Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.

Internet of Things Lab

EXPERIMENT 1

Title: Introduction to various sensors and various actuators & its Application.

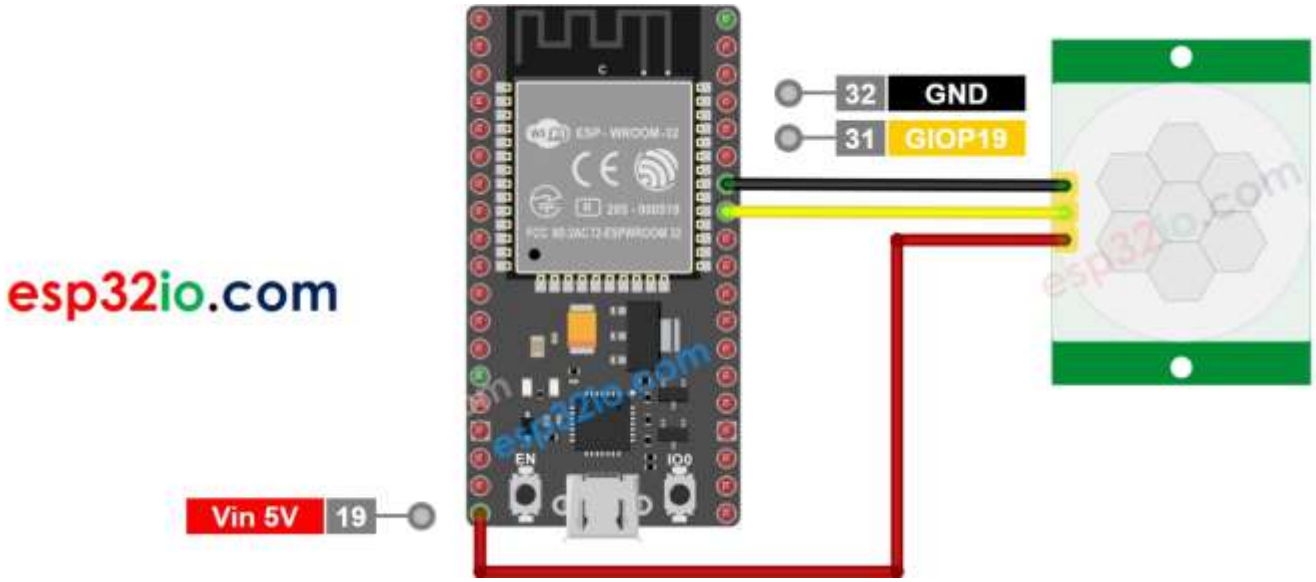
Objective: Introduction to various sensors and various actuators & its Application.

a) PIR Motion Sensor.

- A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view
- Use a pair of pyro-electric sensors to detect heat energy in the surrounding environment. These two sensors sit beside each other, and when the signal differential between the two sensors changes (if a person enters the room, for example), the sensor will engage
- Detect moving objects even in dark with great accuracy.
- PIR sensors can detect the motion of objects without coming in contact with them.

Working: The passive infrared sensor does not radiate energy to space. It receives the infrared radiation from the human body to make an alarm. Any object with temperature is constantly radiating infrared rays to the outside world

Interfacing diagram:



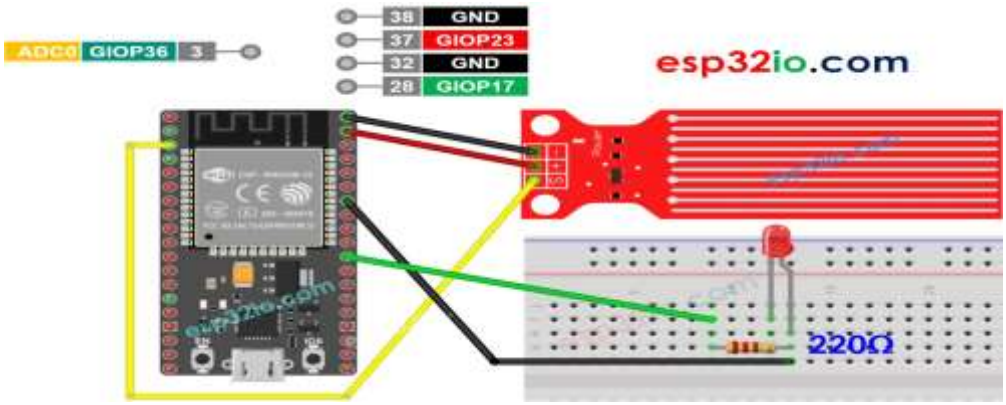
b) float sensor

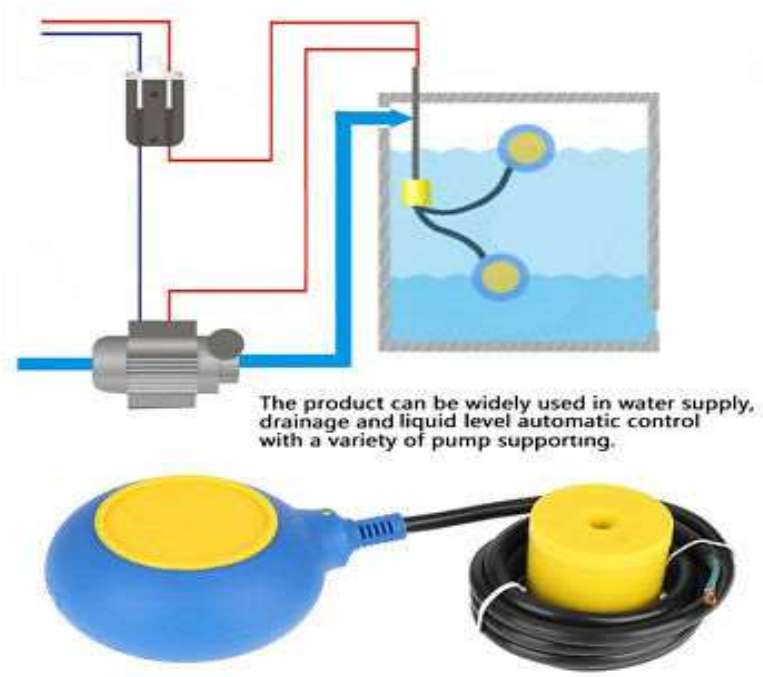
Working: Float level sensors are continuous level sensors featuring a magnetic float that rises and falls as liquid levels change.

The movement of the float creates a magnetic field that actuates a hermetically sealed reed switch located in the stem of the level sensor, triggering the switch to open or close

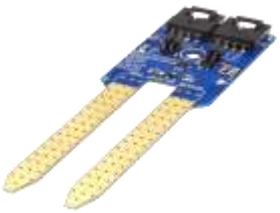
A float switch is a simple device that is used to detect the level of liquid in a tank and control an external component, such as a pump, valve, alarm or other device

Interfacing diagram:





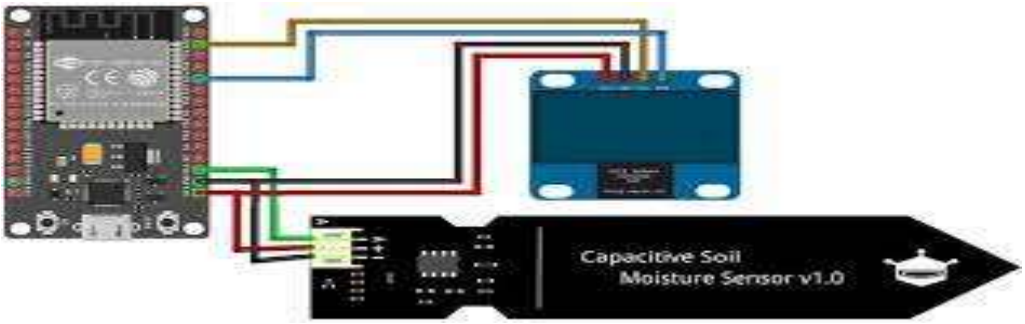
c) Moisture Sensor.



Working: A resistive soil moisture sensor works by using the relationship between electrical resistance and water content to gauge the moisture levels of the soil.

Two probes that are inserted directly into the soil sample

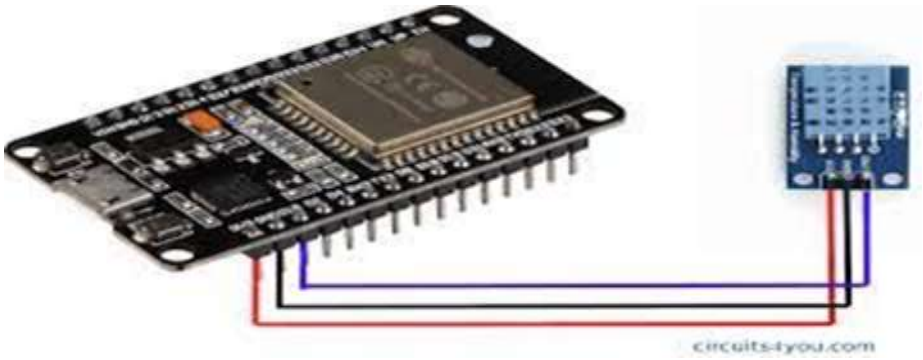
Interfacing diagram:



d) temperature sensor

- **Working:** RTD (Resistance Temperature Detector) As temperature changes, the resistance of any metal changes
- Thermocouples are the most commonly used type of temperature sensor.
- Thermistors
- Semiconductor based ICs.

Interfacing diagram:



e) Touch Sensor.

Working: A touch sensor works like a switch, where when there's contact, touch, or pressure on the surface of a touch sensor, it opens up an electrical circuit and allows currents to flow through it

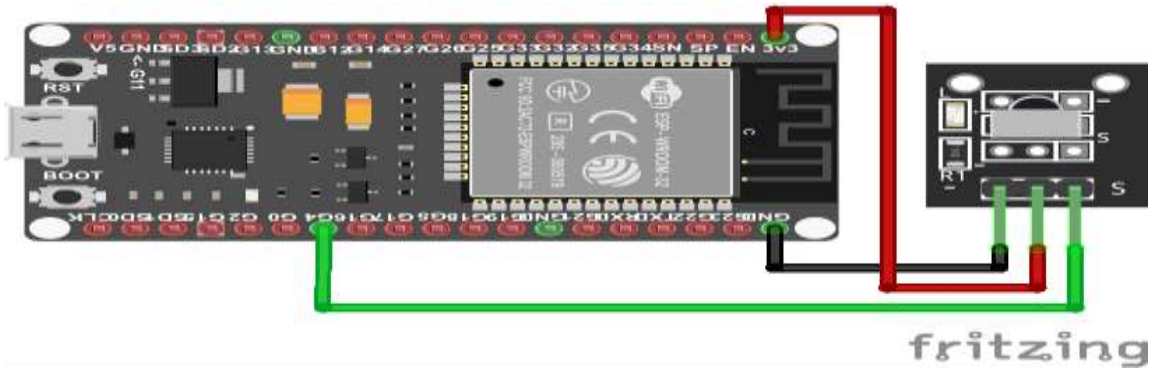
The TTP223-BA6/TTP223N-BA6 TonTouchTM is a touch pad detector IC which offers 1 touch key. The touching detection IC is designed for replacing traditional direct button key with diverse pad size. Low power consumption and wide operating voltage are the contact key features for DC or AC application.

