**30ARDUINO**

1.Two parts-setup and loop

2.setup executes once and loops executes as mentioned.

3.Blinkning the built in LED is the basic program of arduino.

4.Input LED is connected to the 13th pin of digital out.

**Blinking LED:**

1.First initalize digital pin 13 as output in the code.

2.There are two levels:HIGH and LOW

HIGH-switches on the led with 5v

LOW-switches off the led

NOTE: Make sure the idle is connected to the proper board in this case it is the arduino unweo.

**CODE:**

1.First write the code in the idle of arduino.

2.Then save the code.

3.The code gets saved as ino file(folder).

4.Then compile the code.

5.Now click the upload button so that the code gets uploaded in the arduino board and it runs.

NOTE: delay(1000) means that there will be a delay of 1 second.

here the parameter of delay is the milliseconds.

NOTE: The default pin of the inbuilt led of the Arduino board is the 13th pin.

(1 second=1000 milliseconds)

**BREADBOARD(NOTES):**

1.vertical pins of the middle section are of the same potential.

2.horizontal pins of the top and bottom sections are of the same potential.

**BLINKING OF EXTERNAL LED USING ARDUINO(connection to pin 13)**

1.From 13th pin of arduino borad connect to any pin of the breadboard.

2.take one pin of the resistor and connect to the same vertical line of the 13th pin that u connected before.(vertical pins of the same potential(in the middle section))

3.the other is connected to another vertical line of different potential.

4.Connect the positive side of the LED(longer wire of the LED) to the other pin of the resistor(the same vertical line of the other pin)

5.From the negative side of the LED connect to the ground of the arduino.

NOTE: If we want to see the output at a different pin we are supposed to change the code where pin 13 has been mentioned and mention your respective pin(EG:pin 10)

now make your connections to pin 10(one of the pins of the resistor is connected to the same line as that of pin 10)

Now compile the code and upload and run:

EXTERNAL LED BLINKS!!!!!

**Fading LED through code using Arduino**

1.Pulse width modulation means that we can change the brightness of the LED.

2.Brightness of the led can be changed only when u connect to certain pins which tilt(~) symbol [pins:3,5,6,9,10,11]

3.The input wire (the one which has the same connection as one of the pins of the resistor) must be connected to one of the tilt pins.

4.The same connections as the above experiment and the LED glows.

//Initializing LED Pin

int led\_pin = 9;

void setup() {

  //Declaring LED pin as output

  pinMode(led\_pin, OUTPUT);

}

void loop() {

  //Fading the LED

  for(int i=0; i<255; i++){

    analogWrite(led\_pin, i);

    delay(5);

  }

  for(int i=255; i>0; i--){

    analogWrite(led\_pin, i);

    delay(5);

  }

}

NOTE: Pulse width modulation is possible only when u connect to certain pins which have tilt(~) symbol [pins:3,5,6,9,10,11]. The input wire (the one which has the same connection as one of the pins of the resistor) must be connected to one of the tilt pins.

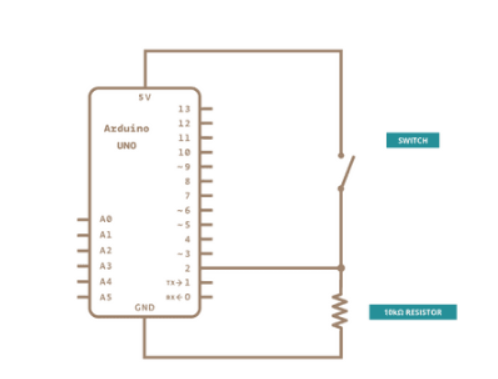
NOTE: In the for loop of the code we have a range of 0-255 which specifies the brightness of the LED.

NOTE: the range of 0-255 mentions the level of brightness of the LED.

Higher the number mentioned as the maximum limit in the for loop, the more brightness and longer the time taken to complete a cycle.(Time taken to complete an entire cycle is more because of the delay(5)+the time taken to complete the for loop)

When the maximum limit is mentioned as 255 the loop runs 255 times and thus more time is taken to execute this loop than time taken to complete a loop with a higher limit of 10.

**Push button and LDR Interfacing:**



1.Connect one end of the switch to 5v.

