

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Praktik Simulasi Relay, Button & LED serta Praktik Simulasi Sensor Jarak (Ultrasonic)

Emilda Citra Diva Revanda

233140701111001

Fakultas Vokasi, Universitas Brawijaya
Email: citraimelda883@gmail.com

Abstract

Eksperimen ini bertujuan untuk mensimulasikan penggunaan relay, button, dan LED serta sensor jarak ultrasonic dalam sistem berbasis IoT. Relay digunakan untuk mengontrol perangkat listrik, button sebagai input, dan LED sebagai indikator. Sensor jarak ultrasonic berfungsi untuk mendeteksi objek berdasarkan pantulan gelombang ultrasonik. Implementasi dilakukan menggunakan ESP32 dan Wokwi sebagai simulator. Hasil eksperimen menunjukkan bahwa sistem bekerja dengan baik, memungkinkan kontrol dan pemantauan secara real-time. Penggunaan teknologi IoT dalam eksperimen ini membuktikan efisiensi dan fleksibilitas dalam pengelolaan perangkat berbasis sensor dan aktuator.

Keywords—Internet of Things, Relay, Button, LED, Ultrasonic Sensor

1. Introduction

1.1 Latar Belakang

Internet of Things (IoT) memungkinkan integrasi antara sensor dan aktuator untuk otomatisasi berbagai aplikasi. Dalam eksperimen ini, relay digunakan untuk mengendalikan perangkat listrik, button sebagai pemicu aksi, LED sebagai indikator visual, dan sensor ultrasonic untuk mengukur jarak. Teknologi ini banyak digunakan dalam smart home, sistem keamanan, dan industri otomatisasi.

1.2 Tujuan Eksperimen

1. Mempelajari implementasi relay, button, dan LED dalam sistem berbasis IoT.
2. Memahami cara kerja sensor ultrasonic dalam mendeteksi jarak.
3. Menganalisis penggunaan ESP32 dan Wokwi sebagai alat simulasi dalam pengujian sistem IoT.

2. Methodology

2.1 Tools & Materials

- **Mikrokontroler:** ESP32
- **Perangkat Lunak:** Wokwi Simulator, Arduino IDE
- **Sensor & Aktuator:** Relay, Button, LED, Sensor Ultrasonic (HC-SR04)

2.2 Implementation Steps

1. Praktik Simulasi Relay, Button & LED

- Menghubungkan relay, button, dan LED ke ESP32.
- Memprogram ESP32 untuk mengontrol relay berdasarkan input dari button.
- Menguji fungsionalitas LED sebagai indikator status relay.

2. Praktik Simulasi Sensor Jarak (Ultrasonic)

- Menghubungkan sensor ultrasonic HC-SR04 ke ESP32.
- Memprogram ESP32 untuk membaca data jarak dan menampilkan hasil di serial monitor.
- Menganalisis keakuratan sensor dalam mendeteksi objek.

3. Results and Discussion

3.1 Experimental Results

Hasil eksperimen menunjukkan bahwa sistem dapat berfungsi sesuai dengan yang diharapkan:

- **Relay & Button:** Button berhasil digunakan untuk mengontrol relay, memungkinkan simulasi sistem saklar elektronik.
- **LED:** LED menyala sebagai indikator ketika relay diaktifkan.
- **Sensor Ultrasonic:** Sensor HC-SR04 mampu mendeteksi objek dalam rentang tertentu dengan akurasi yang cukup baik.
- **Wokwi Simulator:** Memfasilitasi pengujian tanpa perangkat keras fisik, meningkatkan efisiensi dalam tahap pengembangan.