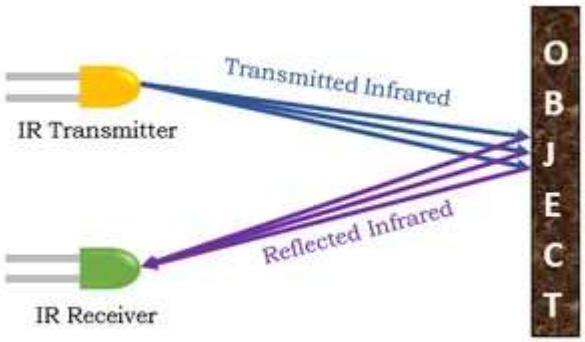
Interfacing diagram:

f) Infrared Sensor.

- **Working:** Active infrared sensors both emit and detect infrared radiation.
- Active IR sensors have two parts: a light emitting diode (LED) and a receiver.
- When an object comes close to the sensor, the infrared light from the LED reflects off of the object and is detected by the receiver.
- Infrared sensors are used in motion detection, night vision, astronomy, art restoration, gas detection, and a variety of other applications

Interfacing diagram:

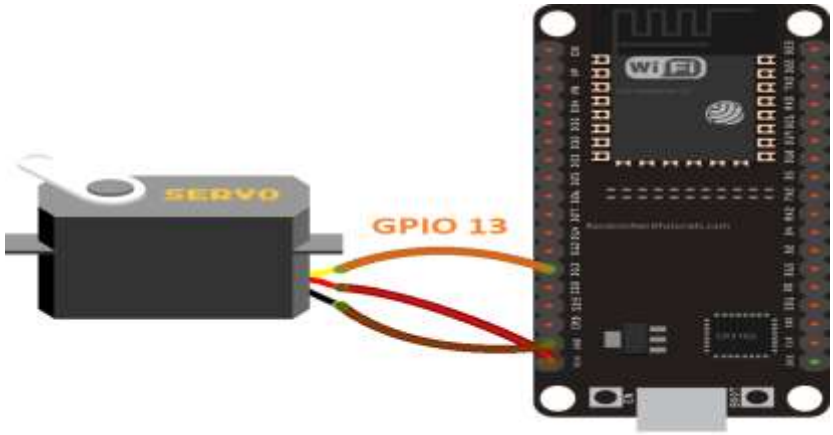




g) Servo Motor

- **Working:** A servo motor is an electromechanical device that produces torque and velocity based on the supplied current and voltage.
- A servo motor works as part of a closed loop system providing torque and velocity as commanded from a servo controller utilizing a feedback device to close the loop.
- The three types include positional rotation, continuous rotation, and linear.

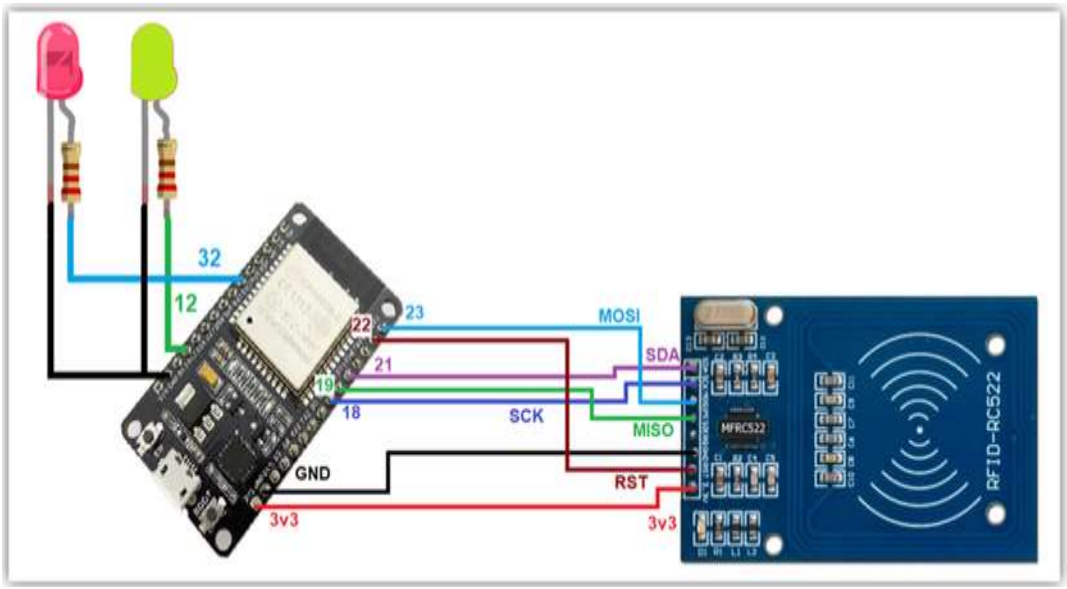
Interfacing diagram:



h) RFID Sensor

- **Working:** The RFID reader is a network-connected device that can be portable or permanently attached.
- It is working under inductive coupling principle, based on a radio frequency or radio waves.
- RIFDs uses electromagnetic field to identify objective or tracking the objects automatically even 100 meters distance.

Interfacing diagram:

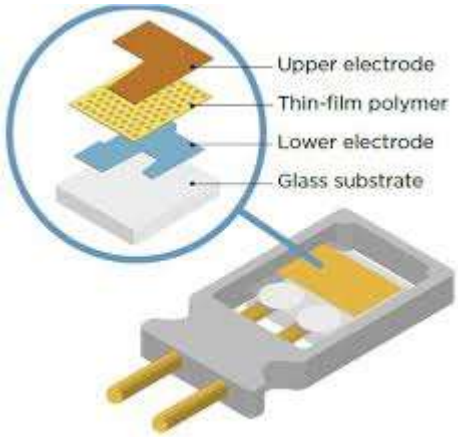


ESP32	RC522
18	SCK
19	MISO
21	SDA
22	RST
23	MOSI
GND	GND
3v3	3v3
LedVermelho	32
LedVerde	12

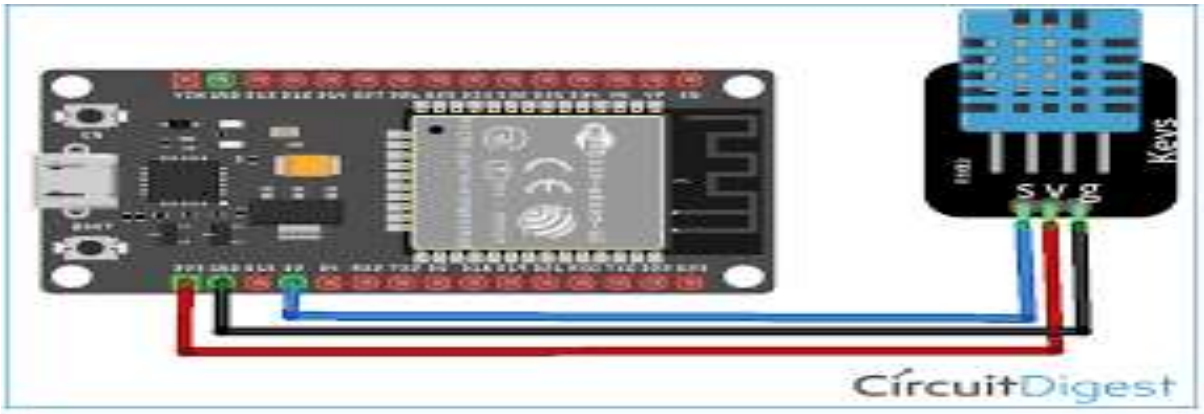
i) Humidity sensor

Working:

- Humidity sensors work by detecting changes that alter electrical currents or temperature in the air.
- There are three basic types of humidity sensors: capacitive, resistive and thermal.
- All three types will monitor minute changes in the atmosphere in order to calculate the humidity in the air
- Absolute humidity is a measure of the mass of the water vapour present in a specified volume.
- Because the mass of water vapour is difficult to measure, a more common measurement called relative humidity is used. Relative humidity (RH) is the percentage of the amount of water that the air can hold at a given temperature.



Interfacing diagram:



Conclusion: Here we have learnt various sensors and various actuators & its Application. There are different sensors which are used for sensing and actuators which are used to sense the motion are operated by interfacing it with esp32 development board and the output is observed respectively.

1. Sensors take input from physical quantities and turn them into electrical signals.

Actuators convert electrical quantities eg. Current or voltage into physical quantities, motion, sound, etc.

2. A mircocontroller is used to take data from the sensor, process it and drive the actuator, which in turn performs some task in the real world.

3. Sensors can collect information about the real world like distance, temperature, humidity, pressure, force, weight, ambient light, etc.

4. Based on these parameters, sensor modules can give us more enhanced information like motion detection, rpm measurement, etc.

5. Examples of actuators are servo motors, dc motors, magnetic actuators, hydraulics, pneumatics, etc.

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

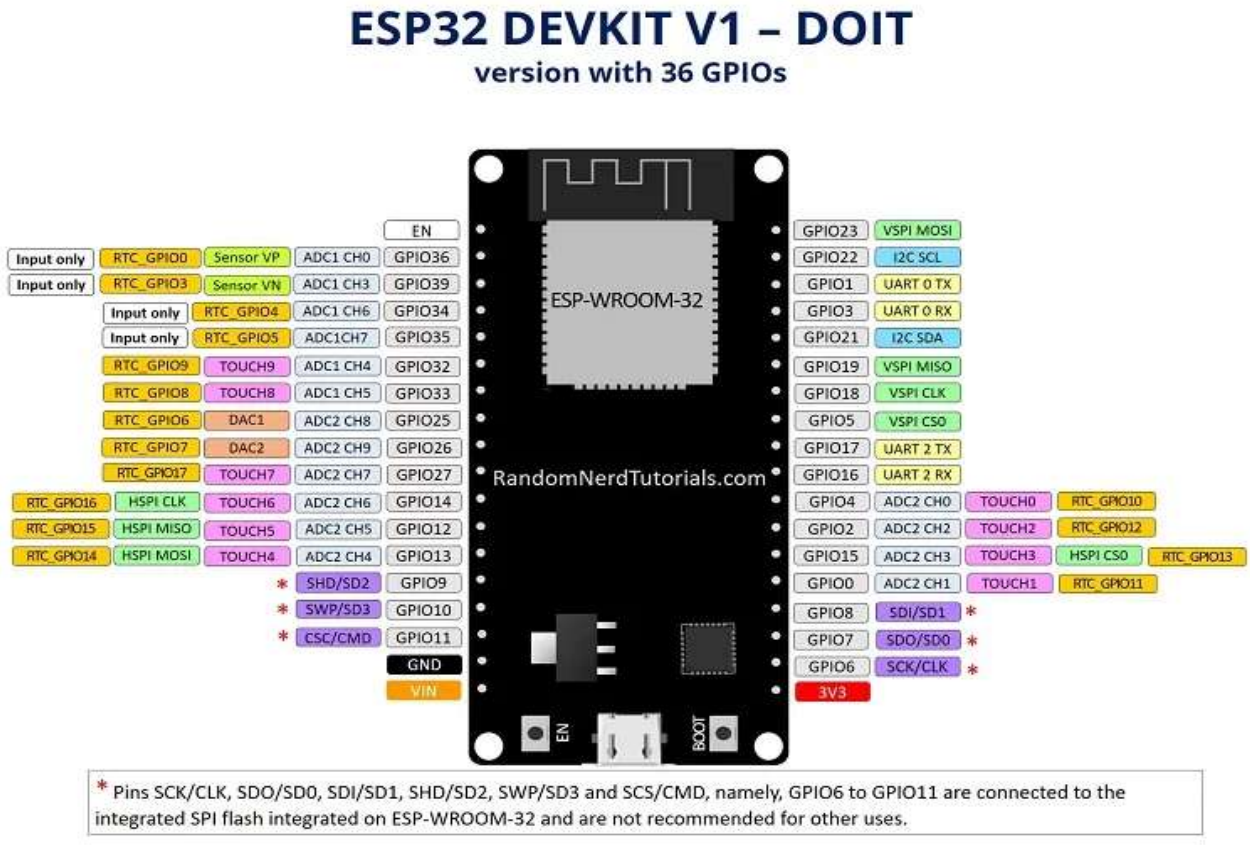
A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.
Internet of Things Lab

