**LAPORAN PRAKTIUM INTERNET OF THINGS (IoT)**

Fakultas Vokasi , Universitas Brawijaya

**Praktik Simulasi Rangkaian Sensor Ultrasonik**

**Menggunakan ESP32**

*Iffah Alayya Azzahro*

*Fakultas Vokasi, Universitas Brawijaya*

*iffaazzahro@gmail.com*

**Abstrak**

Internet of Things (IoT) memungkinkan perangkat elektronik untuk berkomunikasi dan beroperasi secara otomatis, termasuk dalam sistem pengukuran berbasis mikrokontroler. Praktikum ini bertujuan untuk mengimplementasikan dan mensimulasikan rangkaian sensor ultrasonik menggunakan ESP32. Sensor ultrasonik digunakan untuk mengukur jarak berdasarkan waktu tempuh gelombang suara, yang kemudian diolah oleh ESP32 dan ditampilkan melalui Serial Monitor. Pemrograman dilakukan menggunakan Visual Studio Code (VS Code) dengan bahasa pemrograman C++. Hasil praktikum menunjukkan bahwa sistem dapat bekerja dengan baik, di mana sensor ultrasonik dapat mengukur jarak dengan respons yang cepat dan akurat. Praktikum ini membantu mahasiswa memahami prinsip kerja sensor ultrasonik, pemrograman mikrokontroler, serta penerapan IoT dalam sistem pengukuran. Kesimpulannya, praktik ini memberikan pengalaman langsung dalam merancang sistem pengukuran berbasis ESP32, yang dapat menjadi dasar bagi pengembangan sistem IoT yang lebih kompleks.*Keyword : Internet of Things (IoT), ESP32, Relay, Mikrokontroler, LED*

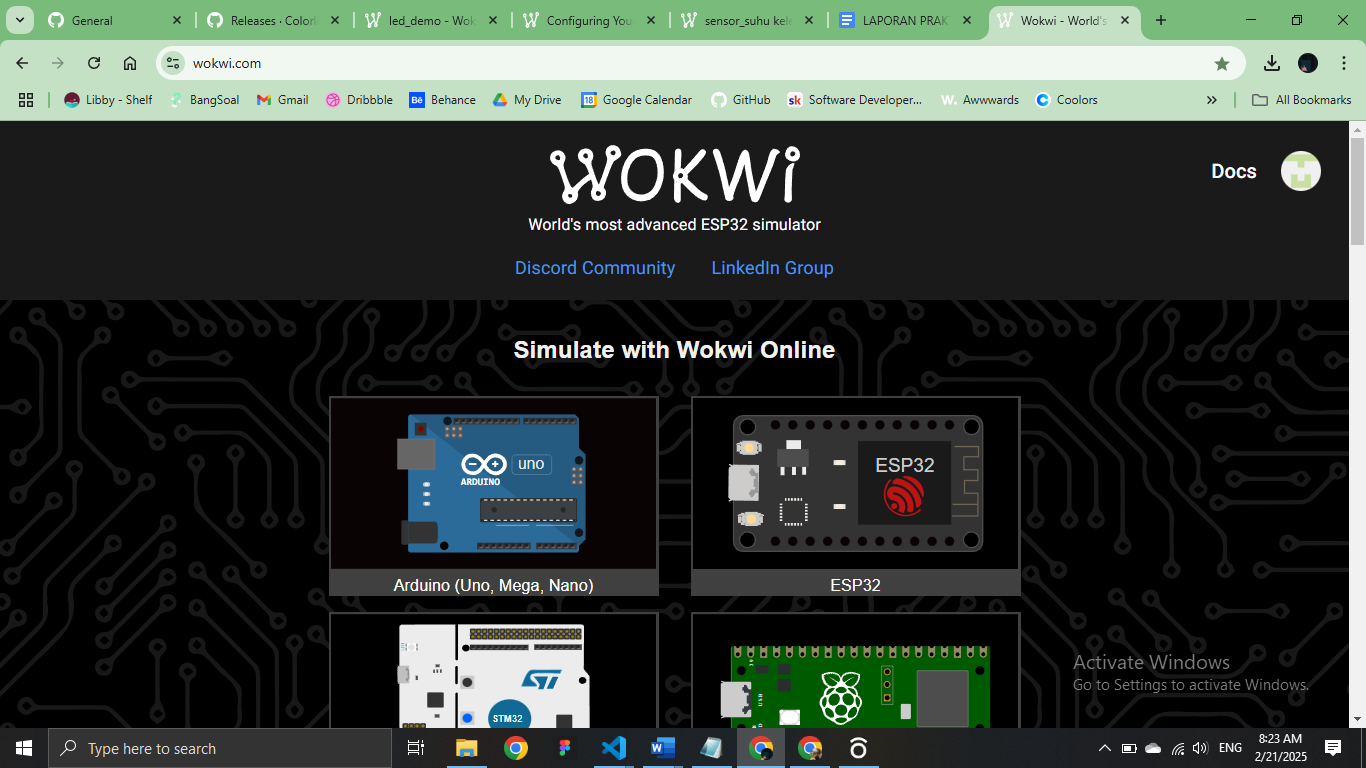
1. **Introduction (Pendahuluan)**
   1. **Latar Belakang**

