NAME – Satyajit Panda (IoT summer internship)

DATE - 27/06/2022 (MON)

Arduino tutorial reference

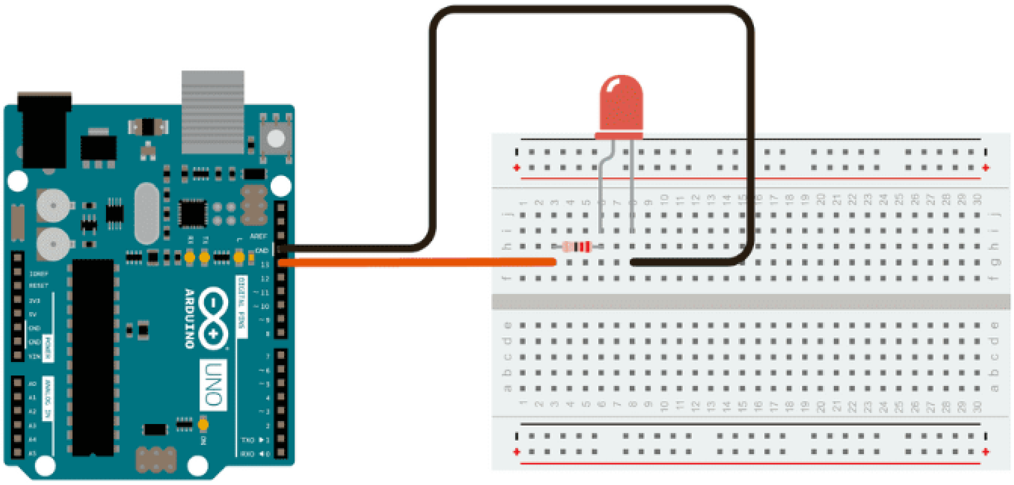
1. Arduino basics
   Programmed to blink the port 13 LED

   Program

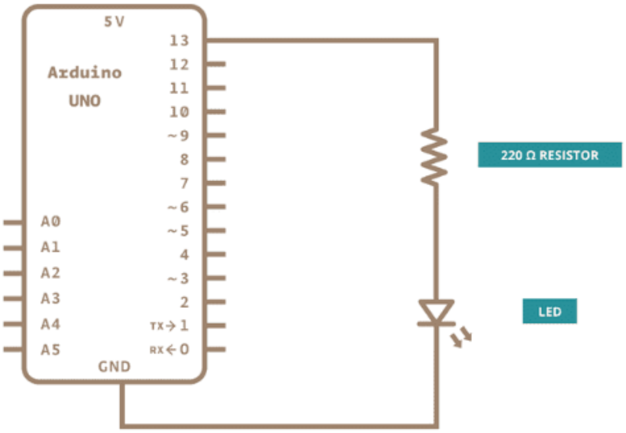
```
void setup() {
 // initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

To control an external led along with the builtin LED

Circuit diagram

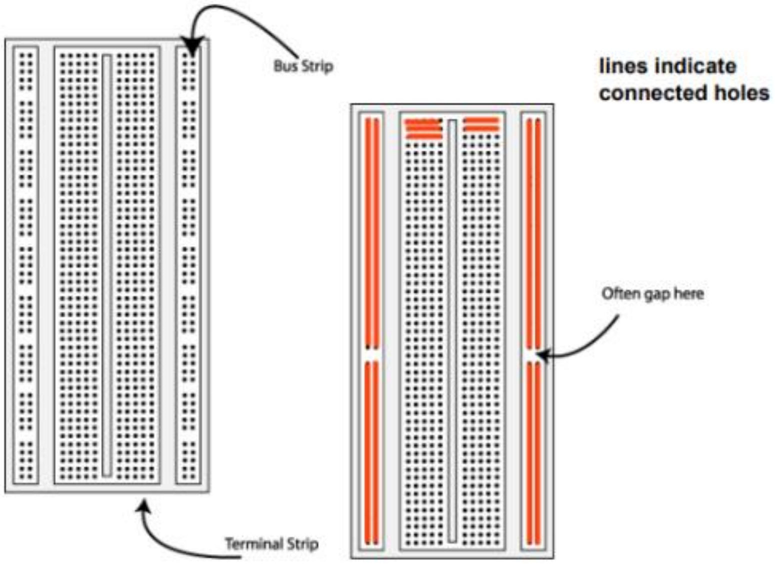


Schematic



2. Bread board connections

Underlying connections of a breadboard

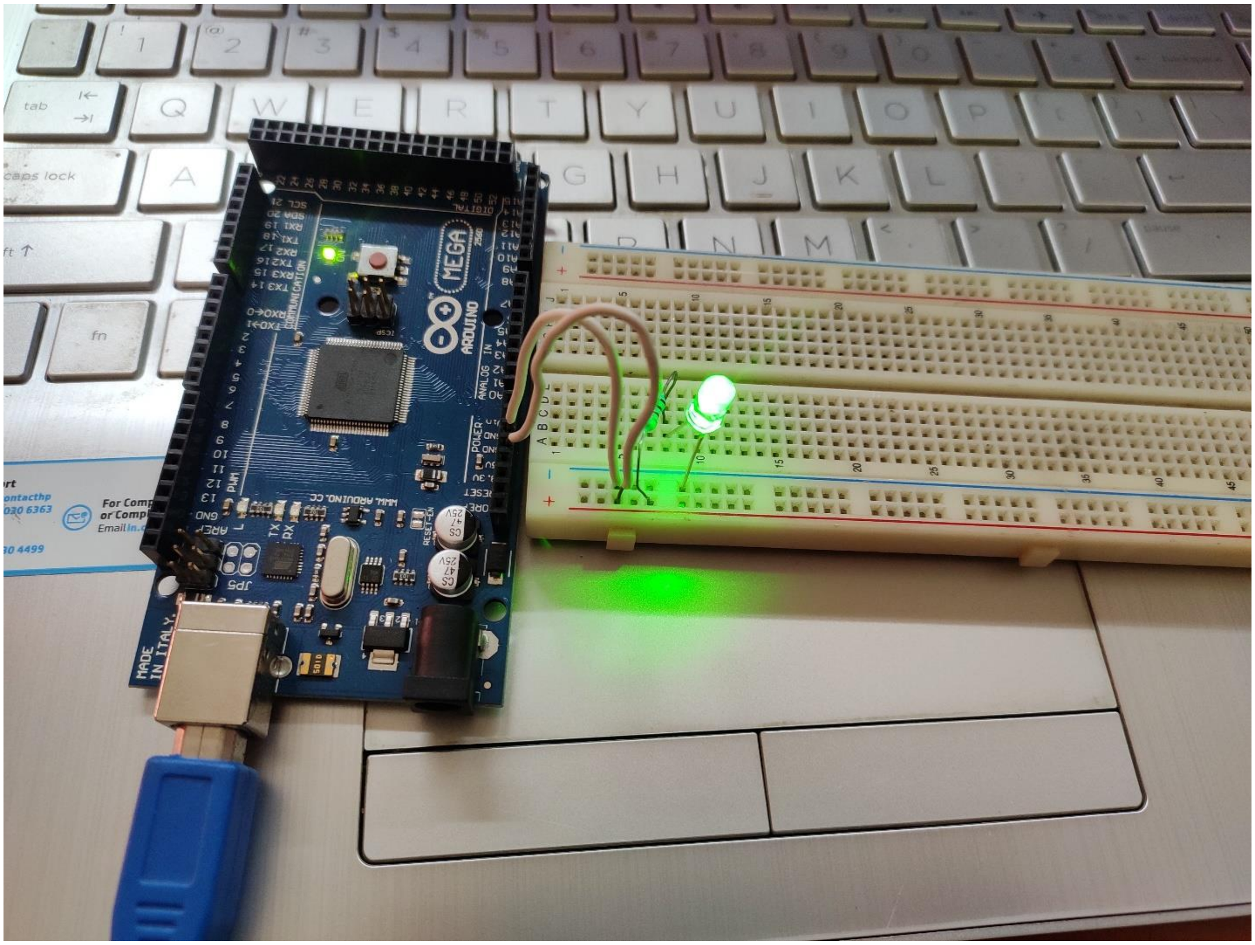


Breadboard Strips

The positive and negative labeled rail like pin holes are called power rails. Generally they are used for external power supplies, where the column like holes are used for electrical components connections.
The positive and negative signs are only for reference and ease of the user. We can use them as per our convenience.

reference

Programmed to connect an LED to the breadboard.



This is the normal analog connection of an LED with the Arduino board

We also can connect it with the digital 13 pin and digital ground to synchronize the external LED with the inbuilt LED (13).

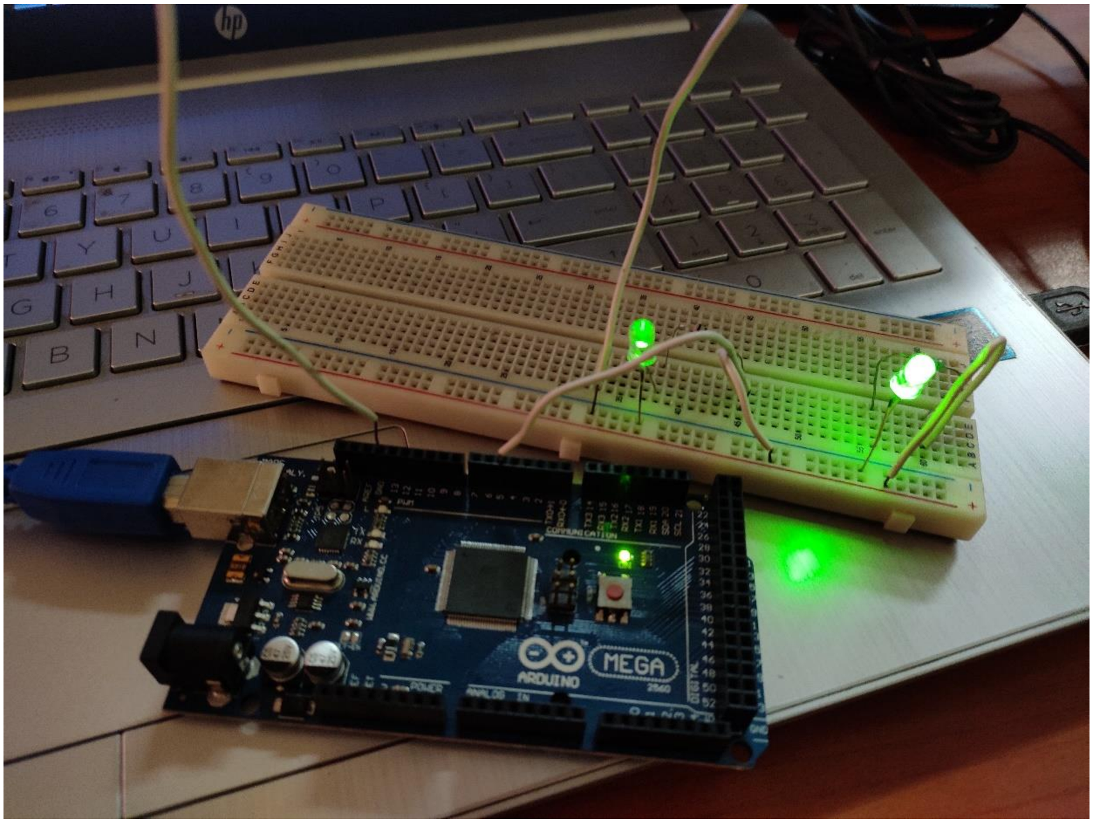
DATE – 28/06/2022 (TUE)

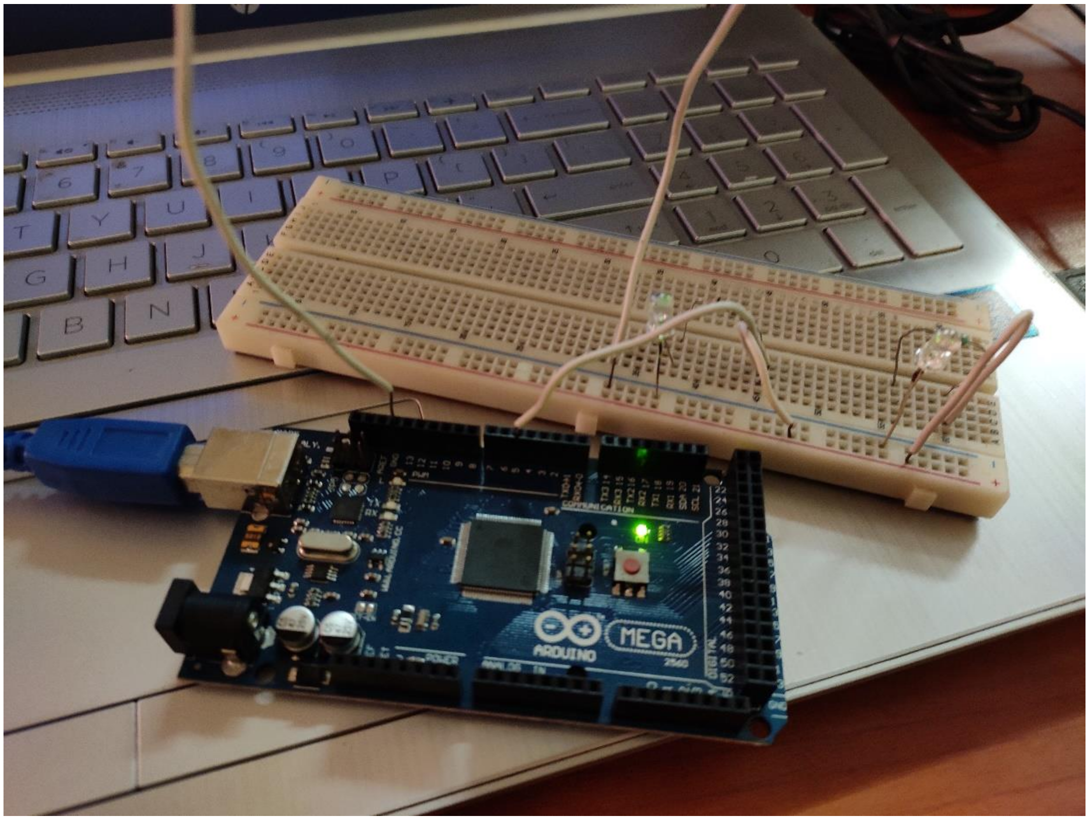
1. Parallel connection on LEDS on bread board and blinking them

Program