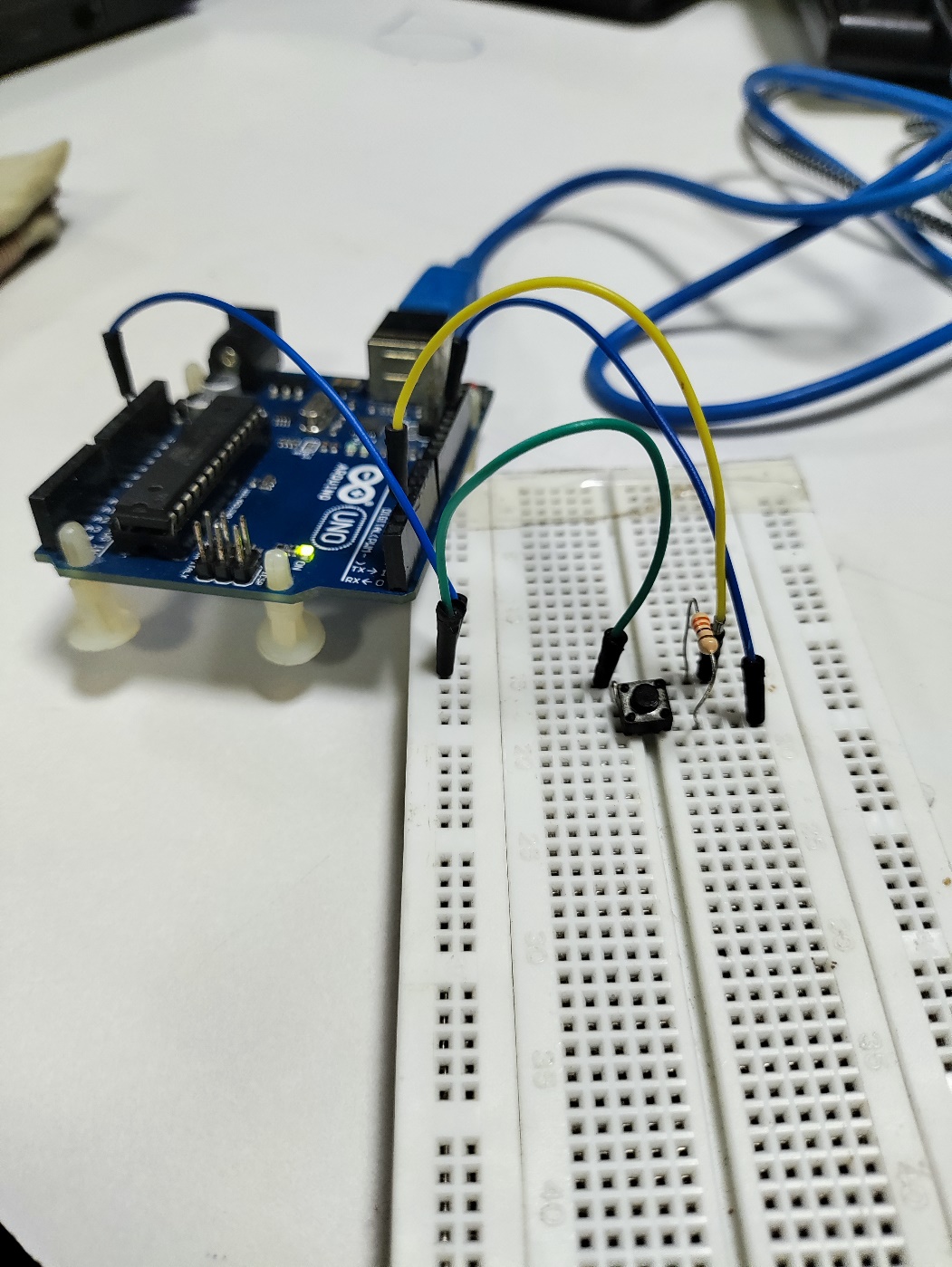
2.Connect other end of the switch to any one of the vertical lines of the middle section.

3.One end of the resistor is connected to the same above vertical line.

