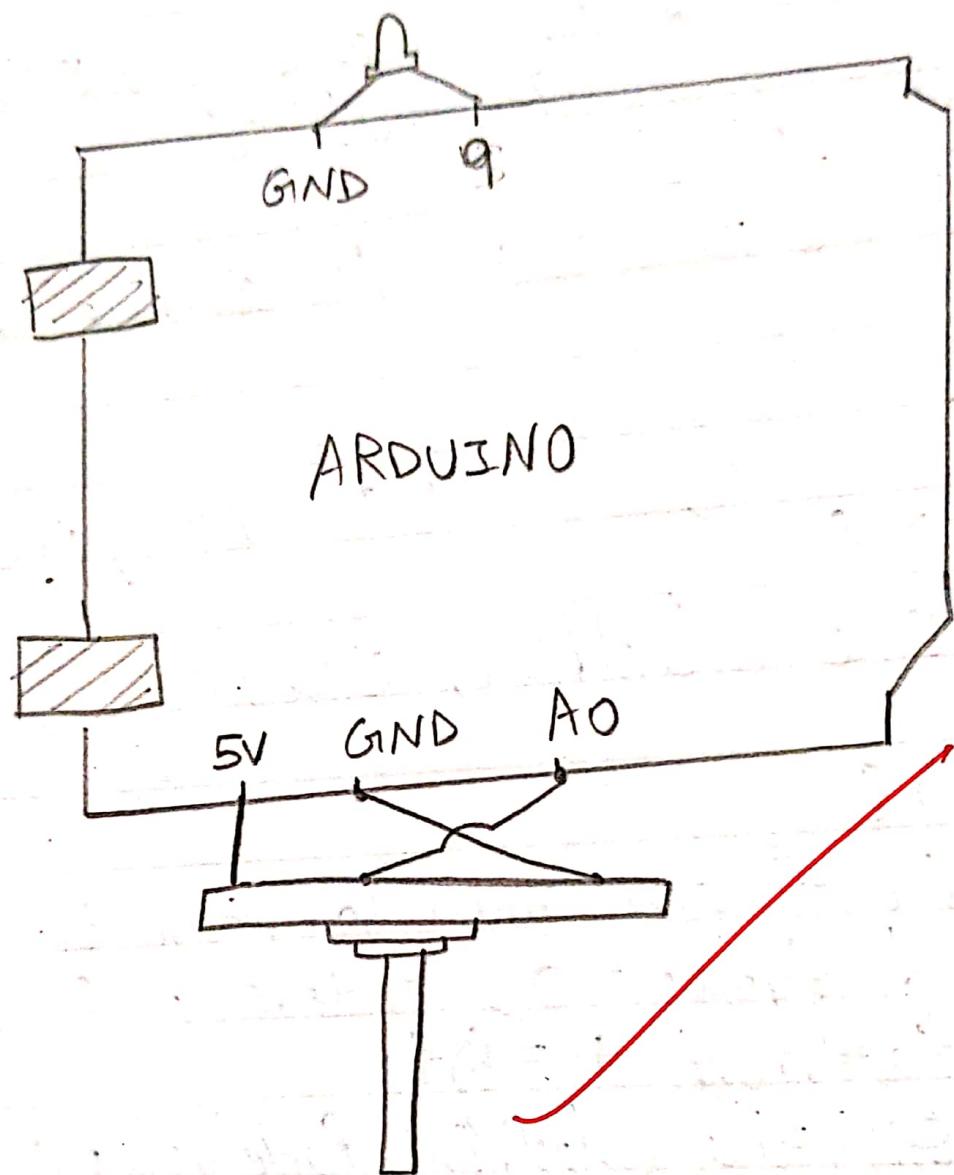


CIRCUIT:



- OBSERVATION: The led glows up and fades out periodically.

- WITH POTENTIOMETER

- AIM: To demonstrate fading of LED using potentiometer.

- COMPONENTS REQUIRED: Arduino board, LED, potentiometer

- CODE :

```
const int analogPinIn = A0;
const int analogPinOut = 9; // PWM pin
int sensorValue = 0; // reads value from potentiometer
int outputValue = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    sensorValue = analogRead(analogPinIn);
    outputValue = map(sensorValue, 0, 1023, 0, 255);
    analogWrite(analogPinOut, outputValue);
    Serial.print("Sensor = ");
    Serial.print(sensorValue); // Serial.print("Output = ");
    Serial.print(outputValue);
    delay(2);
}
```

// end of loop

Date 19-8-19

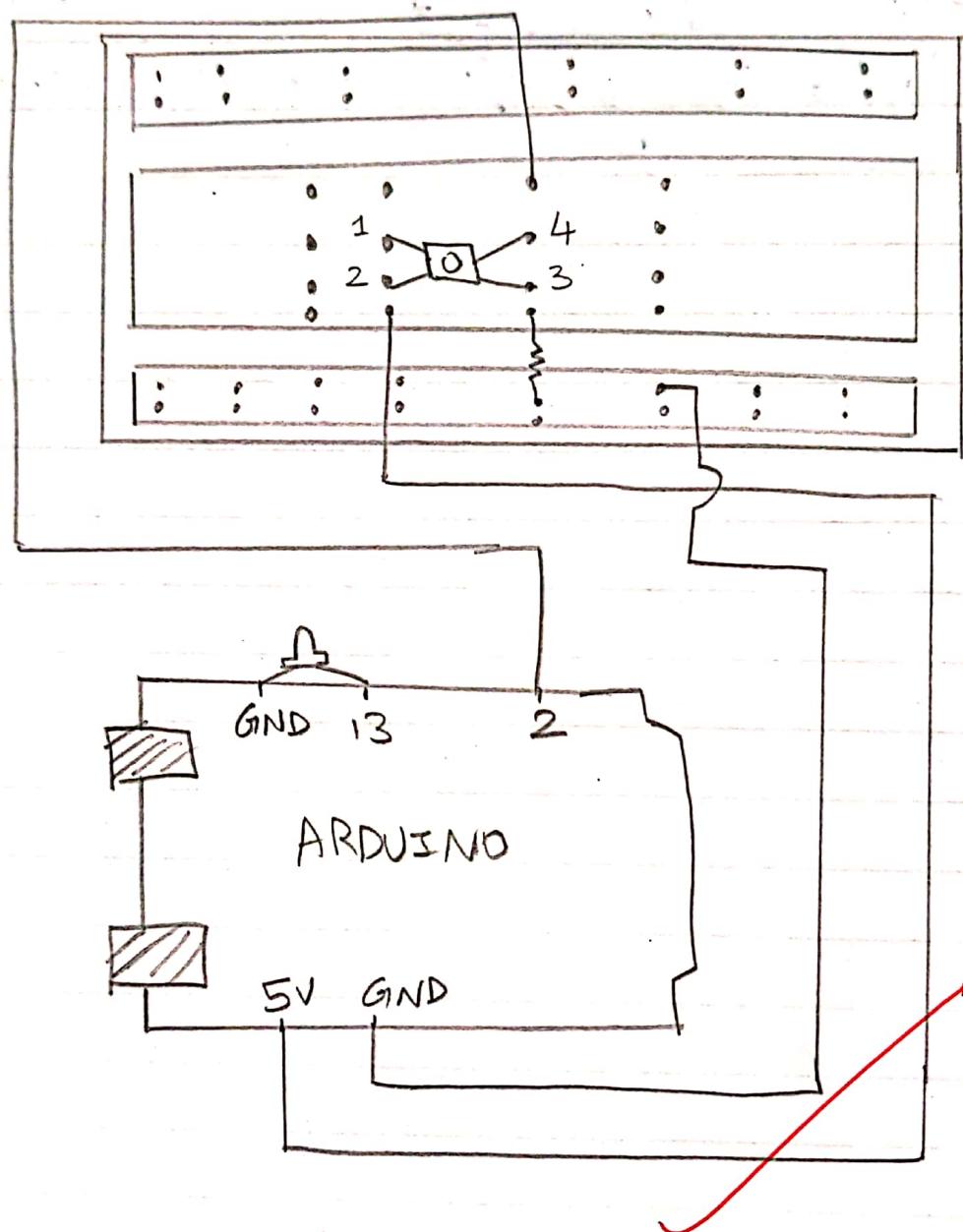
Expt. No. 02

Page No. 5

• OBSERVATION :

By rotating the shaft LED will glow and fade automatically of Potentiometer.

