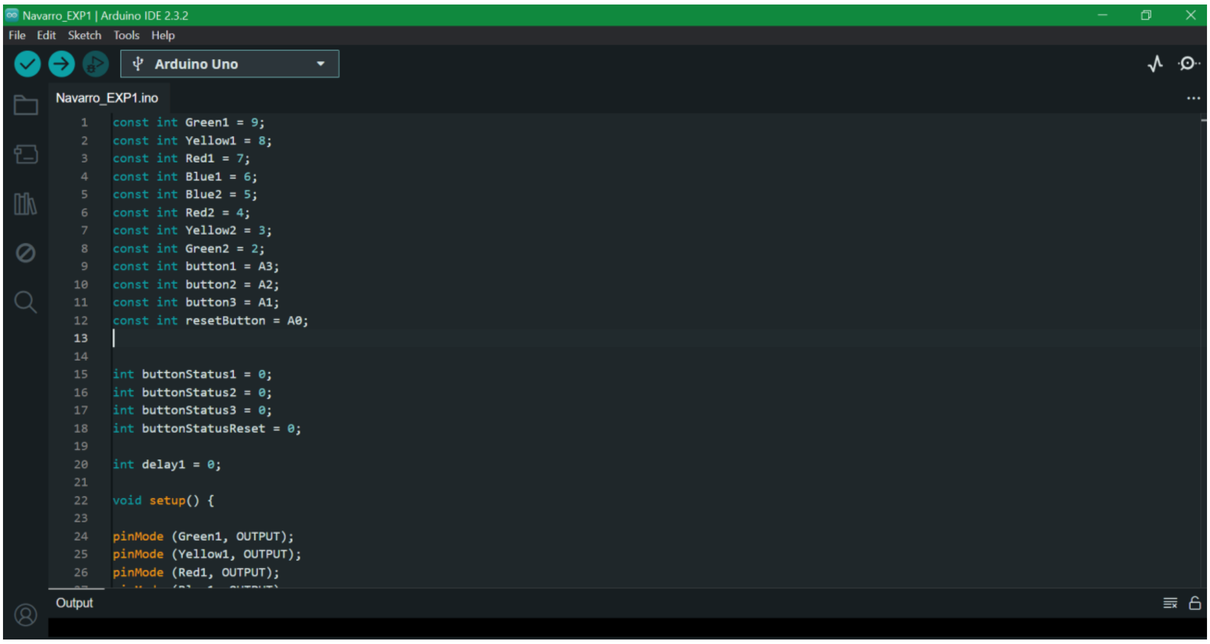


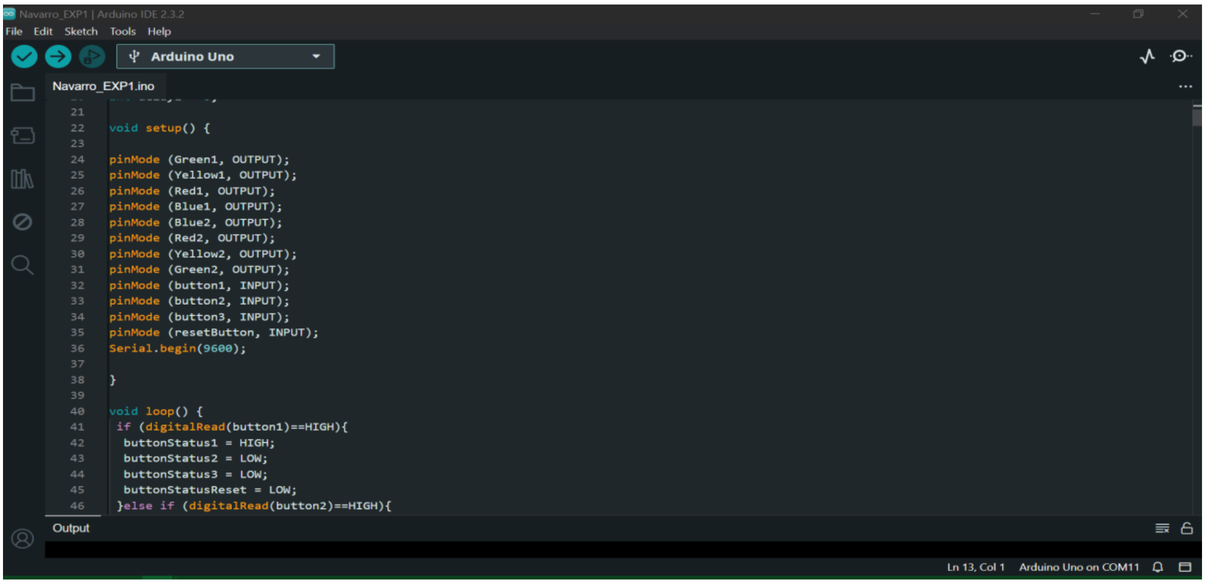
Figure 1: *The completed circuit. The LEDs are connected to the Arduino board.*

