**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA ĐIỆN – ĐIỆN TỬ**

****

**BÀI TẬP LỚN**

**DỰ ÁN HỆ THỐNG VƯỜN THÔNG MINH TÍCH HỢP BLYNK**

**Sinh viên thực hiện:** Cao Đình Thi \_ 211410101

Nguyễn Thành Trung \_ 211413750

Nguyễn Quang Anh \_ 211410307

Nguyễn Khắc Anh Tú \_ 211410588

Vũ Duy Tùng \_ 211410459

**Lớp**: Cử nhân Kỹ thuật điện tử - viễn thông

**Khoa**: Điện – Điện tử

**Giảng viên:** Ngô Thanh Bình

Contents

[I. Giới thiệu 4](#_Toc178885530)

[II. Thành phần của hệ thống 5](#_Toc178885531)

[**1. ESP32** 5](#_Toc178885532)

[**2. Cảm biến DHT11** 6](#_Toc178885533)

[**3. Relay** 6](#_Toc178885535)

[**4. OLED SSD-1306** 7](#_Toc178885536)

[**5. Công tắc bấm, đèn LED, điện trở** 7](#_Toc178885537)

[**- Nút bấm** 8](#_Toc178885538)

[**- Đèn LED** 8](#_Toc178885540)

[**- Điện trở** 9](#_Toc178885541)

[**6. Blynk** 9](#_Toc178885542)

[**-**  **Nền tảng Blynk** 9](#_Toc178885543)

[III, Nguyên lý hoạt động 10](#_Toc178885544)

[**Nguyên lý hoạt động** 10](#_Toc178885545)

[**1. Giám sát nhiệt độ và độ ẩm** 10](#_Toc178885546)

[**2. Kiểm tra điều kiện tưới nước** 10](#_Toc178885547)

[**3. Tưới nước tự động** 11](#_Toc178885548)

[**4. Tưới nước thủ công (nút bấm)** 11](#_Toc178885549)

[**5. Tưới nước từ xa qua Blynk** 11](#_Toc178885550)

[**6. Hiển thị thông tin lên màn hình OLED** 11](#_Toc178885551)

[**7. Điều khiển hệ thống thông qua ứng dụng Blynk** 11](#_Toc178885552)

[**8. Xử lý đa nhiệm với FreeRTOS** 12](#_Toc178885553)

[**9. Tính năng bảo vệ hệ thống** 12](#_Toc178885554)

[IV. Kiến Trúc Hệ Thống 12](#_Toc178885555)

[**1. Sơ đồ phần cứng** 12](#_Toc178885556)

[**a, Sơ đồ mô tả trên wokwi** 12](#_Toc178885557)

[**b, Sơ đồ thực tế** 14](#_Toc178885558)

[**2. Chi tiết chức năng phần mềm** 14](#_Toc178885560)

[**2.1. Khai báo cấu hình Blink** 14](#_Toc178885561)

[**2.2 . Khai báo thư viện** 15](#_Toc178885562)

[**2.3 Khai báo các biến và cấu hình phần cứng** 15](#_Toc178885563)

[**2.4 Trạng thái hệ thống và ngưỡng cảm biến** 18](#_Toc178885564)

[**2.5 Khai báo biến toàn cục và kết nối Blynk** 19](#_Toc178885565)

[**2.6. Cập nhật OLED và LED** 20](#_Toc178885566)

[**2.7 Tác vụ đọc cảm biến DHT** 22](#_Toc178885567)

[**2.8 Tác vụ điều khiển tưới nước tự động** 25](#_Toc178885568)

[**2.9 Tác vụ đọc nút bấm để điều khiển tưới nước thủ công** 27](#_Toc178885569)

[**2.10 Tác vụ xử lý tưới nước từ Blynk** 28](#_Toc178885570)

[**2.11 Cập nhật trạng thái nút Blynk (pin V3)** 30](#_Toc178885571)

[**2.12 Hàm setup()** 32](#_Toc178885572)

[**2.13. Hàm loop()** 38](#_Toc178885573)

[V, Kết Luận 39](#_Toc178885574)

# **I. Giới thiệu**

Trong những năm gần đây, việc ứng dụng công nghệ vào nông nghiệp đang trở thành xu hướng tất yếu nhằm nâng cao năng suất và chất lượng cây trồng. Việc kiểm soát các yếu tố môi trường như nhiệt độ, độ ẩm đất đóng vai trò quan trọng trong quá trình sinh trưởng và phát triển của cây trồng. Hệ thống đo thông số nhiệt độ, độ ẩm và tưới cây tự động sử dụng FreeRTOS và ESP32 ra đời nhằm giải quyết vấn đề này.

**Hệ thống được thiết kế với các mục tiêu:**

* **Giám sát:** Theo dõi liên tục nhiệt độ, độ ẩm đất và độ ẩm không khí xung quanh cây trồng.
* **Tự động hóa:** Tự động tưới nước cho cây dựa trên các thông số thu thập được, đảm bảo cây trồng luôn nhận được lượng nước phù hợp.
* **Hiệu quả:** Tiết kiệm nước và công sức chăm sóc, đồng thời tăng năng suất cây trồng.
* **Linh hoạt:** Dễ dàng điều chỉnh các thông số tưới tiêu cho phù hợp với từng loại cây trồng.

**Hệ thống sử dụng:**

* **ESP32:** Là một vi điều khiển mạnh mẽ, tích hợp sẵn Wi-Fi và Bluetooth, cho phép kết nối Internet và điều khiển từ xa.
* **FreeRTOS:** Hệ điều hành thời gian thực cho phép xử lý nhiều tác vụ đồng thời một cách hiệu quả.
* **Cảm biến:** Cảm biến nhiệt độ, độ ẩm đất và độ ẩm không khí để thu thập dữ liệu môi trường.

**Đề tài này sẽ trình bày:**

* Thiết kế phần cứng của hệ thống, bao gồm sơ đồ mạch và lựa chọn linh kiện.
* Lập trình điều khiển hệ thống sử dụng FreeRTOS trên ESP32.
* Giao diện người dùng để giám sát và điều khiển hệ thống.

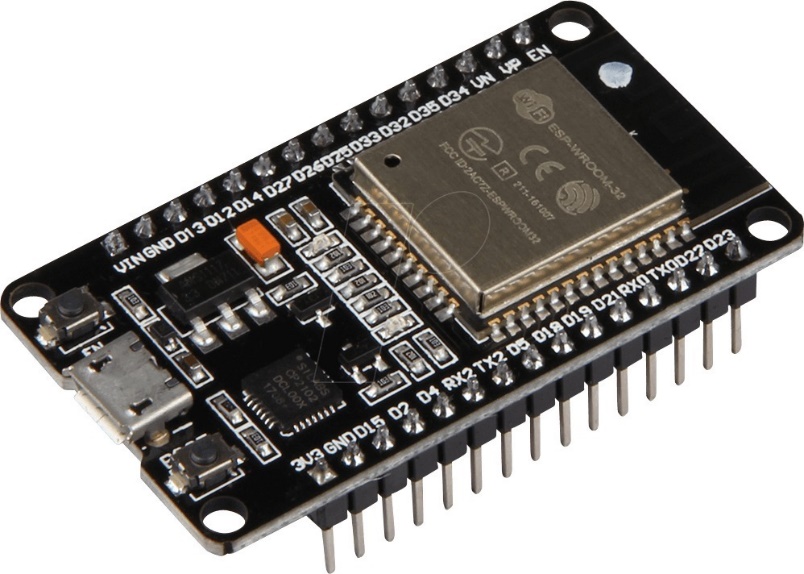
**Kết quả mong đợi:**

* Xây dựng một hệ thống tưới tiêu tự động hiệu quả, tiết kiệm và dễ sử dụng.
* Góp phần ứng dụng công nghệ vào nông nghiệp, nâng cao năng suất và chất lượng cây trồng.

# **II. Thành phần của hệ thống**

## **1. ESP32**

- ESP32 là một vi điều khiển tích hợp WiFi và Bluetooth, được phát triển bởi Espressif Systems. Nó cung cấp một nền tảng mạnh mẽ cho các ứng dụng IoT, cho phép kết nối không dây dễ dàng và linh hoạt.



**- Các tính năng nổi bật :**

* **Kết nối không dây:** Hỗ trợ WiFi 802.11 b/g/n và Bluetooth Classic/Low Energy.
* **Hiệu suất cao:** Dual-Core CPU với tốc độ lên đến 240 MHz và bộ nhớ SRAM từ 160KB đến 520KB.
* **Nhiều cổng I/O:** Cung cấp nhiều chân GPIO và hỗ trợ các giao thức như SPI, I2C, UART, ADC và PWM.
* **Tiết kiệm năng lượng:** Thiết kế với chế độ tiết kiệm năng lượng cho các ứng dụng chạy pin.
* **Hỗ trợ lập trình linh hoạt:** Có thể lập trình bằng Arduino, MicroPython, và ESP-IDF.

- Trong dự án Vườn Thông minh, ESP32 là bộ điều khiển chính, giúp đọc dữ liệu từ cảm biến, điều khiển tưới nước, và kết nối với nền tảng Blynk để điều khiển từ xa.

## **2. Cảm biến DHT11**

## Cảm biến nhiệt độ - độ ẩm DHT11 ( 3C1.2 ) – https://dientubachviet.com/

## 

 **Chức năng**: DHT11 là cảm biến đo nhiệt độ và độ ẩm của môi trường. Dữ liệu này được sử dụng để xác định thời điểm cần tưới nước cho cây.

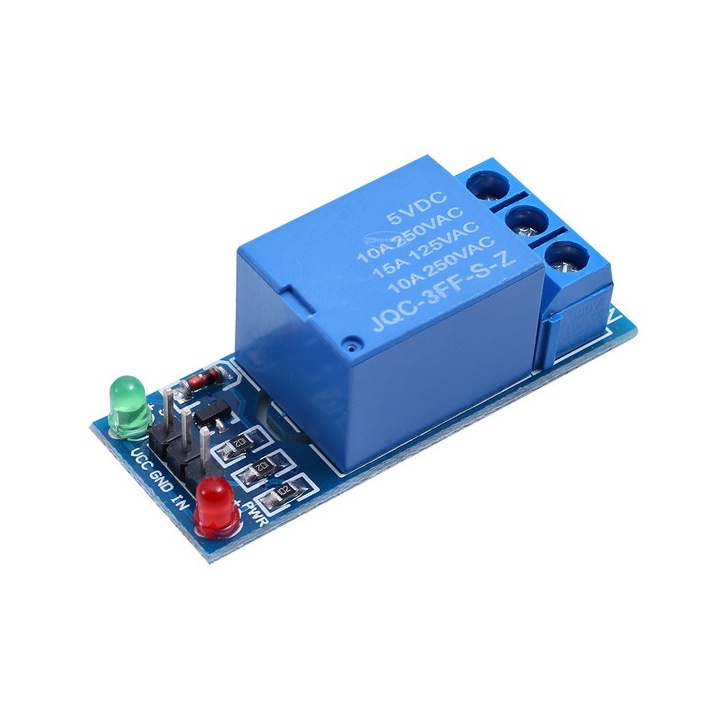
 **Đặc điểm kỹ thuật**:

* Đo nhiệt độ trong khoảng từ 0 đến 50 độ C với độ chính xác ±2°C.
* Đo độ ẩm từ 20% đến 90% với độ chính xác ±5%.
* Tần số đọc dữ liệu khoảng 1 giây một lần.