EXPERIMENT 2

Title: Introduction to ESP32 and Arduino IDE/Visual Studio Code.

Objective: To study Introduction to ESP32 and Arduino IDE/Visual Studio Code.



Theory:

The ESP32 chip comes with 48 pins with multiple functions. Not all pins are exposed in all ESP32

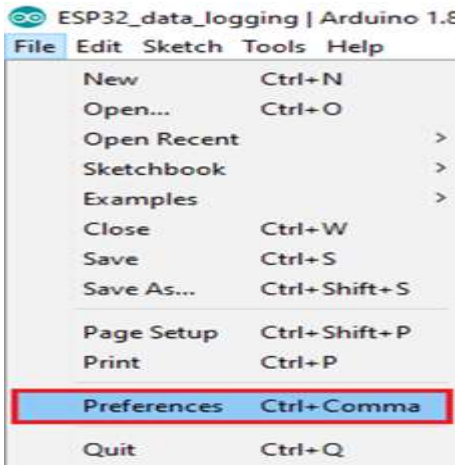
development boards, and some pins cannot be used.

The ADC (analog to digital converter) and DAC (digital to analog converter) features are assigned to specific static pins. However, you can decide which pins are UART, I2C, SPI, PWM, etc – you just need to assign them in the code. This is possible due to the ESP32 chip's multiplexing feature.

Arduino is an open-source physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing and MaxMSP).One very big advantage with ESP32, which has aided its quick adoption and massive popularity, is the provision for programming the ESP32 within the Arduino IDE.

To install the ESP32 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to **File**> **Preferences**

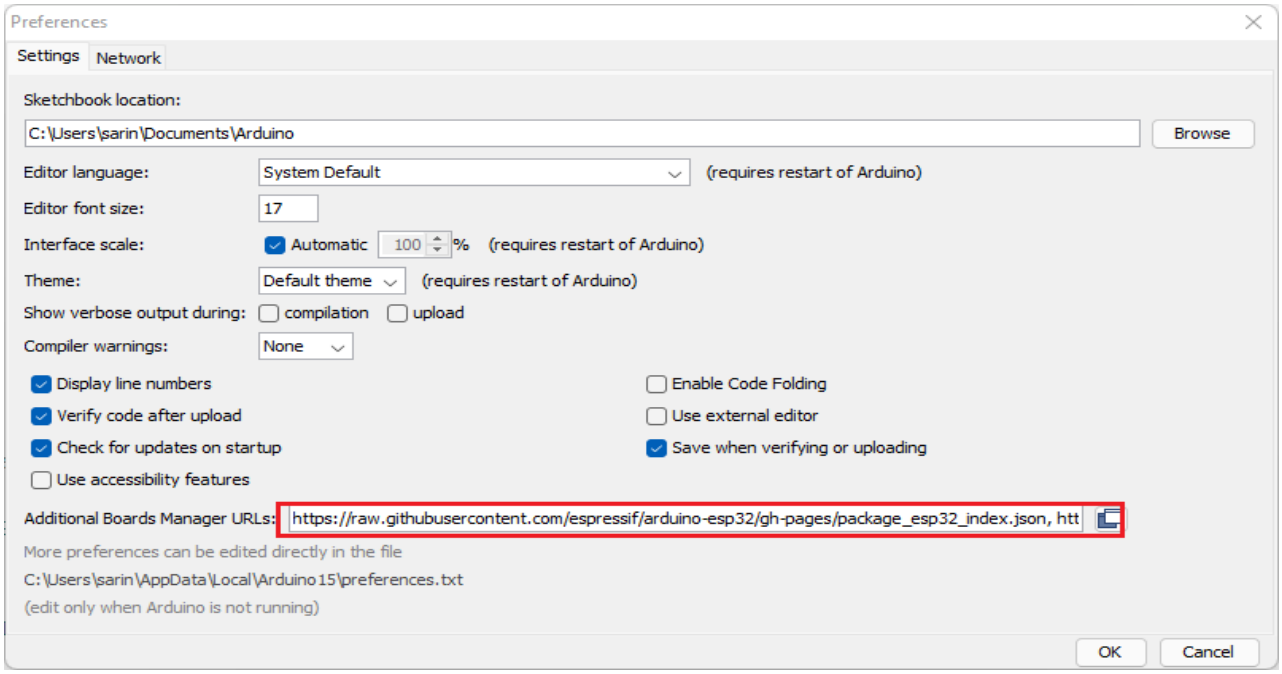


2.Enter the following into the "Additional Board Manager URLs" field:

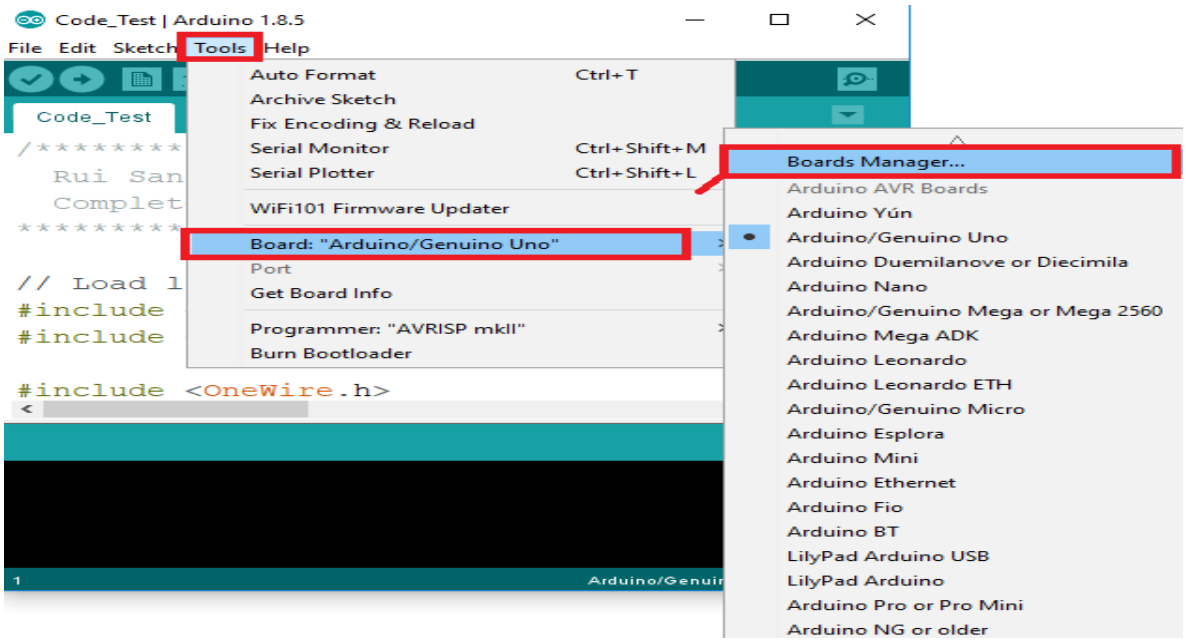
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-

pages/package_esp32_index.json

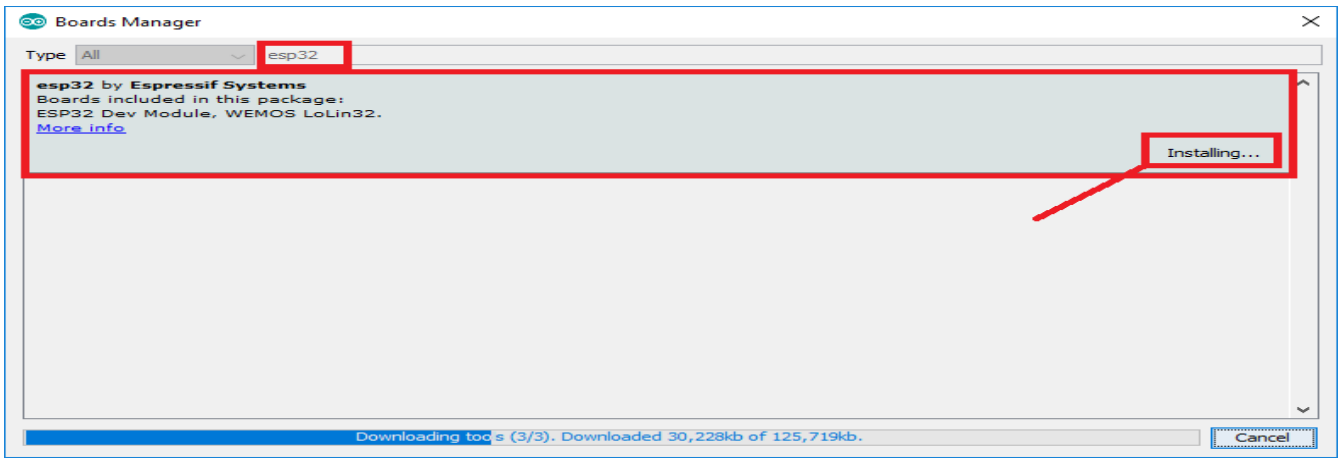
Then, click the “OK” button:



