

**ĐẠI HỌC HUẾ**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC**



**ĐẠI HỌC HUẾ**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC**

**Tên đề tài:**

**Phát triển hệ thống cảnh báo ngập nước với ESP32**

**Học phần:** Phát triển ứng dụng IoT - Nhóm 4

**Giảng viên hướng dẫn:** ThS. Võ Việt Dũng

*Tp Huế, tháng 3 năm 2025*

## MỤC LỤC:

<b>LỜI MỞ ĐẦU</b>	1
1. Lý do chọn đề tài	1
2. Mục tiêu	1
3. Phương pháp	2
4. Bố cục bài tiểu luận	2
<b>NỘI DUNG</b>	4
<b>CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CẢNH BÁO NGẬP NƯỚC</b>	4
1.1. Vai trò của hệ thống cảnh báo ngập nước	4
1.2. Ứng dụng IoT trong cảnh báo ngập nước	4
1.3. Giới thiệu về ESP32	4
<b>CHƯƠNG 2: THIẾT KẾ HỆ THỐNG CẢNH BÁO NGẬP NƯỚC</b>	4
2.1. Yêu cầu hệ thống	4
2.2. Thành phần phần cứng	5
2.3. Thành phần phần mềm	5
2.4. Chi tiết về cảm biến và cách kết nối	5
2.5. Nguồn điện và bảo vệ hệ thống	6
<b>CHƯƠNG 3: NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG</b>	6
3.1. Quy trình hoạt động	6
3.2. Sơ đồ khối hệ thống	8
3.3. Xử lý dữ liệu chi tiết	8
<b>CHƯƠNG 4: MÔ PHỎNG HỆ THỐNG TRÊN WOKWI</b>	9
4.1. Thiết lập sơ đồ trên Wokwi	9
4.2. Gửi tin nhắn qua Telegram	10
4.3. Quy trình thiết lập mô phỏng	10
<b>CHƯƠNG 5: ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN</b>	11
5.1. Ưu điểm của hệ thống	11
5.2. Hạn chế	11
5.3. Hướng phát triển	11
5.4. So sánh với các hệ thống hiện có	11
5.5. Khả năng mở rộng chi tiết	12
<b>CHƯƠNG 6: ỨNG DỤNG THỰC TẾ VÀ TRIỂN KHAI</b>	12
6.1. Ứng dụng trong đời sống	12
6.2. Quy trình triển khai thực tế	12

<b>6.3. Chi phí ước tính</b> .....	13
<b>KẾT LUẬN</b> .....	13
<b>TÀI LIỆU THAM KHẢO</b> .....	14

# LỜI MỞ ĐẦU

## 1. Lý do chọn đề tài

Ngập lụt là một trong những thách thức lớn đối với nhiều quốc gia, đặc biệt tại Việt Nam – nơi có khí hậu nhiệt đới gió mùa và hệ thống sông ngòi dày đặc. Mỗi năm, các đợt mưa lớn, triều cường và lũ lụt gây thiệt hại nghiêm trọng về người và tài sản, ảnh hưởng trực tiếp đến đời sống của hàng triệu người dân. Theo thống kê của Tổng cục Phòng chống thiên tai Việt Nam, chỉ riêng năm 2024, lũ lụt đã gây thiệt hại hàng nghìn tỷ đồng và làm hàng chục người thiệt mạng. Trong bối cảnh biến đổi khí hậu ngày càng phức tạp, việc phát triển các hệ thống cảnh báo sớm trở thành nhu cầu cấp thiết để giảm thiểu rủi ro.

Công nghệ Internet vạn vật (IoT) đang mở ra những giải pháp đột phá trong việc giám sát và quản lý thiên tai. Với sự phổ biến của các vi điều khiển như ESP32 – một thiết bị nhỏ gọn, giá rẻ và tích hợp Wi-Fi – việc xây dựng hệ thống cảnh báo ngập nước thông minh không còn là ý tưởng xa vời. Đề tài "Phát triển hệ thống cảnh báo ngập nước với ESP32" được chọn vì tính ứng dụng cao, khả năng triển khai thực tế và tiềm năng đóng góp vào công tác phòng chống thiên tai. Hơn nữa, việc sử dụng nền tảng mô phỏng Wokwi giúp đơn giản hóa quá trình thiết kế và kiểm tra, phù hợp với mục tiêu học tập và nghiên cứu.

