IE2070
Embedded Systems
2nd Year, 2nd Semester

Individual Assignment

IR controlled Lamp

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

4th of May 2024

Declaration

I certify that this report does not incorporate without acknowledgement, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief it does not contain any material previously published or written by another person, except where due reference is made in text.

Registration Number : IT22630070

Name : S.M.J.MALSHAN

Table of Content

1) Introduction

This project presents the design and implementation of an interactive LED control system using an Arduino Uno microcontroller board. The system is constructed using the Arduino Integrated Development Environment (IDE) and microchip Studio, and prototype is built on a breadboard.

The system controls a set of Light Emitting Diodes (LEDs), consisting of four white LEDs and one RGB (Red, Green, Blue) LED. These LEDs are manipulated using an Infrared (IR) remote control.

Initially, all LEDs are in the OFF state. The volume Up button on the IR remote is assigned to sequentially turn on the white LEDs. Once all white are activated, additional presses of the Volume Up button cycle through the colors of the RGB LED.

Conversely, the Volume Down button on the IR remote is programmed to first cycle the RGB LED through its colors in reverse order. After the completion of the RGB LED color cycle, additional presses of the button sequentially turn OFF the white LEDs.

Furthermore, the system can control the brightness of the white LEDs using the Channel Up/Down buttons on the IR remote. The Channel Up button increases the brightness of the white LEDs, while the Channel Down button decreases their brightness level.

2) Design methodology

3.2. Components

Following components were used to design the lamp:

a) Arduino Uno board
b) White LEDs - 4
c) RGB LED - 1
d) Resistor (220 ohm)
e) IR remote
f) IR receiver (3pin)
g) Jumper Wires
h) Breadboard
i) Power Supply

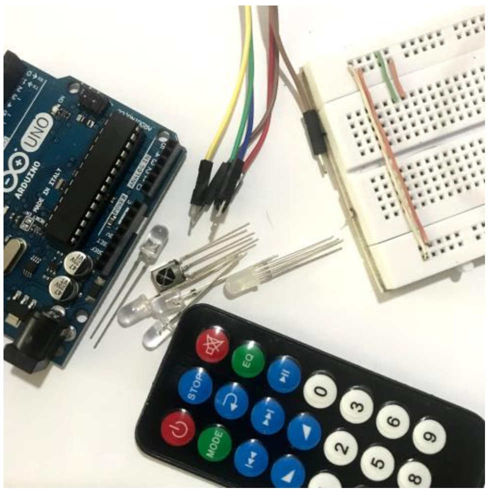


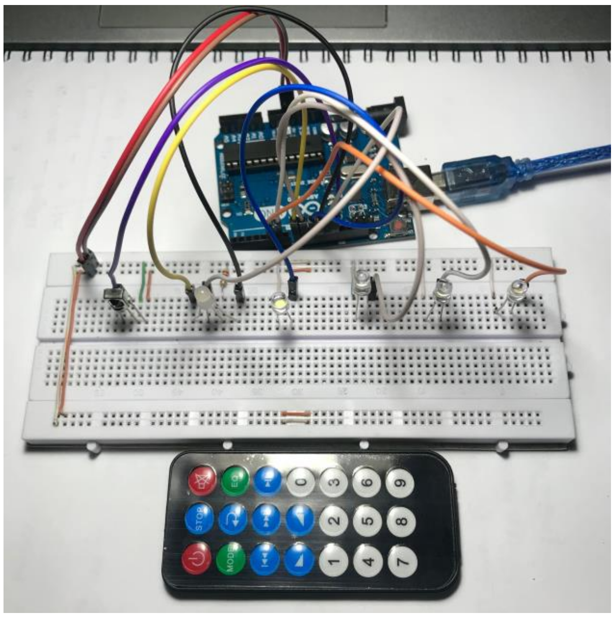
Figure 1: Components

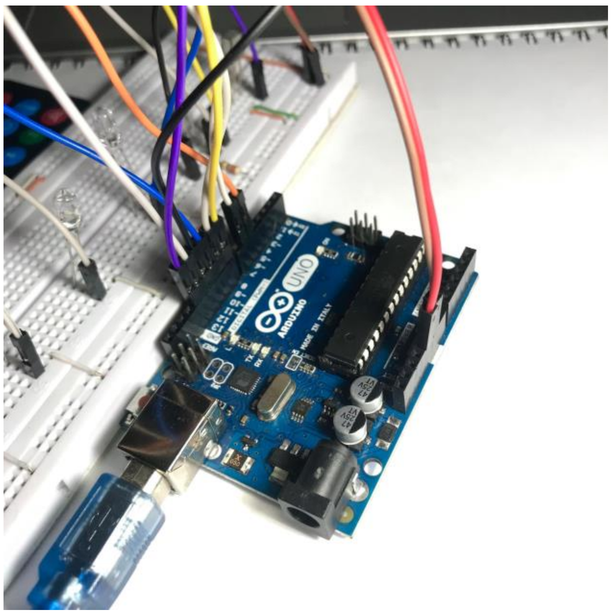
3.2. Scenario

- Initially, all LEDs are in the OFF state.
- Upon pressing the Volume Up button, 1st white LED is turned ON. Again pressing the Volume Up button, 2nd white LED is turned ON. Similarly, the sequence continues till the colors of the RGB are turned ON.
- Additionally, if the Volume Up button is repeatedly pressed, the RGB LED will toggle between the Red, Green, and Blue colors.
- When pressing the Volume Down button, first turn OFF the final color of the RGB LED. Similarly, the sequence continues till all the LEDs are turned OFF.
- Furthermore, the user can control the brightness of the white LEDs using the Channel Up/Down button.

