Department of Electronic and Telecommunication Engineering

University of Moratuwa

Timer Based Aquarium Light

Yalegama M.M.O.A.B : 200740V

This report is submitted as a partial fulfilment of the module

EN2160 - Electronic Design Realization

26th July 2023

Contents

Abstract

The aim of this project was to create a timer-based aquarium light which is a device that can be used in fish tanks to light the fish tank for a specified period. The user has 3 options to select from for the period with the added feature of being able to turn it off and on as normal use. The project utilized the ATmega328p microcontroller and the Arduino Relay module to deliver a switching action based on time. The final design of the product operated at a voltage of 12V and was suitable for use with aquarium lights with rated operating voltage of 12V.

1. Introduction

1.1 Problem Description and Overview

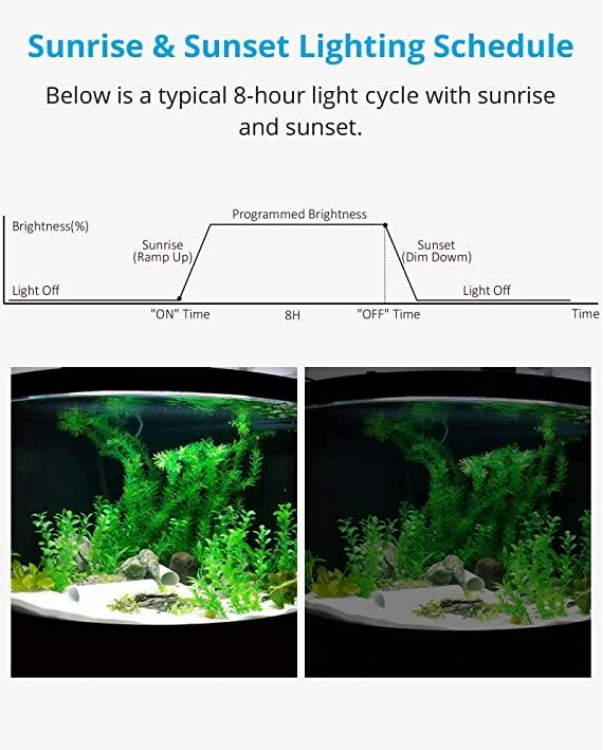
Contrary to popular belief, fish keeping can be a very delicate task in terms of maintenance. Ensure the fish and plants in the aquarium tank get the proper amount of nutrition and exposure to light is critical. The amount of light especially can be a rather tedious to control since it's a daily activity that requires careful timing. The time of exposure to light also depends on several factors such as the species of fish and the amount of ambient light. However, it typically ranges between 8 and 12 hours per day.

This problem is addressed by creating an aquarium light based on a timer that will turn off after one of three specified times, namely, 8, 10, and 12. This is indicated by 3 led lights on the enclosure.

1.2 Functionality and key features

The product consists of an aquarium light that turns off after a set interval of time. The timer options are selected according to the requirements of fish and plant life to thrive. The device can be used in medium sized tanks and light should be attached to the rood of the fish tank. The recommended timer option for a medium sized fish tank is 8 hours and should be turned on between sun rise and sun set.

Reset feature – By holding the on/off button for a 5 second duration it resets any timer feature that had been turned on at that time.



1.3 Specifications

Size – 12 cm length x 7 cm width x 50 cm height

Weight – less than 100g

Enclosure Colour – Black

LED Colours – Green, Yellow and Red and red to indicate 8 hours, 10 hours and 12 hours respectively.

Operation Mode - On or Off

Operating Voltage - Power supply should be connected to 240V / 60Hz supply. LED lights operate at 12V DC.