Perkembangan teknologi IoT semakin pesat dan telah diterapkan dalam berbagai aspek kehidupan, termasuk dalam sistem pengukuran otomatis. Salah satu sensor yang sering digunakan dalam sistem pengukuran berbasis IoT adalah sensor ultrasonik, yang mampu mengukur jarak dengan prinsip pantulan gelombang suara. Dengan menggunakan mikrokontroler ESP32, data hasil pengukuran dapat diolah dan ditampilkan secara real-time, memungkinkan pengguna untuk memperoleh informasi jarak dengan lebih efisien dan akurat.

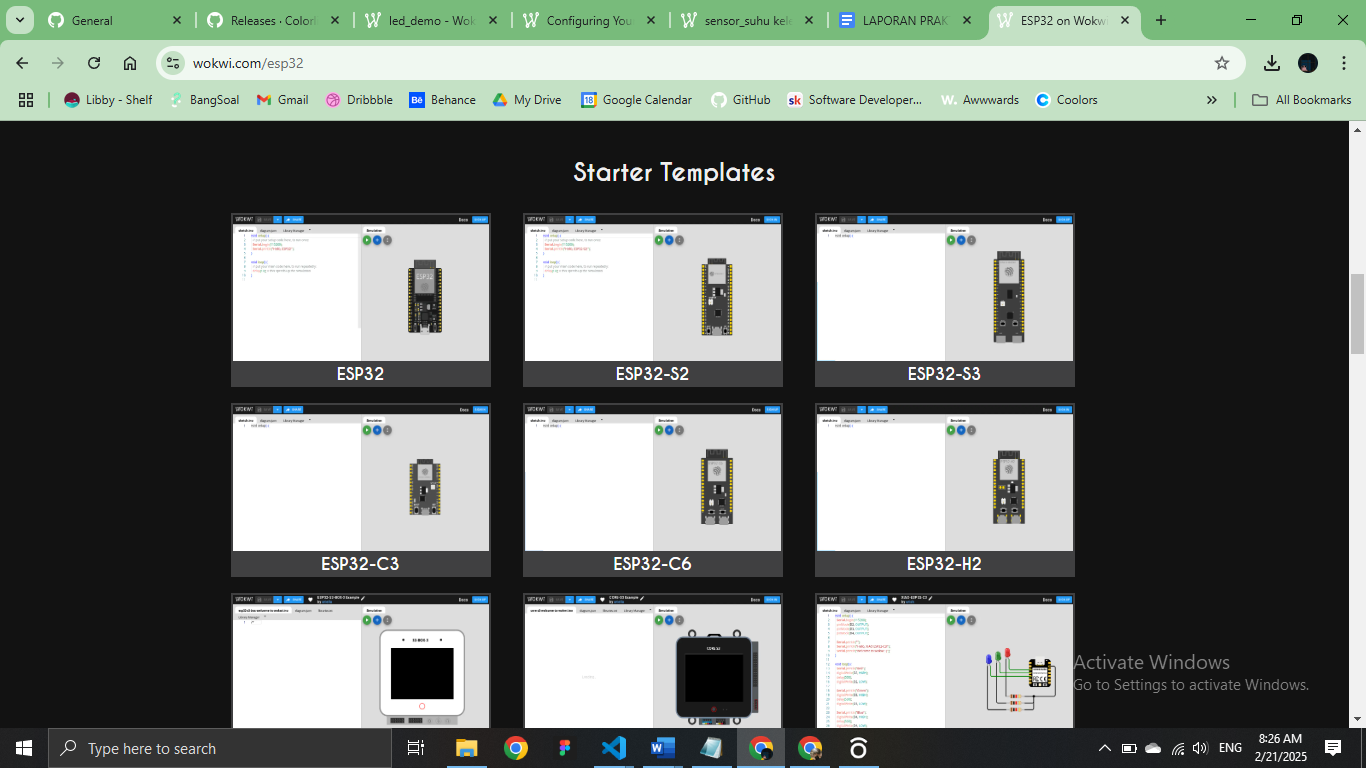
Melalui praktikum ini, mahasiswa mendapatkan pengalaman langsung dalam merancang dan mengimplementasikan rangkaian sensor ultrasonik menggunakan ESP32. Dengan pemrograman di Visual Studio Code (VS Code), mahasiswa dapat memahami cara kerja sensor ultrasonik, pemrosesan data sensor oleh mikrokontroler, serta penerapan IoT dalam sistem pengukuran otomatis. Praktikum ini bertujuan untuk membekali mahasiswa dengan pemahaman dasar dalam pengembangan sistem pengukuran berbasis mikrokontroler yang dapat diterapkan dalam berbagai aplikasi, seperti robotika, sistem keamanan, dan otomasi industri.