- Trong dự án Vườn Thông minh, cảm biến DHT11 được sử dụng để đo nhiệt độ và độ ẩm trong môi trường vườn. Dữ liệu đo được sẽ được gửi đến ESP32 để hiển thị trên màn hình OLED và nền tảng Blynk. Hệ thống sẽ sử dụng thông tin này để tự động điều khiển tưới nước khi độ ẩm thấp hơn ngưỡng quy định.

## **3. Relay**

- Relay là một thiết bị điện tử hoạt động như một công tắc điều khiển từ xa, cho phép bật hoặc tắt dòng điện trong mạch điện lớn bằng cách sử dụng một tín hiệu điện nhỏ hơn.



**- Tính năng chính:**

* **Bật/tắt mạch điện:** Relay có khả năng bật hoặc tắt dòng điện cho các thiết bị khác, như máy bơm nước trong hệ thống tưới cây.
* **Cách ly điện:** Relay cung cấp cách ly giữa mạch điều khiển (thường ở mức điện áp thấp) và mạch tải (mức điện áp cao), bảo vệ các linh kiện nhạy cảm khỏi hư hỏng.
* **Điều khiển từ xa:** Relay có thể được điều khiển bằng tín hiệu từ vi điều khiển (như ESP32), cho phép tự động hóa các quy trình.

**- Ứng dụng trong Dự án Vườn Thông minh**

Trong dự án Vườn Thông minh, relay được sử dụng để điều khiển máy bơm nước. Khi cảm biến DHT22 phát hiện độ ẩm đất thấp hơn ngưỡng quy định, ESP32 sẽ kích hoạt relay để bật máy bơm và tưới nước cho cây. Sau khoảng thời gian đã định, relay sẽ tắt máy bơm, giúp duy trì độ ẩm thích hợp cho cây trồng.

## **4. OLED SSD-1306**

- Màn hình OLED SSD-1306 là một loại màn hình hiển thị sử dụng công nghệ OLED, thường được sử dụng trong các dự án điện tử nhờ vào kích thước nhỏ gọn và khả năng hiển thị sắc nét.



#### - Tính năng chính:

* **Độ phân giải:** Thường có độ phân giải 128x64 pixel, cung cấp hình ảnh rõ nét.
* **Giao tiếp I2C/SPI: Hỗ** trợ cả giao thức I2C và SPI, giúp dễ dàng kết nối với các vi điều khiển như ESP32.
* **Độ tương phản cao:** Màn hình hiển thị màu đen sâu và màu trắng sáng, cho phép nhìn rõ trong nhiều điều kiện ánh sáng.
* **Tiết kiệm năng lượng:** Tiêu thụ ít năng lượng, thích hợp cho các ứng dụng pin.

#### - Ứng dụng trong Dự án Vườn Thông minh

Trong dự án Vườn Thông minh, màn hình SSD-1306 được sử dụng để hiển thị thông tin nhiệt độ và độ ẩm từ cảm biến DHT22, cùng với trạng thái hoạt động của hệ thống như "Watering" (đang tưới nước) và "Normal" (bình thường).

**5. Công tắc bấm, đèn LED, điện trở**

### **- Nút bấm**

### 

### 

Nút bấm là một công tắc cơ học đơn giản cho phép người dùng kích hoạt hoặc tắt một chức năng trong mạch điện.

#### Tính năng chính:

* **Dễ sử dụng**: Cung cấp cách giao tiếp đơn giản với người dùng.
* **Giao tiếp trực tiếp**: Khi được nhấn, nó đóng mạch điện, cho phép dòng điện chạy qua.

#### Ứng dụng trong Dự án Vườn Thông minh

**Nút bấm**: Được sử dụng để kích hoạt chế độ tưới nước thủ công.

### **- Đèn LED**



Đèn LED (Light Emitting Diode) là một thiết bị phát sáng hiệu quả, thường được sử dụng để hiển thị trạng thái hoạt động của hệ thống.

#### Tính năng chính:

* **Tiết kiệm năng lượng**: Tiêu thụ ít điện năng so với các loại đèn truyền thống.
* **Độ bền cao**: Thời gian sử dụng lâu dài, không dễ hỏng hóc.
* **Đa dạng màu sắc**: Có thể có nhiều màu sắc khác nhau để thể hiện các trạng thái khác nhau (ví dụ: LED xanh cho trạng thái hoạt động bình thường, LED đỏ cho trạng thái cảnh báo).

**Đèn LED**: LED xanh để báo hiệu hoạt động bình thường và LED đỏ để cảnh báo trạng thái tưới nước.

### **- Điện trở**

****

Điện trở là một linh kiện điện tử được sử dụng để hạn chế dòng điện trong mạch.

#### Tính năng chính:

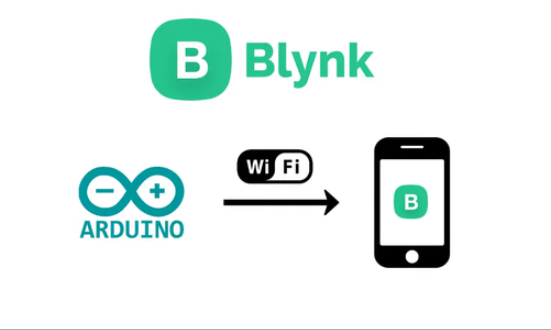
* **Kiểm soát dòng điện**: Bảo vệ các linh kiện nhạy cảm như đèn LED khỏi dòng điện quá cao.
* **Điều chỉnh tín hiệu**: Dùng để tạo ra các mức điện áp khác nhau trong mạch.

#### Ứng dụng trong Dự án Vườn Thông minh

* **Điện trở**: Được sử dụng để hạn chế dòng điện chạy qua đèn LED, đảm bảo chúng hoạt động an toàn và hiệu quả.

**6. Blynk**

### **-** **Nền tảng Blynk**

****

Blynk là một nền tảng IoT cho phép người dùng xây dựng ứng dụng di động để điều khiển và giám sát các thiết bị điện tử từ xa thông qua Internet.

#### Tính năng chính:

* **Giao diện người dùng thân thiện**: Cung cấp giao diện kéo và thả, giúp người dùng dễ dàng thiết kế ứng dụng mà không cần nhiều kiến thức lập trình.
* **Hỗ trợ nhiều loại thiết bị**: Tương thích với nhiều loại vi điều khiển như ESP32, Arduino và Raspberry Pi.
* **Tích hợp nhiều tính năng**: Cung cấp các widget (tiện ích) như nút bấm, thanh trượt, và biểu đồ để người dùng có thể giám sát và điều khiển thiết bị.
* **Giao tiếp qua Internet**: Cho phép điều khiển thiết bị từ bất kỳ đâu thông qua kết nối WiFi hoặc mạng di động.

#### Ứng dụng trong Dự án Vườn Thông minh

Trong dự án Vườn Thông minh, nền tảng Blynk được sử dụng để tạo ứng dụng di động cho phép người dùng theo dõi và điều khiển các chức năng của hệ thống, như tưới nước và kiểm tra nhiệt độ, độ ẩm. Người dùng có thể nhận thông báo và kiểm soát hệ thống một cách thuận tiện từ xa.

**III, Nguyên lý hoạt động**

**Nguyên lý hoạt động**

Hệ thống vườn thông minh hoạt động dựa trên việc giám sát nhiệt độ và độ ẩm của môi trường, sau đó điều khiển tưới nước tự động khi các thông số môi trường vượt qua ngưỡng cài đặt. Bên cạnh đó, hệ thống còn hỗ trợ tính năng điều khiển tưới nước từ xa thông qua ứng dụng Blynk, cho phép người dùng can thiệp thủ công khi cần thiết. Dưới đây là nguyên lý hoạt động chi tiết của hệ thống:

**1. Giám sát nhiệt độ và độ ẩm**

- Cảm biến DHT11 được sử dụng để liên tục đo nhiệt độ và độ ẩm trong không gian xung quanh cây trồng. Mỗi giây, hệ thống sẽ thu thập dữ liệu từ cảm biến này và truyền về ESP32 để xử lý.

- Dữ liệu nhiệt độ và độ ẩm được hiển thị trực tiếp trên màn hình OLED và đồng thời gửi lên ứng dụng Blynk. Người dùng có thể theo dõi các thông số này theo thời gian thực thông qua điện thoại hoặc máy tính bảng có cài đặt ứng dụng Blynk.

**2. Kiểm tra điều kiện tưới nước**

- Hệ thống thiết lập các ngưỡng nhiệt độ và độ ẩm để xác định khi nào cần tưới nước cho cây. Cụ thể:

- Ngưỡng nhiệt độ thấp (25°C) và nhiệt độ cao (35°C) được đặt làm giới hạn an toàn. Nếu nhiệt độ nằm ngoài khoảng này, cây không cần phải tưới.

- Ngưỡng độ ẩm thấp (60%) và độ ẩm cao (80%) giúp xác định mức độ khô hạn của đất. Nếu độ ẩm thấp hơn ngưỡng cài đặt (dưới 60%), hệ thống sẽ bật tưới nước tự động.

- Khi nhiệt độ và độ ẩm nằm trong khoảng an toàn và độ ẩm thấp hơn ngưỡng yêu cầu, hệ thống sẽ tự động kích hoạt relay để bật bơm nước, đảm bảo rằng cây luôn được tưới đủ nước khi cần thiết.

**3. Tưới nước tự động**

- Khi độ ẩm thấp và điều kiện nhiệt độ phù hợp, hệ thống chuyển sang chế độ tưới nước tự động. Trong chế độ này, relay sẽ được kích hoạt để bật máy bơm và bắt đầu tưới nước.

- Máy bơm sẽ hoạt động trong một khoảng thời gian nhất định (mặc định là 5 giây). Sau khi hoàn thành tưới nước, hệ thống chuyển về trạng thái bình thường (Normal) và relay sẽ tắt bơm.

- Màn hình OLED sẽ hiển thị trạng thái “Watering” khi hệ thống đang tưới nước, và trạng thái này cũng được cập nhật trên ứng dụng Blynk.

**4. Tưới nước thủ công (nút bấm)**

- Hệ thống hỗ trợ chế độ tưới nước thủ công thông qua nút bấm vật lý được kết nối với ESP32. Khi người dùng nhấn nút, hệ thống sẽ bỏ qua điều kiện cảm biến và kích hoạt bơm nước ngay lập tức, cho phép người dùng điều khiển tưới nước bằng tay.

- Tương tự như chế độ tự động, bơm nước sẽ hoạt động trong khoảng thời gian 5 giây sau khi nhấn nút, sau đó tự động tắt và chuyển về trạng thái bình thường.

**5. Tưới nước từ xa qua Blynk**

- Một trong những tính năng nổi bật của hệ thống là khả năng tưới nước từ xa thông qua ứng dụng Blynk. Trên ứng dụng Blynk, người dùng có thể nhấn nút điều khiển tưới nước thủ công mà không cần phải có mặt tại vườn.