In configuring the connections, I think I properly manage it well smoothly as also shown in my raw video, LEDs are working properly the way I programmed it. The completed circuit has 8 LEDs with 4 buttons all are connected in their respective ports in the Arduino. So, after I've completed the circuit I direct my focus in programming with the Arduino Uno. The next step is to program the sequence of the LEDs based on the music that I will be playing and assign each buttons it their respective patterns and song. The songs that I've chosen to synch is Limang Taon by Juan Karlos, Huwag na Huwag Mong Sasabihin by Lola Amour version, and Para sa Streets by Hev Abi. In the coding, I first initialized the variables, the in the void setup() loop. I used the function pinMode(). The syntax for the use of pinMode() function is defined by: pinMode. The mode defines if the pin is an input, an output, or an input pull-up. This Arduino code is designed to control a series of LEDs based on the input from several buttons. The code begins by defining which pins on the Arduino will control the LEDs (Green1, Yellow1, Red1, etc.) and which will read input from the buttons. Each button is connected to a specific pin, and when pressed, it triggers different actions.



```
const int Green1 = 9;
const int Yellow1 = 8;
const int Red1 = 7;
const int Blue1 = 6;
const int Blue2 = 5;
const int Red2 = 4;
const int Yellow2 = 3;
const int Green2 = 2;
const int button1 = A3;
const int button2 = A2;
const int button3 = A1;
const int resetButton = A0;


int buttonStatus1 = 0;
int buttonStatus2 = 0;
int buttonStatus3 = 0;
int buttonStatusReset = 0;

int delay1 = 0;

void setup() {

pinMode (Green1, OUTPUT);
pinMode (Yellow1, OUTPUT);
pinMode (Red1, OUTPUT);
```

Figure 2: *Initialization of variables and the definition of the modes of the pins.*

