**LAPORAN PRAKTIKUM INTERNET OF THINGS**

**PRAKTIK PEMBUATAN SENSOR JARA004B**

Fakultas Vokasi, Universitas Brawijaya

**Autors**

Fadhilah Azkia

Fakultas Vokasi, Universitas Brawijaya

Email : [fadhilahazkia95@gmail.com](mailto:fadhilahazkia95@gmail.com)

**Abstract**

Percoobaan ini bertujuan untuk merancang Sensor Jarak ( Ultra Sonic ) menggunakan mikrokontroler ESP32. Sistem ini terdiri dari sensor HRC-SR04 dan ESP32 itu sendiri. Mikrokontroler diprogram untuk mengatur jarak dari setiap benda yang ada didekat sensor. Praktik ini sangat efektif untuk mengetahui bagaimana sensor bekerja dengan terstruktur.

1. **Introduction**  
   1. **Latar Belakang**  
        
      Lampu lalu lintas merupakan sistem yang sangat berguna untuk mengatur aliran kendaraan di persimpangan jalan. Sistem ini dirancang dengan baik dan efektif sesuai dengan waktu yang mempengaruhi kapan lampu menyala, dari lampu satu ke lampu lainnya. Eksperimen ini mensimulasikan bagaimana sistematis dari lampu lalulintas menggunakan mikrokontroler ESP32.
   2. **Tujuan Eksperimen**

* Memahami cara kerja sensor ultrasonik **HC-SR04** dalam mengukur jarak.
* Menerapkan pemrograman mikrokontroler **ESP32** untuk membaca data dari sensor jarak.
* Menampilkan hasil pengukuran jarak dalam **cm** dan **inch** pada Serial Monitor.

1. **Metodologi**
   1. **Alat dan Bahan**

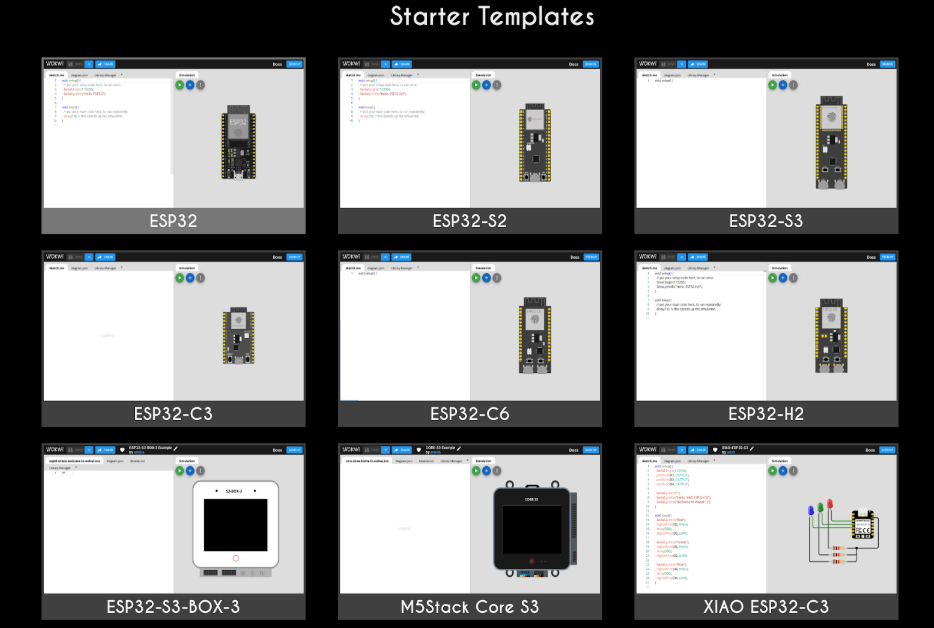
* Hardware : Laptop
* Wokwi online : <https://wokwi.com/>
* **HRC-SR04**
* Visual Studio Code
  1. **Langkah Implementasi**

