

# **ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

## **TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**Khoa Điện tử – Viễn thông**

**Bộ môn Máy tính – Hệ thống nhúng**



**BÁO CÁO ĐỒ ÁN MÔN HỌC THỰC HÀNH KẾT NỐI VÀ  
THU NHẬN DỮ LIỆU TRONG IOT**

**HỆ THỐNG ĐIỀU KHIỂN LED VÀ QUẠT  
THÔNG QUA ỨNG DỤNG BLE SCANNER**

Sinh viên thực hiện:

Trần Văn Trường

Huỳnh Nguyễn Anh Tuấn

MSSV:

20200390

20200397

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 1 NĂM 2024**

# MỤC LỤC

Tiêu đề	Trang
<b>1. GIỚI THIỆU ĐỒ ÁN .....</b>	<b>1</b>
<b>1.1. Chức năng chính.....</b>	<b>1</b>
<b>1.2. Công nghệ sử dụng.....</b>	<b>1</b>
<b>1.3. Các chuẩn giao tiếp có trong hệ thống .....</b>	<b>2</b>
<b>2. SƠ ĐỒ NỐI DÂY CỦA HỆ THỐNG .....</b>	<b>3</b>
<b>3. BLOCK DIAGRAM.....</b>	<b>4</b>
<b>4. CODE.....</b>	<b>5</b>
<b>5. KẾT QUẢ.....</b>	<b>10</b>
<b>TỔNG KẾT .....</b>	<b>11</b>

# 1. GIỚI THIỆU ĐỒ ÁN

Hệ thống điều khiển led và quạt thông qua ứng dụng BLE Scanner, dự án này kết hợp sự linh hoạt của Arduino, module BLE(HM-10) và cảm biến nhiệt độ DS18B20 để tạo ra một hệ thống thông minh có thể điều khiển LED, quạt, đồng thời cung cấp thông tin về nhiệt độ môi trường. Ứng dụng BLE Scanner trên điện thoại di động giúp người dùng dễ dàng điều khiển và giám sát từ xa.

## 1.1. Chức năng chính

- + **Điều khiển LED và quạt:** Người dùng có thể bật/tắt đèn LED và quạt thông qua ứng dụng BLE Scanner. Điều này mang lại sự thuận tiện và tiết kiệm năng lượng khi chỉ cần kích hoạt thiết bị khi cần thiết.
- + **Hiển thị nhiệt độ:** Hệ thống cung cấp thông tin về nhiệt độ môi trường từ cảm biến DS18B20.
- + **Chế độ tự động và thủ công:** Hệ thống có chế độ tự động dựa trên nhiệt độ, nếu nhiệt độ vượt quá mức xác định, quạt sẽ được bật để làm mát và đèn LED sẽ thay đổi trạng thái. Người dùng cũng có thể chuyển sang chế độ thủ công để kiểm soát trực tiếp.

## 1.2. Công nghệ sử dụng

- + **IDE:** Arduino IDE
- + Arduino Uno R3
- + Module HM-10

- + Cảm biến nhiệt độ DS18B20
- + 1 led, 1 quạt tản nhiệt, 2 điện trở 4.7k $\Omega$ .

### 1.3. Các chuẩn giao tiếp có trong hệ thống

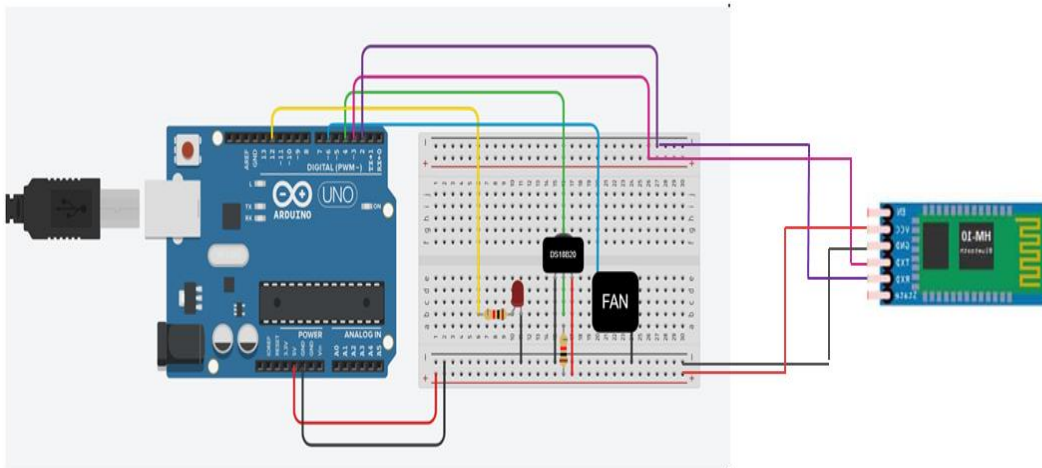
- + **Có dây:** UART (Universal Asynchronous Receiver-Transmitter).

