



BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG
CƠ SỞ VÀ ỨNG DỤNG IOTS
MMH: ITFA436064
Thời gian thực hiện: 01 buổi

Họ và tên:Vũ Tiến Phát-21151309

Hoàng Thị Diễm Quỳnh-21151326

Nguyễn Nam Huy-21151244

Ngô Xuân Thọ-21119376

Trương Nguyễn Quốc Thắng-22119231

1. So sánh chuẩn truyền thông Wifi và chuẩn truyền thông Zigbee

Wifi

- Phạm vi truyền thông
 - + Phạm vi truyền thông lớn: hàng trăm mét
- Tốc độ truyền thông:
 - + Tốc độ truyền thông cao, thích hợp cho việc truyền dữ liệu lớn
- Tiêu thụ năng lượng:
 - + Tiêu thụ nhiều năng lượng nên làm giảm tuổi thọ pin trong các thiết bị di động hoặc các thiết bị IoT yêu cầu hoạt động lâu dài
- Khả năng đa mạng:
 - + WiFi thường không được tối ưu hóa cho các mạng cảm biến mà có nhiều thiết bị kết nối.
- Ứng dụng:
 - + WiFi thường được sử dụng cho các ứng dụng đòi hỏi tốc độ cao như truyền dữ liệu video, trình duyệt web và trò chơi trực tuyến.
- Băng tần: 2,4GHz

Zigbee

- Phạm vi truyền thông
 - + Phạm vi truyền thông nhỏ hơn: từ vài mét đến khoảng 100m.
- Tốc độ truyền thông:
 - + Tốc độ truyền thông thấp hơn, thích hợp cho các ứng dụng cần ít dữ liệu
- Tiêu thụ năng lượng:
 - + Tiêu thụ ít năng lượng hơn, nên thời gian sử dụng của pin sẽ dài hơn
- Khả năng đa mạng:
 - + Zigbee được thiết kế để hỗ trợ các mạng mà có nhiều thiết bị, cho phép hình thành mạng lưới mesh cho phạm vi truyền thông lớn hơn và tính ổn định
- Ứng dụng:
 - + Zigbee thường được sử dụng cho các ứng dụng IoT, cảm biến thông minh, điều khiển hệ thống tự động trong nhà, như đèn, máy lạnh, và thiết bị an ninh.
- Băng tần: 800-900MHz

2. So sánh ESP32 và ESP8266

ESP32:

Là một dòng vi điều khiển hệ thống trên chip giá rẻ. ESP32 là phiên bản nâng cấp của dòng ESP 8266. Dòng ESP32 được tạo ra và phát triển bởi Espressif Systems. ESP có bộ đồng xử lý lõi kép và công suất thấp. Nó được phát triển vì sự thiếu bảo mật trong ESP8266.

ESP8266:

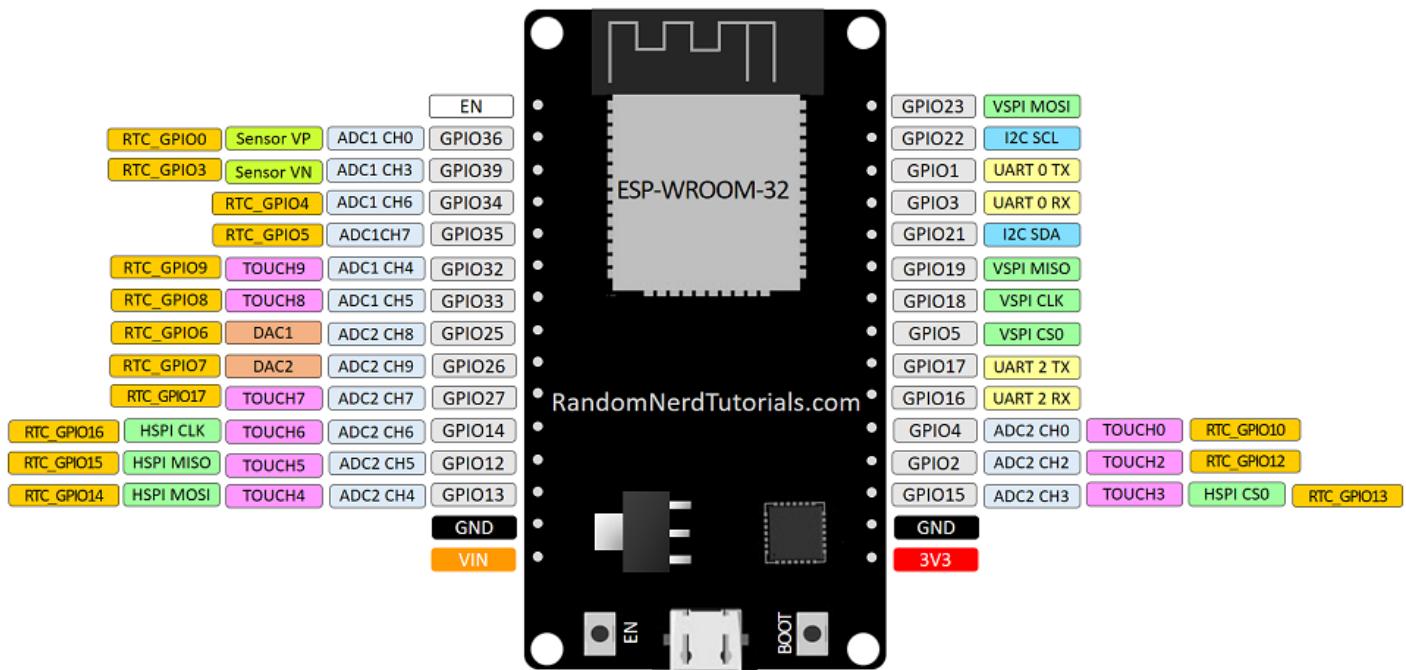
Là một hệ thống hoàn chỉnh hoặc khép kín trên mạch chip (SOC), module Wifi với ngăn xếp giao thức IP/TCP. ESP8266 cho phép truy cập vào bất kỳ bộ điều khiển nào thông qua kết nối Wifi.