## 2. Mục tiêu

Bài tiểu luận hướng đến các mục tiêu cụ thể sau:

- **Thiết kế lý thuyết:** Xây dựng một hệ thống cảnh báo ngập nước dựa trên ESP32, sử dụng cảm biến mực nước và các cảm biến phụ để phát hiện nguy cơ ngập.
- **Mô phỏng:** Triển khai hệ thống trên nền tảng Wokwi, tích hợp các dịch vụ như ThingSpeak, Blynk hoặc Telegram để gửi thông báo khi mực nước vượt ngưỡng.
- **Đánh giá:** Phân tích ưu điểm, hạn chế và tiềm năng ứng dụng thực tế của hệ thống, từ đó đề xuất hướng phát triển trong tương lai.

- **Nâng cao kiến thức:** Cung cấp cái nhìn sâu hơn về ứng dụng IoT trong quản lý thiên tai, đồng thời làm nền tảng cho các dự án thực tế sau này.

### 3. Phương pháp

Để thực hiện bài tiểu luận, các phương pháp sau được áp dụng:

- **Nghiên cứu tài liệu:** Thu thập thông tin từ các nguồn tài liệu kỹ thuật về ESP32, cảm biến mực nước, và các nền tảng IoT như ThingSpeak, Blynk, Telegram. Các tài liệu tham khảo bao gồm sách, bài báo khoa học và hướng dẫn từ cộng đồng lập trình IoT.
- **Phân tích lý thuyết:** Xây dựng nguyên lý hoạt động của hệ thống dựa trên đặc điểm kỹ thuật của ESP32 và các cảm biến, không bao gồm phần lập trình thực tế.
- **Mô phỏng trên Wokwi:** Thiết kế sơ đồ hệ thống trên nền tảng Wokwi, sử dụng các linh kiện có sẵn và mô phỏng dữ liệu ngẫu nhiên cho các cảm biến không được hỗ trợ (như cảm biến mưa).
- **Tổng hợp và đánh giá:** So sánh hệ thống với các giải pháp hiện có, đánh giá tính khả thi và đề xuất cải tiến dựa trên phân tích lý thuyết.

### 4. Bố cục bài tiểu luận

Bài tiểu luận được chia thành các chương chính như sau:

- **Chương 1: Tổng quan về hệ thống cảnh báo ngập nước**  
Giới thiệu vai trò của hệ thống cảnh báo ngập nước, ứng dụng IoT và đặc điểm của ESP32.
- **Chương 2: Thiết kế hệ thống cảnh báo ngập nước**  
Mô tả yêu cầu, thành phần phần cứng/phần mềm, chi tiết về cảm biến và cách kết nối.
- **Chương 3: Nguyên lý hoạt động của hệ thống**  
Trình bày quy trình thu thập, xử lý dữ liệu và cơ chế gửi thông báo qua các nền tảng.

- **Chương 4: Mô phỏng hệ thống trên Wokwi**  
Hướng dẫn thiết lập sơ đồ, kịch bản kiểm tra và cách gửi dữ liệu qua Telegram.
- **Chương 5: Đánh giá và hướng phát triển**  
Phân tích ưu điểm, hạn chế, so sánh với hệ thống khác và đề xuất cải tiến.
- **Chương 6: Ứng dụng thực tế và triển khai**  
Đề xuất các kịch bản ứng dụng, quy trình triển khai và ước tính chi phí.
- **Kết luận:** Tóm tắt nội dung, nhấn mạnh ý nghĩa và tiềm năng của hệ thống.

## **NỘI DUNG**

### **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CẢNH BÁO NGẬP NƯỚC**

#### **1.1. Vai trò của hệ thống cảnh báo ngập nước**

Ngập lụt thường xảy ra do mưa lớn, triều cường hoặc vỡ đập, gây ảnh hưởng nghiêm trọng đến đời sống và kinh tế. Hệ thống cảnh báo ngập nước giúp:

- Phát hiện sớm mực nước vượt ngưỡng nguy hiểm.
- Gửi thông báo kịp thời đến người dân qua các nền tảng trực tuyến.
- Hỗ trợ cơ quan chức năng trong việc lập kế hoạch ứng phó.