```
void setup() {
  // put your setup code here, to run once:
//  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(5,HIGH);
  delay(1000);
  digitalWrite(5,LOW);
  delay(1000);
}
```

Both the LEDs are connected to pin 5, and we use the pin to control at what time interval it should set it high and low.

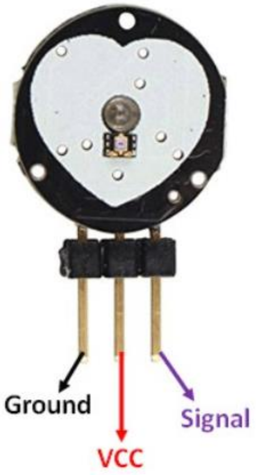

ARDUINO
MEGA
2560


ARDUINO
MEGA
2560

2. Checking BPM using a heart rate pulse sensor

About pulse sensor

The pulse sensor has 3 pins VCC, GND and analog pin.



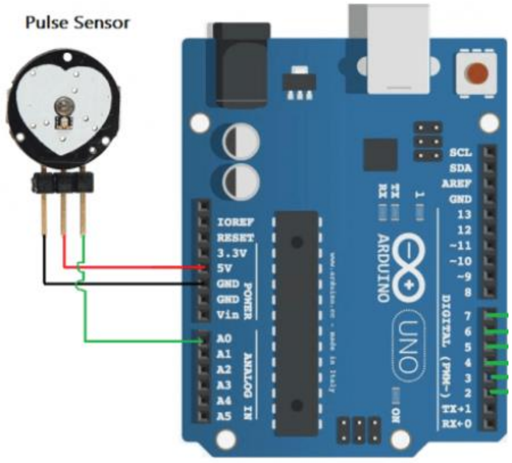
To use it we just have to plug it with the Arduino and place the sensor on our fingertip. It provides live heart rate to use in our projects.

There is a heart logo on the front side of the sensor. On this we have to put our fingers on. The small round hole inside the heart logo, where a flash green led shines is a small ambient light photo sensor.

The heart rate sensor is connected with the Arduino with its power input (red wire) to 3.3/5v supply, blue wire to GND and the ac signal output (purple wire) to A0 AC input of Arduino.

reference

Circuit diagram for the electrical connection



As we are using the serial monitor and serial plotter to visualize the output, we don’t need any external device for this purpose.

Working principle of the sensor

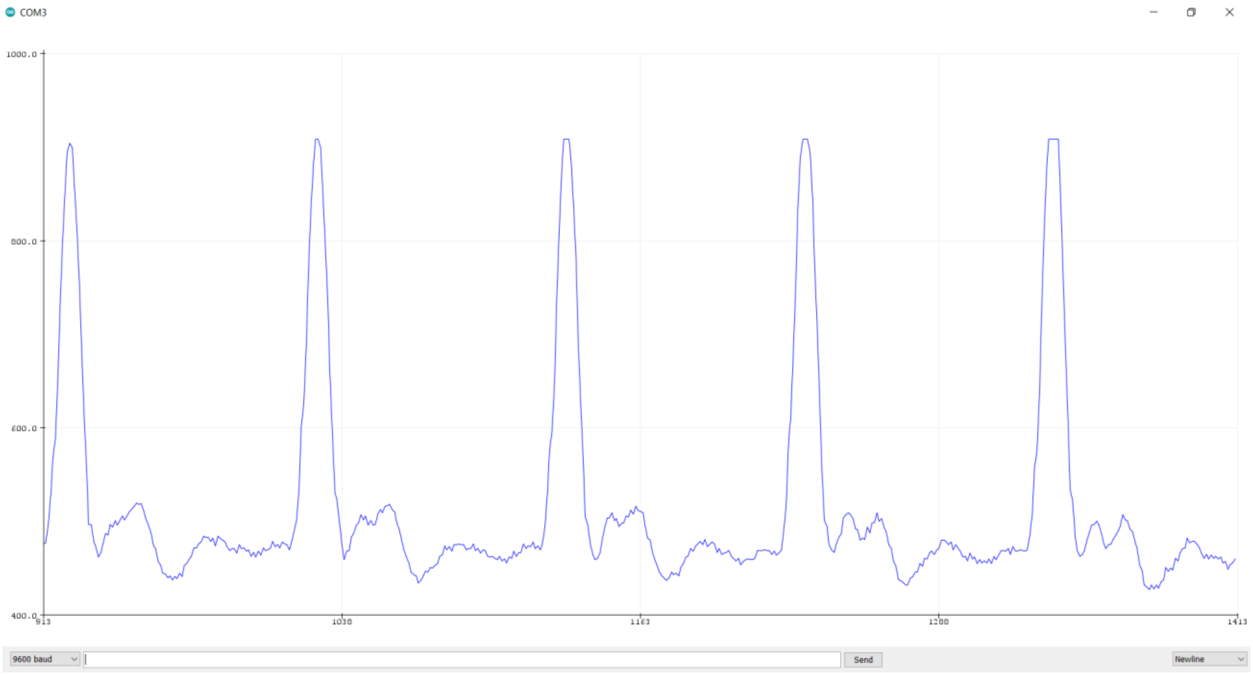
When a heartbeat occurs, blood is pumped through the human body and gets squeezed into the capillary tissues. Consequently, the volume of these capillary tissues increases. But in between the two consecutive heartbeats, this volume inside capillary tissues decreases. This change in volume between the heartbeats affects the amount of light that will transmit through these tissues. This can be measured with the help of a microcontroller.

The pulse sensor module has a light that helps in measuring the pulse rate. When we place the finger on the pulse sensor, the light reflected will change based on the volume of blood inside the capillary blood vessels. This variation in light transmission and reflection can be obtained as a pulse from the output of the pulse sensor. This pulse can be then conditioned to measure heartbeat and then programmed accordingly to read as heartbeat count using Arduino.

Program for observing the heart rate in serial plotter

```
//  Variables
int PulseSensorPurplePin = 0;        // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0
int LED13 = 13;   //  The on-board Arduion LED


int Signal;                // holds the incoming raw data. Signal value can range from 0-1024
int Threshold = 550;            // Determine which Signal to "count as a beat", and which to ingore.


// The SetUp Function:
void setup() {
  pinMode(LED13,OUTPUT);         // pin that will blink to your heartbeat!
   Serial.begin(9600);         // Set's up Serial Communication at certain speed.
}
// The Main Loop Function
void loop() {
  Signal = analogRead(PulseSensorPurplePin);  // Read the PulseSensor's value.
                                              // Assign this value to the "Signal" variable.


   Serial.println(Signal);                    // Send the Signal value to Serial Plotter.
   if(Signal > Threshold){                          // If the signal is above "550", then "turn-on" Arduino's on-Board
LED.
     digitalWrite(LED13,HIGH);
   } else {
     digitalWrite(LED13,LOW);                //  Else, the sigal must be below "550", so "turn-off" this LED.
   }
delay(10);
}
```

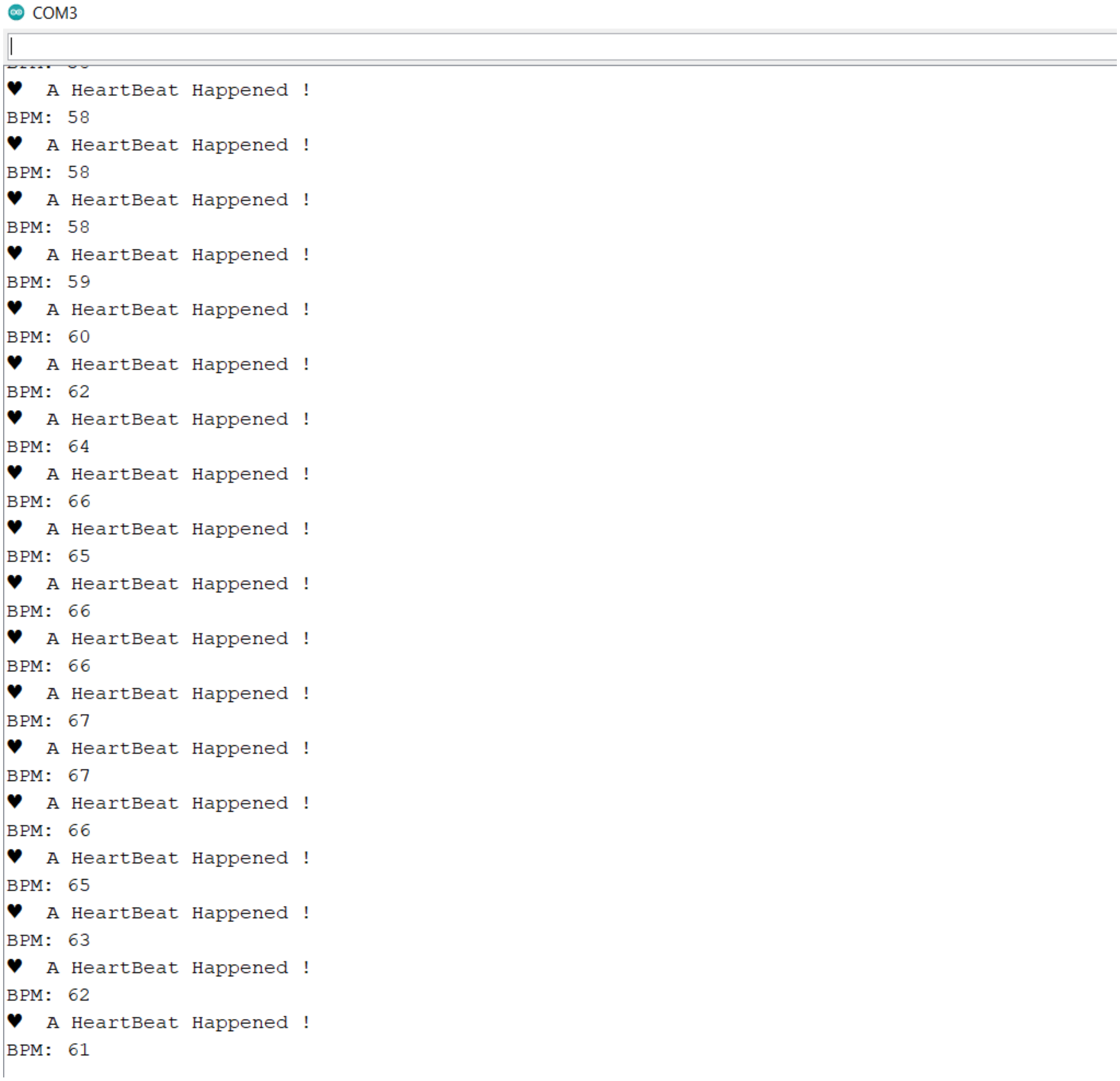
Serial Plotter output



Program to display the BPM in the serial monitor

```
}


#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.



//  Variables

const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0

const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.

int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.

                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default
setting.

                               // Otherwise leave the default "550" value.

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called
"pulseSensor"



void setup() {

  Serial.begin(9600);          // For Serial Monitor



  // Configure the PulseSensor object, by assigning our variables to it.

  pulseSensor.analogInput(PulseWire);

  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.

  pulseSensor.setThreshold(Threshold);



  // Double-check the "pulseSensor" object was created and "began" seeing a signal.

   if (pulseSensor.begin()) {

    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or
on Arduino reset.