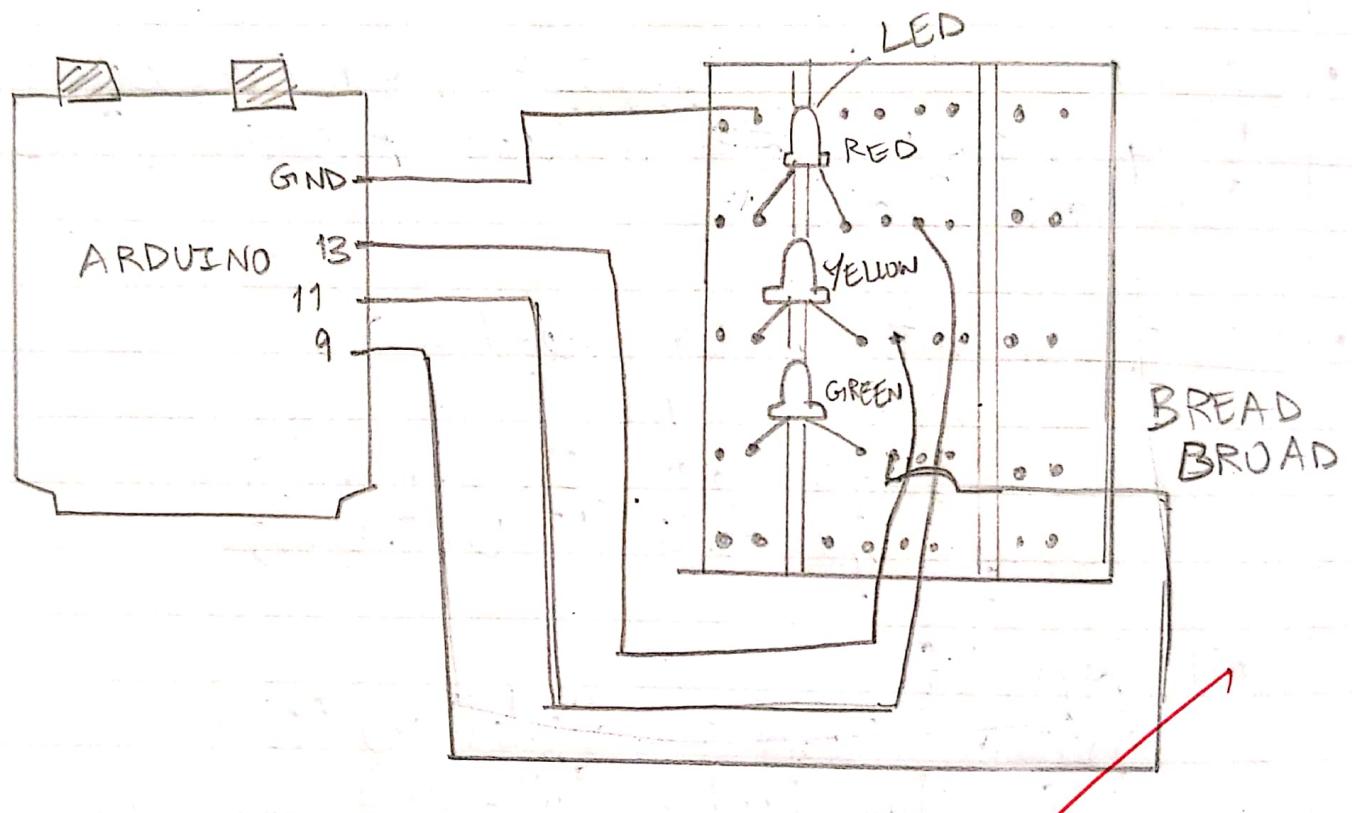
CIRCUIT:



- AIM : To demonstrate LED on/off using Push Button.
- COMPONENTS REQUIRED : Arduino Board, LED, USB Cable, ~~PC~~, Bread Board, Push Button, Resistor
- CODE :

```
int buttonRead = 0;
void setup() {
    pinMode(2, INPUT);
    pinMode(13, OUTPUT);
}
void loop() {
    buttonRead = digitalRead(2);
    if (buttonRead == HIGH)
        digitalWrite(13, HIGH);
    else
        digitalWrite(13, LOW);
}
```
- OBSERVATION : When the button is pressed, the LED glows, and on releasing the button, the LED is off.

CIRCUIT:



- AIM: Design and implement traffic light controller using LEDs.
- COMPONENTS REQUIRED: Arduino board, LEDs (red, green, yellow), USB cable, Bread board, connecting wires.
- CODE:

```

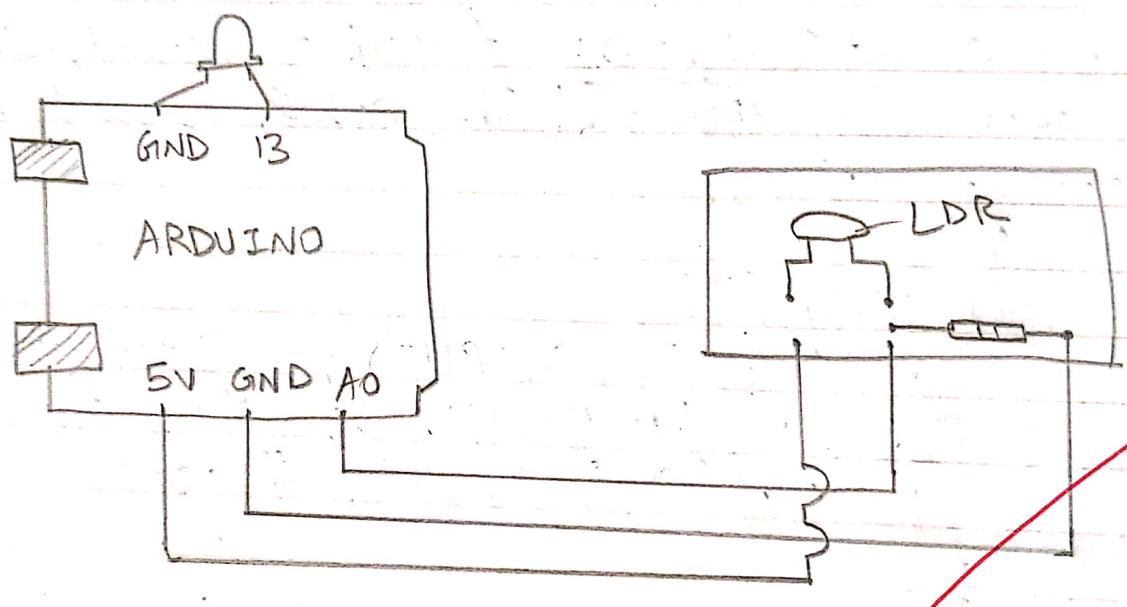
void setup() {
    pinMode(13, OUTPUT); // red led
    pinMode(11, OUTPUT); // yellow led
    pinMode(9, OUTPUT); // green led
}

void loop() {
    digitalWrite(13, HIGH); // red led glows
    digitalWrite(11, LOW);
    digitalWrite(9, LOW);
    delay(10000);
    digitalWrite(13, LOW);
    digitalWrite(11, LOW);
    digitalWrite(9, HIGH); // green led glows
    delay(5000);
    digitalWrite(13, LOW);
    digitalWrite(11, HIGH); // yellow led glows
    digitalWrite(9, LOW);
    delay(1000);
}

// End of loop

```
- OBSERVATION: Traffic lights is simulated.

CIRCUIT:



- AIM: Design and program smart light using LDR sensor and LED.

- COMPONENTS REQUIRED: Arduino board, LED, USB Cable, LDR sensor, Resistor.