* 1. **Tujuan Eksperimen**

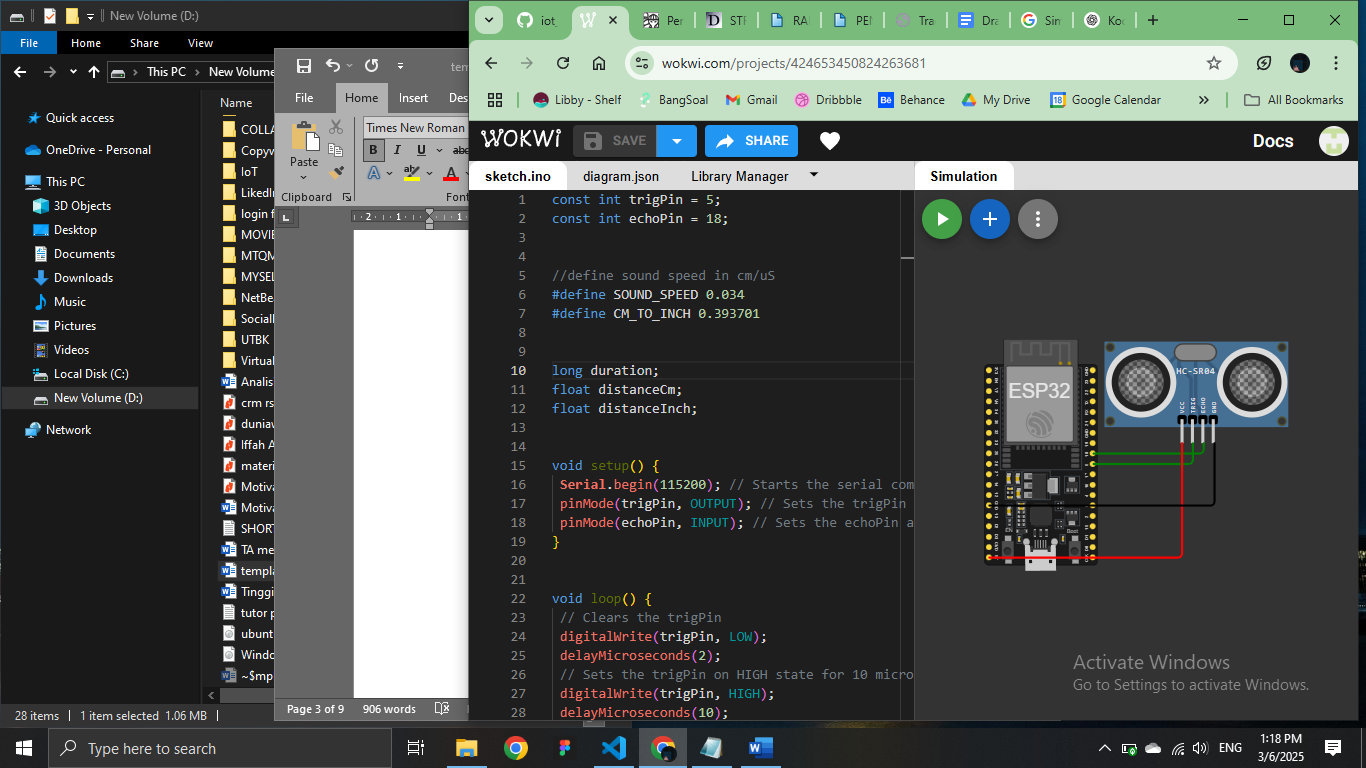
1. Memahami prinsip kerja sensor ultrasonik dalam mengukur jarak berdasarkan pantulan gelombang suara.
2. Mengimplementasikan rangkaian sensor ultrasonik dan menghubungkannya dengan ESP32 serta menguji fungsinya dalam sistem pengukuran sederhana.
3. Mengoptimalkan pemrograman mikrokontroler untuk sistem otomasi sederhana dan memahami cara mengontrol perangkat eksternal melalui kode program.
4. **Methodology (Metodologi)**
   1. **Tools & Materials (Alat dan Bahan)**
5. Laptop
6. Internet
7. Aplikasi ( Visual Studio Code )
8. Website ( Wokwi.com)
   1. **Langkah Implementasi**
9. Pembuatan Akun Wokwi
10. Membuka website Wokwi (<https://wokwi.com>)