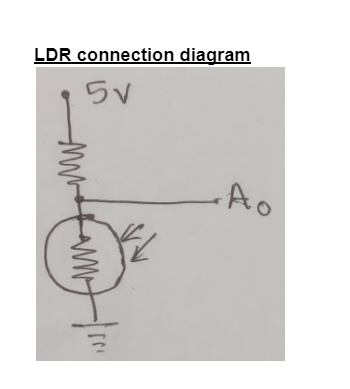
4.2v pin is also connected to a pin of the same vertical line

4.Other end of the resistor is connected to another vertical line which has the same potential as that of the ground.(one wire is connected from any pin from the same vertical line to ground)

5. Now connect the board to the computer and run and upload the code.



**LDR Interfacing:**

****

**CODE:**

// Interfacing Arduino uno with LDR sensor

const int ledPin = 13; // digital pin 5

const int ldrPin = A0; // analog pin 0

void setup() { // The setup() function will only run once, after each powerup or reset of the Arduino board.

Serial.begin(9600);

pinMode(ledPin, OUTPUT); // Here LED is determined as an output or an indicator.

pinMode(ldrPin, INPUT); // Here LDR sensor is determined as input.

}

void loop() { // Void loop is ran again and again and contains main code.

int ldrStatus = analogRead(ldrPin);

if (ldrStatus > 980) {digitalWrite(ledPin, HIGH); // If LDR senses darkness led pin high that means led will glow. Adjust the threshold according to the LDR readings

Serial.print("Darkness over here,turn on the LED :");

Serial.println(ldrStatus);

} else {

digitalWrite(ledPin, LOW); // If LDR senses light led pin low that means led will stop glowing.

Serial.print("There is sufficient light , turn off the LED : ");

Serial.println(ldrStatus);

}

}

**STEPS:**

1.A0-refers to the pin which is in the analog section of the Arduino board.

NOTE: For pulse width modulation we connect the pins to the other side which is digital + analog side.

STEPS:

1.connect 5V to resistor in the breadboard

2.other end of resistor(the other pins of the same vertical line) is connected to one end of LDR(light dependent resistor)

3.The other end of LDR is connected to ground.

4.A0 is connected to the same vertical line as that of (one terminal of resistor and one terminal of LDR)

**EXPLANATION:**

1.When the LDR is not blocked then the inbuilt LED does not glow.(this is because the light intensity is more and thus this means value is lesser than 980 which is seen in serial monitor)

2.When we block the LDR then the light intensity reduces and thus the value of ldrStatus becomes more than 980 and the inbuilt LED starts ti glow.

Application: This can be used to automatically operate the street lights according to the level of darkness.

