

**TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA CÔNG NGHỆ THÔNG TIN**

**SỐ PHÁCH: .....**

**TÊN ĐỀ TÀI TIỂU LUẬN**  
**Tạo hệ thống định vị trong nhà với ESP32 và BLE**

**TÊN LỚP HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG IOT**

**MÃ HỌC PHẦN: TIN4024**

**GIẢNG VIÊN HƯỚNG DẪN: VÕ VIỆT DŨNG**

**HUẾ, THÁNG 04 NĂM 2025**

**TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA CÔNG NGHỆ THÔNG TIN**

**TÊN ĐỀ TÀI TIỂU LUẬN**

**Tạo hệ thống định vị trong nhà với ESP32 và BLE**

**TÊN LỚP HỌC PHẦN : PHÁT TRIỂN ỨNG DỤNG IOT**

**MÃ HỌC PHẦN: TIN4024**

**Giảng viên hướng dẫn :    VÕ VIỆT DŨNG**

**HUẾ, THÁNG 04 NĂM 2025**

**Danh mục từ viết tắt**

Kí hiệu	Tiếng Anh	Tiếng Việt
ESP32	Espressif Systems 32	Là một vi điều khiển phát triển bởi Espressif
BLE	Bluetooth Low Energy	Phiên bản tiết kiệm năng lượng của công nghệ Bluetooth
RSSI	Received Signal Strength Indicator	Chỉ số cường độ tín hiệu
UUID	Universally Unique Identifier	Định danh duy nhất toàn cục
MAC	Media Access Control Address	Địa chỉ điều khiển truy cập phương tiện
GPS	Global Positioning System	Hệ thống định vị toàn cầu

## MỤC LỤC

LỜI MỞ ĐẦU .....	1
NỘI DUNG .....	2
<b>CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG ĐỊNH VỊ TRONG NHÀ .....</b>	<b>2</b>
<b>I. Giới thiệu chung về hệ thống định vị trong nhà.....</b>	<b>2</b>
<b>II. Giới thiệu các ESP32 và Bluetooth Low Energy .....</b>	<b>2</b>
1.Giới thiệu về ESP32 và BLE.....	2
2. Nguyên lý hoạt động của BLE.....	4
<b>CHƯƠNG 2 THIẾT KẾ HỆ THỐNG .....</b>	<b>6</b>
<b>I. Ý tưởng .....</b>	<b>6</b>
<b>II. Nguyên lý hoạt động của hệ thống.....</b>	<b>6</b>
<b>Chương 3. Triển khai và kiểm thử .....</b>	<b>7</b>
<b>I.Môi trường kiểm thử và mô phỏng .....</b>	<b>7</b>
<b>II. Các hàm sinh dữ liệu ngẫu nhiên sử dụng .....</b>	<b>7</b>
Kết luận .....	12

## LỜI MỞ ĐẦU

Trong các ứng dụng hiện đại, việc xác định vị trí của thiết bị trong không gian đóng vai trò quan trọng, đặc biệt là trong các hệ thống quản lý tài sản, hệ thống an ninh và nhà thông minh. Với sự phát triển của công nghệ Bluetooth Low Energy (BLE), các giải pháp định vị trong nhà ngày càng trở nên phổ biến, đặc biệt trong các không gian nhỏ và môi trường phức tạp như trong các tòa nhà, kho bãi hay các khu vực công cộng. ESP32, một vi điều khiển với khả năng hỗ trợ BLE, là một lựa chọn lý tưởng để xây dựng các hệ thống định vị trong nhà.

Trong quá trình phát triển hệ thống định vị trong nhà với ESP32 và BLE, do không có đầy đủ các thiết bị thật và BLE không hỗ trợ để chạy mô phỏng, em đã quyết định sử dụng dữ liệu ngẫu nhiên thay thế cho các giá trị thực tế của BLE trong quá trình quét.

