

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



**TÊN ĐỀ TÀI TIỂU LUẬN:
PHÁT TRIỂN HỆ THỐNG
CẢNH BÁO NGẬP NƯỚC VỚI ESP32**

**PHÁT TRIỂN ỨNG DỤNG IOT – 2024-2025.2.TIN4024.005
GIẢNG VIÊN HƯỚNG DẪN: VÕ VIỆT DŨNG**

HUẾ, THÁNG 4 NĂM 2025

LỜI MỞ ĐẦU

Trong những năm gần đây, tình trạng ngập úng tại các khu vực đô thị và nông thôn đang ngày càng nghiêm trọng, đặc biệt là miền Trung mùa mưa bão. Ngập nước không chỉ gây thiệt hại về tài sản, ảnh hưởng đến đời sống sinh hoạt hàng ngày của người dân mà còn tiềm ẩn nhiều nguy cơ về sức khỏe và an toàn. Việc cảnh báo ngập nước giúp hạn chế các thiệt hại về tài sản, cũng như nhanh chóng chuẩn bị trước những nguy cơ tiềm tàng phía trước là hết sức cần thiết.

Trong thời đại công nghiệp hóa và hiện đại hóa, với những bước tiến vượt bậc về công nghệ, trong đó đặc biệt là công nghệ IoT (Internet of Things - Internet vạn vật), việc ứng dụng các vi điều khiển như ESP32 để xây dựng các hệ thống giám sát môi trường và cảnh báo sớm trở nên dễ tiếp cận và tiết kiệm chi phí hơn bao giờ hết. Đề tài này tập trung thiết kế và triển khai Hệ thống cảnh báo ngập nước sử dụng ESP32, kết hợp với cảm biến siêu âm HC-SR04 để đo mực nước, để hiển thị thông tin, đèn LED và còi để phát cảnh báo, đồng thời tích hợp ứng dụng Blynk giúp người dùng có thể giám sát và điều khiển hệ thống từ xa qua điện thoại.

Bài tiểu luận sẽ trình bày chi tiết từ cơ sở lý thuyết, thiết kế phần cứng, lập trình điều khiển đến việc đánh giá hiệu quả hoạt động của hệ thống. Hy vọng rằng sản phẩm này sẽ góp phần hỗ trợ người dân trong công tác phòng tránh và ứng phó kịp thời với các tình huống ngập lụt, đặc biệt là trong khu vực đô thị hoặc vùng trũng thấp.

MỤC LỤC

LỜI MỞ ĐẦU	1
MỤC LỤC	2
1. Giới thiệu chung.....	4
1.1. Khái quát về Internet of Things (IoT).....	4
1.2. Giới thiệu về ESP32.....	4
1.3. Cảm biến siêu âm HC-SR04	5
1.4. Màn hình OLED	5
1.5. Buzzer.....	6
1.6. Relay.....	7
2. Thiết kế hệ thống.....	7
2.1. Sơ đồ khối hệ thống	7
2.2. Nguyên lý hoạt động	8
2.3. Chu trình xử lý của ESP32	8
3. Mô hình thiết kế	10
3.1. Bảng sơ đồ kết nối phần cứng	10
3.2. Sơ đồ Wokwi	10
3.3. Giao diện Blynk	14
4. Lập trình cho hệ thống	15
4.1. Chương trình tổng quát	15
4.1.1. Mục tiêu của chương trình	15
4.1.2. Cấu trúc chương trình tổng quát.....	15
4.1.3. Luồng hoạt động của hệ thống	16
4.1.4. Mô hình dữ liệu kết nối Blynk	17
4.1.5. Ưu điểm của mô hình.....	17
4.2. Khai báo thông tin Blynk	17
4.3. Thư viện và cấu hình OLED	17

4.4. Khai báo chân phần cứng	18
4.5. Thông tin Wi-Fi	19
4.6. Hàm setup()	19
4.7. Hàm loop().....	20
4.8. Hiển thị lên màn hình OLED	20
4.9. Điều kiện cảnh báo.....	21
4.10. Chu kỳ lặp.....	22
4.11. Tổng kết chức năng chính	22
KẾT LUẬN	23
TÀI LIỆU THAM KHẢO	24

1. Giới thiệu chung

1.1. Khái quát về Internet of Things (IoT)

Internet of Things (IoT) là khái niệm chỉ mạng lưới các thiết bị vật lý được kết nối với nhau thông qua internet để thu thập và trao đổi dữ liệu. Các thiết bị IoT có thể là cảm biến, vi điều khiển, thiết bị điện tử, phần mềm,... và có khả năng nhận dạng, định vị, giám sát và điều khiển trong thời gian thực.

Ứng dụng của IoT ngày càng mở rộng trong nhiều lĩnh vực như:

- Nhà thông minh: điều khiển thiết bị điện tử từ xa, phát hiện rò rỉ nước, khí gas,...
- Y tế: giám sát sức khỏe từ xa, hỗ trợ người già và bệnh nhân mãn tính.
- Nông nghiệp thông minh: theo dõi độ ẩm đất, nhiệt độ, tưới tiêu tự động.
- Giao thông thông minh: giám sát phương tiện, quản lý bãi đỗ xe.
- Môi trường: giám sát chất lượng không khí, mực nước, nhiệt độ,...