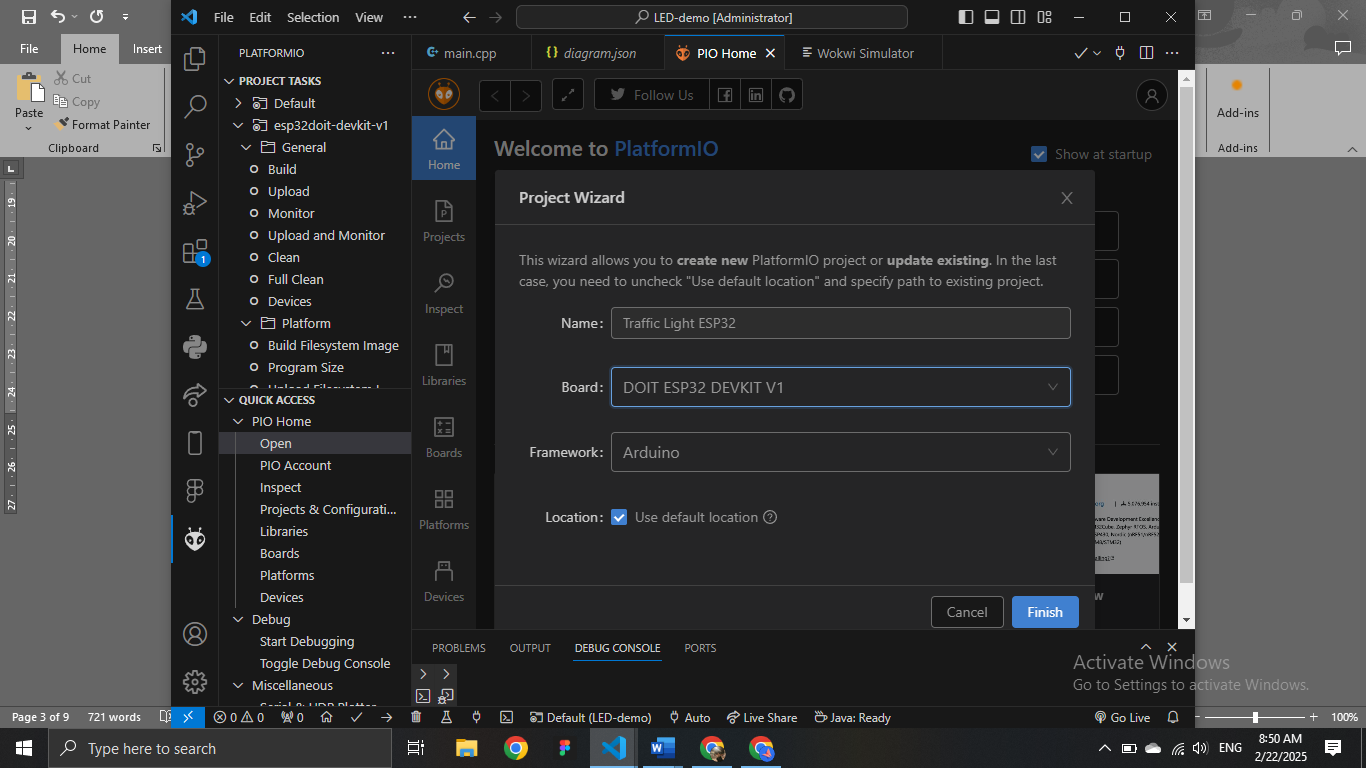
1. Pilih ESP32, lalu pilih salah satu starter template sesuai yang diinginkan