Cụ thể, em đã giả lập quá trình quét các thiết bị BLE bằng cách tạo ra các giá trị ngẫu nhiên cho các thông số như:

- Địa chỉ MAC của các thiết bị.
- Tên thiết bị.
- Cường độ tín hiệu (RSSI) – lấy khoảng cách.
- Kinh độ và vĩ độ của thiết bị (có thể dùng GPS trong các thiết bị thật).
- Cách thức quét các thiết bị.

Mặc dù không thể phản ánh chính xác hành vi của BLE trong môi trường thực tế, việc này giúp em kiểm tra được các thuật toán xử lý dữ liệu, hiển thị thông tin trên giao diện người dùng, cũng như các logic liên quan đến việc quét và nhận diện thiết bị.

# NỘI DUNG

## CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG ĐỊNH VỊ TRONG NHÀ

### I. Giới thiệu chung về hệ thống định vị trong nhà

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu theo dõi và xác định vị trí của thiết bị hoặc đối tượng trong không gian ngày càng trở nên phổ biến. Hệ thống định vị trong nhà (Indoor Positioning System – IPS) đóng vai trò quan trọng trong việc quản lý tài sản, giám sát người dùng, và nâng cao hiệu quả vận hành trong các tòa nhà, kho bãi, văn phòng, bệnh viện, và nhà thông minh.

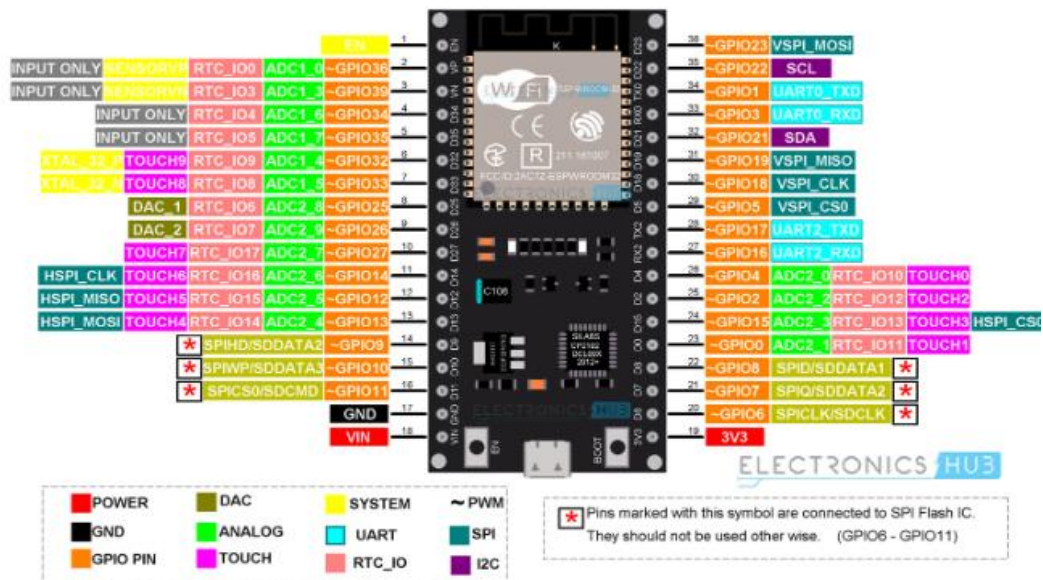
Khác với hệ thống định vị ngoài trời như GPS, hệ thống định vị trong nhà cần dựa vào các công nghệ như BLE, Wi-Fi, RFID hoặc UWB để hoạt động hiệu quả trong môi trường kín. Trong đó, Bluetooth Low Energy (BLE) là một giải pháp phổ biến và tiết kiệm năng lượng, đặc biệt phù hợp cho các ứng dụng trong không gian nhỏ với chi phí thấp.