- Khi người dùng nhấn nút tưới nước trên giao diện Blynk (được gán vào chân ảo V3), hệ thống sẽ nhận lệnh và kích hoạt chế độ tưới nước từ xa. Sau khi tưới nước hoàn thành, hệ thống tự động quay lại trạng thái bình thường và hiển thị kết quả trên ứng dụng.

- Trong suốt quá trình tưới nước, hệ thống cập nhật trạng thái “Blynk Watering” lên màn hình OLED và ứng dụng, giúp người dùng theo dõi quá trình tưới nước.

**6. Hiển thị thông tin lên màn hình OLED**

- Màn hình OLED trong hệ thống đóng vai trò hiển thị thông tin trực tiếp về nhiệt độ, độ ẩm, và trạng thái hệ thống. Các trạng thái như "Normal" (bình thường), "Watering" (đang tưới nước), "Manual Watering" (tưới thủ công), và "Blynk Watering" (tưới qua Blynk) đều được hiển thị trên màn hình OLED.

- Mỗi khi hệ thống thay đổi trạng thái, màn hình OLED sẽ cập nhật nội dung, cho phép người dùng biết hệ thống đang hoạt động trong chế độ nào.

**7. Điều khiển hệ thống thông qua ứng dụng Blynk**

- Giao diện điều khiển trên Blynk cho phép người dùng theo dõi các thông số nhiệt độ và độ ẩm được cập nhật từ cảm biến DHT11. Các giá trị này được truyền về Blynk liên tục qua WiFi.

- Người dùng có thể điều khiển quá trình tưới nước thông qua nút bấm trên giao diện Blynk. Điều này tạo ra sự linh hoạt và tiện lợi khi có thể điều khiển hệ thống từ xa mà không cần có mặt tại vườn.

- Blynk còn cung cấp các tính năng khác như hiển thị các thông số môi trường và trạng thái hoạt động của hệ thống, giúp người dùng dễ dàng quản lý hệ thống thông qua điện thoại.

**8. Xử lý đa nhiệm với FreeRTOS**

- **Hệ thống:** Sử dụng FreeRTOS để quản lý các tác vụ khác nhau như đo nhiệt độ và độ ẩm, điều khiển tưới nước, xử lý nút bấm, và nhận lệnh từ Blynk. FreeRTOS cho phép chạy nhiều tác vụ song song, đảm bảo hệ thống luôn hoạt động mượt mà mà không bị gián đoạn.

**- Các tác vụ được tách biệt và hoạt động độc lập:**

**-Tác vụ đọc DHT1:** Đo nhiệt độ và độ ẩm liên tục.

**- Tác vụ điều khiển tưới nước:** Kiểm tra điều kiện và bật tắt bơm nước.

**- Tác vụ đọc nút bấm:** Xử lý tín hiệu từ nút bấm vật lý.

**- Tác vụ Blynk:** Xử lý lệnh tưới nước từ ứng dụng Blynk.

**9. Tính năng bảo vệ hệ thống**

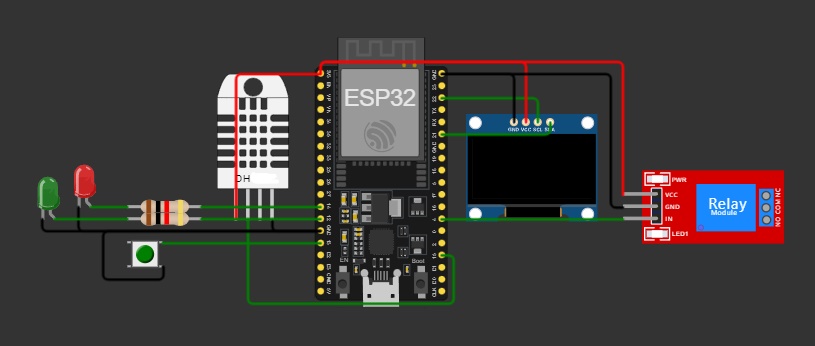
- Hệ thống được thiết kế với cơ chế tự động tắt bơm sau một khoảng thời gian nhất định (5 giây) để tránh việc tưới nước quá mức, đảm bảo tiết kiệm nước và bảo vệ cây trồng.

- Ngoài ra, các ngưỡng nhiệt độ và độ ẩm được thiết lập hợp lý để hệ thống chỉ tưới nước khi thật sự cần thiết, tránh lãng phí tài nguyên và đảm bảo cây trồng được chăm sóc đúng cách.

**IV. Kiến Trúc Hệ Thống**

**1. Sơ đồ phần cứng**

**a, Sơ đồ mô tả trên wokwi**



**1. Cảm biến DHT11 (màu trắng):**

* **Chân dữ liệu (Data) của DHT11:** Kết nối với một chân GPIO 15 của ESP32.
* **Chân VCC của DHT11:** Kết nối với nguồn điện 3.3V của ESP32.
* **Chân GND của DHT11:** Kết nối với chân GND của ESP32.

**2. Màn hình LCD (màu xanh):**

* **Các chân SDA và SCL của LCD:** Kết nối với các chân I2C của ESP32 ( GPIO 21 và 22).
* **Chân VCC của LCD:** Kết nối với nguồn điện 3.3V hoặc 5V của ESP32 .
* **Chân GND của LCD:** Kết nối với chân GND của ESP32.

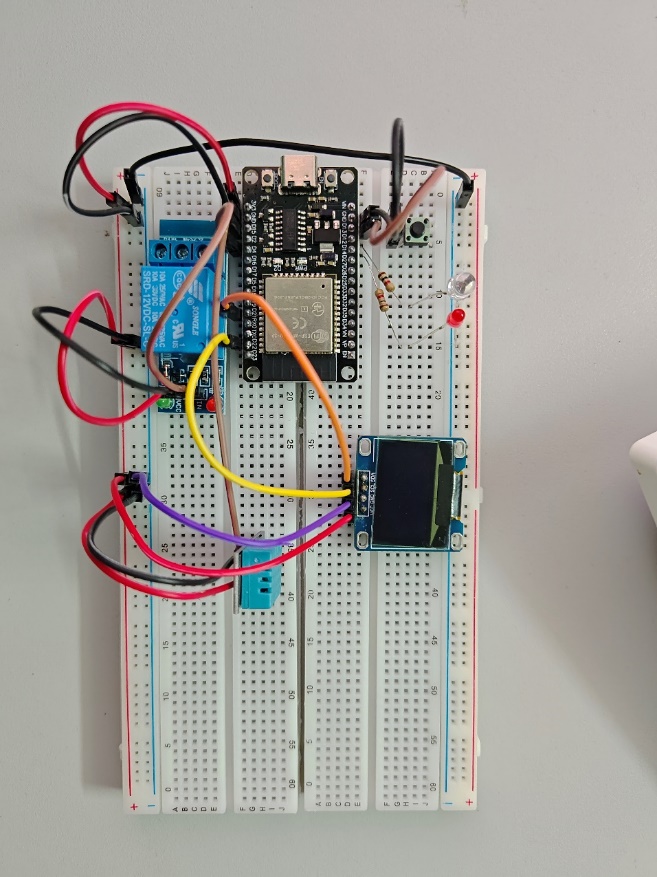
**3. Module Relay (màu đỏ):**

* **Chân IN của Relay:** Kết nối với một chân GPIO 4 của ESP32.
* **Chân VCC của Relay:** Kết nối với nguồn điện 5V của ESP32.
* **Chân GND của Relay:** Kết nối với chân GND của ESP32.

**4. Đèn LED (màu đỏ và xanh lá):**

* **Chân dương (+) của LED đỏ:** Kết nối với một chân GPIO 14 của ESP32 thông qua một điện trở.
* **Chân âm (-) của LED đỏ:** Kết nối với chân GND của ESP32.
* **Chân dương (+) của LED xanh lá:** Kết nối với một chân GPIO 12 của ESP32.
* **Chân âm (-) của LED xanh lá:** Kết nối với chân GND của ESP32 thông qua nút nhấn (màu xanh lá).

**b, Sơ đồ thực tế**



## **2. Chi tiết chức năng phần mềm**

## **2.1. Khai báo cấu hình Blink**



**Đầu tiên, mã nguồn khai báo các thông tin cần thiết để kết nối với ứng dụng Blynk:**

* **BLYNK\_TEMPLATE\_ID**: Định danh của template trong ứng dụng Blynk, dùng để xác định dự án mà thiết bị đang kết nối.
* **BLYNK\_TEMPLATE\_NAME**: Tên của template, giúp người dùng dễ nhận diện dự án trong ứng dụng Blynk.
* **BLYNK\_AUTH\_TOKEN**: Mã xác thực giúp kết nối thiết bị với tài khoản Blynk của người dùng. Đây là thông tin bảo mật quan trọng để đảm bảo rằng chỉ thiết bị đã được xác thực mới có thể kết nối và điều khiển dự án.

## **2.2 . Khai báo thư viện**

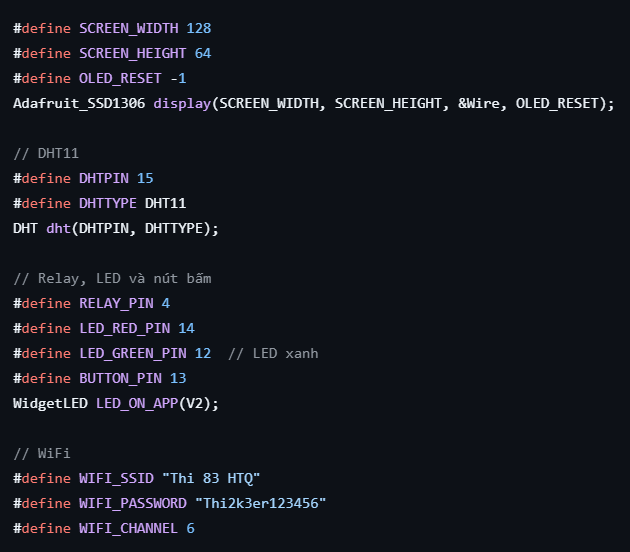


**Các thư viện được sử dụng**:

* **Adafruit\_SSD1306**: Thư viện này cho phép điều khiển màn hình OLED để hiển thị thông tin trạng thái hiện tại của hệ thống.
* **Adafruit\_Sensor và DHT\_U**: Hai thư viện này được sử dụng để tương tác với cảm biến DHT11, giúp đo lường nhiệt độ và độ ẩm trong môi trường.
* **Wire**: Thư viện này hỗ trợ giao tiếp I2C, thường được sử dụng cho các cảm biến và thiết bị ngoại vi.
* **freertos/FreeRTOS.h và freertos/task.h**: Thư viện hỗ trợ lập trình đa tác vụ với FreeRTOS, cho phép thực hiện nhiều tác vụ đồng thời.
* **WiFi.h và WiFiClient.h**: Các thư viện này cho phép ESP32 kết nối với mạng WiFi để giao tiếp với ứng dụng Blynk và các dịch vụ Internet khác.
* **BlynkSimpleEsp32.h**: Thư viện này giúp thiết lập kết nối giữa ESP32 và nền tảng Blynk, cho phép điều khiển từ xa và theo dõi trạng thái qua ứng dụng di động.