2.3.Design implementation

Firstly, I designed a prototype of the lamp on the breadboard according to the given scenario, using above components. I used Pulse Width Modulation (PWM) pins (5,6,9,10) for control the brightness of the white LEDs. IR receiver out is connected to the pin 13 on the Arduino Uno. The IR remote is powered by a 3V battery and Uno board powered by its serial cable.
I hope to assemble the final product on the dotboard.





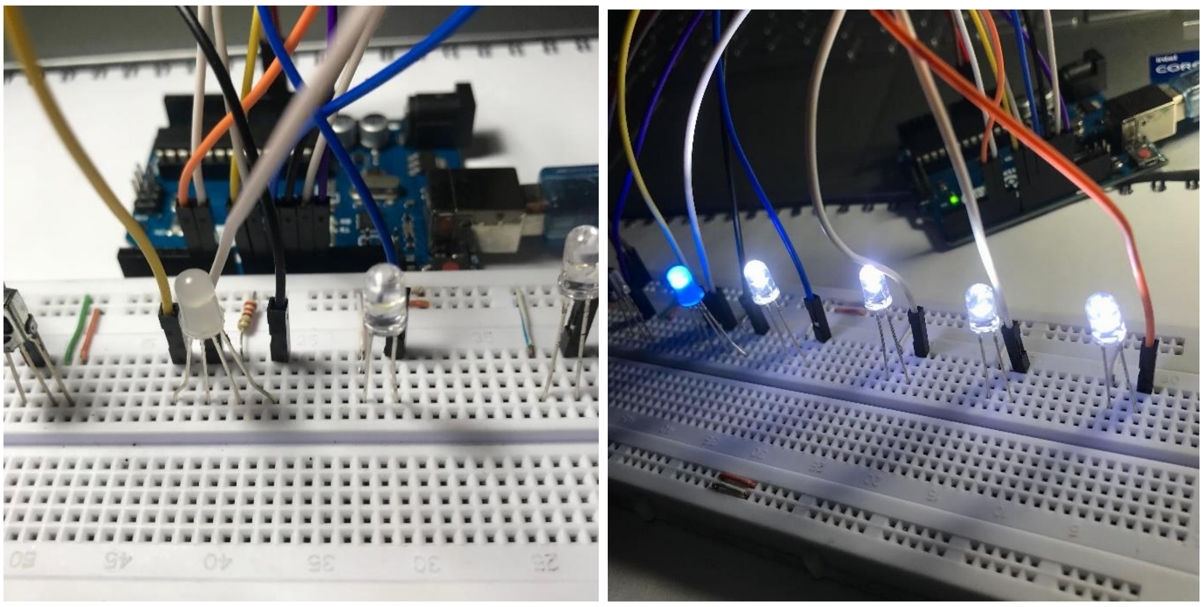
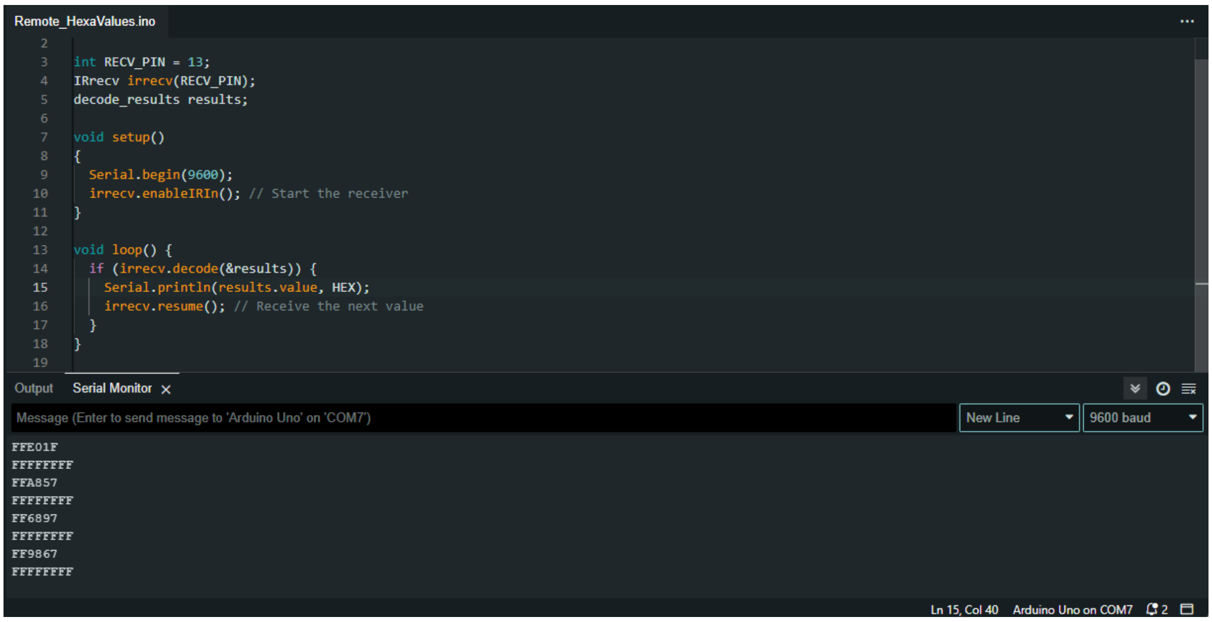


Figure 2: Implementation on the bread board

3) Code Implementation

3.1. Decode the IR remote values



Remote_HexaValues.ino

```

int RECV_PIN = 13;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}

```

Output Serial Monitor

Message (Enter to send message to 'Arduino Uno' on 'COM7') New Line 9600 baud

```
FFE01F
FFFFFFFF
FFA857
FFFFFFFF
FF6897
FFFFFFFF
FF9867
FFFFFFFF
```

