# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

# Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak(Ultrasonic)**

**Author(s)** Muhammad Hafizh Al Furqon

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*muhammad.hafizhalfurqon@gmail.com*](mailto:muhammad.hafizhalfurqon@gmail.com)

Praktikum ini bertujuan untuk memahami prinsip kerja sensor ultrasonik dalam mengukur jarak menggunakan mikrokontroler ESP32. Simulasi dilakukan dengan bantuan Wokwi untuk mendesain skematik rangkaian serta PlatformIO untuk pengembangan perangkat lunak. Implementasi mencakup pembacaan data dari sensor ultrasonik HC-SR04 dan menampilkan hasilnya pada serial monitor. Hasil percobaan menunjukkan bahwa sensor dapat mengukur jarak dengan akurasi yang cukup baik dalam kondisi ideal. Penelitian ini memberikan pemahaman mendasar tentang penggunaan sensor ultrasonik dalam aplikasi berbasis IoT.

**Keywords**—*Wokwi, PlatformIO, ESP32, Simulasi, Sensor Jarak*

**1. Introduction**

**1.1 Latar belakang**

Dalam dunia Internet of Things (IoT), penggunaan sensor ultrasonik menjadi salah satu metode yang efisien dalam mendeteksi jarak. Sensor ultrasonik HC-SR04 bekerja dengan mengirimkan gelombang suara berfrekuensi tinggi dan mengukur waktu yang dibutuhkan untuk menerima pantulan gelombang tersebut. Dengan teknologi ini, berbagai aplikasi dapat dikembangkan, seperti sistem parkir otomatis, robotika, serta perangkat pemantauan jarak. Pada praktikum ini, dilakukan simulasi penggunaan sensor ultrasonik dengan mikrokontroler ESP32 untuk memahami bagaimana sensor ini bekerja dalam sistem otomasi.

**1.2 Tujuan eksperimen**

1. Memahami prinsip kerja sensor ultrasonik dalam mengukur jarak.
2. Mengimplementasikan pembacaan data dari sensor ultrasonik menggunakan ESP32.
3. Menampilkan hasil pengukuran jarak melalui serial monitor.
4. Menguji akurasi sensor dalam kondisi simulasi.

**2. Methodology (Metodologi)**

**2.1 Tools & Materials (Alat dan Bahan)**

* Mikrokontroler: ESP32
* Sensor Ultrasonik HC-SR04
* Kabel Jumper

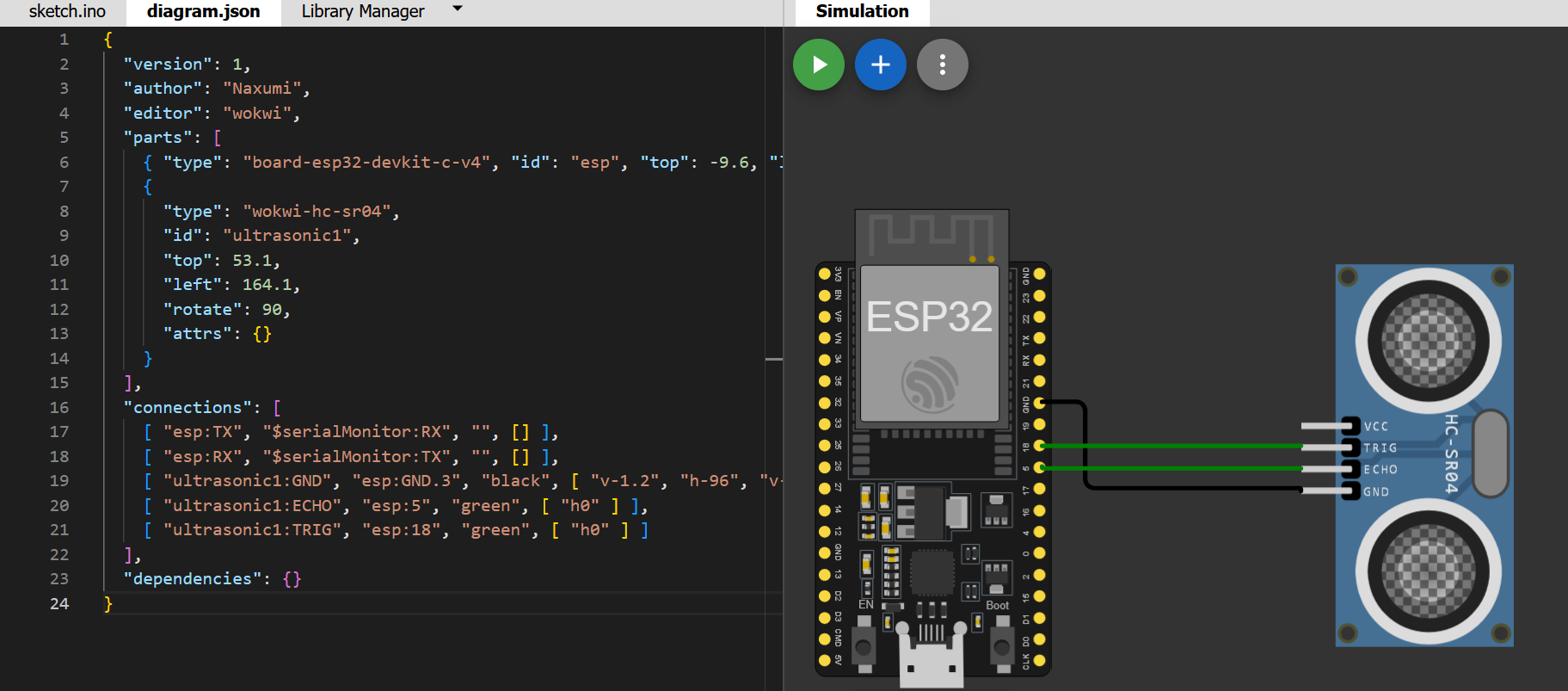
Software:

