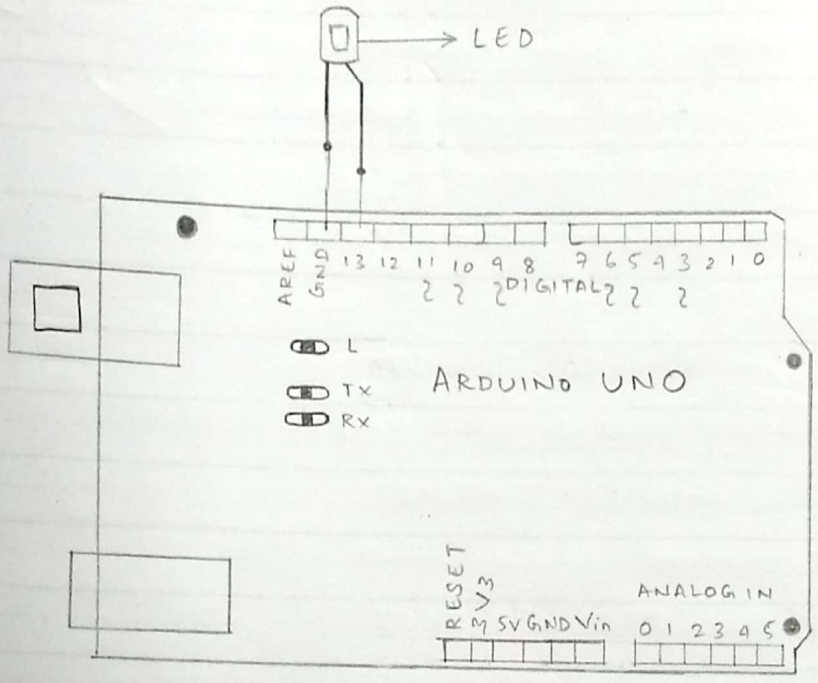
Particulars of the Experiments Performed

CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No.
1a.	13-8-19	LED BLINKING	10	01
1b.	13-8-19	LED FADING USING POTENTIOMETER		02
2a.	20-8-19	TRAFFIC LIGHT CONTROLLER		03-04
2b.	20-8-19	LED ON/OFF USING POTENTIOMETER		05
3.	27-8-19	NIGHTLIGHT SIMULATION USING LDR		06
4.	27-8-19	LED ON/OFF USING PUSH BUTTON		07
5.	27-8-19	SERIAL READ		08
6.	03-9-19	TEMPERATURE SENSOR		09
7.	03-9-19	FLAME SENSOR - FIRE ALERT SYSTEM		10
8.	03-9-19	SOIL MOISTURE SENSOR		11
9.	01-10-19	ULTRASONIC SENSOR AND LCD		12-13
10.	01-10-19	HUMAN PRESENCE DETECTION - PIR		14-15
11.	01-10-19	COLOUR RECOGNITION SENSOR USING RGB LED		16-18
12a.	15-10-19	BLUETOOTH AT COMMANDS		19
12b.	15-10-19	BLUETOOTH CONTROLLED BY MOBILE		20
12c.	15-10-19	BLUETOOTH RELAY MODULE		21-22
13a.	15-10-19	RFID READER		23
13b.	15-10-19	ACCESS CONTROL THROUGH RFID		24-25
14a.	15-10-19	GSM MODULE - CALL TO PARTICULAR NO.		26
14b.	15-10-19	SENDING AND RECEIVING MESSAGES USING GSM		27-28

CIRCUIT DIAGRAM:



Date 13-08-19

Expt. No. 01 Page No. 01

1a. LED BLINKING

AIM: Turns on an LED on for one second, then off for one second repeatedly.

COMPONENTS: Arduino-UNO board, LED

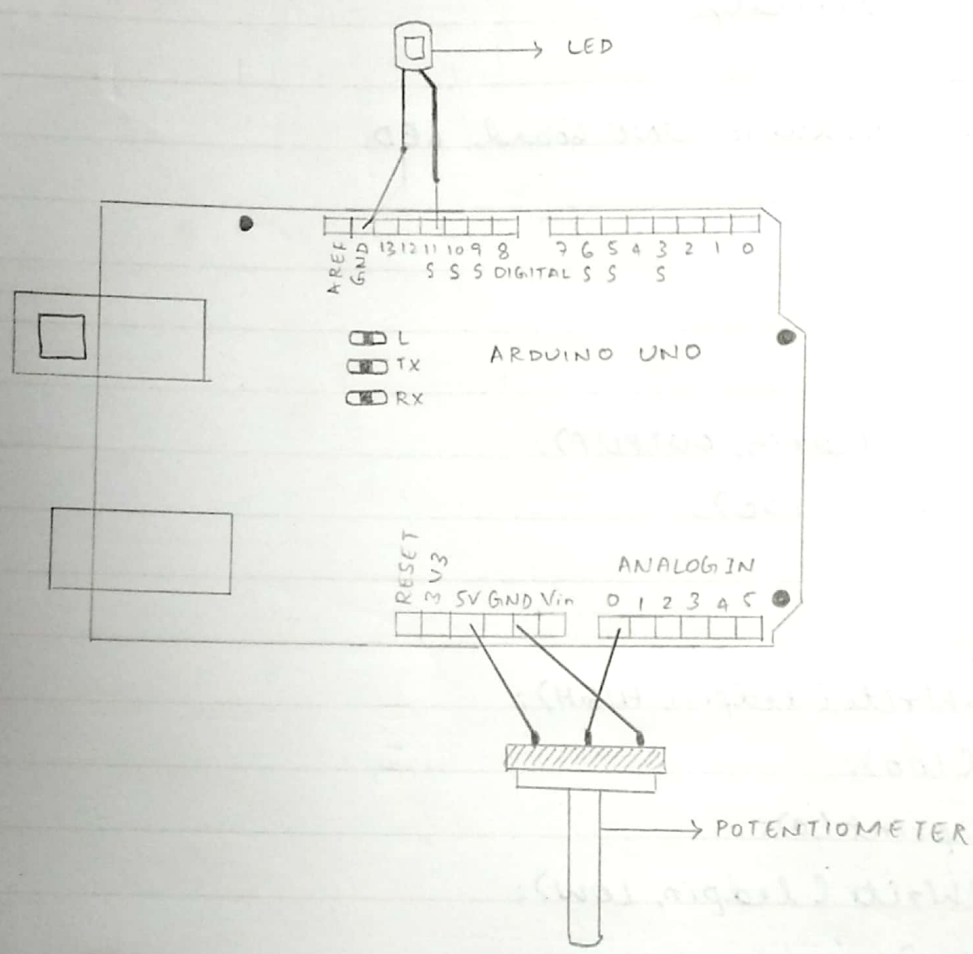
CODE:

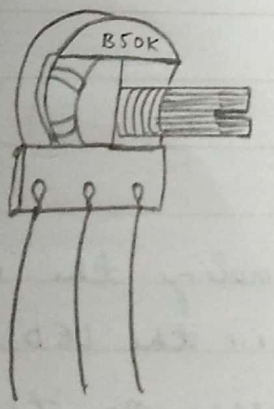
```
int ledpin = 13;
void setup()
{ pinMode (ledpin, OUTPUT);
  Serial.begin (9600);
}
void loop()
{ digitalWrite (ledpin, HIGH);
  delay (100);
  Serial.print (0);
  digitalWrite (ledpin, LOW);
  delay (100);
  Serial.print (1);
}
```

OBSERVATION: After compiling and uploading the code, the LED turns on and off repeatedly i.e the LED blinks, the output of which can also be seen on the serial monitor.

Teacher's Signature :

CIRCUIT DIAGRAM:





POTENTIOMETER

Date 13-08-19

Expt. No. 01 Page No. 02

1b. LED FADING USING POTENTIOMETER

AIM: To control the brightness of an LED using a potentiometer

COMPONENTS: Arduino-UNO board, LED, Potentiometer

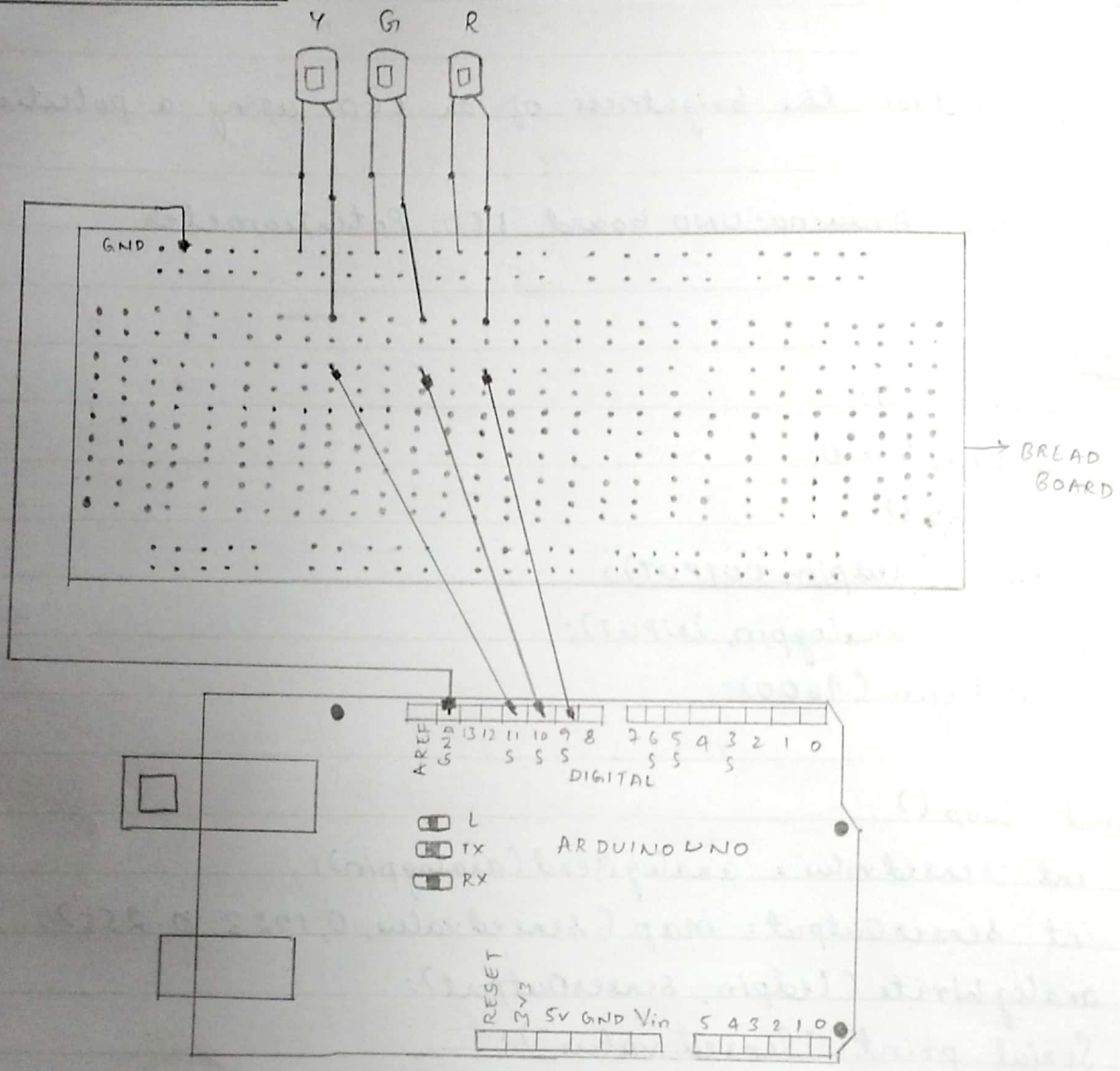
CODE:

```
int ledpin = 11;
int analogpin = 0;
void setup()
{ pinMode(ledpin, OUTPUT);
  pinMode(analogpin, INPUT);
  Serial.begin(9600);
}
void loop()
{ int sensedvalue = analogRead(analogpin);
  int sensorOutput: map(sensedvalue, 0, 1023, 0, 255);
  analogWrite(ledpin, sensorOutput);
  Serial.print("Sensed value");
  Serial.println(sensedvalue);
  Serial.print("Sensor Output");
  Serial.println(sensorOutput);
  delay(300);
}
```