- CODE:

```

int LDR = A0;
int LDRValue = 0;
int light sensitivity = 1000; // threshold

void setup() {
    Serial.begin(9600);
    pinMode(11, OUTPUT);
}

void loop() {
    LDRValue = analogRead(LDR);
    Serial.println(LDRValue);
    delay(150);
    if (LDRValue < light sensitivity)
        digitalWrite(11, HIGH);
    else
        digitalWrite(11, LOW);
    delay(1000);
}
} // end of loop

```

- OBSERVATION: 865

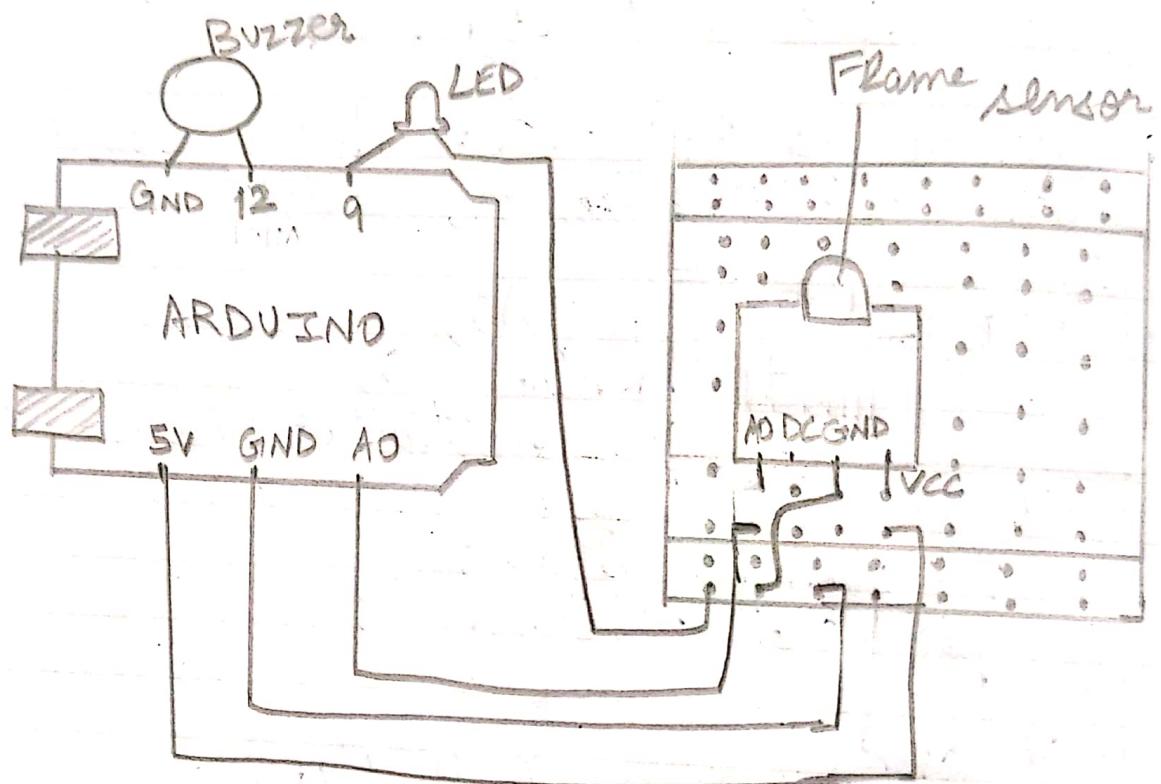
869

1006 // LED glows

901

1.

- CIRCUIT :



FIRE ALERT

- AIM : To design and demonstrate fire alert using Flame sensor
- COMPONENTS REQUIRED : Arduino board, LED, Flame sensor, Buzzer, bread board
- CODE :

```
const int analogPinIn = A0;
const int buzzerPin = 12;
const int ledPin = 9;
int sensorValue = 0;
int threshold = 100; //based on experiment

void setup() {
    pinMode(buzzerPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("Welcome");
    sensorValue = analogRead(analogPinIn);
    Serial.println(sensorValue);
    if (sensorValue < threshold) {
        Serial.println("Fire Detected");
        Serial.println("LED on");
        digitalWrite(ledPin, HIGH);
        digitalWrite(buzzerPin, HIGH);
        delay(1000);
    }
}
```

Teacher's Signature : _____

Date 16-9-19

Expt. No. 06

Page No. 10

```
digitalWrite (ledPin, LOW);  
digitalWrite (buzzerPin, LOW);  
delay (sensorValue);  
} //end of loop
```

• OBSERVATION:

1013

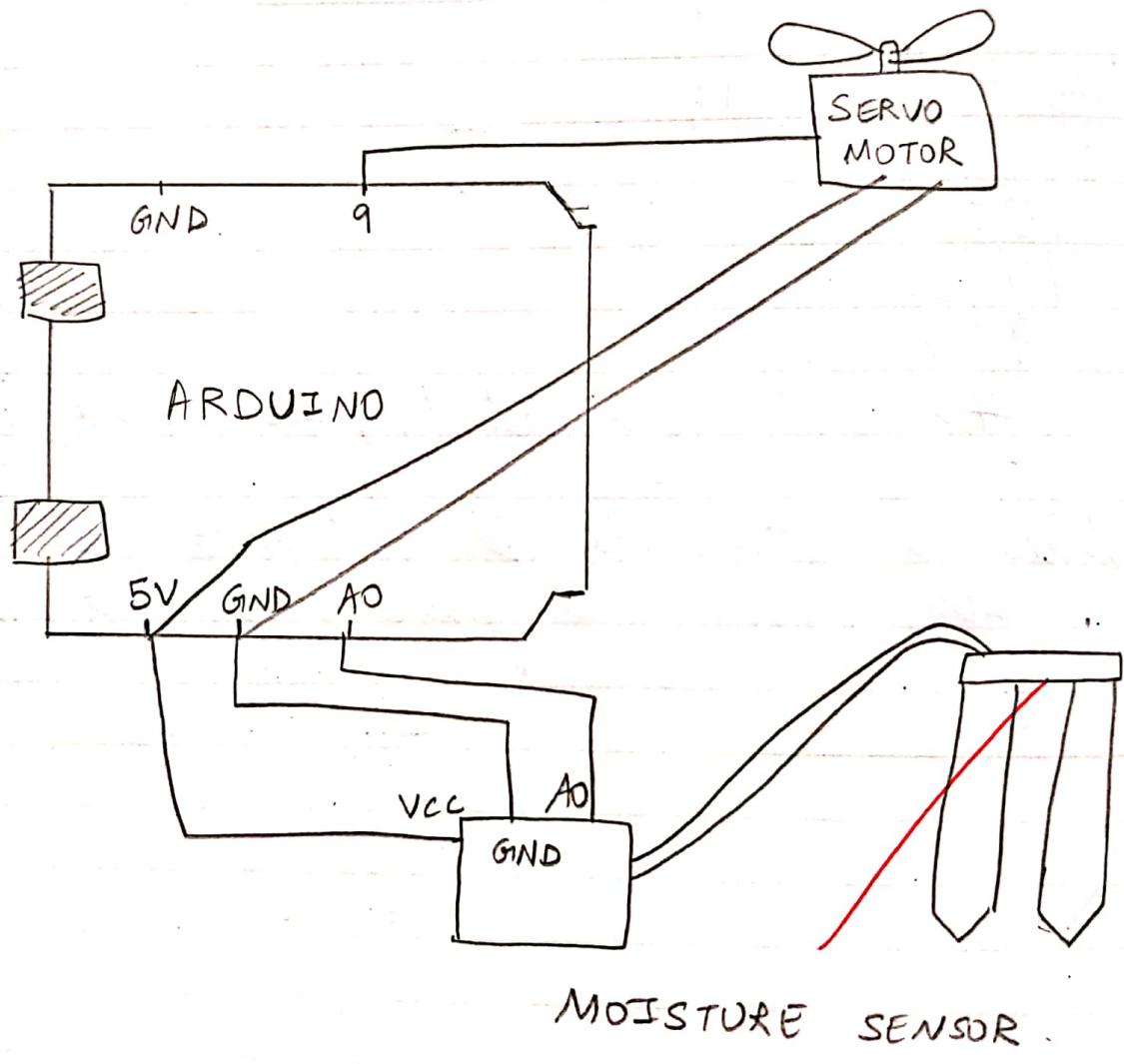
1002

980

40 // Fire Detected, LED glows, Buzzer on

When a flame is brought near the flame sensor, the led glows and the buzzer rings.