1. Membuka wokwi online.
2. Pilih simulasi mengggunakan ‘ESP32’



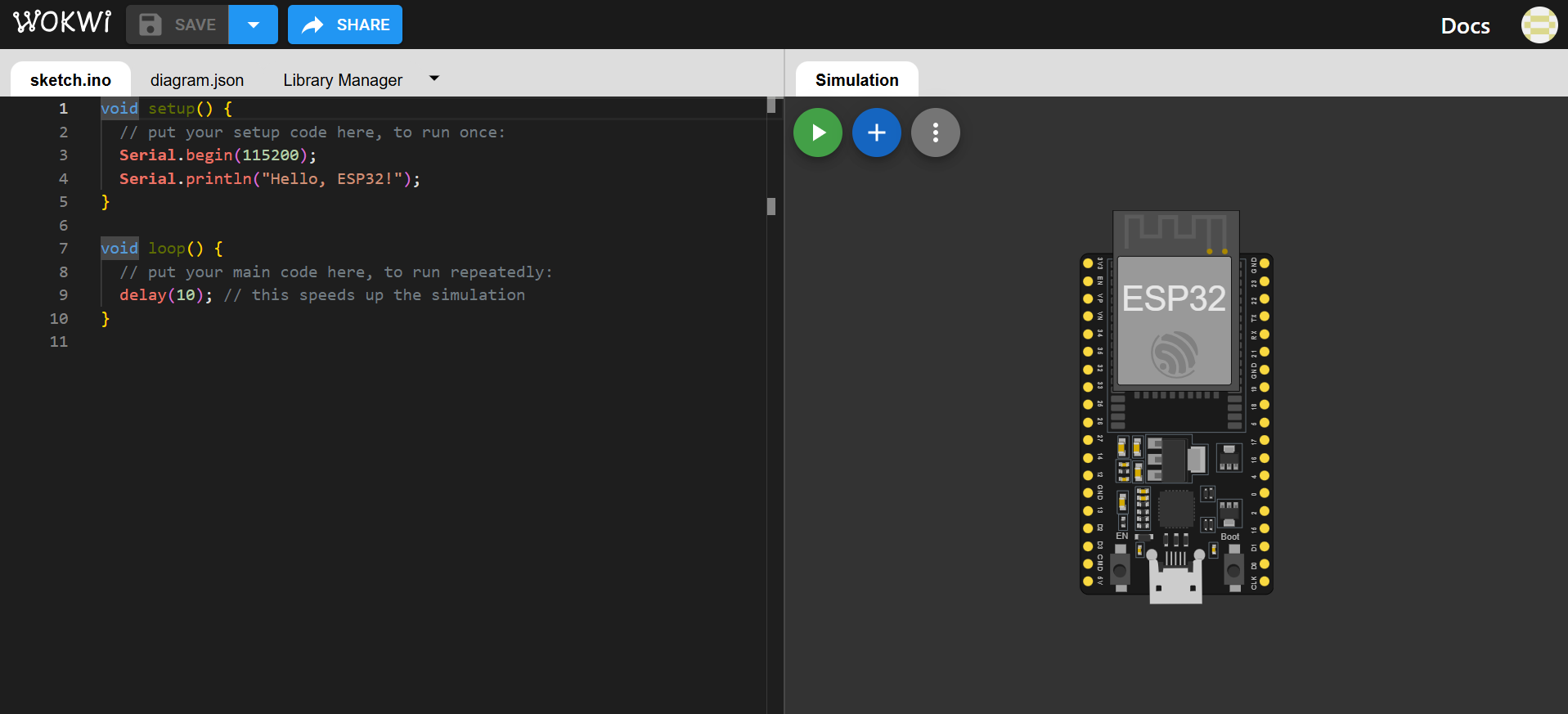
**Gambar 2.2.1** Home page Wokwi online ( wokwi.com ).