OBSERVATION: By turning the knob of the potentiometer, the brightness of the LED is increased and decreased i.e the LED fades.

Teacher's Signature : _____

CIRCUIT DIAGRAM:



Date 20-8-19

Expt. No. 02 Page No. 03

2a. TRAFFIC LIGHT CONTROLLER

AIM: Traffic Signal Simulation using LED's

COMPONENTS: Arduino-UNO board, 3-LED's - Red, Yellow, Green, Bread Board, 4-jumper wires - Male-to-Male

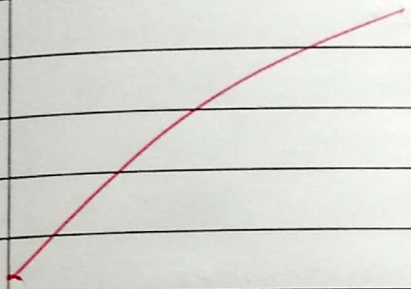
CODE:

```
int red = 9;
int green = 10;
int yellow = 11;
void setup()
{ pinMode (red, OUTPUT);
  pinMode (yellow, OUTPUT);
  pinMode (green, OUTPUT);
}
void loop()
{  digitalWrite (red, HIGH);
   digitalWrite (yellow, LOW);
   digitalWrite (green, LOW);
   delay (1000);
   digitalWrite (yellow, HIGH);
   digitalWrite (red, LOW);
   digitalWrite (green, LOW);
   delay (500);
   digitalWrite (green, HIGH);
   digitalWrite (red, LOW);
```

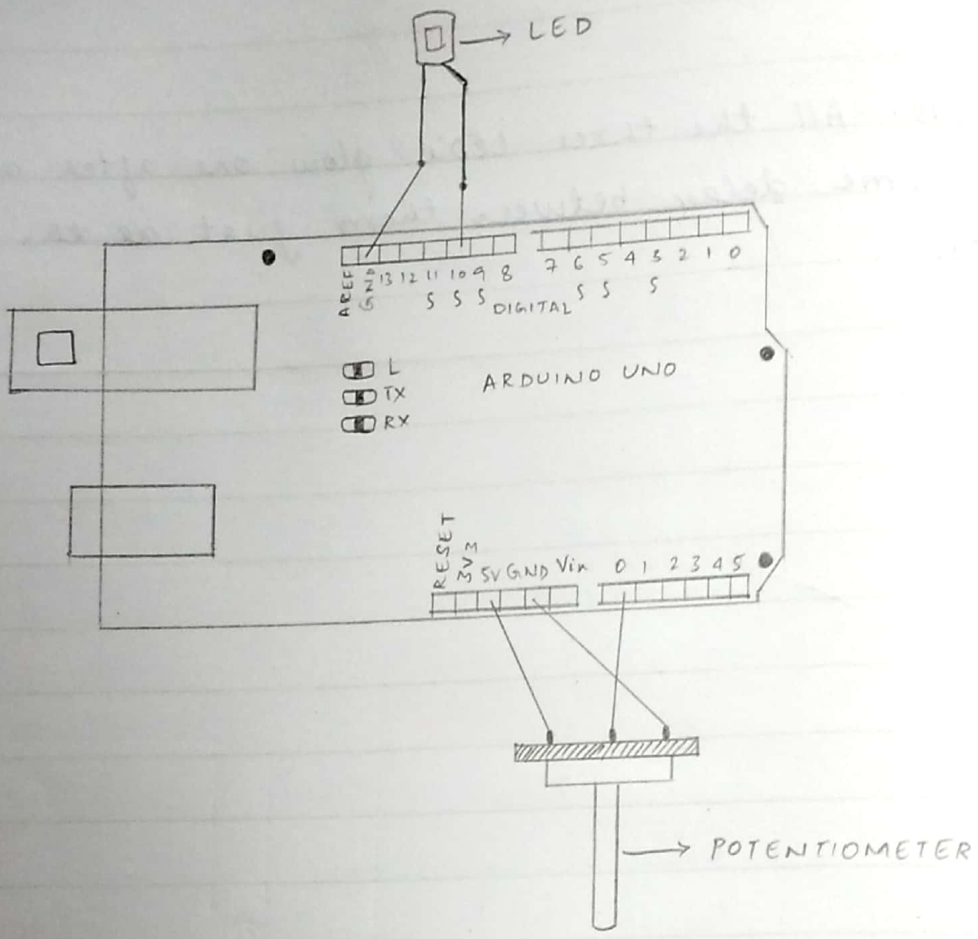
Teacher's Signature :

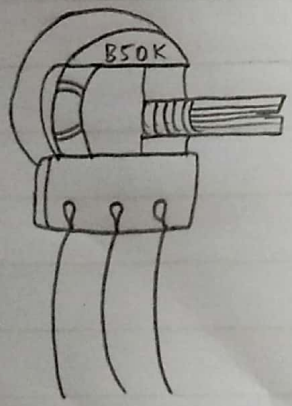
```
    digitalWrite (yellow, LOW);
    delay (2000);
}
```

OBSERVATIONS: All the three LED's glow one after another with some delay between them just as the traffic lights.



CIRCUIT DIAGRAM:-





POTENTIOMETER

Date 20-8-19

Expt. No. 02 Page No. 05

2(b) LED ON/OFF USING POTENTIOMETER

AIM: SWITCHING ON/OFF LED WITH USE OF POTENTIOMETER

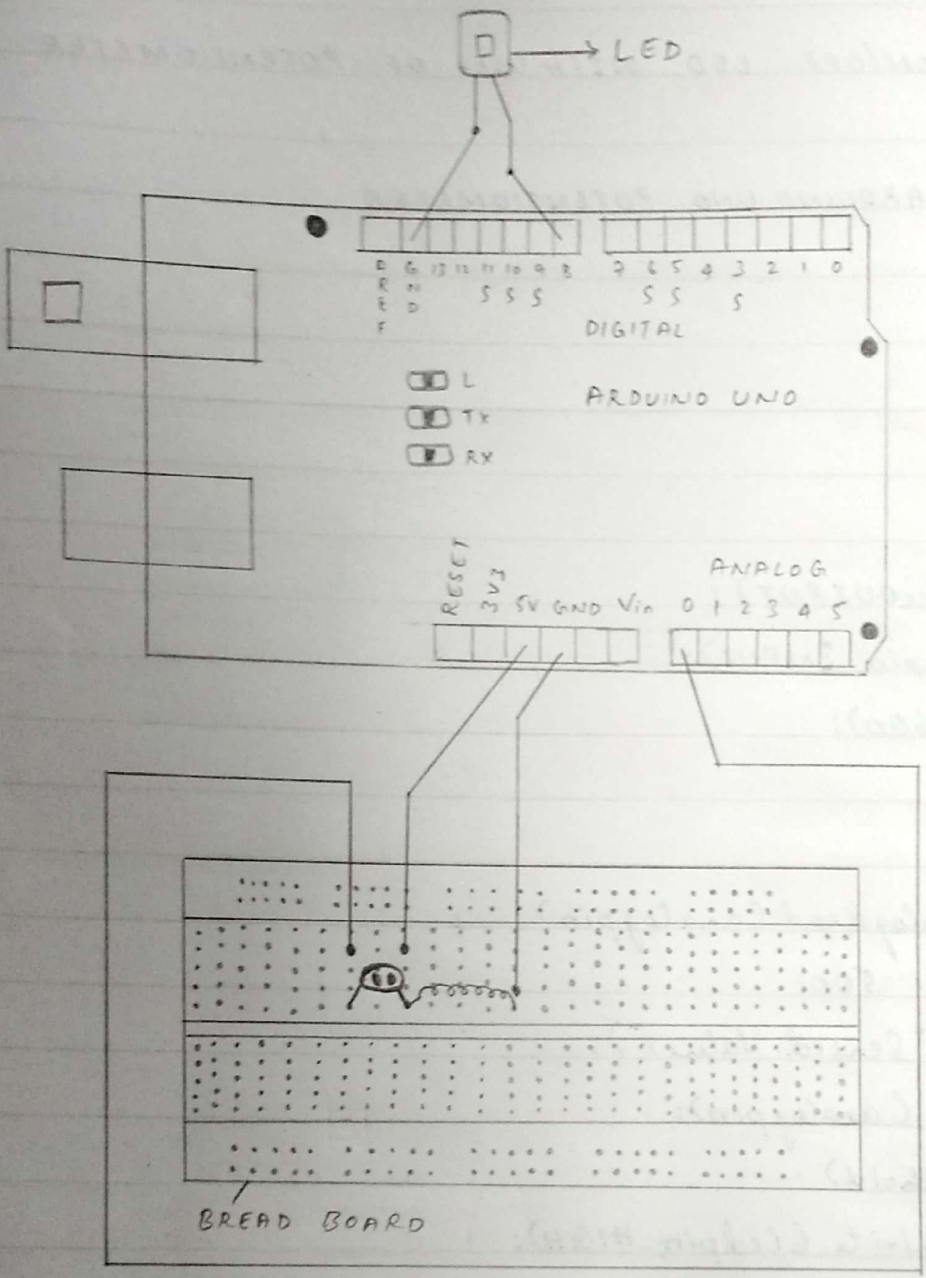
COMPONENTS: LED, ARDUINO UNO, POTENTIOMETER

CODE:

```
int ledpin = 10;
int analogpin = 0;
void setup()
{ pinMode (ledpin, OUTPUT);
  pinMode (analogpin, INPUT);
  Serial.begin (9600);
}
Void loop()
{ int val = analogRead (analogpin);
  int threshold = 500;
  Serial.print ("Sensed Value:");
  Serial.println (analogpin);
  if (val > threshold)
      digitalWrite (ledpin, HIGH);
  else
      digitalWrite (ledpin, LOW);
}
```

OBSERVATION: LED turns on when potentiometer crosses threshold and turns off when it goes below threshold.

