LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktikum Simulasi Sensor Jarak (Ultrasonic)**

****

*Abdur Rohman Syah Sjadja’ah*

Fakultas Vokasi, Universitas Brawijaya

Email: [abdurrohmansyh02@gmail.com](mailto:abdurrohmansyh02@gmail.com)

**Abstract** (Abstrak)

Praktikum ini bertujuan untuk mengetahui bagaimana cara membuat alat Sensor Jarak melalui Website Wokwi dan disimulasikan melalui Visual Studio Code (VSCode ). Simulasi ini ini terdiri dari HC-SR04 dan ESP32 yang nantinya saling terhubung menggunakan kabel. Sistem ini menggunakan sensor ultrasonik HC-SR04, yang terdiri dari pemicu (trigPin) dan gema (echoPin) untuk mengukur jarak berdasarkan waktu yang dibutuhkan gelombang suara untuk merambat dan memantul. ESP32 membaca durasi sinyal yang dipantulkan dan menghitung jarak dalam sentimeter dan inci, menampilkan hasilnya di Serial Monitor. Hasilnya menunjukkan bahwa kemampuan mikrokontroler secara akurat mengukur jarak secara real-time tanpa memerlukan perangkat keras fisik. Eksperimen ini memberikan wawasan mendasar tentang pemrograman mikrokontroler, integrasi komponen elektronik, dan implementasi sistem berbasis IoT.

***Kata kunc : i****Internet of Things, ESP32, HC-SR04, VSCode, Wokwi Simulator*.

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

Dengan pesatnya perkembangan teknologi saat ini keterampilan dalam pemrograman, dan manajemen proyek semakin penting, terutama dalam pengembangan system berbasis mikrokontroler. Wokwi sebagai platform simulator mikrokontroler memungkinkan kita untuk membuat berbagai simulasi IoT tanpa perlu perangkat keras. Tentunya platform Wokwi ini sangat membantu untuk memudahkan pengujian kode sebelum implementasi nyata. Tidak hanya Wokwi, Visual Studio Code (VSCode) adalah teks editor yang populer yang mendukung berbagai bahasa pemrograman yang nantinya memudahkan mengedit kode. Dengan berbagai ekstensi yang mendukung pengembangan mikrokontroler. Fitur seperti debugging memudahkan pengguna menulis kode dengan lebih efisien dan meminimalisir kesalahan.

Melalui praktikum ini, pengguna akan mempelajari cara membuat Sensor Jarak menggunakan ESP32 dan HC-SR04 dengan menggunakan platform Wokwi dan disimulasikan melalui VSCode.

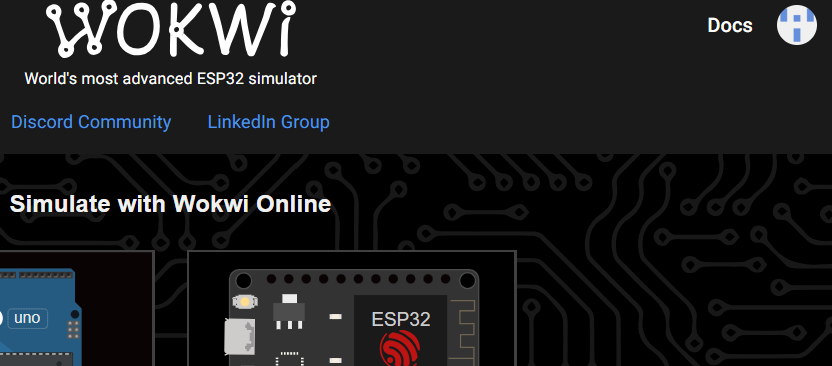
* 1. **Tujuan Eksperimen**

Praktikum ini bertujuan untuk mengetahui langkah-langkah cara mensimulasikan HC-SR04 dan ESP32 menjadi sensor ultrasonic menggunakan platform Wokwi. Dengan Wokwi pengguna dapat melakukan perancangan system ESP32 dan HC-SR04 yang saling terhubung secara virtual. Sementara VSCode digunakan untuk mengedit kode program agar jalannya sesuai dengan yang diharapkan. Melalui praktikum ini Mahasiswa diharapkan dapat mengetahui apa saja tools dan perangkat yang diperlukan untuk melakukan praktik.