#### **1.2. Ứng dụng IoT trong cảnh báo ngập nước**

Internet vạn vật (IoT) cho phép kết nối các thiết bị vật lý với mạng Internet, tạo ra hệ thống giám sát và điều khiển từ xa. Trong bối cảnh cảnh báo ngập nước, IoT giúp:

- Thu thập dữ liệu từ cảm biến theo thời gian thực.
- Truyền dữ liệu đến máy chủ hoặc ứng dụng người dùng.
- Tự động hóa quá trình cảnh báo mà không cần can thiệp thủ công.

#### **1.3. Giới thiệu về ESP32**

ESP32 là vi điều khiển do Espressif Systems phát triển, tích hợp Wi-Fi và Bluetooth, phù hợp cho các ứng dụng IoT. Một số đặc điểm nổi bật:

- Bộ xử lý lõi kép Xtensa LX6.
- Hỗ trợ nhiều giao thức giao tiếp (I2C, SPI, UART).
- Tiêu thụ điện năng thấp, dễ tích hợp với cảm biến và module khác.

### **CHƯƠNG 2: THIẾT KẾ HỆ THỐNG CẢNH BÁO NGẬP NƯỚC**

#### **2.1. Yêu cầu hệ thống**

Hệ thống cần đáp ứng các yêu cầu sau:

- Đo lường mực nước chính xác bằng cảm biến mực nước.
- Gửi thông báo khi mực nước vượt ngưỡng cho phép.

- Tích hợp thêm các cảm biến phụ (nhiệt độ, độ ẩm) để cung cấp thông tin môi trường.
- Hỗ trợ mô phỏng trên Wokwi và gửi dữ liệu qua Telegram.

## **2.2. Thành phần phần cứng**

### **2.2.1. ESP32**

ESP32 đóng vai trò trung tâm, xử lý dữ liệu từ cảm biến và gửi thông báo qua Wi-Fi.

### **2.2.2. Cảm biến mực nước (HC-SR04)**

Cảm biến HC-SR04 là cảm biến siêu âm được sử dụng để đo khoảng cách từ cảm biến đến bề mặt nước. Dựa vào khoảng cách đo được, hệ thống có thể xác định mức nước hiện tại.

### **2.2.3. Cảm biến phụ**

- Cảm biến nhiệt độ và độ ẩm (DHT22): Cung cấp thông tin về điều kiện môi trường, hỗ trợ đánh giá nguy cơ ngập lụt.

### **2.2.4. Module cảnh báo**

- LED: Làm tín hiệu cảnh báo
- Buzzer: Mô phỏng âm thanh cảnh báo khi mực nước vượt ngưỡng.

## **2.3. Thành phần phần mềm**

### **2.3.1. Wokwi**

Wokwi là nền tảng mô phỏng IoT trực tuyến, hỗ trợ ESP32 và nhiều cảm biến. Hệ thống sẽ được thiết kế và kiểm tra trên Wokwi trước khi triển khai thực tế.

### **2.3.2. Dịch vụ gửi thông báo**

- **Telegram:** Gửi tin nhắn tức thời qua bot Telegram.

## **2.4. Chi tiết về cảm biến và cách kết nối**

### **2.4.1. Cảm biến mực nước (HC-SR04)**

Cảm biến HC-SR04 hoạt động dựa trên nguyên lý đo khoảng cách bằng sóng siêu âm. Cảm biến này có cấu tạo bao gồm:

- Chân Trigger (Tín hiệu kích hoạt): Gửi tín hiệu siêu âm để bắt đầu quá trình đo.



- Chân Echo (Tín hiệu phản hồi): Nhận tín hiệu sóng siêu âm phản hồi sau khi bị vật thể (mức nước) phản xạ lại.
- Chân VCC: Nguồn cung cấp (5V hoặc 3.3V tùy vào nguồn cấp).
- Chân GND: Nối đất.

#### 2.4.2. Cảm biến DHT22

Cảm biến DHT22 đo nhiệt độ và độ ẩm không khí, hữu ích trong việc đánh giá điều kiện môi trường liên quan đến ngập lụt:

- **DHT22:** Độ chính xác (nhiệt độ  $\pm 0.5^{\circ}\text{C}$ , độ ẩm  $\pm 2\%$ ), thích hợp cho ứng dụng thực tế. Cảm biến kết nối với ESP32 qua giao thức 1-Wire, sử dụng một chân GPIO (ví dụ: GPIO 4) và cần điện trở kéo lên (pull-up resistor)  $4.7\text{k}\Omega$  giữa chân VCC và Data.