CIRCUIT DIAGRAM:



Date 27-8-19

Expt. No. 03 Page No. 06

3. NIGHTLIGHT SIMULATION

AIM: Simulate a night light using LDR

COMPONENTS: LED, LDR, Resistor, Breadboard, Jumper Wires, Arduino Board

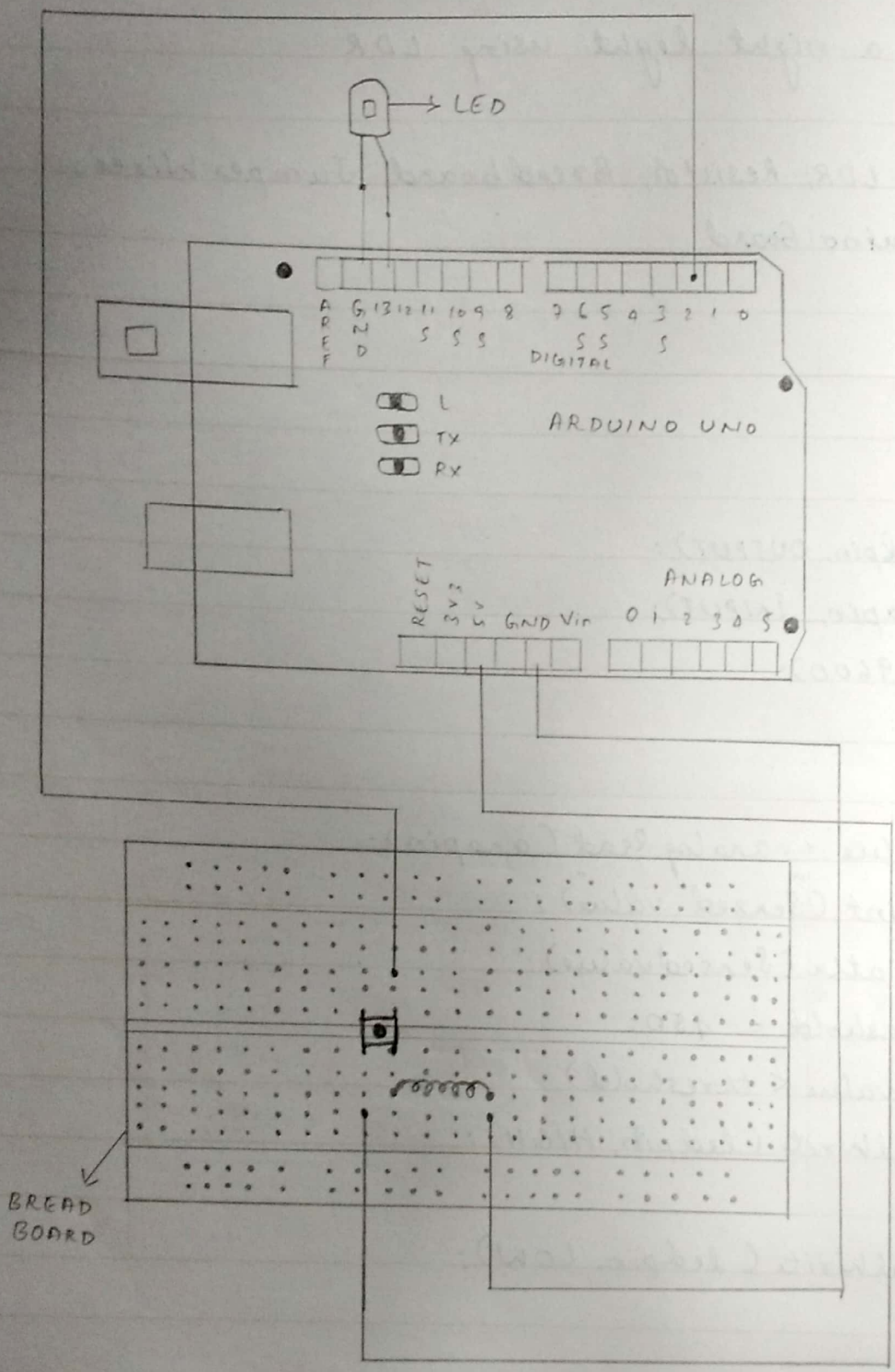
CODE:

```
int ledpin = 8;
int anapin = 0;
void setup()
{  pinMode (ledpin, OUTPUT);
   pinMode (anapin, INPUT);
   Serial.begin (9600);
}
void loop()
{  int sensedvalue = analogRead (anapin);
      Serial.print ("Sensed value");
      Serial.println (Sensedvalue);
      int threshold = 450;
      if (sensedvalue < threshold)
         digitalWrite (ledpin, HIGH);
      else
         digitalWrite (ledpin, LOW);
}
```

OBSERVATION: When lights are switched off in the room, LED should switch on. When lights are switched on in the room, LED should switch off immediately.

Teacher's Signature : _____

CIRCUIT DIAGRAM:



Date 27-8-19

Expt. No. 04 Page No. 07

4. LED ON/OFF USING PUSH BUTTON

AIM: Turn on LED ON/OFF using a Push button

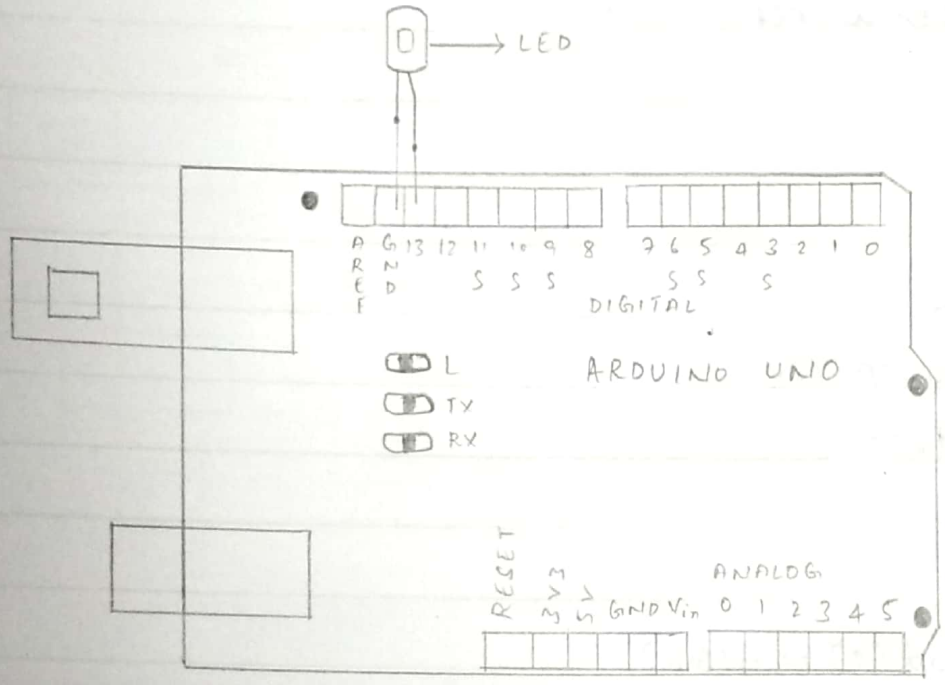
COMPONENTS: Arduino Board, Breadboard, LED, Push button, Resistor, Jumper wires

CODE:

```
void setup()
{ pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
void loop()
{ if(digitalRead(2) == HIGH)
    digitalWrite(13, HIGH);
  else
    digitalWrite(13, LOW);
  int val = digitalRead(2);
  Serial.print("Value:");
  Serial.println(val);
}
```

OBSERVATION: On pressing the Push Button, the LED glows and on releasing it the LED switches off.

CIRCUIT DIAGRAM:



Date 27-8-19

Expt. No. 05 Page No. 08

8. SERIAL READ

AIM: To control blinking of LED based on input in serial monitor.

COMPONENTS: Arduino Board, LED

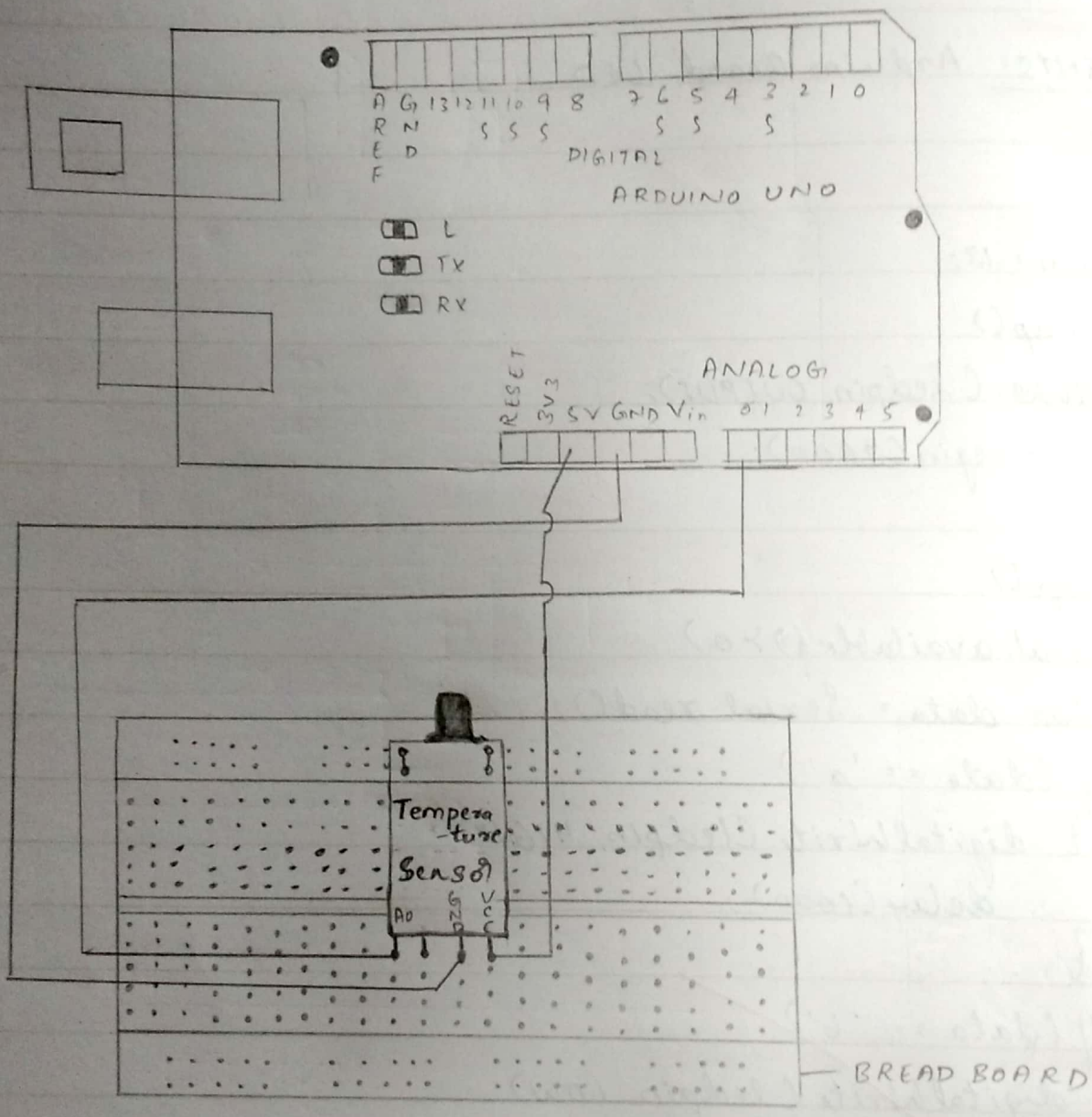
CODE:

```
int ledpin = 13;
void setup()
{ pinMode (ledpin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{ if (Serial.available()>0)
  { char data = Serial.read();
    if (data == 'a')
    { digitalWrite (ledpin, HIGH);
      delay(1000);
    }
    if (data == 'b')
    { digitalWrite (ledpin, LOW);
      delay(1000);
    }
} }
```

