**Experiment No. 5**

**Title:** Write a program using Arduino to control LED (One or more ON/OFF).

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-**

void setup()

{

pinMode(13, OUTPUT);

}

void loop()

{

digitalWrite(13, HIGH);

delay(1000); // Wait for 1000 millisecond(s) digitalWrite(13, LOW);

delay(1000); // Wait for 1000 millisecond(s)

}

**Circuit Diagram:-**

A diagram of a circuit

Description automatically generated with low confidence

**Output:-**

A picture containing electronic engineering, electronics, circuit component, electronic component

Description automatically generatedA picture containing electronic engineering, electronics, circuit component, electronic component

Description automatically generated

**Experiment No. 6**

**Title:** Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is

greater than 200.

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

# Program:-

int counter = 0; void setup()

{

Serial.begin(9600); pinMode(7,OUTPUT); pinMode(8,OUTPUT); pinMode(9,OUTPUT);

}

void loop() { if(counter == 250)

{

counter=0;

}

if(counter < 250)

{

Serial.println(counter);

}

counter = counter + 1; delay(100);

if(counter < 100 )

{

digitalWrite(7,LOW); digitalWrite(8,LOW); digitalWrite(9,HIGH);

}

if(counter > 101 && counter < 200 )

{

digitalWrite(7,LOW); digitalWrite(8,HIGH); digitalWrite(9,LOW);

}

if( counter > 200 )

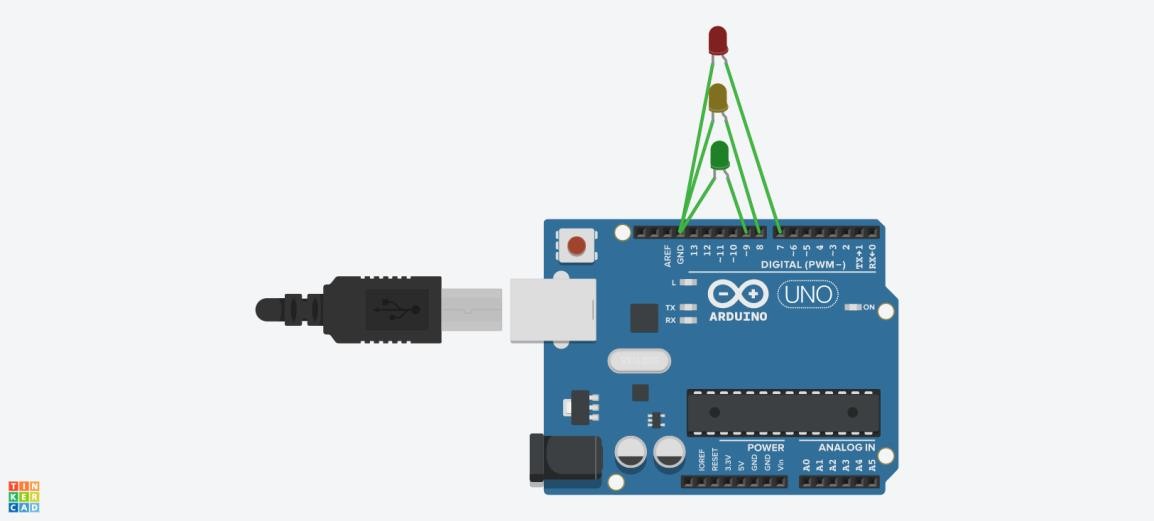
{

digitalWrite(7,HIGH); digitalWrite(8,LOW); digitalWrite(9,LOW);

}

}

# Circuit Diagram:-



**Output:-**

A computer screen shot of a blue circuit board

Description automatically generated with low confidenceA computer screen shot of a blue circuit board

Description automatically generated with low confidenceA computer screen shot of a blue circuit board

Description automatically generated with low confidence

**Experiment No. 7**

**Title:** Create a program so that when the user enters ‘b’ the blue light blinks, ‘g’ the green light

is illuminated ‘y’ the yellow light is illuminated and

‘r’ the red light is illuminated.