#### 2.4.3. Module cảnh báo

- LED: Được nối với GPIO 2, sáng lên khi mực nước vượt ngưỡng.
- Buzzer: Nối với GPIO 5, phát âm thanh cảnh báo (tần số mô phỏng khoảng 500-1000 Hz).

### 2.5. Nguồn điện và bảo vệ hệ thống

Hệ thống sử dụng nguồn 3.3V từ ESP32 cho các cảm biến. Để bảo vệ:

- **Chống nước:** Các linh kiện cần được đặt trong hộp kín IP65 khi triển khai thực tế.
- **Ổn áp:** Dùng thêm module ổn áp nếu nguồn điện không ổn định.

## CHƯƠNG 3: NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG

### 3.1. Quy trình hoạt động

#### 1. Thu thập dữ liệu:

- Cảm biến mực nước (HC-SR04): Cảm biến đo khoảng cách từ cảm biến đến mặt nước và gửi tín hiệu phản hồi về ESP32. Dựa trên thời gian phản hồi, ESP32 tính toán được mức nước hiện tại.

- Cảm biến nhiệt độ và độ ẩm (DHT22): Cảm biến này đo nhiệt độ và độ ẩm trong môi trường xung quanh và gửi dữ liệu đến ESP32 để xử lý.

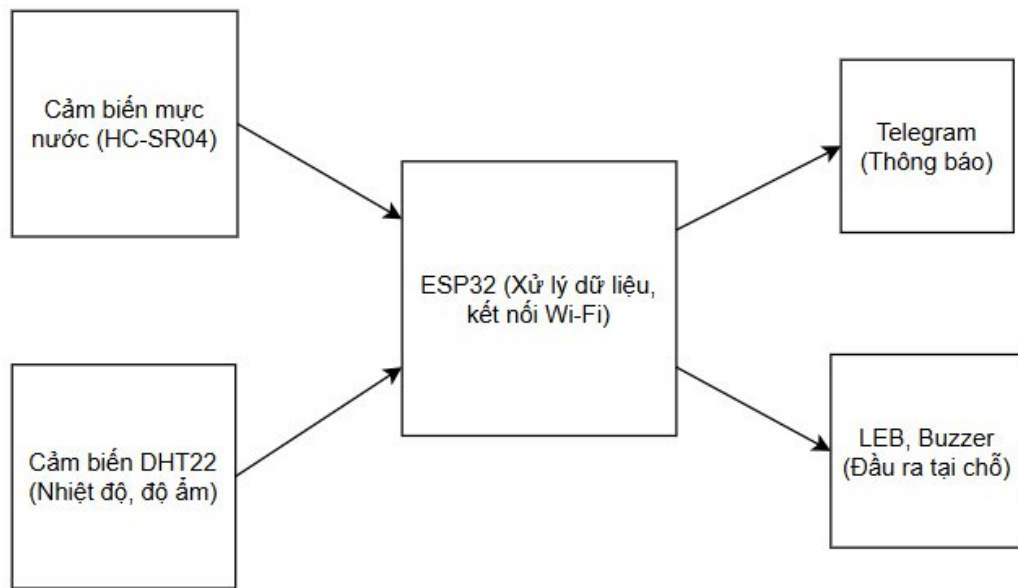
## **2. Xử lý dữ liệu:**

- Đo khoảng cách từ cảm biến HC-SR04:  
ESP32 sẽ kích hoạt cảm biến HC-SR04 bằng tín hiệu Trigger, sau đó đo thời gian phản hồi Echo từ cảm biến. Thời gian phản hồi này được dùng để tính toán khoảng cách giữa cảm biến và bề mặt nước.
- Chuyển đổi thời gian thành mức nước:  
Khoảng cách đo được từ cảm biến HC-SR04 sẽ được chuyển đổi thành mức nước trong bể chứa hoặc khu vực giám sát.

## **3. Cảnh báo:**

- Kích hoạt cảnh báo vật lý (LED và Buzzer):  
Khi mức nước vượt quá ngưỡng định sẵn, ESP32 sẽ kích hoạt các thiết bị cảnh báo vật lý để người sử dụng nhận biết.
- Gửi thông báo qua Wi-Fi:  
ESP32 sử dụng kết nối Wi-Fi để gửi thông báo cảnh báo đến các hệ thống giám sát hoặc ứng dụng di động.