Ln 15, Col 40 Arduino Uno on COM7

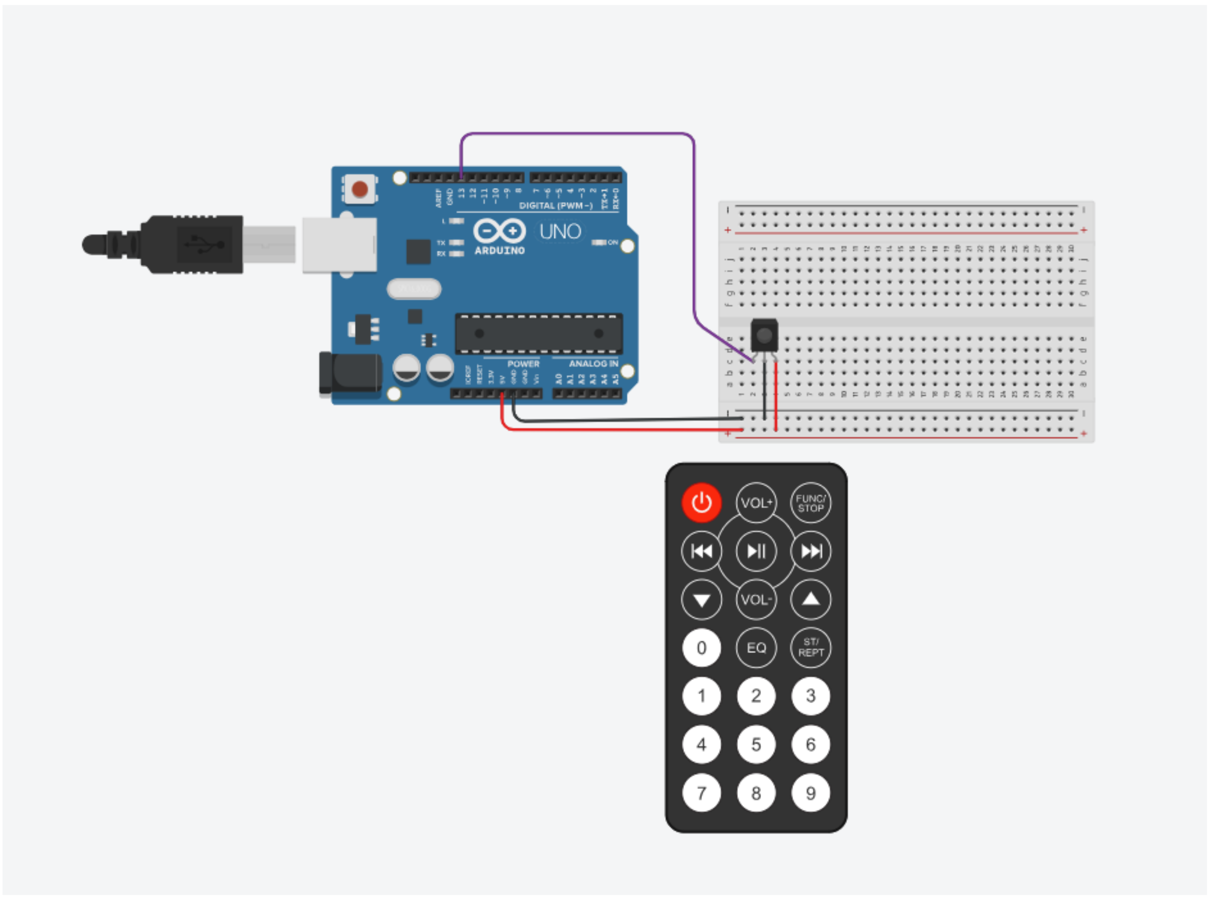
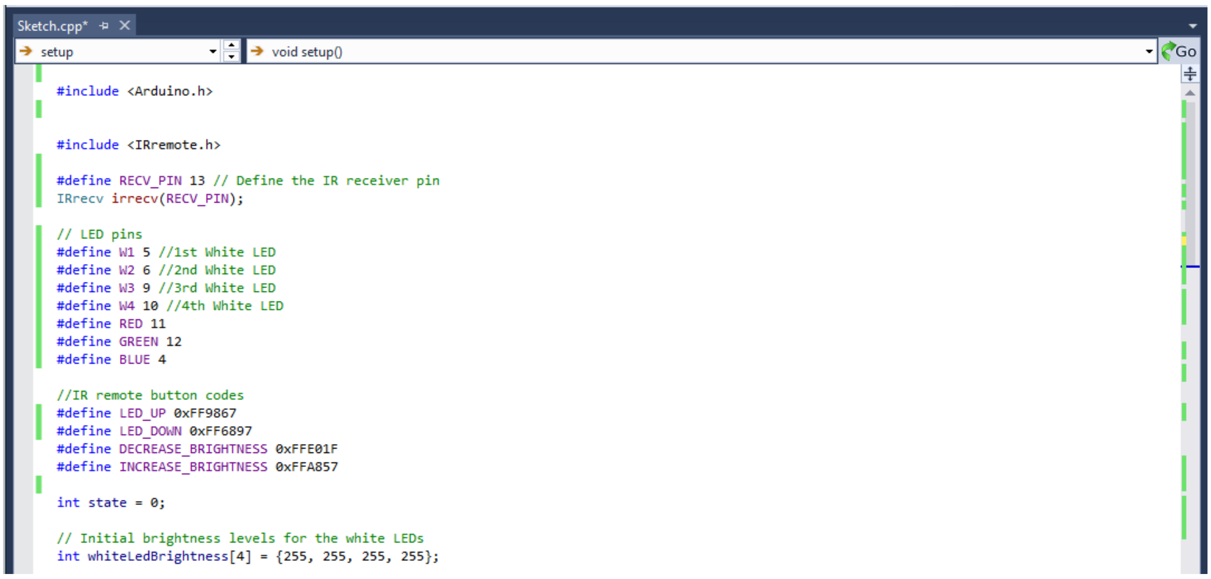
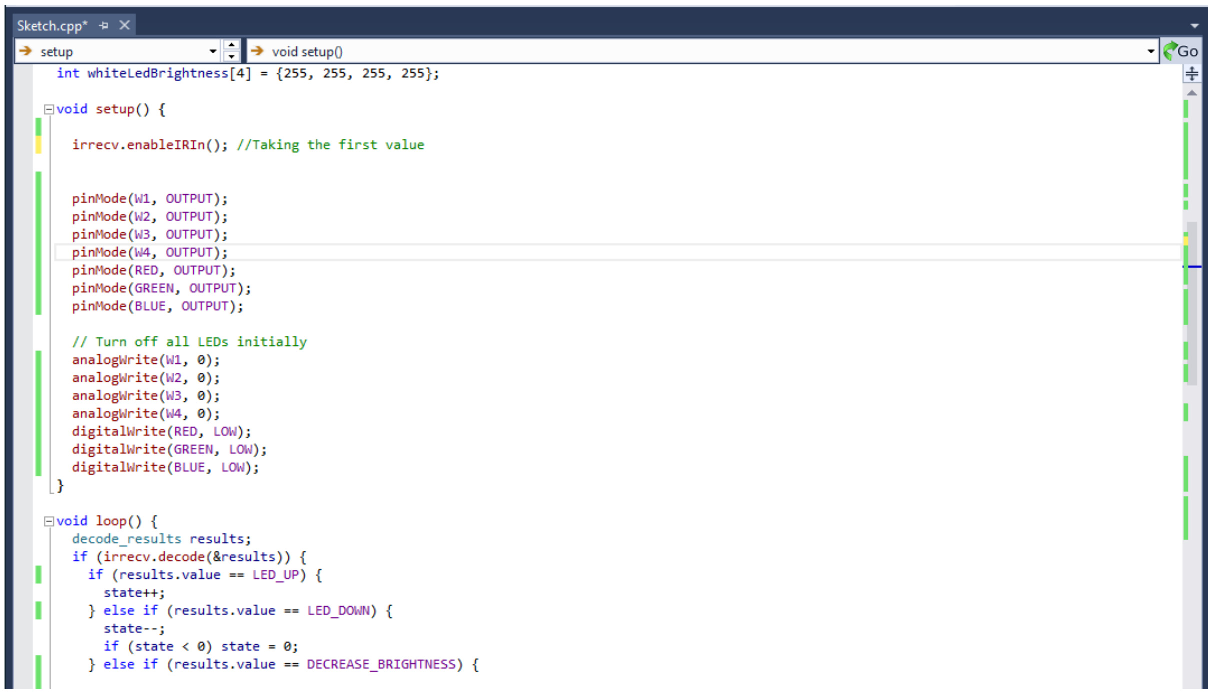