### II. Giới thiệu các ESP32 và Bluetooth Low Energy

#### 1. Giới thiệu về ESP32 và BLE

##### ESP32:

ESP32 là một vi điều khiển mạnh mẽ của Espressif Systems, tích hợp sẵn khả năng kết nối Wi-Fi và BLE. ESP32 có thể được sử dụng để phát triển các ứng dụng trong nhiều lĩnh vực, bao gồm định vị trong nhà, vì nó có khả năng quét các tín hiệu BLE và xử lý dữ liệu để xác định vị trí của các thiết bị trong không gian.



Hình 1. Sơ đồ chân của vi điều khiển ESP32

## Đặc điểm kỹ thuật của ESP32

ESP32 có nhiều phiên bản khác nhau, nhưng phổ biến nhất là ESP32 DevKit V1. Một số thông số quan trọng:

- Bộ xử lý: Dual-core Xtensa LX6, tốc độ lên đến 240 MHz
- Bộ nhớ RAM: 520 KB
- Bộ nhớ Flash: 4MB (tùy phiên bản)
- Kết nối không dây: WiFi 802.11 b/g/n, Bluetooth 4.2/BLE
- Số chân GPIO: Lên đến 36 GPIO (có thể lập trình đa chức năng)
- Giao tiếp ngoại vi: SPI, I2C, UART, ADC, DAC, PWM
- Nguồn điện hoạt động: 3.3V

## BLE:

BLE (Bluetooth Low Energy) là một công nghệ không dây tiết kiệm năng lượng, thuộc chuẩn Bluetooth 4.0 trở lên, được thiết kế cho các ứng dụng IoT và thiết bị đeo (wearable) cần hoạt động lâu dài với pin nhỏ. BLE được tích hợp trên ESP32 và dễ dàng sử dụng qua thư viện BLEDevice.

## 2. Nguyên lý hoạt động của BLE

BLE hoạt động bằng cách sử dụng tần số 2,4 GHz để truyền thông tin giữa các thiết bị. Nó sử dụng một giao thức kết nối hai bước để thiết lập kết nối giữa hai thiết bị BLE: Giao thức kết nối và Giao thức giao tiếp.

Giao thức kết nối là quá trình mà hai thiết bị BLE trao đổi các thông tin để thiết lập kết nối giữa chúng. Giao thức này bao gồm hai giai đoạn: quảng bá và quá trình kết nối.

Trong giai đoạn quảng bá, thiết bị BLE sẽ gửi các thông điệp quảng bá đến các thiết bị khác trong vùng phủ sóng của nó. Các thông điệp này chứa các thông tin cơ bản về thiết bị như tên, địa chỉ MAC và loại thiết bị. Các thiết bị khác trong vùng phủ sóng có thể nhận thông điệp này và quyết định liệu chúng có muốn kết nối với thiết bị gửi thông điệp hay không.

Trong giai đoạn kết nối, khi một thiết bị nhận được yêu cầu kết nối từ một thiết bị khác, nó sẽ phản hồi bằng một thông điệp kết nối và thiết lập kết nối giữa hai thiết bị. Sau khi kết nối được thiết lập, hai thiết bị sẽ chuyển sang giai đoạn giao tiếp.

Giao thức giao tiếp là quá trình mà hai thiết bị BLE trao đổi các thông tin dữ liệu. Nó sử dụng một giao thức chuyển giao dữ liệu hai chiều, cho phép hai thiết bị gửi và nhận dữ liệu đồng thời. Các thông điệp dữ liệu này có thể chứa các thông tin về trạng thái của thiết bị, dữ liệu cảm biến, âm thanh hoặc hình ảnh.

Ưu điểm của BLE:

- Tiêu thụ năng lượng thấp, phù hợp với thiết bị hoạt động lâu dài.
- Độ tin cậy cao
- Giá thành rẻ, tích hợp sẵn trong nhiều thiết bị như điện thoại, smartwatch.
- Hỗ trợ định vị trong nhà ở quy mô nhỏ đến trung bình.