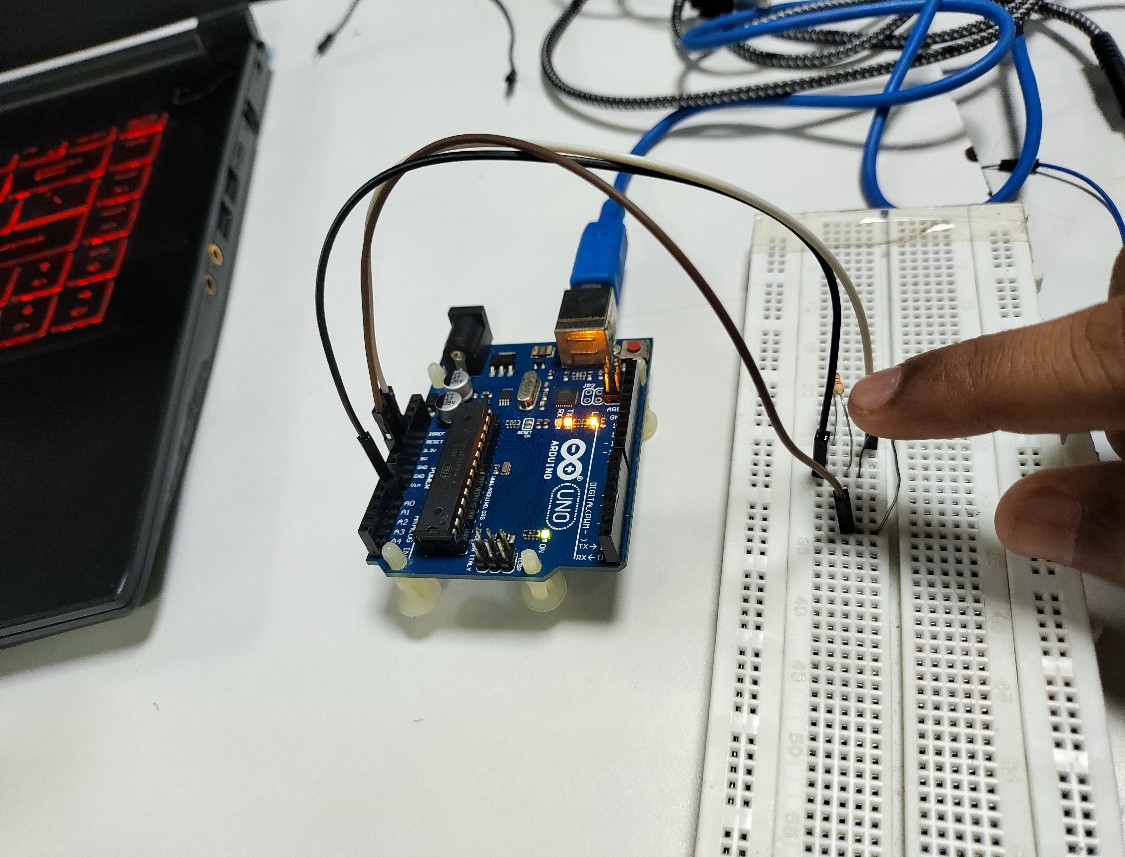


FIG1: when the LDR is blocked. Thus the inbuilt LED glows

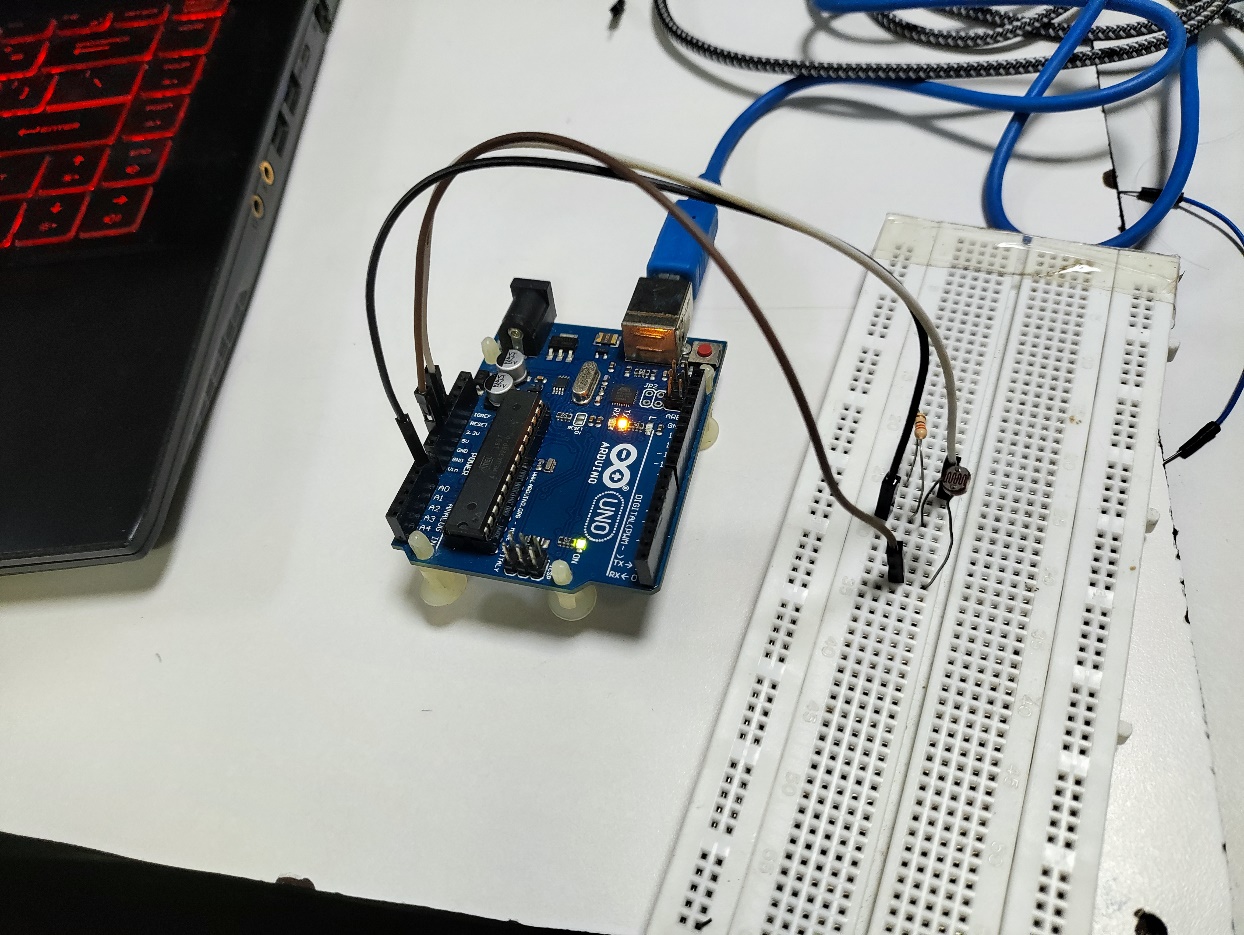


FIG2: The LDR is not blocked and thus there is high light intensity sensed. This causes the inbuilt LED TO not glow.

**IR Interfacing**

1.Connect the IR sensor to the breadboard.(IR sensor has 3 different ports).

2.The different ports of the IR sensor are mentioned behind it.(VCC,GND,OUT)

3.5V connection is given to VCC

4.ground is connected to ground.

5.A0 which is present on the left side of the Arduino board is given to output.

NOTE: The working of the IR sensor can be verified by the red light glowing in the IR sensor.

NOTE: IR sensors are usually used to detect motion. In this case when there is motion on top of the IR sensor then the inbuilt LED glows.

int IRSensor = A0; // connect IR sensor AO to arduino analog pin A0

int LED = 13; // connect Led to arduino pin 9

void setup()

{

int IRSensor = A0;

Serial.begin(9600); //initialise serial monitor

pinMode (LED, OUTPUT); // Led pin OUTPUT

}

void loop()

{

int statusSensor = analogRead (IRSensor);

Serial.println(statusSensor); //display output on serial monitor

if(statusSensor<100)

digitalWrite(LED,HIGH); //if light is not present,LED on

else

digitalWrite(LED,LOW); //if light is present,LED off

}

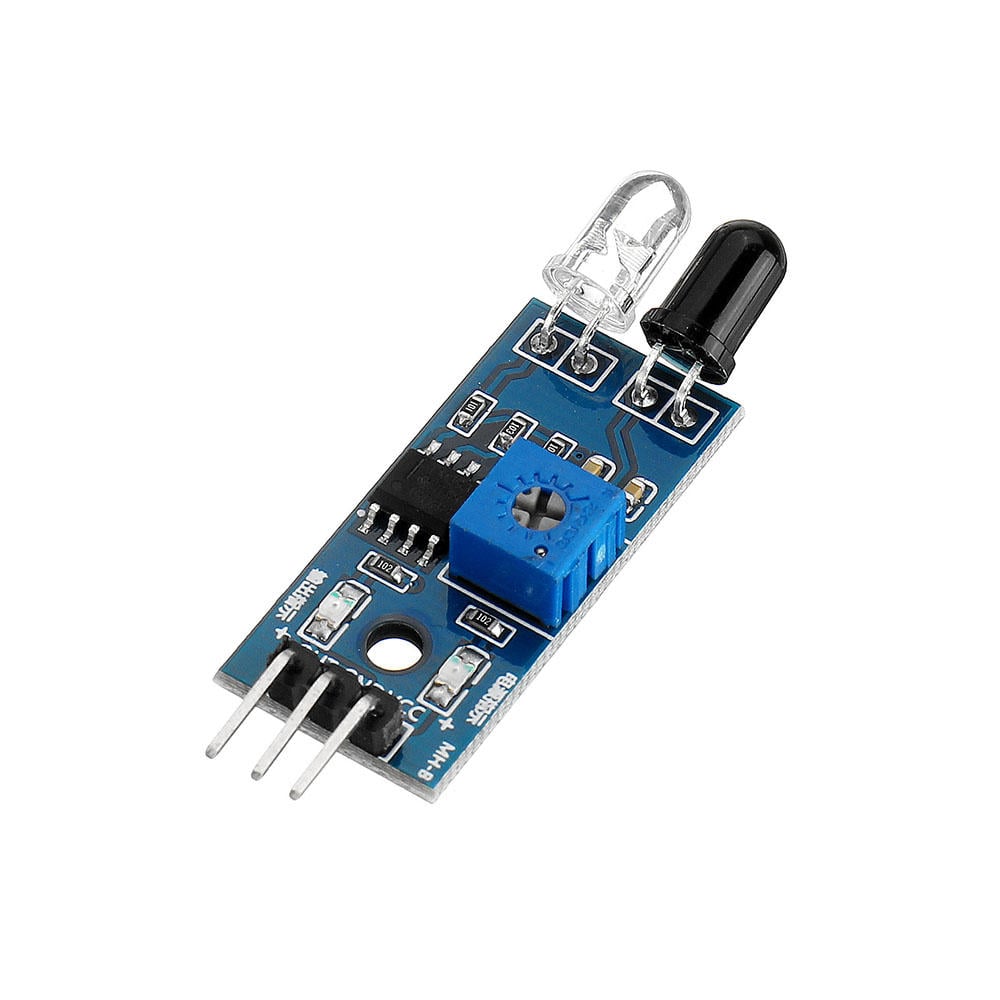


FIG1: IR SENSOR

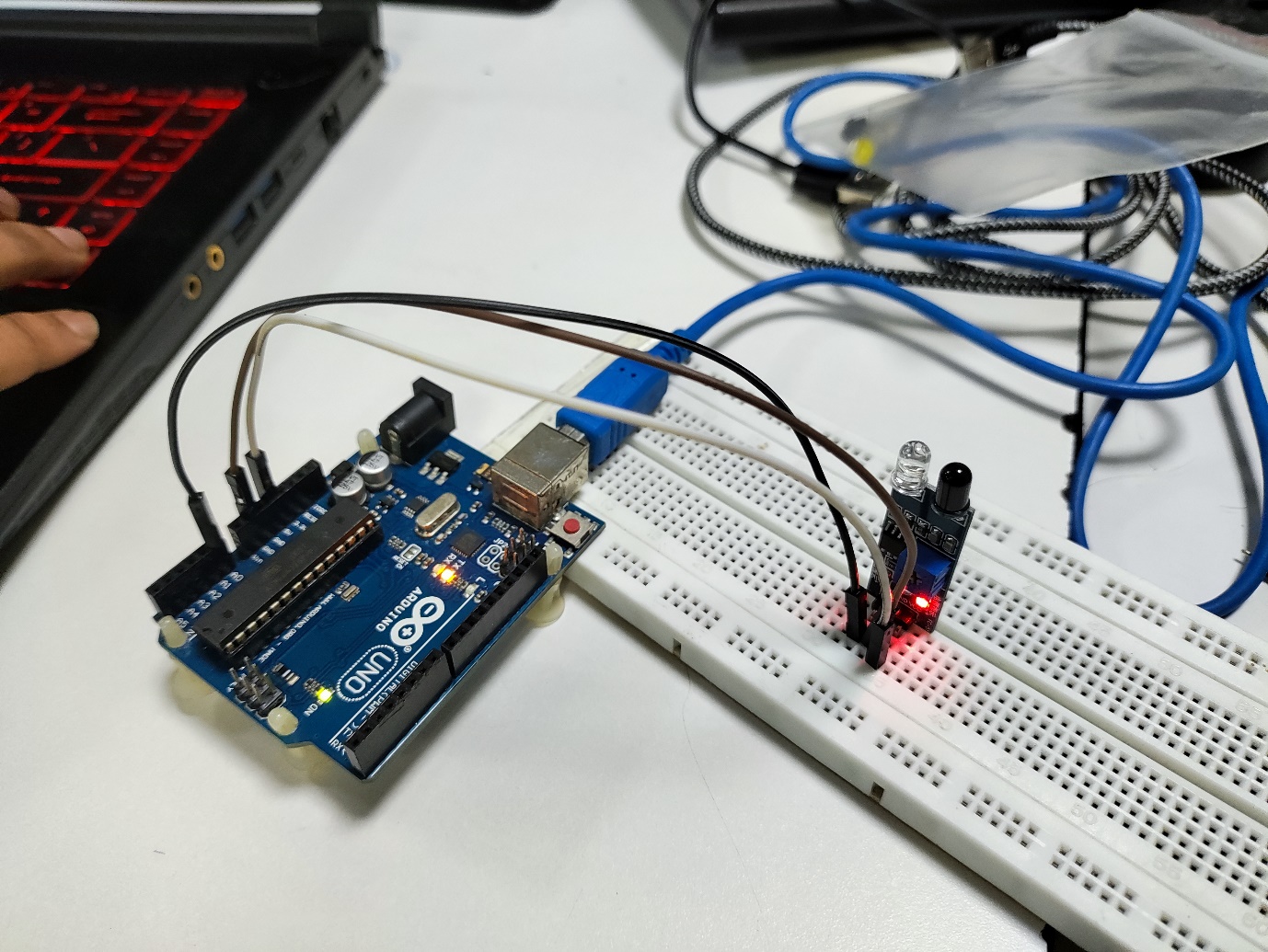


FIG 2:Connections

NOTE: if(statusSensor<100) this value of 100 must be changed according to the value of the serial monitor.

**BUZZER:**

1.To the above setup connect 8th pin to the positive terminal of the buzzer and negative terminal to ground.(the connections are given in such a way that we hold the wires together)

2.According to the flow the sound is played. In the below code, when executed we can hear (Sa Ri Ga Ma Pa Da Ni Sa)

NOTE: When the sensor detects motion just once sound is heard for one complete cycle.

**DC Motor Inerfacing**

1.Positive terminal of battery is connected is connected to positive terminal of motor driver and negative terminal of battery is connected to negative terminal of motor driver.

2.The DC motor can be connected at any terminal.

3.VCC is connected to vcc of motor driver.

4.ground of Arduino board is connected to Arduino ground of motor driver.

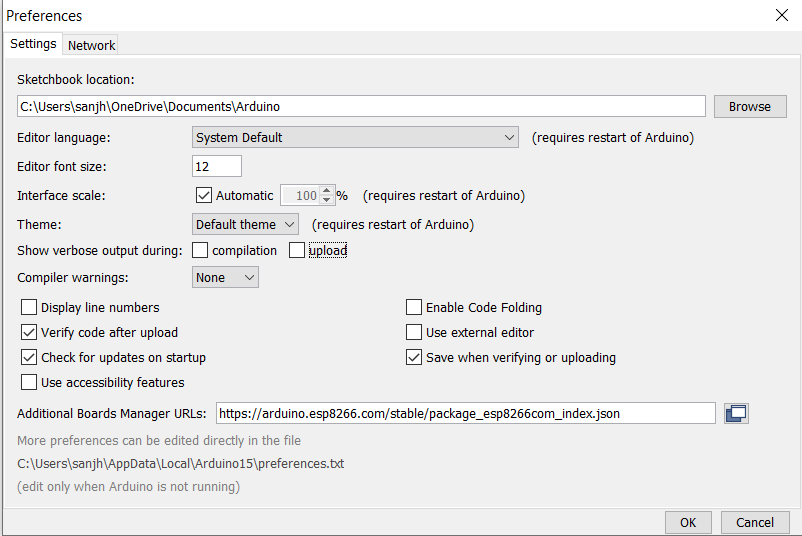
5.The the two inputs are connected to the respective pins of the side where you connected the motor.

**NODE MCU:**

1.Open Arduino IDE.

2.Open File->preferences->Additional Boards Manager->copy the URL

3.URL: https://arduino.esp8266.com/stable/package\_esp8266com\_index.json

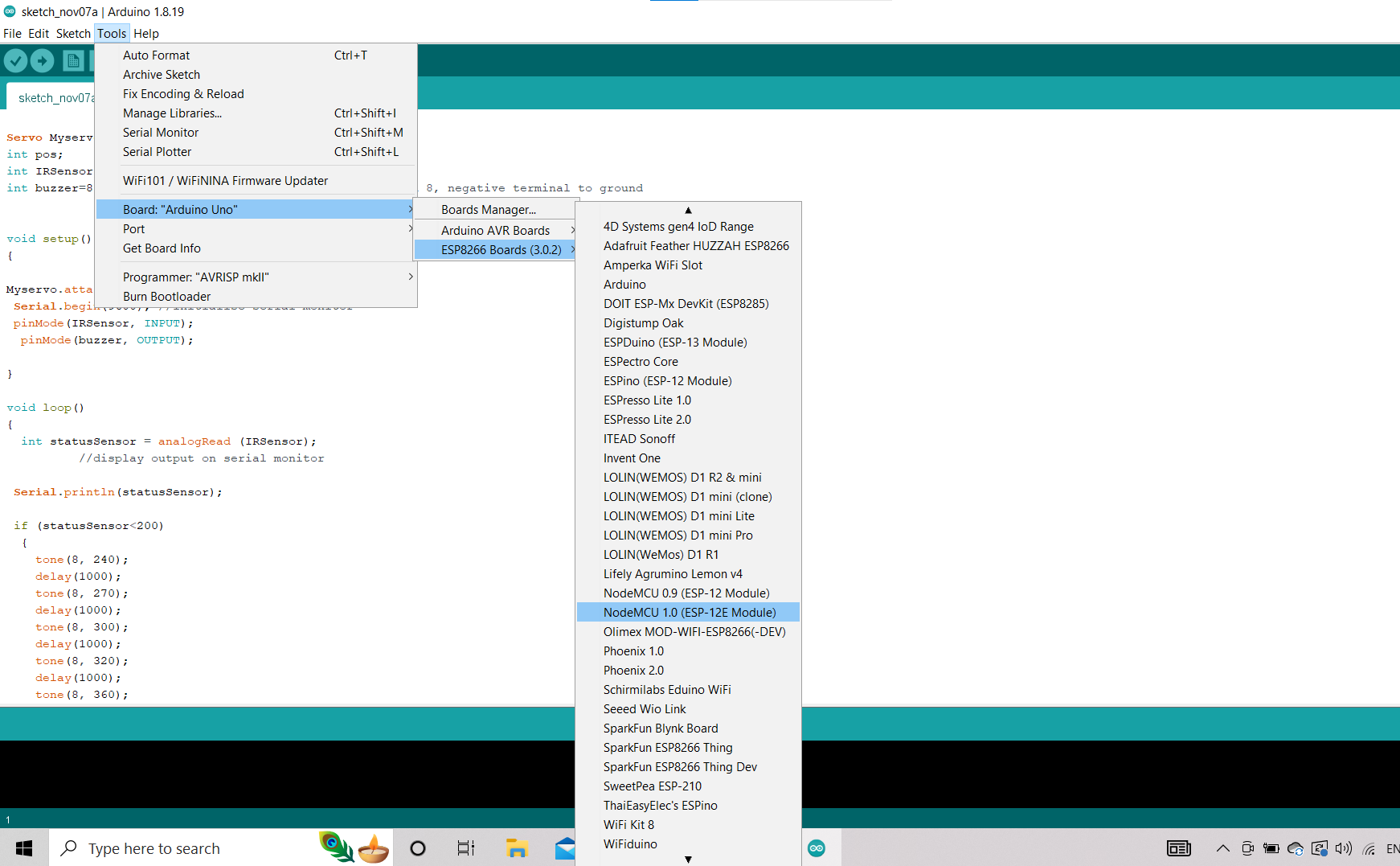


4.Enter ok.

5.Tools->board->boards manager->wait for proper instalment->Boards manager->in the search bar search for esp8266->install the package by ESP8266 community version 3.0.2.

6. Connect the NodeMCU to system.

7.Then go to tools->port->the port to which u have connected the node mcu.

8. From the above select NodeMCU(ESP-12E Module)