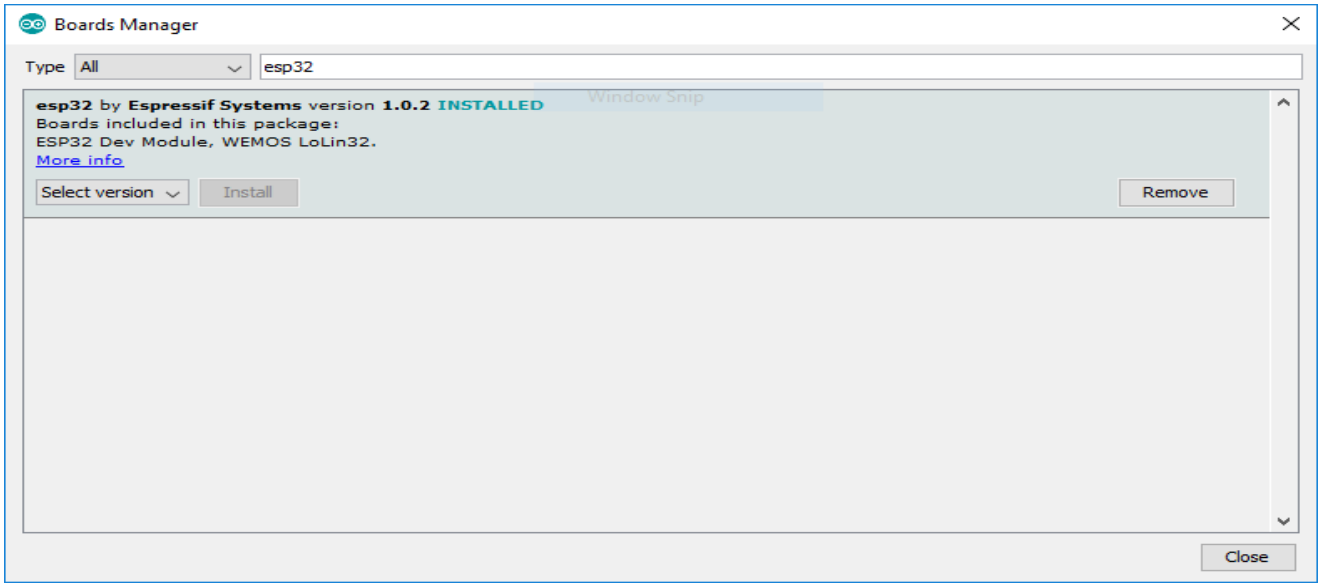
Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma
2.Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager**



3. Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**“:

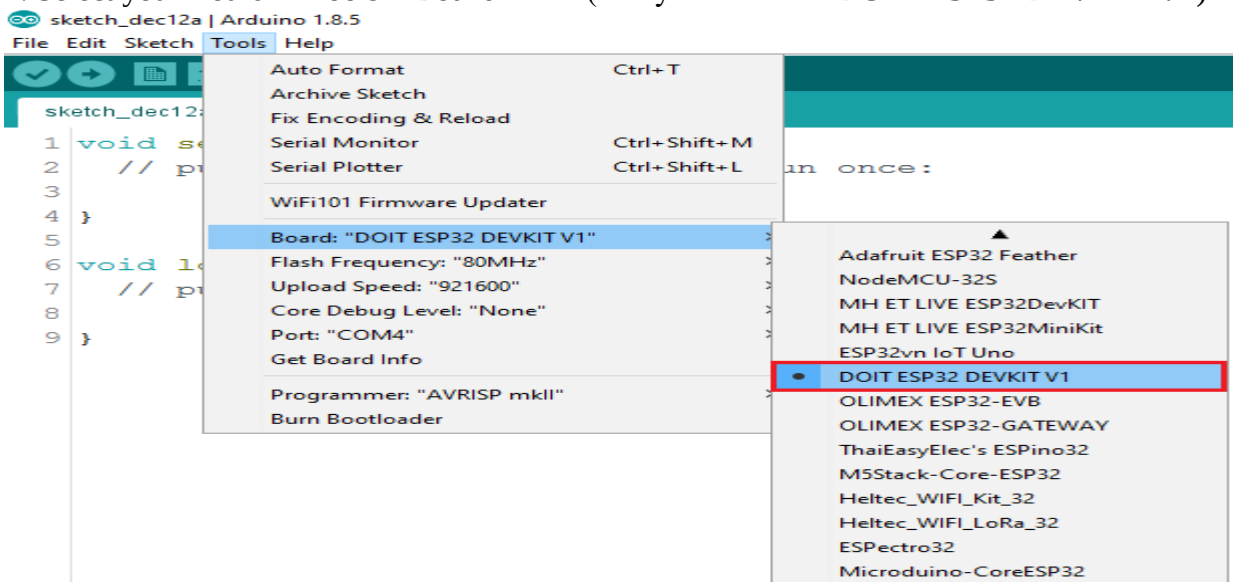


4. That's it. It should be installed after a few seconds.

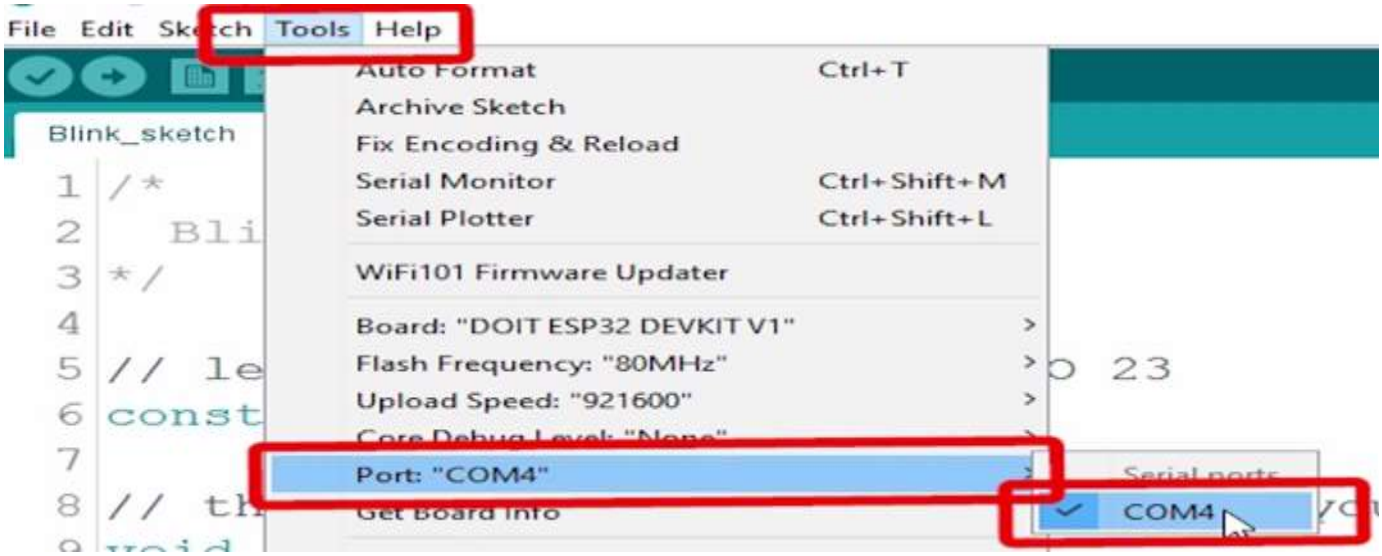


Testing the Installation

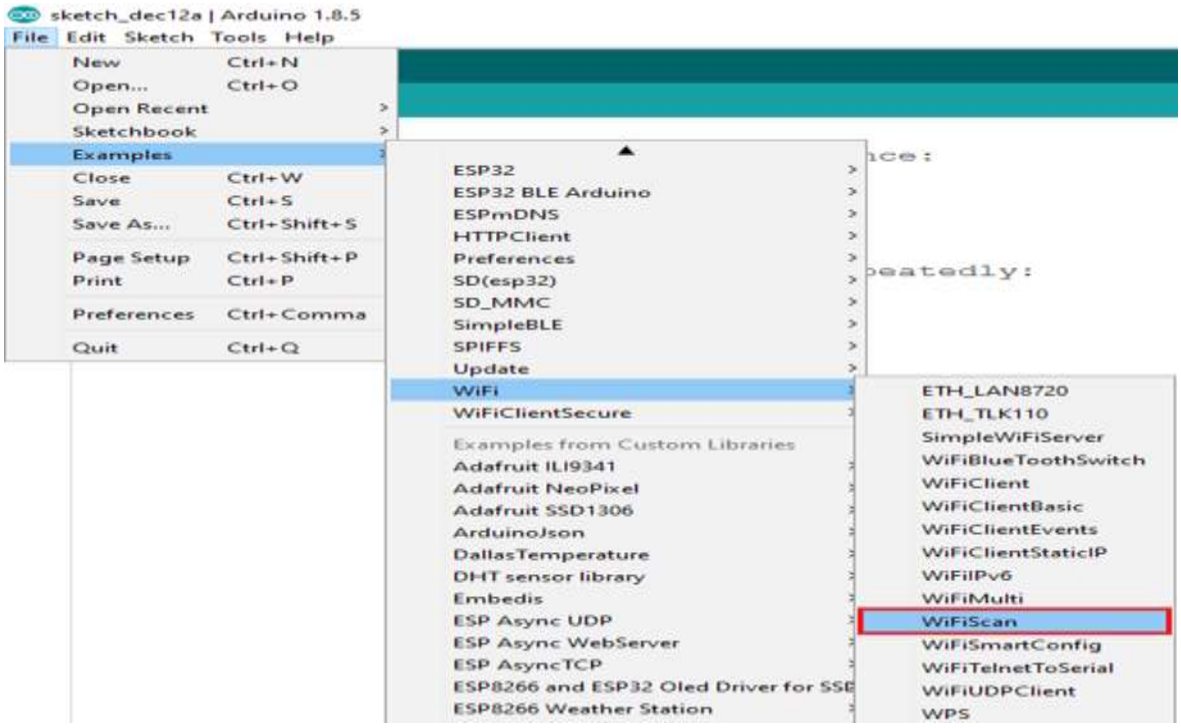
Plug the ESP32 board to your computer. With your Arduino IDE open, follow these steps:
1. Select your Board in **Tools** > **Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)