### **2.3 Khai báo các biến và cấu hình phần cứng**

Sau khi khai báo thư viện, mã nguồn tiếp tục khai báo các biến và cấu hình cho phần cứng của hệ thống như sau:



**- Kích thước màn hình OLED**: Các biến SCREEN\_WIDTH và SCREEN\_HEIGHT được thiết lập để xác định kích thước của màn hình OLED là 128x64 pixel, phù hợp với các loại màn hình nhỏ gọn phù hợp sử dụng trong dự án .



- **Cảm biến DHT11**:

* DHTPIN xác định chân GPIO mà cảm biến DHT11 được kết nối (chân 15).
* DHTTYPE xác định loại cảm biến được sử dụng là DHT11. Đối với cảm biến DHT11, nó có khả năng đo nhiệt độ từ 0 đến 50°C và độ ẩm từ 20 đến 90%.
* Biến DHT dht(DHTPIN, DHTTYPE) khởi tạo đối tượng cảm biến để sử dụng các phương thức đọc dữ liệu từ cảm biến.



- **Relay, LED và nút bấm**:

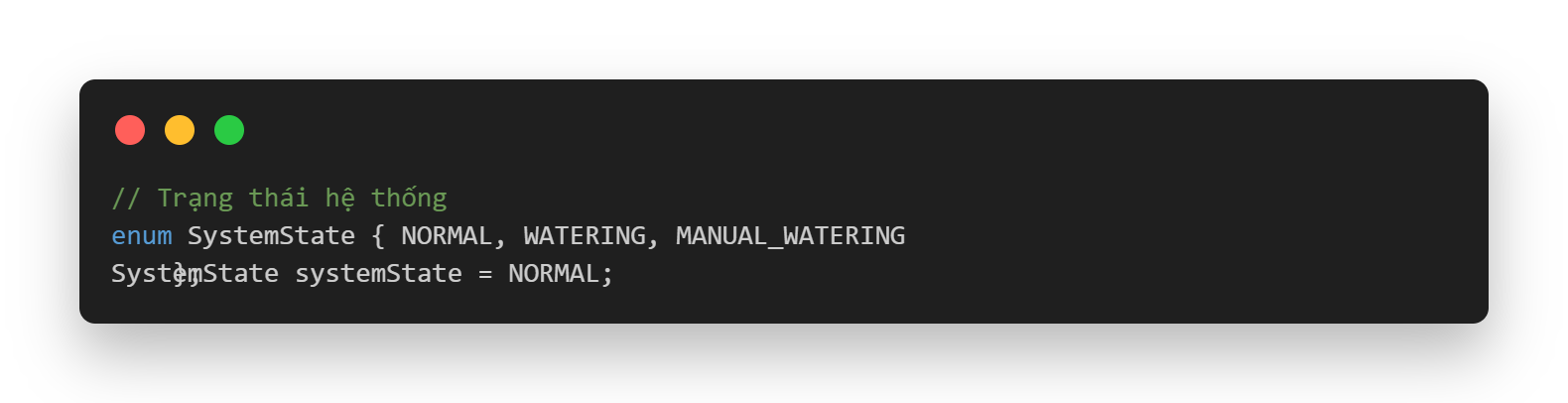
* RELAY\_PIN, LED\_RED\_PIN, LED\_GREEN\_PIN, và BUTTON\_PIN được định nghĩa cho các chân GPIO khác nhau, giúp điều khiển các thiết bị như bơm nước (relay), LED báo trạng thái và nút bấm cho chế độ tưới thủ công.
* WidgetLED LED\_ON\_APP(V2) cho phép điều khiển LED từ ứng dụng Blynk để phản ánh trạng thái hoạt động.



- **Khai báo WiFi**:

* WIFI\_SSID và WIFI\_PASSWORD chứa thông tin xác thực cho mạng WiFi mà hệ thống sẽ kết nối. Ở đây Wifi sẽ có ID : “ Thi 83 HTQ ” và mật khẩu : “ Thi2k3er123456 ”
* WIFI\_CHANNEL chỉ định kênh WiFi (mặc định là 6) để tránh xung đột với các kênh khác trong môi trường.

### **2.4 Trạng thái hệ thống và ngưỡng cảm biến**

**- Trạng thái hệ thống :** 

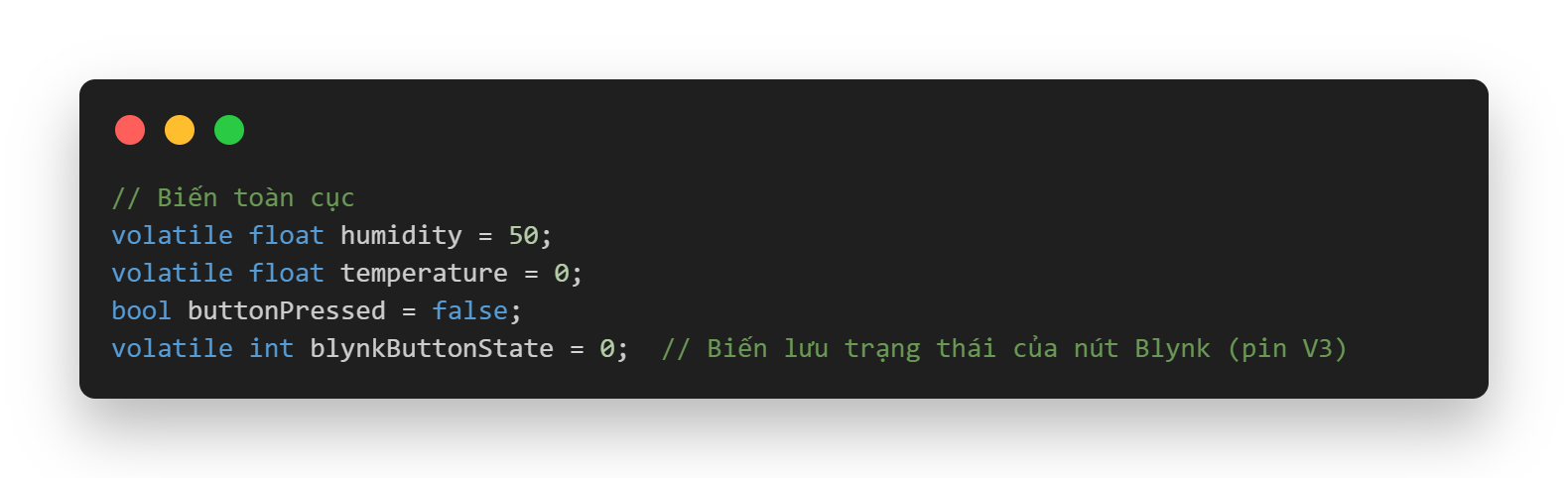
* **enum SystemState**: Định nghĩa các trạng thái của hệ thống, bao gồm:
* **NORMAL**: Trạng thái bình thường, hệ thống không đang tưới nước.
* **WATERING**: Hệ thống đang trong quá trình tưới nước.
* **MANUAL\_WATERING**: Hệ thống đang ở chế độ tưới thủ công thông qua nút bấm.
* **SystemState systemState = NORMAL**: Khởi tạo biến systemState với trạng thái mặc định là NORMAL, cho thấy hệ thống bắt đầu trong trạng thái bình thường.

**- Ngưỡng cảm biến:** 

* **volatile float temp\_threshold\_low**: Ngưỡng nhiệt độ tối thiểu, thiết lập là 25°C. Nếu nhiệt độ đo được thấp hơn ngưỡng này, hệ thống sẽ xem xét tưới nước.
* **volatile float temp\_threshold\_high**: Ngưỡng nhiệt độ tối đa, thiết lập là 35°C. Nếu nhiệt độ đo được vượt quá ngưỡng này, hệ thống sẽ không tưới nước.
* **volatile float humidity\_threshold\_low**: Ngưỡng độ ẩm tối thiểu, thiết lập là 60%. Nếu độ ẩm đo được thấp hơn ngưỡng này, hệ thống sẽ kích hoạt tưới nước.
* **volatile float humidity\_threshold\_high**: Ngưỡng độ ẩm tối đa, thiết lập là 80%. Nếu độ ẩm đo được vượt quá ngưỡng này, hệ thống sẽ không tưới nước.

### **2.5 Khai báo biến toàn cục và kết nối Blynk**

**- Khai báo biến toàn cục:**



 **volatile float humidity**: Biến lưu trữ độ ẩm hiện tại. Từ khóa volatile được sử dụng để cho biết rằng giá trị của biến này có thể được thay đổi bởi các tác vụ khác trong FreeRTOS, giúp tránh việc biên dịch tối ưu hóa sai.

 **volatile float temperature**: Tương tự, biến này lưu trữ nhiệt độ hiện tại của môi trường.

 **bool buttonPressed**: Biến boolean này theo dõi trạng thái của nút bấm. Nó sẽ là true nếu nút bấm được nhấn, ngược lại là false.

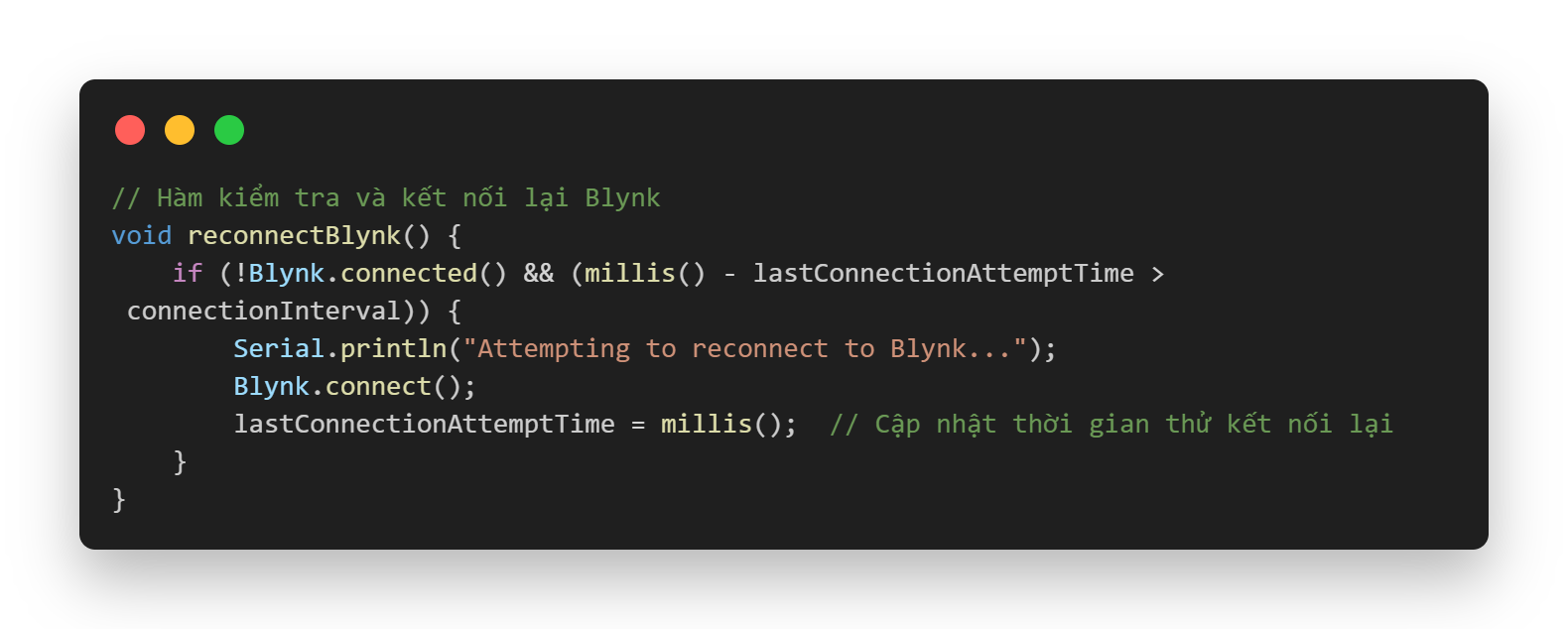
 **volatile int blynkButtonState**: Biến này lưu trạng thái của nút Blynk, được gán cho pin V3. Đây là nơi mà trạng thái nút bấm trên ứng dụng Blynk sẽ được lưu.