Một trong những chức năng chính của ESP8266 là lưu trữ ứng dụng nào hoặc giảm tải tất cả các chức năng mạng Wifi. Nó có độ bền cao và có thể hoạt động ổn định ngay cả trong môi trường công nghiệp khắc nghiệt.

Điều này là do phạm vi nhiệt độ hoạt động rộng của nó. Nó cũng cung cấp kiến trúc tiết kiệm năng lượng và bộ xử lý Tensilica 32-bit.

So sánh	ESP32	ESP8266
Tần số đồng hồ	160 hoặc 240 Hz	80 Hz
Bluetooth	BLE	Không
Cảm biến Hall	Có	Không
Giao diện camera	Không	Không
Cảm biến nhiệt độ	Có	Không
Cảm biến cảm ứng	10	Không
Bảo mật	Mã hóa flash Khởi động bảo mật. OTP 1024-bit	Không
Tiêu thụ công suất thấp	Cảm biến sâu 10uA	20 uA
Nhiệt độ	Có	Không
Đồng xử lý	ULP	Không
Tổng GPIO	39	17
Mã hóa	RSA,RNG,ECC,SHA-2,AES	17
SPI	4	2
USB OTG	Không	Không
Vị điều khiển	LX6 Xtensa 32-bit lõi đơn hoặc lõi kép	L 106 Xtensa lõi đơn 32-bit
ROM	448KB	Không
CAN	2	Không
Ethernet	10/100 Mbps	Không
SPIRAM bên ngoài	Lên tới 16MB	Lên tới 16MB

3. Sơ đồ chân kết nối ESP32 (hoặc ESP8266) (**lập bảng** chức năng các chân).



Tên các chân	Chức năng
EN/RST	Điều khiển nguồn, khởi động lại module
GPIO	Sử dụng làm đầu vào hoặc đầu ra
UART	Được sử dụng cho truyền dữ liệu qua cổng serial UART, cho phép kết nối với các thiết bị sử dụng giao thức serial như máy tính, module Bluetooth, và nhiều thiết bị ngoại vi khác
I2C	Được sử dụng cho giao tiếp I2C, cho phép kết nối với các cảm biến và thiết bị I2C khác.
SPI	Được sử dụng cho giao tiếp SPI, phù hợp cho việc kết nối với các mạch điều khiển, màn hình, và nhiều thiết bị khác
ADC	Được sử dụng để đo giá trị tín hiệu analog, cho phép kết nối với các cảm biến analog hoặc đo các điện áp ngoại vi.
DAC	Cho phép tạo tín hiệu analog từ dữ liệu kỹ thuật số, thích hợp cho các ứng dụng âm thanh và video.
RTC	Đánh thức ESP32 khi trong chế độ Low Power Mode. Sử dụng như 1 chân ngắt ngoài.
Touch GPIO	Hỗ trợ chức năng cảm ứng, cho phép xác định khi