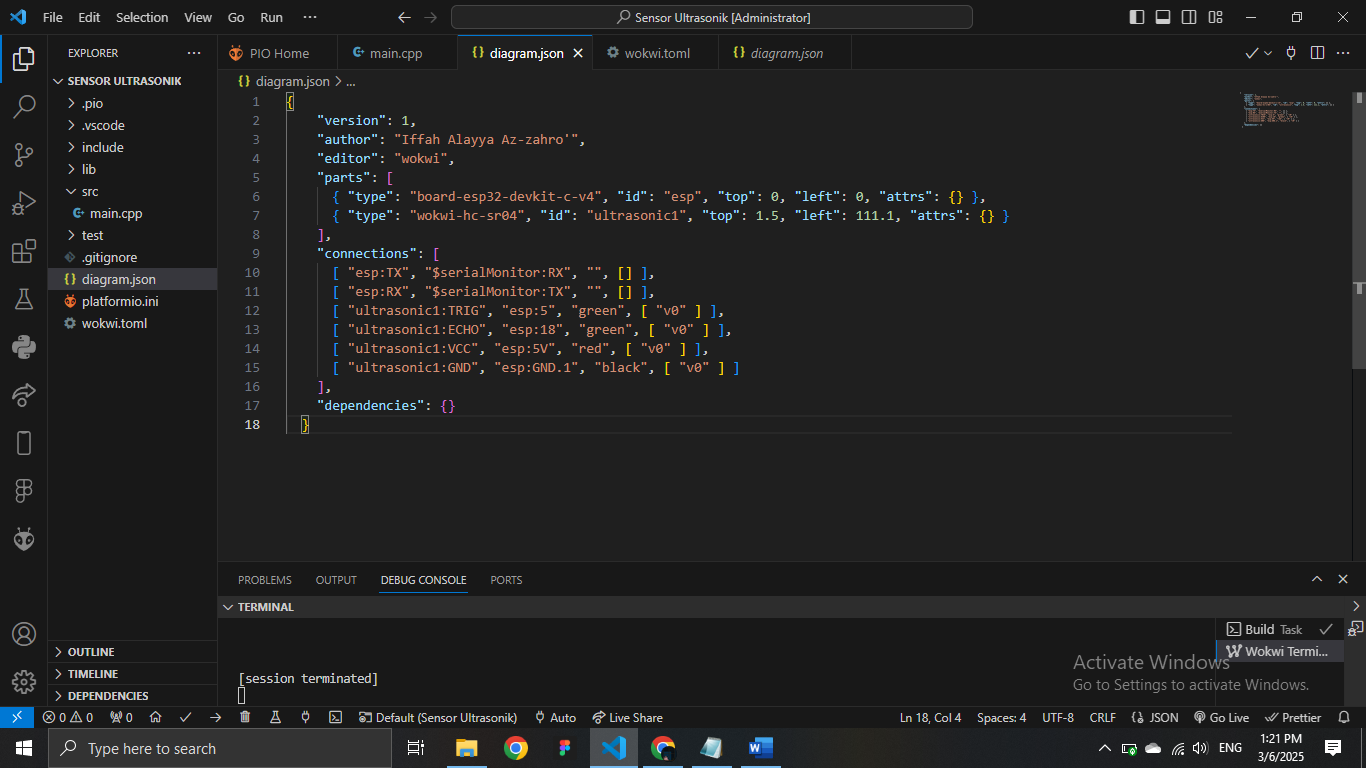
1. Kemudian rangkai alat sensor ultrasonik dan masukkan code pada sketch.io untuk memulai simulasi pada alat tersebut.

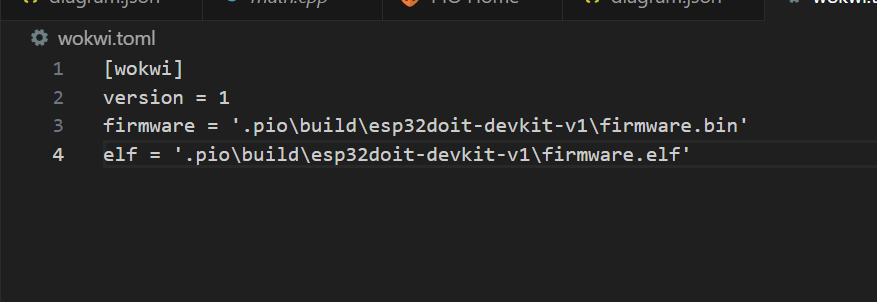


1. Mulai Implementasi Code pada vscode.
2. Membuka Aplikasi Visual Studio Code , kemudian pilih platform.io dan pilih create new project.