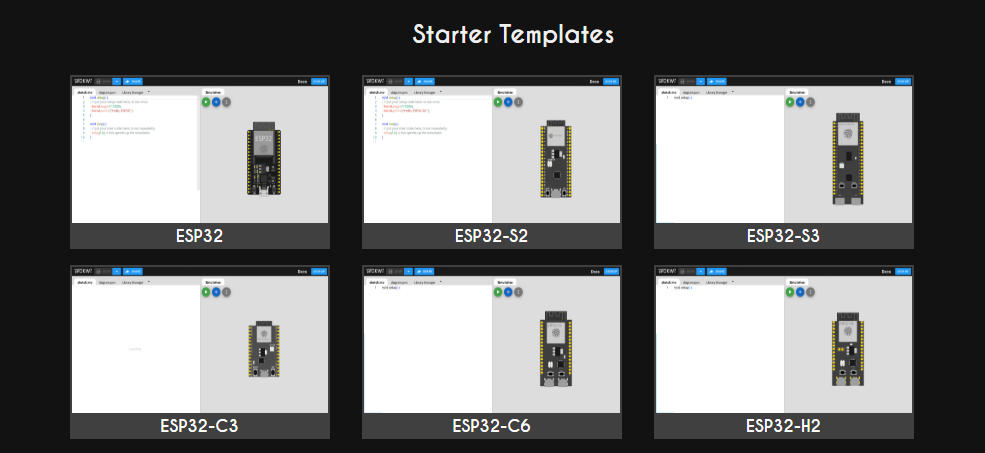
1. **Methodology** (Metodologi)
   1. **Tools & Materials** (Alat & Bahan)

* Laptop
* Internet
* Aplikasi (VSCode)
* Website Wokwi (<https://wokwi.com/>)
  1. **Implementation Steps** (Langkah Implementasi)

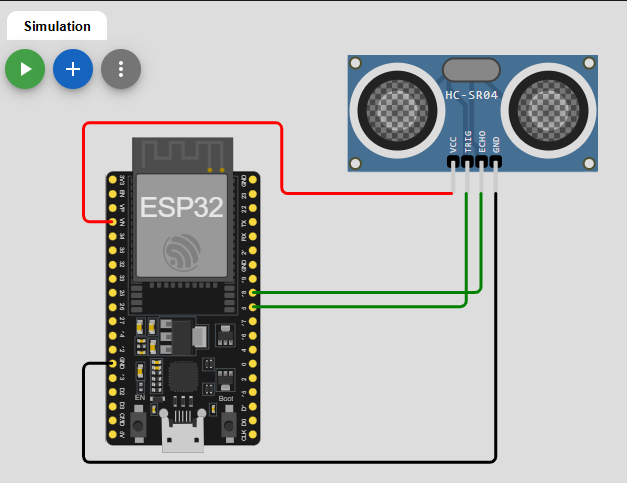
1. Buka website <https://wokwi.com/> lalu sign up terlebih dahulu menggunakan akun Github kalian. Jika sudah Sign up maka foto profil Github akan terlihat di pojok kanan atas