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-**

int RED = 13;

int GREEN = 12; int YELLOW = 11; int BLUE = 10;

char ledToBlink; void setup() {

pinMode(RED,OUTPUT); pinMode(YELLOW,OUTPUT); pinMode(BLUE,OUTPUT); pinMode(GREEN,OUTPUT);

Serial.begin(9600);

}

void loop() { int temp;

if (Serial.available() > 0) { ledToBlink = Serial.read();

if ((ledToBlink == 'r') || (ledToBlink == 'R')) { Serial.print("Blinking Red LED\n");

temp = 13;

}

else if((ledToBlink == 'g') || (ledToBlink == 'G')) { Serial.print("Blinking Green LED\n");

temp = 12;

}

else if((ledToBlink == 'y') || (ledToBlink == 'Y')){ Serial.print("Blinking Yellow LED\n");

temp = 11;

}

else if((ledToBlink == 'b') || (ledToBlink == 'B')){ Serial.print("Blinking Blue LED\n");

temp = 10;

}

else{

}

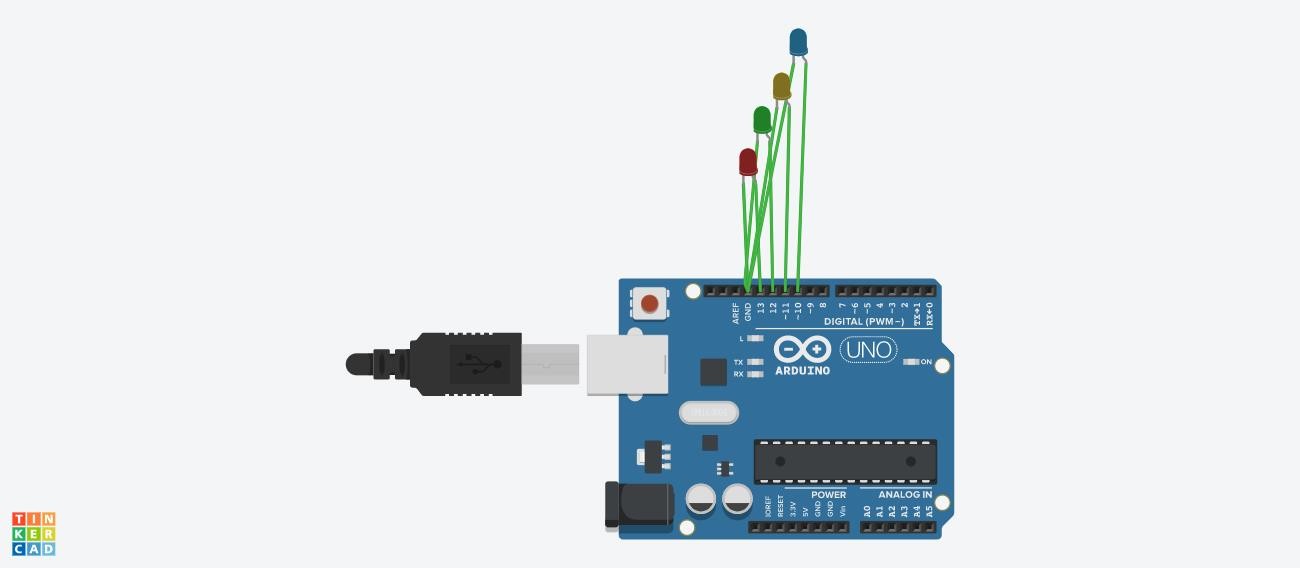
Serial.print("\nInvalid Choice Try Again \n");

digitalWrite(temp, HIGH); delay(1000); digitalWrite(temp, LOW);

}

}

**Circuit Diagram:-**



**Output:-**

A picture containing text, software, diagram, operating system

Description automatically generated

A picture containing text, software, screenshot

Description automatically generated

**Experiment No. 10**

**Title:** Write a program read the temperature

sensor and send the values to the serial monitor on the computer.