• CIRCUIT:



SOIL MOISTURE SENSOR

- AIM: To demonstrate soil moisture sensor.
- COMPONENTS REQUIRED: Arduino, servo motor, bread board, cup of water, moisture sensor

CODE :

```
#include <Servo.h>
Servo myservo; //servo object
int pos=0; //angle of rotation
int sensorPin = A0;
int sensorValue = 0;
void setup() {
    myservo.attach(9);
    Serial.begin(9600);
}
void loop() {
    sensorValue = analogRead(sensorPin);
    Serial.println(sensorValue);
    if (sensorValue > 500){ //threshold = 500
        for (pos=0; pos <= 180; pos += 1)
        { //goes from 0° to 180° in steps of 1°
            myservo.write(pos);
            delay(15);
        }
        for (pos = 180; pos >= 0; pos -= 1)
        { myservo.write(pos);
    }
}
```

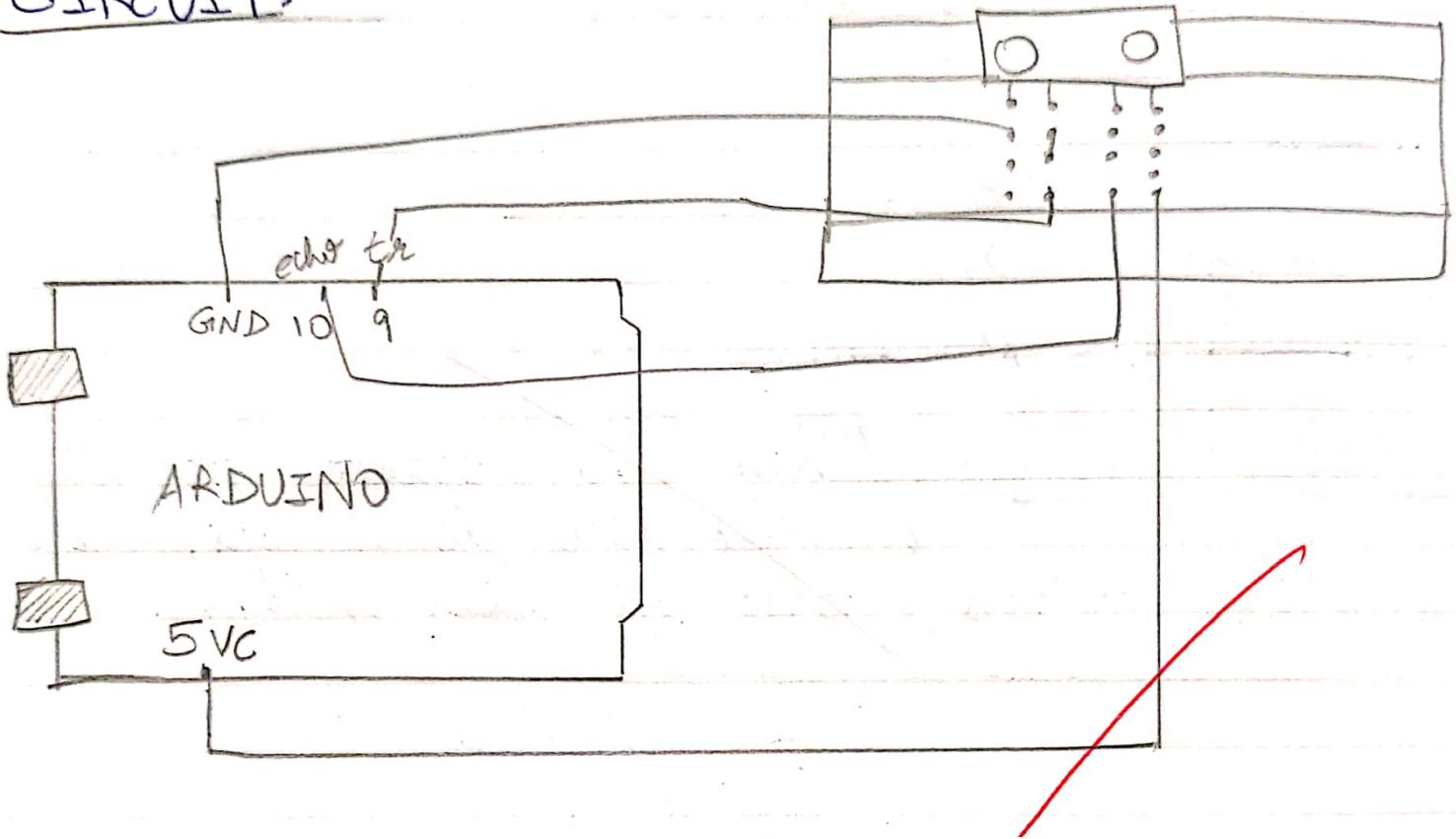
Teacher's Signature : _____

```
mysero.write( pos );  
delay( 15 );
```

```
} //end of loop if  
delay( 1000 );  
} //end of loop
```

- OBSERVATION : When the ~~water~~ moisture in the soil is less, the motor runs else it does not.

- CIRCUIT:



Ultrasonic Sensor

- AIM: To measure the distance between the sensor and objects placed in front of it.
- COMPONENTS REQUIRED: Ultrasonic sensor, wires, Arduino board.
- CODE:

```

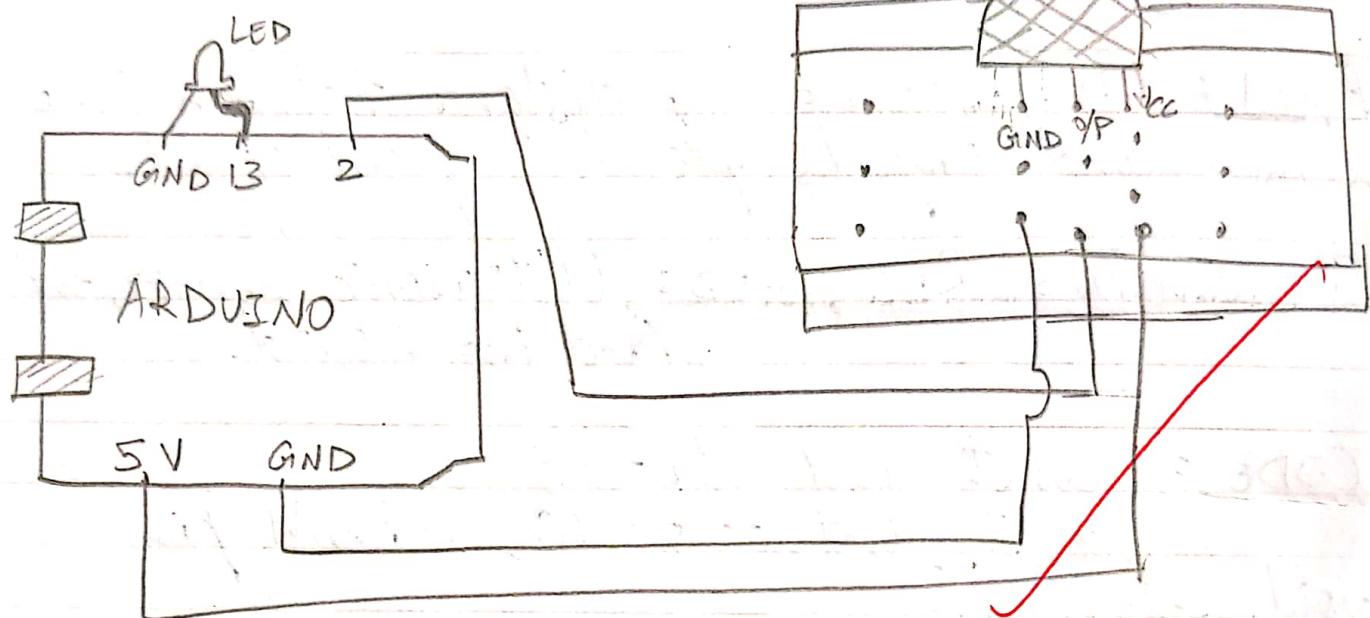
const int trig = 9;
const int echo = 10; // PWM pin

void setup() {
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    Serial.begin(9600);
}

void loop() {
    // distance = duration / 29 / 2;
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    long duration = pulseIn(echo, HIGH);
    double distance = duration * 0.034 / 2;
    Serial.print("In Distance : ");
    Serial.print(distance);
}

```
- OBSERVATION: The distance between the object and sensor is shown in the serial monitor.