OBSERVATION: On entering 'a' in serial monitor the led turns on and on entering 'b' it turns off.

Teacher's Signature :

CIRCUIT DIAGRAM:-



Expt. No. 06 Date 3-9-19

Page No. 09

6. TEMPERATURE SENSOR

AIM: To design and implement a temperature monitor using temperature sensor.

COMPONENTS: Arduino-UNO, Breadboard, Temperature Sensor - LM35, wires

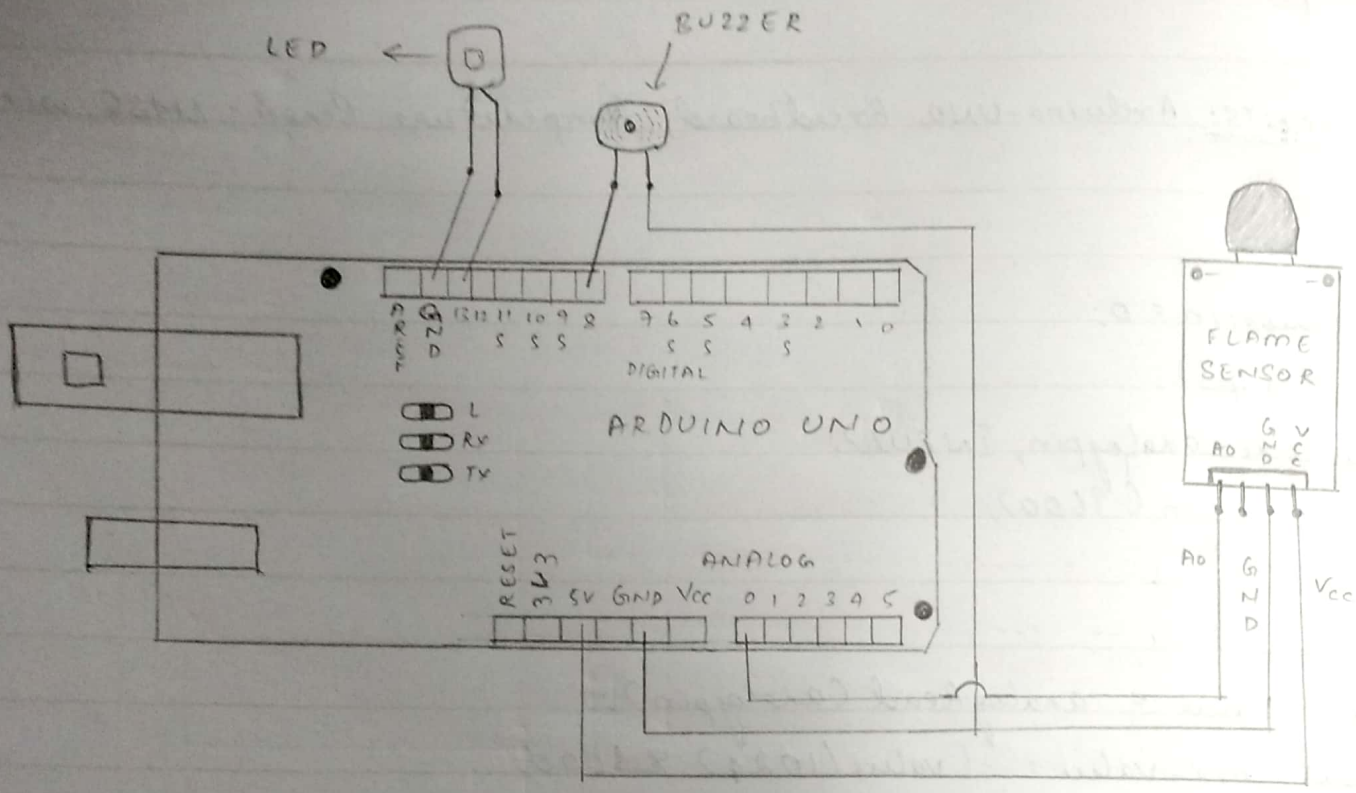
CODE:

```
int analogpin = 0;
void setup()
{ pinMode(analogpin, INPUT);
  Serial.begin(9600);
}
void loop()
{ float value = analogRead(analogpin);
  float mv_value = (value/1024) * 5000;
  float temp_in_C = mv_value/10;
  Serial.print("Temp in °C = ");
  Serial.println(temp_in_C);
  delay(1000);
  float temp_F = ((9/5) * temp_in_C) + 32;
  Serial.print("Temp in °F = ");
  Serial.println(temp_in_F);
  delay(1000);
}
```

OBSERVATIONS: Temperature displayed on monitor increments when fingers touch the sensor and decrements when a cold object touches it.

Teacher's Signature : _____

CIRCUIT DIAGRAM:



Date 3-9-19

Expt. No. 07 Page No. 10

7. FLAME SENSOR - FIRE ALERT SYSTEM

AIM: To design and implement a fire alarm system using flame sensor and buzzer.

COMPONENTS: Arduino-UNO, Bread board, LED, buzzer, Flame sensor-LM393

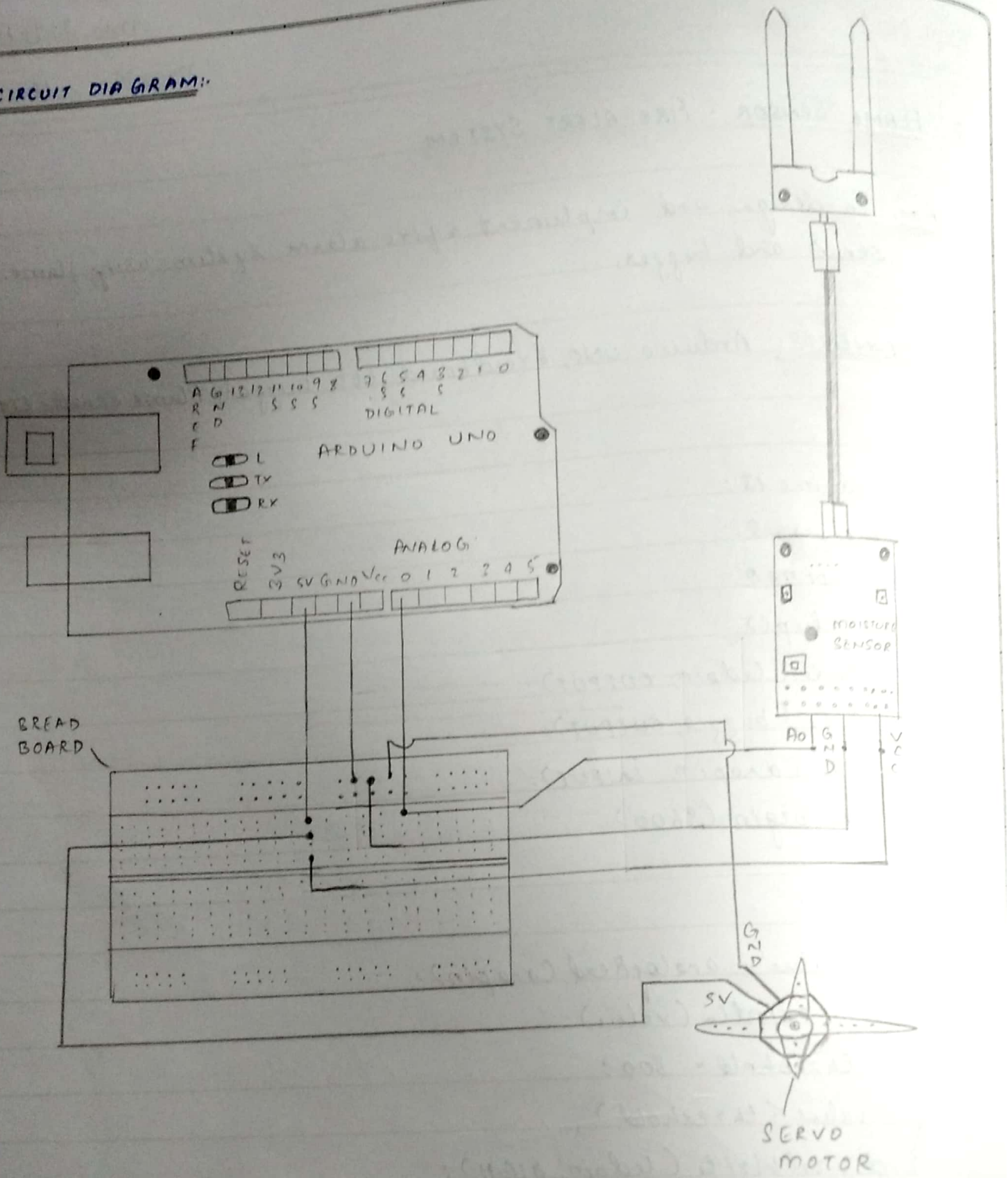
CODE:

```
int ledpin = 13;
int buzzer = 8;
int anapin = 0;
void setup()
{ pinMode (ledpin, OUTPUT);
  pinMode (buzzer, OUTPUT);
  pinMode (anapin, INPUT);
   Serial.begin (9600);
}
void loop()
{ int value = analogRead (anapin);
  Serial.println (value);
  int threshold = 300;
  if (value < threshold)
   { digitalWrite (ledpin, HIGH);
     digitalWrite (buzzer, HIGH); }
  else { digitalWrite (ledpin, LOW);
         digitalWrite (buzzer, LOW); }
}
```

OBSERVATION: When flame is brought near sensor buzzer rings.