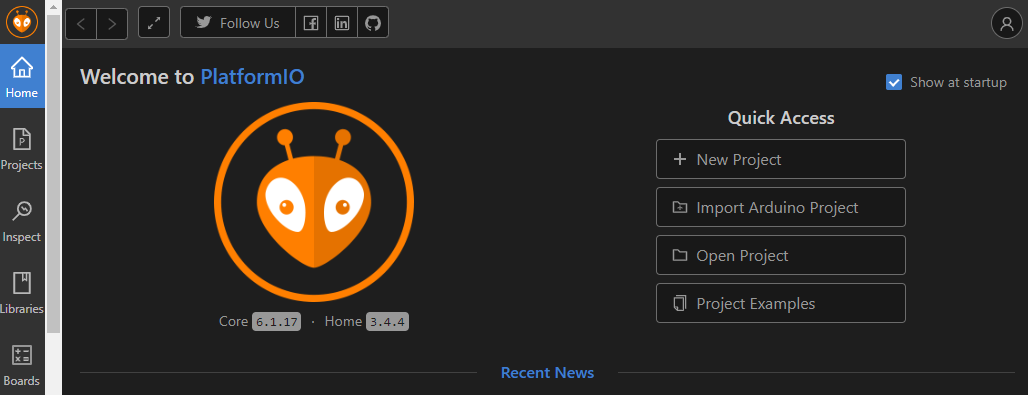
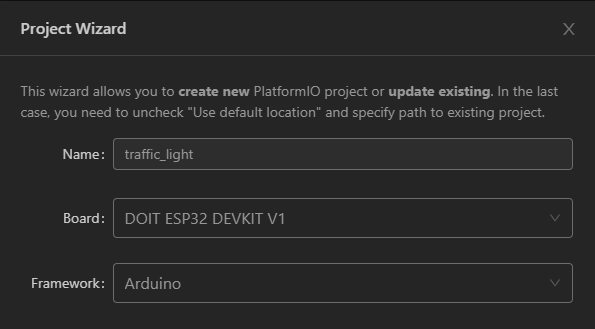
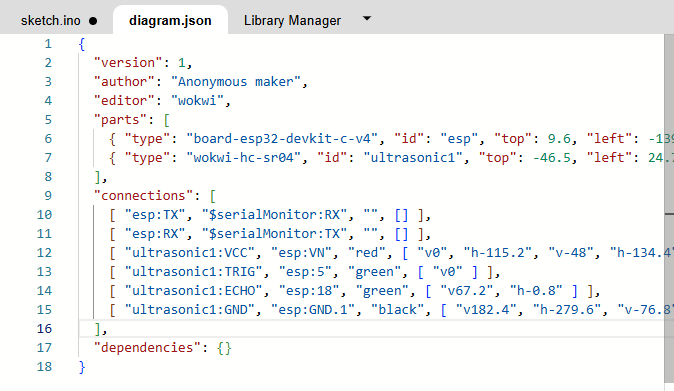
****

1. Pilih opsi ESP32
2. Scroll ke bawah sampai menemukan Starter Template, lalu pilih ESP32 paling atas sebelah kiri



1. Buat simulasi sesuai gambar di bawah ini



1. ****Selanjutnya buka VSCode dan pilih PlatformIO. Klik new project dan isi seperti gambar di bawah ini. Tunggu hingga project selesai dibuat
2. Selanjutnya buat file baru yaitu diagram.json dan wokwi.toml. Copy kode diagram.json dari wokwi dan paste ke file diagram.json yang ada di VSCode. File wokwi.toml bisa diisi sesuai gambar di bawah ini

Kode file wokwi.toml

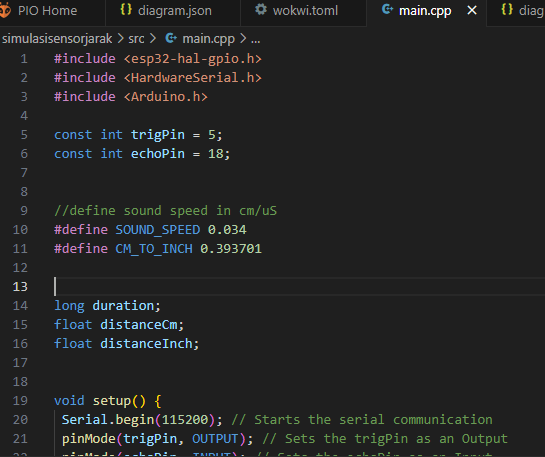
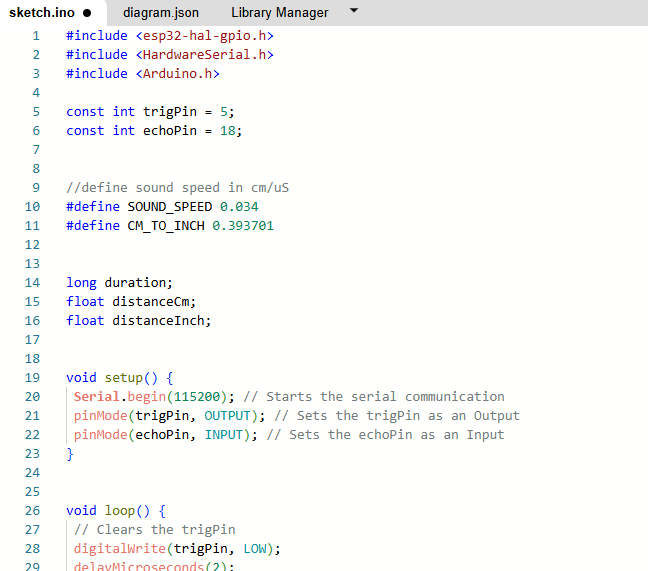
[wokwi]