Nhược điểm của BLE:

- Độ chính xác thấp
- Tín hiệu dễ bị ảnh hưởng bởi vật cản và môi trường (nhiều).
- Dựa vào RSSI nên cần thuật toán lọc để tăng độ tin cậy.



## 2. Ứng dụng của ESP32 và BLE trong hệ thống định vị trong nhà

ESP32 có thể quét và thu tín hiệu từ các thiết bị BLE (ví dụ như điện thoại, beacon, smartwatch). Dựa trên giá trị RSSI của từng thiết bị, hệ thống có thể:

- Ước lượng khoảng cách từ ESP32 đến thiết bị BLE.
- Xác định vị trí tương đối của thiết bị trong không gian nhỏ.
- Phát hiện khi thiết bị ra khỏi vùng cho phép → gửi cảnh báo.

Hệ thống có thể áp dụng trong:

- Giám sát thiết bị có giá trị (laptop, điện thoại) trong văn phòng.
- Định vị bệnh nhân/nhân viên trong bệnh viện, nhà máy.
- Giám sát trẻ em/người già trong nhà.

## CHƯƠNG 2 THIẾT KẾ HỆ THỐNG

### I. Ý tưởng

Xác định thiết bị BLE nằm trong vùng phủ sóng của ESP32.

Ước lượng khoảng cách thiết bị dựa trên chỉ số RSSI.

Phát hiện khi thiết bị rời khỏi vùng an toàn.

Gửi cảnh báo kịp thời qua ứng dụng Telegram đến người dùng quản lý.

Sử dụng telegram để tìm thiết bị.

### II. Nguyên lý hoạt động của hệ thống.

Trong hệ thống định vị trong nhà, ESP32 được cấu hình hoạt động ở chế độ scanner để quét và nhận diện các tín hiệu BLE từ các thiết bị xung quanh. Quá trình hoạt động của BLE trên ESP32 được tóm tắt như sau:

- Thiết bị phát (Advertiser): Các thiết bị BLE như điện thoại, smartwatch, laptop... sẽ phát ra các gói tin quảng bá (advertising packets) định kỳ, chứa thông tin định danh như địa chỉ MAC hoặc UUID (là danh sách được khai báo sẵn).
- ESP32 quét (Scanning): ESP32 liên tục quét các gói tin BLE trong vùng lân cận. Mỗi khi bắt được một gói tin, ESP32 sẽ trích xuất các thông tin như địa chỉ MAC, chỉ số cường độ tín hiệu (RSSI), tên thiết bị...
- Xác định thiết bị tin cậy: Dựa vào danh sách các thiết bị đã được định nghĩa trước, ESP32 sẽ kiểm tra xem thiết bị nào đang trong vùng phủ sóng. Nếu một thiết bị tin cậy không còn xuất hiện hoặc RSSI giảm dưới ngưỡng, hệ thống có thể kết luận rằng thiết bị đang rời khỏi khu vực an toàn.
- Gửi thông báo: Khi phát hiện thiết bị nằm trong danh sách tin cậy sẽ xác định khoảng cách, vị trí gửi lên telegram. Nếu vượt qua ngưỡng khoảng cách cho phép, ESP32 sẽ gửi cảnh báo đến người dùng qua Telegram app

## Chương 3. Triển khai và kiểm thử

### I. Môi trường kiểm thử và mô phỏng

Trong quá trình phát triển em đã sử dụng Visual Studio Code (VS Code) kết hợp với PlatformIO làm môi trường phát triển và mô phỏng:

### II. Các hàm sinh dữ liệu ngẫu nhiên sử dụng

#### 1. Danh sách thiết bị: mô phỏng tên thiết bị và địa chỉ MAC

```
std::vector<TrustedDevice> trustedDevices = {  
    {"TTL_Laptop", "AB:CD:EF:12:34:56"},  
    {"TTL_Phone", "11:22:33:44:55:66"}};
```