Roll No:SA&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-** int baselineTemp = 0; int celsius = 0; int fahrenheit = 0; void setup()

{

pinMode(A2, INPUT);

pinMode(13, OUTPUT); Serial.begin(9600);

} void

loop()

{

celsius = map(((analogRead(A2) - 20) \* 3.04), 0, 1023, -40, 125);

fahrenheit = ((celsius \* 9) / 5 + 32); Serial.print(celsius);

Serial.print(" C, ");

Serial.print(fahrenheit)

; Serial.println(" F"); if (celsius > 40)

{

digitalWrite(13,HIGH);

}

else

{

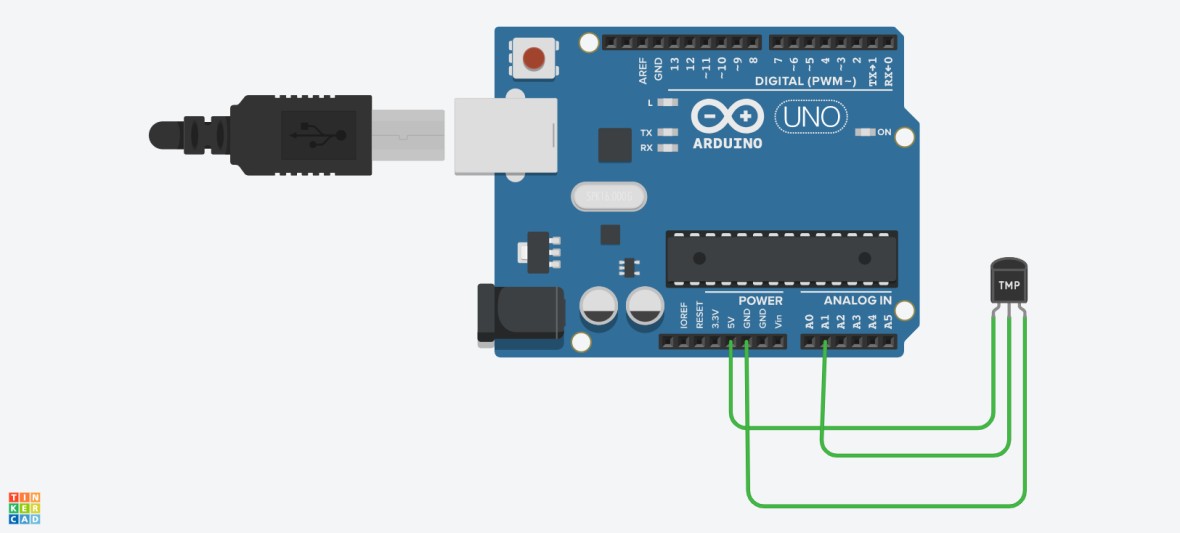
digitalWrite(13,LOW);

}

delay(1000);

}

**Circuit Diagram:-**



**Output:-**

A picture containing text, screenshot, software, multimedia software

Description automatically generated

**Experiment No. 11**

**Title:** Write a program so it displays the temperature

In Fahrenheit as well as the maximum and minimum Temperatures it has seen.

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-**

int val;

int tempPin = 0; float tfmax;

float tfmin = 100; void setup()

{

Serial.begin(9600);

}

void loop()

{

val = analogRead(tempPin); float mv = ( val/1024.0)\*500; float cel = mv;

float farh = (cel\*9)/5 + 32;

if (farh > tfmax)

{

tfmax = farh;

}

if (farh < tfmin)

{

tfmin = farh;