* Visual Studio Code (VSCode)
* Extension PlatformIO (untuk proses compile)
* Extension Wokwi (untuk simulasi dan pembuatan diagram)
* Github (untuk repository dan version control)

Diagram: File diagram.json yang diambil dari website Wokwi sebagai acuan skematik

**2.2 Implementation Steps (Langkah Implementasi)**

1. Buat topologi dan code terlebih dahulu di wokwi.com agar nanti diagram.json bisa digunakan di Visual Studio Code



1. Buat project baru menggunakan PlatformIO, sertakan diagram.json yang sudah terbuat di wokwi.com, dan file konfigurasi wokwi.toml beserta path firmware dan elf yang didapatkan setelah meng-build code-nya

struktur

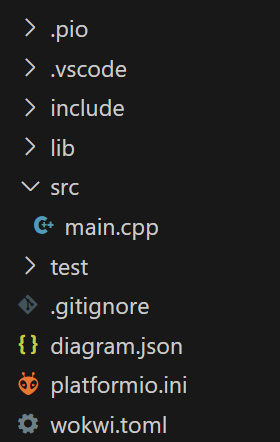


diagram.json

{

"version": 1,

"author": "Naxumi",

"editor": "wokwi",

"parts": [

{ "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -9.6, "left": -14.36, "attrs": {} },

{

"type": "wokwi-hc-sr04",

"id": "ultrasonic1",

"top": 53.1,

"left": 164.1,

"rotate": 90,

"attrs": {}

}

],

"connections": [

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

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

[ "ultrasonic1:GND", "esp:GND.3", "black", [ "v-1.2", "h-96", "v-38.4" ] ],

[ "ultrasonic1:ECHO", "esp:5", "green", [ "h0" ] ],

[ "ultrasonic1:TRIG", "esp:18", "green", [ "h0" ] ]