### 3.2. Sơ đồ khối hệ thống



### 3.3. Xử lý dữ liệu chi tiết

#### 3.3.1. Đọc tín hiệu từ cảm biến mực nước

Tín hiệu từ cảm biến mực nước (HC-SR04) sẽ được ESP32 xử lý thông qua các bước sau:

1. **Đặt ngưỡng mực nước nguy hiểm:**

Để xác định mức nước nguy hiểm, người dùng sẽ thiết lập ngưỡng mực nước.

2. **Đo khoảng cách và chuyển đổi tín hiệu:**

ESP32 sử dụng cảm biến HC-SR04 để đo khoảng cách từ cảm biến đến mặt nước.

3. **So sánh giá trị đo được với ngưỡng mực nước:**

Giá trị ADC thu được sẽ được so sánh với ngưỡng mực nước nguy hiểm. Nếu giá trị đo được vượt qua ngưỡng này, điều này cho thấy mực nước đã đạt đến mức nguy hiểm.

4. **Kích hoạt cảnh báo:**

Nếu mực nước vượt qua ngưỡng, ESP32 sẽ kích hoạt các biện pháp cảnh báo

như bật LED, phát âm thanh từ buzzer, gửi thông báo qua các nền tảng như Wi-Fi hoặc Telegram Bot để cảnh báo người sử dụng.

### 3.3.2. Xử lý dữ liệu từ DHT22

Dữ liệu nhiệt độ và độ ẩm được đọc qua giao thức 1-Wire. ESP32 phân tích:

- Nhiệt độ cao bất thường ( $>35^{\circ}\text{C}$ ) có thể liên quan đến thời tiết cực đoan.
- Độ ẩm  $>90\%$  kết hợp với mưa lớn là dấu hiệu nguy cơ ngập.

### 3.4. Cơ chế gửi thông báo qua Telegram

- ESP32 gửi tin nhắn qua bot Telegram sử dụng API HTTPS.
- Nội dung mẫu: "Cảnh báo ngập nước! Mức nước: [giá trị], Thời gian: [timestamp]".