}

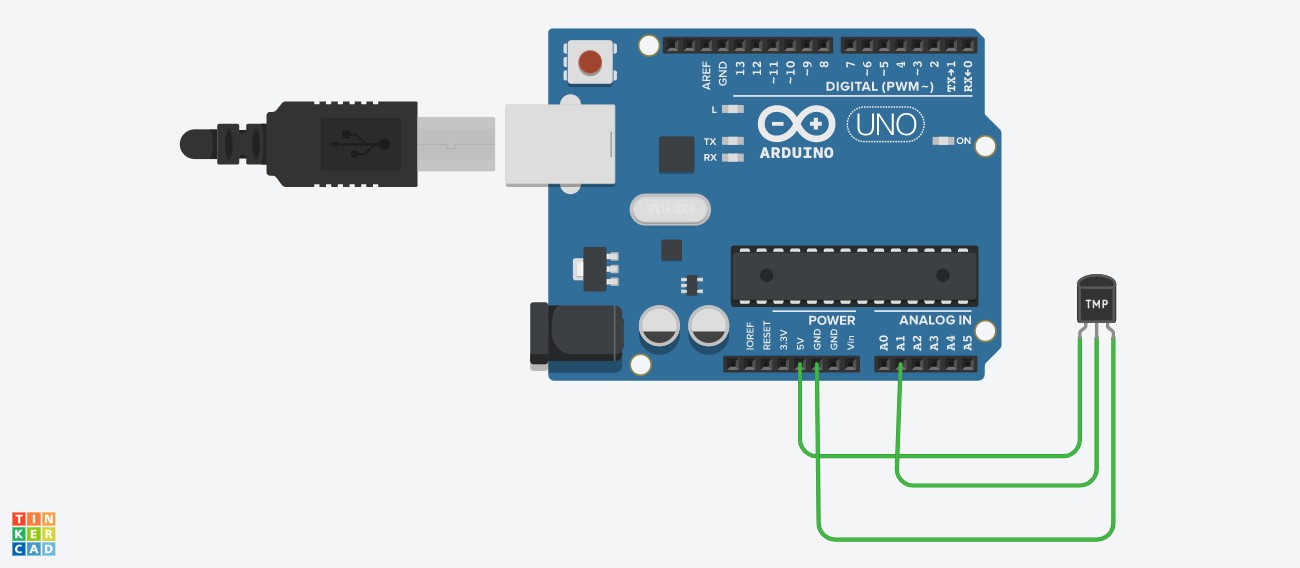
Serial.print("TEMPRATURE = "); Serial.print(farh); Serial.print("\*F");

Serial.println();

Serial.print("Max Temp:"); Serial.print(tfmax); Serial.println(); Serial.print("Min Temp:"); Serial.print(tfmin); Serial.println(); Serial.println(); delay(2000);

}

**Circuit Diagram:-**



**Output:-**

A picture containing text, screenshot, electronic engineering, electronics

Description automatically generated

**Experiment No. 12**

**Title:** Write a program to show the temperature and Shows a graph of the recent measurements.

Roll No: SA&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-**

#include <SoftwareSerial.h> SoftwareSerial bt(8, 9); // RX, TX

#include "dht.h" #define dataPin A0

dht DHT; int i = 0; int temp; int hum;

void setup() {

Serial.begin(9600); bt.begin(9600); Serial.println("Ready");

delay(2000); pinMode(13, OUTPUT);

}

void loop(){

int readData = DHT.read11(dataPin);

hum = DHT.humidity; temp = DHT.temperature;

Serial.println();

Serial.println(hum); Serial.print("Humidity: ");

Serial.print(",");

Serial.print("Temp: "); Serial.println(temp); Serial.print("o"); //degree symbol Serial.print("C ");

Serial.print(",");

bt.print(temp); //send distance to MIT App bt.print(";");

bt.print(hum); //send distance to MIT App bt.println(";");

if(temp> 31)

{

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second

}

else

{

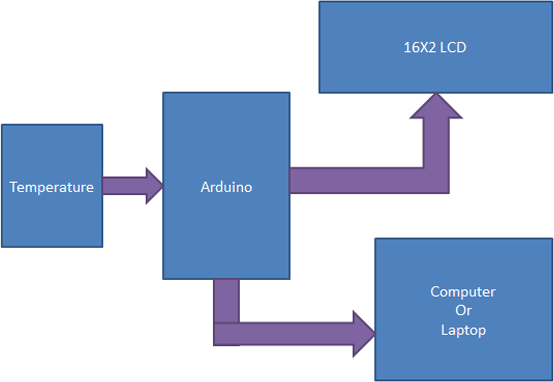
digitalWrite(13, LOW); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second

}

delay(1000);

}

**Diagram:-**



**Experiment No. 14**

**Title:** Understanding the connectivity of Raspberry- Pi /Arduino circuit with IR sensor. Write an Application to detect obstacle and notify user using

LED’s.