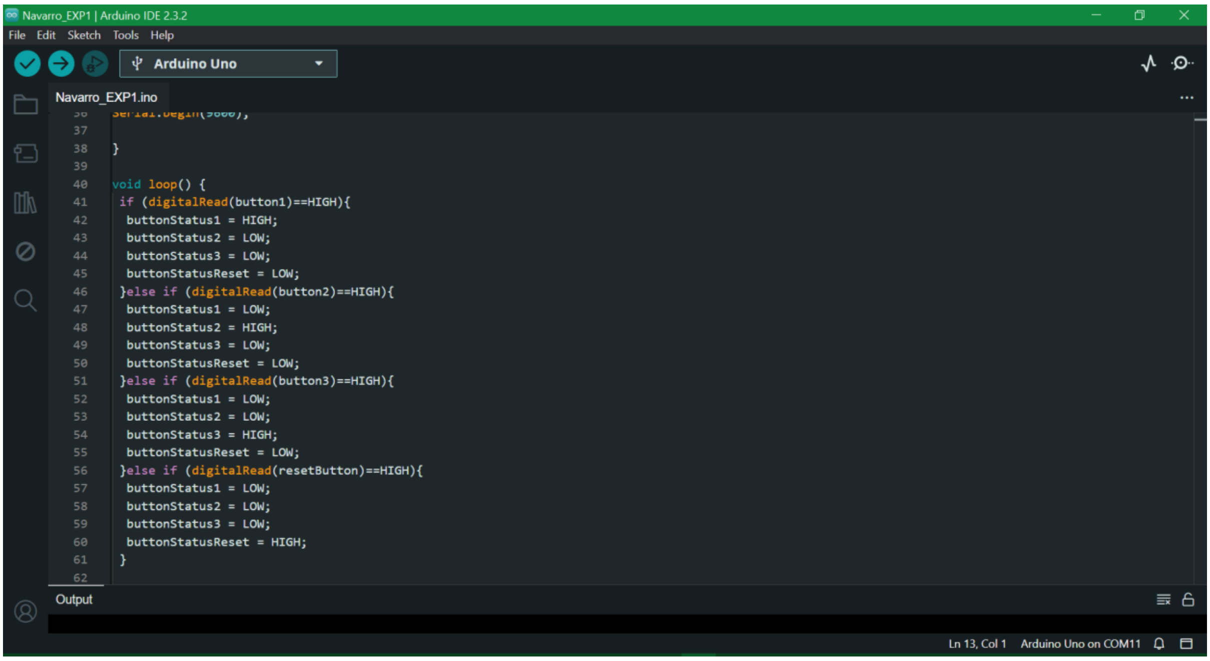


```

void setup() {

pinMode (Green1, OUTPUT);
pinMode (Yellow1, OUTPUT);
pinMode (Red1, OUTPUT);
pinMode (Blue1, OUTPUT);
pinMode (Blue2, OUTPUT);
pinMode (Red2, OUTPUT);
pinMode (Yellow2, OUTPUT);
pinMode (Green2, OUTPUT);
pinMode (button1, INPUT);
pinMode (button2, INPUT);
pinMode (button3, INPUT);
pinMode (resetButton, INPUT);
Serial.begin(9600);

}

void loop() {
 if (digitalRead(button1)==HIGH){
  buttonStatus1 = HIGH;
  buttonStatus2 = LOW;
  buttonStatus3 = LOW;
  buttonStatusReset = LOW;
 }else if (digitalRead(button2)==HIGH){
```

Figure 3: *Initialization of variables and the definition of the modes of the pins (continuation).*

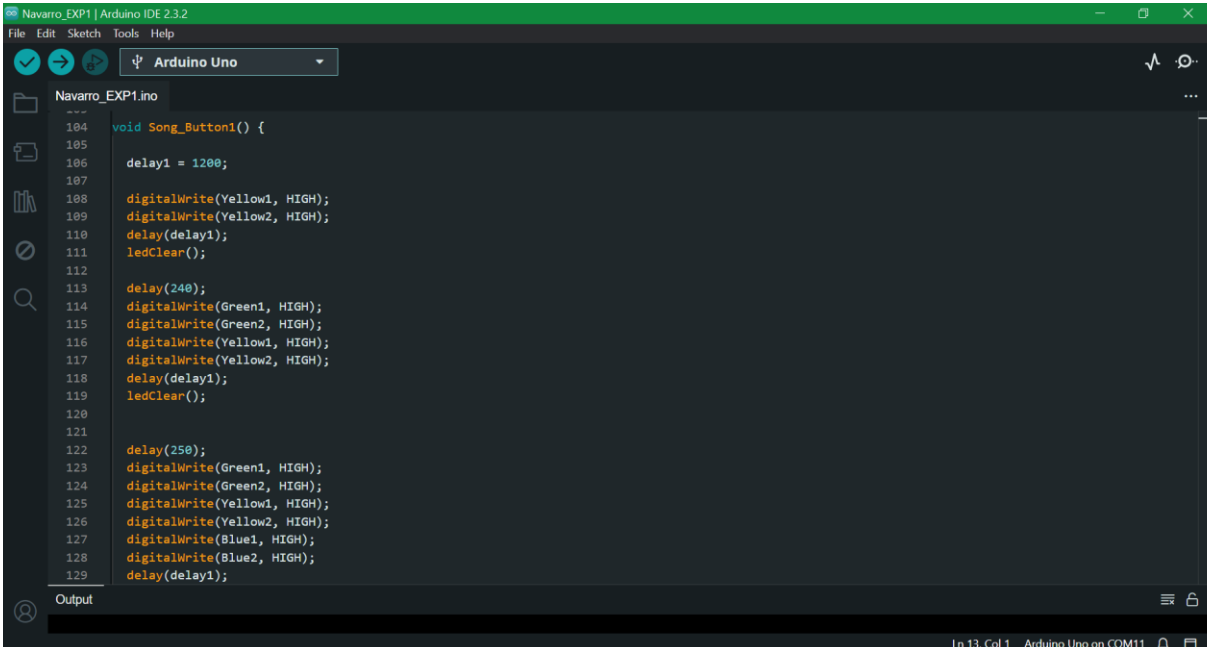
In the loop function, the Arduino constantly checks the status of the buttons. If a button is pressed, it sets the corresponding button status to HIGH and others to LOW. Depending on which button is active, a specific function is called to light up the LEDs in a particular sequence.