Teacher's Signature : _____

CIRCUIT DIAGRAM:-



Expt. No. 08 Date 3-9-19

Page No. 11

8. SOIL MOISTURE SENSOR

AIM: To design and implement smart irrigation system using soil moisture sensor and servo motor.

COMPONENTS: Arduino-UNO, Moisture Sensor, Servo motor, Bread board, wires.

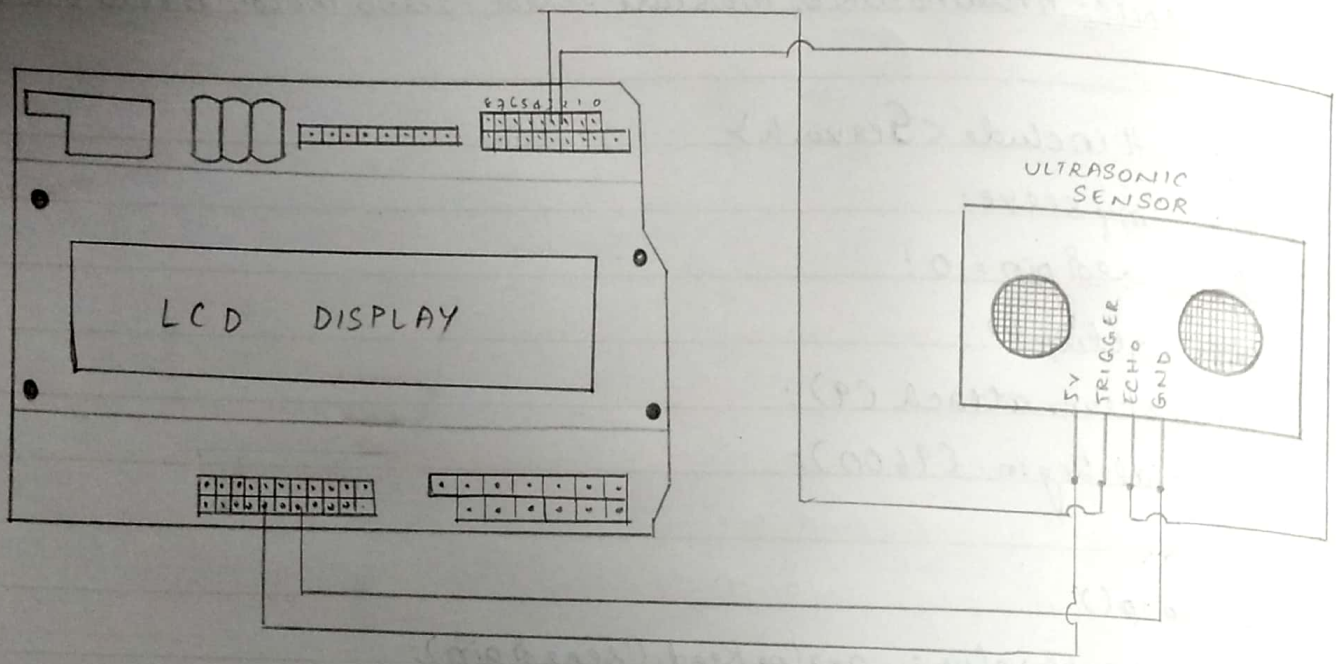
CODE:

```
#include <Servo.h>
Servo myservo;
int sensorpin = 0;
void setup()
{  myservo.attach(9);
   Serial.begin(9600);
}
void loop()
{ int sensorvalue = analogRead(sensorpin);
  Serial.print(sensorvalue);
  if(sensorvalue > 500)
  { for(int pos=0; pos<180; pos+=1)
    { myservo.write(pos); delay(150); }
    for(int pos=180; pos>=0; pos-=1)
    { myservo.write(pos); delay(150); }
  }
}
```

OBSERVATION: Servo motor rotates when moisture sensor taken out of water and stops rotating when dipped in water.

Teacher's Signature : _____

CIRCUIT DIAGRAM:-



Date 1-10-19

Expt. No. 09 Page No. 12

9. REVERSE PARKING SENSOR USING ULTRASONIC SENSOR AND LCD

AIM: Measuring the distance using Ultrasonic sensor which helps in reverse parking system.

COMPONENTS: Arduino board, Bread board, Ultrasonic Sensor, LCD, Jumper Wires

CODE:

```
#include <Liquid Crystal.h>
LiquidCrystal lcd (8, 9, 4, 5, 6, 7);
int trigPin = 3;
int echoPin = 2;
void setup()
{ lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop()
{ long duration, inches, cm;
  lcd.setCursor(0, 1);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  duration = pulseIn(echopin, HIGH);
```

Teacher's Signature : _____

Date 1-10-19

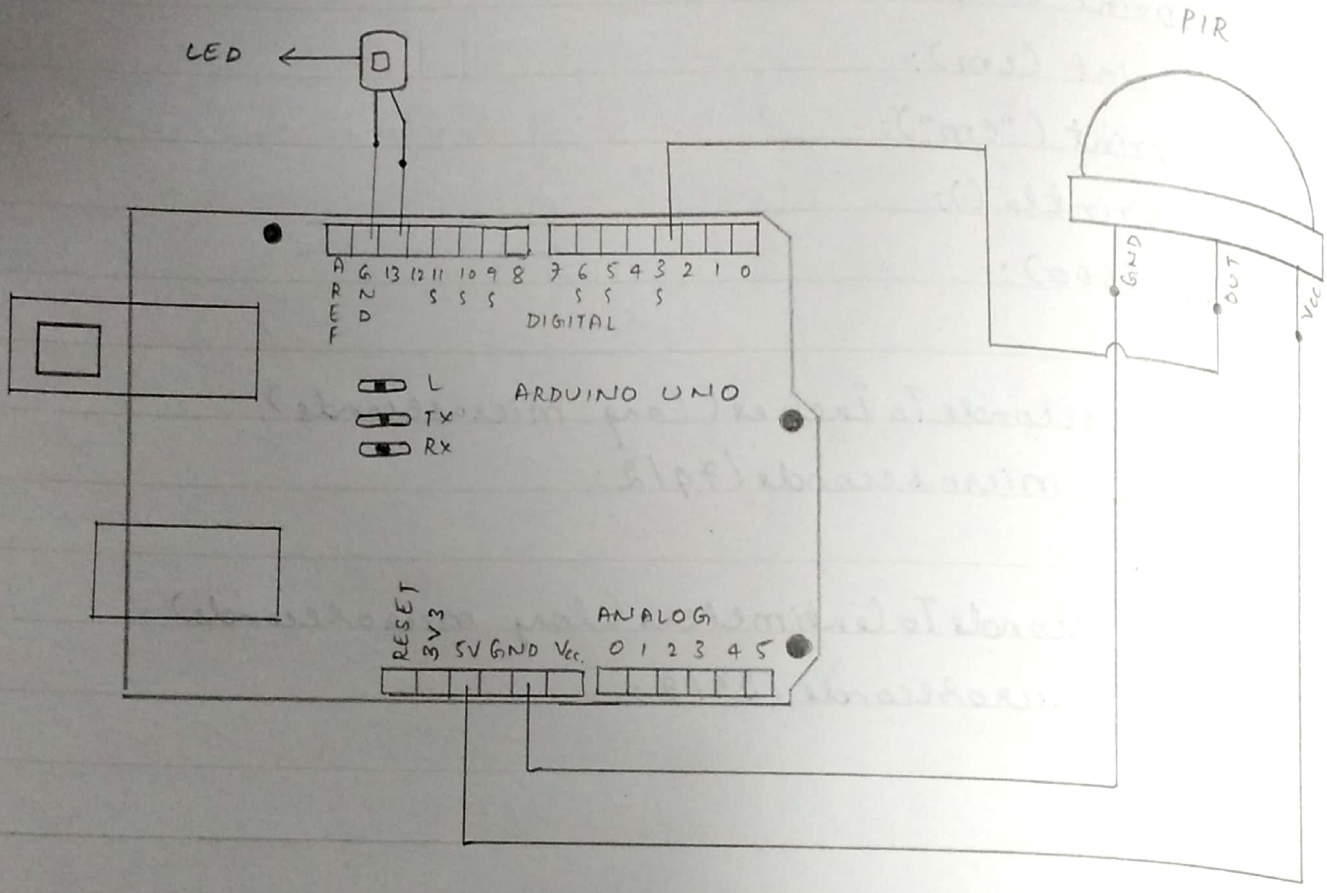
Expt. No. 9 Page No. 13

```
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);
    lcd.print(inches);
    lcd.print("in");
    lcd.print(cm);
    lcd.print("cm");
    lcd.println();
    delay(500);
}

long microsecondsToInches(long microseconds)
{ return microseconds/74/2;
}

long microsecondsToCentimeters(long microseconds)
{ return microseconds/29/2;
}
```

CIRCUIT DIAGRAM:-



Date 1-10-19

Expt. No. 10 Page No. 14

10. HUMAN PRESENCE DETECTION USING PIR

AIM: To simulate human presence detection.

COMPONENTS: Arduino Board, Bread Board, PIR, Jumper Wires.