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

# Program:-

void setup()

{

pinMode(2, INPUT); pinMode(3,OUTPUT);

}

void loop()

{

digitalWrite(3, LOW); int read=digitalRead(2); if(read == HIGH)

{

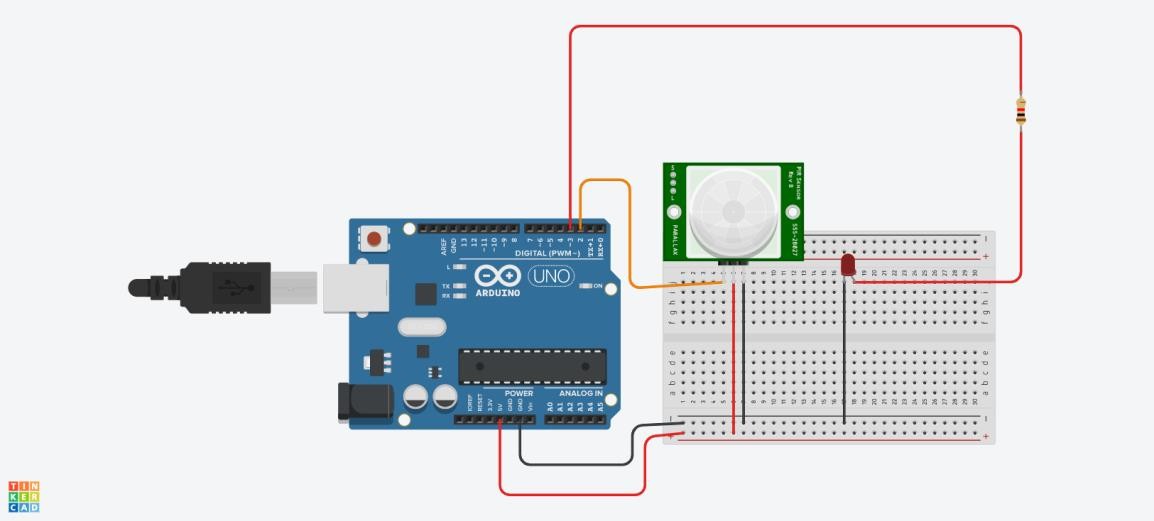
digitalWrite(3,HIGH); delay(1000);

}

delay(1000);

}

# Circuit Diagram:-



**Output:-**

A computer screen shot of a computer

Description automatically generated with low confidence

A computer screen shot of a computer

Description automatically generated with low confidence

**Experiment No. 15**

**Title:** Study of ThingSpeak – an API and Web Service for the Internet of Thing

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

# Program:-

String ssid = "Simulator Wifi"; // SSID to connect to String password = ""; // Our virtual wifi has no password

String host = "api.thingspeak.com"; // Open Weather Map API const int httpPort = 80;

String url = "/update?api\_key=TBRSBCUPJXYW5F7P&field1="; int setupESP8266(void) {

// Start our ESP8266 Serial Communication

Serial.begin(115200); // Serial connection over USB to computer Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266 delay(10); // Wait a little for the ESP to respond

if (!Serial.find("OK ")) return 1;

// Connect to 123D Circuits Simulator Wifi Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\""); delay(10); // Wait a little for the ESP to respond

if (!Serial.find("OK ")) return 2;

// Open TCP connection to the host:

Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort); delay(50); // Wait a little for the ESP to respond

if (!Serial.find("OK ")) return 3; return 0;

}

void anydata(void) {

int temp = map(analogRead(A0),20,358,-40,125);

// Construct our HTTP call

String httpPacket = "GET " + url + String(temp) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";

int length = httpPacket.length();