  }

}
```

Serial Monitor output



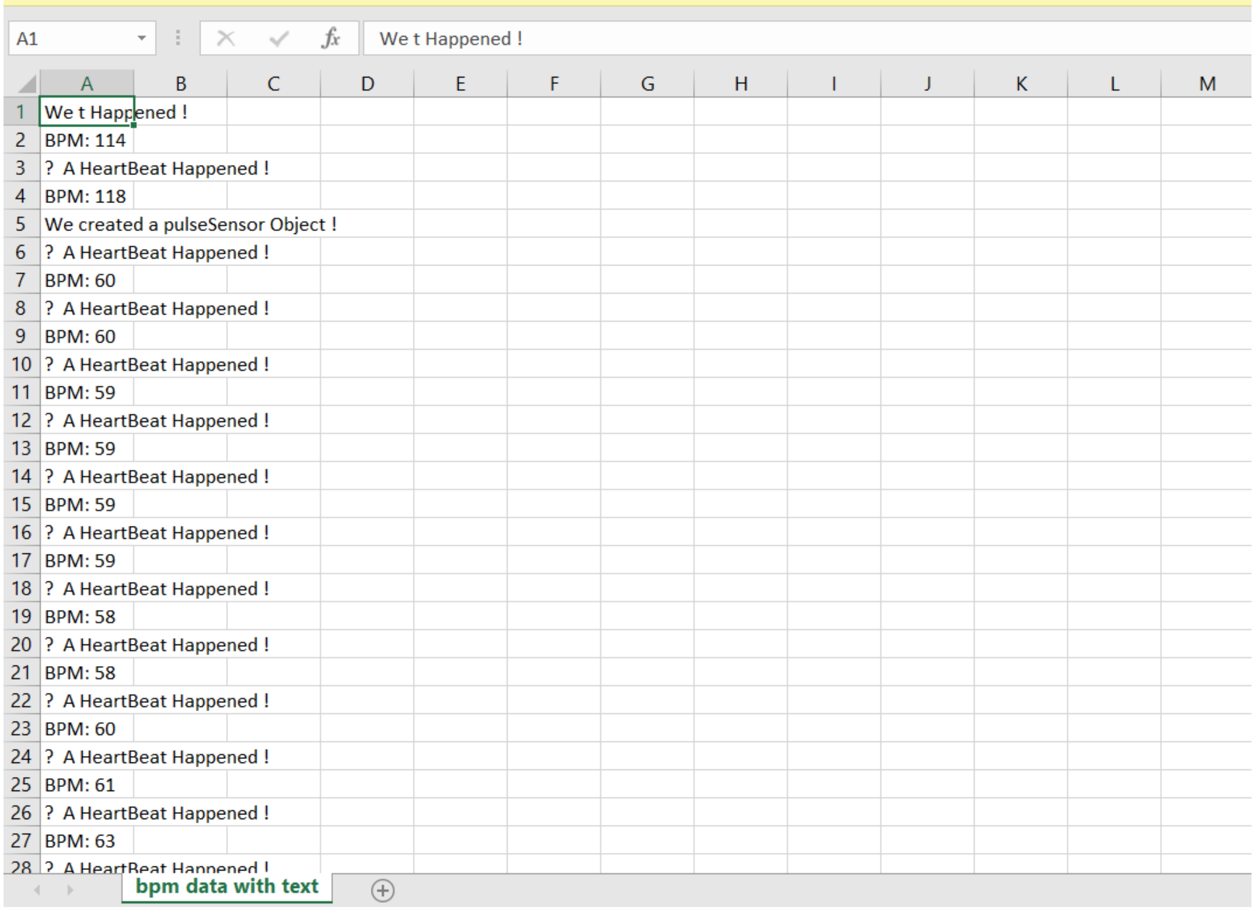
```
♥  A HeartBeat Happened !
BPM: 58
♥  A HeartBeat Happened !
BPM: 58
♥  A HeartBeat Happened !
BPM: 58
♥  A HeartBeat Happened !
BPM: 59
♥  A HeartBeat Happened !
BPM: 60
♥  A HeartBeat Happened !
BPM: 62
♥  A HeartBeat Happened !
BPM: 64
♥  A HeartBeat Happened !
BPM: 66
♥  A HeartBeat Happened !
BPM: 65
♥  A HeartBeat Happened !
BPM: 66
♥  A HeartBeat Happened !
BPM: 66
♥  A HeartBeat Happened !
BPM: 67
♥  A HeartBeat Happened !
BPM: 67
♥  A HeartBeat Happened !
BPM: 66
♥  A HeartBeat Happened !
BPM: 65
♥  A HeartBeat Happened !
BPM: 63
♥  A HeartBeat Happened !
BPM: 62
♥  A HeartBeat Happened !
BPM: 61
```

DATE – 29/06/2022 (WED)

1. Converting the serial monitor data into a csv file and storing it

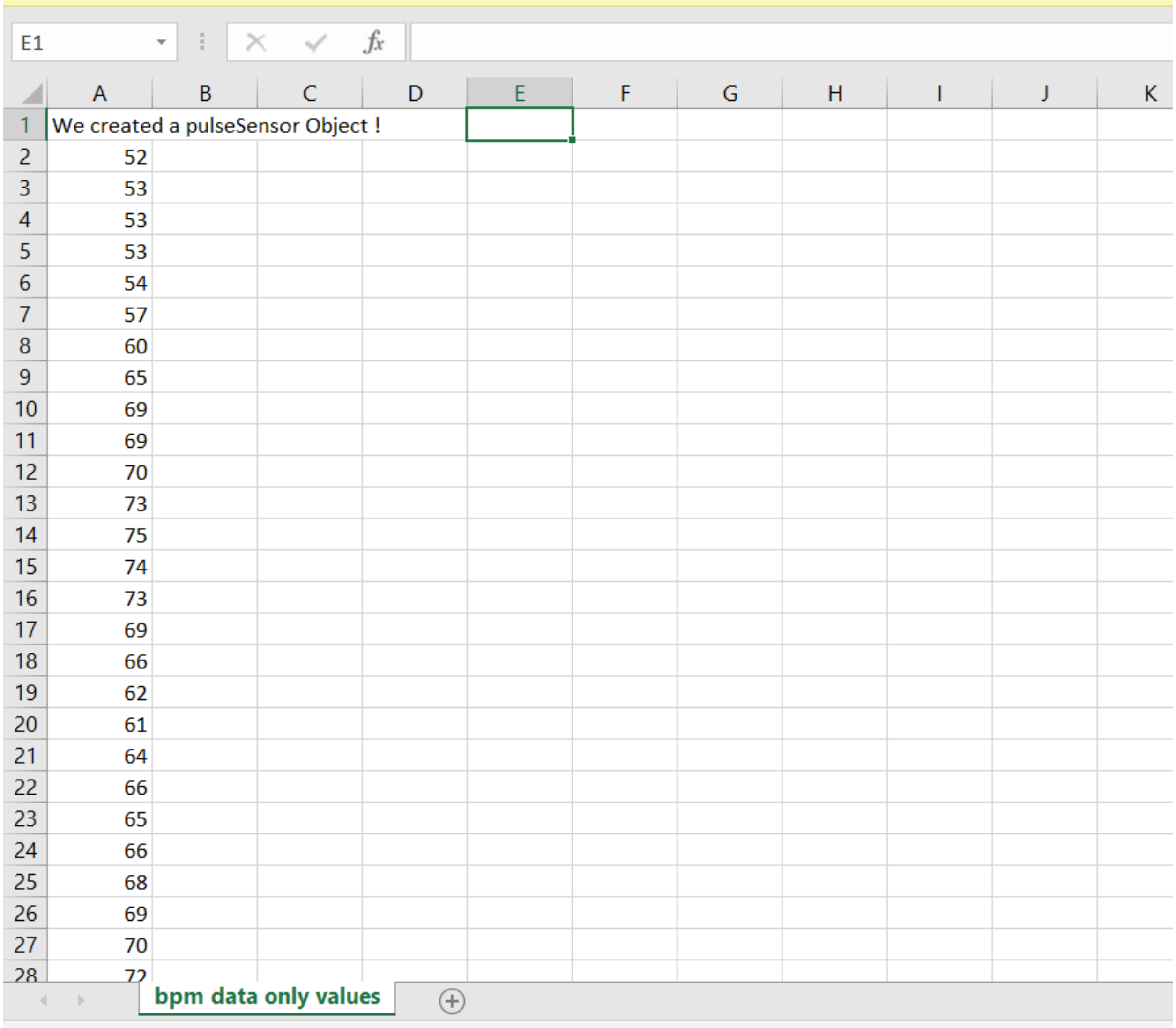
   Coolterm software is used to export the serial monitor data and to store it in a txt file, then we convert this txt into a csv(comma separated values) file.

   [referenece]()



We t Happened !

	A
1	We t Happened !
2	BPM: 114
3	? A HeartBeat Happened !
4	BPM: 118
5	We created a pulseSensor Object !
6	? A HeartBeat Happened !
7	BPM: 60
8	? A HeartBeat Happened !
9	BPM: 60
10	? A HeartBeat Happened !
11	BPM: 59
12	? A HeartBeat Happened !
13	BPM: 59
14	? A HeartBeat Happened !
15	BPM: 59
16	? A HeartBeat Happened !
17	BPM: 59
18	? A HeartBeat Happened !
19	BPM: 58
20	? A HeartBeat Happened !
21	BPM: 58
22	? A HeartBeat Happened !
23	BPM: 60
24	? A HeartBeat Happened !
25	BPM: 61
26	? A HeartBeat Happened !
27	BPM: 63
28	? A HeartBeat Happened !



	A	B	C	D	E	F	G	H	I	J	K
1	We created a pulseSensor Object !										
2	52										
3	53										
4	53										
5	53										
6	54										
7	57										
8	60										
9	65										
10	69										
11	69										
12	70										
13	73										
14	75										
15	74										
16	73										
17	69										
18	66										
19	62										
20	61										
21	64										
22	66										
23	65										
24	66										
25	68										
26	69										
27	70										
28	72										

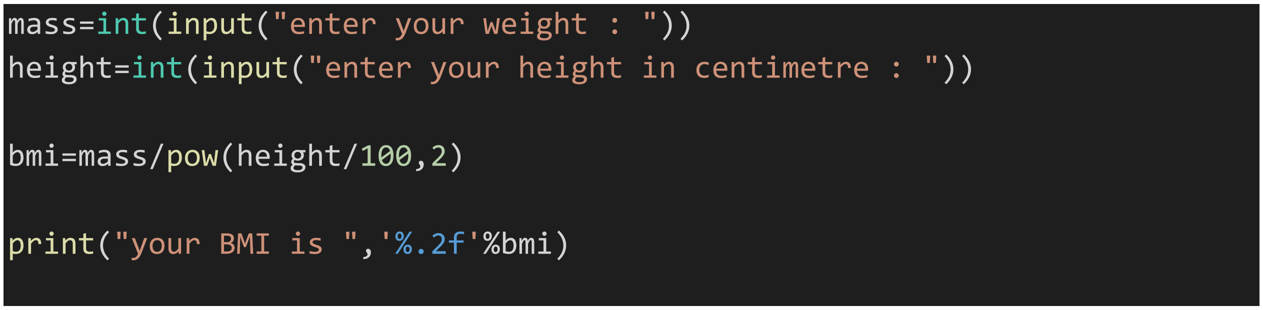
2. Read about basics of python

reference

DATE – 30/06/2022 (THU)

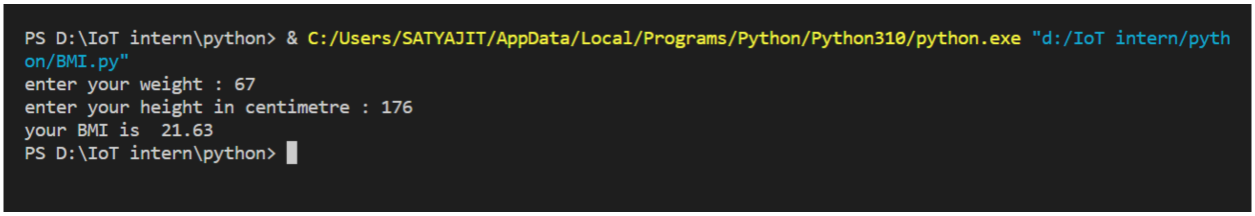
Basic python programs done.

1. Program to calculate BMI



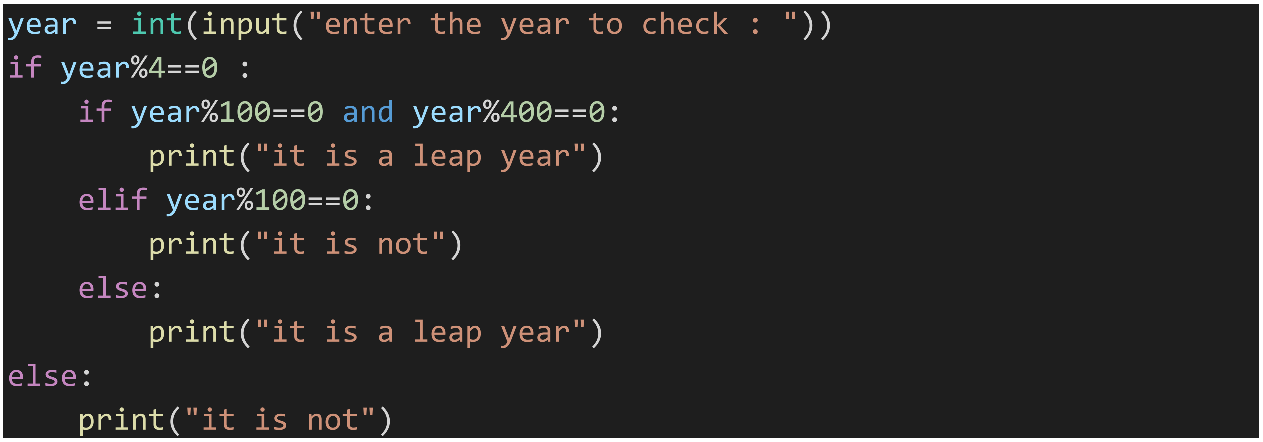
```
mass=int(input("enter your weight : "))
height=int(input("enter your height in centimetre : "))

bmi=mass/pow(height/100,2)

print("your BMI is ",'%.2f'%bmi)
```

<u>output</u>



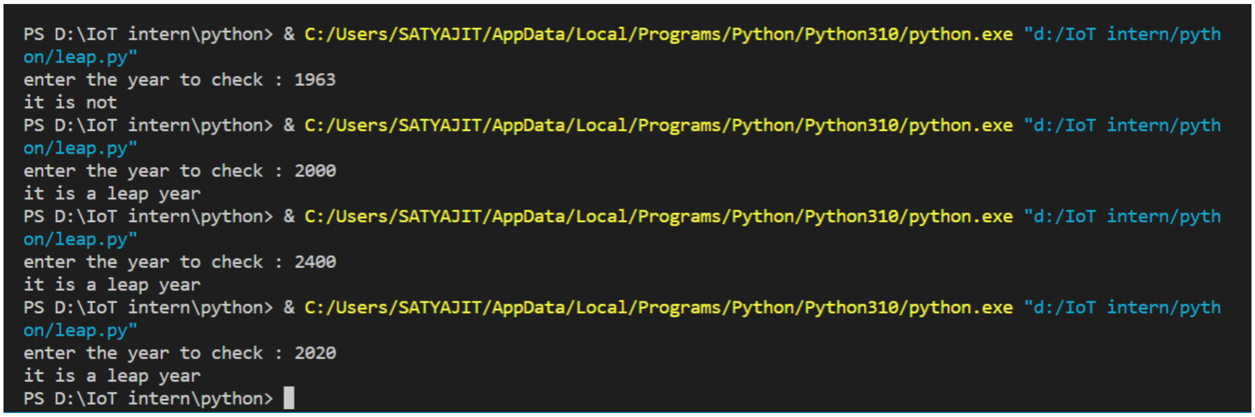
```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/BMI.py"
enter your weight : 67
enter your height in centimetre : 176
your BMI is  21.63
PS D:\IoT intern\python> 
```

2. Program to know a year is leap year or not



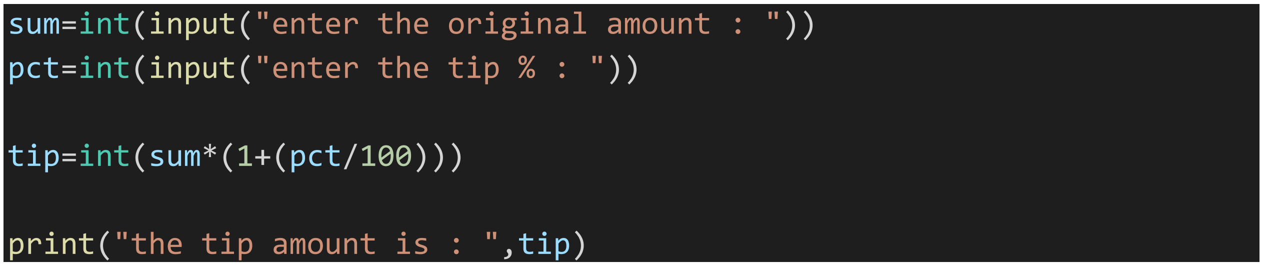
```
year = int(input("enter the year to check : "))
if year%4==0 :
    if year%100==0 and year%400==0:
        print("it is a leap year")
    elif year%100==0:
        print("it is not")
    else:
        print("it is a leap year")
else:
    print("it is not")
```

output



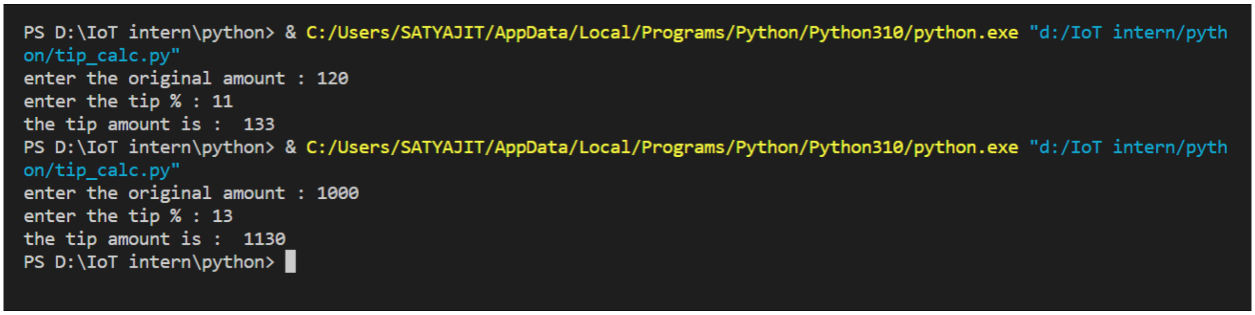
```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/leap.py"
enter the year to check : 1963
it is not
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/leap.py"
enter the year to check : 2000
it is a leap year
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/leap.py"
enter the year to check : 2400
it is a leap year
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/leap.py"
enter the year to check : 2020
it is a leap year
PS D:\IoT intern\python> ▋
```

3. Program for tip caluculator