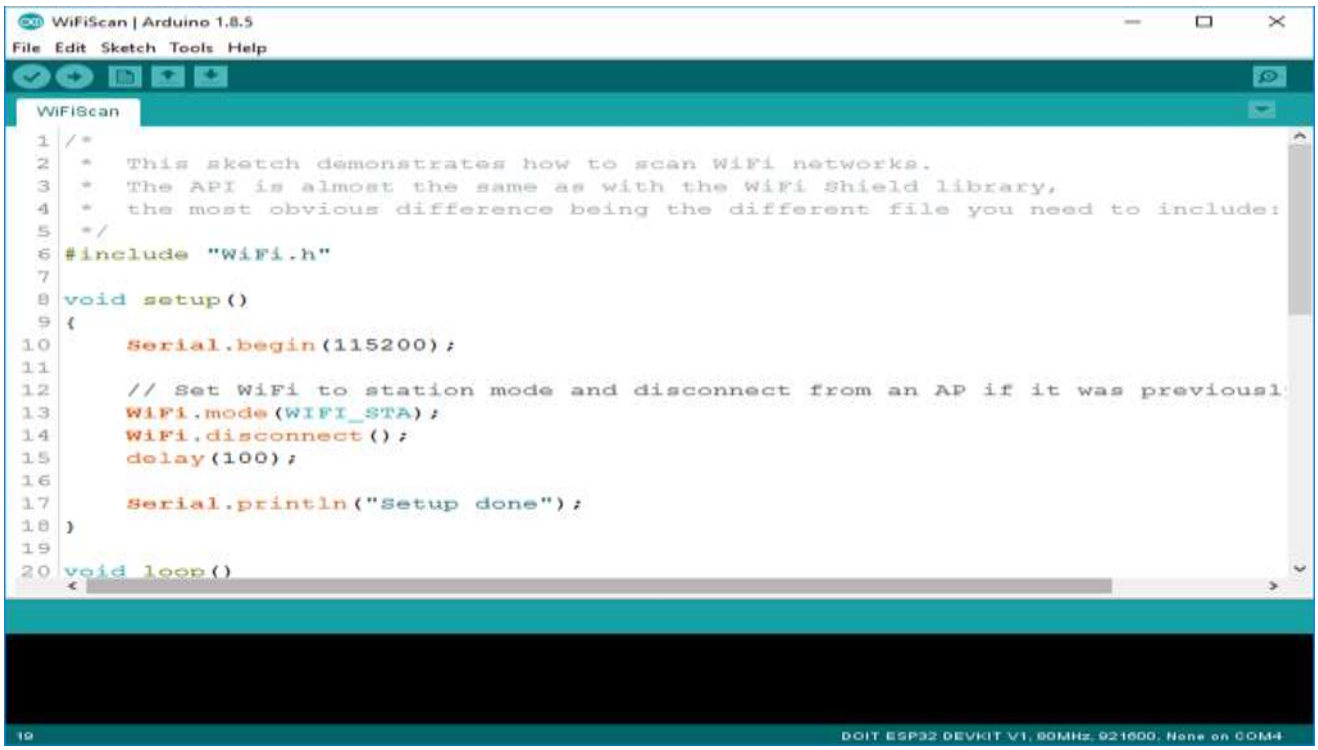
2. Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the CP210x USB to UART Bridge VCP Drivers):



3. Open the following example under File > Examples > WiFi (ESP32 > WiFiScan

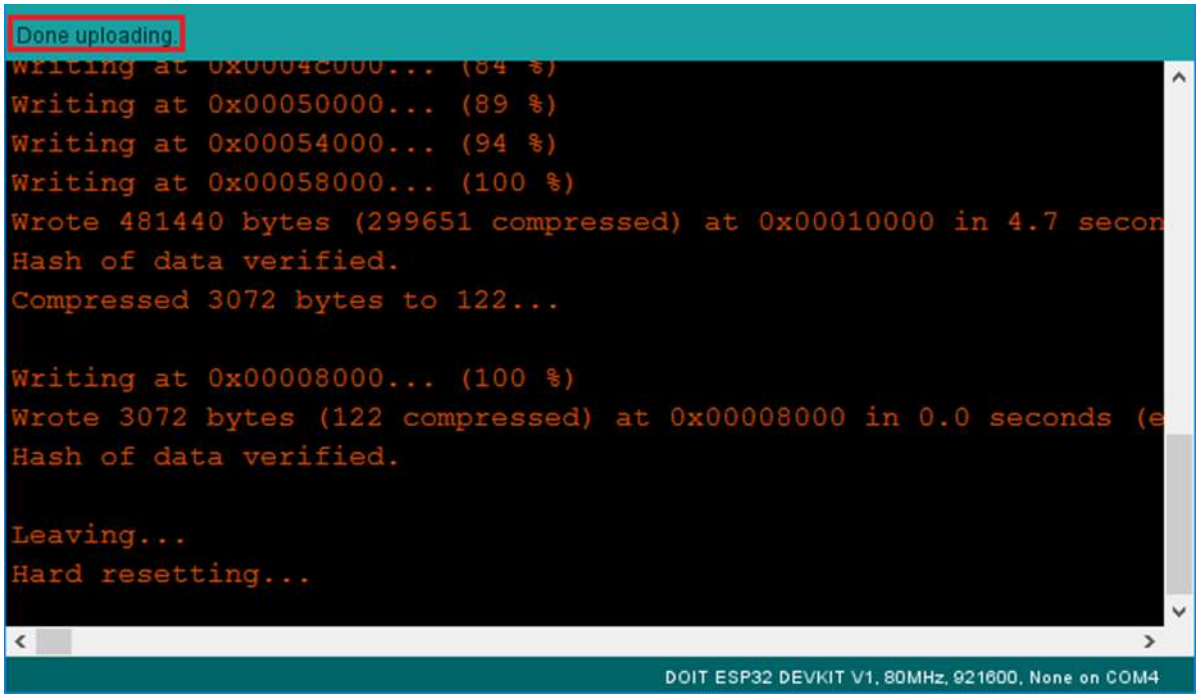


4. A new sketch opens in your Arduino IDE:



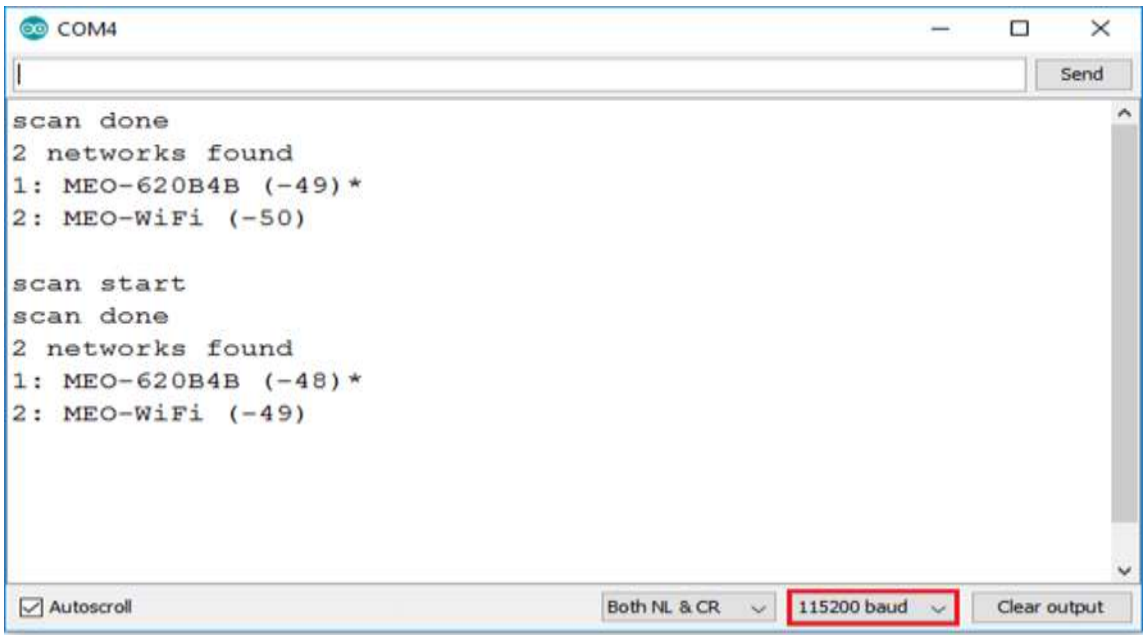
5. Press the Upload button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

6. If everything went as expected, you should see a “Done uploading.” message.



7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:

8. Press the ESP32 on-board Enable button and you should see the networks available near your ESP32:

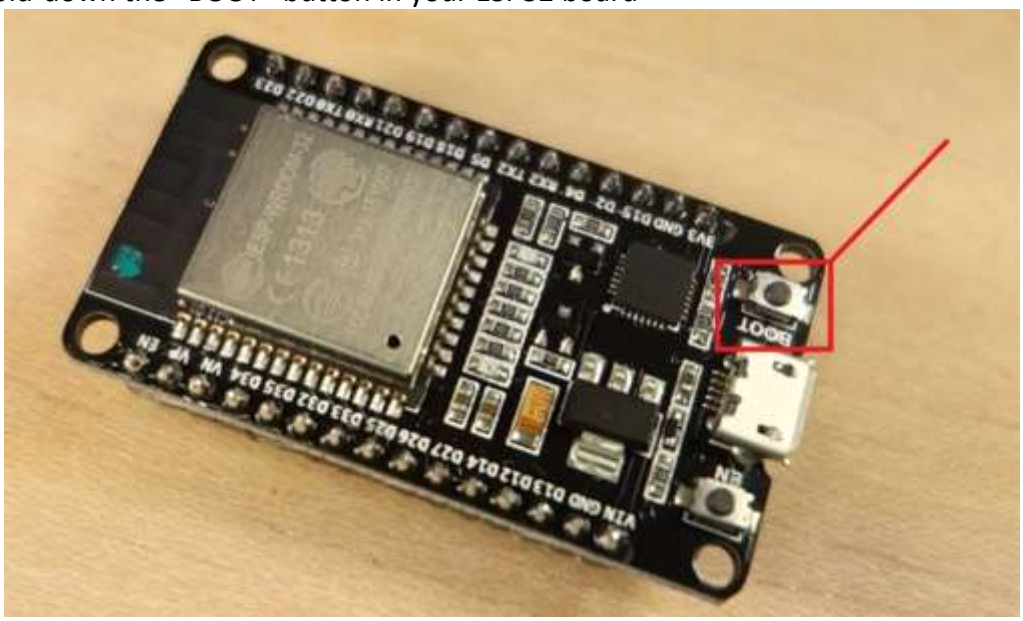


Troubleshooting

If you try to upload a new sketch to your ESP32 and you get this error message “A fatal error occurred: Failed to connect to ESP32: Timed out… Connecting…“. It means that your ESP32 is not in flashing/uploading mode.