// Send our message length Serial.print("AT+CIPSEN D="); Serial.println(length);

delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

// Send our http request Serial.print(httpPacket);

delay(10); // Wait a little for the ESP to respond if (!Serial.find("SEND OK\r\n")) return;

}

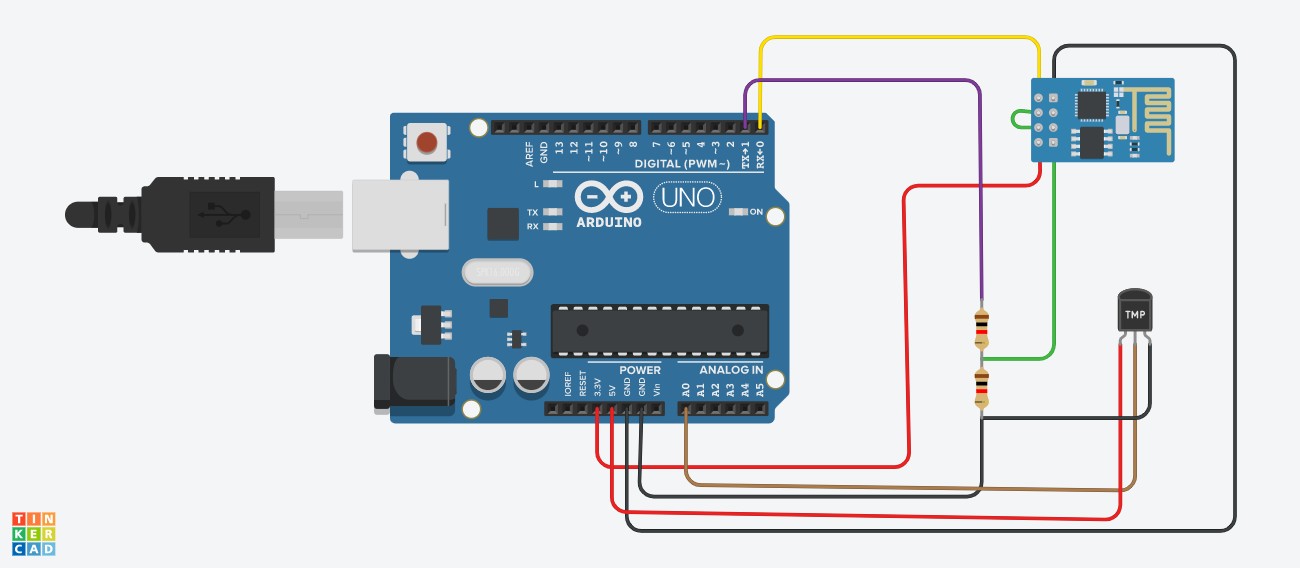
void setup() { setupESP8266();

}

void loop() { anydata(); delay(10000);

}

# Circuit Diagram:-



**Output:-**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A computer screen shot of a circuit board

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Experiment No. 16**

**Title:** Write an application to control the operation of Hardware simulated traffic signals

Roll No:SAI&DC75

Batch: S9

Date of Performance: \_ \_ /\_ \_/\_ \_ \_ \_

Date of Assessment: \_ \_ /\_ \_/\_ \_ \_ \_

|  |  |
| --- | --- |
| **Particulars** | **Marks** |
| **Attendance (05)** |  |
| **Journal (05)** |  |
| **Performance (05)** |  |
| **Understanding (05)** |  |
| **Total (20)** |  |
| **Signature of Staff Member** |  |

**Program:-**

void setup()

{

pinMode(13, OUTPUT); pinMode(12, OUTPUT); pinMode(7, OUTPUT);

}

void loop()

{

digitalWrite(13, HIGH);

delay(1000); // Wait for 1000 millisecond(s) digitalWrite(13, LOW);

delay(1000); // Wait for 1000 millisecond(s) digitalWrite(12, HIGH);

delay(1000); // Wait for 1000 millisecond(s) digitalWrite(12, LOW);