```
sum=int(input("enter the original amount : "))
pct=int(input("enter the tip % : "))

tip=int(sum*(1+(pct/100)))

print("the tip amount is : ",tip)
```

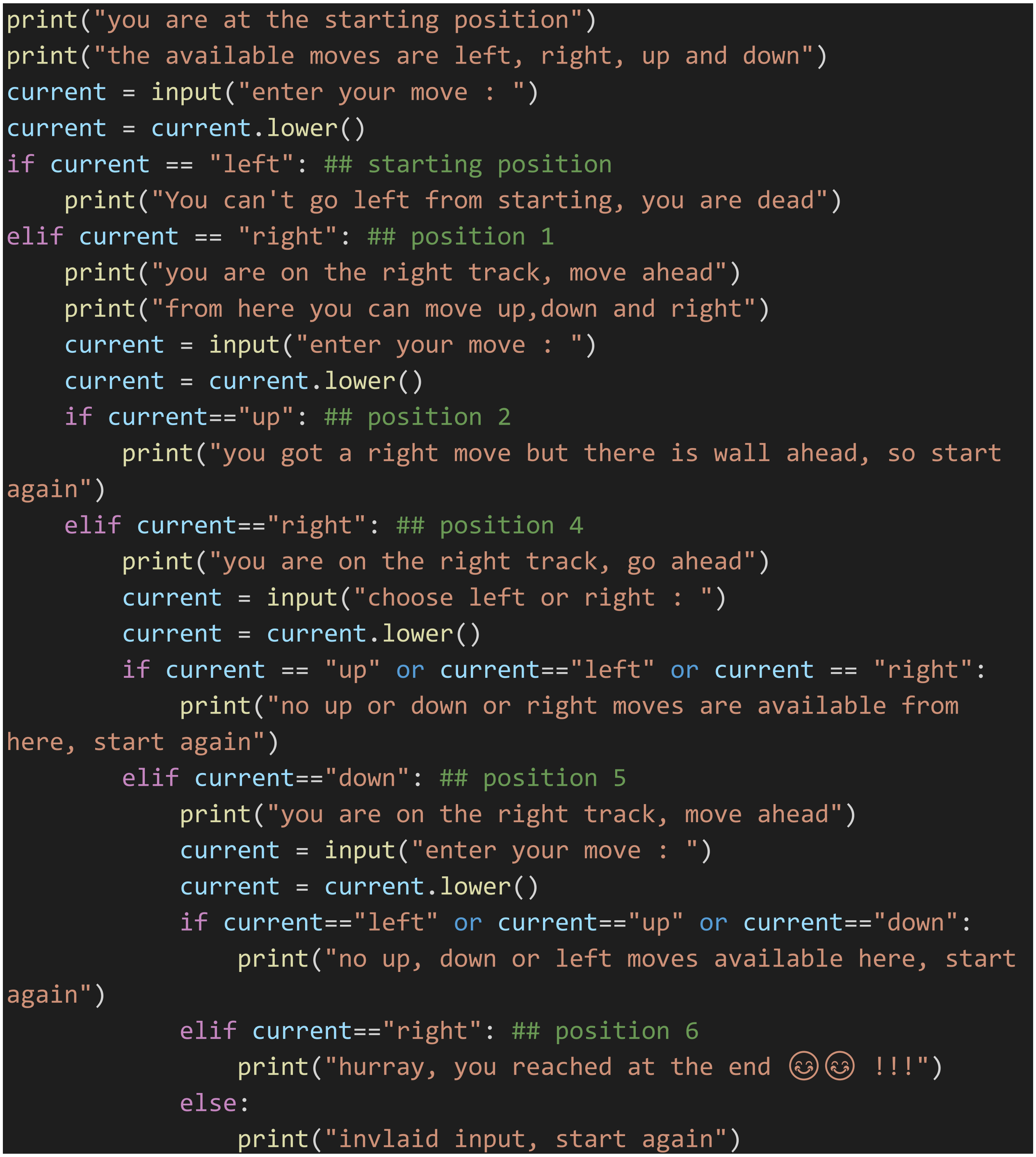
output



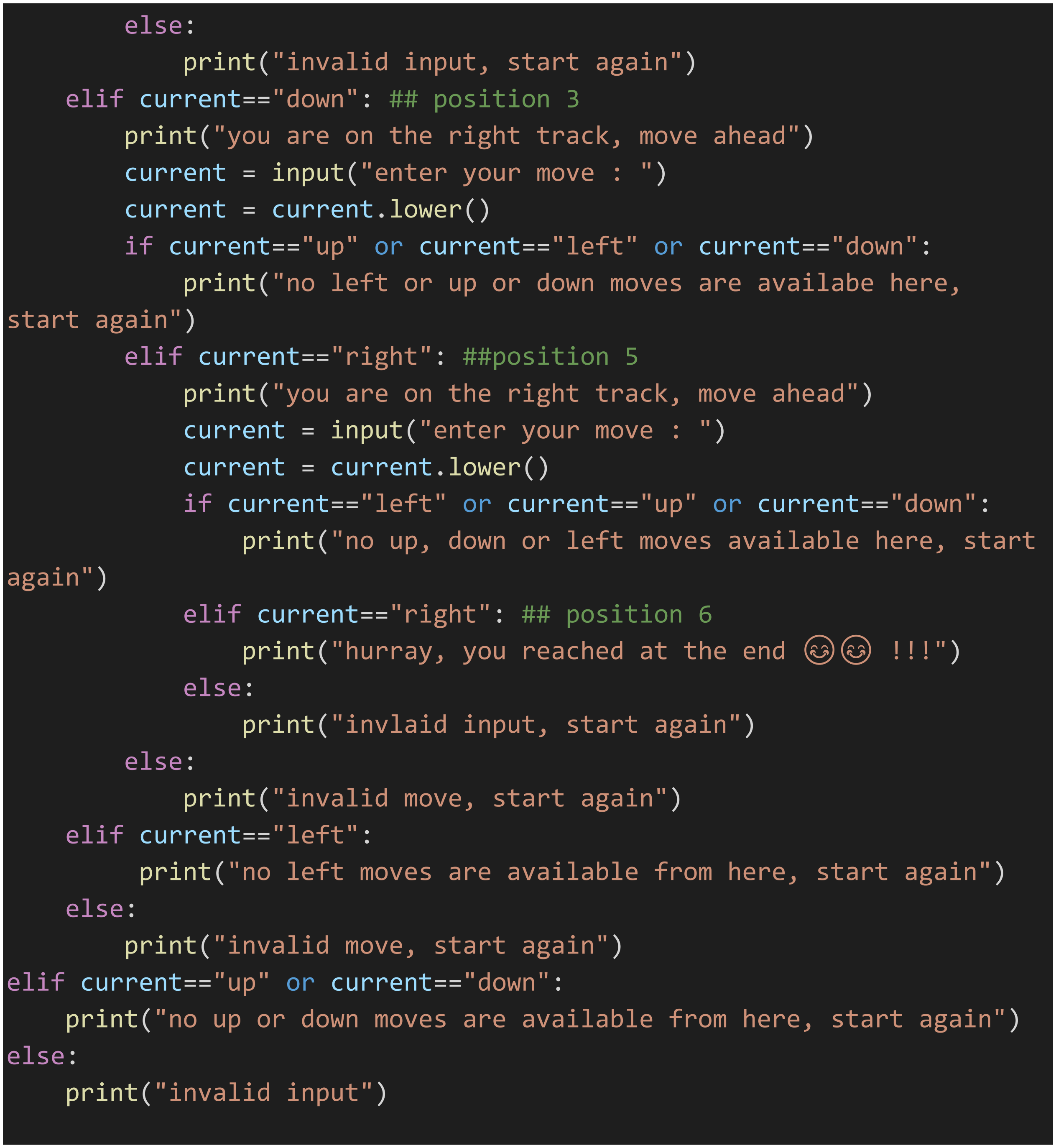
```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/tip_calc.py"
enter the original amount : 120
enter the tip % : 11
the tip amount is :  133
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/tip_calc.py"
enter the original amount : 1000
enter the tip % : 13
the tip amount is :  1130
PS D:\IoT intern\python> ▋
```

DATE – 04/07/2022 (MON)

4. The maze problem

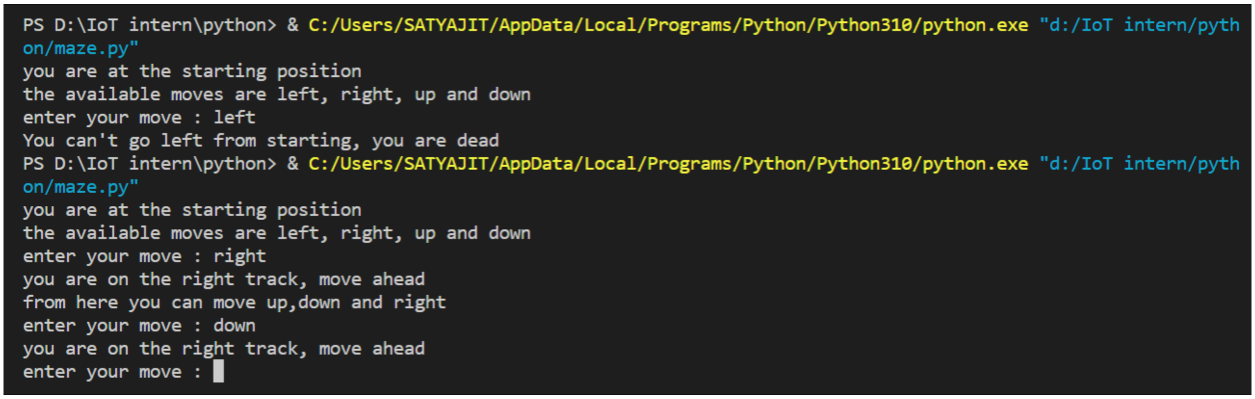


```
print("you are at the starting position")
print("the available moves are left, right, up and down")
current = input("enter your move : ")
current = current.lower()
if current == "left": ## starting position
    print("You can't go left from starting, you are dead")
elif current == "right": ## position 1
    print("you are on the right track, move ahead")
    print("from here you can move up,down and right")
    current = input("enter your move : ")
    current = current.lower()
    if current=="up": ## position 2
        print("you got a right move but there is wall ahead, so start
again")
    elif current=="right": ## position 4
        print("you are on the right track, go ahead")
        current = input("choose left or right : ")
        current = current.lower()
        if current == "up" or current=="left" or current == "right":
            print("no up or down or right moves are available from
here, start again")
        elif current=="down": ## position 5
            print("you are on the right track, move ahead")
            current = input("enter your move : ")
            current = current.lower()
            if current=="left" or current=="up" or current=="down":
                print("no up, down or left moves available here, start
again")
            elif current=="right": ## position 6
                print("hurray, you reached at the end 😊😊 !!!")
            else:
                print("invlaid input, start again")
```



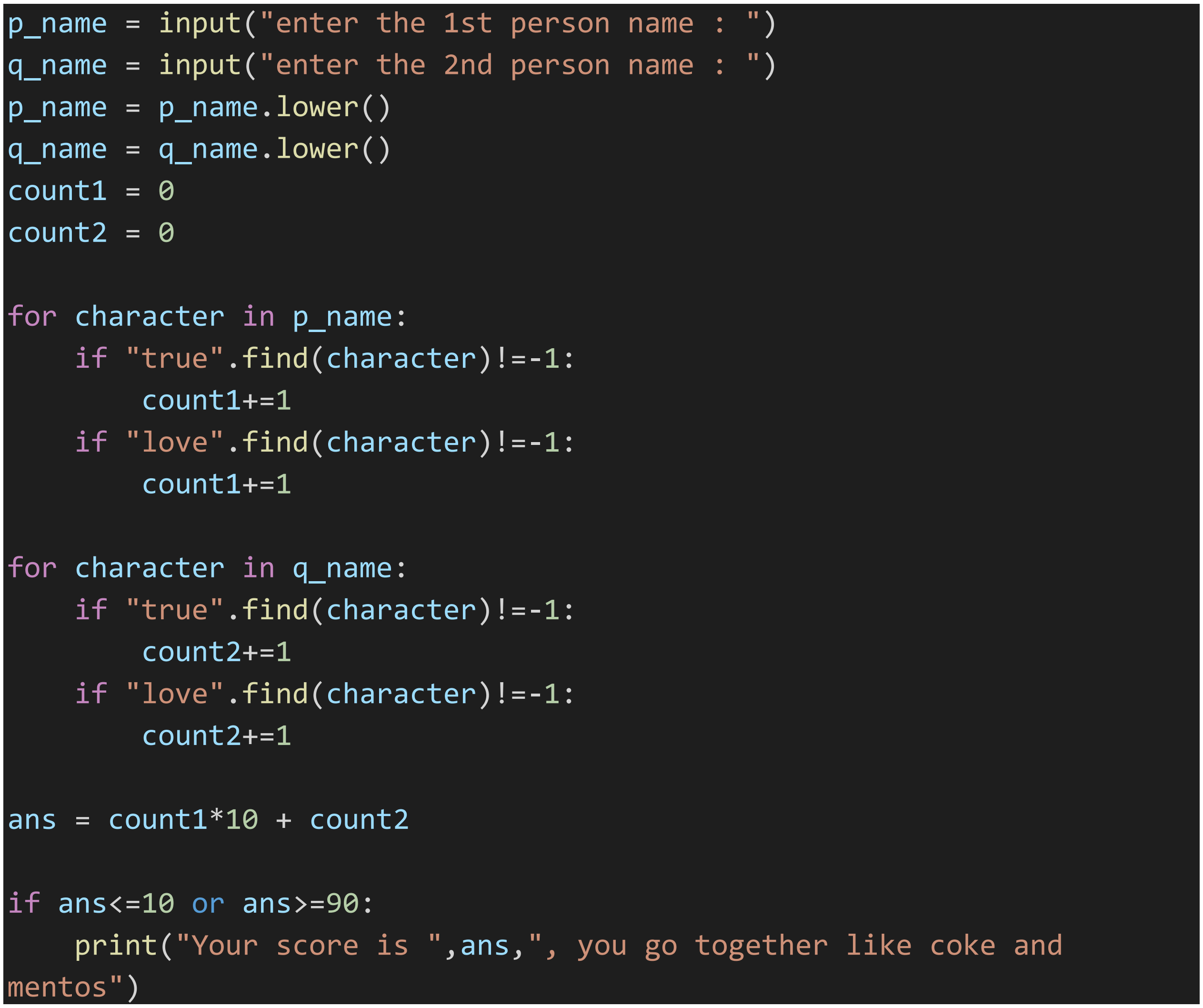
```
        else:
            print("invalid input, start again")
    elif current=="down": ## position 3
        print("you are on the right track, move ahead")
        current = input("enter your move : ")
        current = current.lower()
        if current=="up" or current=="left" or current=="down":
            print("no left or up or down moves are availabe here, 
start again")
        elif current=="right": ##position 5
            print("you are on the right track, move ahead")
            current = input("enter your move : ")
            current = current.lower()
            if current=="left" or current=="up" or current=="down":
                print("no up, down or left moves available here, start 
again")
            elif current=="right": ## position 6
                print("hurray, you reached at the end 😊😊 !!!")
            else:
                print("invlaid input, start again")
        else:
            print("invalid move, start again")
    elif current=="left":
         print("no left moves are available from here, start again")
    else:
        print("invalid move, start again")
elif current=="up" or current=="down":
    print("no up or down moves are available from here, start again")
else:
    print("invalid input")
```

output

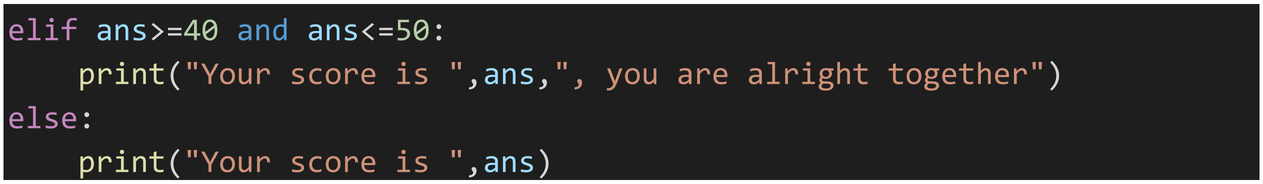


```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/maze.py"
you are at the starting position
the available moves are left, right, up and down
enter your move : left
You can't go left from starting, you are dead
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/maze.py"
you are at the starting position
the available moves are left, right, up and down
enter your move : right
you are on the right track, move ahead
from here you can move up,down and right
enter your move : down
you are on the right track, move ahead
enter your move :
```

5. True-love calculator program

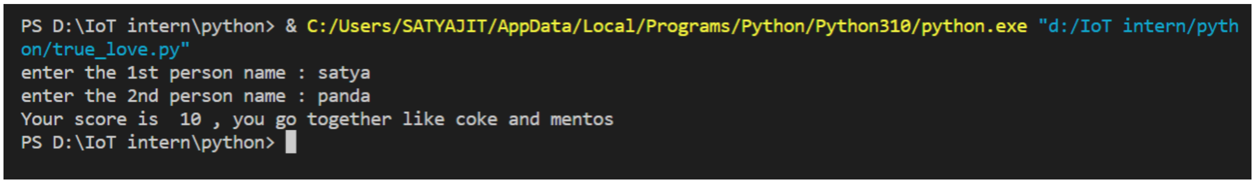