Figure 3: Circuit diagram of the decoding hexa values.

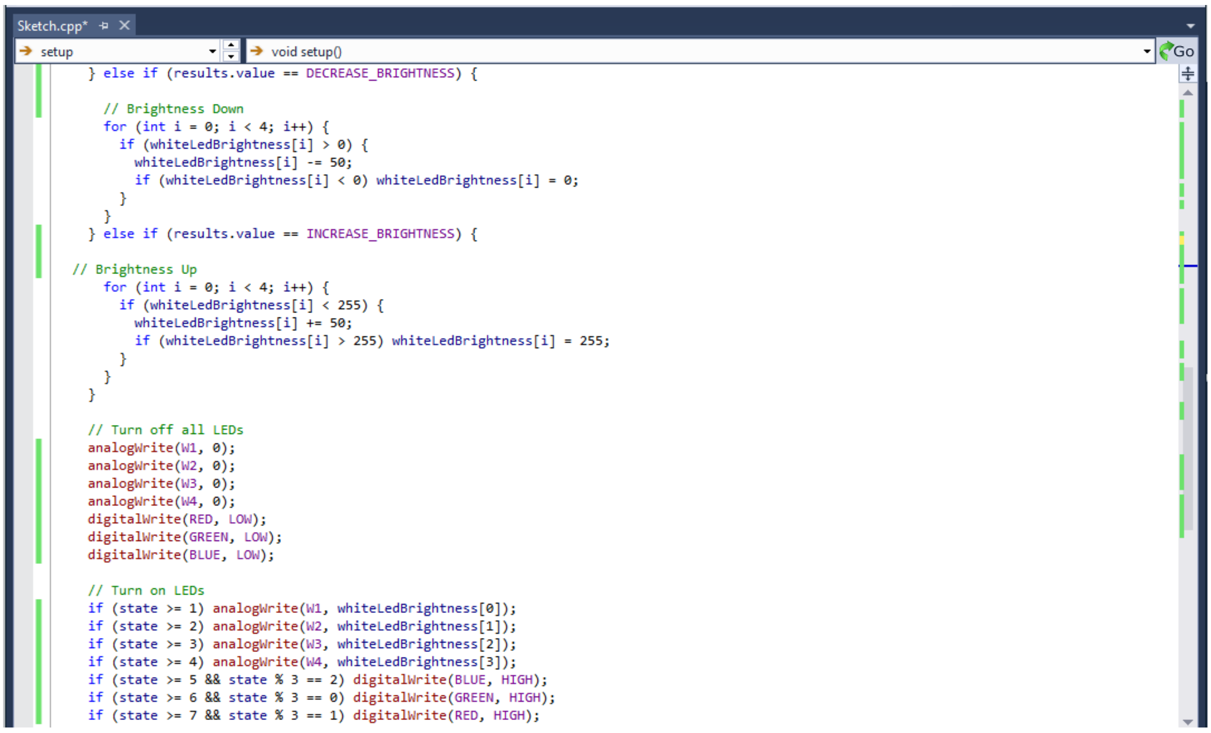
3.2. Source code



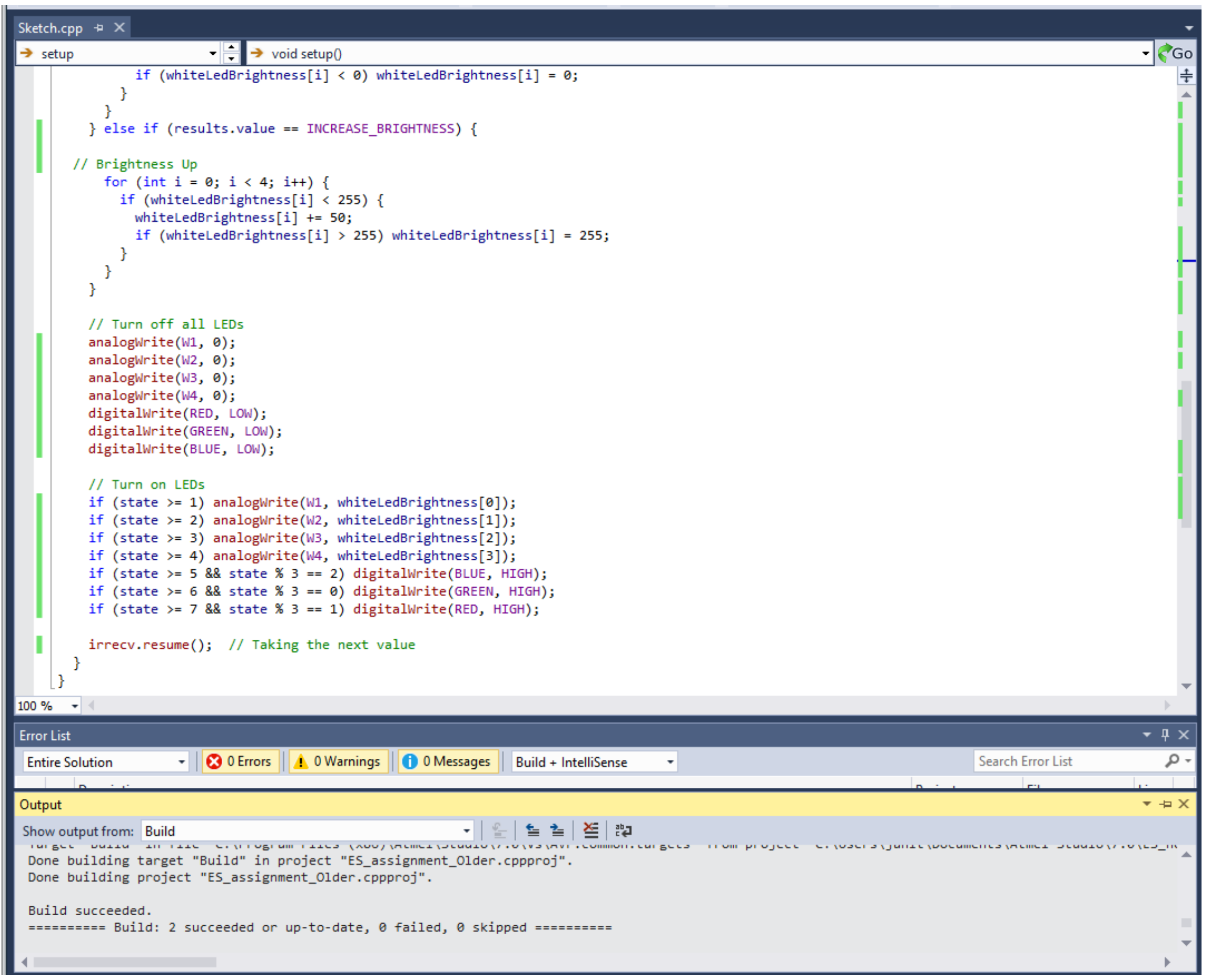
```
#include <Arduino.h>


#include <IRremote.h>

#define RECV_PIN 13 // Define the IR receiver pin
IRrecv irrecv(RECV_PIN);

// LED pins
#define W1 5 //1st White LED
#define W2 6 //2nd White LED
#define W3 9 //3rd White LED
#define W4 10 //4th White LED
#define RED 11
#define GREEN 12
#define BLUE 4

//IR remote button codes
#define LED_UP 0xFF9867
#define LED_DOWN 0xFF6897
#define DECREASE_BRIGHTNESS 0xFFE01F
#define INCREASE_BRIGHTNESS 0xFFA857

int state = 0;

// Initial brightness levels for the white LEDs
int whiteLedBrightness[4] = {255, 255, 255, 255};
```



```
int whiteLedBrightness[4] = {255, 255, 255, 255};

void setup() {

  irrecv.enableIRIn(); //Taking the first value


  pinMode(W1, OUTPUT);
  pinMode(W2, OUTPUT);
  pinMode(W3, OUTPUT);
  pinMode(W4, OUTPUT);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);

  // Turn off all LEDs initially
  analogWrite(W1, 0);
  analogWrite(W2, 0);
  analogWrite(W3, 0);
  analogWrite(W4, 0);
  digitalWrite(RED, LOW);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
}

void loop() {
  decode_results results;
  if (irrecv.decode(&results)) {
    if (results.value == LED_UP) {
      state++;
    } else if (results.value == LED_DOWN) {
      state--;
      if (state < 0) state = 0;
    } else if (results.value == DECREASE_BRIGHTNESS) {
```