For example, when button1 is pressed, the Song_Button1() function is triggered. This function turns on and off various LEDs in a pattern, with specific delays in between. The LEDs light up in sequences, creating a visual effect. Similar functions exist for the other buttons (Song_Button2() and Song_Button3()), each controlling the LEDs in different ways. The code is a straightforward example of how to control LEDs with buttons, allowing for the creation of various light patterns depending on the button pressed.


Navarro_EXP1 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Navarro_EXP1.ino
Output
Ln 13, Col 1 Arduino Uno on COM11

```
  Serial.begin(9600);

}

void loop() {
 if (digitalRead(button1)==HIGH){
  buttonStatus1 = HIGH;
  buttonStatus2 = LOW;
  buttonStatus3 = LOW;
  buttonStatusReset = LOW;
 }else if (digitalRead(button2)==HIGH){
  buttonStatus1 = LOW;
  buttonStatus2 = HIGH;
  buttonStatus3 = LOW;
  buttonStatusReset = LOW;
 }else if (digitalRead(button3)==HIGH){
  buttonStatus1 = LOW;
  buttonStatus2 = LOW;
  buttonStatus3 = HIGH;
  buttonStatusReset = LOW;
 }else if (digitalRead(resetButton)==HIGH){
  buttonStatus1 = LOW;
  buttonStatus2 = LOW;
  buttonStatus3 = LOW;
  buttonStatusReset = HIGH;
 }

```

Figure 4: *The initialization of temp and reading of the value of the analog pin as well as the conditions for the sequence to play.*


Navarro_EXP1 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Navarro_EXP1.ino
Output
Ln 13, Col 1 Arduino Uno on COM11

```
void Song_Button1() {

  delay1 = 1200;

  digitalWrite(Yellow1, HIGH);
  digitalWrite(Yellow2, HIGH);
  delay(delay1);
  ledClear();

  delay(240);
  digitalWrite(Green1, HIGH);
  digitalWrite(Green2, HIGH);
  digitalWrite(Yellow1, HIGH);
  digitalWrite(Yellow2, HIGH);
  delay(delay1);
  ledClear();


  delay(250);
  digitalWrite(Green1, HIGH);
  digitalWrite(Green2, HIGH);
  digitalWrite(Yellow1, HIGH);
  digitalWrite(Yellow2, HIGH);
  digitalWrite(Blue1, HIGH);
  digitalWrite(Blue2, HIGH);
  delay(delay1);
```

Figure 5: *The initialization of temp and reading of the value of the analog pin as well as the conditions for the sequence to play (continuation).*

CONCLUSION

The experiment allowed me to explore the basics of embedded systems by creating a simple yet engaging project that mimicked the Symphony of Lights in Hong Kong. I started by carefully positioning four colored LEDs on a breadboard to create a dynamic light display. Although I initially struggled with setting up the reset button, I eventually figured out how to configure it correctly. With the help of supplementary videos, I learned how to connect buttons to the Arduino, each one controlling a different LED sequence. After successfully connecting all the components, I focused on programming the Arduino to control the LED patterns based on button inputs. Through this process, I learned how to use functions like pinMode() and how to create loops that respond to button presses, triggering different light sequences. I successfully synched my chosen three pieces of music with the 8 LEDs manually. The experiment was a valuable experience that helped me understand the fundamentals of circuit design and programming in embedded systems.