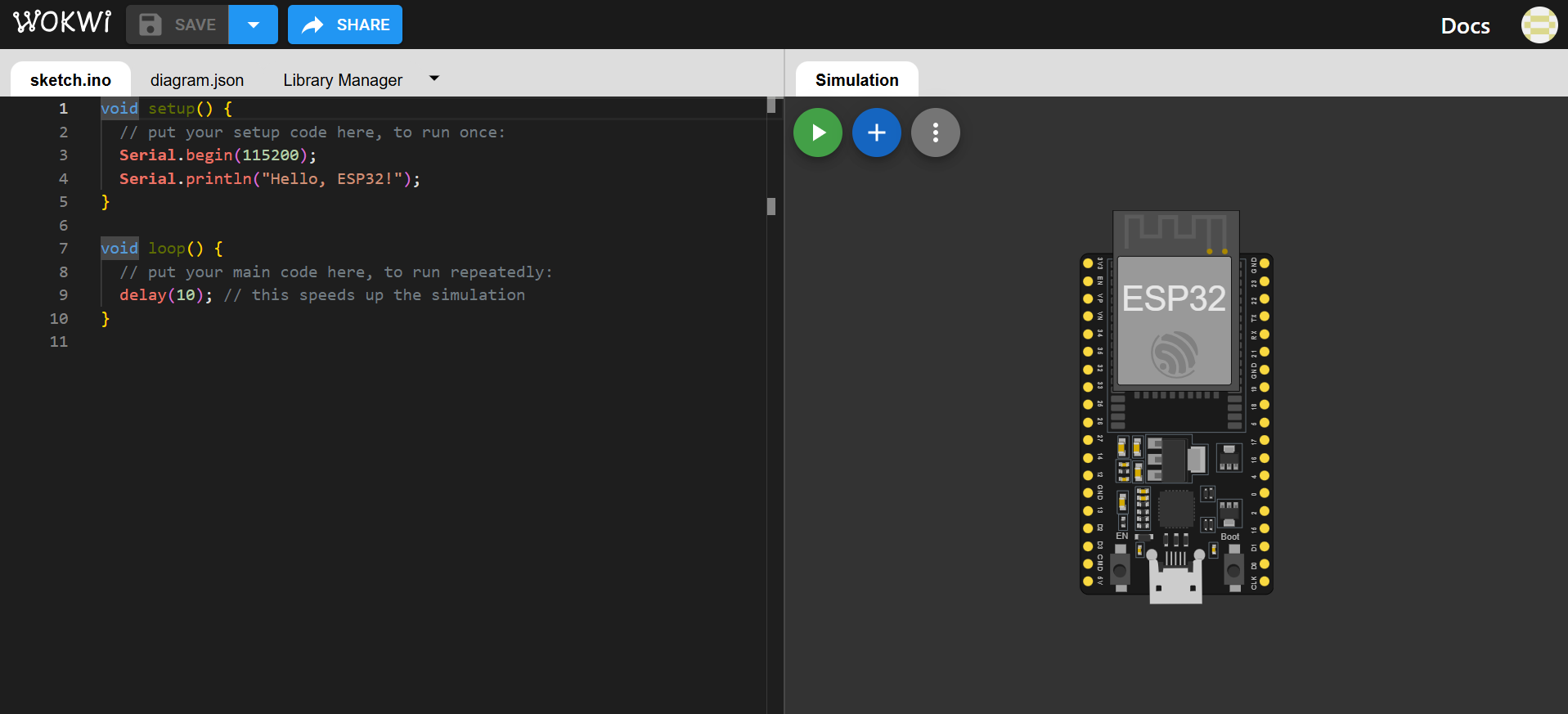
1. Selanjutnya memilih starter templates ‘ESP32’