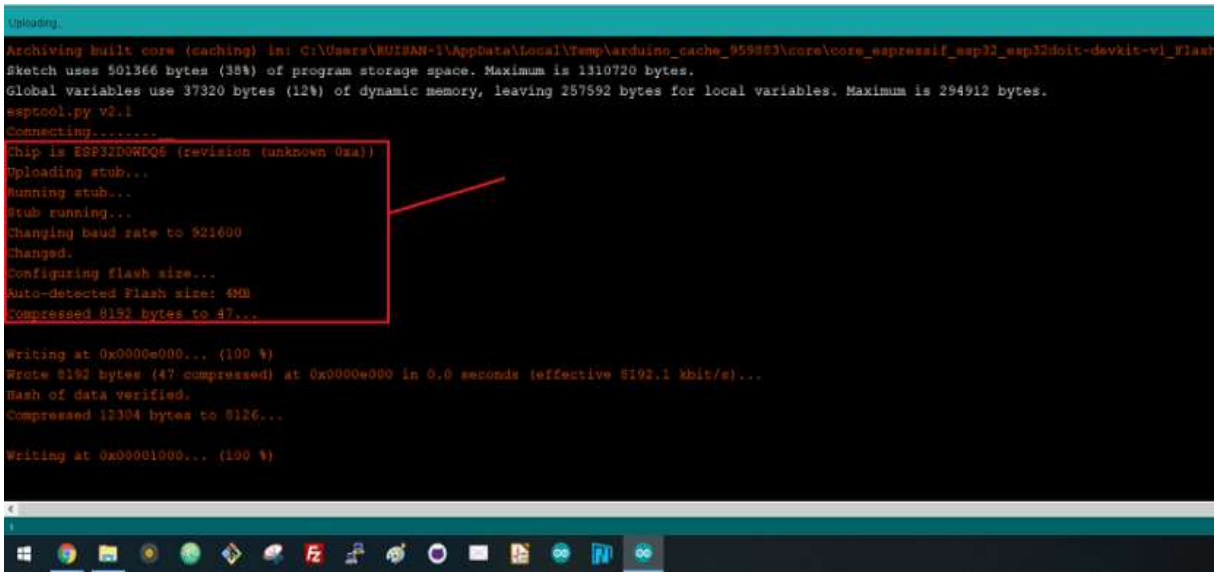
Having the right board name and COM por selected, follow these steps:

•Hold-down the “BOOT” button in your ESP32 board



•Press the “Upload” button in the Arduino IDE to upload your sketch

•After you see the “Connecting….” message in your Arduino IDE, release the finger from the “BOOT” button:



•After that, you should see the “Done uploading” message

That’s it. Your ESP32 should have the new sketch running. Press the “ENABLE” button to restart the ESP32 and run the new uploaded sketch.

Conclusion:

1. Developing and IOT system requires the use of microcontrollers. A large number of microcontrollers are available in the market, each with special features, pros and cons.
2. Arduino UNO, ESP32, STM32, AT89C51, RaspberryPi, ESP8266 are some examples of microcontroller boards used to develop IOT systems.
3. Out of these, we will use ESP32 for our practicals.
4. Programming these devices needs the use of an integrated development environment or coding platform. We also need to debug and compile the code on a computer before downloading it onto the board.
5. Arduino IDE and Visual Studio Code are examples of programming environments, of which we will use Arduino IDE as it is the simplest to use.

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.

Internet of Things Lab

EXPERIMENT 3

Title: To measure sensor data and display on serial monitor.

Objective: Write a program to measure sensor data and display on serial monitor.

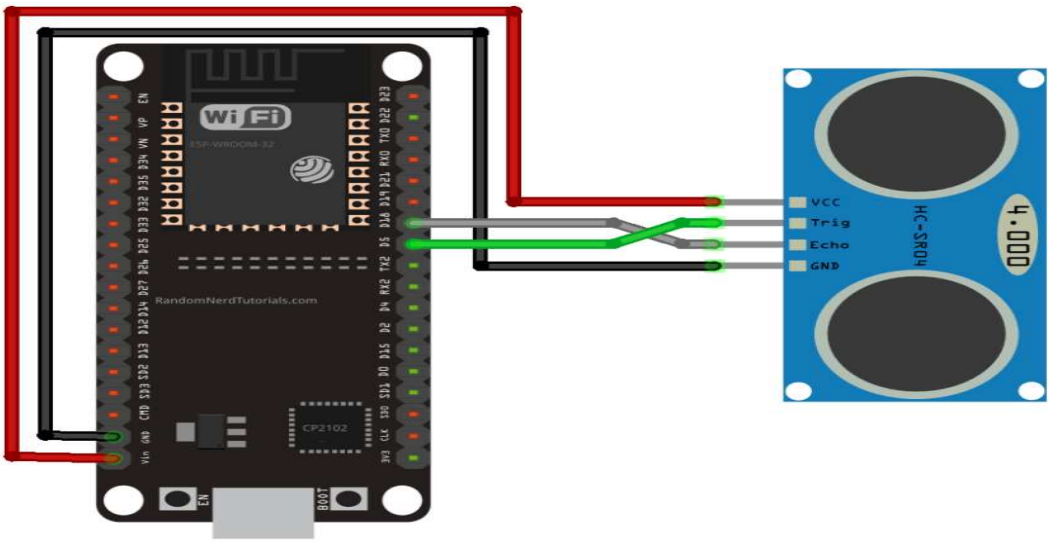
A. Interfacing of Ultrasonic Sensor

B. Interfacing of PIR Sensor

Software used: Ardino IDE

Theory:

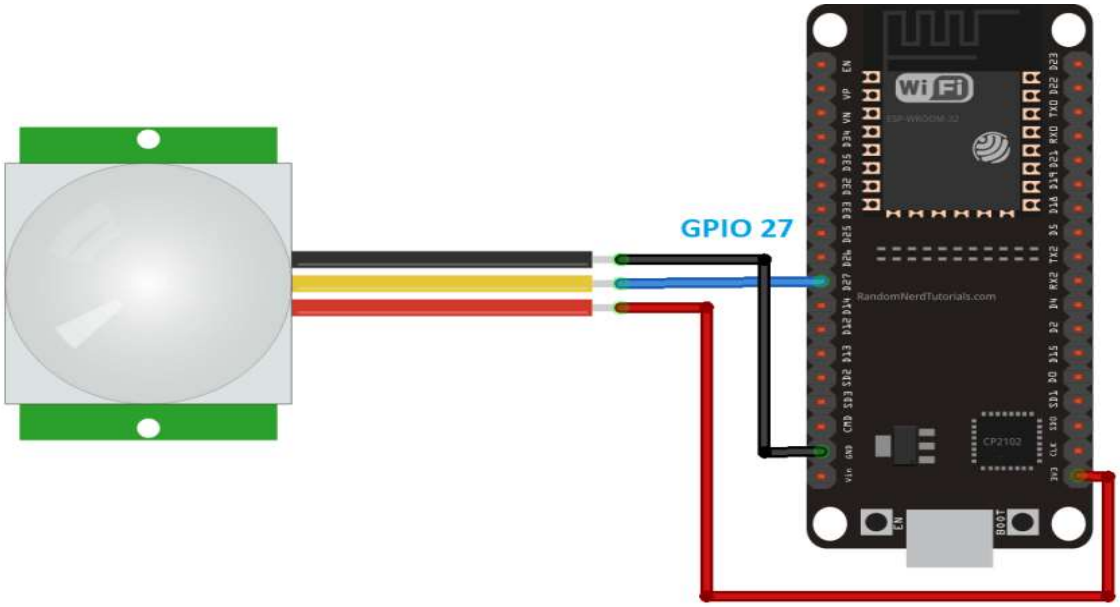
A. Interfacing of Ultrasonic Sensor with ESP32



Ultrasonic Sensor	ESP32
VCC	VIN
Trig	GPIO 5
Echo	GPIO 18
GND	GND

The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

B. Interfacing of PIR Sensor with ESP32



We are implementing hardware interrupt (Hardware interrupts are the external interrupts that are caused by an external event) method. Where we are using a PIR sensor to generate interrupts. Interrupt generated via PIR motion sensor comes under the category of a hardware interrupt. PIR motion sensor is mostly used in home automation applications where we can make home appliances to respond automatically over human presence. Appliance connected to ESP32 will respond automatically (as per the instructions provided) whenever an interrupt is being triggered by the PIR motion sensor.

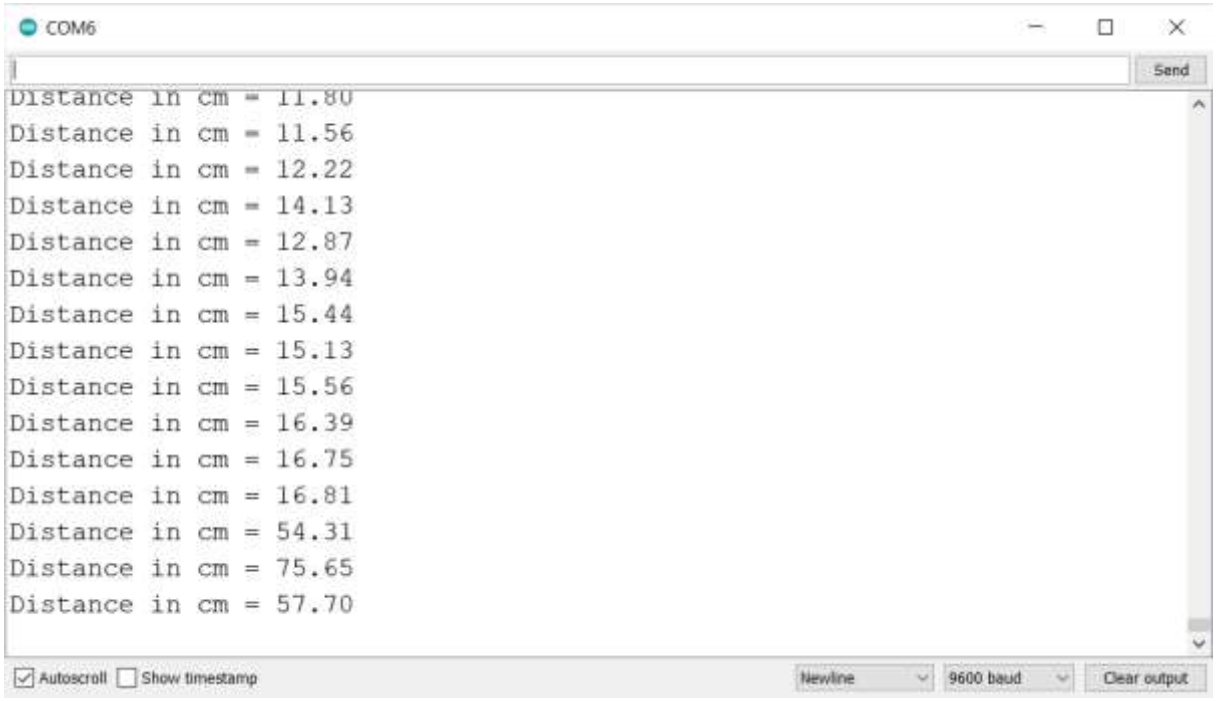
Algorithm / code :

Code: 3A Distance measurement using ultrasonic sensor

```
const int triggerPin = 5;
 const int echoPin = 18;
```

```
  #define velocity 0.034
  long duration;
  float distance;

void setup() {
  // put your setup code here, to run once:
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

  //sending the trigger pulse
  digitalWrite(triggerPin,LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin,LOW);

  //receiving echo pulse
  duration = pulseIn(echoPin, HIGH);

  //calculating distance
  distance = (duration*velocity)/2;
  //displaying on serial monitor
  Serial.print("Distance in cm = ");
  Serial.println(distance);

  //making it easy to read on serial monitor
  delay(1000);
}
```