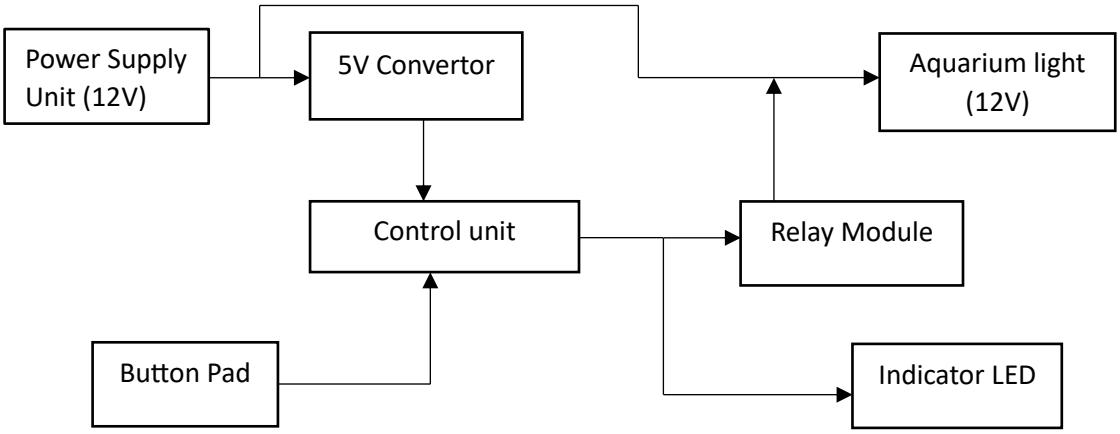
Timing Options - Three sets of timers. 8H, 10H and 12H.

Additional features –

- Easy to set up device.
- Simple and easy to use User Interface

2. Implementation

2.1 Functional Block diagram



2.2 Mechanism

The power supply unit will supply a 12 V DC voltage that will be interrupted by a relay that will activated by a control signal that will come from the control unit. The control unit will be given instructions from the user using the button pad. The control unit will also send signals to the indicator LED to show the user which option of timer is being activated at the specific time.

2.3 Component Selection

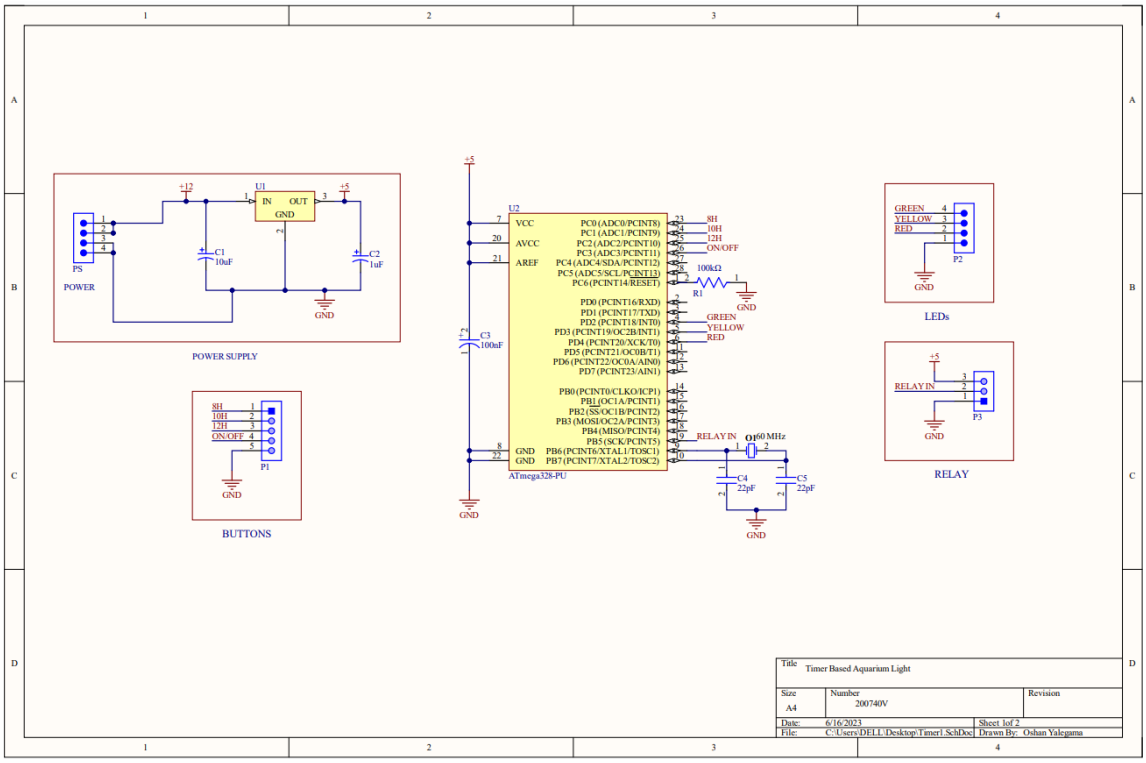
Power Supply Unit : 12 V 2A Power Adaptor