```
p_name = input("enter the 1st person name : ")
q_name = input("enter the 2nd person name : ")
p_name = p_name.lower()
q_name = q_name.lower()
count1 = 0
count2 = 0

for character in p_name:
    if "true".find(character)!=-1:
        count1+=1
    if "love".find(character)!=-1:
        count1+=1

for character in q_name:
    if "true".find(character)!=-1:
        count2+=1
    if "love".find(character)!=-1:
        count2+=1

ans = count1*10 + count2

if ans<=10 or ans>=90:
    print("Your score is ",ans,", you go together like coke and
mentos")
```



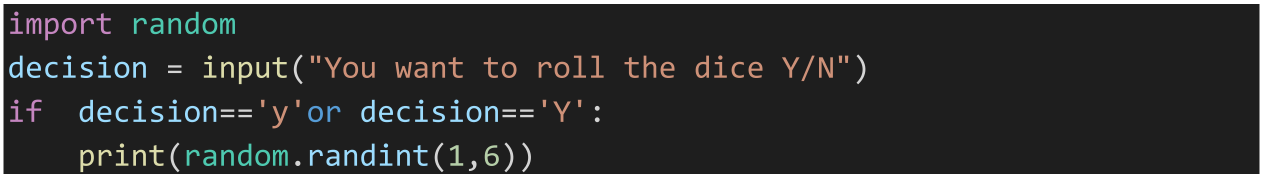
```
elif ans>=40 and ans<=50:
    print("Your score is ",ans,", you are alright together")
else:
    print("Your score is ",ans)
```

output



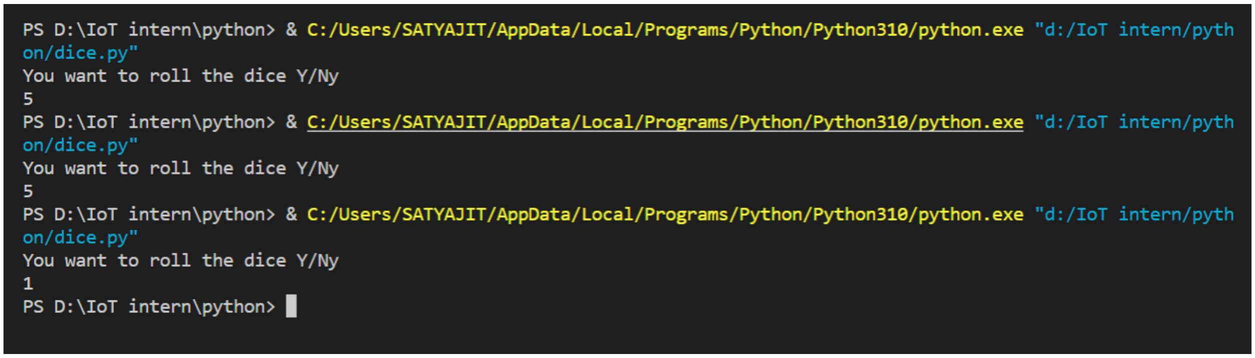
```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/true_love.py"
enter the 1st person name : satya
enter the 2nd person name : panda
Your score is  10 , you go together like coke and mentos
PS D:\IoT intern\python> █
```

6. Dice game problem



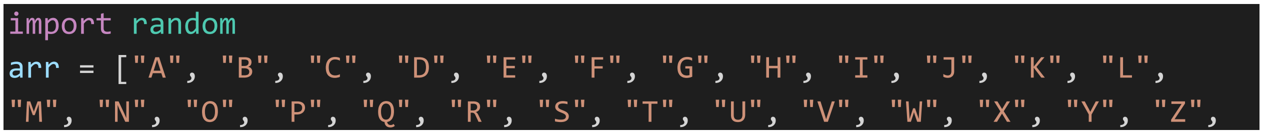
```
import random
decision = input("You want to roll the dice Y/N")
if  decision=='y'or decision=='Y':
    print(random.randint(1,6))
```

output

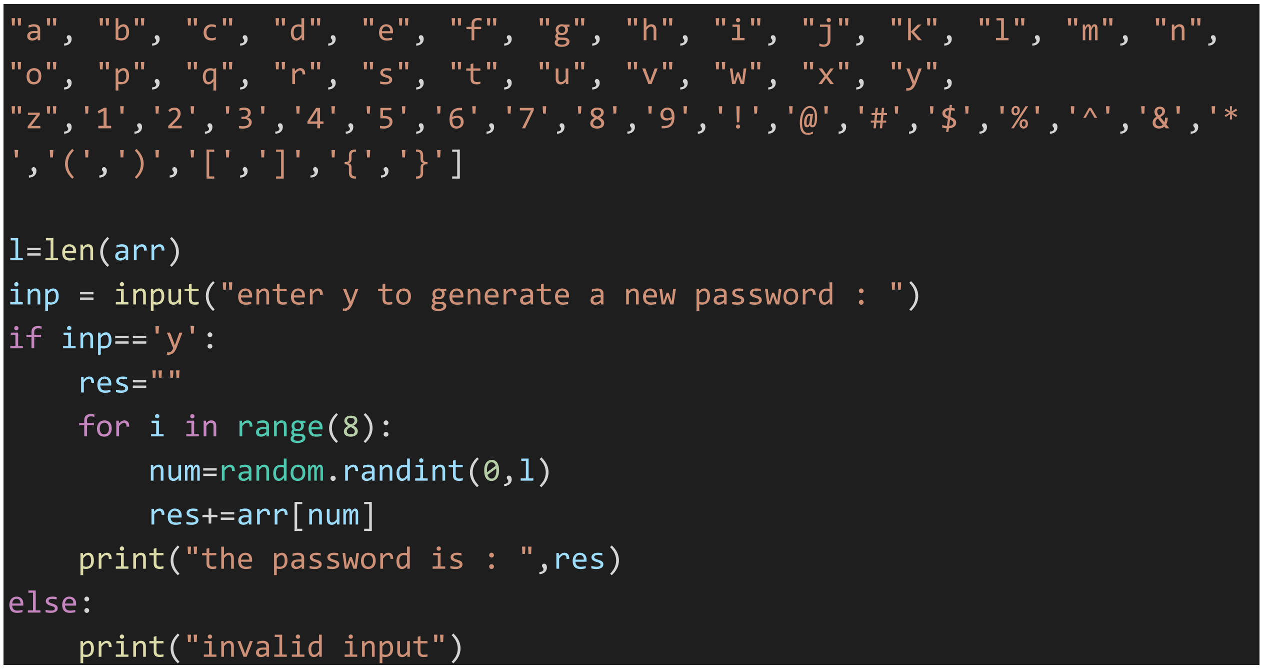


```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/dice.py"
You want to roll the dice Y/Ny
5
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/dice.py"
You want to roll the dice Y/Ny
5
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/dice.py"
You want to roll the dice Y/Ny
1
PS D:\IoT intern\python> █
```

7. Password generator problem

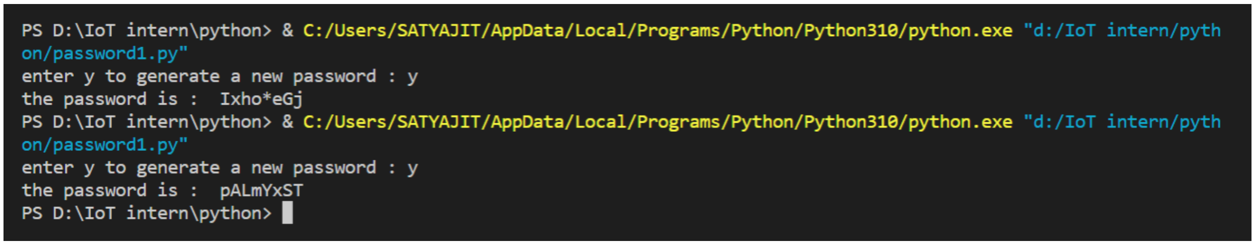


```
import random
arr = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L",
"M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z",
```



```
"a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n",
"o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z",'1','2','3','4','5','6','7','8','9','!','@','#','$','%','^','&','*
','(',')','[',']','{','}']

l=len(arr)
inp = input("enter y to generate a new password : ")
if inp=='y':
    res=""
    for i in range(8):
        num=random.randint(0,l)
        res+=arr[num]
    print("the password is : ",res)
else:
    print("invalid input")
```

output



```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/password1.py"
enter y to generate a new password : y
the password is :  Ixho*eGj
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/pyth
on/password1.py"
enter y to generate a new password : y
the password is :  pALmYxST
PS D:\IoT intern\python>
```

Connecting Arduino with things speak to view the sensor data online

DATE – 05/07/2022 (TUE)

Exploring how we can send Arduino sensor data online

DATE – 06/07/2022 (WED)

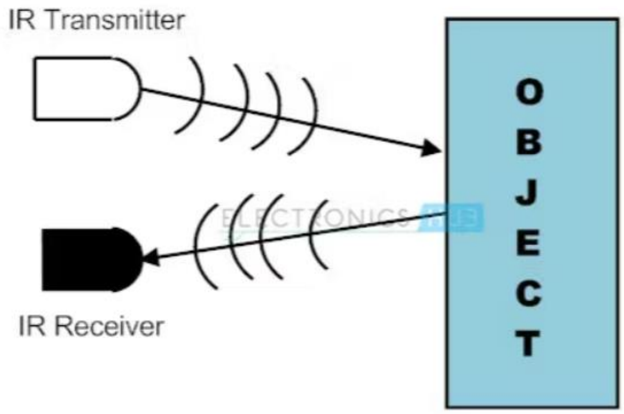
1. How to use IR sensor data with Arduino

Components used – Arduino, jumper wires, IR sensor, LED light, cable

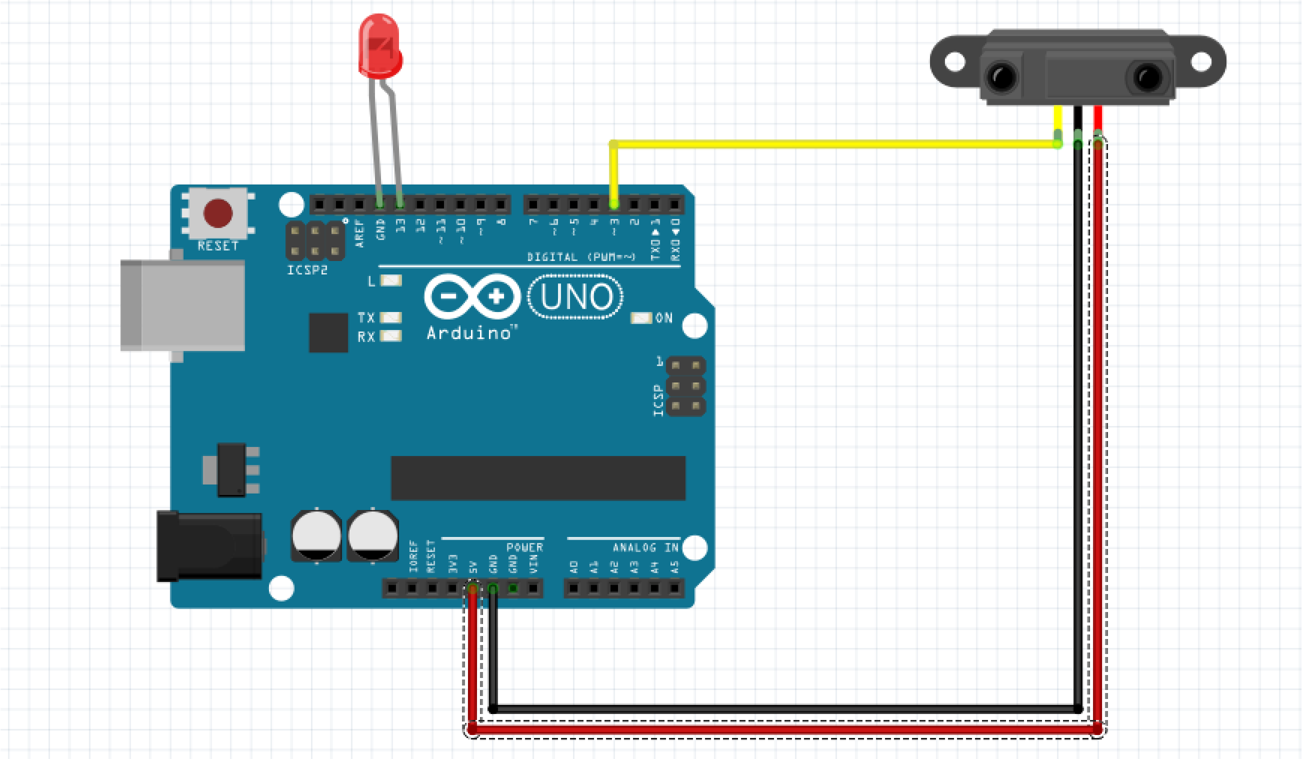
Infrared Sensor is an electronic instrument that is used to detect any type of obstacle/object

We are using an Arduino diecimila here.

The sender and receiver end of the IR sensor is as below.



The circuit connection is like

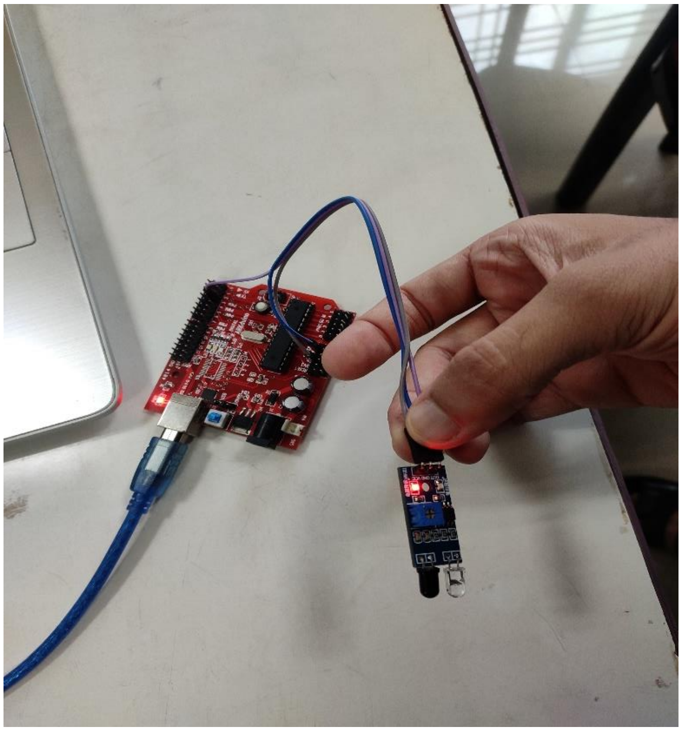


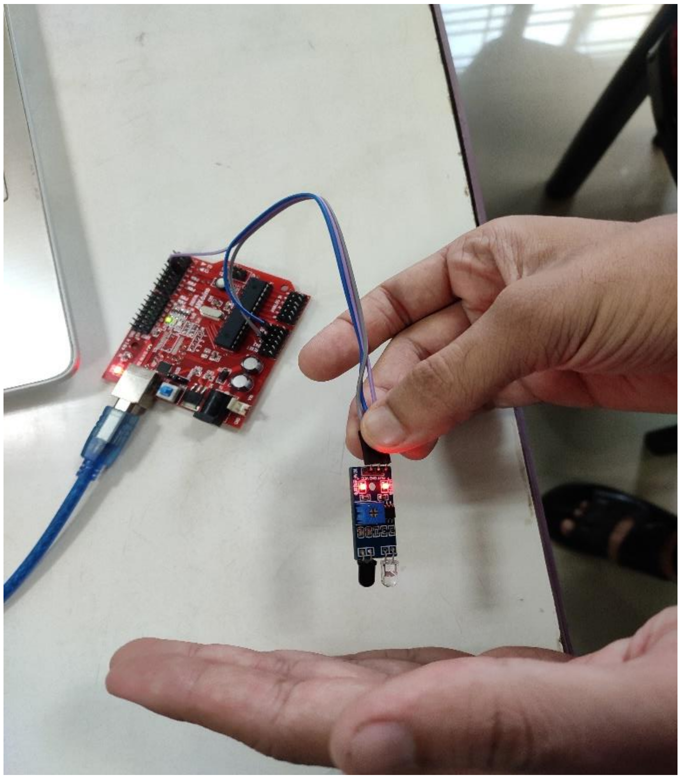
Code