**- Khai báo biến kiểm tra và kết nối lại Blynk :**



 **unsigned long lastConnectionAttemptTime**: Biến này lưu thời điểm cuối cùng mà hệ thống cố gắng kết nối lại với Blynk. Nó được sử dụng để theo dõi thời gian giữa các lần kết nối.

 **unsigned long connectionInterval**: Biến này định nghĩa khoảng thời gian giữa các lần thử kết nối lại với Blynk, ở đây được thiết lập là 10 giây (10000 mili giây).

**- Hàm kiểm tra và kết nối lại Blynk** 

 **void reconnectBlynk()** : Hàm này có nhiệm vụ kiểm tra trạng thái kết nối với Blynk và thực hiện kết nối lại nếu cần thiết.

 **if (!Blynk.connected())**: Kiểm tra xem thiết bị có đang kết nối với Blynk hay không. Nếu không, nó sẽ tiếp tục kiểm tra thời gian giữa các lần kết nối.

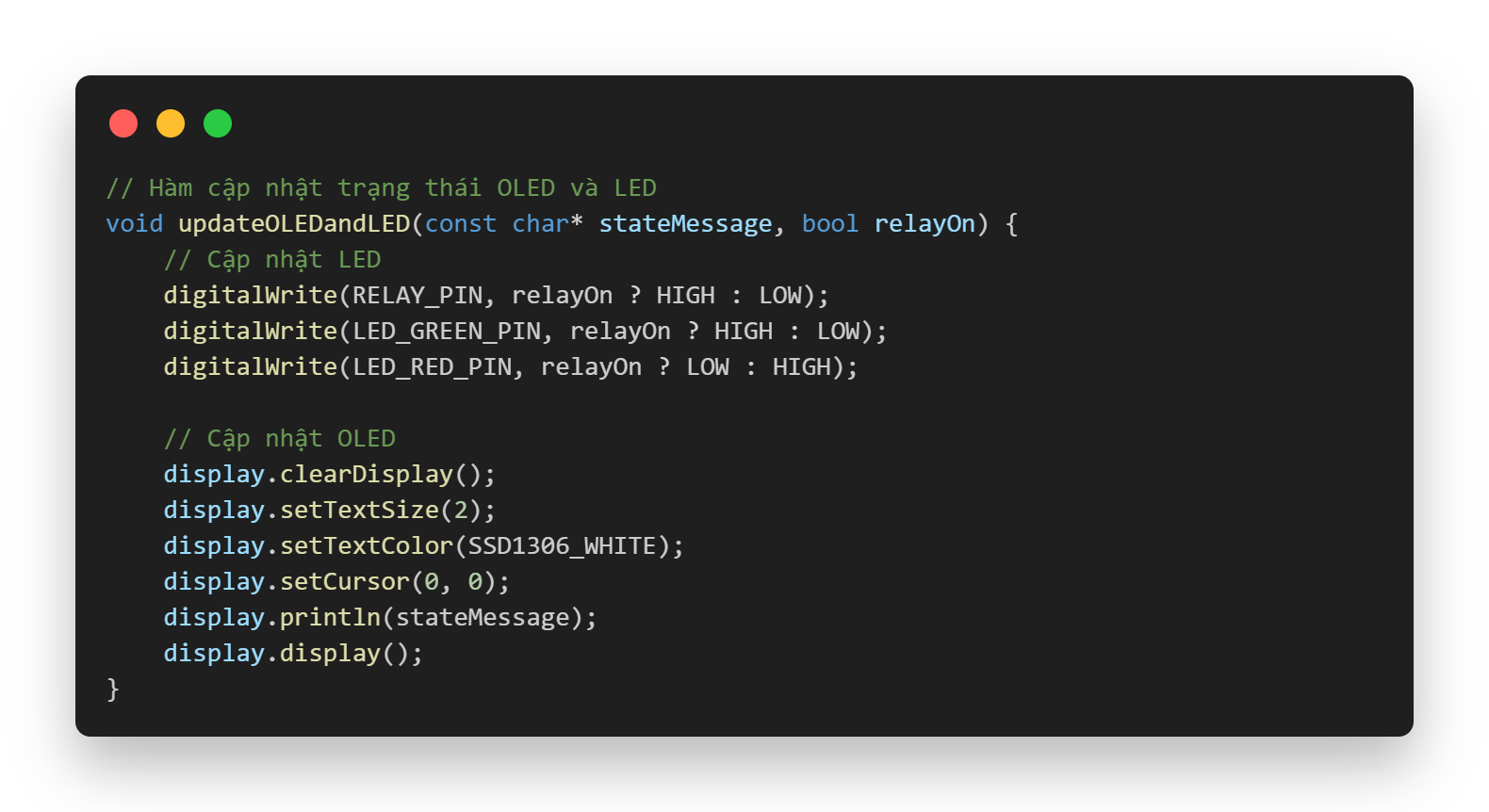
 **(millis() - lastConnectionAttemptTime > connectionInterval)**: So sánh thời gian hiện tại (sử dụng hàm millis()) với thời điểm cố gắng kết nối lại lần cuối. Nếu thời gian đã qua lớn hơn connectionInterval, nó sẽ thực hiện kết nối.

 **Blynk.connect():** Lệnh để thiết bị cố gắng kết nối lại với Blynk.

 **lastConnectionAttemptTime = millis():** Cập nhật thời gian thử kết nối lại mỗi khi thực hiện kết nối.

### **2.6. Cập nhật OLED và LED**

Hàm updateOLEDandLED được sử dụng để cập nhật trạng thái của màn hình OLED và các đèn LED dựa trên trạng thái của hệ thống. Hàm này nhận vào hai tham số: một thông điệp trạng thái và một giá trị boolean để xác định trạng thái của relay (điều khiển bơm tưới nước).



- **Điều khiển Relay và LED :**

* **digitalWrite(RELAY\_PIN, relayOn ? HIGH : LOW);**: Điều khiển pin của relay dựa trên tham số relayOn. Nếu relayOn là true, relay sẽ được bật (HIGH), ngược lại sẽ tắt (LOW).
* **digitalWrite(LED\_GREEN\_PIN, relayOn ? HIGH : LOW);**: Tương tự, LED xanh sẽ sáng khi relayOn là true và tắt khi là false.
* **digitalWrite(LED\_RED\_PIN, relayOn ? LOW : HIGH);**: Đèn LED đỏ sẽ sáng khi relayOn là false và tắt khi relayOn là true, nhằm chỉ ra rằng hệ thống đang hoạt động (tưới nước) hay không.

- **Cập nhật màn hình OLED**:

 **display.clearDisplay();**: Xóa nội dung hiện tại trên màn hình OLED để chuẩn bị cho thông điệp mới.

 **display.setTextSize(2);**: Thiết lập kích thước chữ cho văn bản hiển thị. Trong trường hợp này, kích thước là 2, giúp văn bản dễ đọc hơn.

 **display.setTextColor(SSD1306\_WHITE);**: Đặt màu chữ hiển thị là trắng (SSD1306\_WHITE).

 **display.setCursor(0, 0);**: Đặt vị trí con trỏ để bắt đầu viết văn bản từ góc trên bên trái của màn hình.

 **display.println(stateMessage);**: In thông điệp trạng thái được truyền vào hàm (stateMessage) lên màn hình OLED.

 **display.display();**: Cập nhật màn hình để hiển thị các thay đổi vừa thực hiện.

Hàm updateOLEDandLED giúp người dùng theo dõi trạng thái của hệ thống vườn thông minh một cách trực quan thông qua màn hình OLED và các đèn LED. Khi bơm tưới nước hoạt động, màn hình sẽ hiển thị thông điệp tương ứng, đồng thời đèn LED xanh sẽ sáng để báo hiệu. Điều này không chỉ nâng cao trải nghiệm người dùng mà còn giúp đảm bảo tính chính xác trong quá trình hoạt động của hệ thống.

### **2.7 Tác vụ đọc cảm biến DHT**

Hàm task\_readDHT22 được thiết kế để đọc dữ liệu từ cảm biến DHT11 (hoặc DHT22) để thu thập thông tin về độ ẩm và nhiệt độ trong môi trường vườn thông minh. Tác vụ này hoạt động trong một vòng lặp vô hạn, liên tục lấy mẫu dữ liệu từ cảm biến và gửi thông tin này đến ứng dụng Blynk để theo dõi từ xa.



**- Khởi tạo tác vụ:**



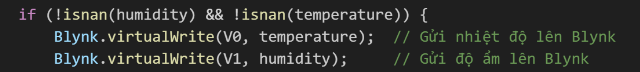
Tác vụ bắt đầu với một vòng lặp vô hạn while (true), cho phép chương trình liên tục đọc dữ liệu từ cảm biến mà không dừng lại.

**- Đọc dữ liệu từ cảm biến:**



Thực hiện việc đọc giá trị độ ẩm và nhiệt độ từ cảm biến DHT. Kết quả được lưu vào các biến toàn cục humidity và temperature.

**- Kiểm tra dữ liệu hợp lệ và gửi lên Blynk :**



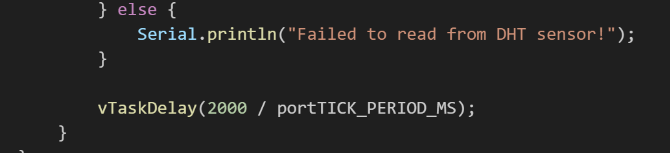
* Đoạn mã này sử dụng hàm isnan() để kiểm tra xem giá trị đọc từ cảm biến có hợp lệ hay không. Nếu một trong hai giá trị không hợp lệ (ví dụ: cảm biến không phản hồi), hàm sẽ in ra thông báo lỗi.
* Nếu dữ liệu hợp lệ, giá trị nhiệt độ sẽ được gửi lên pin ảo V0 và độ ẩm lên pin ảo V1 trong ứng dụng Blynk. Điều này giúp người dùng theo dõi các thông số môi trường từ xa.