CODE:

```
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;
int pirPin = 3;
int ledPin = 13;
void setup()
{ Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(pirPin, INPUT);
  digitalWrite(pirPin, LOW);
  Serial.print("calibrating sensor ");
  for(int i=0; i<calibrationTime; i++)
  { Serial.print(".");
    delay(100);
  }
  Serial.println(" done");
  Serial.println("SENSOR ACTIVE");
  delay(10);
}
```

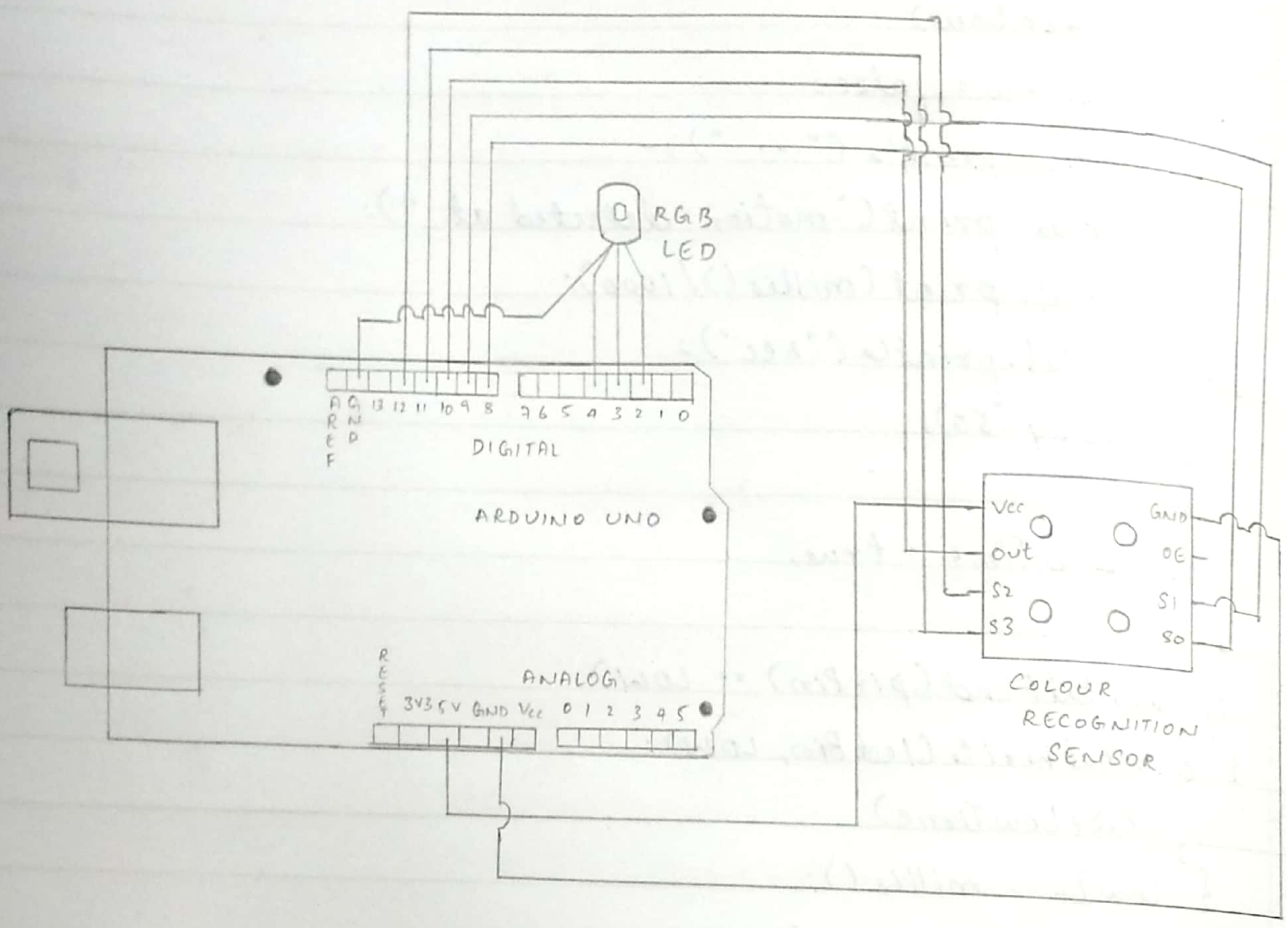
Teacher's Signature :

Date 1-10-19

Expt. No. 10 Page No. 15

```
void loop()
{ if(digitalRead(pirPin) == HIGH)
  { digitalWrite(ledPin, HIGH);
    if(lockLow)
    { lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println("sec");
      delay(50);
    }
    takeLowTime = true;
  }
  if(digitalRead(pirPin) == LOW)
{ digitalWrite(ledPin, LOW);
  if(takeLowTime)
  { lowIn = millis();
    takeLowTime = false;
  }
  if(!lockLow && millis() - lowIn > pause)
  { lockLow = true;
    Serial.print("motion ended at ");
    Serial.print((millis() - pause)/1000);
    Serial.println("sec");
    delay(50);
  }
 }
}
```

CIRCUIT DIAGRAM:



Expt. No. 11 Date 1-10-19

Page No. 16

11. COLOUR RECOGNITION SENSOR USING RGB LED

AIM: To identify all the RGB colours using colour recognition sensor.

COMPONENTS: Arduino Board, RGB led, Colour Recognition Sensor, Bread Board, Jumper wires

CODE:

```
const int s0 = 8;
const int s1 = 9;
const int s2 = 12;
const int s3 = 11;
const int out = 10;
int redled = 2;
int greenled = 3;
int blueled = 4;
int red = 0, green = 0, blue = 0;
void setup()
{ Serial.begin(9600);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(blueLed, OUTPUT);
```

Teacher's Signature :

```
    digitalWrite (S0, HIGH);
    digitalWrite (S1, HIGH);
}
void loop()
{ color();
  Serial.print("R Intensity:");
  Serial.print(red, DEC);
  Serial.print("G Intensity:");
  Serial.print(green, DEC);
  Serial.print("B intensity:");
  Serial.print(blue, DEC);
  if(red<blue && red<green)
  { Serial.println(" - (Red Colour)");
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, LOW);
  }
  else if(blue<red && blue<green)
  { Serial.println(" - (Blue colour)");
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, HIGH);
  }
  else if(green<red && green<blue)
  { Serial.println(" - (Green Colour)");
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    digitalWrite(blueLed, LOW); }
```

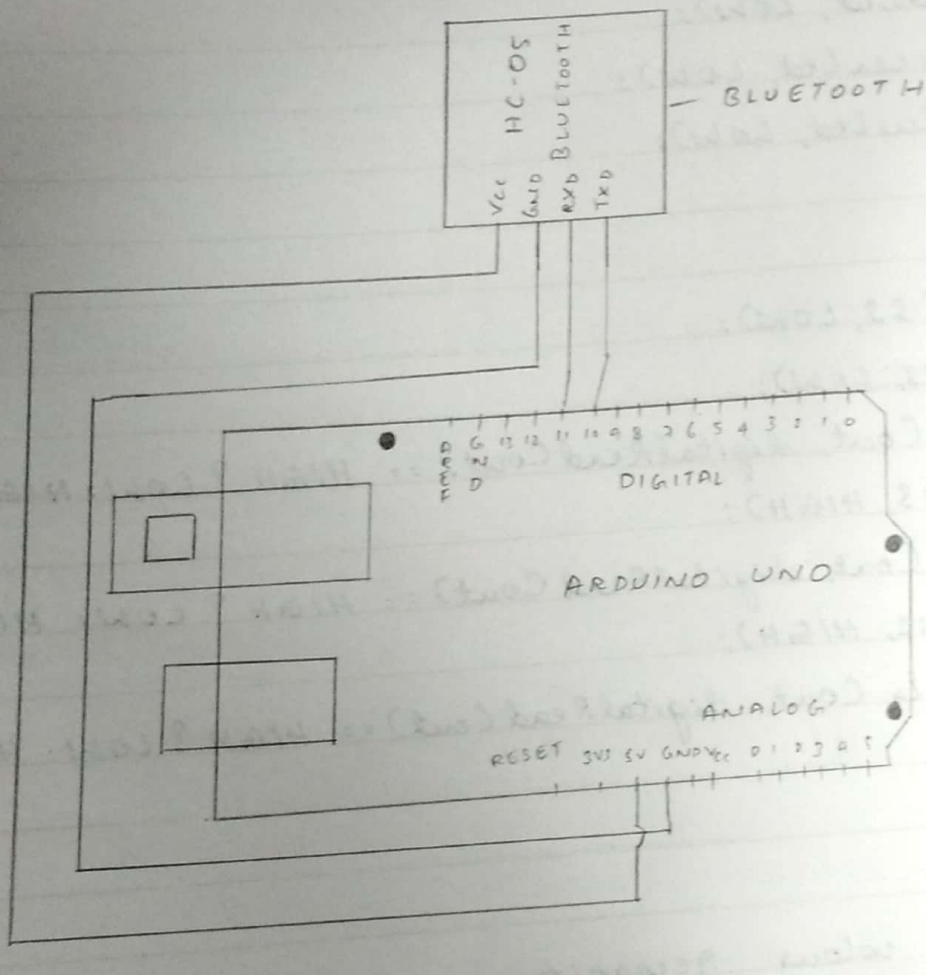
Teacher's Signature :

```
else
{ Serial.println();}
delay(300);
digitalWrite(redLed, LOW);
digitalWrite(greenLed, LOW);
digitalWrite(blueLed, LOW);
}
void color()
{ digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  red = pulseIn(Out, digitalRead(Out) == HIGH ? LOW: HIGH);
  digitalWrite(S3, HIGH);
  blue = pulseIn(Out, digitalRead(Out) == HIGH ? LOW: HIGH);
  digitalWrite(S2, HIGH);
  green = pulseIn(Out, digitalRead(Out) == HIGH ? LOW: HIGH);
}
```

OBSERVATIONS:-

When the colour recognition sensor was placed infront of any object, the RGB led glowed with the colour nearest to the detected colour.

CIRCUIT DIAGRAM:



Expt. No. 12 Date 15-10-19

Page No. 19

12a. BLUETOOTH AT COMMANDS

AIM: Working on AT commands making use of Bluetooth module.