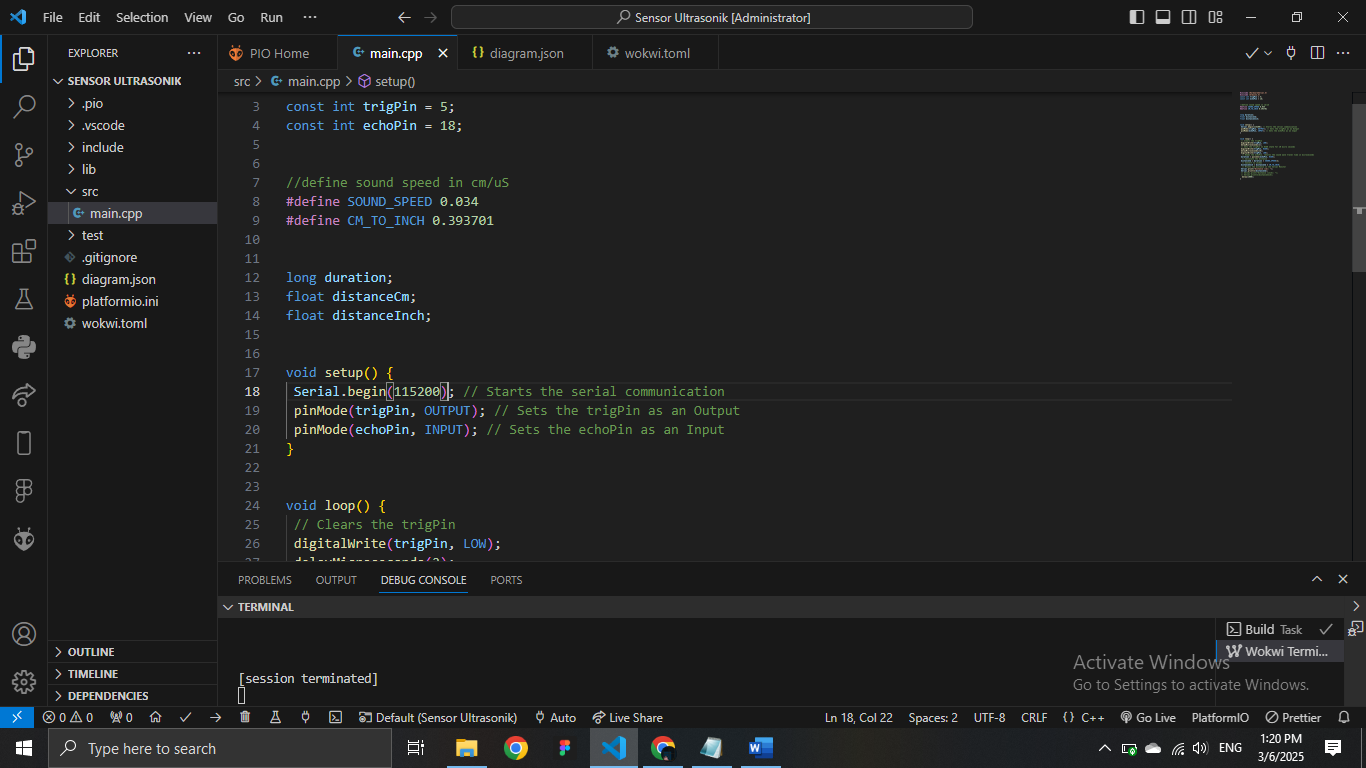


1. Buat file diagram.json dan wokwi.toml dan isi dengan kode masing masing, Paste Code Diagram.json.

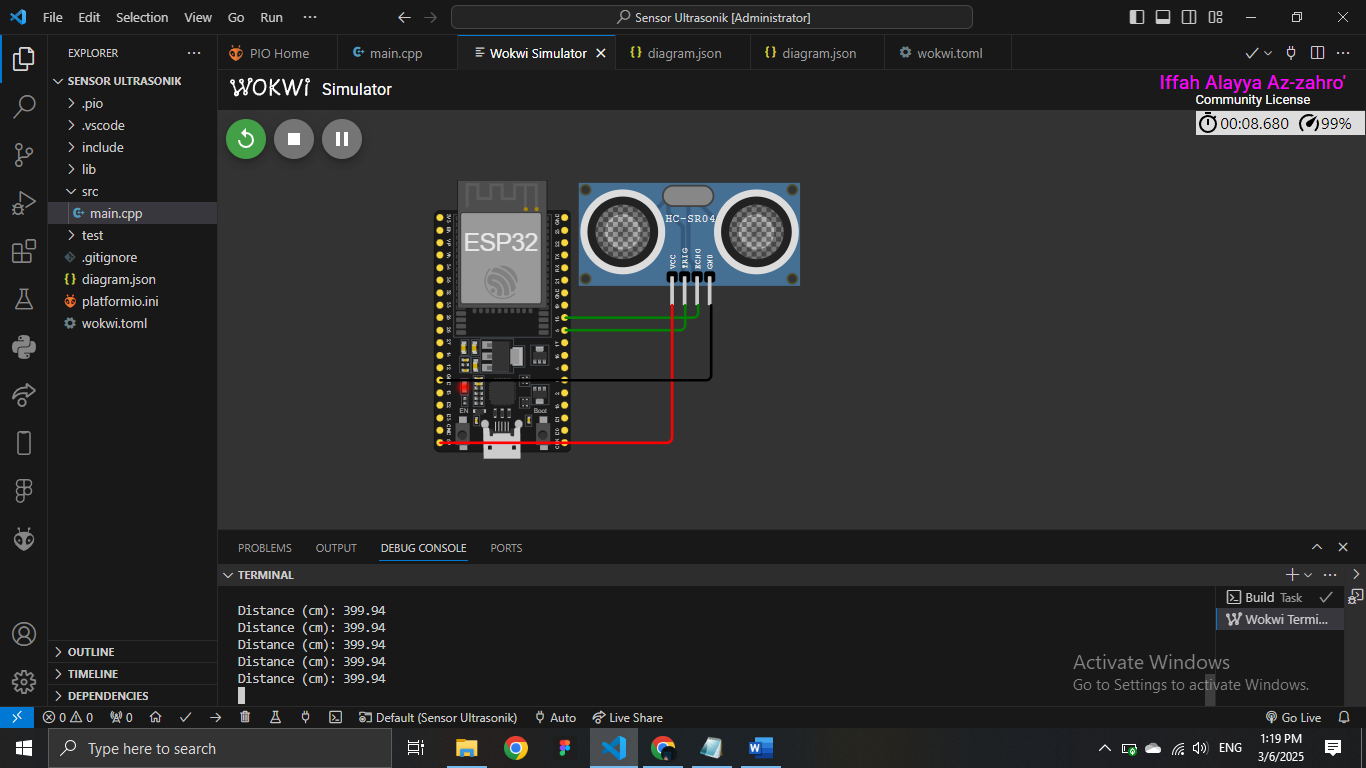




1. Masukkan Code ke dalam main.cpp, lalu compile.

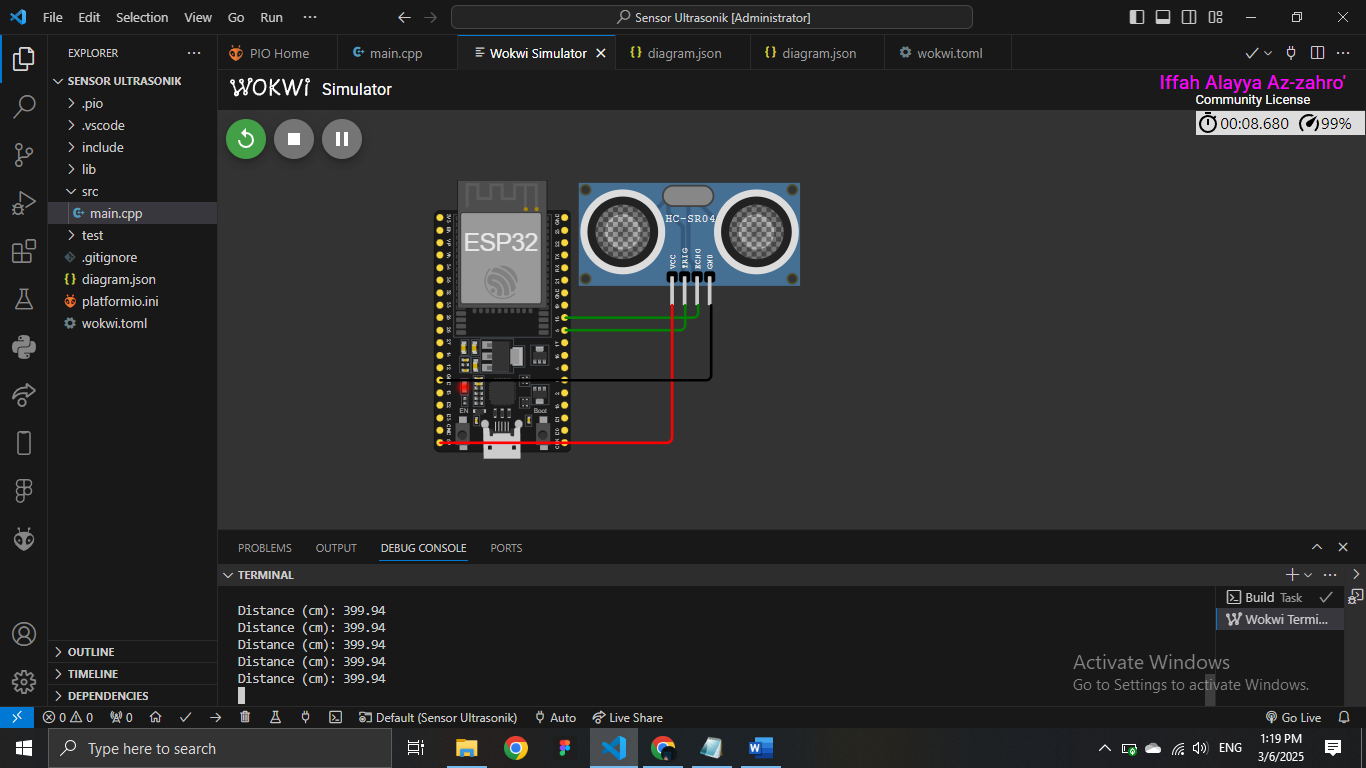


1. Jika tidak ada yang error , lanjutkan simulasi dengan ‘start wokwi simulator’



1. **Results and Discussion (Hasil dan Pembahasan)**
   1. Experimental Results (Hasil Eksperimen)

Hasil dari praktikum adalah pengguna dapat menjalankan Traffic Lights sesuai dengan tahap tahap yang diberikan, Jika Simulasi Traffic Lights menyala dan berjalan dengan baik maka semua lampu akan menyala sesuai dengan kode yang telah dibuat.



1. **Appendix (Lampiran)**
   1. Kode program main.cpp pada VSCode
2. #include <HardwareSerial.h>
3. #include <Arduino.h>
4. const int trigPin = 5;