## CHƯƠNG 4: MÔ PHỎNG HỆ THỐNG TRÊN WOKWI

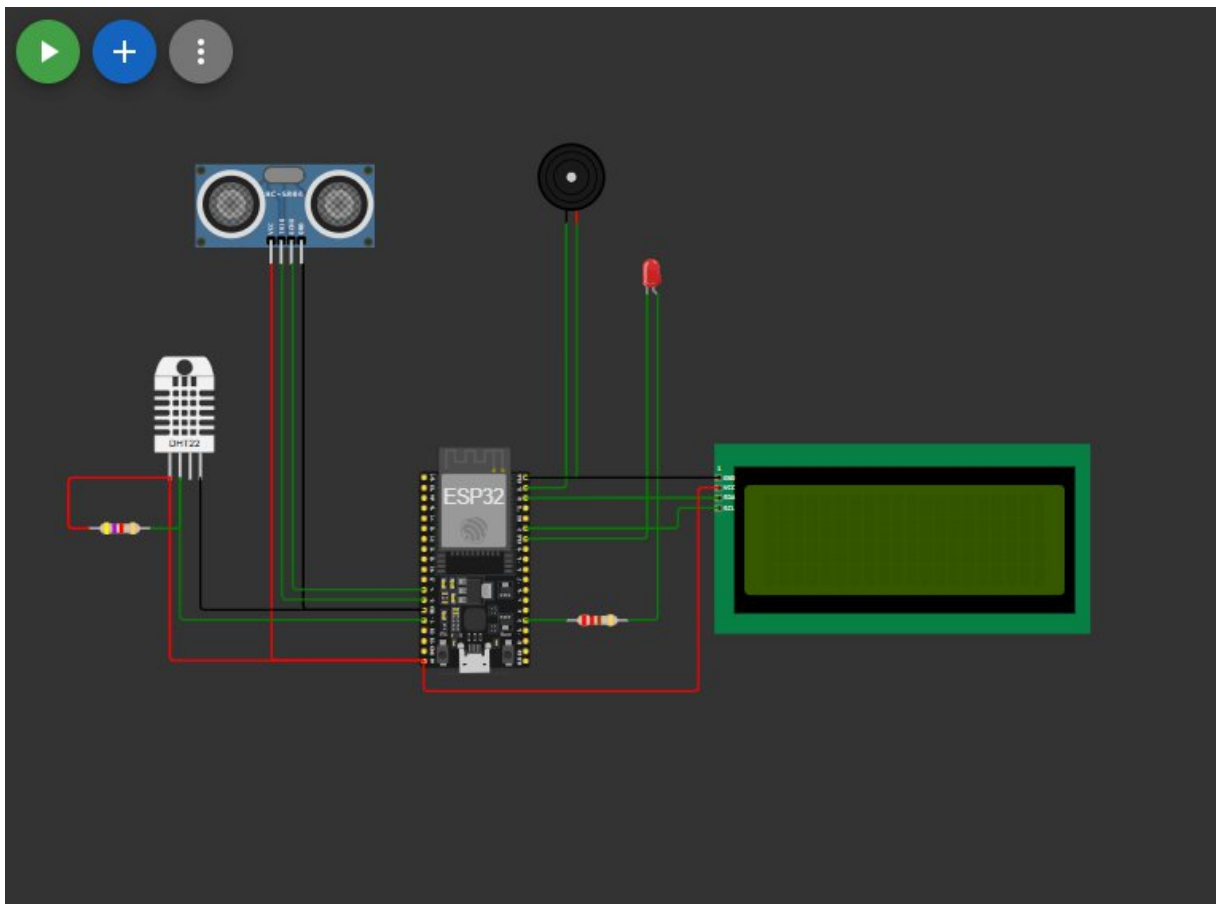
### 4.1. Thiết lập sơ đồ trên Wokwi

Sơ đồ mô phỏng bao gồm:

- **ESP32:** Kết nối với các cảm biến và module.
- **HC-SR04:**
  - Chân VCC: Kết nối với chân 5V của ESP32
  - Chân GND: Kết nối với chân GND của ESP32.
  - Chân TRIG: Kết nối với GPIO 12 của ESP32
  - Chân ECHO: Kết nối với GPIO 14 của ESP32.
- **Cảm biến DHT22:**
  - Chân VCC: Kết nối với chân 5V của ESP32
  - Chân GND: Kết nối với chân GND của ESP32.
  - Chân Data: Kết nối với GPIO13 của ESP32
- **LED:**
  - Chân dương: Kết nối với GPIO2 của ESP32 qua điện trở  $220\Omega$

- Chân âm: Kết nối với GND của ESP32
- **Buzzer:**
  - Chân dương: Kết nối với GPIO23 của ESP32
  - Chân âm: Kết nối với GND của ESP32
- **Màn hình LCD I2C 16x2:**
  - Chân VCC: Kết nối với 5V của ESP32
  - Chân GND: Kết nối với GND của ESP32
  - Chân SDA: Kết nối với **GPIO 22** của ESP32
  - Chân SCL: Kết nối với **GPIO 21** của ESP32
- **Wi-Fi:** ESP32 sử dụng mạng "Wokwi-GUEST" để gửi dữ liệu.

#### Sơ đồ Wokwi:



#### 4.2. Gửi tin nhắn qua Telegram

- ESP32 sử dụng bot Telegram để gửi tin nhắn dạng: "Cảnh báo: Mức nước vượt ngưỡng tại [thời gian]".

### 4.3. Quy trình thiết lập mô phỏng

#### 1. Tạo dự án trên Wokwi:

- Chọn ESP32 từ danh sách linh kiện.
- Thêm cảm biến DHT22, LED, Buzzer và cảm biến siêu âm HC-SR04.

#### 2. Kết nối linh kiện:

- GPIO 14 nối với cảm biến mực nước.
- GPIO 4 nối với chân Data của DHT22.
- GPIO 2 nối với LED qua điện trở 220Ω.
- GPIO 23 nối với Buzzer.

#### 3. Cấu hình Wi-Fi:

- Sử dụng SSID "Wokwi-GUEST" và để trống mật khẩu.

#### 4. Mô phỏng dữ liệu:

- Giá trị cảm biến mực nước thay đổi từ 0-4095 (tăng dần để kiểm tra ngưỡng).
- Giá trị mưa ngẫu nhiên được thêm vào kịch bản.

## CHƯƠNG 5: ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Ưu điểm của hệ thống

- Chi phí thấp, dễ triển khai nhờ ESP32 và Wokwi.
- Linh hoạt trong việc gửi thông báo Telegram.
- Có thể mở rộng với nhiều cảm biến khác.

### 5.2. Hạn chế

- Mô phỏng trên Wokwi không phản ánh hoàn toàn điều kiện thực tế.
- Phụ thuộc vào kết nối Wi-Fi, có thể bị gián đoạn nếu mất mạng.