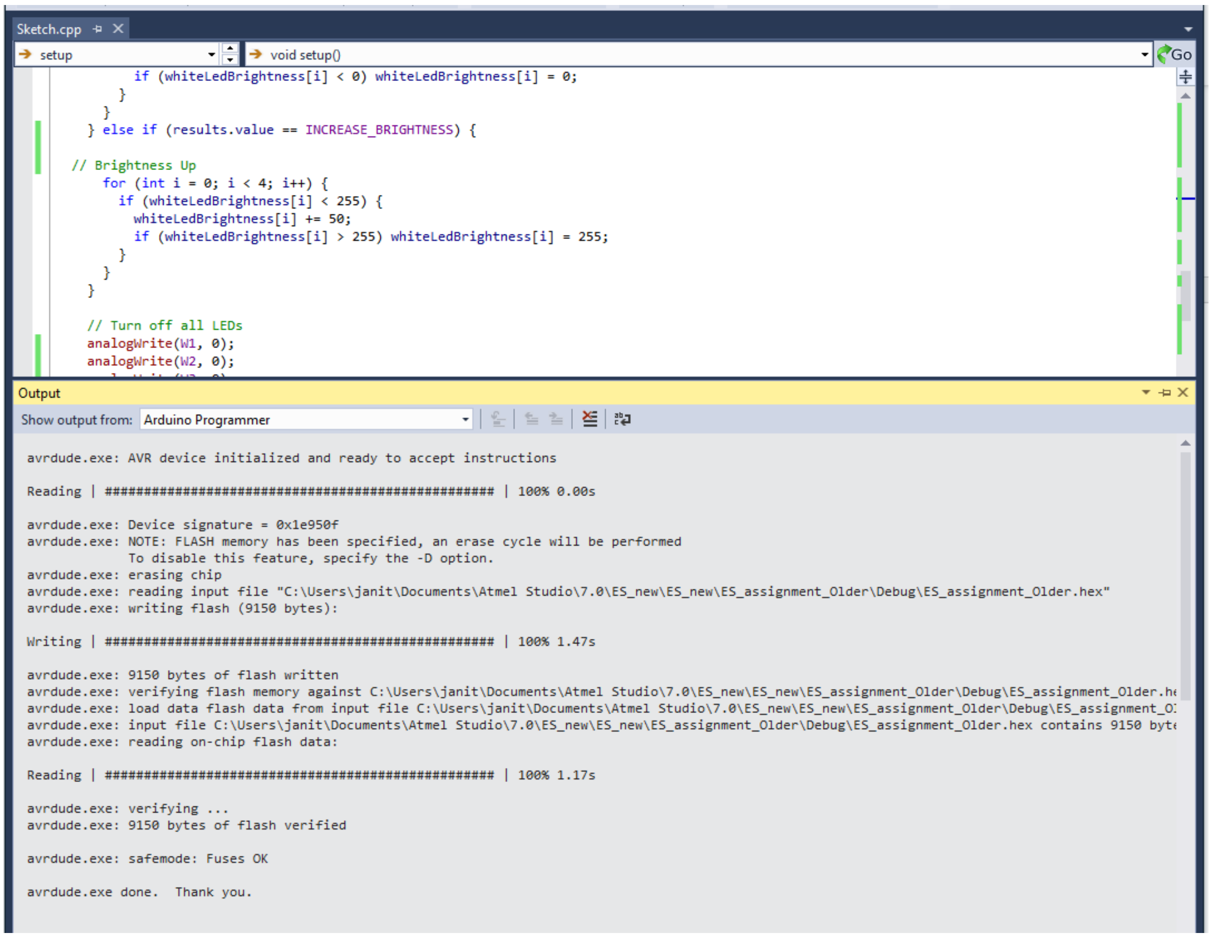


```
    } else if (results.value == DECREASE_BRIGHTNESS) {

      // Brightness Down
      for (int i = 0; i < 4; i++) {
        if (whiteLedBrightness[i] > 0) {
          whiteLedBrightness[i] -= 50;
          if (whiteLedBrightness[i] < 0) whiteLedBrightness[i] = 0;
        }
      }
    } else if (results.value == INCREASE_BRIGHTNESS) {

  // Brightness Up
      for (int i = 0; i < 4; i++) {
        if (whiteLedBrightness[i] < 255) {
          whiteLedBrightness[i] += 50;
          if (whiteLedBrightness[i] > 255) whiteLedBrightness[i] = 255;
        }
      }
    }

    // Turn off all LEDs
    analogWrite(W1, 0);
    analogWrite(W2, 0);
    analogWrite(W3, 0);
    analogWrite(W4, 0);
    digitalWrite(RED, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(BLUE, LOW);

    // Turn on LEDs
    if (state >= 1) analogWrite(W1, whiteLedBrightness[0]);
    if (state >= 2) analogWrite(W2, whiteLedBrightness[1]);
    if (state >= 3) analogWrite(W3, whiteLedBrightness[2]);
    if (state >= 4) analogWrite(W4, whiteLedBrightness[3]);
    if (state >= 5 && state % 3 == 2) digitalWrite(BLUE, HIGH);
    if (state >= 6 && state % 3 == 0) digitalWrite(GREEN, HIGH);
    if (state >= 7 && state % 3 == 1) digitalWrite(RED, HIGH);
```



```
            if (whiteLedBrightness[i] < 0) whiteLedBrightness[i] = 0;
          }
        }
      } else if (results.value == INCREASE_BRIGHTNESS) {

    // Brightness Up
        for (int i = 0; i < 4; i++) {
          if (whiteLedBrightness[i] < 255) {
            whiteLedBrightness[i] += 50;
            if (whiteLedBrightness[i] > 255) whiteLedBrightness[i] = 255;
          }
        }
      }

      // Turn off all LEDs
      analogWrite(W1, 0);
      analogWrite(W2, 0);
      analogWrite(W3, 0);
      analogWrite(W4, 0);
      digitalWrite(RED, LOW);
      digitalWrite(GREEN, LOW);
      digitalWrite(BLUE, LOW);

      // Turn on LEDs
      if (state >= 1) analogWrite(W1, whiteLedBrightness[0]);
      if (state >= 2) analogWrite(W2, whiteLedBrightness[1]);
      if (state >= 3) analogWrite(W3, whiteLedBrightness[2]);