CIRCUIT:



PIR Sensor

- AIM: To detect the human motion in a given area using PIR sensor.
- COMPONENTS REQUIREMENT: Arduino board, PIR sensor, led, wires, bread board
- CODE:

```

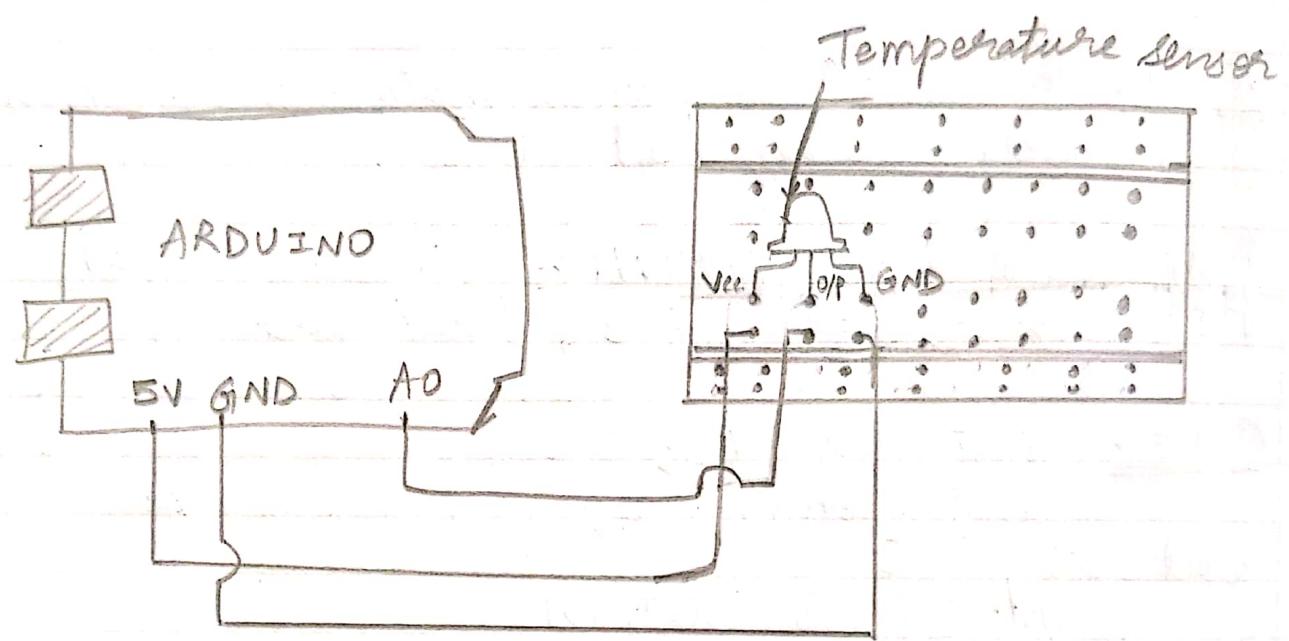
int led = 13;
int sensor = 2;

void setup() {
    pinMode(led, OUTPUT);
    pinMode(sensor, INPUT);
    Serial.begin(9600);
}

void loop() {
    val = digitalRead(sensor);
    digitalWrite(led, val);
    Serial.println(val);
    if (val == 0)
        Serial.println("Where are you ?");
    else
        Serial.println("Hey ! I got you !");
}

```
- OBSERVATION: When a human motion is detected the serial monitor displays: "Hey ! I got you !", so, it displays, "Where are you ?" ↙ ↘ ↙ ↘

• CIRCUIT :



TEMPERATURE SENSOR

- AIM: To use temperature sensor to display the temperature of the surrounding area.
- COMPONENTS REQUIRED: Arduino, temperature sensor, bread board, connecting wires
- CODE:

```

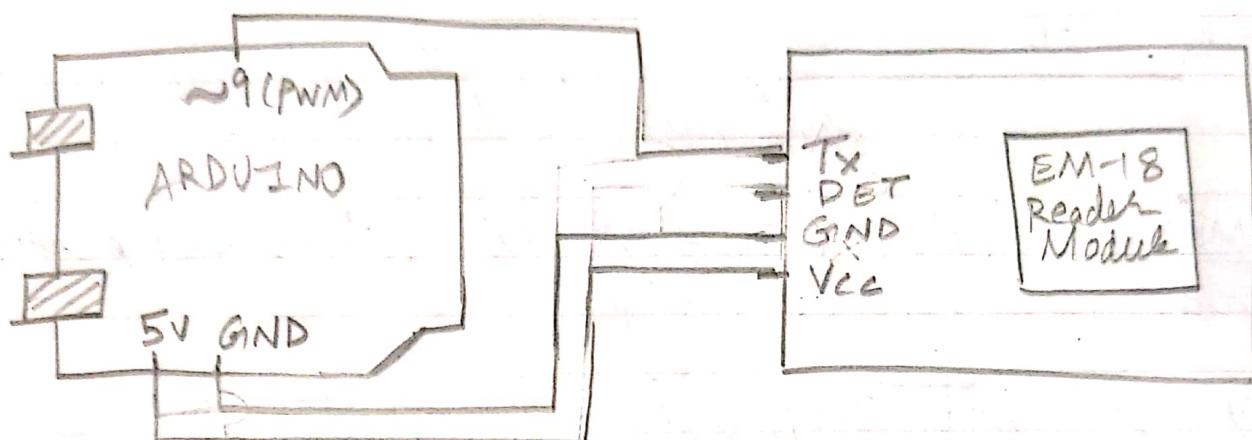
void setup() {
    Serial.begin(9600);
}

void loop() {
    int rawVoltage = analogRead(A0);
    // Conversion of sensor value (rawVoltage) to temperature value
    float millivolts = (rawVoltage / 1024.0) * 5000;
    float celsius = millivolts / 10; // scale factor for LM35
    Serial.print(celsius);
    Serial.print(" degrees Celsius, ");
    Serial.print((celsius + 9) / 5 + 32); // °F = °C * 9/5 + 32
    Serial.println(" degrees Fahrenheit");
    delay(1000);
}

```

- OBSERVATION: The temperature sensor senses the temperature of the surrounding area which is displayed on the serial monitor.

- CIRCUIT:



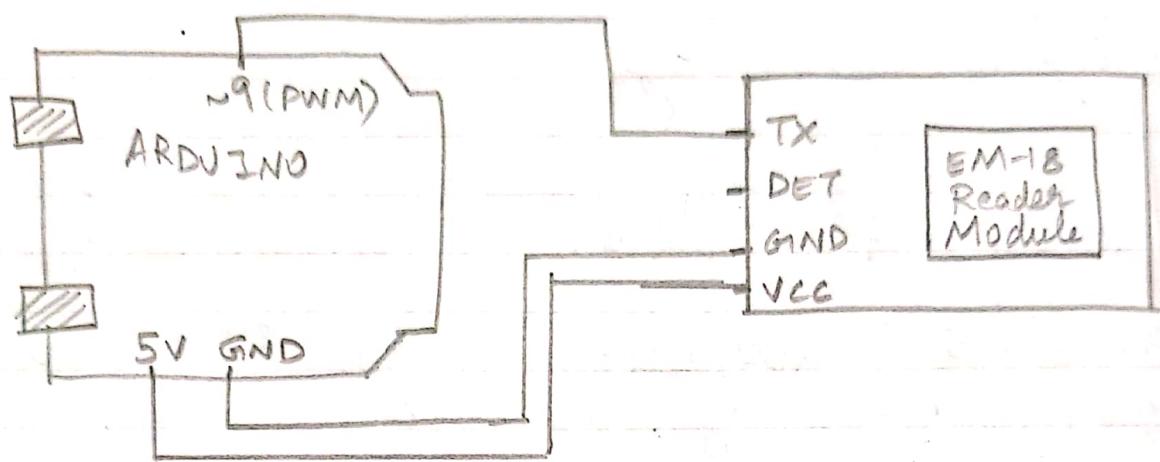
RFID Reader and RFID Tag Count

- Reading a Single Tag
- AIM: To read a single RFID tag using Arduino and RFID module.
- COMPONENTS REQUIRED: Arduino, RFID Reader Module, RFID Tag, connecting wires.
- CODE:

```
#include <SoftwareSerial.h>
int rxpin = 9, txpin = 10;
SoftwareSerial myserial(rxpin, txpin);
void setup() {
    myserial.begin(9600); //Setting baud rate of Software Serial Library
    Serial.begin(9600); //Setting baud rate of Serial Monitor
}
void loop() {
    if (myserial.available() > 0)
        Serial.write(myserial.read());
}
```
- OBSERVATION: The RFID tag is read by the RFID Reader module and is displayed on the Serial monitor.

Teacher's Signature : _____

- CIRCUIT:



• Reading Multiple Tags

• AIM: To read multiple RFID tags using Arduino and RFID module and display them.

• COMPONENTS REQUIRED: Arduino, RFID Reader module, RFID tags.