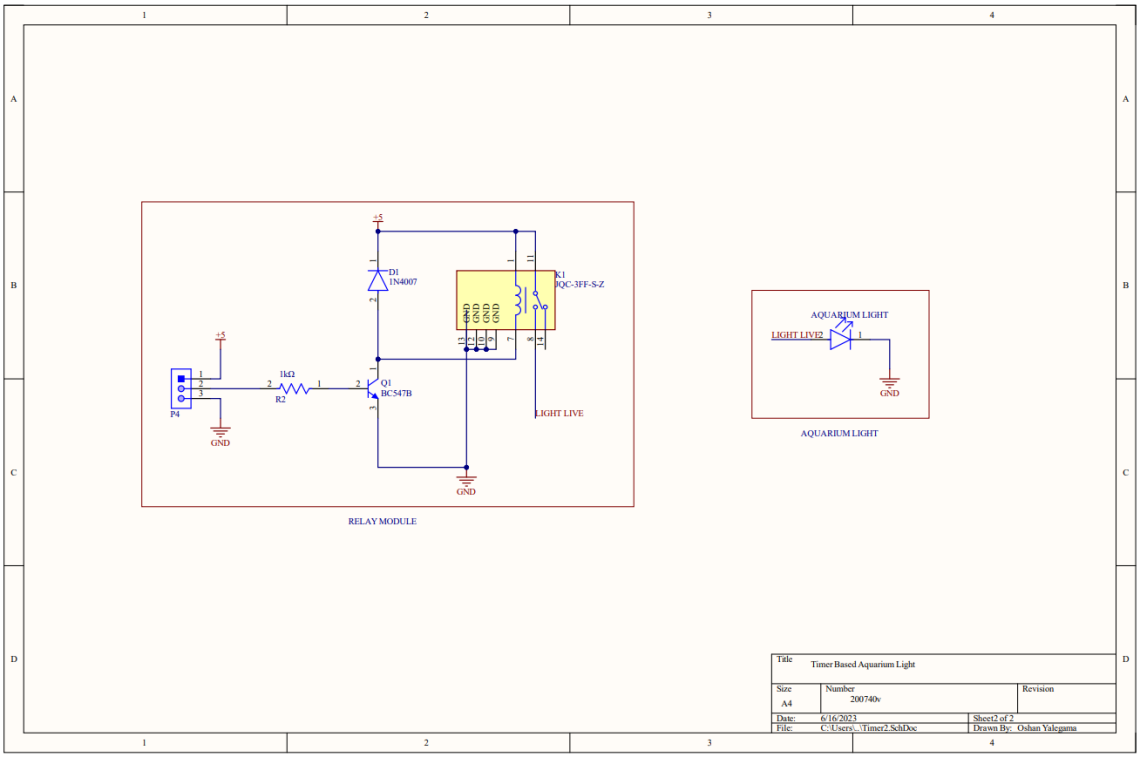
5V Convertor : Capacitors (470 uF, 1 uF), LM7805, Diode IN4007

Control Unit : 22 AWG wire, 220 ohm resistors, 10 uF capacitors, ATMega328P, 16 MHz crustal oscillator