4. Appendix

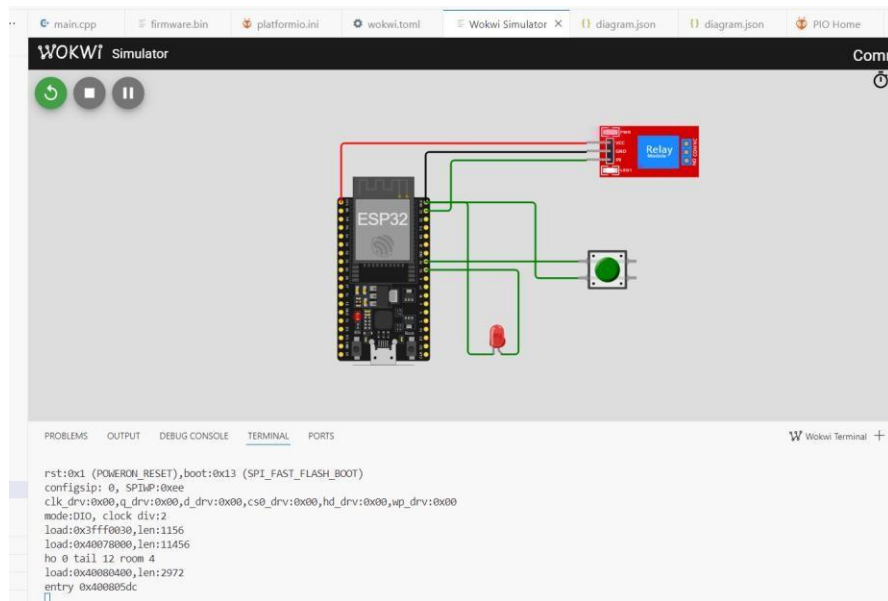
LED RELAY BUTTON

```
{  
  "version": 1,  
  "author": "Anonymous  
maker", "editor": "wokwi",  
  "parts": [  
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 9.6, "left": -215.96, "attrs": { } },  
    { "type": "wokwi-relay-module", "id": "relay1", "top": -47.8, "left": 67.2, "attrs": { } },
```

```

{
  "type": "wokwi-
  pushbutton", "id": "btn1",
  "top": 92.6,
  "left": 57.6,
  "attrs": { "color": "green", "xray": "1" }
},
{
  "type": "wokwi-led",
  "id": "led1",
  "top": 169.2,
  "left": -53.8,
  "attrs": { "color": "red" }
}
],
"connections": [
  [ "esp:TX", "$serialMonitor:RX", "", [] ],
  [ "esp:RX", "$serialMonitor:TX", "", [] ],
  [ "relay1:VCC", "esp:3V3", "red", [ "h-278.4", "v67.2" ] ],
  [ "relay1:GND", "esp:GND.2", "black", [ "h-182.4", "v57.2" ] ],
  [ "relay1:IN", "esp:23", "green", [ "h-153.6", "v57.4" ] ],
  [ "btn1:1.1", "esp:19", "green", [ "h-163.2", "v38.4" ] ],
  [ "btn1:2.1", "esp:GND.2", "green", [ "h-48", "v-86.2", "h0" ] ],
  [ "led1:A", "esp:18", "green", [ "v0", "h19.2", "v-96" ] ],
  [ "led1:C", "esp:GND.2", "green", [ "v0", "h-28.4", "v-172.8" ] ]
],
"dependencies": {}
}

```



JARAK

```
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): ");

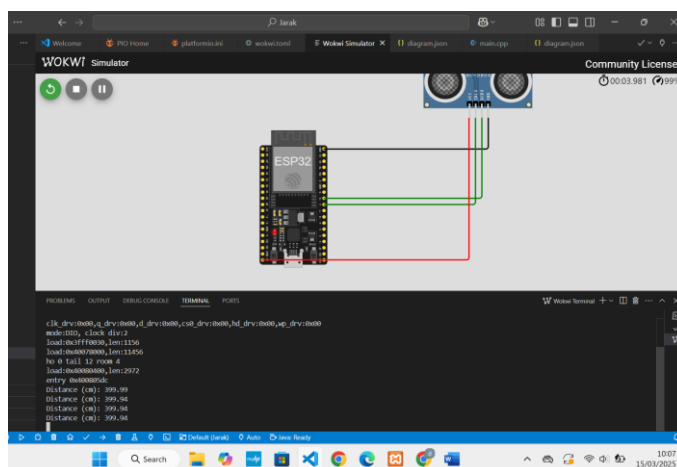
Serial.println(distanceCm);

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

// Serial.println(distanceInch);

delay(1000);
}

```



```
#include <Arduino.h>
```

```
const int trigPin = 5;
```

```
const int echoPin = 18;
```

```
#define SOUND_SPEED 0.034
```

```
#define CM_TO_INCH 0.393701
```

```
long duration;
```

```
float distanceCm;
```

```
float distanceInch;
```

```

void setup() {
    Serial.begin(115200); // Memulai komunikasi serial
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop() {
    // Membersihkan trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Mengaktifkan trigPin selama 10 microseconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Membaca echoPin, mendapatkan waktu perjalanan gelombang suara dalam microseconds
    duration = pulseIn(echoPin, HIGH);

    // Menghitung jarak dalam cm
    distanceCm = duration * SOUND_SPEED / 2;

    // Menghitung jarak dalam inci
    distanceInch = distanceCm * CM_TO_INCH;

    // Menampilkan hasil ke Serial Monitor
    Serial.print("Distance (cm): ");
    Serial.println(distanceCm);

    delay(1000);
}

```