#### 1. Hàm mô phỏng BLE quét các thiết bị: Mô phỏng quá trình quét của BLE

```
bool isDeviceTrusted(String macAddress)  
{  
    for (const auto &device : trustedDevices)  
    {  
        if (macAddress.equalsIgnoreCase(device.mac) ||  
            macAddress.equalsIgnoreCase(device.name))  
        {  
            return true;  
        }  
    }  
    return false;  
}  
  
void scanDevices()  
{  
    if (targetDevice.length() > 0)  
    {  
        bool found = false;  
        for (const auto &device : trustedDevices)  
        {  
            if (device.name.equalsIgnoreCase(targetDevice))  
            {  
                foundDeviceName = device.name;  
                Serial.println("✔ Tìm thấy: " + foundDeviceName);  
                found = true;  
                break;  
            }  
        }  
  
        if (!found)
```

```

    {
        Serial.println("✗ Không tìm thấy thiết bị: " + targetDevice);
        bot.sendMessage(GROUP_ID, "✗ Thiết bị *" + targetDevice + "*" không tồn tại trong danh sách tin cậy.", "Markdown");
        targetDevice = "";
        foundDeviceName = "";
    }
}
else
{
    if (currentDeviceIndex >= trustedDevices.size())
        currentDeviceIndex = 0;
    foundDeviceName = trustedDevices[currentDeviceIndex++].name;
    Serial.println("✓ Thiết bị tin cậy phát hiện: " + foundDeviceName);
}
}
}

```

- Hàm sinh ngẫu nhiên kinh độ và vĩ độ và tính khoảng cách:

```

double deg2rad(double deg)
{
    return (deg * (3.14159265358979323846 / 180.0));
}

double haversine(double lat1, double lon1, double lat2, double lon2)
{
    const double R = 6371000;
    double phi1 = deg2rad(lat1), phi2 = deg2rad(lat2);
    double deltaPhi = deg2rad(lat2 - lat1), deltaLambda = deg2rad(lon2 - lon1);
    double a = sin(deltaPhi / 2) * sin(deltaPhi / 2) + cos(phi1) * cos(phi2) * sin(deltaLambda / 2) * sin(deltaLambda / 2);
    return R * 2 * atan2(sqrt(a), sqrt(1 - a));
}

```

- Hàm tính khoảng cách khi có sự chênh lệch kinh độ và vĩ độ nếu có:

```

DeviceData getMockData()
{
    DeviceData data;

    bool over10m = random(0, 2); // 0 hoặc 1

    if (over10m)
    {
        // Vị trí di chuyển xa hơn một chút (tối đa 10m)
        float deltaLat = random(-100, 100) / 100000.0;
        float deltaLng = random(-100, 100) / 100000.0;
        data.lat = baseLat + deltaLat;
        data.lng = baseLng + deltaLng;
    }
}

```

```

else
{
    // Vị trí di chuyển gần hơn (tối đa 5m)
    float deltaLat = random(-50, 50) / 100000.0;
    float deltaLng = random(-50, 50) / 100000.0;
    data.lat = baseLat + deltaLat;
    data.lng = baseLng + deltaLng;
}

data.distance = haversine(baseLat, baseLng, data.lat, data.lng);

return data;
}

```

- Hàm gửi kết quả lên telegram và nhận lệnh từ telegram:

```

void sendTelegramAlert(float lat, float lng, String deviceName)
{
    double distance = haversine(baseLat, baseLng, lat, lng);
    String message;

    if (distance > 10.0)
    {
        message = "⚠️ *Cảnh báo*: Thiết bị *" + deviceName + "* vượt quá phạm vi 10m!\n";
    }
    else
    {
        message = "✅ Thiết bị *" + deviceName + "* đang trong phạm vi an toàn.\n";
    }

    message += "📏 Khoảng cách: " + String(distance, 1) + "m\n";
    message += "🌐 Vị trí: " + String(lat, 6) + "," + String(lng, 6);
    message += "\n📍 [Xem bản đồ](https://www.google.com/maps?q=" + String(lat, 6) + "," + String(lng, 6) + ")";

    bool success = bot.sendMessage(GROUP_ID, message, "Markdown");

    if (success)
    {
        if (distance > 10.0)
            Serial.println("📧 Gửi *cảnh báo* Telegram thành công!");
        else
            Serial.println("✅ Đã gửi thông tin thiết bị trong phạm vi an toàn đến Telegram.");
    }
    else
    {