Button Pad : 4 push buttons

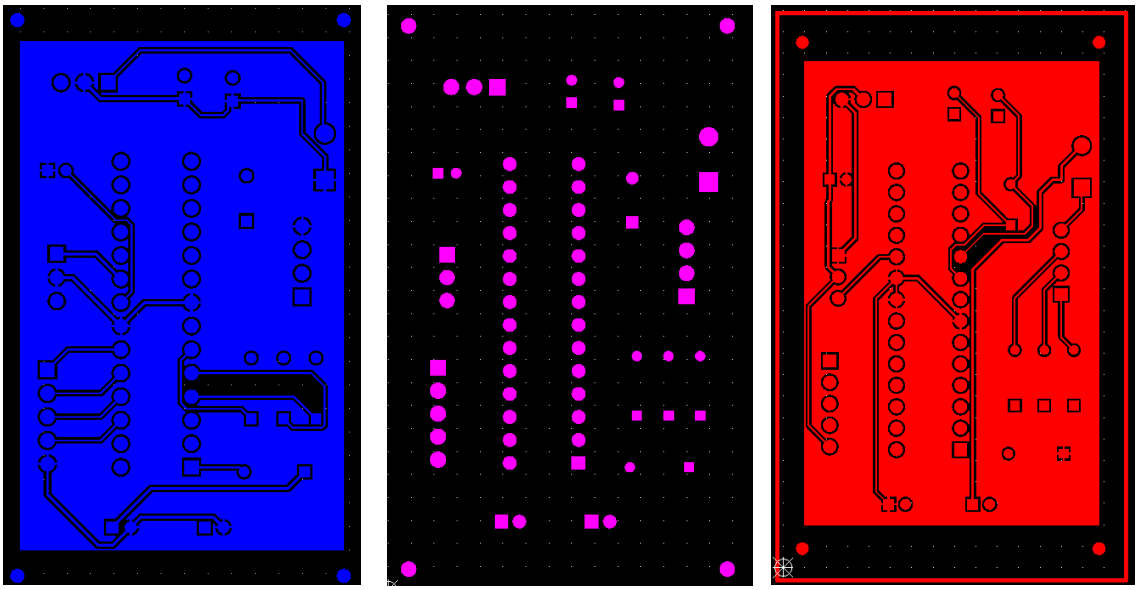
2.4 Schematic Diagram

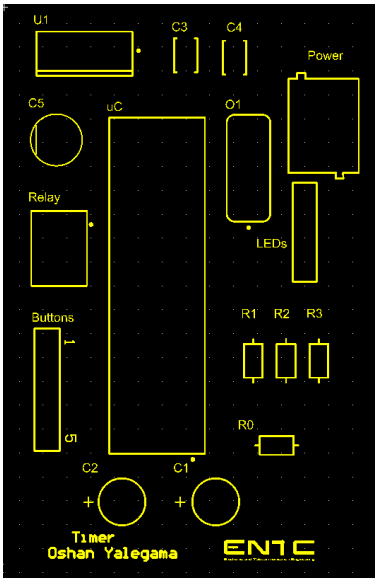




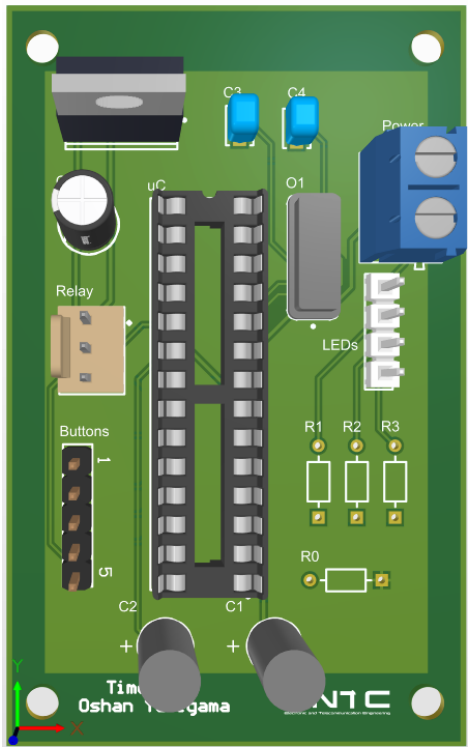
2.5 PCB Design

The final PCB design files sent for manufacturing are as follows.