- **Cập nhật hiển thị OLED**:



* Nếu trạng thái của hệ thống là NORMAL, hàm sẽ xóa nội dung hiện có trên màn hình OLED và sau đó cập nhật các giá trị nhiệt độ và độ ẩm.
* **Kiểm tra ngưỡng :**
* **temperature > temp\_threshold\_high**: Kiểm tra nếu nhiệt độ vượt quá ngưỡng cao. Nếu đúng, không cần tưới.
* **temperature < temp\_threshold\_low**: Kiểm tra nếu nhiệt độ thấp hơn ngưỡng thấp. Nếu đúng, không cần tưới.
* **humidity > humidity\_threshold\_high**: Kiểm tra nếu độ ẩm vượt quá ngưỡng cao. Nếu đúng, không cần tưới.

**- Xử lý lỗi và đọc lại :**



- Nếu không thể đọc dữ liệu từ cảm biến, hàm sẽ in ra thông báo lỗi trên console để thông báo cho người dùng.

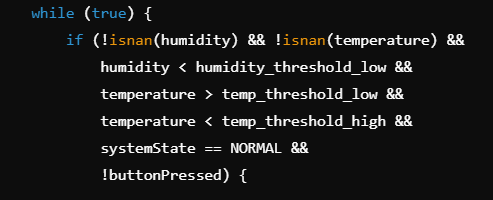
- Cuối cùng, hàm tạm dừng trong 2 giây trước khi quay lại đầu vòng lặp để đọc lại dữ liệu từ cảm biến.

### **2.8 Tác vụ điều khiển tưới nước tự động**

Tác vụ điều khiển tưới nước tự động, trong đó hệ thống sẽ tự động tưới nước cho cây khi các điều kiện nhất định được thỏa mãn.

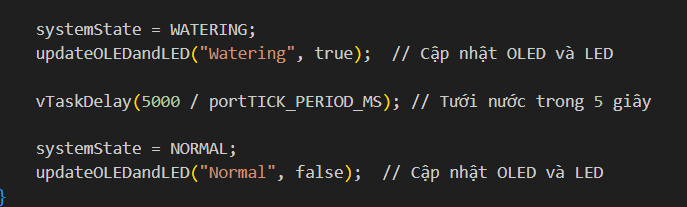


**- Kiểm tra điều kiện :**



* **!isnan(humidity) và** **!isnan(temperature)**: Đảm bảo rằng dữ liệu từ cảm biến không bị lỗi (có thể đọc được).
* **humidity < humidity\_threshold\_low:** Kiểm tra xem độ ẩm có thấp hơn ngưỡng quy định hay không.
* **temperature > temp\_threshold\_low && temperature < temp\_threshold\_high:** Đảm bảo nhiệt độ nằm trong khoảng an toàn.
* **systemState == NORMAL:** Hệ thống đang ở trạng thái bình thường, chưa thực hiện tưới.
* **!buttonPressed:** Đảm bảo nút bấm không được nhấn, tránh xung đột với các tác vụ thủ công.

**- Điều khiển tưới nước :**



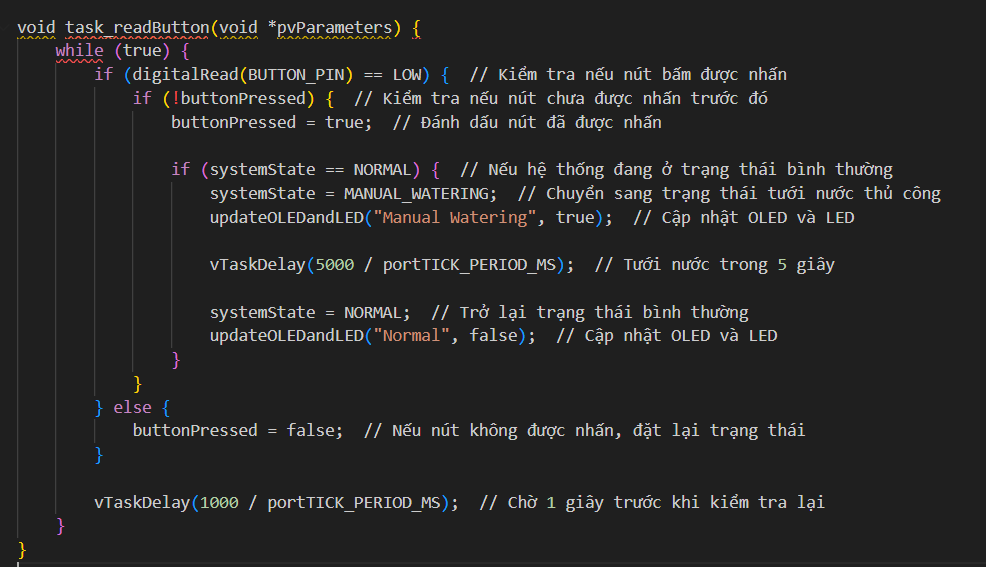
* Nếu tất cả điều kiện trên được thỏa mãn, trạng thái của hệ thống sẽ chuyển sang WATERING, và hàm updateOLEDandLED sẽ được gọi để cập nhật thông điệp trên màn hình OLED và trạng thái của các đèn LED.
* Hệ thống sẽ tưới nước trong vòng 5 giây bằng cách sử dụng vTaskDelay(5000 / portTICK\_PERIOD\_MS).
* **Quay lại trạng thái bình thường**: Sau khi tưới nước, trạng thái hệ thống được đặt lại thành NORMAL, và màn hình OLED sẽ hiển thị thông điệp bình thường.

**- Thời gian chờ :**



Cuối cùng, vTaskDelay(1000 / portTICK\_PERIOD\_MS) sẽ làm cho tác vụ chờ 1 giây trước khi lặp lại kiểm tra các điều kiện tưới nước một lần nữa.

### **2.9 Tác vụ đọc nút bấm để điều khiển tưới nước thủ công**

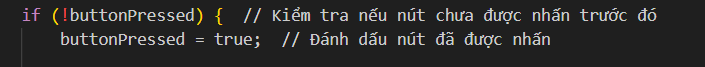
Tác vụ đọc nút bấm cho phép người dùng điều khiển tưới nước thủ công cho hệ thống vườn thông minh. 

**- Kiểm tra trạng thái nút bấm :**



Hệ thống sử dụng digitalRead(BUTTON\_PIN) == LOW để kiểm tra nếu nút bấm đã được nhấn. Khi nút nhấn, nó sẽ trả về trạng thái LOW.

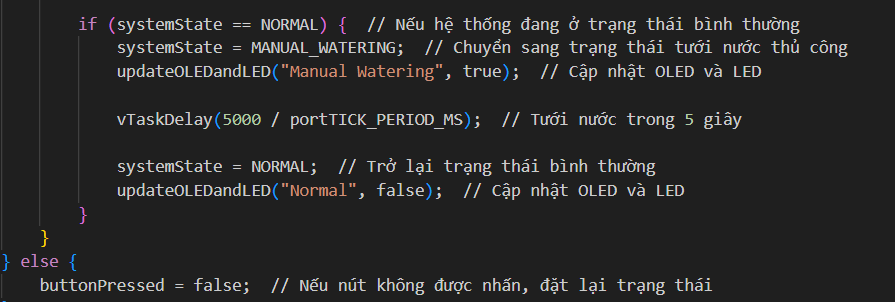
- Quản lý trạng thái nút :



 Nếu nút chưa được nhấn trước đó (!buttonPressed), hệ thống sẽ đánh dấu là nút đã nhấn bằng cách đặt biến buttonPressed thành true.

 Điều này giúp ngăn tình trạng nhấn nhiều lần liên tục khi người dùng giữ nút quá lâu.

- **Kích hoạt tưới nước thủ công** và hoàn tất tưới nước:



 Nếu hệ thống đang ở trạng thái bình thường (NORMAL), khi nhấn nút, hệ thống sẽ chuyển sang trạng thái MANUAL\_WATERING (tưới nước thủ công).

 Hàm updateOLEDandLED được gọi để hiển thị trạng thái hiện tại trên màn hình OLED và đèn LED sẽ báo hiệu tình trạng hoạt động của hệ thống.

 **Hoàn tất tưới nước**: Sau khi tưới nước trong 5 giây, hệ thống sẽ tự động trở lại trạng thái NORMAL và lại cập nhật OLED và LED để thông báo cho người dùng.

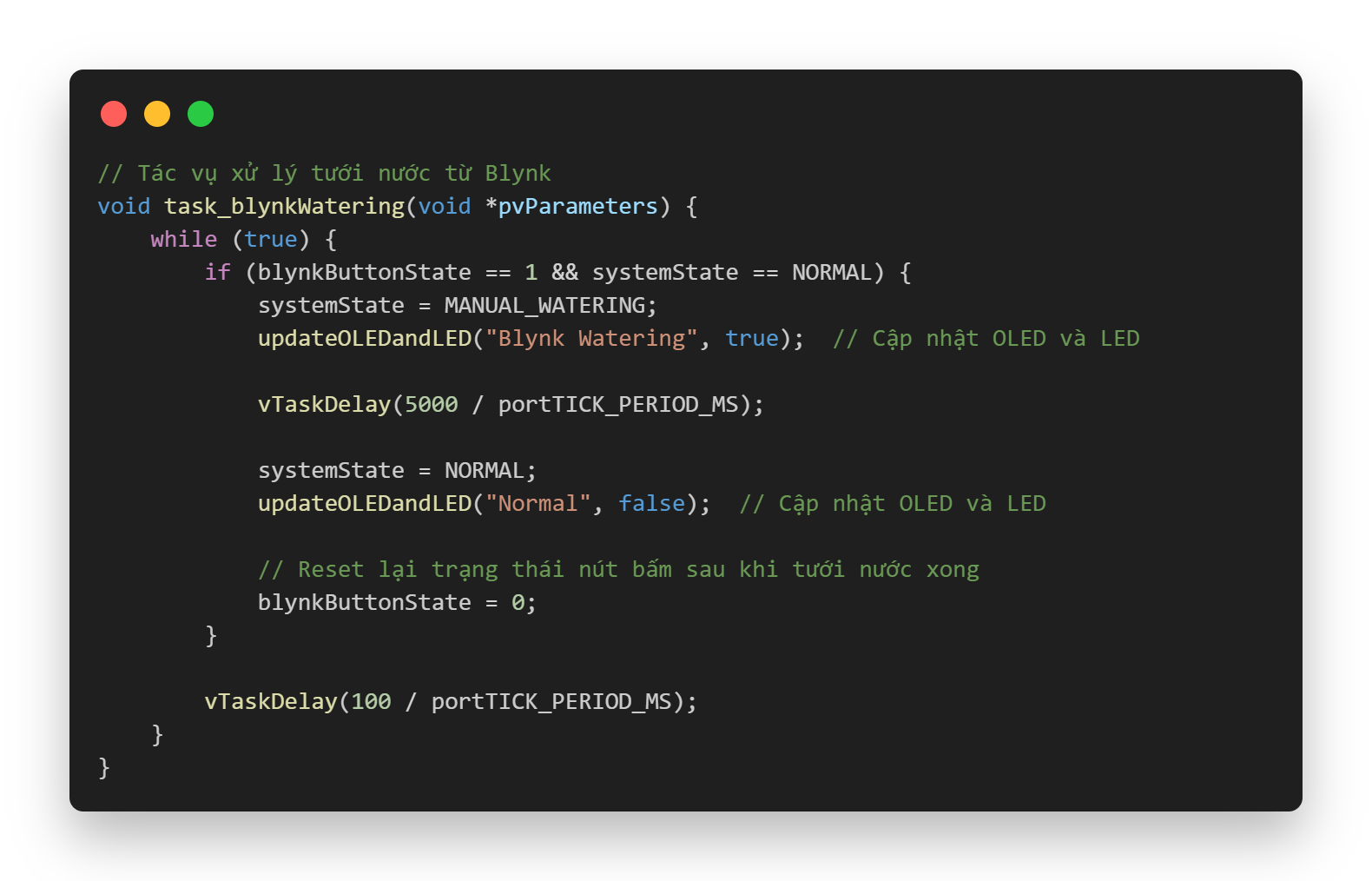
**- Chờ giữa các lần kiểm tra :**



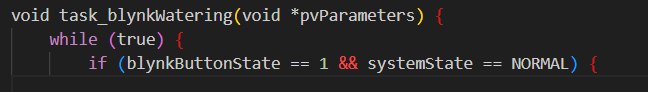
Hệ thống sẽ chờ 1 giây trước khi lặp lại việc kiểm tra trạng thái nút bấm.