### 5.3. Hướng phát triển

- Tích hợp năng lượng mặt trời để hệ thống hoạt động độc lập.

- Thêm cảm biến lưu lượng nước để dự đoán tốc độ ngập.
- Phát triển ứng dụng di động riêng thay vì dùng nền tảng thứ ba.

#### 5.4. So sánh với các hệ thống hiện có

- **Hệ thống truyền thống:** Dùng phao cơ học hoặc cảm biến đắt tiền, chi phí cao, khó bảo trì.
- **Hệ thống ESP32:** Giá rẻ, dễ mở rộng, tích hợp IoT, nhưng phụ thuộc vào mạng Internet.

#### 5.5. Khả năng mở rộng chi tiết

- **Thêm cảm biến lưu lượng:** Đo tốc độ dòng nước để dự đoán thời gian ngập.
- **Tích hợp GPS:** Xác định vị trí hệ thống, hữu ích khi triển khai nhiều điểm.
- **Học máy:** Phân tích dữ liệu lịch sử để dự báo ngập lụt chính xác hơn.

### CHƯƠNG 6: ỨNG DỤNG THỰC TẾ VÀ TRIỂN KHAI

#### 6.1. Ứng dụng trong đời sống

Hệ thống có thể được triển khai tại:

- **Khu dân cư:** Gắn ở cống thoát nước hoặc khu vực trũng để cảnh báo người dân.
- **Nông nghiệp:** Giám sát mực nước tại ruộng lúa, ao cá.
- **Giao thông:** Đặt tại các điểm ngập thường xuyên để cảnh báo tài xế.

#### 6.2. Quy trình triển khai thực tế

1. **Chọn vị trí lắp đặt:**
  - Khu vực dễ ngập, có nguồn điện hoặc pin dự phòng.
2. **Lắp đặt phần cứng:**
  - Đặt cảm biến mực nước ở độ sâu phù hợp, cố định ESP32 trong hộp chống nước.
3. **Kết nối mạng:**

- Sử dụng Wi-Fi tại chỗ hoặc module SIM (như SIM800L) nếu không có Wi-Fi.

#### **4. Kiểm tra và bảo trì:**

- Định kỳ kiểm tra cảm biến và nguồn điện để đảm bảo hoạt động ổn định.

### **6.3. Chi phí ước tính**

- ESP32: ~100.000 VNĐ.
- Cảm biến mực nước: ~20.000 VNĐ.
- DHT11: ~25.000 VNĐ.
- LED + Buzzer + linh kiện phụ: ~30.000 VNĐ.
- Màn hình LCD I2C 16x2: ~50.000 VNĐ.
- Tổng cộng: ~225.000 VNĐ (chưa tính hộp chống nước và nguồn điện).

## **KẾT LUẬN**

Hệ thống cảnh báo ngập nước dựa trên ESP32 là một giải pháp IoT hiệu quả, kết hợp cảm biến mực nước và các cảm biến phụ để giám sát và thông báo kịp thời. Việc mô phỏng trên Wokwi giúp kiểm tra tính khả thi trước khi triển khai thực tế. Với khả năng gửi dữ liệu qua ThingSpeak, Blynk hoặc Telegram, hệ thống mang lại sự tiện lợi và linh hoạt cho người dùng. Trong tương lai, hệ thống có thể được cải tiến để ứng dụng rộng rãi trong quản lý thiên tai và bảo vệ cộng đồng.



## TÀI LIỆU THAM KHẢO

1. VietnamPlus. (2025, January 4). *Thiên tai năm 2024: Tổn thương nghiêm trọng và những bài học lớn*. <https://www.vietnamplus.vn/thien-tai-nam-2024-ton-thuong-nghiem-trong-va-nhung-bai-hoc-lon-post1005874.vnp>
2. Espressif Systems. (2024). *ESP32 Series Datasheet* (Version 4.2). Espressif Systems.  
[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
3. Adafruit Industries. (2024). *DHT22 Temperature and Humidity Sensor Guide*. Adafruit Learning System. <https://learn.adafruit.com/dht>
4. Wokwi Team. (2025). *Wokwi ESP32 Simulator Documentation*. Wokwi.  
<https://docs.wokwi.com/guides/esp32>
5. Telegram. (2025). *Telegram Bot API Documentation*. Telegram.  
<https://core.telegram.org/bots/api>