**Gambar 2.2.2** Halaman starter templates.

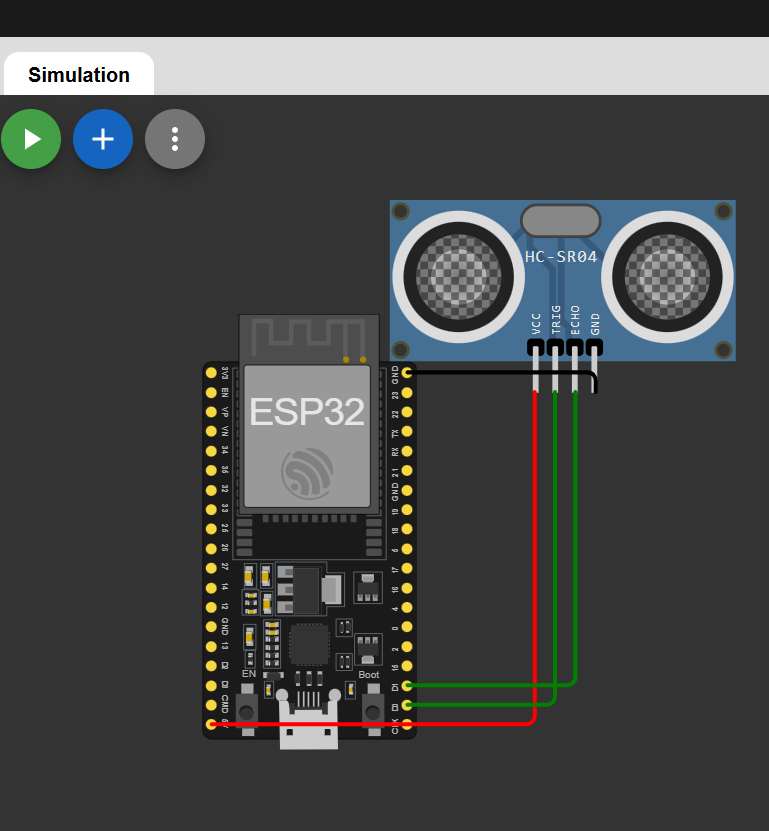
1. Tampilannya akan seperti yang ada di bawah ini.





**Gambar 2.2.3** Halaman project wokwi online.

1. Menambahkan Sensor HRC-SR04 yang kita butuhkan pada halaman.



**Gambar 2.2.4** Gambar device setelah di pasangkan.

1. Memasukkan source code sesuai dengan yang dibutukan pada main.cpp.

const int trigPin = 5;

const int echoPin = 18;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;

void setup() {

**Serial**.begin(115200); // Starts the serial communication

 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

 pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop() {

 // Clears the trigPin

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Sets the trigPin on HIGH state for 10 micro seconds

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

 duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

 distanceCm = duration \* SOUND\_SPEED/2;

  // Convert to inches

 distanceInch = distanceCm \* CM\_TO\_INCH;