U1
C3
C4
Power
C5
uC
O1
Relay
LEDs
Buttons
R1
R2
R3
R0
C2
C1
Timer
Oshan Yalegama

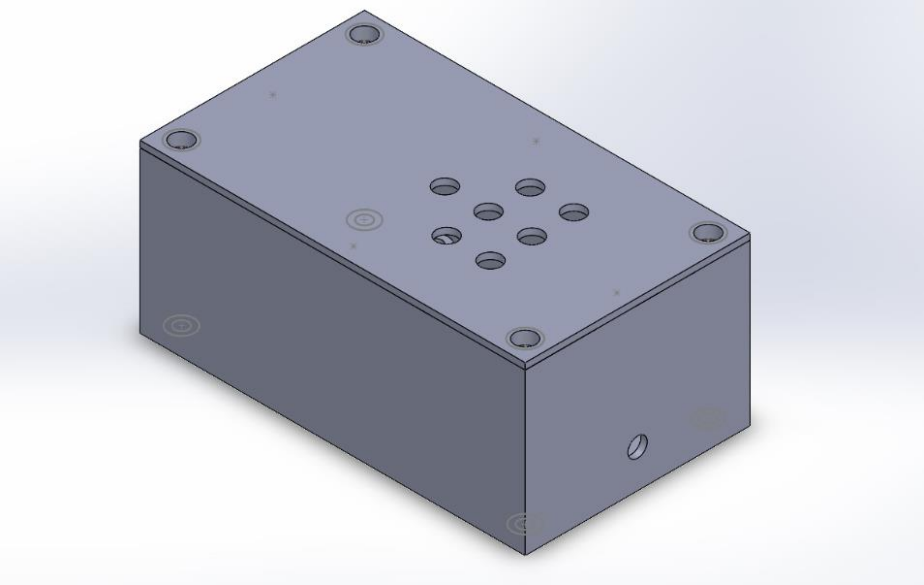
2.6 Final PCB

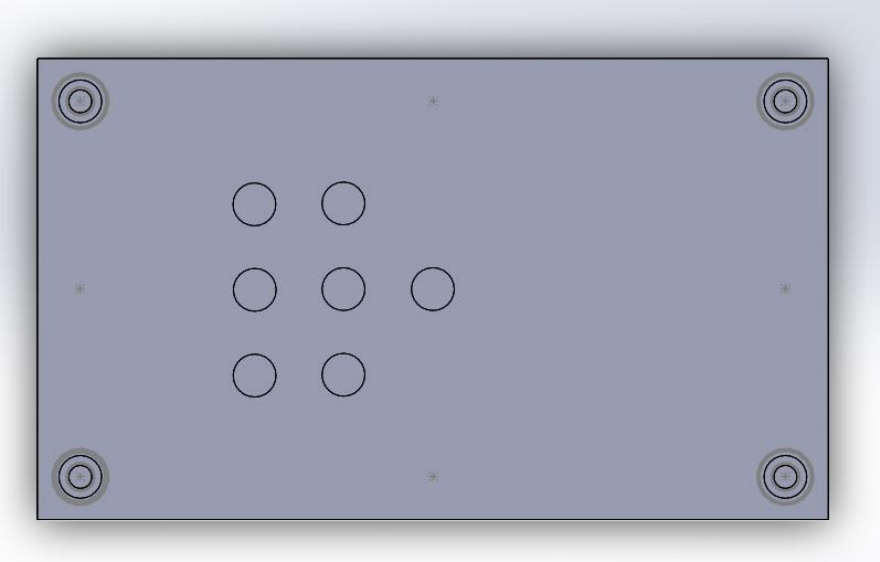

C3
C4
uC
O1
Relay
LEDs
Buttons
R1
R2
R3
R0
C2
C1

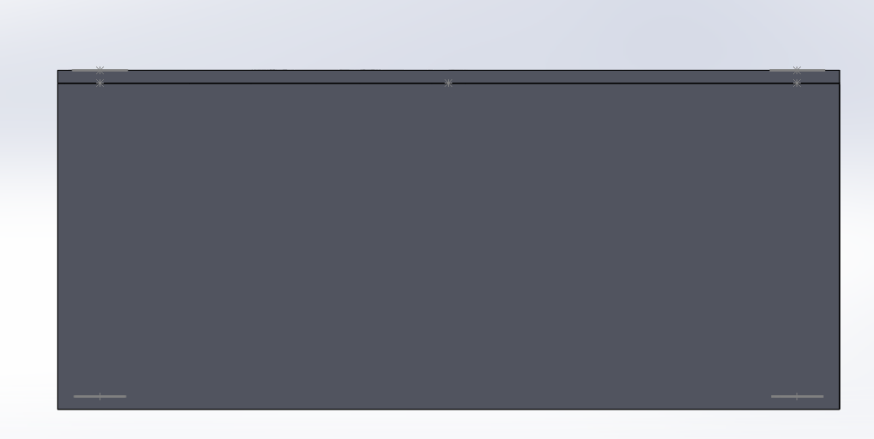
3. Enclosure

3.1 Enclosure Design

The enclosure was designed on Solidworks and was 3d printed by a local 3d printing shop.







3.2 The 3D Printed Enclosure

4. Assembly Instructions

4.1 Components required

12 V 2A Power Adaptor

Capacitors (470 uF, 1 uF)

LM7805 voltage regulator

Diode IN4007

22 AWG wire

220 ohm resistors

10 uF capacitors

ATMega328P

16 MHz crustal oscillator

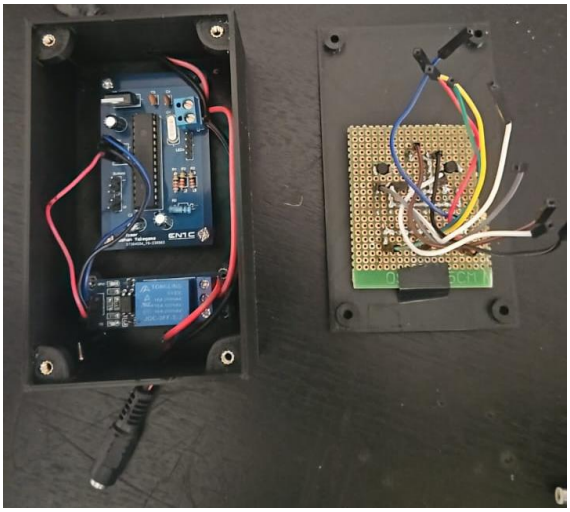
4 push buttons

4 indicator LEDs

4.2 Parts Assembly

All components are through hole and hence should be soldered as appropriate.

The PCB and Relay Module should be mounted to the platforms on the enclosure using 3mm screws and 2mm screws respectively. The button pad should be mounted to the enclosure lid using the supports printed. This is demonstrated by the following image.