```
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(3,INPUT);
  Serial.begin(9600);
}
void loop()
{
  if (digitalRead(3)== LOW)
  {
    digitalWrite(13,HIGH);
    delay(10);
  }
  else
  {
    digitalWrite(13,LOW);
    delay(20);
  }
}
```

reference

Output pictures



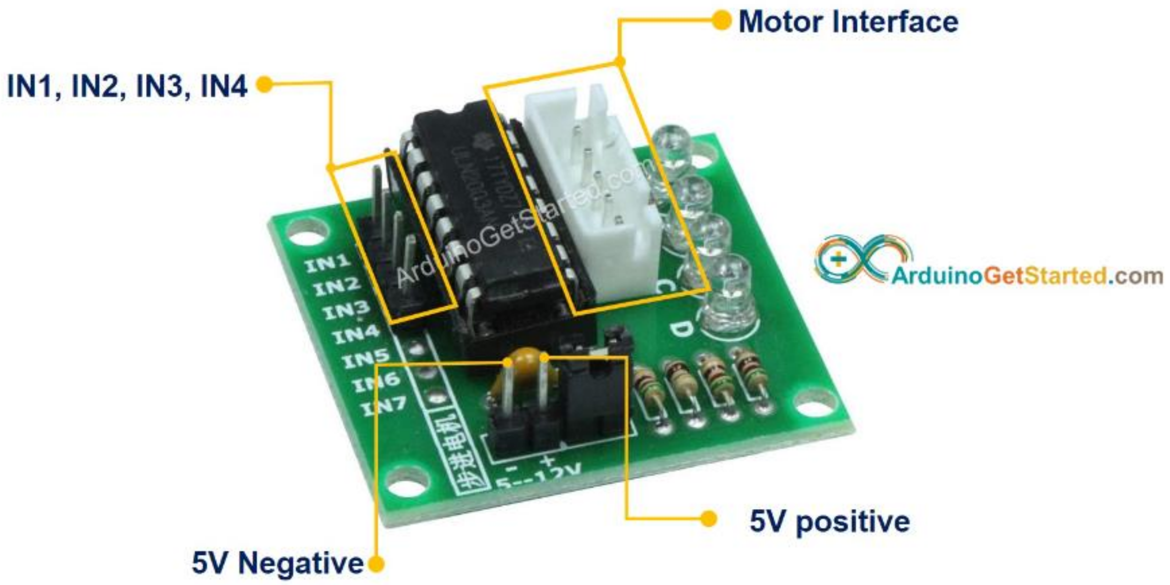


DATE – 07/07/2022

2. Stepper motor connection with Arduino

Components used – Arduino diecimila, 28BYJ-48 stepper motor, driver

The driver is connected with the Arduino with 5v power supply and ground.

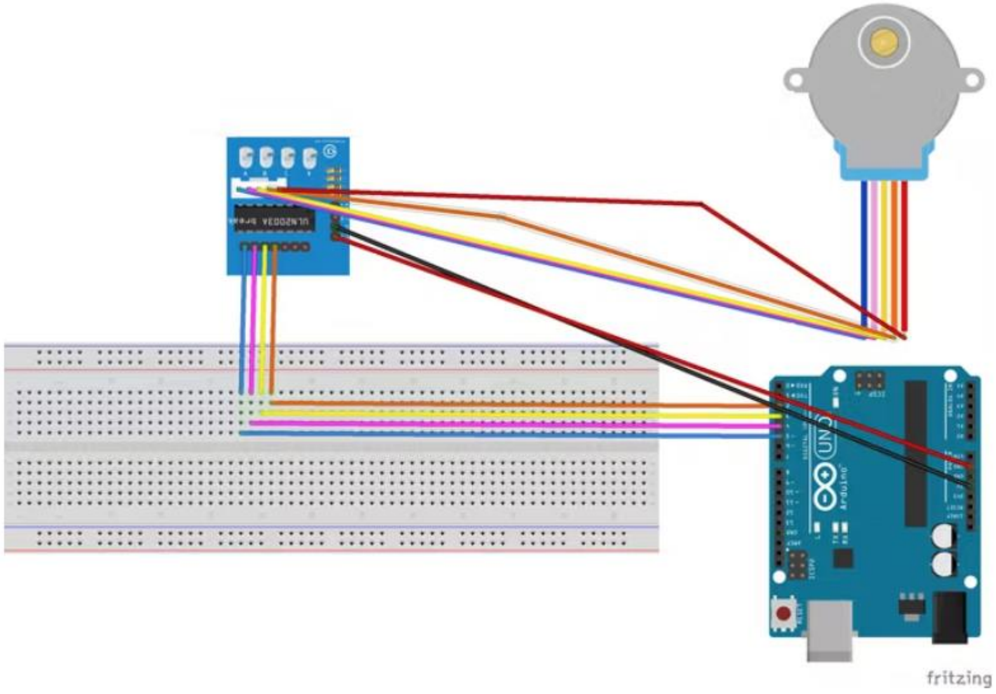


Connection

The IN1, IN2, IN3, IN4 are connected to the 12, 11, 10 and 9 digital pins of the Arduino diecimila.

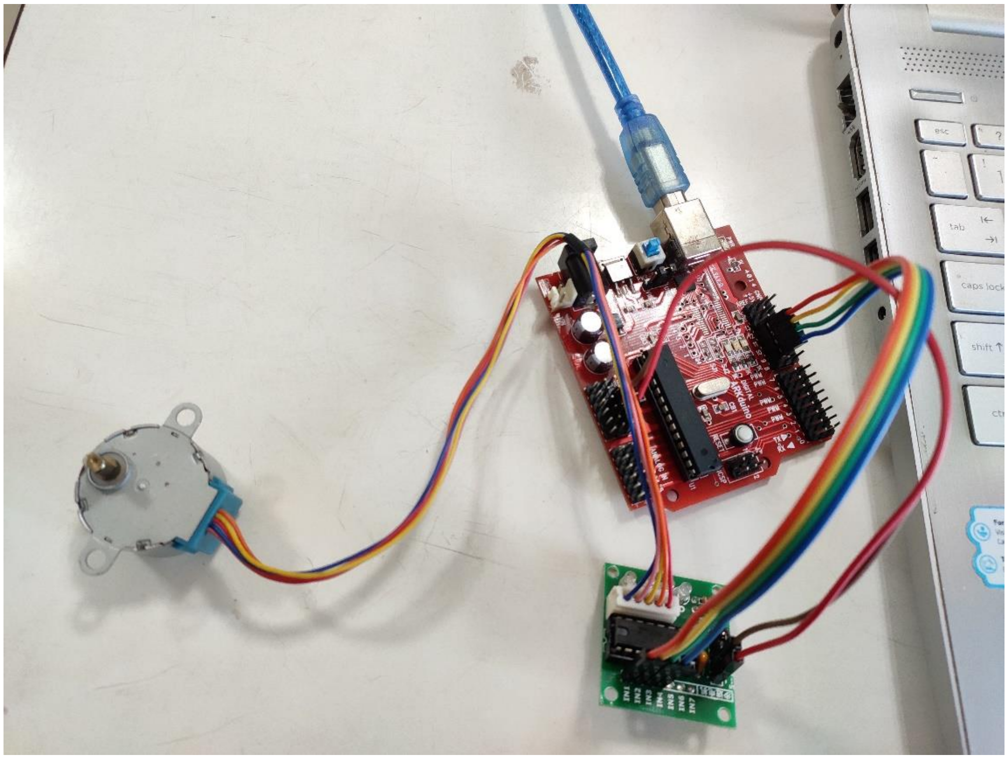
The step motor 28BYJ-48 (5V DC) is connected with the driver in the provided port.

The Arduino is connected to the system and the code is uploaded.



Circuit Diagram for motor

Circuit photo



Code

```
#define STEPPER_PIN_1 9
#define STEPPER_PIN_2 10
#define STEPPER_PIN_3 11
#define STEPPER_PIN_4 12
int step_number = 0;
void setup() {
```

```
pinMode(STEPPER_PIN_1, OUTPUT);
pinMode(STEPPER_PIN_2, OUTPUT);
pinMode(STEPPER_PIN_3, OUTPUT);
pinMode(STEPPER_PIN_4, OUTPUT);

}

void loop() {

  OneStep(false);
  delay(2);


}



void OneStep(bool dir){
    if(dir){
switch(step_number){
  case 0:
  digitalWrite(STEPPER_PIN_1, HIGH);
  digitalWrite(STEPPER_PIN_2, LOW);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, LOW);
  break;
  case 1:
  digitalWrite(STEPPER_PIN_1, LOW);
  digitalWrite(STEPPER_PIN_2, HIGH);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, LOW);
  break;
  case 2:
  digitalWrite(STEPPER_PIN_1, LOW);
  digitalWrite(STEPPER_PIN_2, LOW);
  digitalWrite(STEPPER_PIN_3, HIGH);
  digitalWrite(STEPPER_PIN_4, LOW);
  break;
  case 3:
  digitalWrite(STEPPER_PIN_1, LOW);
  digitalWrite(STEPPER_PIN_2, LOW);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, HIGH);
  break;
}
```

```
}else{
  switch(step_number){
case 0:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, HIGH);
break;
case 1:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, HIGH);
digitalWrite(STEPPER_PIN_4, LOW);
break;
case 2:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, HIGH);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, LOW);
break;
case 3:
digitalWrite(STEPPER_PIN_1, HIGH);
```

```
  digitalWrite(STEPPER_PIN_2, LOW);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, LOW);



}
  }
step_number++;
  if(step_number > 3){
    step_number = 0;
  }
}
```

DATE – 08/07/2022

Attempts to upload sensor data to online

Wifi scanned but the connection not made

The esp device is connected with the computer with a micro usb cable.

Code

```
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop()
{
    Serial.println("scan start");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0) {
        Serial.println("no networks found");
    } else {
        Serial.print(n);
```

DATE – 11/07/2022

Manual data uploaded to thingspeak.

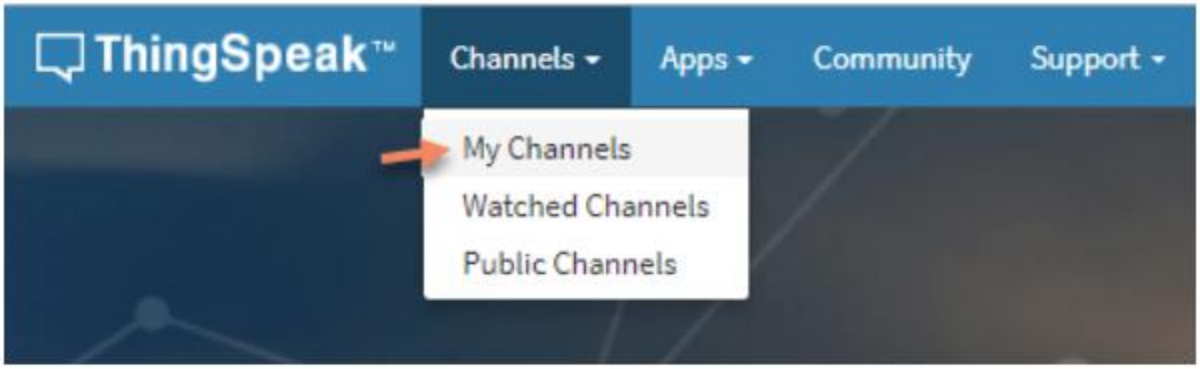
Normal signal from heart rate sensor worked, but the bpm convertor did not work with esp.

Thingspeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to Thingspeak from your devices, create instant visualization of live data, and send alerts.

To use thingspeak in our project, we have to first create a new channel and connect that channel with our code with the channel ID and API keys.

Creating a channel in thingspeak reference

1- 1st we have to sign in by giving the mathworks credentials.
2- Then go to channels->my channels



3 - On the channel page click on new channel.

4 - Then select and name the number of filds we want to upload.

5 - Finally save the channel.

We can access the channel ID and the read and write API keys now.

Code to connect the esp to the wifi only

```
#include "WiFi.h"

const char* ssid = "dlink";
const char* password =  "";

void setup() {

  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }

  Serial.println("Connected to the WiFi network");

}

void loop() {}
```

DATE – 12/07/2022

Fedora downloaded and installation on virtual box.

Brave browser installation on fedora.

DATE – 13/07/2022

VS code installation on fedora

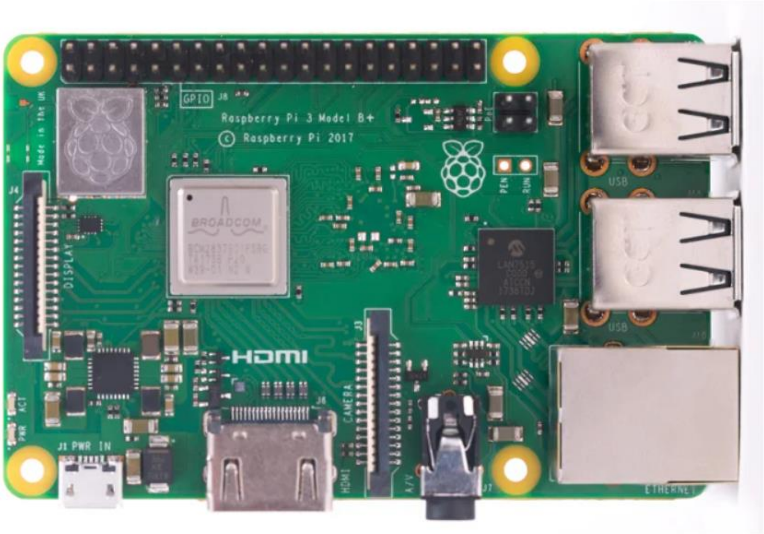
Downloading raspberry pi imager, vnc viewer and putty.

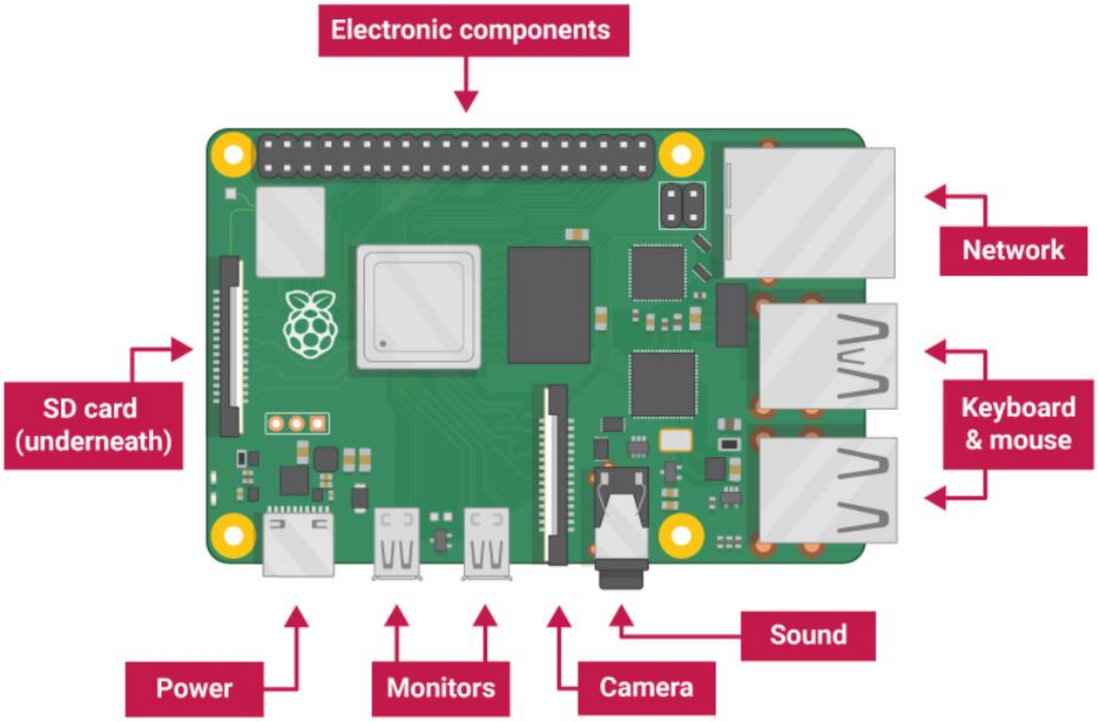
DATE – 14/07/2022

<u>Raspberry pi set up</u>

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything we'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

We can use raspberry pi to learn code, or to do projects related to IoT and electronics. Raspberry pi operates in a open source eco system. Its main supported operating system PI os is open source also.





Tutorial reference

First we have to download and install putty and vnc viewer software in our laptop. Then we insert the sd card in the raspberry pi and connect it with a LAN cable with the laptop. We have to provide the raspberry pi a external power supply through the micro usb also.

Putty configuration

1. We open putty and enter the host name as raspberrypi.local and leave all those settings as default and hit open.
2. If everything goes right, we will see a pop up warning, press yes.
3. Now the raspberry terminal will open. We have to enter the id as “pi” and password as “raspberry”. We can do things from here also. It is the same raspberry pi terminal.
4. Now we want to view our raspberry interface through vnc viewer. So type the below command in the raspberry pi terminal.

