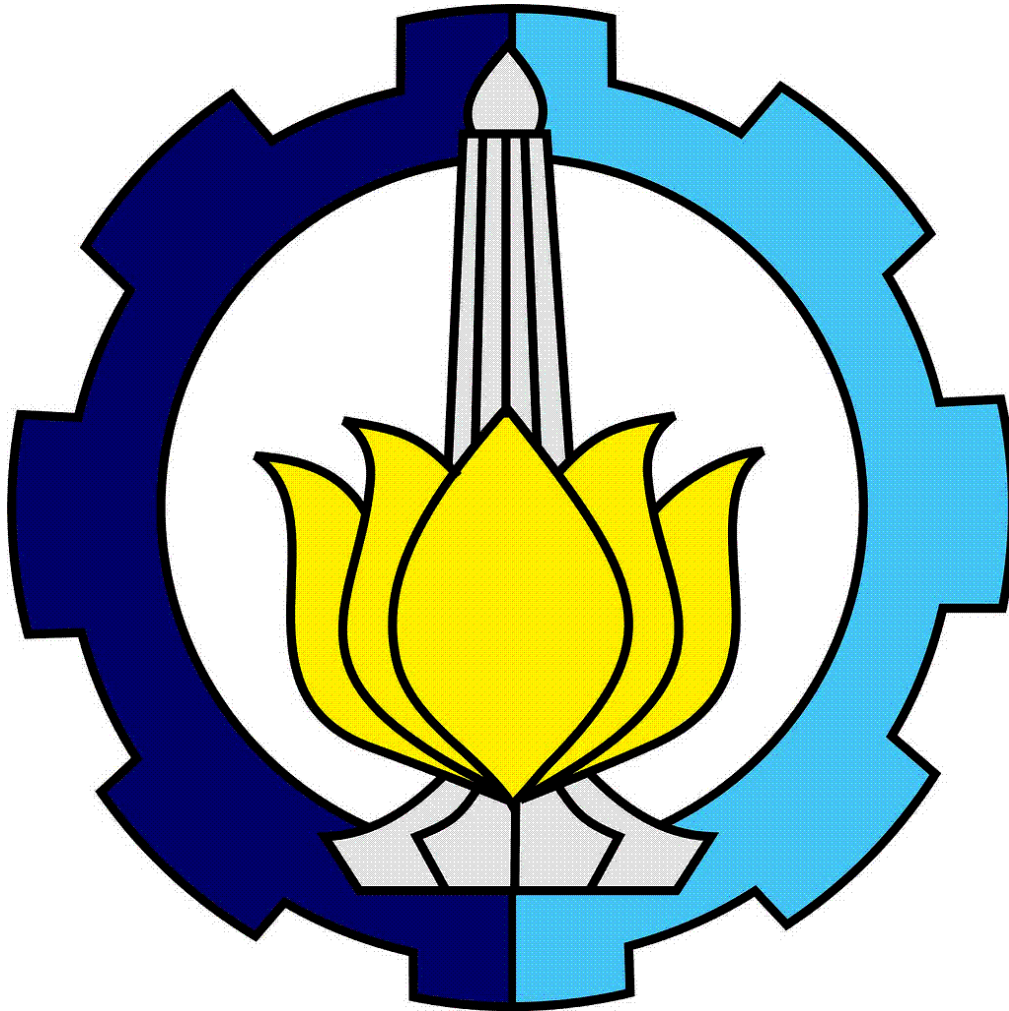


**LAPORAN UTS-2
INTERNET OF THINGS**



**APLIKASI SEDERHANA BUZZER, LED, OLED, DHT11, SENSOR ULTRASONIC,
SERVO, dan DC MOTOR MENGGUNAKAN ESP-32**

Anggota Kelompok :

Fransiskus Benyamin Sitompul (5027211021)

Muhamad Ilham Yumna (5027211024)

Wisnu Adjie Saka (5027211051)

Penjelasan Project

Pada kesempatan kali ini kami membuat laporan IoT untuk aplikasi sederhana yang menggunakan ESP-32 dengan berbagai komponen seperti buzzer, LED, OLED, DHT11, sensor ultrasonik, servo, dan DC motor, dan berhasil mengoperasikannya melalui aplikasi Blynk pada perangkat mobile. Tujuan kami pada project kali ini adalah dapat membuat aplikasi sederhana buzzer, led, oled, dht11, sensor ultrasonic, servo, dc motor menggunakan ESP-32 dan melakukan connection terhadap WIFI. Berikut adalah alat-alat yang kita gunakan pada project kali ini :

- **ESP-32**, merupakan sebuah mikrokontroler yang dirancang untuk aplikasi Internet of Things (IoT).
- **Sensor Ultrasonic**, merupakan sensor yang digunakan untuk mengukur jarak antara sensor dan suatu objek menggunakan gelombang suara ultrasonik.
- **LED**, merupakan sebuah komponen elektronik yang dapat mengeluarkan cahaya ketika diberi arus listrik.
- **OLED**, merupakan sebuah jenis teknologi layar elektronik yang menggunakan bahan organik yang dapat menghasilkan cahaya ketika diberi arus listrik.
- **DHT11**, merupakan sensor suhu dan kelembaban yang sering digunakan pada proyek DIY dan IoT.
- **Servo**, merupakan sebuah motor listrik yang dilengkapi dengan sistem kontrol yang presisi untuk menggerakkan suatu objek dalam sudut tertentu.
- **Dc Motor dan L293D**, dc motor adalah sebuah jenis motor listrik yang menggunakan arus searah (DC) untuk menghasilkan gerakan pada porosnya, sedangkan L293D adalah sebuah IC (Integrated Circuit) yang berfungsi sebagai driver motor DC. Dalam penggunaannya, L293D dihubungkan antara mikrokontroler dan motor DC. Mikrokontroler mengirimkan sinyal kontrol ke L293D untuk mengontrol arah putaran dan kecepatan motor.

Berikut merupakan code dari aplikasi yang telah kelompok kami buat :

```
#include <DHT.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Adafruit_SSD1306.h>
#include <Ultrasonic.h>
#include <ESP32Servo.h>

#define BLYNK_TEMPLATE_ID "TMPL6HhXzV39h"
```

```
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "ez0R7JDb6eCo20Xb3UYQn3Kqd4DmLV6K"

#define LED1_PIN 32
#define LED2_PIN 33
#define LED3_PIN 25
#define LED4_PIN 26
#define LED5_PIN 27
#define LED6_PIN 14
#define LED7_PIN 12
#define LED8_PIN 13

#define DHT_PIN      23
#define DHT_TYPE     DHT11

#define TRIGGER_PIN  17
#define ECHO_PIN     5

#define OLED_ADDRESS 0x3C
#define OLED_SDA 21
#define OLED_SCL 22

#define BUZZER 4

#define enA 19
#define in1 1
#define in2 3

#define SERVOPIN 2
int servoPosition = 90;

Adafruit_SSD1306 display(128, 64, &Wire, OLED_ADDRESS);
Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN, 30000UL);
DHT dht(DHT_PIN, DHT_TYPE);
Servo servo;
BlynkTimer timer;

char ssid[] = "Departemen Tanpa Internet";
char pass[] = "fqkj4876";

float distance;
float temperature;
float humidity;
```

```

int speed;

void updateLEDs(int ledCount) {
    digitalWrite(LED1_PIN, ledCount >= 1 ? HIGH : LOW);
    digitalWrite(LED2_PIN, ledCount >= 2 ? HIGH : LOW);
    digitalWrite(LED3_PIN, ledCount >= 3 ? HIGH : LOW);
    digitalWrite(LED4_PIN, ledCount >= 4 ? HIGH : LOW);
    digitalWrite(LED5_PIN, ledCount >= 5 ? HIGH : LOW);
    digitalWrite(LED6_PIN, ledCount >= 6 ? HIGH : LOW);
    digitalWrite(LED7_PIN, ledCount >= 7 ? HIGH : LOW);
    digitalWrite(LED8_PIN, ledCount >= 8 ? HIGH : LOW);
}

BLYNK_WRITE(V0) {
    int ledCount = param.asInt();
    updateLEDs(ledCount);
}

BLYNK_WRITE(V4) {
    // set the servo position based on the value of virtual pin V4
    servoPosition = map(param.asInt(), 0, 1023, 0, 180);
    servo.write(servoPosition);
}

BLYNK_WRITE(V5) { // This function will be called when Button Widget 1
in Blynk app is pressed
    int buttonState = param.asInt(); // Get the state of the button (HIGH
or LOW)

    if (buttonState == HIGH) {
        // Play tone 1
        tone(BUZZER, 523);
    } else {
        noTone(BUZZER);
    }
}

BLYNK_WRITE(V6) { // This function will be called when Button Widget 2
in Blynk app is pressed
    int buttonState = param.asInt(); // Get the state of the button (HIGH
or LOW)

    if (buttonState == HIGH) {

```

```

    // Play tone 2
    tone(BUZZER, 784);
} else {
    noTone(BUZZER);
}
}

BLYNK_WRITE(V7) { // This function will be called when Button Widget 3
in Blynk app is pressed
    int buttonState = param.asInt(); // Get the state of the button (HIGH
or LOW)

    if (buttonState == HIGH) {
        // Play tone 3
        tone(BUZZER, 440);
    } else {
        noTone(BUZZER);
    }
}

BLYNK_WRITE(V8) { // This function will be called when Button Widget 4
in Blynk app is pressed
    int buttonState = param.asInt(); // Get the state of the button (HIGH
or LOW)

    if (buttonState == HIGH) {
        // Play tone 4
        tone(BUZZER, 698);
    } else {
        noTone(BUZZER);
    }
}

BLYNK_WRITE(V9) {
    speed = param.asInt();
    analogWrite(enA, speed);

    if (speed > 0) {
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
    } else if (speed < 0) {
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
    }
}

```

```
    } else {  
        digitalWrite(in1, LOW);  
        digitalWrite(in2, LOW);  
    }  
    Serial.println(speed);  
}  
  
void sendDistance()  
{  
    distance = ultrasonic.read();  
    Blynk.virtualWrite(V1, distance);  
}  
  
void sendTempHumid()  
{  
    temperature = dht.readTemperature();  
    humidity = dht.readHumidity();  
    if (isnan(temperature) || isnan(humidity)) {  
        Serial.println("Failed to read from DHT sensor!");  
        return;  
    }  
    Blynk.virtualWrite(V2, temperature);  
    Blynk.virtualWrite(V3, humidity);  
}  
  
void updateDisplay()  
{  
    display.clearDisplay();  
    display.setCursor(0,0);  
    display.setTextSize(1);  
    display.print("Distance: ");  
    display.print(distance);  
    display.print(" cm");  
    display.setCursor(0,8);  
    display.setTextSize(1);  
    display.print("Temp: ");  
    display.print(temperature);  
    display.print(" C");  
    display.setCursor(0,16);  
    display.print("Humid: ");  
    display.print(humidity);  
    display.print(" %");  
    display.display();  
}
```

```

    display.setCursor(0,24);
    display.setTextSize(1);
    display.print("Motor speed: ");
    display.print(speed);
}

void setup()
{
    Serial.begin(115200);
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    dht.begin();
    display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS);
    display.clearDisplay();
    display.display();
    display.setText_color(WHITE);
    timer.setInterval(2000L, sendTempHumid);
    timer.setInterval(1000L, sendDistance);
    timer.setInterval(1000L, updateDisplay);

    pinMode(LED1_PIN, OUTPUT);
    pinMode(LED2_PIN, OUTPUT);
    pinMode(LED3_PIN, OUTPUT);
    pinMode(LED4_PIN, OUTPUT);
    pinMode(LED5_PIN, OUTPUT);
    pinMode(LED6_PIN, OUTPUT);
    pinMode(LED7_PIN, OUTPUT);
    pinMode(LED8_PIN, OUTPUT);

    servo.attach(2);

    pinMode(BUZZER, OUTPUT);

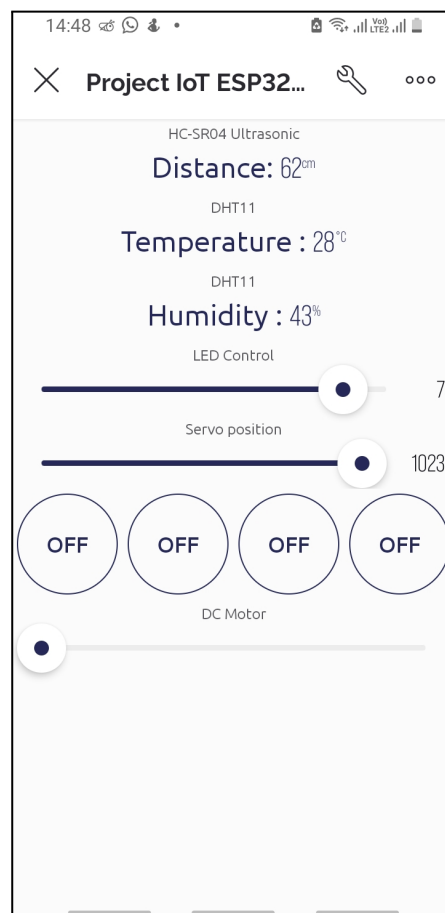
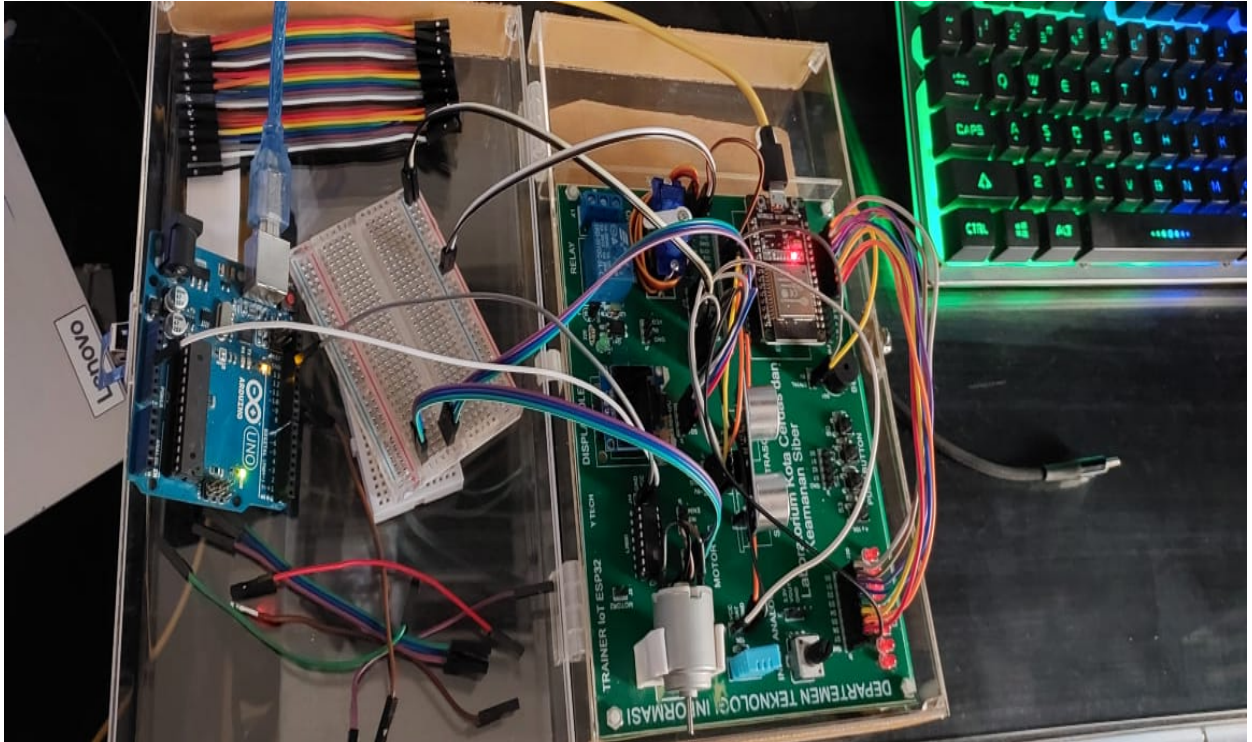
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
}

void loop()
{
    Blynk.run();
    timer.run();
}

```

Dokumentasi

Berikut adalah dokumentasi alat dan aplikasi IOT kelompok kami :



Dan terakhir merupakan output simulasi dari aplikasi IOT yang telah kelompok kami buat dalam bentuk video :

https://drive.google.com/file/d/1XdGhc1IFE6l_Ts7tFlnB40dp6_DV2wlq/view?usp=drivesdk