```

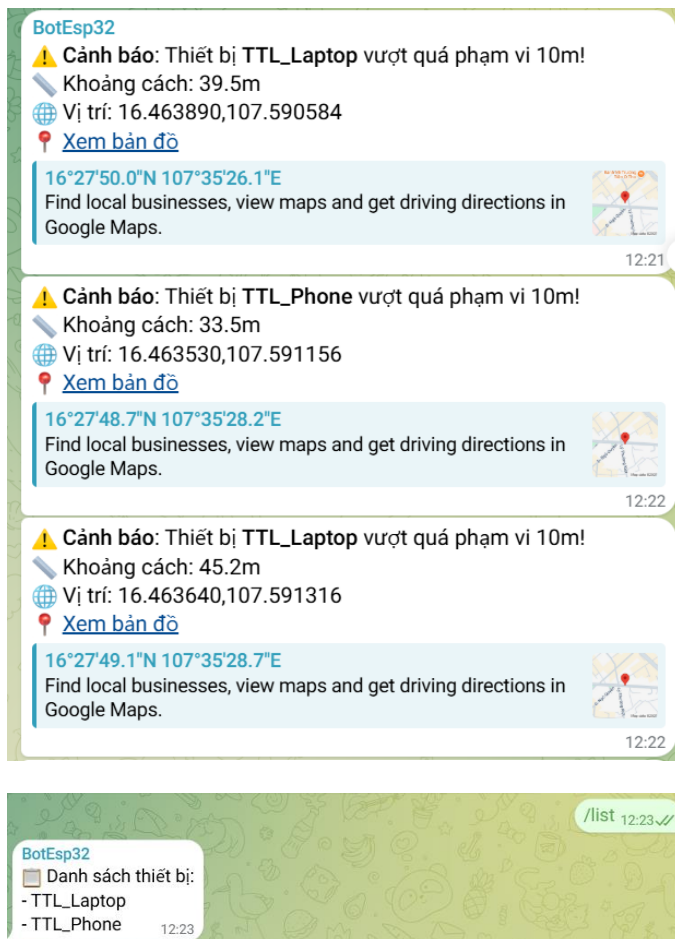
```

        Serial.println("❌ Gửi thông báo Telegram thất bại!");
    }
}
bool scanAllMode = false;
void handleTelegramCommands()
{
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    for (int i = 0; i < numNewMessages; i++)
    {
        String chat_id = String(bot.messages[i].chat_id);
        if (chat_id != GROUP_ID)
            continue;
        String text = bot.messages[i].text;
        if (text.startsWith("/scan_"))
        {
            targetDevice = text.substring(6);
            bot.sendMessage(chat_id, "🔍 Đang tìm kiếm thiết bị: " + targetDevice,
                "");
        }
        else if (text == "/all")
        {
            targetDevice = "";
            scanAllMode = true;
            bot.sendMessage(chat_id, "🔍 Đang quét tất cả thiết bị tin cậy...", "");
        }
        else if (text == "/list")
        {
            String deviceList = "📁 Danh sách thiết bị:\n";
            for (const auto &device : trustedDevices)
            {
                deviceList += "- " + device.name + "\n";
            }
            bot.sendMessage(chat_id, deviceList, "");
        }
    }
}
}

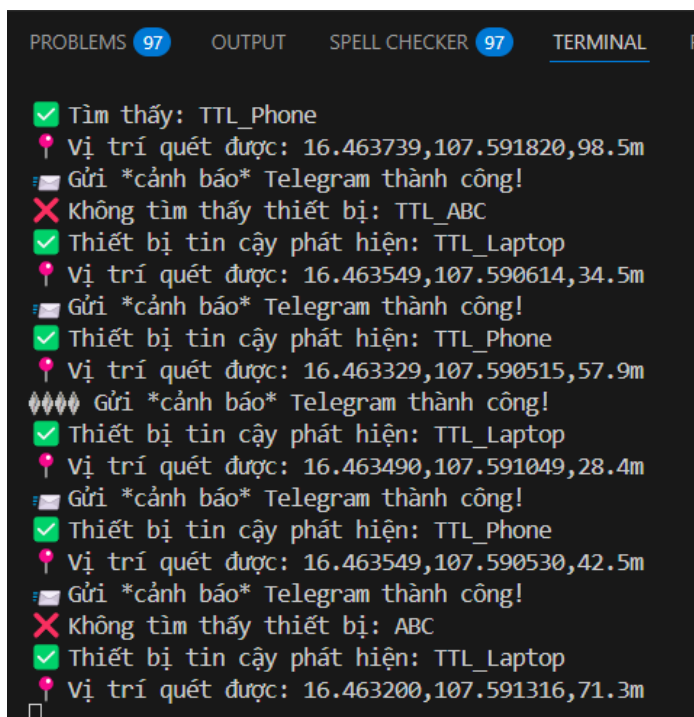
```