      if (state >= 4) analogWrite(W4, whiteLedBrightness[3]);
      if (state >= 5 && state % 3 == 2) digitalWrite(BLUE, HIGH);
      if (state >= 6 && state % 3 == 0) digitalWrite(GREEN, HIGH);
      if (state >= 7 && state % 3 == 1) digitalWrite(RED, HIGH);

      irrecv.resume();  // Taking the next value
    }
}
```

```
Done building target "Build" in project "ES_assignment_Older.cppproj".
Done building project "ES_assignment_Older.cppproj".

Build succeeded.
========== Build: 2 succeeded or up-to-date, 0 failed, 0 skipped ==========
```



```
          if (whiteLedBrightness[i] < 0) whiteLedBrightness[i] = 0;
        }
      }
    } else if (results.value == INCREASE_BRIGHTNESS) {

  // Brightness Up
      for (int i = 0; i < 4; i++) {
        if (whiteLedBrightness[i] < 255) {
          whiteLedBrightness[i] += 50;
          if (whiteLedBrightness[i] > 255) whiteLedBrightness[i] = 255;
        }
      }
    }

    // Turn off all LEDs
    analogWrite(W1, 0);
    analogWrite(W2, 0);
```

```
avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude.exe: Device signature = 0x1e950f
avrdude.exe: NOTE: FLASH memory has been specified, an erase cycle will be performed
             To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file "C:\Users\janit\Documents\Atmel Studio\7.0\ES_new\ES_new\ES_assignment_Older\Debug\ES_assignment_Older.hex"
avrdude.exe: writing flash (9150 bytes):

Writing | ################################################## | 100% 1.47s

avrdude.exe: 9150 bytes of flash written
avrdude.exe: verifying flash memory against C:\Users\janit\Documents\Atmel Studio\7.0\ES_new\ES_new\ES_assignment_Older\Debug\ES_assignment_Older.he
avrdude.exe: load data flash data from input file C:\Users\janit\Documents\Atmel Studio\7.0\ES_new\ES_new\ES_assignment_Older\Debug\ES_assignment_O
avrdude.exe: input file C:\Users\janit\Documents\Atmel Studio\7.0\ES_new\ES_new\ES_assignment_Older\Debug\ES_assignment_Older.hex contains 9150 byt
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 1.17s

avrdude.exe: verifying ...
avrdude.exe: 9150 bytes of flash verified

avrdude.exe: safemode: Fuses OK

avrdude.exe done.  Thank you.
```