5. const int echoPin = 18;
6. //define sound speed in cm/uS
7. #define SOUND\_SPEED 0.034
8. #define CM\_TO\_INCH 0.393701
9. long duration;
10. float distanceCm;
11. float distanceInch;
12. void setup() {
13. Serial.begin(115200); // Starts the serial communication
14. pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
15. pinMode(echoPin, INPUT); // Sets the echoPin as an Input
16. }
17. void loop() {
18. digitalWrite(trigPin, LOW);
19. delayMicroseconds(2);
20. // Sets the trigPin on HIGH state for 10 micro seconds
21. digitalWrite(trigPin, HIGH);
22. delayMicroseconds(10);
23. digitalWrite(trigPin, LOW);
24. // Reads the echoPin, returns the sound wave travel time in microseconds
25. duration = pulseIn(echoPin, HIGH);
26. // Calculate the distance
27. distanceCm = duration \* SOUND\_SPEED/2;
28. // Convert to inches
29. distanceInch = distanceCm \* CM\_TO\_INCH;
30. // Prints the distance in the Serial Monitor
31. Serial.print("Distance (cm): ");
32. Serial.println(distanceCm);
33. delay(1000);
34. }
    1. Kode program diagram json
35. {
36. "version": 1,
37. "author": "Iffah Alayya Az-zahro'",
38. "editor": "wokwi",
39. "parts": [
40. { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
41. { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 1.5, "left": 111.1, "attrs": {} }
42. ],
43. "connections": [
44. [ "esp:TX", "$serialMonitor:RX", "", [] ],
45. [ "esp:RX", "$serialMonitor:TX", "", [] ],
46. [ "ultrasonic1:TRIG", "esp:5", "green", [ "v0" ] ],
47. [ "ultrasonic1:ECHO", "esp:18", "green", [ "v0" ] ],
48. [ "ultrasonic1:VCC", "esp:5V", "red", [ "v0" ] ],
49. [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ]
50. ],
51. "dependencies": {}
52. }