The PCB and Relay orientations are as in the image.

The enclosure should be closed with the lid and tightened using 3mm screws.

5. Bill of Materials

The following list has been compiled with an assumed USD conversion rate of Rs. 332.

Component	Supplier	Unit Price	Quantity	Total Price(LKR)
Atmega 328P	LCSC Electronics	4.98 USD	2	1653.36
IC Socket	LCSC Electronics	0.0464 USD	5	77.024
Crystal oscillator	LCSC Electronics	0.1126 USD	2	74.7664
10uF capacitor	LCSC Electronics	0.043 USD	10	142.76
22pF capacitor	LCSC Electronics	0.01 USD	20	33.2
10k ohm resistor	LCSC Electronics	0.086 USD	10	285.22
220 ohm resistor	LCSC Electronics	0.086 USD	20	570.44
LM7805	LCSC Electronics	0.178 USD	2	118.192
LED	LCSC Electronics	0.138 USD	10	458.16
Arudino Relay Module	Tronic.lk	180 LKR	2	360
12V Power Adapter	Tronic.lk	1100 LKR	1	1100
PCB	JLCPCB	7.42 USD	1	2463.44
Enclosure	Xyder Labs	4800 LKR	1	4800
Total	**Rs. 12,136. 5924**			

6. Tests for Functionality

Power delivery test.