• CODE:

```
#include <SoftwareSerial.h>
int rxpin = 9, txpin = 10;
SoftwareSerial myserial(rxpin, txpin);
int readcount = 0, tagcount = 0;
int j = 0, k = 0; // loop variables
char data_temp, RFID_data[12], data_store[10][12];
boolean disp_control;
void setup() {
    myserial.begin(9600);
    Serial.begin(9600);
}
```

```
void loop() {
    ReceiveData();
    StoreData();
    PrintData();
}
```

```
void ReceiveData() {
    if (myserial.available() > 0) {
```

```

data_temp = mySerial.read();
RFID_data[readcount] = data_temp;
readcount++;
} //end of if
} //end of method

```

```

void storeData() {
    if (readCount == 12) {
        dispControl = true;
        for (k = tagCount; k <= tagCount; k++)
            for (j = 0; j < 12; j++)
                dataStore[k][j] = RFID_data[j];
        readCount = 0;
        tagCount++;
    } //end of if
} //end of method

```

```

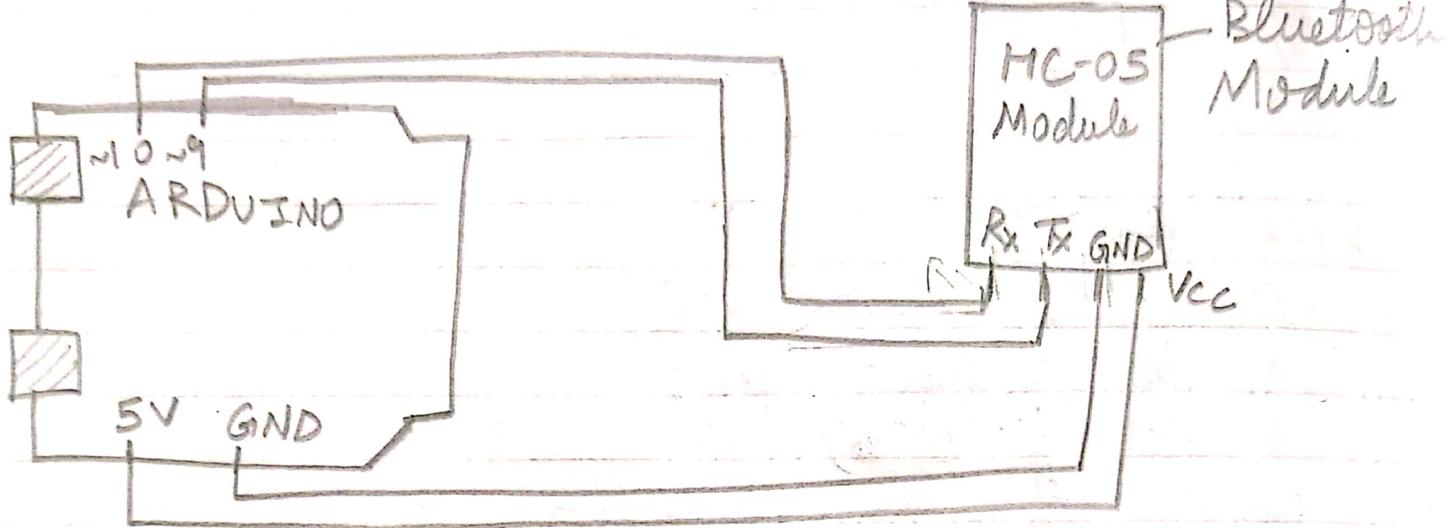
void PrintData() {
    if (dispControl == true) {
        for (k = 0; k <= tagCount; k++) {
            for (j = 0; j < 12; j++)
                Serial.write(dataStore[k][j]);
            Serial.println();
        } //end of loop
        dispControl = false;
    } //end of if
} //end of method

```

- OBSERVATION: Multiple RFID tags are read & displayed

Teacher's Signature : _____

CIRCUIT:



BLUETOOTH MODULE

- AIM: To connect to bluetooth module and demonstrate basic AT commands.
- COMPONENTS REQUIRED: Arduino, HC-05 module, bread board, connecting wires.
- CODE:

```

#include <SoftwareSerial.h>
int rxpin = 9, txpin = 10;
SoftwareSerial BTserial(rxpin, txpin); // RX | TX

void setup() {
    Serial.begin(9600);
    Serial.println("Enter AT commands:");
    BTserial.begin(38400); // HC-05 default speed in
} // AT command mode

void loop() {
    // Keep reading from HC-05 & send to Arduino Serial
    // monitor
    if (BTserial.available())
        Serial.write(BTserial.read());
}

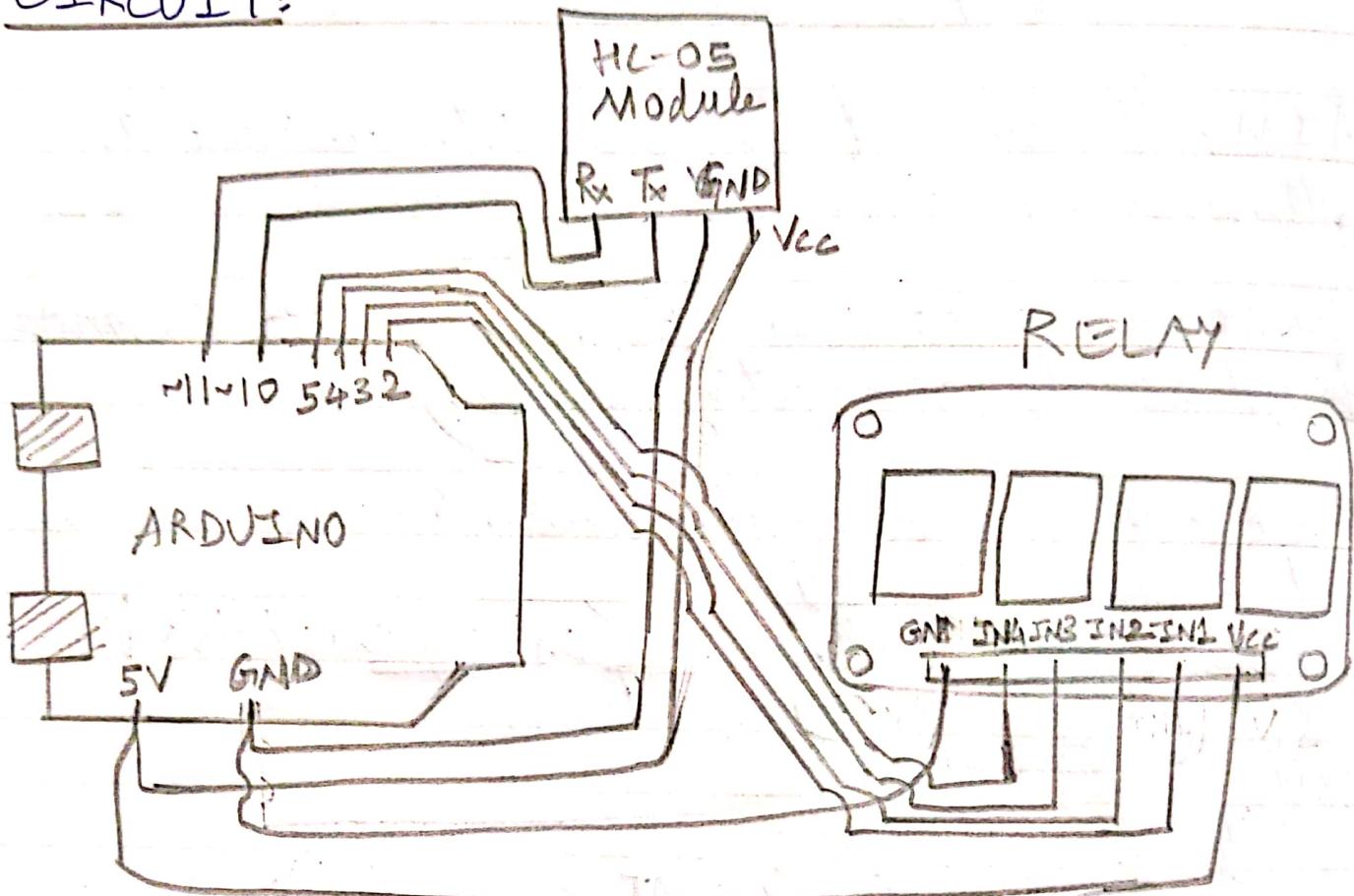
// Keep reading from Arduino serial monitor & send to HC-05
if (Serial.available())
    BTserial.write(Serial.read());
}

```

// End of loop
- OBSERVATION: Basic AT commands were executed.