Output on serial monitor

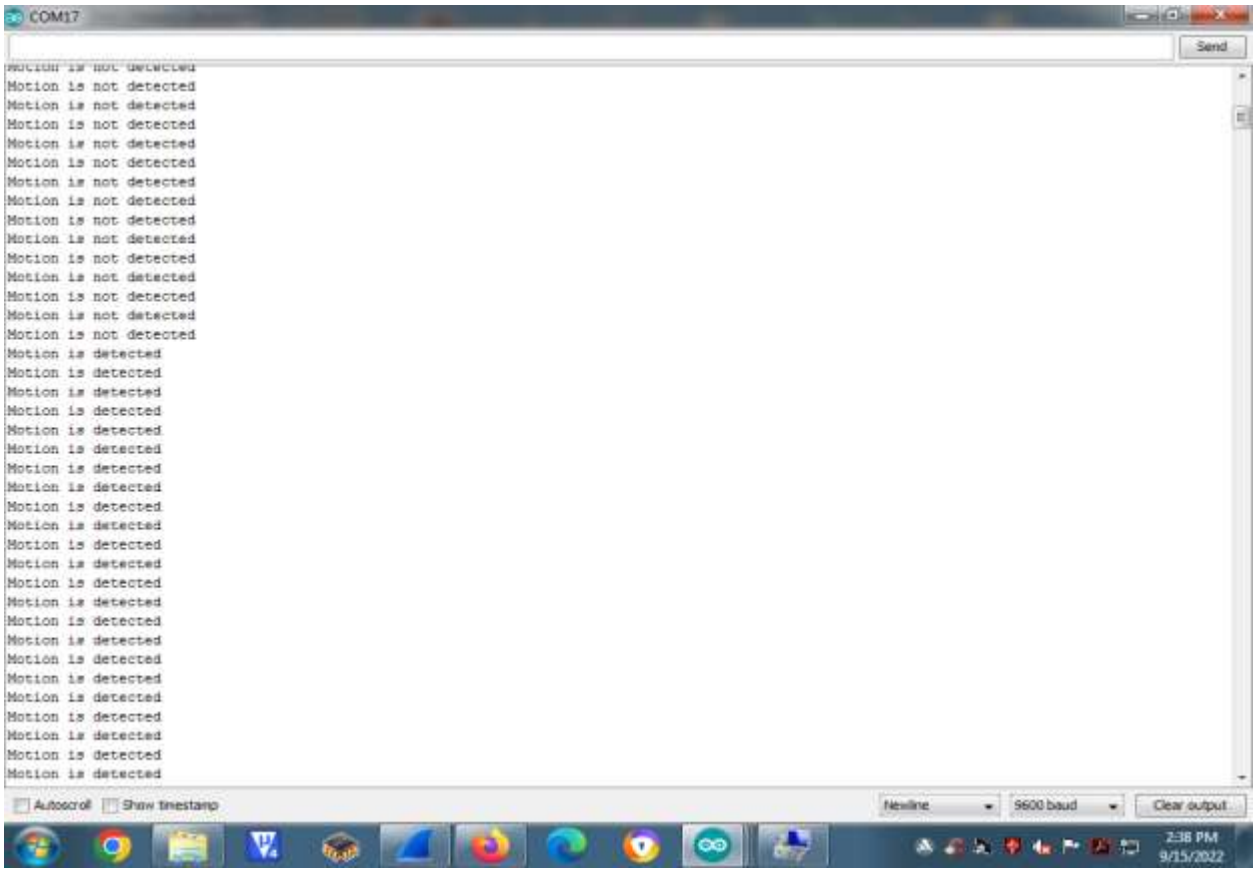


Code 3b:PIR MOTION SENSOR

```
int inputPin = 14;

int pirState = LOW;

int Value = 0;

void setup() {

  // put your setup code here, to run once:

  pinMode(inputPin,INPUT);

  Serial.begin(9600);

}


void loop() {

  // put your main code here, to run repeatedly:

 bool Value = digitalRead(inputPin);

 delayMicroseconds(2);

 if (Value == LOW){

  Serial.println("Motion is not detected");

  }

  else

  {

   Serial.println("Motion is detected");

            delay(200);
```

```
        }

}
```

Output on serial Monitor



Conclusion:

1. Ultrasonic sensor works by sending out a burst of ultrasound and listening for its echo. The microcontroller sends a pulse to trigger the sensor, and then waits for a pulse on the same pin using the pulseIn() function.
2. Uses of ultrasonic sensor are: distance measurement, fluid level measurement, collision prevention, etc.
3. PIR motion sensor provides a high output whenever motion is detected.
4. It can be used in automatic doors, burglar alarms, automatic lighting, etc.

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.
Internet of Things Lab

EXPERIMENT 4

Title: Write a program to control actuator.

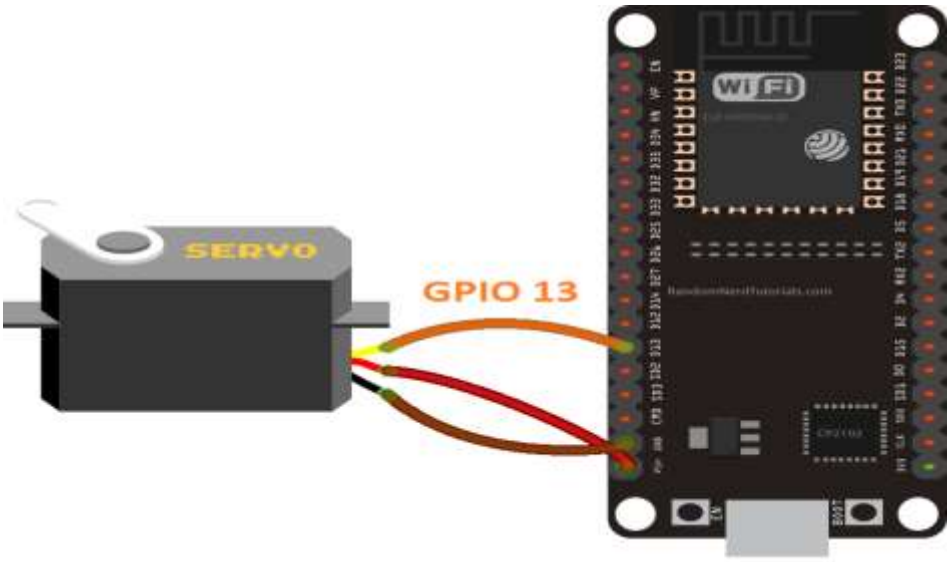
Objective: To Write a program to control actuator.

Software used: Ardino IDE

Theory:

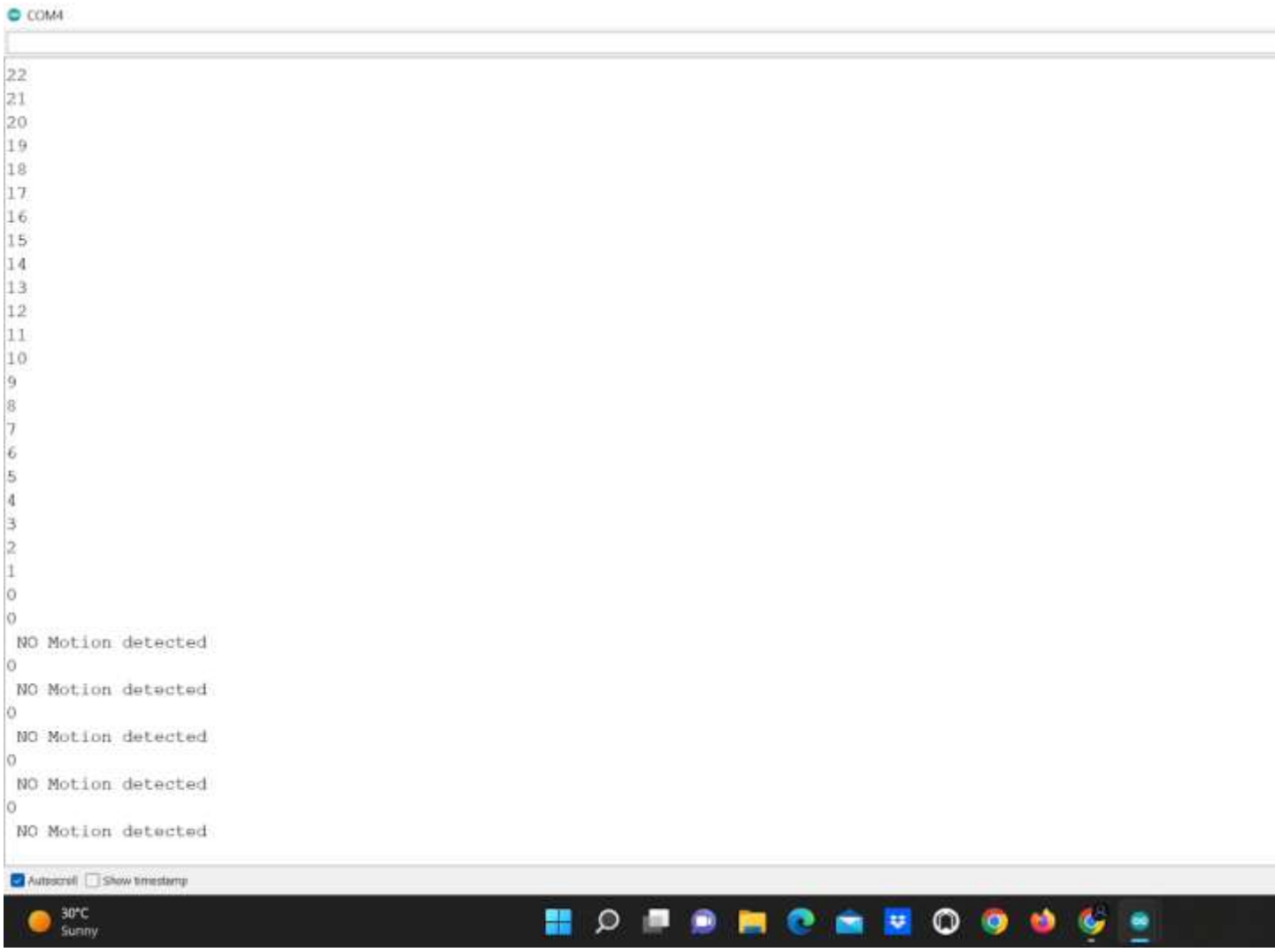
- Servo motors have three wires: power, ground, and signal. The power is usually red, the GND is black or brown, and the signal wire is usually yellow, orange, or white.
- You can position the servo's shaft in various angles from 0 to 180º. Servos are controlled using a pulse width modulation (PWM) signal. This means that the PWM signal sent to the motor will determine the shaft's position.
- To control the motor you can simply use the PWM capabilities of the ESP32 by sending a 50Hz signal with the appropriate pulse width. Or you can use a library to make this task much simpler.
- Servo motors are geared DC motors that have an integrated servomechanism with a feedback loop to allow precise positioning of the motor shaft. A high gear ratio allows a small servo to have an impressive torque rating.
- Most servos are limited in rotation to either 180 or 270 degrees, with 180-degree servo motors being more common. There are specially modified servo motors that can rotate beyond 360-degrees, but we won't be working with those today.
- Servo motors come in a wide range of sizes and can be controlled either with an analog PWM signal or with a digital I/O signal. The inexpensive servos we use for hobbyist applications are usually analog servo motors, which are the types we will be using today.