Giao tiếp giữa Arduino và module HM10 sử dụng giao tiếp UART. Cụ thể, sử dụng thư viện SoftwareSerial trên Arduino để tạo một cổng UART ảo (virtual UART) và kết nối chân TX của HM10 với chân RX của Arduino và ngược lại.

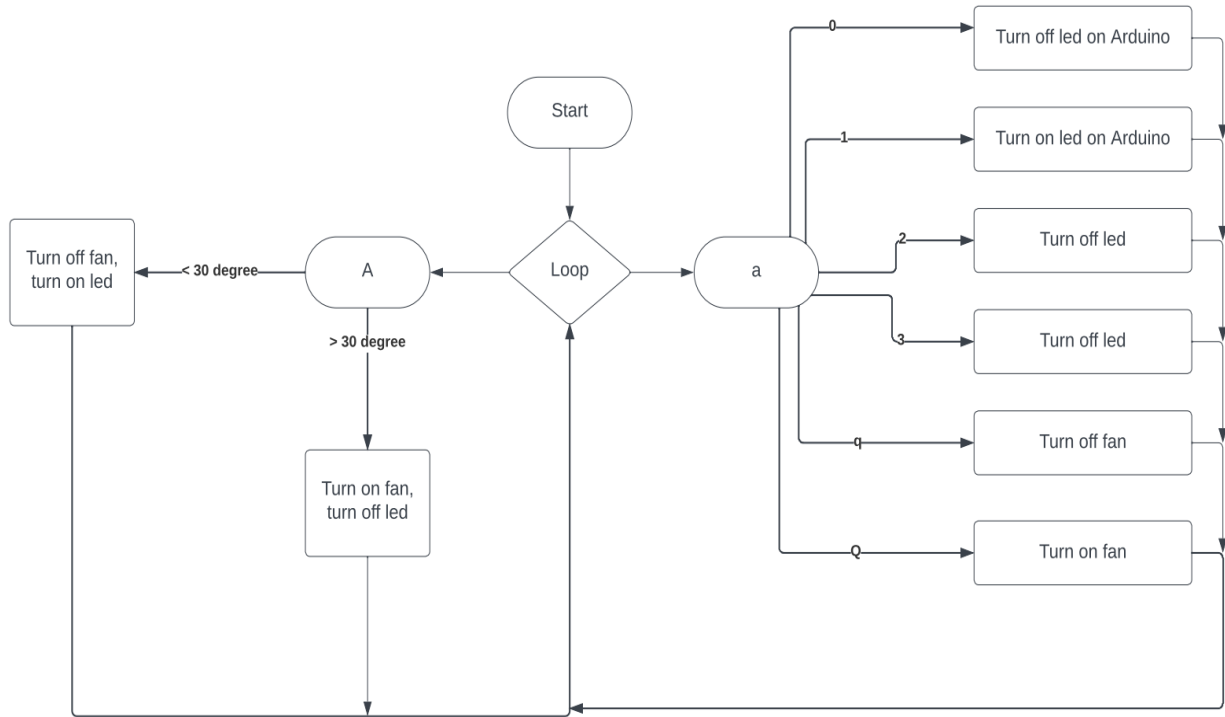
- + **Không dây:** BLE (Bluetooth Low Energy).

Sử dụng để truyền thông giữa Arduino và ứng dụng BLE Scanner trên điện thoại thông qua module BLE HM-10.

## 2. SƠ ĐỒ NỐI DÂY CỦA HỆ THỐNG



### 3. BLOCK DIAGRAM



## 4. CODE

```
#include <SoftwareSerial.h>

#include <DallasTemperature.h>

SoftwareSerial mySerial(2, 3);

int ledPin13 = 13; // LED ở chân D13 (có sẵn trên Arduino)
int ledPin12 = 12; // LED ở chân D12
int fan = 6;       // Quạt ở chân D6
int temp = 4;      // Chân Data của DS18B20 kết nối vào chân D4

bool manualControl = false; // Biến để xác định liệu điều khiển là
                             // tự động (dựa trên nhiệt độ) hay thủ công (tự ứng dụng)

OneWire oneWire(temp);
DallasTemperature sensors(&oneWire);

void setup() {
    mySerial.begin(9600);
    Serial.begin(9600);

    pinMode(ledPin13, OUTPUT);
    pinMode(ledPin12, OUTPUT);
    pinMode(fan, OUTPUT);
```

```

}

void loop() {

    // Kiểm tra và điều khiển quạt và đèn dựa trên dữ liệu từ ứng
    dụng

    checkAppCommands();

    // Đọc nhiệt độ từ cảm biến DS18B20

    sensors.requestTemperatures();

    float temperature = sensors.getTempCByIndex(0);

    // Nếu đang trong chế độ tự động (dựa trên nhiệt độ)

    if (!manualControl) {

        // Hiển thị nhiệt độ dưới dạng số thập phân trên Serial Monitor

        Serial.print("T:");

        Serial.print(temperature, 2); // Hiển thị 2 chữ số thập phân

        Serial.println();

        // Hiển thị nhiệt độ dưới dạng số thập phân trên BLE Scanner

        mySerial.print("T:");

        mySerial.print(temperature, 2); // Hiển thị 2 chữ số thập phân

        mySerial.println();

        // Kiểm tra và điều khiển quạt và đèn dựa trên nhiệt độ
    }
}