## Kết quả kiểm thử:

1. Kết quả gửi lên telegram và nhận lệnh:



Hiện thị monitor:



## Kết luận

Trong đề tài này, em đã xây dựng một hệ thống định vị trong nhà dựa trên vi điều khiển ESP32 và công nghệ Bluetooth Low Energy (BLE) với các dữ liệu sinh ngẫu nhiên. Hệ thống cho phép quét và nhận diện các thiết bị BLE xung quanh, xác định khoảng cách dựa trên cường độ tín hiệu RSSI, và gửi cảnh báo khi thiết bị vượt quá phạm vi an toàn đến người dùng thông qua Telegram.

Dù trong quá trình thực hiện em gặp phải giới hạn về phần cứng và môi trường mô phỏng BLE không hỗ trợ đầy đủ, em đã linh hoạt sử dụng dữ liệu ngẫu nhiên để mô phỏng hành vi của thiết bị thật. Việc này cho phép em kiểm thử được thuật toán xử lý, giao tiếp Telegram và tính năng cảnh báo.

Hệ thống có thể mở rộng thêm các chức năng như định vị theo nhiều điểm (triangulation), lưu lịch sử vị trí, hoặc tích hợp bản đồ trực tiếp. Đây là một hướng đi tiềm năng cho các ứng dụng trong giám sát tài sản, nhà thông minh, và chăm sóc sức khỏe.



### **Tài liệu tham khảo:**

[1] IoT Zone, "BLE là gì? Tổng quan về công nghệ Bluetooth Low Energy," *IoT Zone Blog*. <https://www.iotzone.vn/blog/ble-la-gi/>

[2] Random Nerd Tutorials, "ESP32 Bluetooth Low Energy (BLE) with Arduino IDE," *Random Nerd Tutorials Blog*. <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>

TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA.....

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập – Tự do – Hạnh phúc**

## PHIẾU ĐÁNH GIÁ TIỂU LUẬN

**Học kỳ ..... Năm học ...-...**

[illegible]

**Điểm kết luận:** Bằng số..... Bằng chữ:.....

*Thừa Thiên Huế, ngày ..... tháng ..... năm 20...*

**CBCbT1**  
(Ký và ghi rõ họ tên)

**CBCChT2**  
(Ký và ghi rõ họ tên)