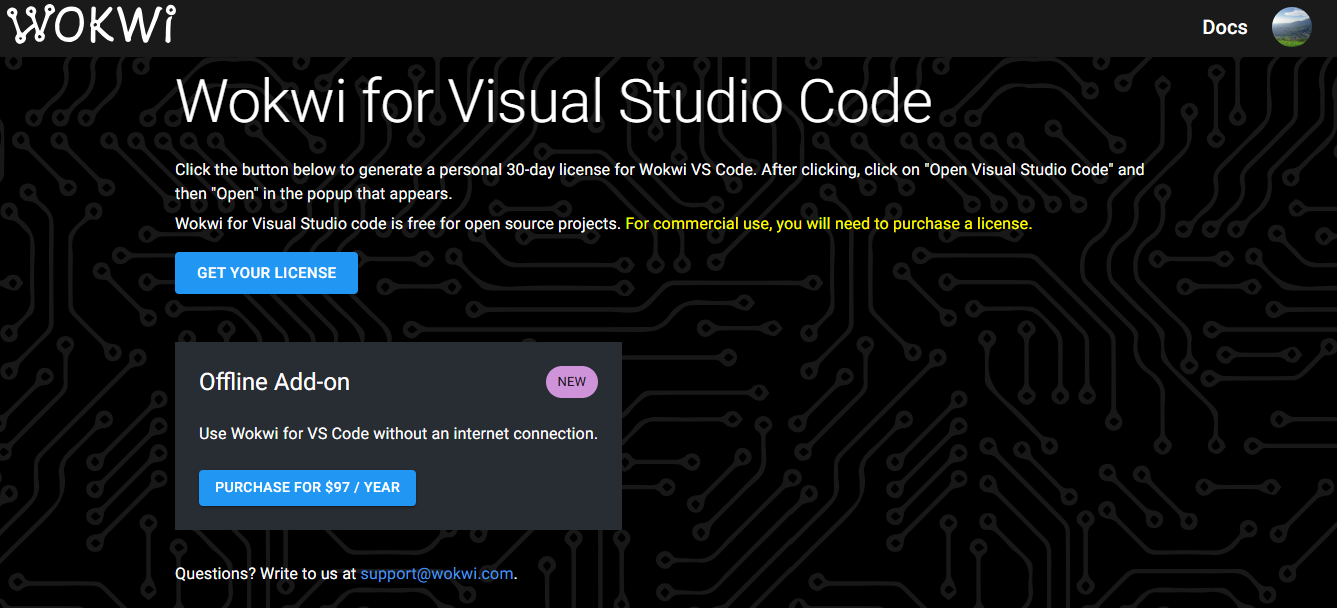
version = 1

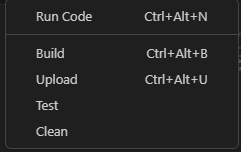
firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