A dc voltage of 12 V should be delivered to the output connector when the device is in timer mode and should have 0 V otherwise. A dc voltage of 12 V should be supplied to the input of the PCB, and this can be checked by probing the input terminal block. If not, there might be a lose connection inside the enclosure.

Relay module operation

The operation of the Relay module can be tested by checking whether the green indicator light is on when the device is not in timer mode.

Button pad operation

The buttons can be tested by simply testing whether each button activates the necessary timing of the device. For this, it must be ensured that the microcontroller is properly operating. The on off button should also provide properly resetting the device's timing configuration.

7. References

How Long Should You Leave a Aquarium Lights On? - AquariumNexus

(20) Arduino Timer Control Relay Devices - YouTube

Building an Arduino on a Breadboard | Arduino Documentation | Arduino Documentation

220V TO 12V DC CONVERTER CIRCUIT DIAGRAM - Electronic Projects, Power Supply Circuits, Circuit Diagram symbols, Audio Amplifier Circuit pdf & Engineering Projects (circuitsarena.com)

LM7805 * LM7806 * LM7808 * LM7809 * LM7810 * LM7812 * LM7815 * LM7818 * LM7824 * LM7805A * LM7806A * LM7808A *LM7809A * LM7810A * LM7812A * LM7815A * LM7818A * LM7824A 3-Terminal 1A Positive Voltage Regulator (Preliminary) (components101.com)