	nào người dùng chạm vào các điểm cảm ứng trên bảng mạch.
GND,Vin,5V	Cấp nguồn

4. Các bước cài đặt và giao tiếp ESP32 (hoặc ESP 8266) với phần mềm Arduino IDE

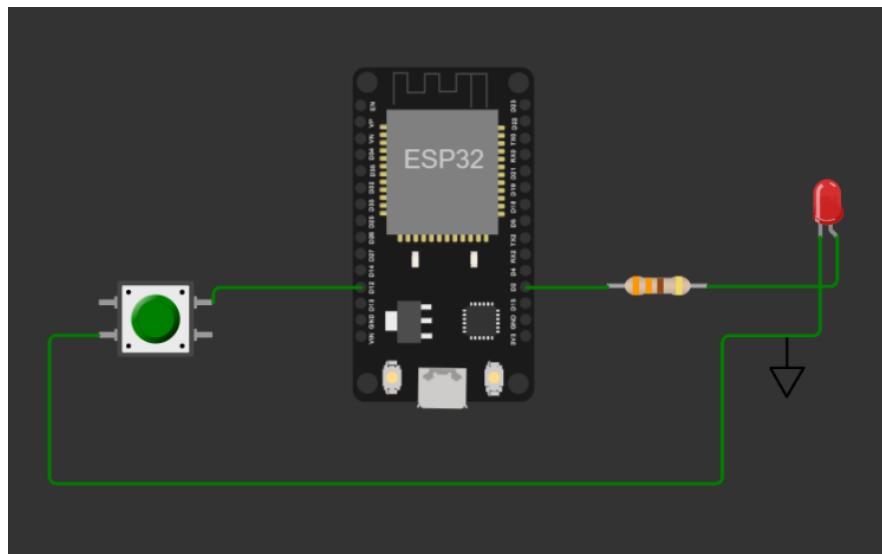
- Các bước cài đặt Arduino IDE và thư viện cho ESP 32:
- Bước 1: Tải IDE trên trang arduino.cc
- Khi tải cần phải chọn đúng hệ điều hành mình đang sử dụng
- Sau khi tải về thì mở file tải về lên và set up như mặc định
- Bước 2: Sau khi tải xong ta mở đến phần giao diện của Arduino và cài đặt thư viện cho ESP Ta vào File -> Preferences. Sau đó 1 cửa sổ hiện lên. Ta chọn biểu tượng tờ giấy bên phải đường link ở phần Additional Boards Manager URLs và dán đường link:
https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json vào sau đó bấm ok.
- Bước 3 cài đặt thư viện ESP32:
- Vào Tools-> Board -> Boards-> Boards Manager Cửa sổ cài đặt cho các board, vào thanh tìm kiếm "esp" rồi chọn esp32 -> Install

5. Các **bước thực hiện** và **giải thích code** chương trình chớp tắt LED kết hợp giao tiếp nút nhấn trên ESP32 (hoặc ESP8266) và hình ảnh chụp các kết quả (link đến video clip upload youtube nếu có)

Các bước thực hiện:

Bước 1: Thống kê linh kiện: ESP32, Led, nút nhấn, trổ.

Bước 2: Thiết kế sơ đồ nguyên lý



Bước 3: Vẽ sơ đồ khối

Bước 4: Viết chương trình

```

int tt =0; // Tạo biến trạng thái
void setup() {
    pinMode(2, OUTPUT); // Khai báo chân D0 là chân Output
    pinMode(12, INPUT_PULLUP); // Khai báo chân D12 là chân Input, nút nhấn kéo lên
}

void loop() {
    // Viết hàm đọc nút nhấn có chống dội
    // Do xuống trở lên nên nút nhấn ở trạng thái HIGH
    // Khi nhấn nút sẽ chuyển xuống trạng thái LOW
    // Nếu đọc được trạng thái LOW thì sẽ tăng biến trạng thái
    if(digitalRead(12)==LOW){
        delay(20);
        if(digitalRead(12)==LOW){
            tt++;
            if(tt > 2) tt=1;
            while(digitalRead(12) == LOW);
        }
    }
    // Nếu biến trạng thái = 1 thì Led sẽ chớp tắt với chu kỳ 2s
    // Nếu biến trạng thái = 2 thì Led sẽ tắt
    if(tt == 1){
        digitalWrite(2, HIGH);
        delay(1000);
        digitalWrite(2, LOW);
        delay(1000);
    }
    if(tt == 2) {
        digitalWrite(2, LOW);
    }
}