4) Circuit diagram

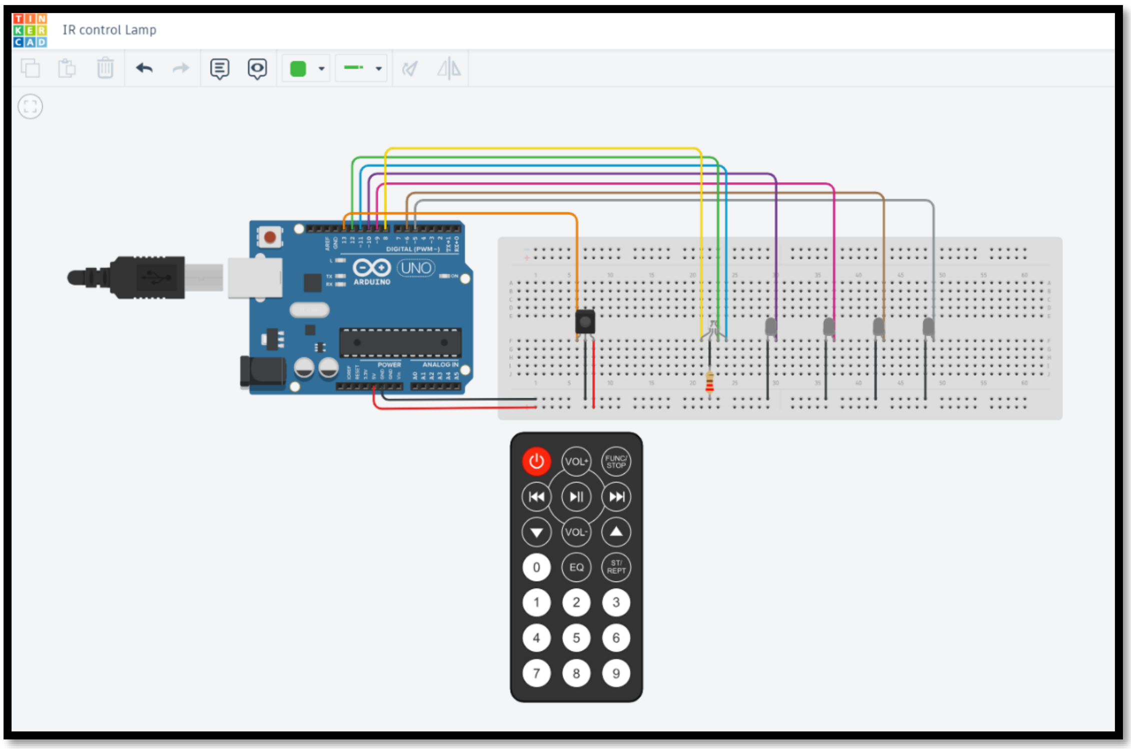


Figure 4: Graphical Circuit Diagram.