  // Prints the distance in the Serial Monitor

**Serial**.print("Distance (cm): ");

**Serial**.println(distanceCm);

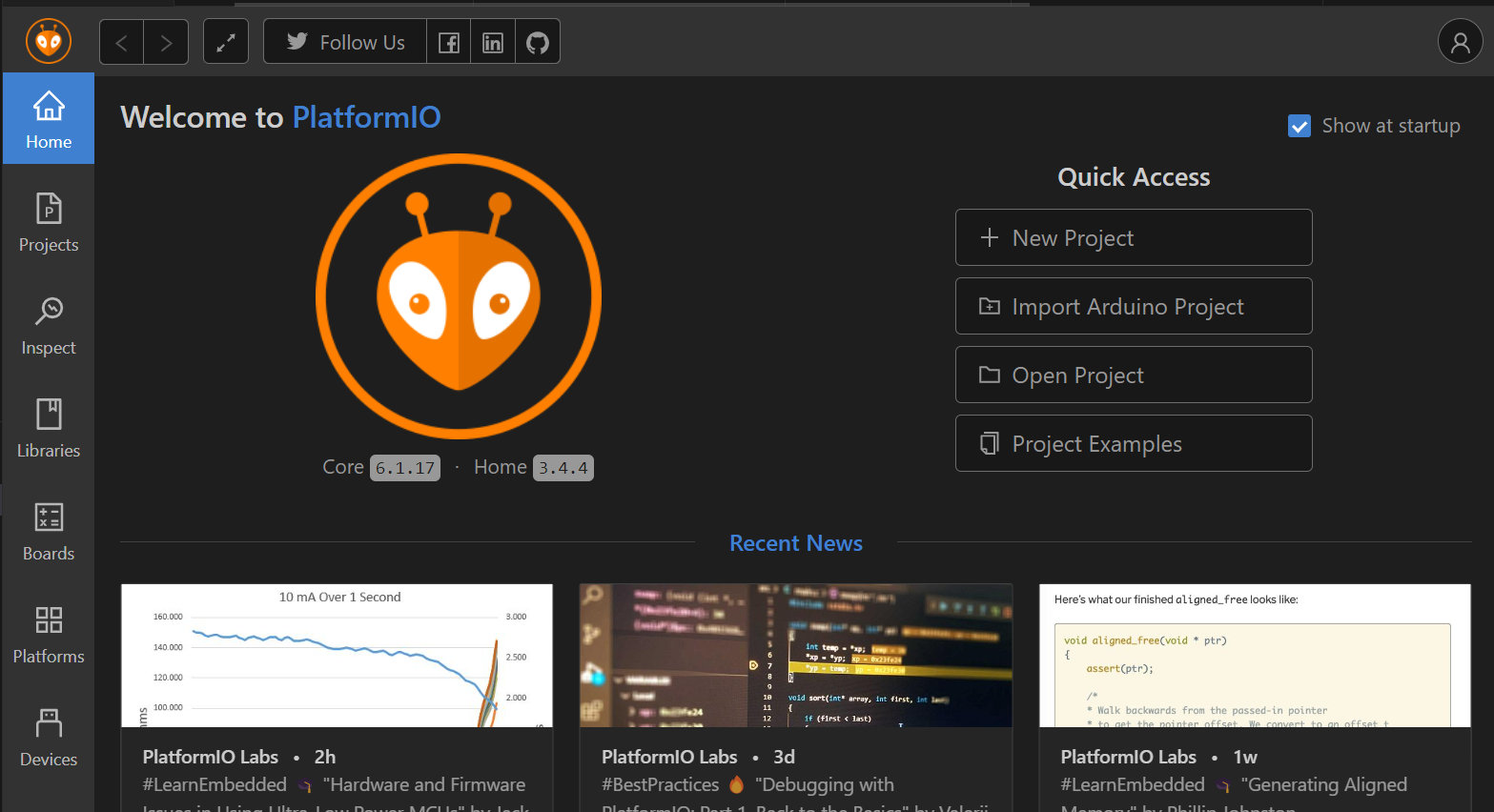
 // Serial.print("Distance (inch): ");

 // Serial.println(distanceInch);

  delay(1000);