```

Bước 5: Chạy chương trình

Video kết quả link: https://youtu.be/lJEQzw_0lzk

- Thực hiện giao tiếp ESP32 và 2 Led đơn điều khiển chớp tắt 2 Led (có sử dụng FreeRTOS). Giải thích code và chụp hình ảnh (kèm clip nếu có) kết quả thực hiện.

```

// Khai báo từng task
TaskHandle_t Task1;
TaskHandle_t Task2;

// Khai báo chân cho 2 Led
const int led_1 = 32;
const int led_2 = 25;

void setup(){
    Serial.begin(115200);
    // Khai báo trạng thái OUTPUT cho 2 chân nối với Led
    pinMode(led_1, OUTPUT);
    pinMode(led_2, OUTPUT);
    // Thiết lập nhiệm vụ chạy độc lập cho từng Led
    xTaskCreatePinnedToCore(Task1code, "Task1", 10000, NULL, 1, &Task1, 0);
    delay(500);
    xTaskCreatePinnedToCore(Task2code, "Task2", 10000, NULL, 1, &Task2, 1);
    delay(500);
}
// Nhiệm vụ của task1: chớp tắt Led_1 với chu kỳ 1s
void Task1code( void * parameter){

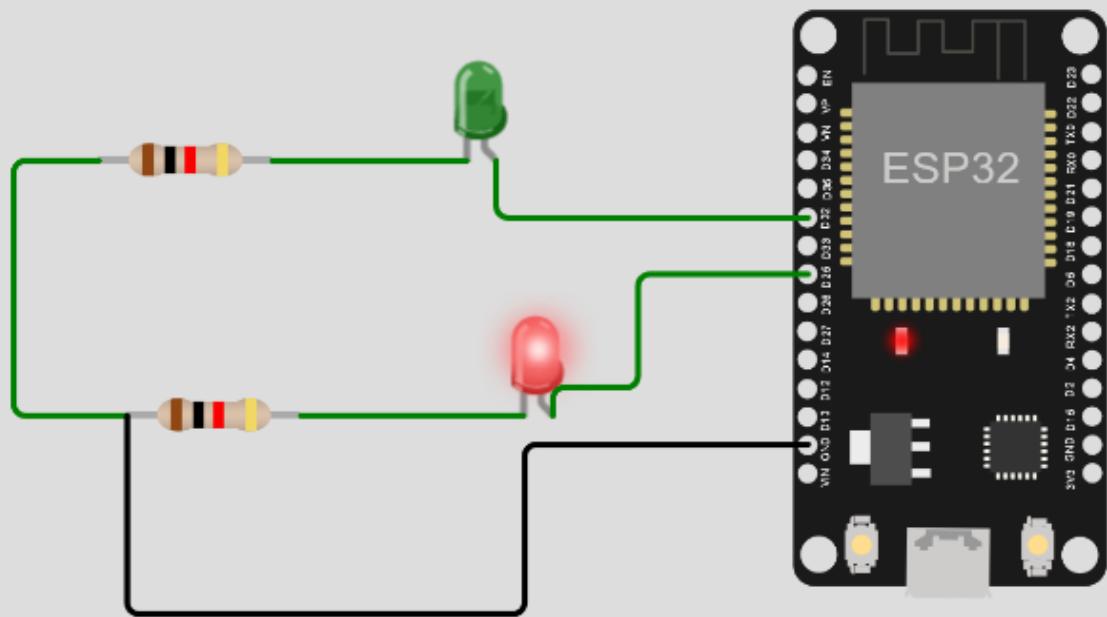
    for(;){
        digitalWrite(led_1, HIGH);
        delay(500);
        digitalWrite(led_1, LOW);
        delay(500);
    }
}

// Nhiệm vụ của task2: chớp tắt Led_1 với chu kỳ 1s
void Task2code( void * parameter){

    for(;){
        digitalWrite(led_2, HIGH);
        delay(500);
        digitalWrite(led_2, LOW);
        delay(500);
    }
}

void loop(){}

```



```
load:0x3fff0030,len:1156  
load:0x40078000,len:11456  
ho 0 tail 12 room 4  
load:0x40080400,len:2972  
entry 0x400805dc  
Task1 is running on core0  
Task2 is running on core1
```

Video kết quả link:

<https://drive.google.com/file/d/1EHL0OicxRRvDBkPACHkda4jtrNINoFeF/view?usp=sharing>

Tài liệu tham khảo (liệt kê theo IEEE)

Hình ảnh làm việc nhóm.