Teacher's Signature : _____

• CIRCUIT:



Relay	IN1	→	2	Arduino
Relay	IN2	→	3	Arduino
Relay	IN3	→	4	Arduino
Relay	IN4	→	5	Arduino

HOME AUTOMATION USING BLUETOOTH AND RELAY

- AIM: To control working of relay (connected to LED's) through Arduino using Android mobile using HC-05 module and Arduino.
- COMPONENTS REQUIRED: Arduino, HC-05 module, 4-channel relay, bread board, connecting wires, Android phone.
- CODE:

```
#include <SoftwareSerial.h>
int expin = 10, txpin = 11;
SoftwareSerial BTserial(expin, txpin);
// Pin 10 RX, Pin 11 TX connected to → Bluetooth TX, RX
#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);
```

Teacher's Signature: _____

```
digitalWrite(relay3, HIGH);  
digitalWrite(relay4, HIGH);
```

```
BTSerial.begin(9600); // or 38400 depending on module  
Serial.begin(9600);
```

} // end

```
void loop() {
```

```
if (BTSerial.available() > 0) {  
    val = BTSerial.read();  
    Serial.println(val);  
}
```

// Relay is on

```
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);
```

// Relay all on

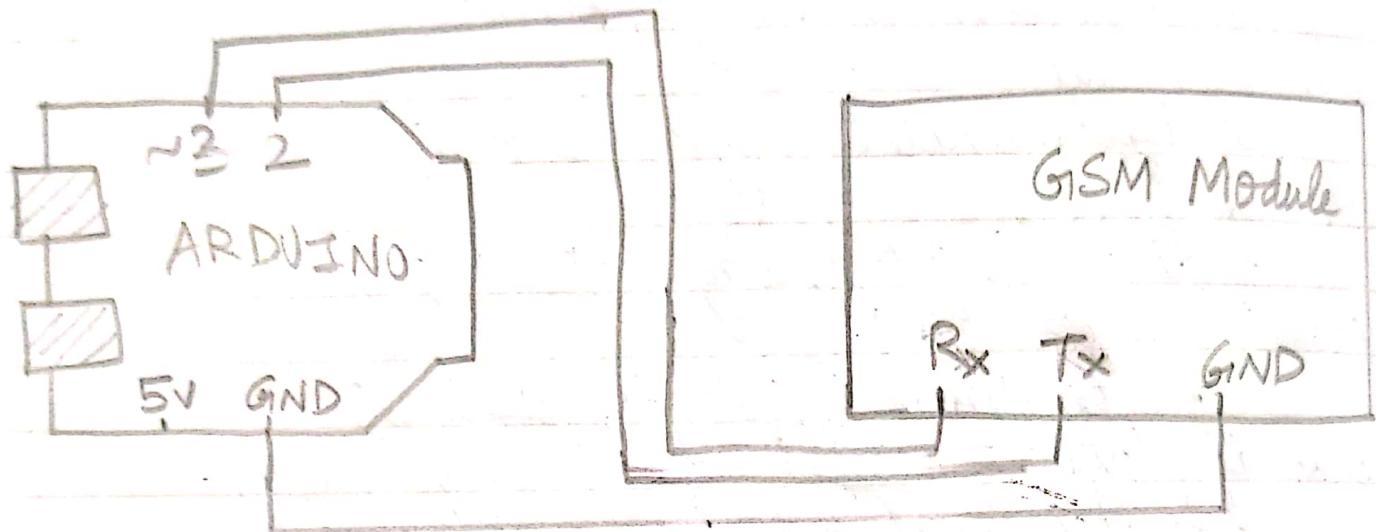
```
else if (val == '9') {  
    digitalWrite(relay1, LOW);  
    digitalWrite(relay2, LOW);  
    digitalWrite(relay3, LOW);  
    digitalWrite(relay4, LOW);  
}
```

```
//relay is off  
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') { //relay all off  
    digitalWrite(relay1, HIGH);  
    digitalWrite(relay2, HIGH);  
    digitalWrite(relay3, HIGH);  
    digitalWrite(relay4, HIGH);  
}
```

//end of loop.

- OBSERVATION: The relay is controlled through Android phone using Arduino and Bluetooth

CIRCUIT:



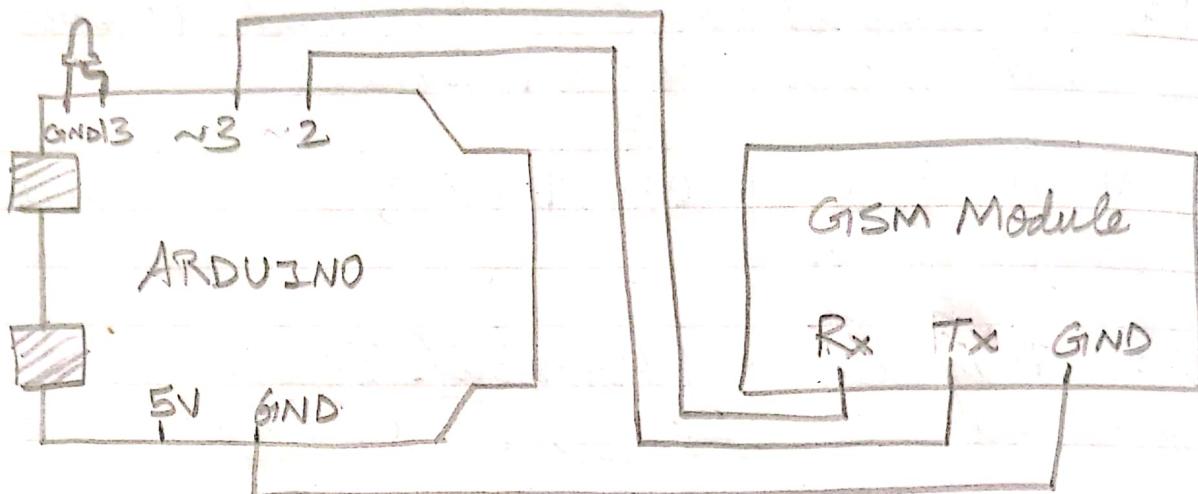
GSM Module

- AIM: To demonstrate connection and working of GSM module using Arduino.
- COMPONENTS REQUIRED: Arduino, GSM module, connecting wires.
- CODE: #include <SoftwareSerial.h>
SoftwareSerial cell(2,3); //Rx, Tx

```
void setup() {  
    cell.begin(9600);  
    delay(500);  
    Serial.begin(9600);  
    Serial.println("CALLING....");  
    cell.println("ATD +919880424607");  
    //ATD- Attention Dial  
    delay(20000);  
}  
void loop() { }
```

- OBSERVATION: The GSM module calls the number specified in the code.

- CIRCUIT:



CONTROLLING LED USING GSM MODULE

- AIM: To control LED using GSM Module and Arduino.
- COMPONENTS REQUIRED: Arduino, GSM module, LED, Connecting wires.
- CODE :