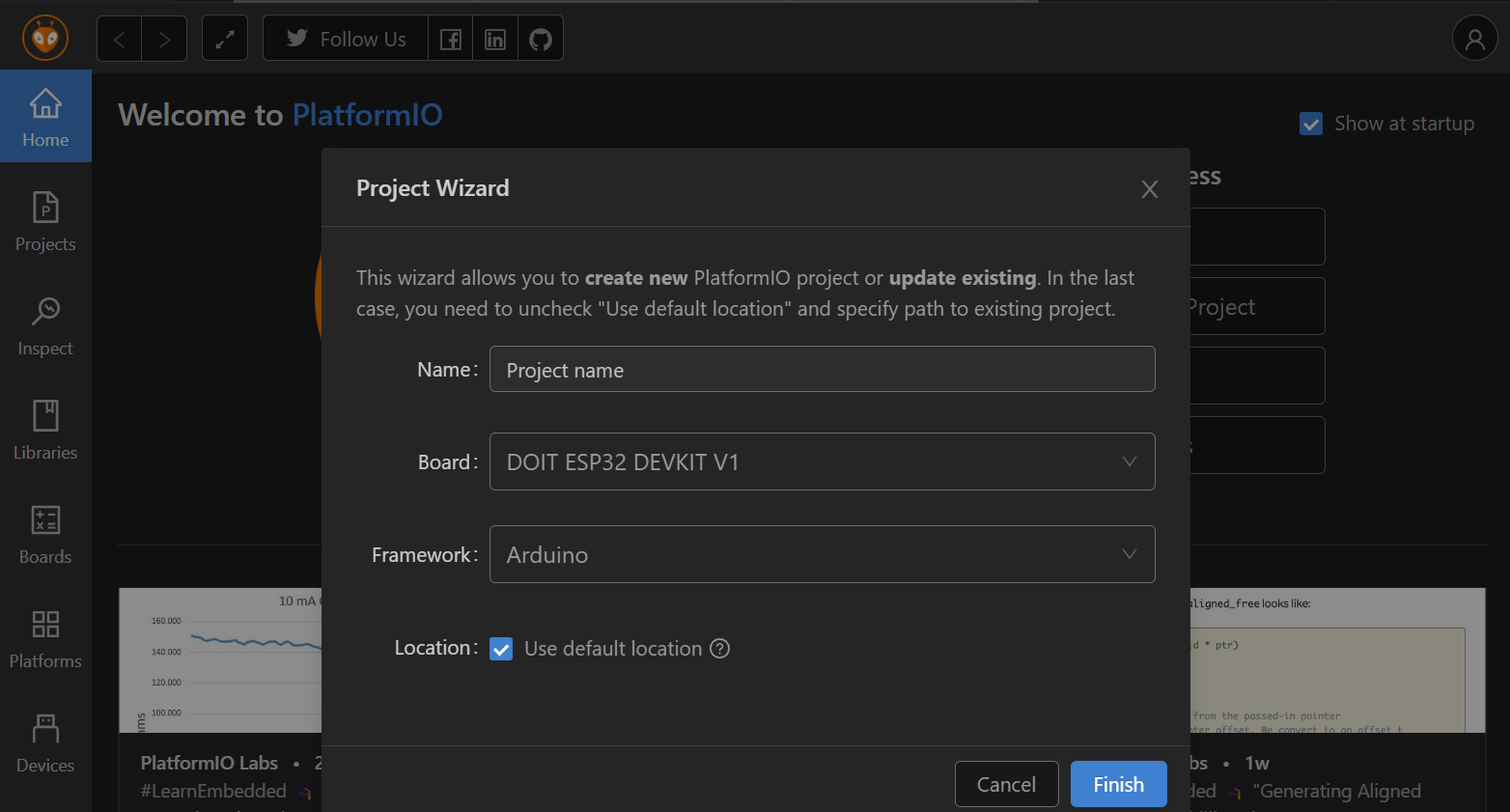
}

1. Selanjutnya masuk ke aplikasi Visual Studio Code, dikarenakan akun wokwi gratis memiliki traffic yang sangat besar sehingga untuk menjalankan project kita membutuhkan waktu yang sangat lama.
2. Membuat project baru di Visual Studio Code dengan extensi PlatformIO.