```



```

    if (temperature > 30.0) {

        digitalWrite(fan, HIGH); // Bật quạt

        digitalWrite(ledPin12, LOW);

    } else {

        digitalWrite(fan, LOW); // Tắt quạt

        digitalWrite(ledPin12, HIGH);

    }

}

delay(1000); // Đợi 1 giây để giảm tần suất kiểm tra nhiệt độ

}

void checkAppCommands() {

    while (mySerial.available()) {

        char command = mySerial.read();

        Serial.println("DATA RECEIVED:");

        switch (command) {

            case '0':

                digitalWrite(ledPin13, LOW);

                Serial.println("LED 13 off");

                manualControl = true; // Chuyển sang chế độ điều khiển thủ
công

                break;

```

công	<pre> case '1':      digitalWrite(ledPin13, HIGH);      Serial.println("LED 13 on");      manualControl = true; // Chuyển sang chế độ điều khiển thủ </pre>
công	<pre>     break;  case '2':      digitalWrite(ledPin12, LOW);      Serial.println("LED 12 off");      manualControl = true; // Chuyển sang chế độ điều khiển thủ </pre>
công	<pre>     break;  case '3':      digitalWrite(ledPin12, HIGH);      Serial.println("LED 12 on");      manualControl = true; // Chuyển sang chế độ điều khiển thủ </pre>
công	<pre>     break;  case 'Q':      digitalWrite(fan, HIGH);      Serial.println("Fan on");      manualControl = true; // Chuyển sang chế độ điều khiển thủ </pre>
	<pre>     break; </pre>

```
case 'q':  
    digitalWrite(fan, LOW);  
    Serial.println("Fan off");  
    manualControl = true; // Chuyển sang chế độ điều khiển thủ  
công  
    break;  
case 'A':  
    manualControl = false; // Chuyển sang chế độ tự động  
    break;  
case 'a':  
    manualControl = true; // Chuyển sang chế độ thủ công  
    break;  
default:  
    // Handle unexpected characters or commands  
    break;  
}  
}  
}
```

## 5. KẾT QUẢ

Khi chạy chương trình thì trên ứng dụng BLE Scanner sẽ hiển thị một kết nối mang tên module thiết bị. Kết nối vào thiết bị đó sau đó điều khiển các chức năng cơ bản như sau:

+ Nhập ‘a’: Chế độ điều khiển thủ công.

- Nhập ‘0’: LED trên board Arduino Uno R3 tắt.
- Nhập ‘1’: LED trên board Arduino Uno R3 sáng.
- Nhập ‘2’: LED trên board mạch tắt.
- Nhập ‘3’: LED trên board mạch sáng.
- Nhập ‘Q’: Quạt dừng.
- Nhập ‘q’: Quạt chạy.

+ Nhập ‘A’: Chế độ điều khiển tự động.

Hệ thống tự động kiểm tra khi nhiệt độ lớn hơn 30 °C thì sẽ bật quạt và tắt led còn nếu dưới 30 °C sẽ tắt quạt và bật đèn.

**Link mô phỏng kết quả:** <https://www.youtube.com/watch?v=sxtqYa1OsWE>

## TỔNG KẾT:

### ☞ Ưu điểm của hệ thống:

- + **Thuận tiện và linh hoạt:** Người dùng có thể dễ dàng điều khiển thiết bị từ bất kỳ đâu thông qua ứng dụng BLE Scanner trên điện thoại di động.
- + **Tiết kiệm năng lượng:** Chế độ tự động dựa trên nhiệt độ giúp giảm tiêu thụ năng lượng khi không cần thiết.
- + **Đơn giản và dễ sử dụng:** Giao diện đơn giản và dễ sử dụng trên ứng dụng BLE Scanner.

### ☞ Nhược điểm của hệ thống:

- + **Hạn chế trong việc mở rộng chức năng:** Dự án này có thể có hạn chế trong việc mở rộng chức năng nếu người dùng muốn tích hợp thêm các thiết bị hoặc cảm biến khác.
- + **Phụ thuộc vào kết nối BLE:** Dự án phụ thuộc vào kết nối BLE để truyền dữ liệu giữa Arduino và ứng dụng BLE Scanner trên điện thoại di động. Nếu có sự cố trong kết nối Bluetooth, có thể làm ảnh hưởng đến khả năng điều khiển và giám sát từ xa.
- + **Ứng dụng giới hạn:** Sự phụ thuộc vào ứng dụng BLE Scanner có thể làm giảm tính linh hoạt, đặc biệt nếu người dùng muốn tích hợp hệ thống vào các hệ sinh thái tương tác khác.