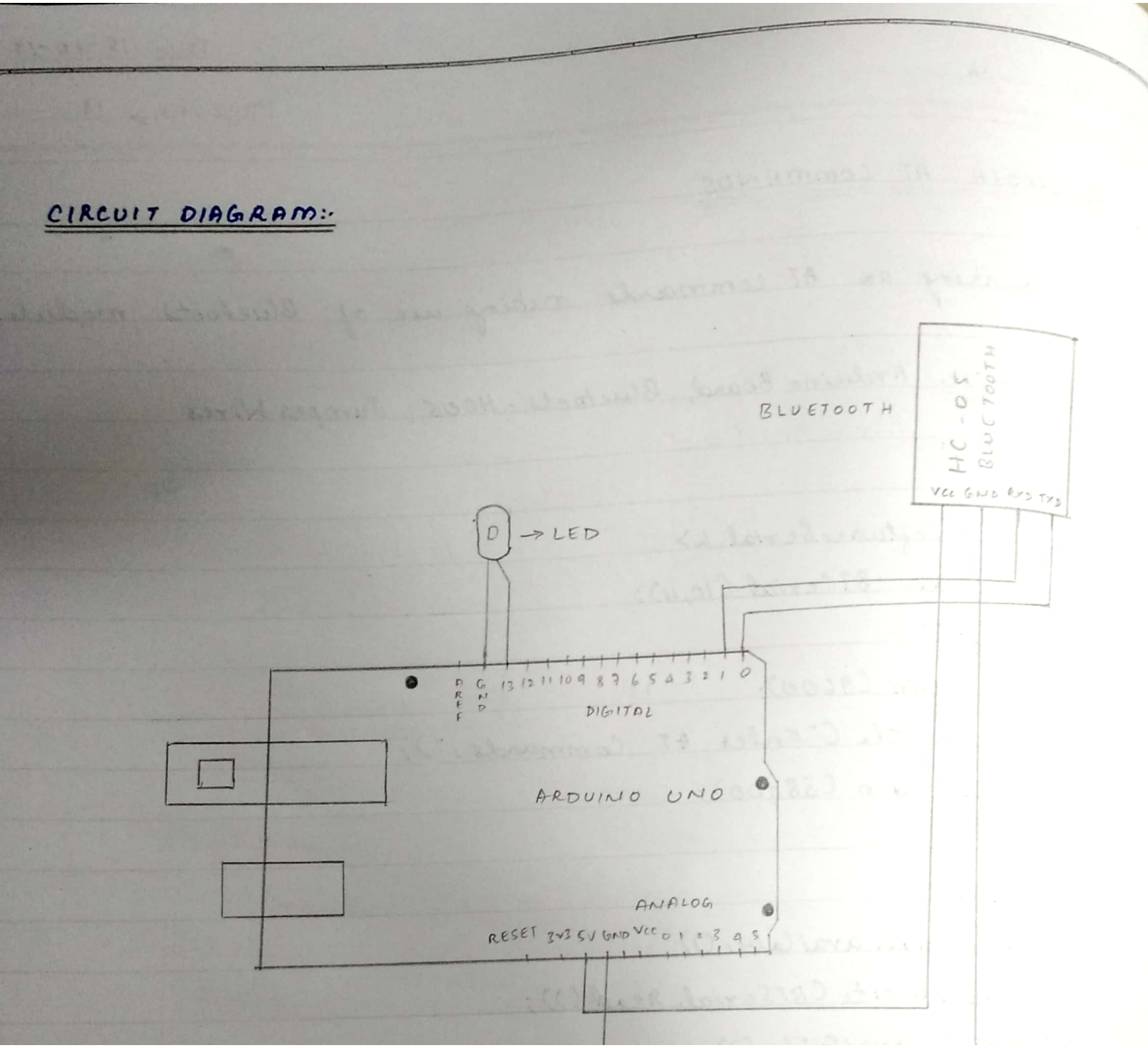
COMPONENTS: Arduino Board, Bluetooth - HC05, Jumper Wires

CODE:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10,11);
void setup()
{ Serial.begin(9600);
  Serial.println("Enter AT Commands:");
  BTSerial.begin(38400);
}
void loop()
{ if(BTSerial.available())
    Serial.write(BTSerial.read());
  if(Serial.available())
    BTSerial.write(Serial.read());
}
```

OBSERVATIONS: By entering the AT commands in the Serial monitor, we can observe the results.

CIRCUIT DIAGRAM:-



Date 15-10-19

Expt. No. 12 Page No. 20

12b. BLUETOOTH CONTROLLED BY MOBILE

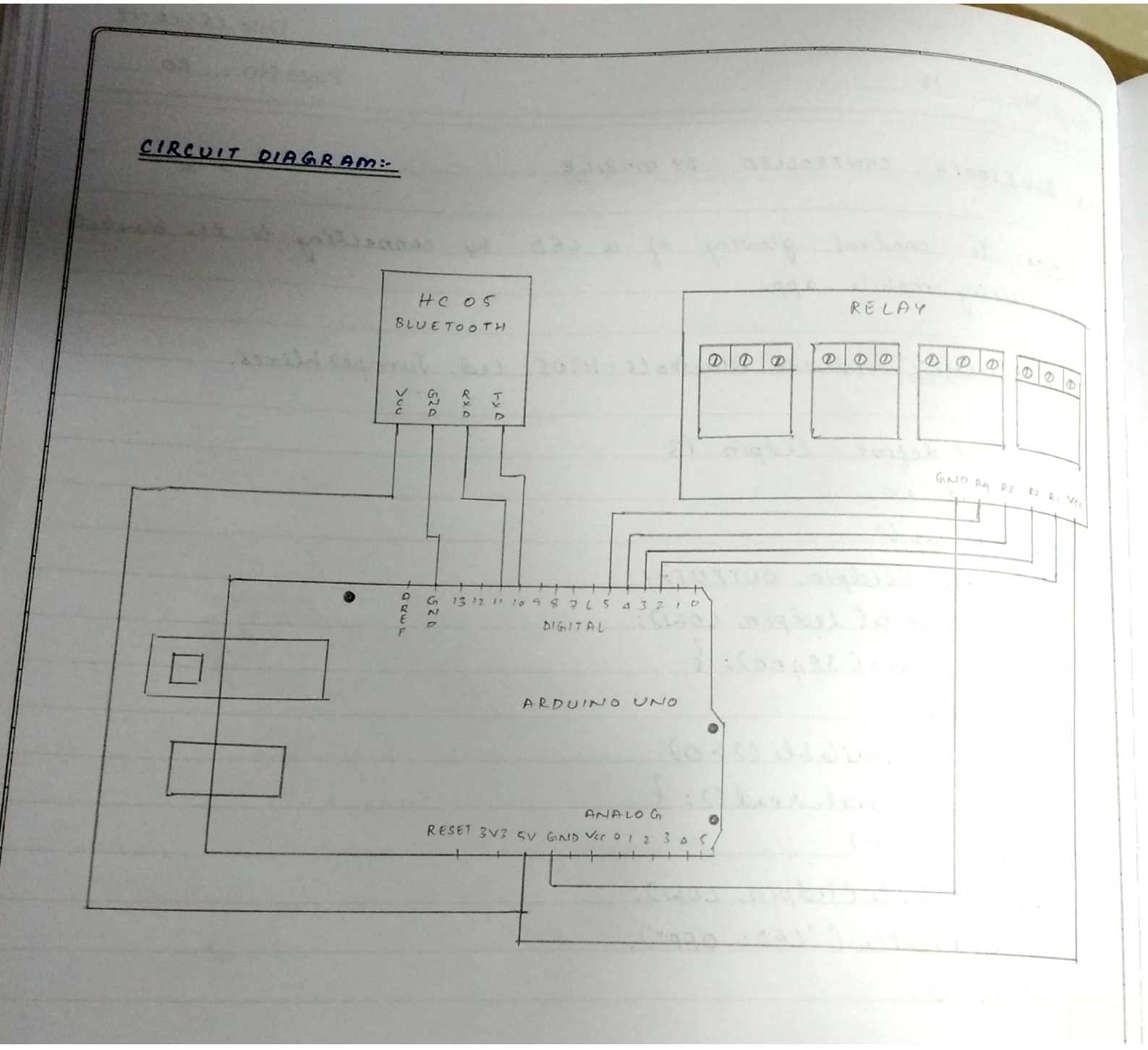
AIM: To control glowing of a LED by connecting to the bluetooth using mobile app.

COMPONENTS: Arduino, Bluetooth-HC05, Led, Jumper wires.

CODE:-

```
# define ledpin 13
int state = 0;
void setup()
{ pinMode(ledpin, OUTPUT);
  digitalWrite(ledpin, LOW);
  Serial.begin(38400); }
void loop()
{ if (Serial.available()>0)
  { state = Serial.read(); }
if (state == '0')
{ digitalWrite(ledpin, LOW);
  Serial.println("LED: OFF");
  state = 0;
}
else if (state == '1')
{ digitalWrite(ledpin, HIGH);
  Serial.println("LED: ON");
  State = 0;
}
}
```

CIRCUIT DIAGRAM:-



Expt. No. 12 Date 15-10-19

Page No. 21

12c. BLUETOOTH RELAY MODULE

AIM: Working with Bluetooth Relay using Mobile bluetooth app.

COMPONENTS: Arduino Board, Bluetooth- HC05, Relay module, Jumper Wires.

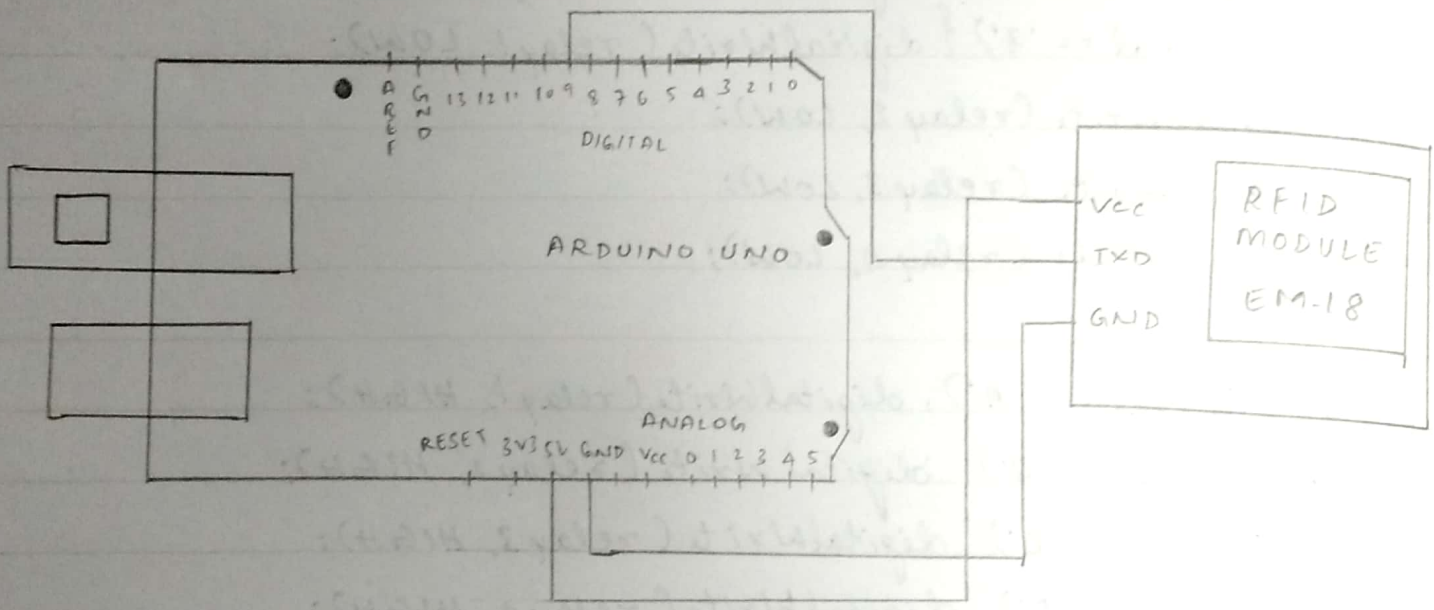
CODE:-

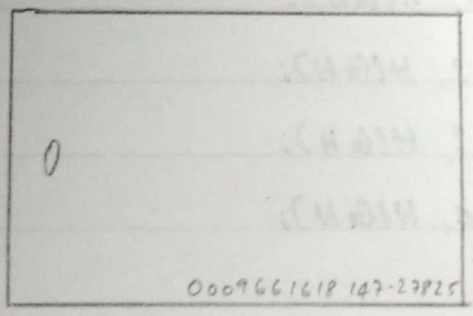
```
#include <SoftwareSerial.h>
SoftwareSerial mySerial (10, 11);
#define relay1 2
#define relay2 3
#define relay3 4
#define relay4 5
char val;
void setup()
{ pinMode (relay1, OUTPUT);
  pinMode (relay2, OUTPUT);
  pinMode (relay3, OUTPUT);
  pinMode (relay4, OUTPUT);
  digitalWrite (relay1, HIGH);
  digitalWrite (relay2, HIGH);
  digitalWrite (relay3, HIGH);
  digitalWrite (relay4, HIGH);
  mySerial.begin (9600);
  Serial.begin (9600);
}
```