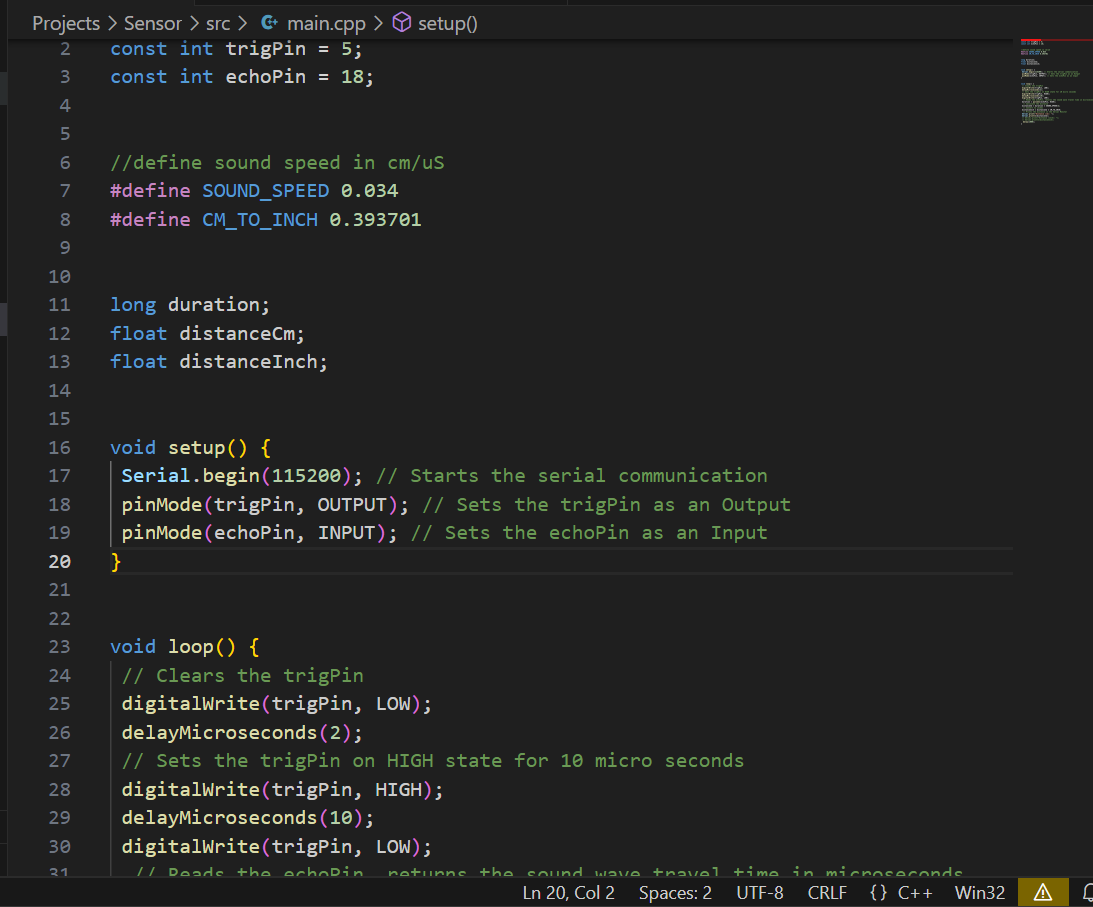
**Gambar 2.2.5** Halaman utama PlatformIO.

1. Memasukkan nama project, memilih board ‘DOIT ESP32 DEVKIT V1’ lalu finish.

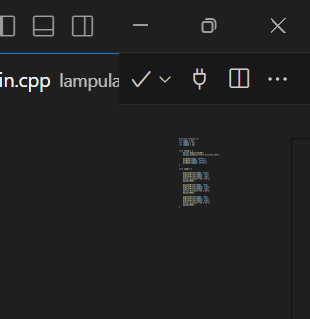


**Gambar 2.2.6** Opsi memilih board dan framework.

1. Pada folder yang sudah dibuat, cari folder src->file main.cpp kemudian salin source code yang tadi sudah dibuat pada wokwi online.



**Gambar 2.2.7** Main.cpp



Vscode akan melakukan proses kompilasi seperti tampilan berikut.tunggu hingga succes.

1. Setelah proses compiling berhasil. anda mendaparkan 2 file penting yang akan dipakai pada proses simulasi. Perhatikan ada file **firmware.bin** dan **firmware.elf**

File ini akan muncul setelah dibuild.

1. Buatlah file wokwi.toml

Isikan dengan source code ini.

[wokwi]

version = 1

firmware = 'path-to-your-firmware.hex'

elf = 'path-to-your-firmware.elf'

1. Gantikan firmware dan elf sesuai Relative Path.
2. Membuat diagram.json kemudian salin dari wokwi online juga.

{

  "version": 1,

  "author": "Anonymous maker",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -56.1, "left": 91.9, "attrs": {} }