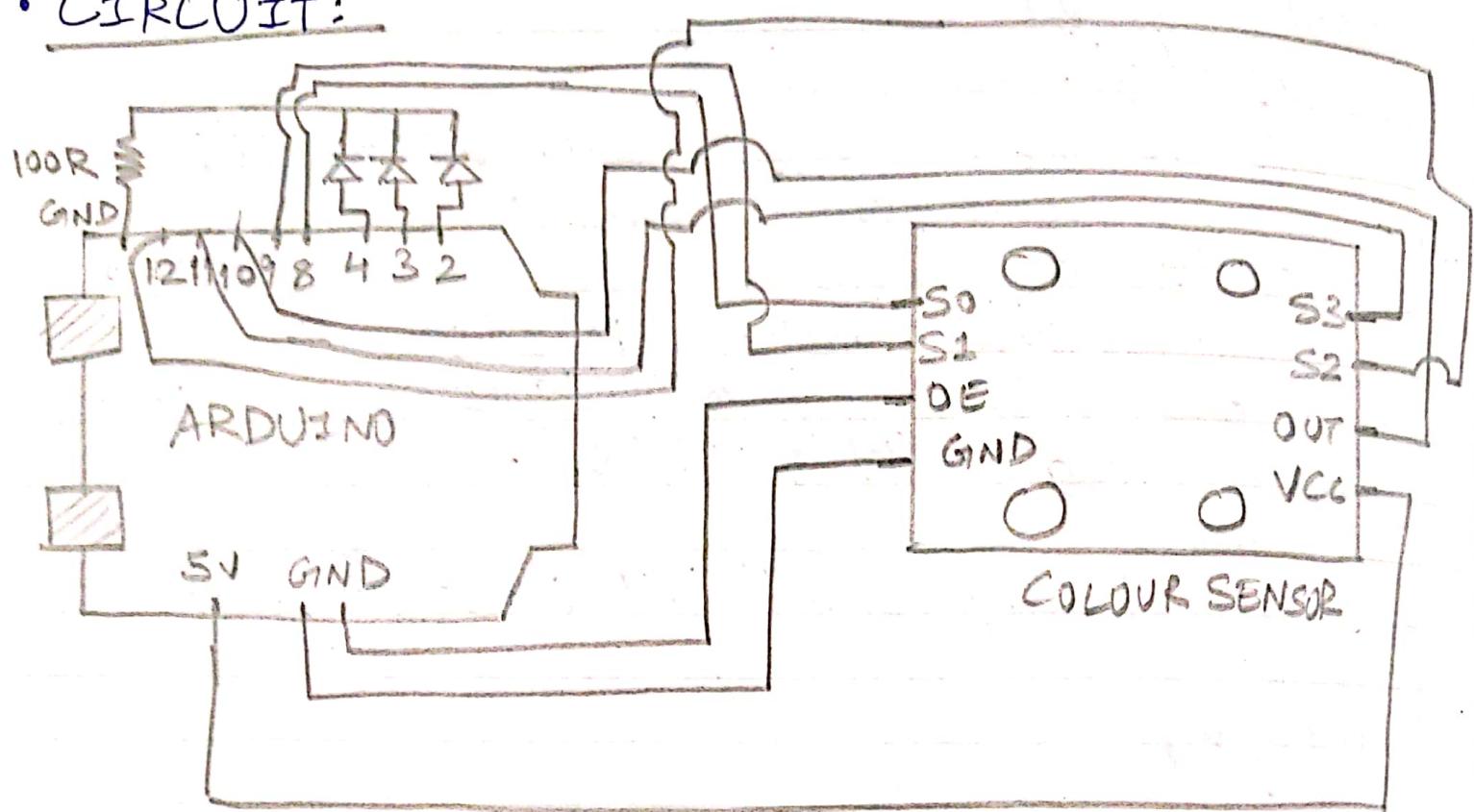
```
#include <SoftwareSerial.h>
int arpin = 2, txpin = 3;
SoftwareSerial cell(arpin, txpin); // RX | TX
void readfn() {
    if (cell.available() > 0) {
        while (cell.available())
            Serial.write(cell.read());
    }
}
void setup() {
    pinMode(13, OUTPUT);
    Serial.begin(9600);
    cell.begin(9600);
    cell.println("AT");
    delay(1000);
    readfn();
    // NEW SMS Alert
    cell.println("AT+CNMI=1,2,0,0,0");
}
```

Teacher's Signature : _____

```
void loop() {
    if (cell.available()) {
        String message = cell.readString();
        Serial.println(message);
        if (message.indexOf("SWITCH ON") > 0)
            digitalWrite(13, HIGH);
        else if (message.indexOf("SWITCH OFF") > 0)
            digitalWrite(13, LOW);
        else
            Serial.println("Nothing to do ...");
    }
}
```

- OBSERVATION: The LED turns on and off depending on the SMS sent to the GSM module.

CIRCUIT:



COLOUR SENSOR → ARDUINO

VCC	→	5V
GND	→	GND
S0	→	8
S1	→	9
S2	→	12
S3	→	11
OUT	→	10
OE	→	GND

S2	S3	Color
L	L	Red
L	H	Blue
H	L	Clear
H	H	Green

COLOR SENSOR

- AIM : To identify the RGB Colors using Color sensor
- COMPONENTS REQUIRED : Arduino, RGB LED, Resistor 100 Ω , Colour sensor TCS230, bread board, connecting wires.
- CODE :

```

const int s0 = 8;
const int s1 = 9;
const int s2 = 12;
const int s3 = 11;
const int out = 10;

//LED pins connected to Arduino
int redLed = 2;
int greenLed = 3;
int blueLed = 4;

// Variables
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);
}

```

Teacher's Signature : _____

```
pinMode(redLed, OUTPUT);
pinMode(greenLed, OUTPUT);
pinMode(blueLed, OUTPUT);
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 && red < 20) {
        serial.println("- (Red Colour)");
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, LOW);
    } else if (blue < red && blue < green) {
        serial.println("- (Blue Color)");
        digitalWrite(redLed, LOW);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, HIGH);
    } else if (green < red && green < blue) {
        serial.println("- (Green color)");
    }
}
```

Teacher's Signature : _____

```
digitalWrite(redLed, LOW);  
digitalWrite(greenLed, HIGH);  
digitalWrite(blueLed, LOW);
```

}
else {

```
    Serial.println();
```

}
delay(300);

```
digitalWrite(redLed, LOW);  
digitalWrite(greenLed, LOW);  
digitalWrite(blueLed, LOW);
```

} // end of loop

void color() {

```
    digitalWrite(s2, LOW);  
    digitalWrite(s3, LOW);
```

red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
 digitalWrite(s3, HIGH);

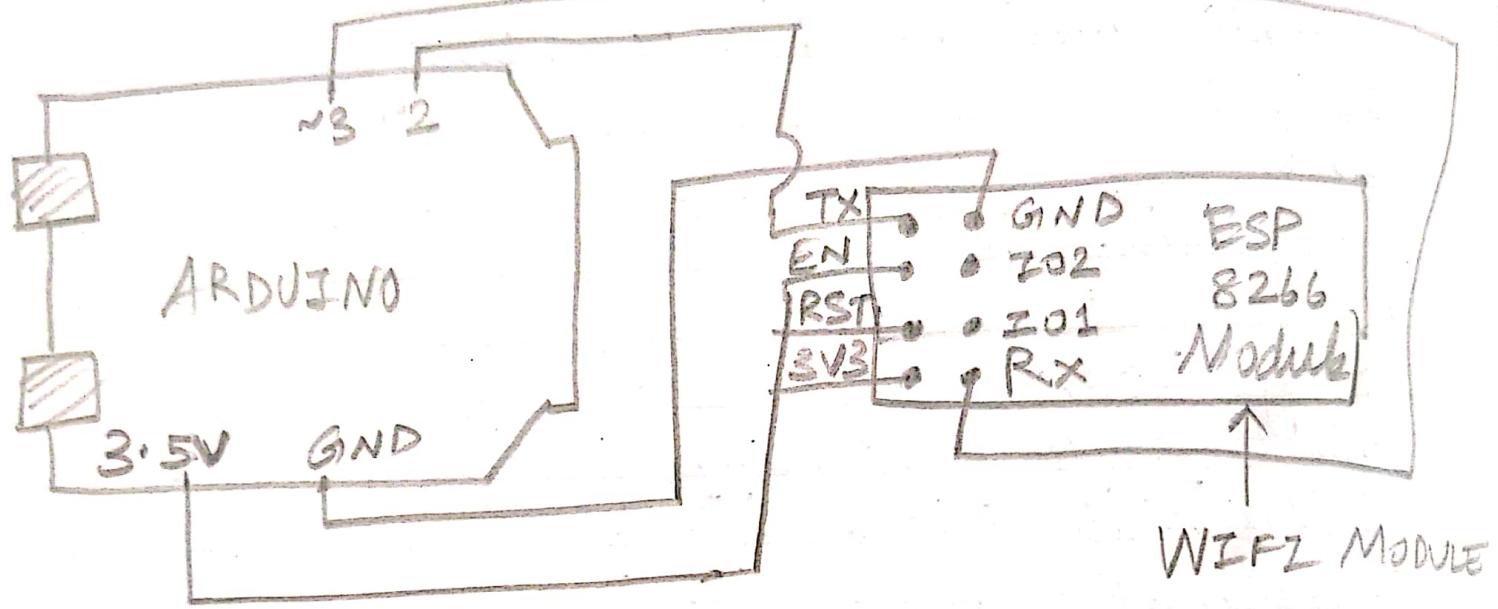
blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
 digitalWrite(s2, HIGH);

green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}

} // end

- OBSERVATION: The sensor senses colors which are indicated using the RGB LED.

CODE CIRCUIT:



WIFI MODULE

• AIM: To demonstrate interfacing of Wifi ESP8266 module with Arduino.

• COMPONENTS REQUIRED: Arduino, Wifi ESP8266 module, connecting wires.

```
#include <SoftwareSerial.h>
SoftwareSerial wifi(2,3); // RX | TX
void setup() {
    pinMode(9, OUTPUT);
    Serial.begin(9600);
    wifi.begin(9600);
}
void loop() {
    if (Serial.available()) {
        while (Serial.available()) {
            wifi.write(Serial.read());
        }
    }
    if (wifi.available()) {
        while (wifi.available())
            Serial.write(wifi.read());
    }
}
```

• OBSERVATIONS: AT commands are entered & executed by Wifi module.

Teacher's Signature : _____