],

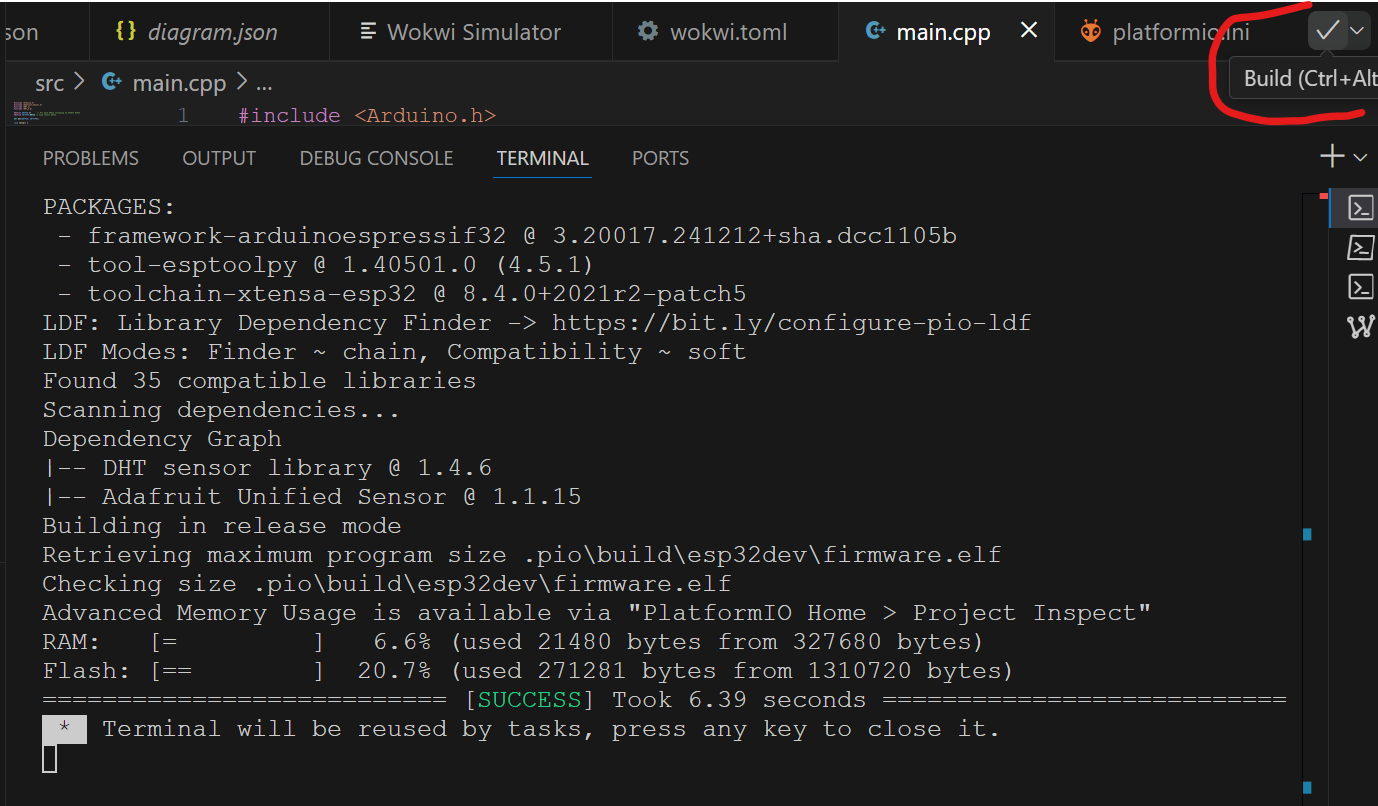
"dependencies": {}

}

wokwi.toml

| [wokwi] version = 1 firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin' elf = '.pio\build\esp32doit-devkit-v1\firmware.elf' |
| --- |

Build



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

Praktikum ini berhasil mensimulasikan penggunaan sensor ultrasonik dengan ESP32 menggunakan Wokwi dan PlatformIO. Sensor dapat membaca jarak berdasarkan waktu tempuh gelombang ultrasonik dan menampilkan hasilnya pada serial monitor. Data hasil percobaan menunjukkan bahwa sensor bekerja dengan baik dalam kondisi ideal.