```
sudo raspi -config
```

5. It will open a configuration menu. We will go to interfacing options. Choose vnc -> enable it

6. The set up is done here. Now we will open the vnc viewer and enter our address “raspberrypi.local”.
7. It will ask for the user id and password. We will enter that as we have set it up and click ok. Our raspberry pi interface window will come now.

Configuration tutorial reference

Link1

Link2

DATE – 19/07/2022

Soil moisture testing

Soil moisture sensor is used to build smart gardens when we need an automated system for irrigation purpose.

The fork-shaped probe with two exposed conductors, acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water content in the soil.

This resistance is inversely proportional to the soil moisture:

- The more water in the soil means better conductivity and will result in a lower resistance.
- The less water in the soil means poor conductivity and will result in a higher resistance.

The sensor produces an output voltage according to the resistance, which by measuring we can determine the moisture level.

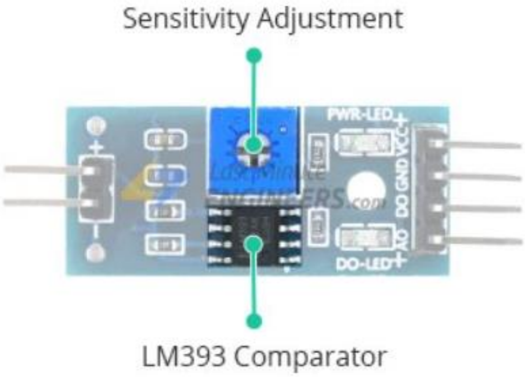
A typical soil moisture sensor has two components.

The Probe - The sensor contains a fork-shaped probe with two exposed conductors that goes into the soil or anywhere else where the water content is to be measured.

The module - The sensor also contains an electronic module that connects the probe to the Arduino.

The module produces an output voltage according to the resistance of the probe and is made available at an Analog Output (AO) pin.

The same signal is fed to a LM393 High Precision Comparator to digitize it and is made available at an Digital Output (DO) pin.

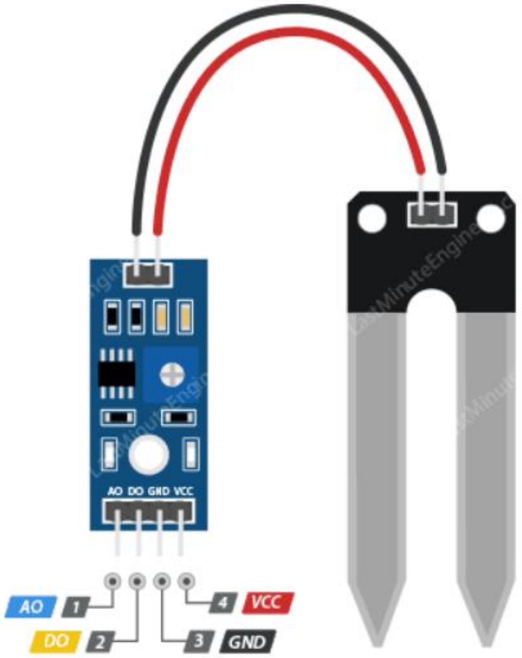


Apart from this, the module has two LEDs. The Power LED will light up when the module is powered. The Status LED will light up when the digital output goes LOW.

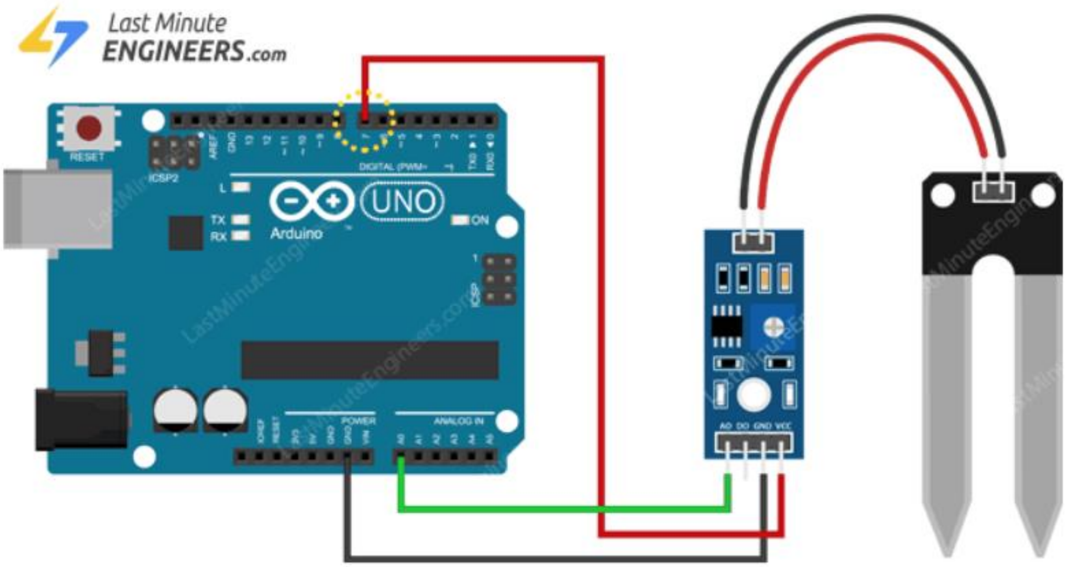
Connecting the probe with the module

There are 4 pins in the module.

1. The A0 pin gives us an analog signal between the supply value to 0V and will be connected to one of the analog inputs on your Arduino.
2. The digital output pin gives Digital output of internal comparator circuit. You can connect it to any digital pin on an Arduino or directly to a 5V relay or similar device.
3. The VCC pin supplies power for the sensor. It is recommended to power the sensor with between 3.3V – 5V.
4. Another is an GND pin.



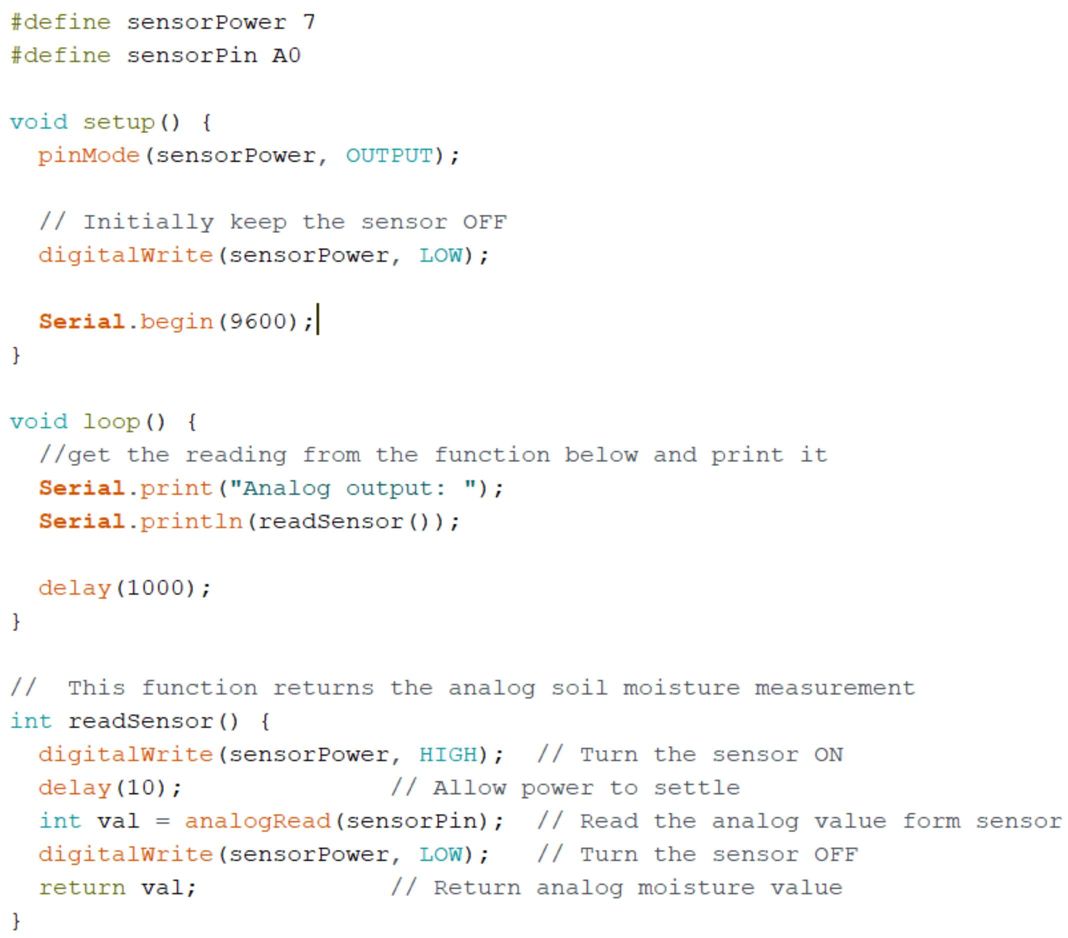
Connection



The VCC pin is connected with the 3.3v/5v power supply and the ground pin to the GND.

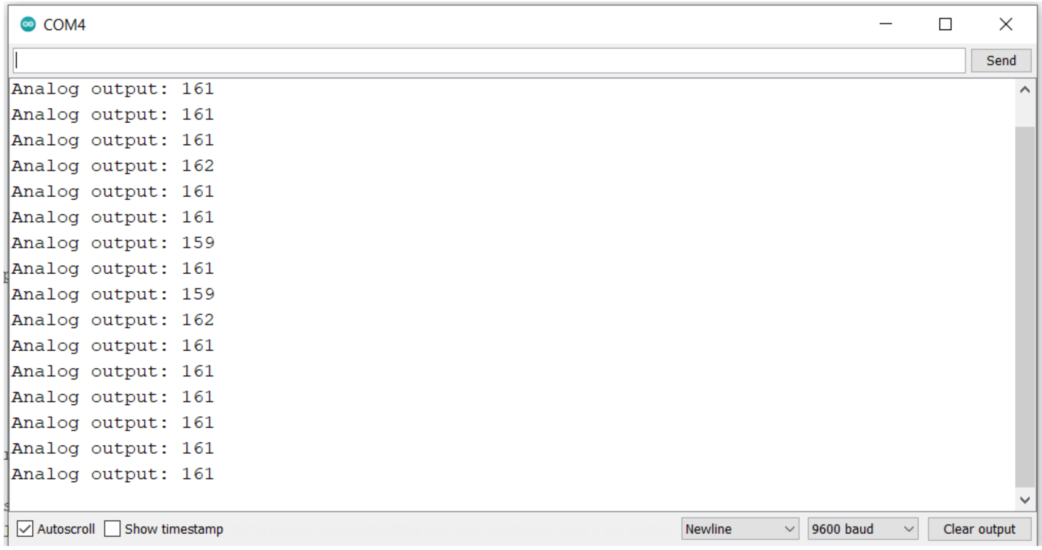
The A0 pin on module is connected with the A0 ADC pin on Arduino.

Program to get only the Analog signal value



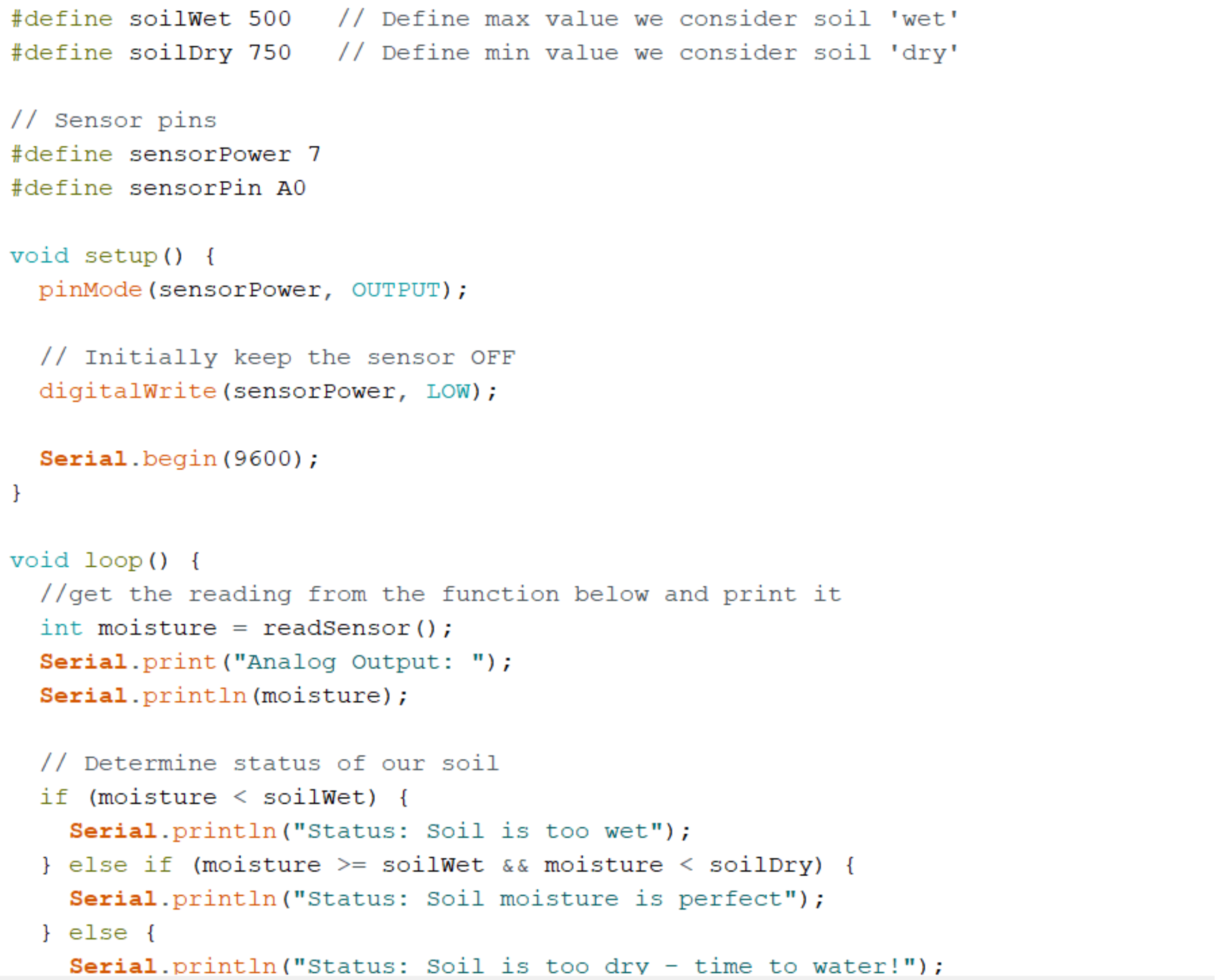
```
#define sensorPower 7
#define sensorPin A0

void setup() {
  pinMode(sensorPower, OUTPUT);