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

1. Copy kode sketch.ino di wokwi dan paste ke folder src/main.cpp di VSCode

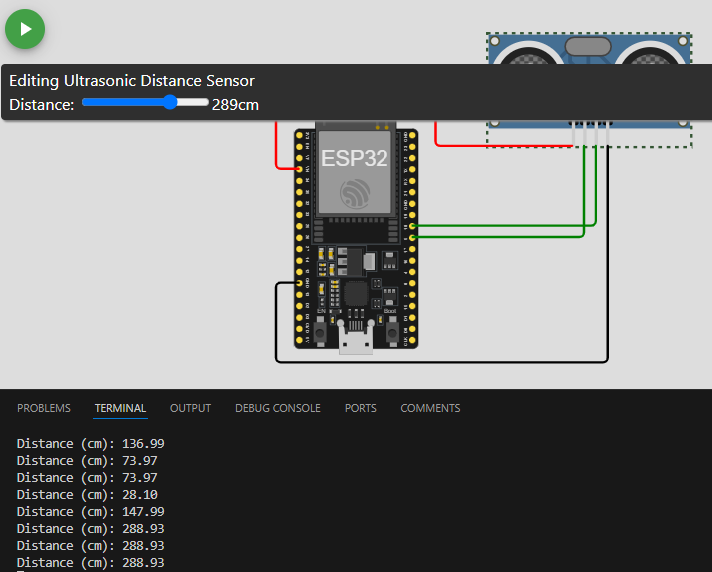


1. Dapatkan lisensi wokwi untuk VSCode klik “GET YOUR LICENSE”. License wokwi dapat dicari di sini (<https://wokwi.com/license>)
2.  Untuk melakukan simulasi melalui VSCode. Klik “Build” terlebih dahulu di file main.cpp



1. Setelah itu ketik “>Wokwi Start Simulator” dan simulasi bisa dijalankan



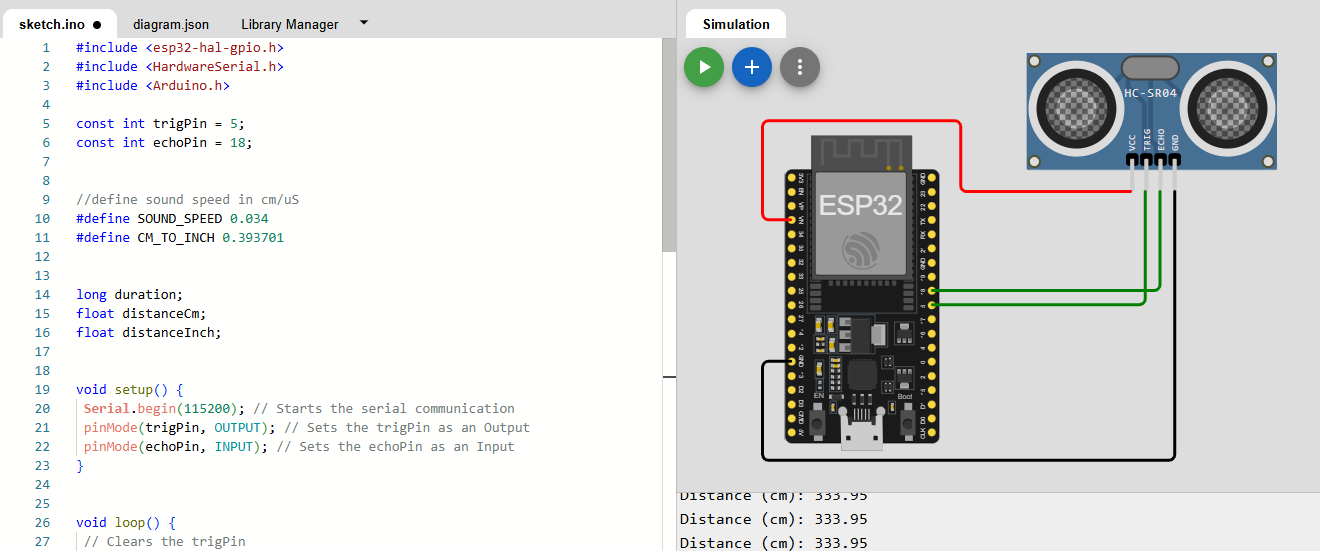


**3. Results and Discussion** (Hasil dan Pembahasan)

**3.1 Experimental Results** (Hasil Eksperimen)

Simulasi sistem sensor jarak menggunakan ESP32 di Wokwi Simulator telah berhasil diimplementasikan. Proses kompilasi dan build menggunakan PlatformIO di Visual Studio Code (VSCode) menunjukkan hasil (SUCCESS) tanpa error. Hasil pengujian menunjukkan bahwa sensor ultrasonic pada HC-SR04 mampu membaca jarak objek di depannya dengan akurasi tinggi, data jarak berhasil ditampilkan di Serial Monitor, menunjukkan nilai yang berubah sesuai dengan yang kita ubah pada HC-SR04. ESP32 merespon perubahan jarak secara real-time dengan waktu pembacaan yang cepat dan stabil.

**Appendix** (Lampiran)



Kode Program main.cpp

#include <esp32-hal-gpio.h>

#include <HardwareSerial.h>

#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);

}