**4. Appendix (Lampiran, jika diperlukan)**

**4.1 Kode Program (src/main.cpp)**

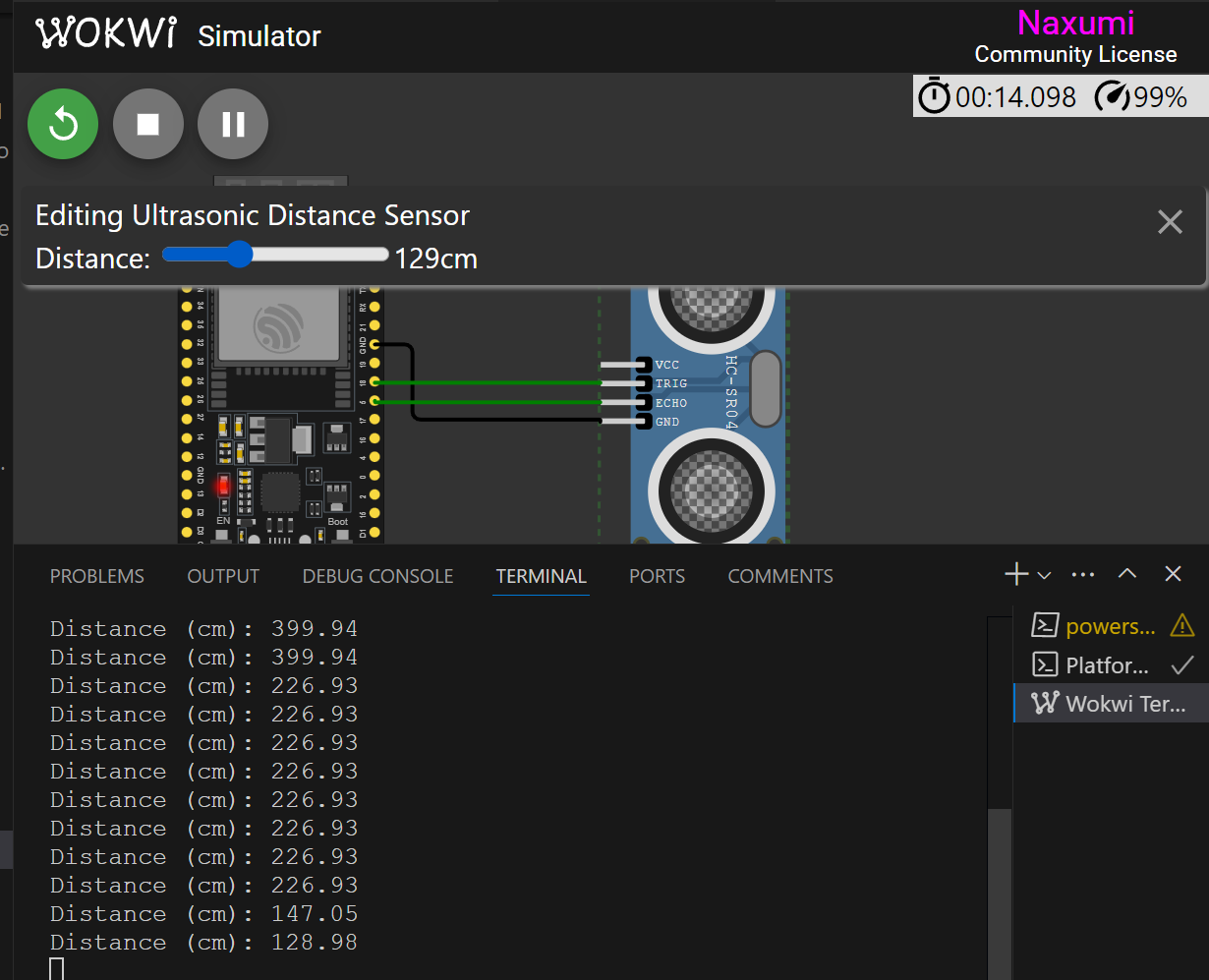
| #include <Arduino.h>  const int trigPin = 18;  const int echoPin = 5;  //define sound speed in cm/uS  #define SOUND\_SPEED 0.034  #define CM\_TO\_INCH 0.393701  long duration;  float distanceCm;  float distanceInch;  void setup() {  Serial.begin(115200); // Starts the serial communication  pinMode(trigPin, 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);  } |
| --- |

**4.2 Diagram Skematik (diagram.json)**

File diagram.json yang dihasilkan dari website Wokwi disimpan dalam folder proyek (misalnya, docs/diagram.json) dan berisi informasi visual mengenai koneksi antara LED dan ESP32. Contoh struktur file:

| {  "version": 1,  "author": "Naxumi",  "editor": "wokwi",  "parts": [  { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -9.6, "left": -14.36, "attrs": {} },  {  "type": "wokwi-hc-sr04",  "id": "ultrasonic1",  "top": 53.1,  "left": 164.1,  "rotate": 90,  "attrs": {}  }  ],  "connections": [  [ "esp:TX", "$serialMonitor:RX", "", [] ],  [ "esp:RX", "$serialMonitor:TX", "", [] ],  [ "ultrasonic1:GND", "esp:GND.3", "black", [ "v-1.2", "h-96", "v-38.4" ] ],  [ "ultrasonic1:ECHO", "esp:5", "green", [ "h0" ] ],  [ "ultrasonic1:TRIG", "esp:18", "green", [ "h0" ] ]  ],  "dependencies": {}  } |
| --- |

**4.3 Screenshot**

****