### **2.10 Tác vụ xử lý tưới nước từ Blynk**

Tác vụ này cho phép người dùng dễ dàng điều khiển hệ thống tưới nước từ xa thông qua ứng dụng Blynk, trong khi vẫn đảm bảo cung cấp thông tin trạng thái rõ ràng qua màn hình OLED và LED. Điều này không chỉ tăng cường trải nghiệm người dùng mà còn giúp tối ưu hóa quy trình tưới nước trong vườn thông minh.

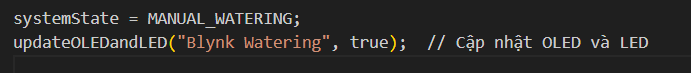
****

 Hàm task\_blynkWatering được thiết kế để xử lý việc tưới nước thông qua ứng dụng Blynk. Vòng lặp chính của hàm sẽ liên tục kiểm tra trạng thái của nút bấm từ Blynk và trạng thái của hệ thống để quyết định có nên thực hiện tưới nước hay không.



 **Kiểm tra trạng thái nút Blynk**: Nếu blynkButtonState bằng 1 (nghĩa là nút đã được nhấn từ ứng dụng Blynk) và systemState là NORMAL (hệ thống không đang tưới nước), hệ thống sẽ tiến hành tưới nước.

 **Chuyển trạng thái hệ thống**: Hệ thống sẽ chuyển sang trạng thái MANUAL\_WATERING để báo hiệu rằng quá trình tưới nước đã bắt đầu.

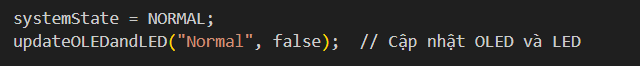


 **Cập nhật thông tin trên OLED và LED**: Hàm updateOLEDandLED sẽ được gọi với thông điệp "Blynk Watering", và giá trị true sẽ được truyền vào để bật bơm tưới nước. Điều này sẽ làm cho màn hình OLED hiển thị thông báo rằng hệ thống đang tưới nước và đèn LED sáng lên.

 **Thực hiện tưới nước trong 5 giây**: Hệ thống sẽ dừng lại trong 5000 ms (5 giây) để cho phép bơm hoạt động. Điều này đảm bảo rằng nước được tưới đủ thời gian.

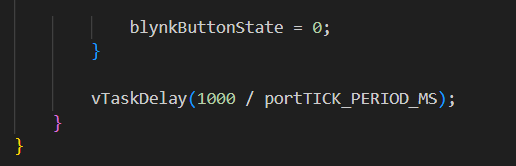


 **Quay lại trạng thái bình thường**: Sau khi thời gian tưới nước kết thúc, hệ thống sẽ trở về trạng thái NORMAL. Đồng thời, thông báo trên màn hình OLED sẽ được cập nhật lại để phản ánh trạng thái bình thường.



 **Reset trạng thái nút Blynk**: Cuối cùng, biến blynkButtonState sẽ được reset về 0 để chuẩn bị cho lần nhấn nút tiếp theo từ ứng dụng Blynk.

 **Tạm dừng một giây**: Sau mỗi lần kiểm tra trạng thái, hàm sẽ tạm dừng một giây trước khi quay lại vòng lặp, nhằm tiết kiệm tài nguyên.



**2.11 Cập nhật trạng thái nút Blynk (pin V3)**

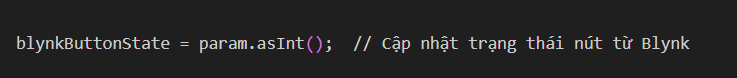
Hàm BLYNK\_WRITE(V3) giúp kết nối giữa ứng dụng Blynk và hệ thống phần cứng, cho phép người dùng điều khiển các chức năng của hệ thống vườn thông minh một cách dễ dàng.



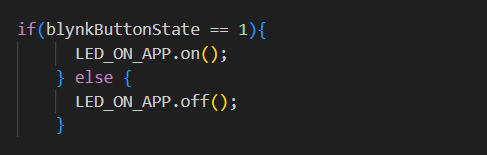
- Hàm BLYNK\_WRITE(V3) được sử dụng để xử lý sự kiện khi trạng thái của nút bấm trong ứng dụng Blynk thay đổi. Khi người dùng nhấn hoặc nhả nút trong ứng dụng Blynk, hàm này sẽ được gọi tự động để cập nhật trạng thái của nút bấm.



**- Khai báo hàm**: Hàm này định nghĩa một hàm xử lý cho pin V3 trên Blynk. Mỗi khi người dùng tương tác với nút trên ứng dụng, hàm sẽ được kích hoạt.



- **Cập nhật trạng thái nút**: Biến blynkButtonState sẽ được cập nhật bằng giá trị số nguyên từ tham số param. Nếu người dùng nhấn nút, giá trị này sẽ là 1; nếu người dùng nhả nút, giá trị sẽ là 0.

****

**- Kiểm tra trạng thái nút**: Hàm sẽ kiểm tra xem blynkButtonState có bằng 1 hay không. Nếu có, điều này có nghĩa là nút đã được nhấn.

**- Bật LED**: Nếu nút đã được nhấn, hàm sẽ gọi LED\_ON\_APP.on(); để bật đèn LED tương ứng trong ứng dụng Blynk, giúp người dùng biết rằng hệ thống đã nhận lệnh.

**- Trường hợp ngược lại**: Nếu trạng thái nút không phải là 1, tức là người dùng đã nhả nút.

**- Tắt LED**: Trong trường hợp này, hàm sẽ gọi LED\_ON\_APP.off(); để tắt đèn LED, cho biết rằng hệ thống đã nhận lệnh tắt.

**2.12 Hàm setup()**

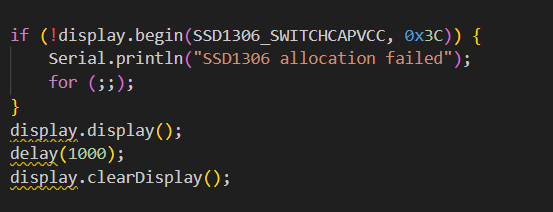
Hàm setup() được gọi một lần khi khởi động hệ thống. Hàm này có nhiệm vụ khởi tạo các thành phần của hệ thống, bao gồm giao tiếp nối tiếp, màn hình OLED, cảm biến, relay, LED, nút bấm, kết nối WiFi và Blynk, cũng như các tác vụ FreeRTOS.



**- Khởi động giao tiếp nối tiếp:** Thiết lập tốc độ baud là 115200, cho phép gửi và nhận dữ liệu qua cổng nối tiếp, giúp kiểm tra và gỡ lỗi.



**- Khởi động màn hình OLED:**



* **Khởi động màn hình**: Gọi hàm display.begin() để khởi động màn hình OLED với địa chỉ I2C là 0x3C.
* **Kiểm tra**: Nếu không khởi động thành công, in ra thông báo lỗi và dừng chương trình.
* **Hiển thị và làm sạch**: Hiển thị nội dung trên màn hình, chờ một giây rồi làm sạch màn hình

**- Khởi động cảm biến DHT và khởi tạo các pin :**

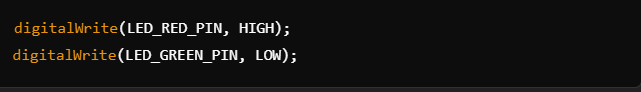


* **Khởi động cảm biến**: Gọi hàm dht.begin() để khởi động cảm biến nhiệt độ và độ ẩm DHT.



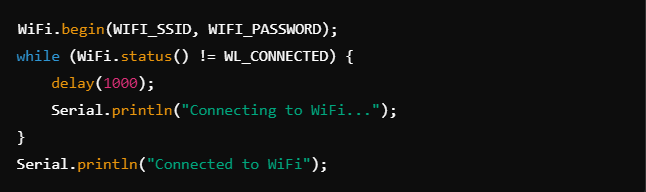
* **Khởi tạo pin**: Thiết lập các pin cho relay, đèn LED và nút bấm:
* RELAY\_PIN, LED\_RED\_PIN, và LED\_GREEN\_PIN được cấu hình là đầu ra (OUTPUT).
* BUTTON\_PIN được cấu hình là đầu vào với chế độ kéo lên (INPUT\_PULLUP).

**- Bật LED đỏ:**



* **Đặt trạng thái LED**: Bật đèn LED đỏ và tắt đèn LED xanh, sử dụng để báo hiệu trạng thái ban đầu.

**- Kết nối WiFi:**

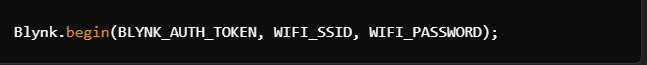


 **Kết nối WiFi**: Gọi WiFi.begin() với SSID và mật khẩu đã định nghĩa trước đó.

 **Kiểm tra kết nối**: Sử dụng vòng lặp while để kiểm tra trạng thái kết nối. Nếu chưa kết nối, in ra thông báo và chờ 1 giây.

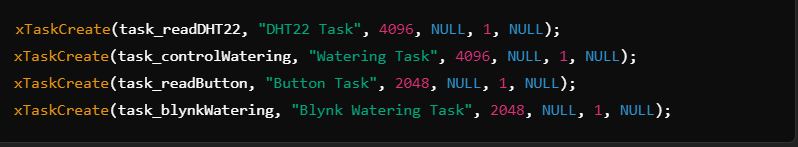
 **Thông báo kết nối**: Khi kết nối thành công, in ra thông báo.