  ],

  "connections": [

    [ "esp:TX", "$serialMonitor:RX", "", [] ],

    [ "esp:RX", "$serialMonitor:TX", "", [] ],

    [ "ultrasonic1:VCC", "esp:5V", "red", [ "v0" ] ],

    [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],

  [ "ultrasonic1:TRIG", "esp:D0", "green", [ "v153.6", "h-77.2" ]],

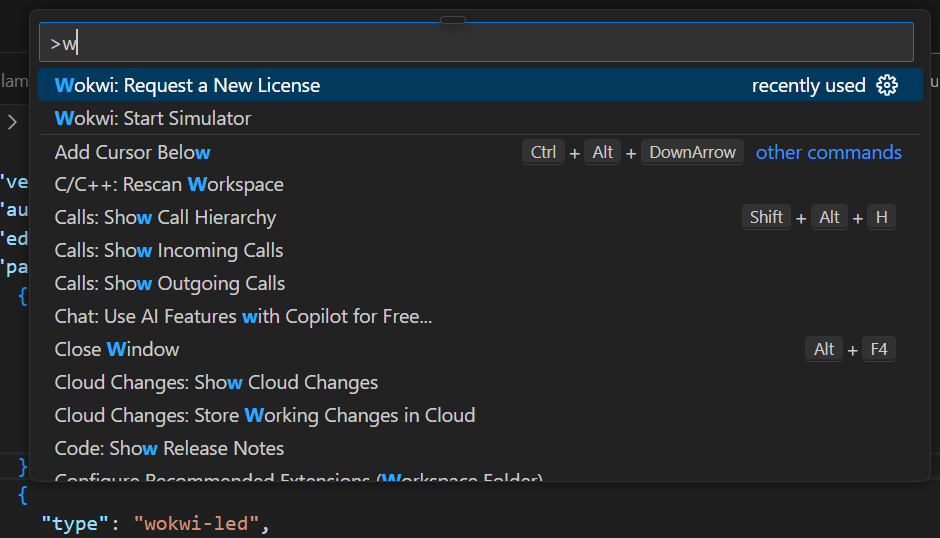
    [ "ultrasonic1:ECHO", "esp:D1", "green", [ "v0" ] ]

  ],

  "dependencies": {}

}

Sebelum memulai simulasi, pastikan telah malakukan request a new license dengan menjalankan perintah **> Wokwi: Request a New License**



**Gambar 2.2.7** Request License.

1. Jalankan simulasinya dengan memilih opsi ‘Start Simulator’

**Gambar 2.2.8** Gambar hasil simulasi.

1. **Hasil dan Pembahasan**
   1. **Hasil Eksperimen**

Dari eksperimen ini, diperoleh hasil bahwa sensor ultrasonik dapat mengukur jarak dengan cukup akurat. Nilai jarak yang diperoleh ditampilkan dalam satuan **cm** dan **inch** melalui **Serial Monitor**.

Hasil eksperimen menunjukkan bahwa:

1. Semakin jauh objek dari sensor, semakin besar nilai jarak yang terbaca.
2. Semakin dekat objek, semakin kecil nilai jaraknya.
3. Sensor HC-SR04 bekerja optimal pada jarak 2 cm hingga 400 cm.
4. Jika tidak ada objek dalam jangkauan sensor, nilai yang dihasilkan bisa tidak akurat atau tidak terdeteksi.
5. **Appendix**

* Main.cpp
* #include <Arduino.h>
* const int trigPin = 5;
* const int echoPin = 18;
* //define sound speed in cm/uS
* #define SOUND\_SPEED 0.034
* #define CM\_TO\_INCH 0.393701
* long duration;
* float distanceCm;
* float distanceInch;
* void setup() {
* Serial.begin(115200); // Starts the serial communication
* pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
* pinMode(echoPin, INPUT); // Sets the echoPin as an Input
* }
* void loop() {
* // Clears the trigPin
* digitalWrite(trigPin, LOW);
* delayMicroseconds(2);
* // Sets the trigPin on HIGH state for 10 micro seconds
* digitalWrite(trigPin, HIGH);
* delayMicroseconds(10);
* digitalWrite(trigPin, LOW);
* // Reads the echoPin, returns the sound wave travel time in microseconds
* duration = pulseIn(echoPin, HIGH);
* // Calculate the distance
* distanceCm = duration \* SOUND\_SPEED/2;
* // Convert to inches
* distanceInch = distanceCm \* CM\_TO\_INCH;
* // Prints the distance in the Serial Monitor
* Serial.print("Distance (cm): ");
* Serial.println(distanceCm);
* // Serial.print("Distance (inch): ");
* // Serial.println(distanceInch);
* delay(1000);
* }
* Wokwi.toml

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'