  // Initially keep the sensor OFF
  digitalWrite(sensorPower, LOW);

  Serial.begin(9600);
}

void loop() {
  //get the reading from the function below and print it
  Serial.print("Analog output: ");
  Serial.println(readSensor());

  delay(1000);
}

//  This function returns the analog soil moisture measurement
int readSensor() {
  digitalWrite(sensorPower, HIGH);  // Turn the sensor ON
  delay(10);                        // Allow power to settle
  int val = analogRead(sensorPin);  // Read the analog value form sensor
  digitalWrite(sensorPower, LOW);   // Turn the sensor OFF
  return val;                       // Return analog moisture value
}
```

Output

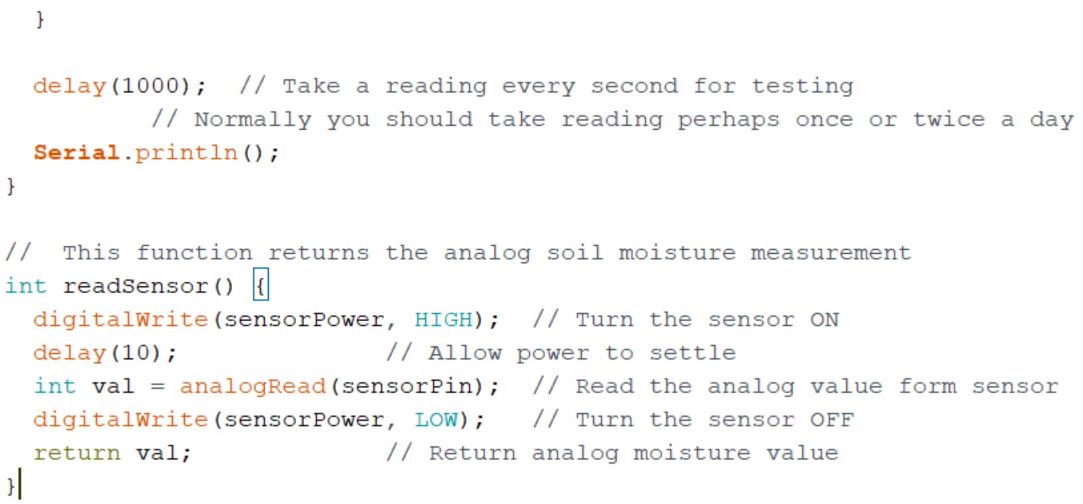


```
Analog output: 161
Analog output: 161
Analog output: 161
Analog output: 162
Analog output: 161
Analog output: 161
Analog output: 159
Analog output: 161
Analog output: 159
Analog output: 162
Analog output: 161
Analog output: 161
Analog output: 161
Analog output: 161
Analog output: 161
Analog output: 161
```

Program to set a threshold and check relative wetness

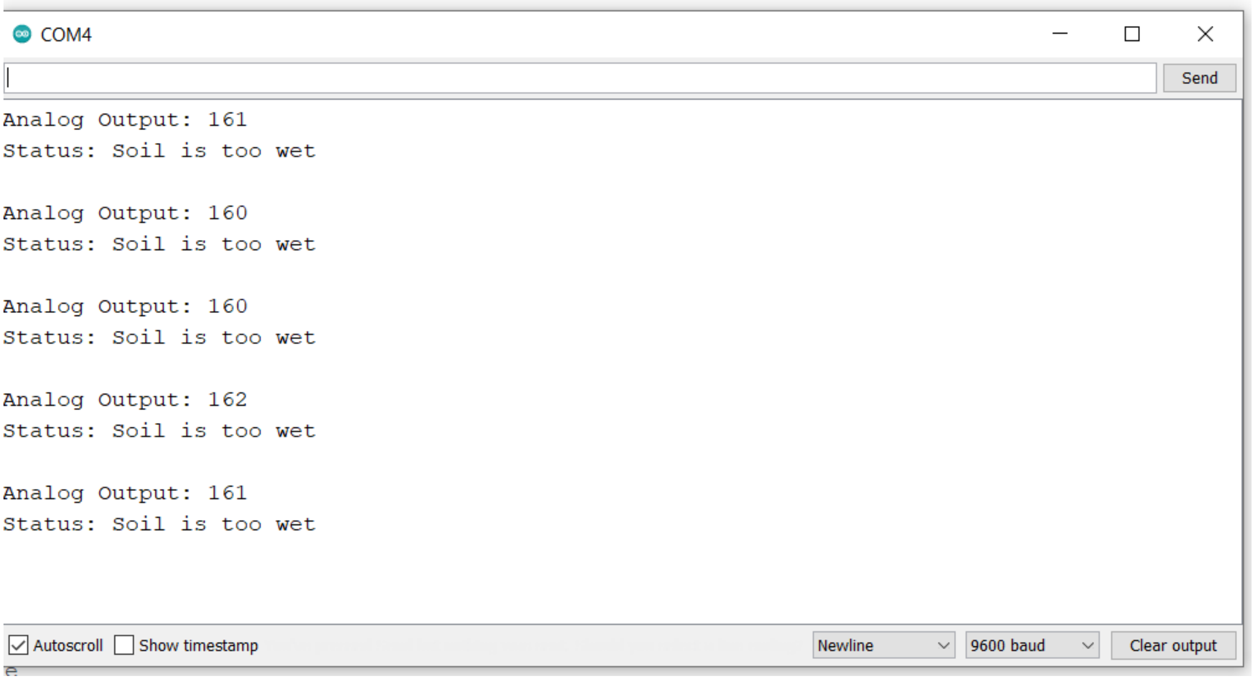


```
#define soilWet 500   // Define max value we consider soil 'wet'
#define soilDry 750   // Define min value we consider soil 'dry'

// Sensor pins
#define sensorPower 7
#define sensorPin A0

void setup() {
  pinMode(sensorPower, OUTPUT);

  // Initially keep the sensor OFF
  digitalWrite(sensorPower, LOW);

  Serial.begin(9600);
}

void loop() {
  //get the reading from the function below and print it
  int moisture = readSensor();
  Serial.print("Analog Output: ");
  Serial.println(moisture);

  // Determine status of our soil
  if (moisture < soilWet) {
    Serial.println("Status: Soil is too wet");
  } else if (moisture >= soilWet && moisture < soilDry) {
    Serial.println("Status: Soil moisture is perfect");
  } else {
    Serial.println("Status: Soil is too dry - time to water!");
```



```
  }

  delay(1000);  // Take a reading every second for testing
          // Normally you should take reading perhaps once or twice a day
  Serial.println();
}

//  This function returns the analog soil moisture measurement
int readSensor() {
  digitalWrite(sensorPower, HIGH);  // Turn the sensor ON
  delay(10);              // Allow power to settle
  int val = analogRead(sensorPin);  // Read the analog value form sensor
  digitalWrite(sensorPower, LOW);   // Turn the sensor OFF
  return val;             // Return analog moisture value
}
```

Output

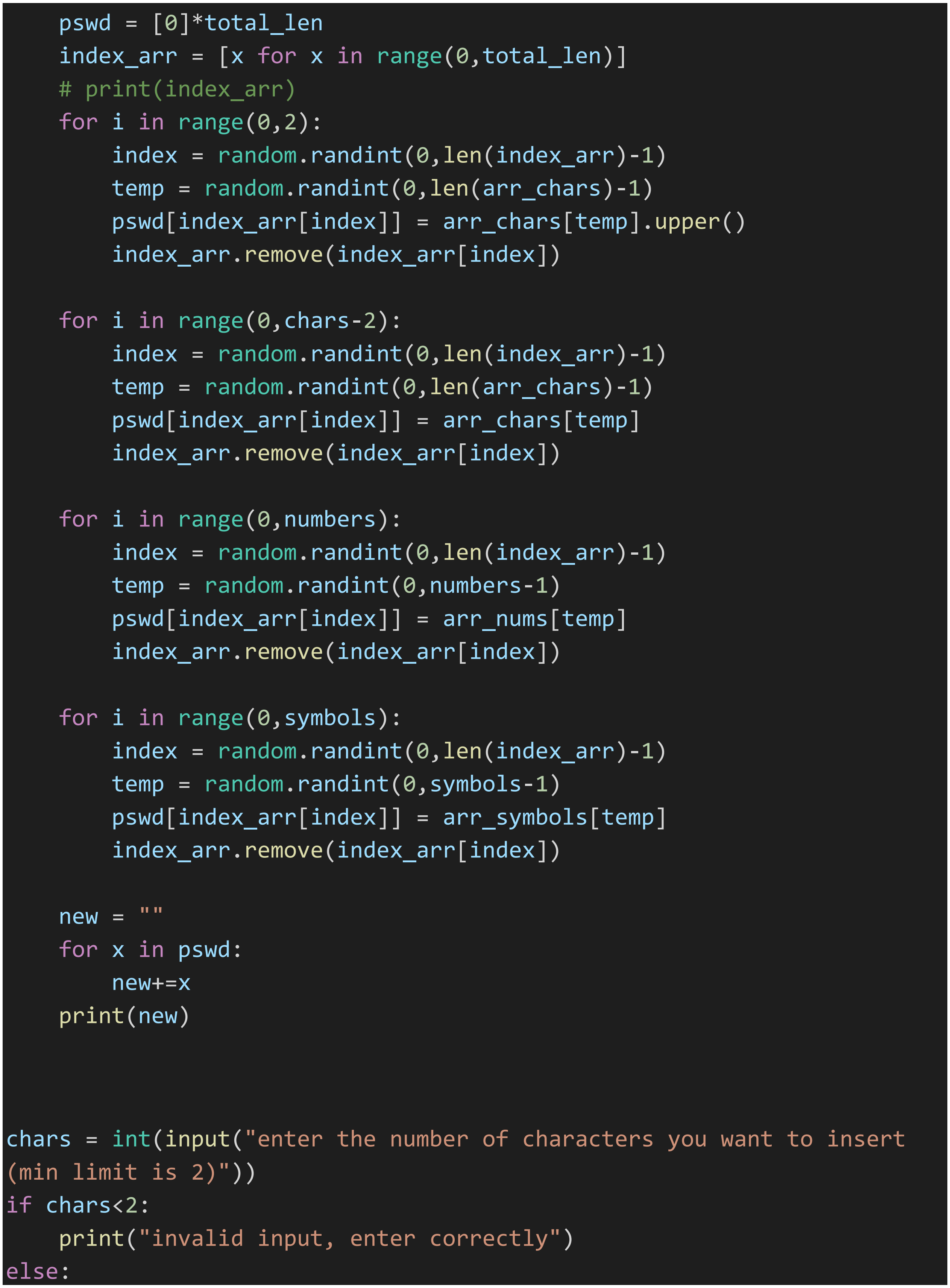


```
Analog Output: 161
Status: Soil is too wet

Analog Output: 160
Status: Soil is too wet

Analog Output: 160
Status: Soil is too wet

Analog Output: 162
Status: Soil is too wet

Analog Output: 161
Status: Soil is too wet
```

reference

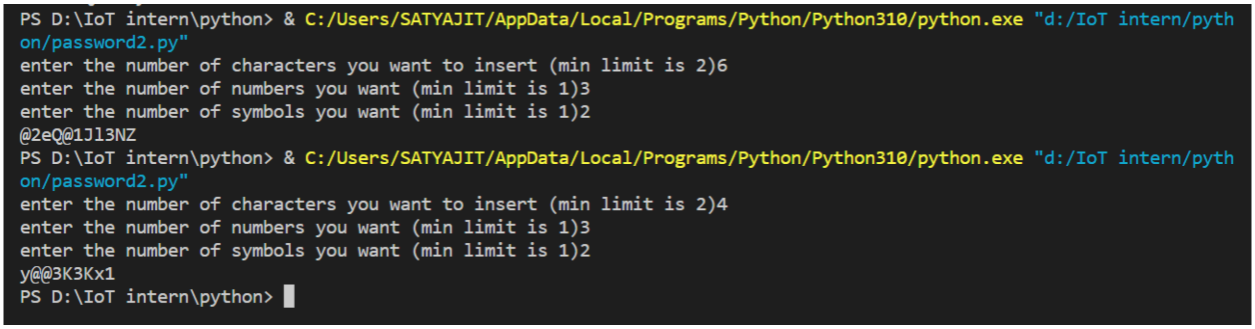
Password generator program 2

```
import random
arr_chars = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K",
"L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y",
"Z", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m",
"n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]
arr_nums = ['1','2','3','4','5','6','7','8','9']
arr_symbols =
['!','@','#','$','%','^','&','*','(',')','[',']','{','}']

def generate(chars,numbers,symbols):
    total_len = chars+numbers+symbols
```



```python
    pswd = [0]*total_len
    index_arr = [x for x in range(0,total_len)]
    # print(index_arr)
    for i in range(0,2):
        index = random.randint(0,len(index_arr)-1)
        temp = random.randint(0,len(arr_chars)-1)
        pswd[index_arr[index]] = arr_chars[temp].upper()
        index_arr.remove(index_arr[index])

    for i in range(0,chars-2):
        index = random.randint(0,len(index_arr)-1)
        temp = random.randint(0,len(arr_chars)-1)
        pswd[index_arr[index]] = arr_chars[temp]
        index_arr.remove(index_arr[index])

    for i in range(0,numbers):
        index = random.randint(0,len(index_arr)-1)
        temp = random.randint(0,numbers-1)
        pswd[index_arr[index]] = arr_nums[temp]
        index_arr.remove(index_arr[index])

    for i in range(0,symbols):
        index = random.randint(0,len(index_arr)-1)
        temp = random.randint(0,symbols-1)
        pswd[index_arr[index]] = arr_symbols[temp]
        index_arr.remove(index_arr[index])

    new = ""
    for x in pswd:
        new+=x
    print(new)



chars = int(input("enter the number of characters you want to insert (min limit is 2)"))
if chars<2:
    print("invalid input, enter correctly")
else:
    numbers = int(input("enter the number of numbers you want (min limit is 1)"))
    if numbers<1:
        print("invalid input, enter correctly")
    else:
        symbols = int(input("enter the number of symbols you want (min limit is 1)"))
        if symbols<1:
            print("invalid input, enter correctly")
        else:
            generate(chars,numbers,symbols)
```

output



```
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/python/password2.py"
enter the number of characters you want to insert (min limit is 2)6
enter the number of numbers you want (min limit is 1)3
enter the number of symbols you want (min limit is 1)2
@2eQ@1J13NZ
PS D:\IoT intern\python> & C:/Users/SATYAJIT/AppData/Local/Programs/Python/Python310/python.exe "d:/IoT intern/python/password2.py"
enter the number of characters you want to insert (min limit is 2)4
enter the number of numbers you want (min limit is 1)3
enter the number of symbols you want (min limit is 1)2
y@@3K3Kx1
PS D:\IoT intern\python> 
```