**- Kết nối Blynk:**



* **Khởi động Blynk**: Sử dụng hàm Blynk.begin() để kết nối với dịch vụ Blynk bằng mã xác thực (BLYNK\_AUTH\_TOKEN) cùng với thông tin WiFi.

**- Tạo các tác vụ FreeRTOS:**

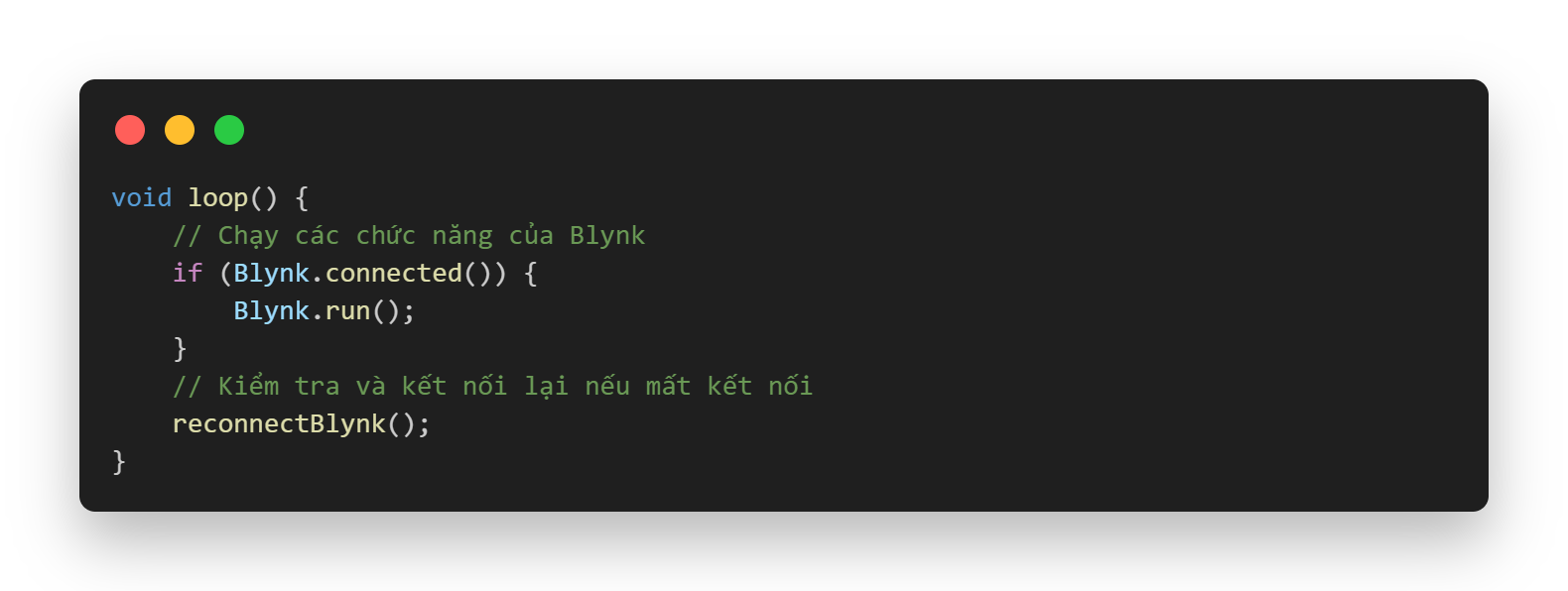


**Tạo tác vụ FreeRTOS**: Khởi tạo các tác vụ cho FreeRTOS để xử lý các chức năng riêng biệt:

 task\_readDHT22: Đọc dữ liệu từ cảm biến DHT22.

 task\_controlWatering: Điều khiển bơm tưới nước tự động.

 task\_readButton: Đọc trạng thái nút bấm.

 task\_blynkWatering: Xử lý tưới nước từ ứng dụng Blynk.

**+ Tác vụ đọc DHT22:**

****

 **Hàm thực thi**: task\_readDHT22

* Đọc dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22.

 **Tên tác vụ**: "DHT22 Task"

* Giúp dễ dàng xác định tác vụ.

 **Kích thước ngăn xếp**: 4096

* Duy trì đủ bộ nhớ cho các thao tác trong tác vụ.

 **Tham số**: NULL

* Không truyền tham số nào.

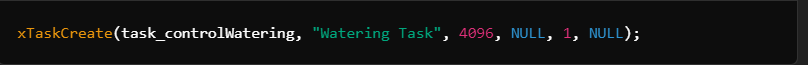
 **Ưu tiên**: 1

* Mức độ ưu tiên cho tác vụ.

 **ID tác vụ**: NULL

* Không lưu ID.

**+ Tác vụ điều khiển tưới nước:**



 **Hàm thực thi**: task\_controlWatering

* Tự động điều khiển hệ thống tưới nước dựa trên dữ liệu cảm biến.

 **Tên tác vụ**: "Watering Task"

* Dễ nhận diện trong hệ thống.

 **Kích thước ngăn xếp**: 4096

* Đủ bộ nhớ cho các thao tác cần thiết.

 **Tham số**: NULL

* Không có tham số nào được truyền.

 **Ưu tiên**: 1

* Chỉ định mức độ ưu tiên.

 **ID tác vụ**: NULL

* Không lưu ID.

**+ Tác vụ đọc nút bấm:**



 **Hàm thực thi**: task\_readButton

* Theo dõi trạng thái nút bấm để điều khiển tưới nước thủ công.

 **Tên tác vụ**: "Button Task"

* Giúp xác định tác vụ trong danh sách.

 **Kích thước ngăn xếp**: 2048

* Đủ bộ nhớ cho các thao tác.

 **Tham số**: NULL

* Không có tham số nào được truyền.

 **Ưu tiên**: 1

* Mức độ ưu tiên cho tác vụ.

 **ID tác vụ**: NULL

* Không lưu ID.

**+ Tác vụ xử lý tưới nước từ Blynk:**



 **Hàm thực thi**: task\_blynkWatering

* Nhận lệnh từ ứng dụng Blynk để điều khiển tưới nước.

 **Tên tác vụ**: "Blynk Watering Task"

* Dễ dàng nhận diện và theo dõi.

 **Kích thước ngăn xếp**: 2048

* Đủ bộ nhớ cho các thao tác cần thiết.

 **Tham số**: NULL

* Không truyền tham số.

 **Ưu tiên**: 1

* Mức độ ưu tiên cho tác vụ.

 **ID tác vụ**: NULL

* Không lưu ID.

### **2.13. Hàm loop()C:\Users\Admin\Downloads\superloop.png**

Hàm loop() là hàm chính trong chương trình Arduino, chạy liên tục sau khi hàm setup() đã hoàn tất.

Hàm loop() là phần chính của chương trình, đảm bảo rằng các chức năng của Blynk hoạt động liên tục và kiểm tra kết nối. Nếu mất kết nối, nó sẽ cố gắng khôi phục kết nối để duy trì hoạt động của hệ thống tưới nước thông minh.

**- Chạy các chức năng của Blynk:**



**Kiểm tra kết nối Blynk**: Đoạn mã này kiểm tra xem thiết bị có kết nối với máy chủ Blynk hay không. Nếu có, hàm Blynk.run() sẽ được gọi để xử lý các tác vụ liên quan đến Blynk, như cập nhật dữ liệu và xử lý các lệnh từ ứng dụng.

**- Kiểm tra và kết nối lại nếu mất kết nối:**



**Gọi hàm reconnectBlynk()**: Nếu thiết bị không còn kết nối với Blynk, hàm này sẽ được gọi để thử kết nối lại. Điều này đảm bảo rằng thiết bị luôn duy trì kết nối với mạng và máy chủ Blynk, giúp hệ thống hoạt động liên tục mà không bị gián đoạn.

# **V, Kết Luận**

Kết luận lại, đồ án "Hệ thống đo thông số nhiệt độ, độ ẩm và tưới cây tự động sử dụng FreeRTOS và ESP32" đã hoàn thành mục tiêu đề ra là xây dựng một hệ thống tưới tiêu tự động, hiệu quả và dễ sử dụng. Hệ thống đã được thử nghiệm và cho thấy khả năng hoạt động ổn định, đáp ứng được yêu cầu cơ bản về giám sát môi trường và điều khiển tưới tiêu.

Thông qua việc ứng dụng vi điều khiển ESP32 kết hợp với hệ điều hành thời gian thực FreeRTOS, hệ thống có khả năng xử lý đa nhiệm, thu thập dữ liệu từ các cảm biến và điều khiển bơm tưới một cách hiệu quả. Giao diện người dùng được thiết kế trực quan, cho phép người dùng dễ dàng theo dõi các thông số môi trường và điều khiển hệ thống.

Tuy nhiên, hệ thống vẫn còn tồn tại một số hạn chế nhất định. Ví dụ, việc chỉ sử dụng các cảm biến nhiệt độ và độ ẩm đất cơ bản có thể chưa phản ánh đầy đủ nhu cầu nước của cây trồng. Độ chính xác của cảm biến cũng có thể bị ảnh hưởng bởi các yếu tố môi trường.

Trong tương lai, hệ thống có thể được cải thiện bằng cách:

* **Tích hợp thêm các cảm biến**: Cảm biến ánh sáng, cảm biến mưa, cảm biến độ pH để thu thập đầy đủ thông tin về môi trường.
* **Ứng dụng trí tuệ nhân tạo (AI)**: Xây dựng mô hình AI để dự đoán nhu cầu nước của cây trồng, từ đó tối ưu hóa việc tưới tiêu.
* **Nâng cao tính năng điều khiển**: Cho phép điều khiển từ xa qua Internet, chia vùng tưới tiêu và điều khiển riêng biệt cho từng vùng.

Tóm lại, đồ án này đã góp phần ứng dụng công nghệ vào nông nghiệp, mang lại giải pháp tưới tiêu hiệu quả, tiết kiệm, góp phần nâng cao năng suất và chất lượng cây trồng. Hy vọng rằng với những hướng phát triển trong tương lai, hệ thống sẽ ngày càng hoàn thiện và được ứng dụng rộng rãi trong thực tiễn.

**Tài liệu tham khảo:**

* **Ứng dụng hệ điều hành FreeRTOS và vi điều khiển ESP32 trong hệ thống trồng rau thủy canh:** (<https://vjst.vn/Images/Tapchi/2019/3A/40-3A-2019.pdf>) Bài báo này trình bày về ứng dụng của Freetos và ESP32 trong hệ thông trồng rau thủy canh
* **Trang chủ FreeRTOS:** (<https://www.freertos.org/> )Cung cấp tài liệu, hướng dẫn và mã nguồn của FreeRTOS.
* **Trang chủ Espressif:** (<https://www.espressif.com/>) Cung cấp thông tin về ESP32, bao gồm tài liệu, phần mềm và công cụ phát triển.
* **Random Nerd Tutorials:** (<https://randomnerdtutorials.com/>) Website chứa nhiều hướng dẫn chi tiết về ESP32 và các ứng dụng IoT.
* **[Lập trình ESP32] Bài 44 - Hệ thống tưới cây tự động - YouTube:** (<https://m.youtube.com/watch?v=GNACny4xJVI>)