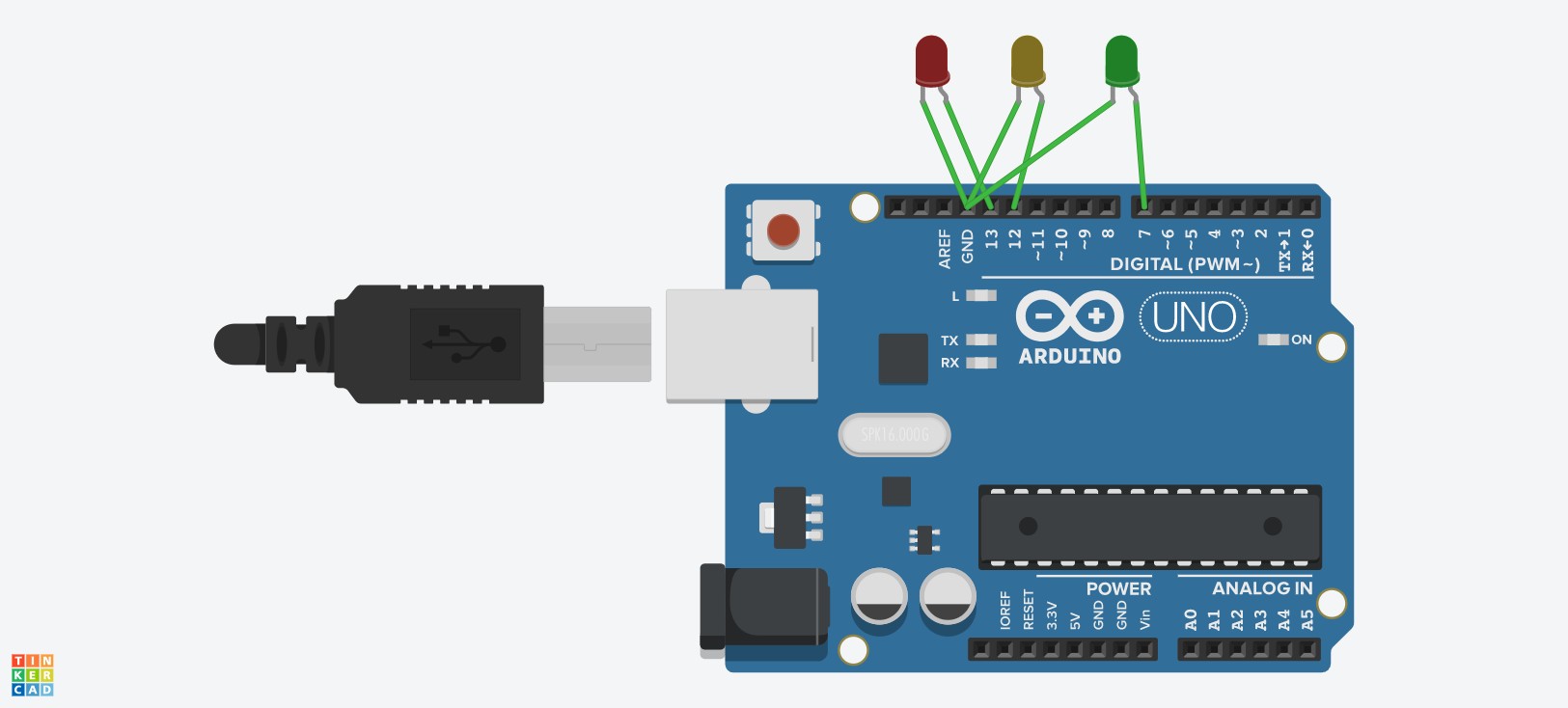
delay(1000); // Wait for 1000 millisecond(s) digitalWrite(7, HIGH);

delay(2000); // Wait for 1000 millisecond(s) digitalWrite(7, LOW);

delay(2000); // Wait for 1000 millisecond(s)

}

**Circuit Diagram:-**



**Output:-**

A computer screen shot of a blue circuit board

Description automatically generated with low confidence