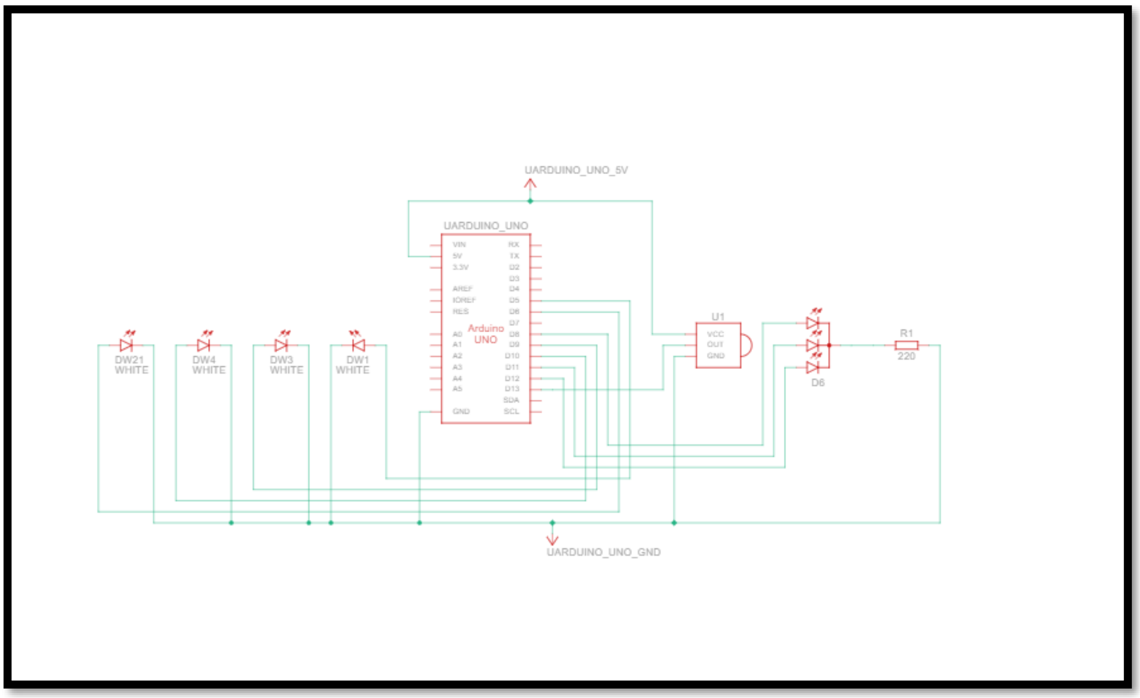


Figure 5: Systematic Circuit diagram

5) Testing and validation

- Component Testing – All components (Arduino Uno board, LEDs, IR receiver, IR remote and jumper wires) were tested individually to verify they are working correctly.
- Circuit Testing – After constructing the circuit, I tested in few ways to ensure that the circuit was correctly wired and that there were no short circuits. For instance, I checked whether there are any short circuits, using the multimeter.
- Functional Testing – Initially, I built up the circuit with two radio button switches assuming as the volume button up(turn on) and volume down button(turn off) for turn on LEDs and turn off LEDs. After successfully completed the first stage, I added the IR receiver and IR remote to do the same functionality. In the final stage, I added the brightness control feature by using next button(brightness up) and previous button(brightness down) on the IR remote.
- Range Testing – In tested the range of the IR remote to determine how far away it could be used to control the lap. The IR remote was worked properly in between 3 meters range. The detection angle of the IR sensor was around 35 degrees.
- User Testing – The lamp was tested by multiple users to collect their feedbacks on its usability and functionality.

6) Discussion

The lamp's performance was generally reliable within the effective range of the IR remote and could be improved by using a more powerful IR transmitter or more sensitive receiver. The lamp is user-friendly, easy to control. However, the remote and receiver should be in one line without any barriers. Therefore, I decided to use radio frequency or Bluetooth for control the lamp in the future, for improve the user experience. The lamp is powered by a external power supply unit and I hope to convert it into a portable device by adding a rechargeable battery.

7) References

1) Back-End, R. [@RoboticsBackEnd]. (2022, February 28). *Arduino - turn LED on and off with Push Button*. Youtube. https://www.youtube.com/watch?v=ZoaUlquC6x8
2) *Decode IR Remote control signals of any Remote using arduino*. (n.d.). Projecthub.Arduino.Cc. Retrieved May 4, 2024, from https://projecthub.arduino.cc/agarwalkrishna3009/decode-ir-remote-control-signals-of-any-remote-using-arduino-9b8e30

3) Logs, A. [@AhmadLogs]. (2023, May 21). *How To Use Infrared Remote using Arduino - new method*. Youtube. https://www.youtube.com/watch?v=m-CE0_o_epc

4) Luna, B. (2022, April 28). *The best guide to validation testing*. Qubika. https://qubika.com/blog/validation-testing/

5) (N.d.). Jaspersoft.com. Retrieved May 4, 2024, from https://www.jaspersoft.com/articles/how-to-create-a-report