Interfacing diagram



Algorithm / code :

```
//servo motor interfacing

#include <ESP32Servo.h>
static const int ServoPin = 12;
Servo Servo1;

void setup() {
 Serial.begin(9600);
 Servo1.attach(ServoPin);
}

void loop() {
  for(int PosDegrees=0;PosDegrees<=90;PosDegrees++){
   Servo1.write(PosDegrees);
   Serial.println(PosDegrees);
   //delay(200);
  }
   for(int PosDegrees=90;PosDegrees>=0;PosDegrees--){
   Servo1.write(PosDegrees);
   Serial.println(PosDegrees);
   //delay(200);
  }
  }
```



Conclusion:

1. Actuators form the last part of an IOT system.
They are used to control various parameters in the real world.
2. One of the actuators is a servo motor, which consists of a dc motor with a servo mechanism and gear arrangement.
3. It can rotate precisely to the specified number of degrees.
4. Servo motors are used in robotics, cameras, antenna positioning, sun tracking solar array, automatic doors, remote controlled toys, etc.

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.
Internet of Things Lab

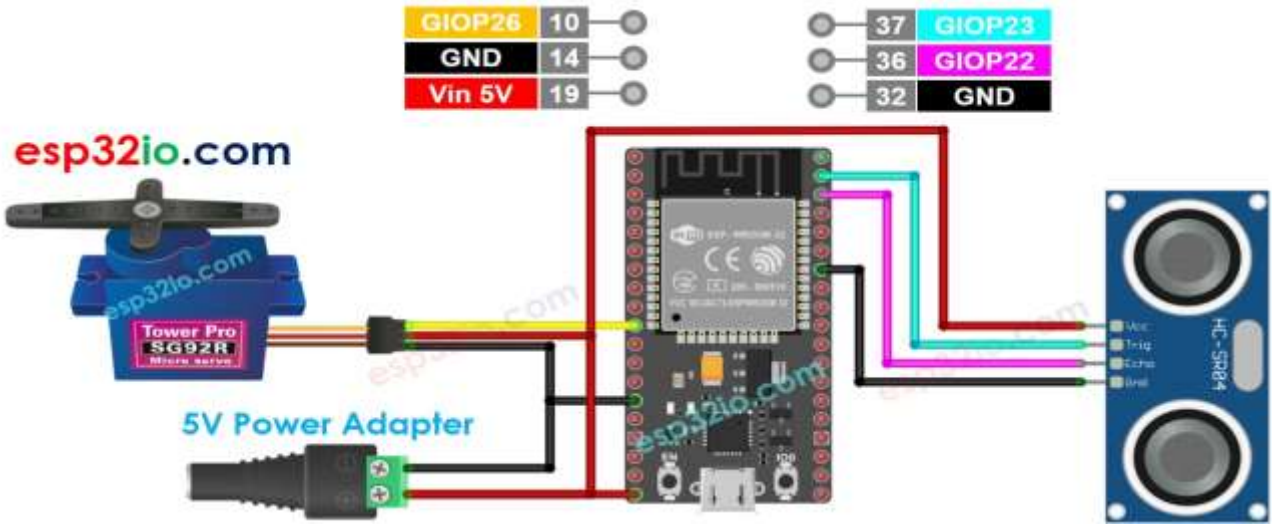
EXPERIMENT 5

Title: Study to control actuators based on real time sensor data

Objective: Write a program to control Actuators based on real time sensor data.

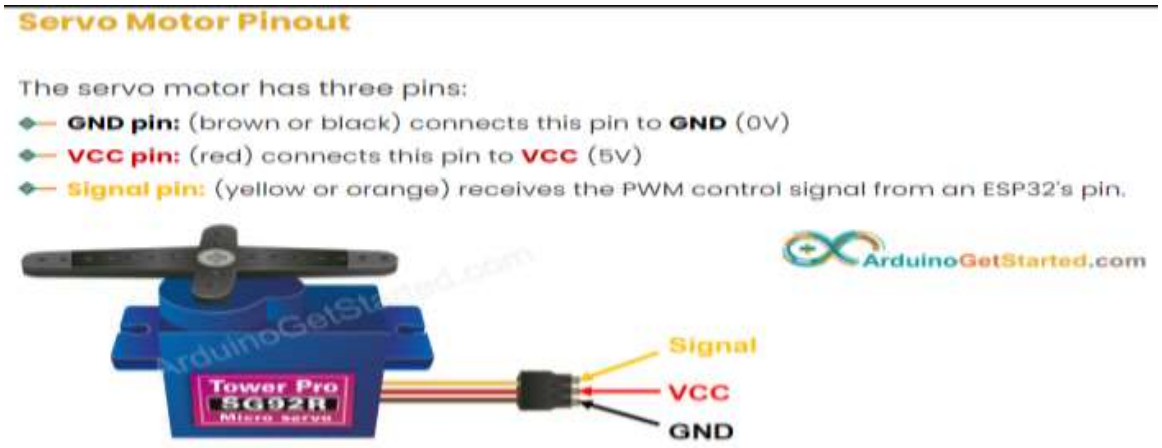
Software used: Ardino IDE

Theory:



- The esp32 automatically rotates a servo motor to 90 degrees if an object is close to ultrasonic sensor.
- The esp32 automatically rotates the servo motor to 0 degrees if an object tis far from the

ultrasonic sensor



The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

- Servo motors are geared DC motors that have an integrated servomechanism with a feedback loop to allow precise positioning of the motor shaft. A high gear ratio allows a small servo to have an impressive torque rating.
- Most servos are limited in rotation to either 180 or 270 degrees, with 180-degree servo motors being more common. There are specially modified servo motors that can rotate beyond 360-degrees, but we won't be working with those today.

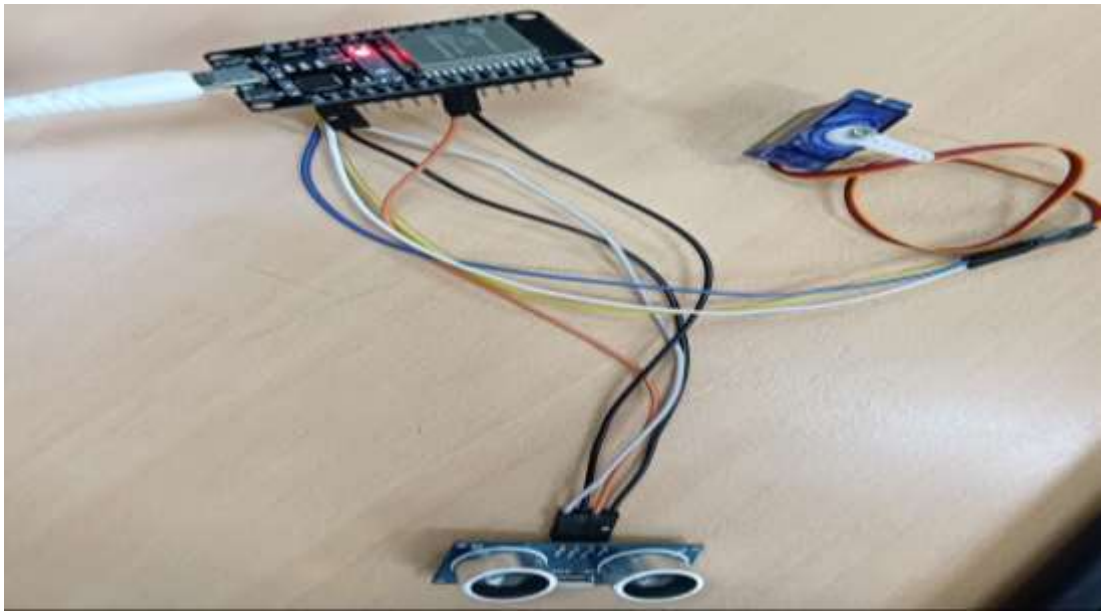
Servo motors come in a wide range of sizes and can be controlled either with an analog PWM signal or with a digital I/O signal. The inexpensive servos we use for hobbyist applications are usually analog servo motors, which are the types we will be using today

Algorithm / code :

```
#include<ESP32Servo.h>
#define TRIG_PIN  5  // ESP32 pin GIOP05 connected to Ultrasonic Sensor's TRIG pin
#define ECHO_PIN  18  // ESP32 pin GIOP18 connected to Ultrasonic Sensor's ECHO pin
#define SERVO_PIN 13  // ESP32 pin GIOP13 connected to Servo Motor's pin
#define Distance  50 // centimeters
 Servo servo;
 float duration_us, distance_cm;
void setup() {
  Serial.begin (9600);       // initialize serial port
  pinMode(TRIG_PIN, OUTPUT); // set ESP32 pin to output mode
  pinMode(ECHO_PIN, INPUT);  // set ESP32 pin to input mode
  servo.attach(SERVO_PIN);   // attaches the servo on pin 13 to the servo object
  servo.write(0);
}
void loop() {
// generate 10-microsecond pulse to TRIG pin
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
// measure duration of pulse from ECHO pin
  duration_us = pulseIn(ECHO_PIN, HIGH);
  // calculate the distance
  distance_cm = 0.017 * duration_us;
 if (distance_cm < Distance)
    servo.write(90); // rotate servo motor to 90 degree
  else
    servo.write(0);  // rotate servo motor to 0 degree
  // print the value to Serial Monitor
  Serial.print("distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");
delay(500);
}
```

COM4

```
distance:  1200.66  cm
distance:  1200.54  cm
distance:  1200.69  cm
distance:  1200.66  cm
distance:  1200.57  cm
distance:  1200.54  cm
distance:  1200.66  cm
distance:  1200.63  cm
distance:  1200.63  cm
distance:  480.08  cm
distance:  4.13  cm
distance:  1200.52  cm
distance:  62.10  cm
distance:  60.86  cm
distance:  245.14  cm
distance:  245.11  cm
distance:  245.05  cm
distance:  244.94  cm
distance:  244.60  cm
distance:  244.37  cm
```



Conclusion:

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.
Internet of Things Lab

EXPERIMENT 6

Title: Study of Implementation of a standalone web server using ESP32

Objective: Write a program to implement a standalone web server using of ESP32

Software used:

Theory:

Algorithm / code :

Conclusion:

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech.
Internet of Things Lab

EXPERIMENT 7

Title: Study of web based application through ESP32

Objective: To develop a web application through ESP32

Software used:

Theory:

Algorithm / code :

Conclusion:

Maharshi Karve Stree Shikshan Samstha's
Cummins College of Engineering for Women
Karve Nagar, Pune
(An autonomous Institute affiliated to Savitribai Phule Pune University)

"शीलं परं भूषणम्"

Department of Electronics and Telecommunication

A. Y. 2022-23 **SEM - I**

T. Y. B. Tech. **DIV.:- A, B, C**

Internet of Things Lab

EXPERIMENT 8

Title: Mini Project: Develop an IoT system for given application

Objective: Mini Project: Develop an IoT system for given application

Software used:

Theory:

Algorithm / code :

Conclusion: