

**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

……   ……

BÁO CÁO BÀI TẬP LỚN

LẬP TRÌNH NHÚNG ĐIỀU KHIỂN TỪ XA VỚI ESP8266

Giáo viên giảng dạy: Lê Hữu Vinh

Sinh viên thực hiện:

1. Cao Hoài An **0306221099**

2. Phan Thành Thảo **0306221481**

LỚP: CĐ TH 22DĐE

KHÓA: 2022 – 2025

TP. HCM, tháng 12 năm 2024

**MỤC LỤC**

[1) Giới thiệu bài tập lớn 1](#_Toc184216618)

[2) Chuẩn bị phần cứng 1](#_Toc184216619)

[3) Sơ đồ mạch 4](#_Toc184216620)

[4) Quy trình xử lý 5](#_Toc184216621)

[5) Source code, giao diện điều khiển và kết quả 5](#_Toc184216622)

[6) Kết Luận 9](#_Toc184216623)

[7) Tài Liệu Tham Khảo 9](#_Toc184216624)

# Giới thiệu bài tập lớn

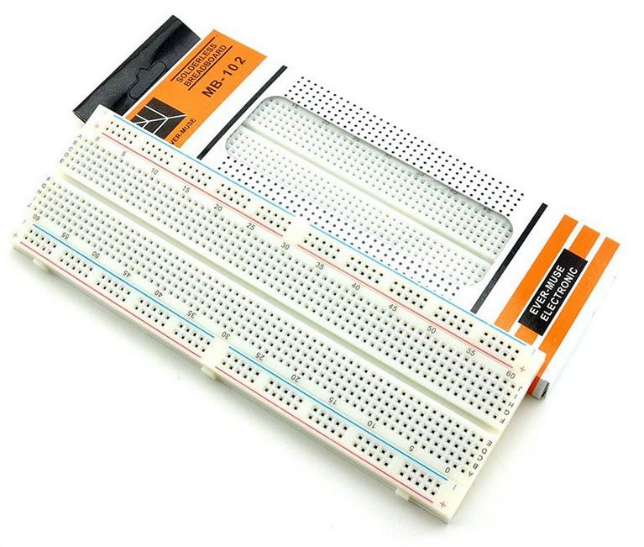
Ngày nay, với sự phát triển mạnh mẽ của công nghệ IoT (Internet of Things), việc ứng dụng các thiết bị kết nối mạng vào đời sống trở nên ngày càng phổ biến, trong đó module WiFi ESP8266 đóng vai trò quan trọng nhờ tính năng mạnh mẽ và chi phí thấp. Kết hợp với Flutter – một framework phát triển ứng dụng đa nền tảng hiện đại, bài tập lớn này hướng tới việc xây dựng một hệ thống điều khiển từ xa cho các thiết bị thông qua mạng WiFi. Cụ thể, ứng dụng Flutter sẽ cung cấp giao diện thân thiện để người dùng điều chỉnh tốc độ chớp tắt của LED thứ nhất, thay đổi độ sáng của LED thứ hai và bật hoặc tắt loa. Trên ESP8266, hệ thống sẽ được lập trình để nhận tín hiệu điều khiển từ ứng dụng và tương tác trực tiếp với các thiết bị phần cứng. Mục tiêu của bài tập là xây dựng một hệ thống giao tiếp ổn định giữa ứng dụng và module ESP8266, đồng thời giúp sinh viên nâng cao kỹ năng lập trình phần mềm và phần cứng, mở ra những ứng dụng IoT thực tiễn trong tương lai.

# Chuẩn bị phần cứng

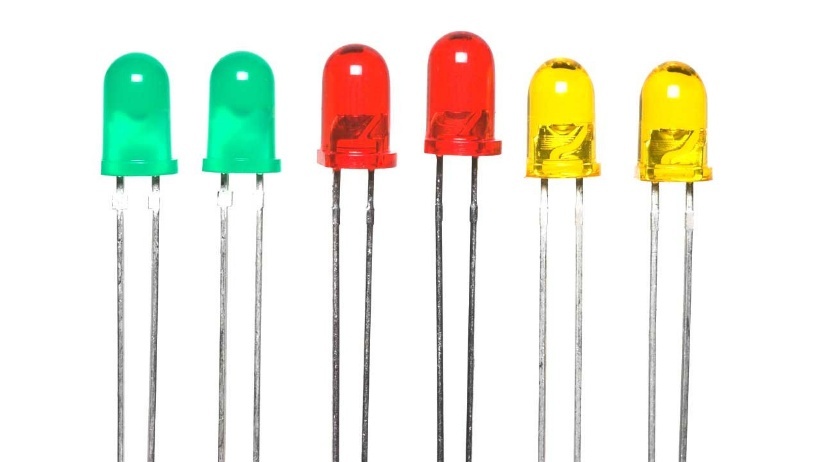
Để thực hiện kết nối chúng ta cần chuẩn bị các phần cứng sau:



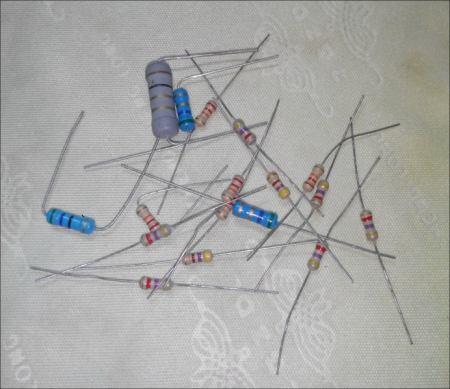
***Hình 2.1. Dây điện***



***Hình 2.2. Bread board***



***Hình 2.3. Đèn led***



***Hình 2.4. Điện trở***

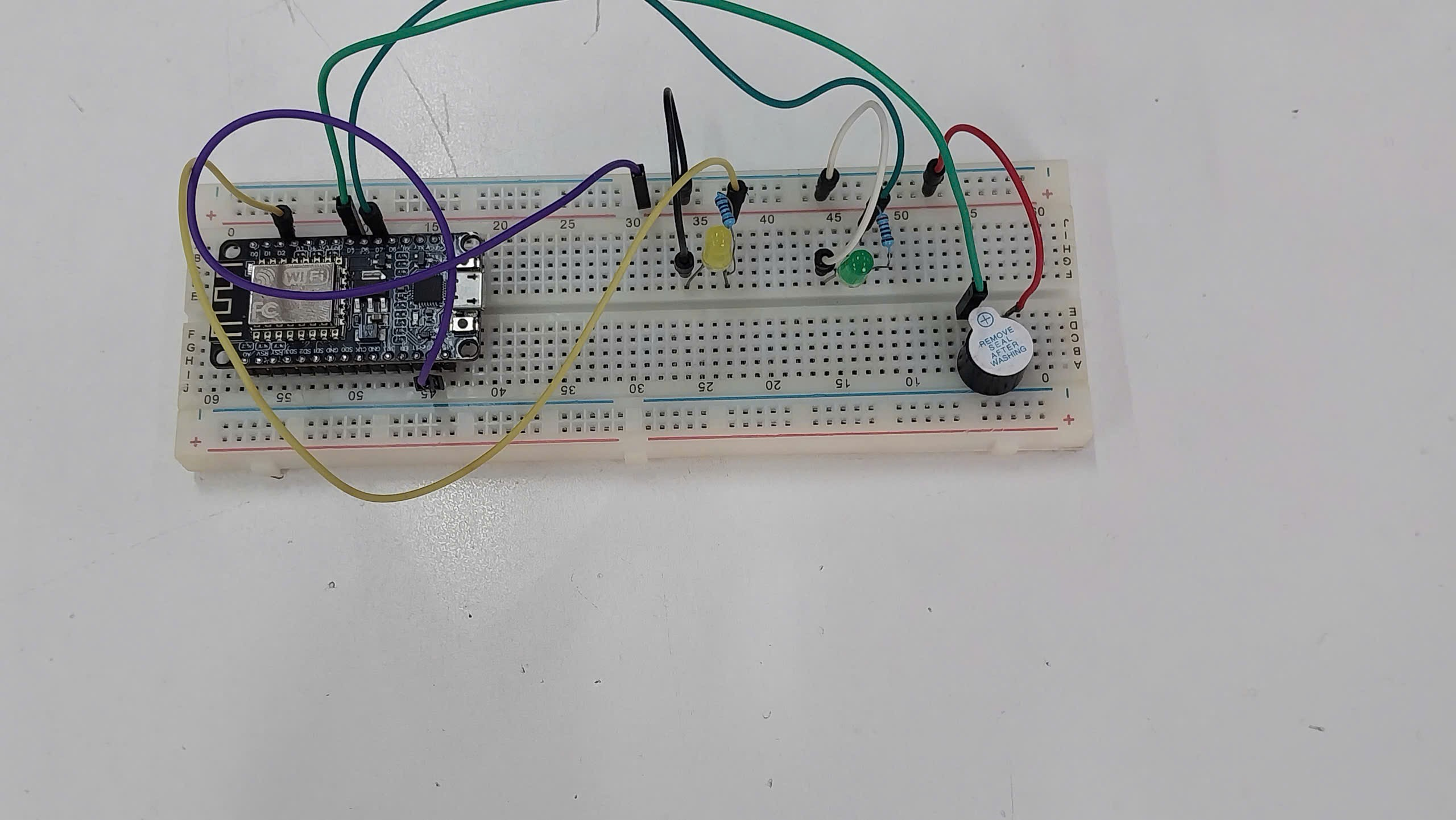
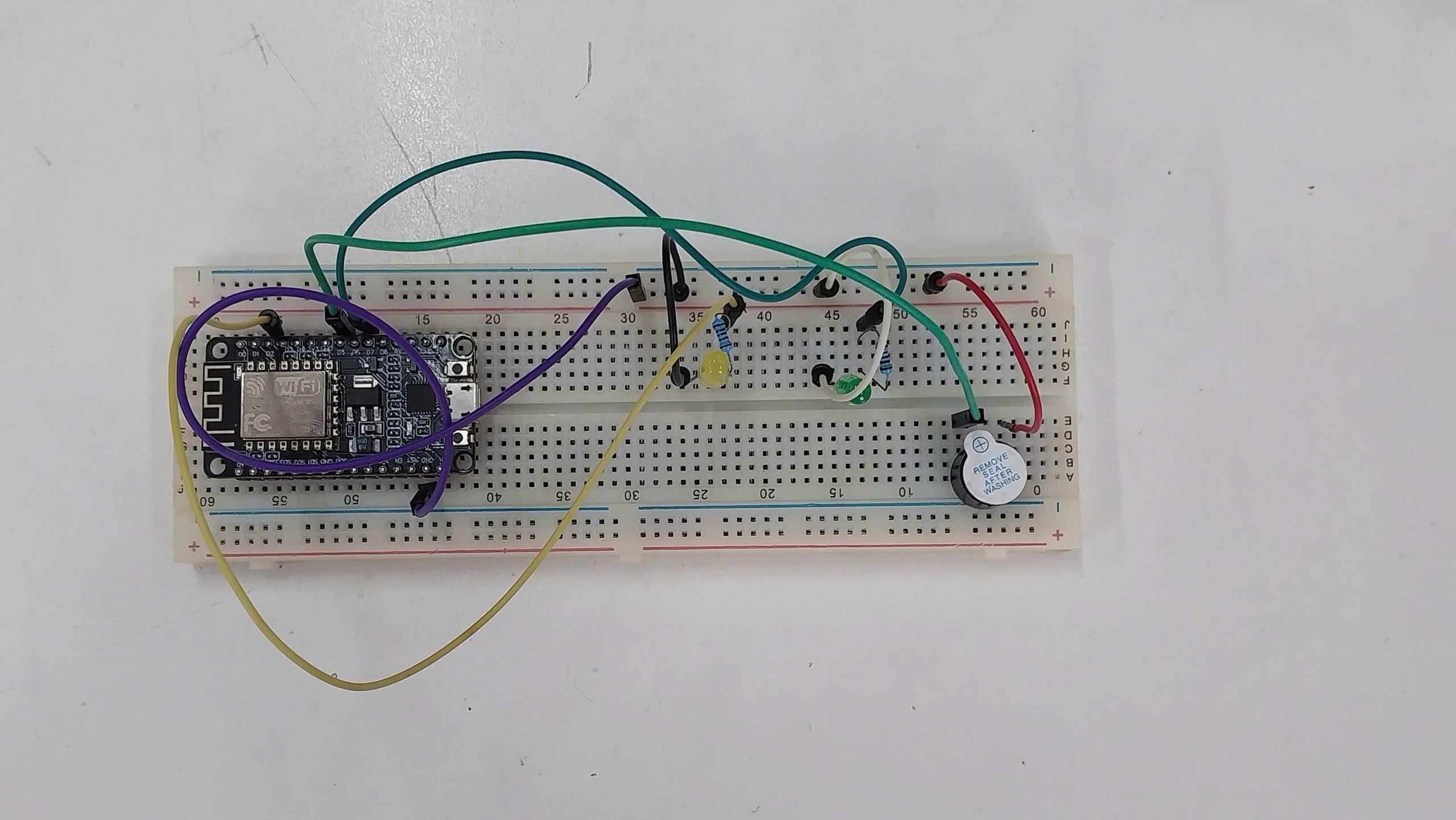


***Hình 2.5. Buzzer (Loa)***

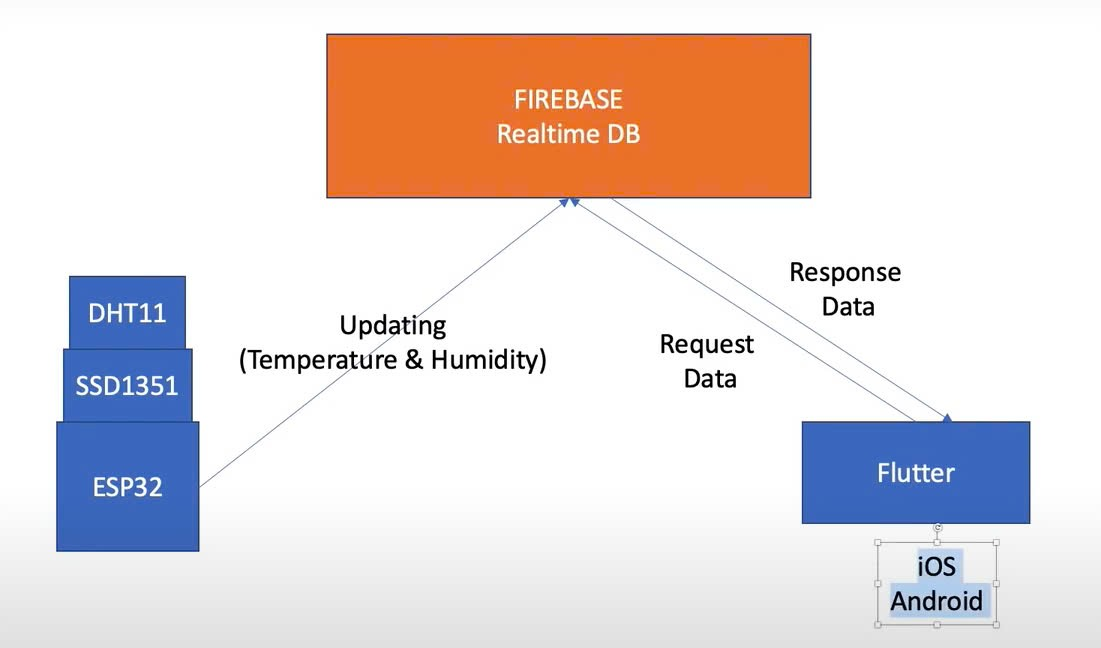


***Hình 2.6. ESP8266***

# Sơ đồ mạch

# Quy trình xử lý



***Hình 4.1 Quy trình xử lý gửi nhận trên firebase***

# Source code, giao diện điều khiển và kết quả

1. **Source code**

#include <ESP8266WiFi.h>

#include <FirebaseESP8266.h>

// Cấu hình WiFi

#define WIFI\_SSID "Galaxy A23 1CFF"

#define WIFI\_PASSWORD "hoaian0123"

// Cấu hình Firebase

#define FIREBASE\_HOST "esp8266-7280d-default-rtdb.asia-southeast1.firebasedatabase.app"

#define FIREBASE\_AUTH "42NYpgaX0hdhBdRd94pxhyNK42mr9e4m7xxtZ01L"

// Chọn chân LED để điều chỉnh độ sáng (D2 là chân hỗ trợ PWM)

#define LED\_PIN\_1 4

#define BUZZER\_PIN 14 // Chân GPIO14 (D5) dùng cho loa

#define LED\_PIN\_2 13

int brightness = 0;

int blinkSpeed = 500; // Thời gian chớp LED mặc định là 500ms (tốc độ)

FirebaseData firebaseData;

FirebaseConfig firebaseConfig;

FirebaseAuth firebaseAuth;

void setup()

{

    Serial.begin(115200);

    // Kết nối WiFi

    WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

    Serial.print("Đang kết nối WiFi...");

    while (WiFi.status() != WL\_CONNECTED)

    {

        delay(1000);

        Serial.print(".");

    }

    Serial.println("\nKết nối WiFi thành công!");

    // Cấu hình Firebase

    firebaseConfig.host = FIREBASE\_HOST;

    firebaseConfig.signer.tokens.legacy\_token = FIREBASE\_AUTH;

    // Bắt đầu Firebase

    Firebase.begin(&firebaseConfig, &firebaseAuth);

    Firebase.reconnectWiFi(true);

    // Khởi tạo LED và Buzzer

    pinMode(LED\_PIN\_1, OUTPUT);  // Chân LED BUILTIN này hiện là led tích hợp của esp nên nó bị đảo ngược

    pinMode(LED\_PIN\_2, OUTPUT);  // Chân LED BUILTIN này hiện là led tích hợp của esp nên nó bị đảo ngược

    pinMode(BUZZER\_PIN, OUTPUT); // Chân Buzzer

}

void loop()

{

    // Đọc trạng thái LED từ Firebase

    if (Firebase.getInt(firebaseData, "/led/status"))

    {

        int ledStatus = firebaseData.intData();

        // Nếu LED được bật, điều chỉnh độ sáng LED

        if (ledStatus == 1)

        {

            if (Firebase.getInt(firebaseData, "/led/brightness"))

            {

                brightness = firebaseData.intData();

                brightness = constrain(brightness, 0, 255); // Giới hạn độ sáng từ 0-255

                analogWrite(LED\_PIN\_2, brightness);         // Điều chỉnh độ sáng LED

                Serial.print("Độ sáng hiện tại: ");

                Serial.println(brightness);

            }

            else

            {

                Serial.println("Không thể đọc độ sáng: " + firebaseData.errorReason());

            }

            // Đọc tốc độ chớp LED từ Firebase

            if (Firebase.getInt(firebaseData, "/led/blink\_speed"))

            {

                blinkSpeed = firebaseData.intData();

                Serial.print("Tốc độ chớp LED: ");

                Serial.println(blinkSpeed);

            }

            else

            {

                Serial.println("Không thể đọc tốc độ chớp: " + firebaseData.errorReason());

            }

            // Chớp LED với tốc độ được điều chỉnh

            digitalWrite(LED\_PIN\_1, HIGH); // Bật LED

            delay(blinkSpeed);             // Chờ thời gian chớp

            digitalWrite(LED\_PIN\_1, LOW);  // Tắt LED

            delay(blinkSpeed);             // Chờ thời gian chớp

        }

        else

        {

            digitalWrite(LED\_PIN\_1, LOW); // Tắt LED nếu trạng thái là tắt

            digitalWrite(LED\_PIN\_2, LOW); // Tắt LED nếu trạng thái là tắt

            Serial.println("LED đã tắt.");

        }

    }

    else

    {

        Serial.println("Không thể đọc trạng thái LED: " + firebaseData.errorReason());

    }

    // Đọc trạng thái Loa (Buzzer) từ Firebase

    if (Firebase.getInt(firebaseData, "/buzzer/status"))

    {

        int buzzerStatus = firebaseData.intData();

        // Bật loa nếu trạng thái là 1, tắt nếu trạng thái là 0

        if (buzzerStatus == 1)

        {

            digitalWrite(BUZZER\_PIN, HIGH); // Bật loa

            Serial.println("Loa đã bật.");

        }

        else

        {

            digitalWrite(BUZZER\_PIN, LOW); // Tắt loa

            Serial.println("Loa đã tắt.");

        }

    }

    else

    {

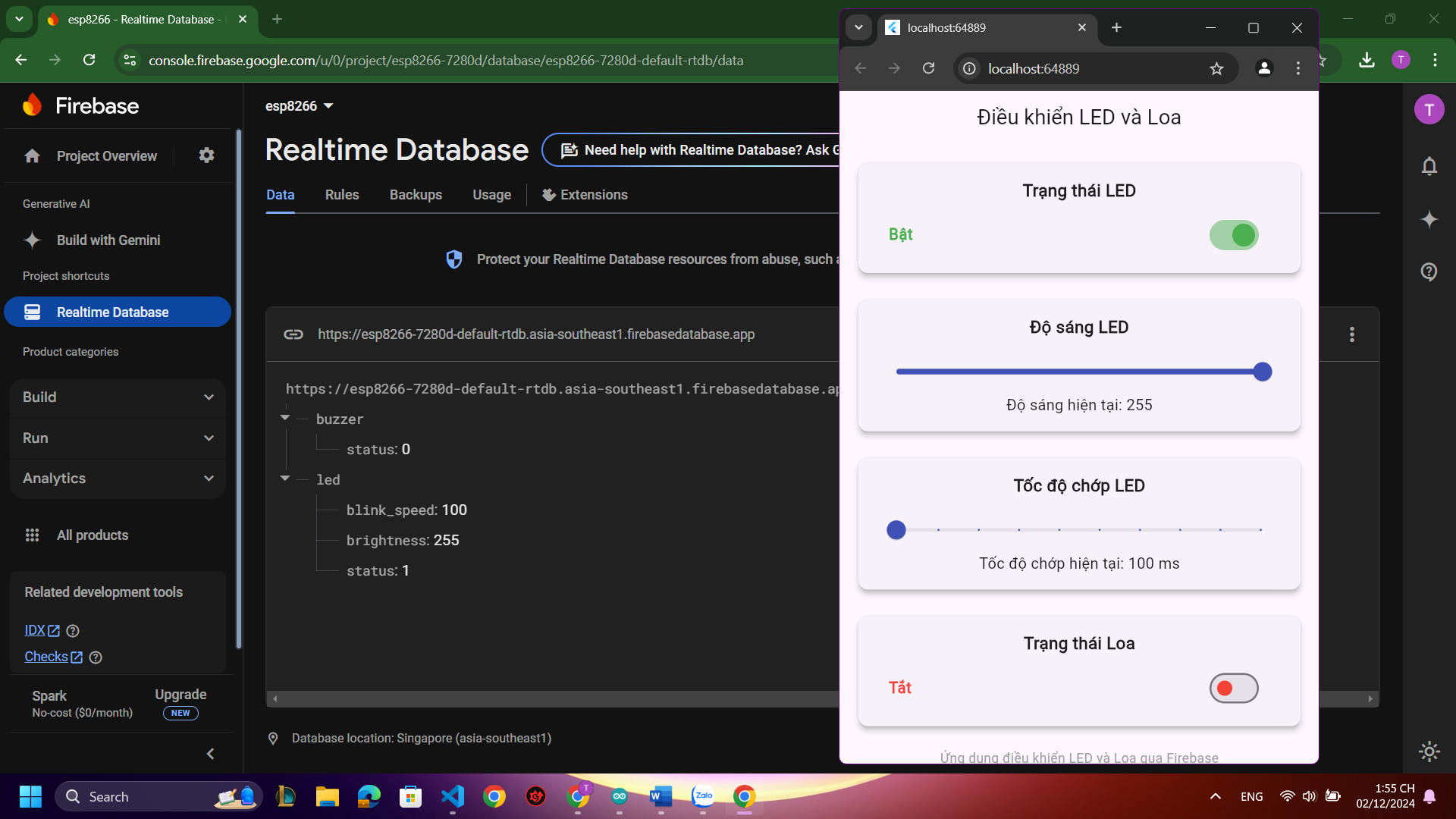
        Serial.println("Không thể đọc trạng thái Loa: " + firebaseData.errorReason());

    }

    delay(1000); // Giảm tải Firebase

}

1. **Giao diện điều khiển**

****

1. **Kết quả**

Dưới đây là video ngắn về kết quả của nhóm làm được.

[](result.mp4)

# Kết Luận

Tóm lại, với việc kết hợp Flutter và ESP8266, bài tập lớn này không chỉ mang lại kiến thức thực tế về lập trình ứng dụng và hệ thống nhúng mà còn mở ra hướng tiếp cận các giải pháp IoT hiện đại. Hệ thống điều khiển tốc độ chớp LED, độ sáng LED và bật/tắt loa được xây dựng trong bài tập là một minh chứng cụ thể cho sự tích hợp hiệu quả giữa phần mềm và phần cứng, đồng thời giúp sinh viên làm quen với các công nghệ tiên tiến, chuẩn bị tốt hơn cho những thách thức trong lĩnh vực công nghệ thông tin và tự động hóa.

# Tài Liệu Tham Khảo

Youtube: <https://www.youtube.com/watch?v=nVrACWPXi8g>

Youtube: <https://youtu.be/HHKmQC8CbIQ?si=f-F8kjT0Z6I9cUw_>