Appendix

The following code was used to program the microcontroller using the Arduino platform and using the Arduino IDE.

```
//OUTPUT PINS

int relayPin = 13;  // The pin we're using

int ledPin1 = 2; // red(button 1)

int ledPin2 = 3; // yellow(button 2)

int ledPin3 = 4; // green(button 3)

//DURATIONS

long duration1 = 10000;  // The duration for button 1 (1 minute)

long duration2 = 20000;  // The duration for button 2 (5 minutes)

long duration3 = 30000;  // The duration for button 3 (30 minutes)

//BUTTONS

int buttonPin1 = A0;  // The pin for button 1

int buttonPin2 = A1;  // The pin for button 2

int buttonPin3 = A2;  // The pin for button 3

int buttonPin4 = A3; //the toggle pin

//BUTTON STATES

bool buttonState1 = false;  // The state of button 1 (pressed or not pressed)//A0,23, 8H

bool buttonState2 = false;  // The state of button 2 (pressed or not pressed)//A1,24

bool buttonState3 = false;  // The state of button 3 (pressed or not pressed)//A2.25

bool buttonState4 = false;  // variable for storing the toggle button state

//BUTTON CHARACTERISTICS

unsigned long startTime;  // The time the pin was turned on

unsigned long pressTime;

int resetTime = 5000;

// OTHER VARIABLES

bool reset = false;
```

```
int ledState = LOW; //variable for storing the led state


//SETUP
void setup() {
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(relayPin, OUTPUT);
  pinMode(buttonPin1, INPUT_PULLUP);  // Set the button 1 pin as an input with pullup resistor
  pinMode(buttonPin2, INPUT_PULLUP);  // Set the button 2 pin as an input with pullup resistor
  pinMode(buttonPin3, INPUT_PULLUP);  // Set the button 3 pin as an input with pullup resistor
  pinMode(buttonPin4, INPUT_PULLUP); // Set the button 4 pin as an input with pullup resistor
  Serial.begin(9600);  // Initialize serial communication
}


//CODE
void loop() {
  //READING THE STATE OF THE BUTTONS
  buttonState1 = digitalRead(buttonPin1);  // Read the state of button 1
  buttonState2 = digitalRead(buttonPin2);  // Read the state of button 2
  buttonState3 = digitalRead(buttonPin3);  // Read the state of button 3
  buttonState4 = digitalRead(buttonPin4); // Read the state of button 3


  //TOGGLING
  if (buttonState4== LOW) {
    Serial.println("yeah");
    ledState = !ledState;
    digitalWrite(relayPin, ledState);
    delay(500);  // Debounce delay to prevent rapid toggling
```

```
}

//PRESSING BUTTON 1
if (buttonState1 == LOW) {  // If button 1 is pressed
 Serial.println("yeah");
 digitalWrite(ledPin1, HIGH); //turn led1 on
 digitalWrite(relayPin, HIGH);//turn relay on
 startTime = millis();  // Store the time the pin was turned on
 while (millis() - startTime <= duration1) {  // While relay is on for less than duration1 time
  //RESET Code
  buttonState4 = digitalRead(buttonPin4);//check the toggle button
  if (buttonState4== LOW){//if the toggle button is pressed
   pressTime = millis();
   reset = true;
   while (millis() - pressTime <= resetTime){//if it's released before the reset time
    buttonState4= digitalRead(buttonPin4);
    if (buttonState4== HIGH){
     reset = false;//don't reset
    }
   }
   if (reset){
    Serial.println("Stopped");
    break;
   }
  }
 }
 digitalWrite(ledPin1, LOW); //Turn led off
 digitalWrite(relayPin, LOW);  // Turn the relay off
 delay(1000);  // Wait for a second before starting again
```

```
}

if (buttonState2 == LOW) {  // If button 1 is pressed
  digitalWrite(ledPin2, HIGH);
  digitalWrite(relayPin, HIGH);  // Turn the pin on
  startTime = millis();  // Store the time the pin was turned on
  while (millis() - startTime <= duration2) {  // While the pin should be on
    //COLOUR CHANGING CODE
    if (millis() - startTime >= (duration2- duration1)) {
      digitalWrite(ledPin2, LOW);
      digitalWrite(ledPin1, HIGH);
    //RESET Code
    buttonState4= digitalRead(buttonPin4);
    if (buttonState4== LOW){
      pressTime = millis();
      reset = true;
      while (millis() - pressTime <= resetTime){
        buttonState4= digitalRead(buttonPin4);
        if (buttonState4== HIGH){
          reset = false;
        }
      }
      if (reset){
        Serial.println("Stopped");
        break;
      }
    }
    delay(1000);  // Wait for one second
   }
```

```
  }
  digitalWrite(ledPin2, LOW);
  digitalWrite(ledPin1, LOW);
  digitalWrite(relayPin, LOW);  // Turn the relay off
  delay(1000);  // Wait for a second before starting again
}

if (buttonState3 == LOW) {  // If button 1 is pressed
  digitalWrite(ledPin3, HIGH);
  digitalWrite(relayPin, HIGH);  // Turn the pin on
  startTime = millis();  // Store the time the pin was turned on
  while (millis() - startTime <= duration3) {  // While the pin should be on
    //COLOUR CHANGING CODE
    if (millis() - startTime >= (duration3-duration2)) {
      if (millis() - startTime >= (duration3 - duration1)) {
        digitalWrite(ledPin2, LOW);
        digitalWrite(ledPin1, HIGH);
      }
      else{
        digitalWrite(ledPin3, LOW);
        digitalWrite(ledPin2, HIGH);
      }
    }
    //RESET Code
    buttonState4= digitalRead(buttonPin4);
    if (buttonState4== LOW){
      pressTime = millis();
      reset = true;
      while (millis() - pressTime <= resetTime){
```

```
        buttonState4= digitalRead(buttonPin4);
        if (buttonState4== HIGH){
          reset = false;
        }
       }
       if (reset){
        Serial.println("Stopped");
        break;
       }
      }
     delay(1000);  // Wait for one second
    }
    digitalWrite(ledPin3, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin1, LOW);
    digitalWrite(relayPin, LOW);
    delay(1000);  // Wait for a second before starting again
  }
}
```