```
void loop()
{ if (mySerial.available()>0)
   { val = mySerial.read();
     Serial.println(val); }
  if (val == '1') digitalWrite(relay1, LOW);
 else if (val == '2') digitalWrite(relay2, LOW);
 else if (val == '3') digitalWrite(relay3, LOW);
 else if (val == '4') digitalWrite(relay4, LOW);
 else if (val == '9') { digitalWrite(relay1, LOW);
   digitalWrite(relay2, LOW);
   digitalWrite(relay3, LOW);
   digitalWrite(relay4, LOW);
 }
 else if (val == 'A') digitalWrite(relay1, HIGH);
 else if (val == 'B') digitalWrite(relay2, HIGH);
 else if (val == 'C') digitalWrite(relay3, HIGH);
 else if (val == 'D') digitalWrite(relay4, HIGH);
 else if (val == 'I')
 { digitalWrite(relay1, HIGH);
   digitalWrite(relay2, HIGH);
   digitalWrite(relay3, HIGH);
   digitalWrite(relay4, HIGH);
 }
}
```

OBSERVATIONS: By using the Bluetooth relay mobile application we can control the Bluetooth Relay module.

CIRCUIT DIAGRAM:-





RFID TAG

Expt. No. 13 Date 15-10-19

Page No. 23

13a. RFID READER

AIM: To read the code present on RFID tag:

COMPONENTS: Arduino Board, RFID tags, RFID Reader Module - EM18, Jumper Wires.

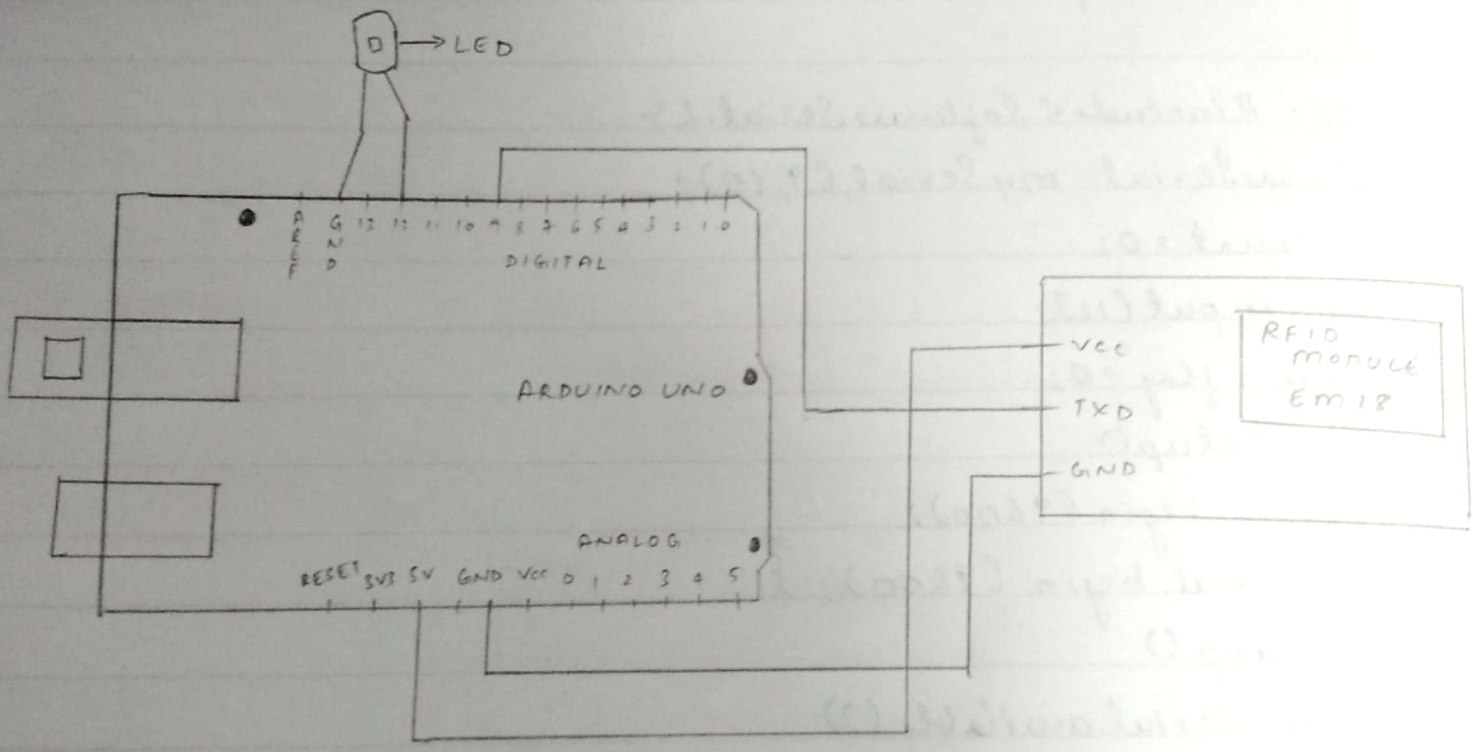
CODE:

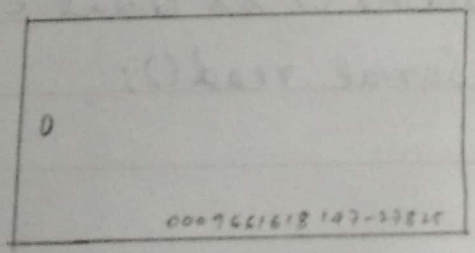
```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10);
int count=0;
char input[12];
boolean flag=0;
void setup()
{ Serial.begin(9600);
  mySerial.begin(9600); }
void loop()
{ if(mySerial.available())
 { count=0;
   while(mySerial.available() && count<12)
   { input[count]=mySerial.read();
     count++;
     delay(5); }
   Serial.print(input);
} }
```

OBSERVATION: On Scanning the RFID tag onto the RFID reader module, the code will be seen on serial monitor.

Teacher's Signature : _____

CIRCUIT DIAGRAM:-





RFID TAG

Date 15-10-19

Expt. No. 13 Page No. 24

13b. ACCESS CONTROL THROUGH RFID

AIM: To grant access for a particular RFID tag already configured, and if yes the LED will glow.

COMPONENTS: Arduino Board, RFID Tag, RFID Reader module - EM18, Jumper Wires.

CODE:-

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial (9,10);
#define LEDPIN 12
char tag[] = "3C0087059 7F9";
char input[12];
int count = 0;
boolean flag = 0;
void setup()
{ Serial.begin (9600);
  mySerial.begin (9600);
  pinMode (LEDPIN, OUTPUT);
}
void loop()
{ if(mySerial.available ())
  { count = 0;
    while (mySerial.available () && count <12)
    { input [count] = mySerial.read ();
      Serial.write (input [count]);
      count ++;
```

Teacher's Signature :

```
    delay (5);
}

if (count == 12)
{   count = 0;
    flag = 1;
  while (count < 12 && flag != 0)
  { if (input[count] == tag[count])
    flag = 1;
    else flag = 0;
    count ++;
} }

 if (flag == 1)
 { Serial.println ("Access Allowed!");
    digitalWrite (LEDPIN, HIGH);
    delay (2000);
    digitalwrite (LEDPIN, LOW); }
else { Serial.println ("Access Denied");
       digitalWrite (LEDPIN, LOW);
       delay (2000); }
  for (count = 0; count < 12; count ++)
   { input[count] = 'F'; }
   count = 0;
} }
```

OBSERVATION : If the scanned RFID tag matches with the known tag, then access is allowed and LED will glow.

CIRCUIT DIAGRAM:-

Date 15-10-19

Expt. No. 14

Page No. 26

14a. GSM MODULE: CALL TO A PARTICULAR NUMBER

AIM:- Call using Arduino and GSM Module - to a specified numbers inside the program.

COMPONENTS:- Arduino Board, GSM Module, Jumpes wires.

CODE:-

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);
void setup()
{ cell.begin(9600);
  delay(500);
  Serial.begin(9600);
  Serial.println("CALLING.....");
  cell.println("ATD+9876543210;");
  delay(10000);
}

void loop()
{ }
```

OBSERVATIONS:- A call will be made to the above mentioned numbers.

CIRCUIT DIAGRAM:

14b. SENDING AND RECEIVING MESSAGES USING GSM

AIM: Send and Receive SMS using Arduino and GSM Module to a specified number & from a specific number to the SIM card loaded in the GSM Module.

COMPONENTS: Arduino Board, GSM module, Jumper wires.

CODE:-

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);
void setup()
{ mySerial.begin(9600);
  Serial.begin(9600);
  delay(100);
}
void loop()
{ if(Serial.available()>0)
  switch(Serial.read())
  { case 's': SendMessage();
              break;
    case 'r': ReceiveMessage();
              break;
  }
 if(mySerial.available()>0)
  Serial.write(mySerial.read());
}
```

Teacher's Signature : _____

Date 15-10-19

Expt. No. 14

Page No. 28

```
void sendMessage()
{ mySerial.println("AT+CMGF=1");
  delay(1000);
 mySerial.println("AT+CMGS=\"+919742000666\"\r");
  delay(1000);
 mySerial.println("I am SMS from GSM module");
  delay(1000);
}
void ReceiveMessage()
{ mySerial.println("AT+CNMI=2,2,0,0,0");
  delay(1000);
}
```