Trong đề tài này, IoT sẽ đóng vai trò quan trọng trong việc giám sát mực nước và đưa ra cảnh báo sớm thông qua các cảm biến, bộ vi điều khiển và ứng dụng kết nối từ xa. Hệ thống có thể giúp người dùng theo dõi tình trạng mực nước ở các khu vực dễ ngập và phản ứng kịp thời khi có nguy cơ xảy ra ngập lụt.

1.2. Giới thiệu về ESP32

ESP32 là một vi điều khiển tích hợp Wi-Fi và Bluetooth do công ty Espressif Systems phát triển. Đây là phiên bản nâng cấp của ESP8266 với nhiều tính năng mạnh mẽ hơn như:

- CPU lõi kép 32-bit, tốc độ lên đến 240 MHz
- Tích hợp Wi-Fi chuẩn IEEE 802.11 b/g/n và Bluetooth v4.2
- Hỗ trợ nhiều giao tiếp như UART, SPI, I2C, PWM, ADC, DAC
- Có khả năng tiết kiệm năng lượng với nhiều chế độ ngủ

ESP32 rất phù hợp trong các dự án IoT, tự động hóa, hệ thống cảnh báo,... vì vừa mạnh mẽ, vừa tiết kiệm chi phí và dễ lập trình. Trong đề tài này, ESP32 giữ vai trò trung tâm trong việc thu thập dữ liệu từ cảm biến siêu âm HC-SR04, xử lý tín hiệu, hiển thị thông tin lên màn hình OLED, kích hoạt hệ thống cảnh báo như LED, còi, relay và gửi dữ liệu lên nền tảng Blynk để người dùng có thể giám sát từ xa bằng điện thoại.

1.3. Cảm biến siêu âm HC-SR04

Cảm biến siêu âm HC-SR04 là loại cảm biến dùng để đo khoảng cách bằng phương pháp không tiếp xúc vật lý. Nó hoạt động dựa trên nguyên lý phát và nhận sóng siêu âm để xác định khoảng cách đến vật thể phía trước.

Cảm biến gồm hai phần chính:

- Bộ phát (Trigger): Phát ra sóng siêu âm với tần số khoảng 40kHz.
- Bộ thu (Echo): Thu lại sóng siêu âm phản xạ từ vật thể.

Khi sóng siêu âm phát ra gặp vật cản, nó sẽ dội lại và được thu lại bởi bộ thu. Dựa trên thời gian từ khi phát đến khi thu lại sóng, vi điều khiển tính được khoảng cách đến vật thể.

Thông số kỹ thuật nổi bật:

- Khoảng cách đo: từ 2cm đến 400cm
- Độ phân giải: khoảng 3mm
- Điện áp hoạt động: 5V DC
- Giao tiếp dễ dàng với vi điều khiển như ESP32

Trong đề tài giám sát mực nước, HC-SR04 được dùng để đo khoảng cách từ cảm biến đến mặt nước, từ đó tính ra mực nước thực tế. Thông tin này sẽ được xử lý để điều khiển cảnh báo và hiển thị lên màn hình.

1.4. Màn hình OLED

Màn hình OLED (Organic Light-Emitting Diode) là một loại màn hình hiển thị sử dụng các hợp chất hữu cơ có khả năng phát sáng khi có dòng điện chạy qua. Mỗi điểm ảnh có thể tự phát sáng mà không cần đến đèn nền như LCD, giúp hiển thị hình ảnh với màu sắc rực rỡ, độ tương phản cao và tiết kiệm năng lượng.

Ưu điểm nổi bật:

- Hiển thị rõ nét, độ tương phản cao: Màu đen hiển thị sâu hơn, văn bản và số liệu hiển thị rõ ngay cả trong điều kiện ánh sáng yếu.
- Tiêu thụ điện năng thấp: Do không cần đèn nền, phù hợp cho các thiết bị IoT chạy bằng pin.

- Kích thước nhỏ gọn: Dễ dàng tích hợp vào các mạch điện tử, phù hợp với các dự án như giám sát môi trường, cảnh báo...

Trong đề tài giám sát mực nước và cảnh báo ngập úng, màn hình OLED (loại 0.96 inch, giao tiếp I2C) được sử dụng để hiển thị thông tin đo được từ cảm biến siêu âm HC-SR04, chẳng hạn như:

- Khoảng cách mực nước
- Trạng thái bơm (đang bật/tắt)
- Mức cảnh báo (an toàn, cảnh báo, nguy hiểm)

Điều này giúp người dùng có thể dễ dàng theo dõi và đánh giá tình hình mực nước trong thời gian thực ngay trên thiết bị.

1.5. Buzzer

Buzzer (còn gọi là còi điện tử) là một linh kiện phát ra âm thanh đơn giản, hoạt động dựa trên nguyên lý rung dao động bằng dòng điện. Nó được sử dụng rộng rãi trong các thiết bị cảnh báo nhờ đặc tính phát ra tiếng "bíp" rõ ràng và dễ nhận biết.

Có hai loại chính:

- Buzzer chủ động: chỉ cần cấp nguồn điện (thường là 3.3V hoặc 5V) là có thể phát ra âm thanh.
- Buzzer thụ động: cần có tín hiệu dao động (tín hiệu PWM) để tạo ra âm thanh.

Cấu tạo cơ bản gồm:

- Tấm rung bằng kim loại (thường là piezo)
- Mạch dao động (với loại chủ động)
- Vỏ nhựa bao bọc, có lỗ phát âm thanh

Trong hệ thống giám sát mực nước, buzzer chủ động được sử dụng để cảnh báo người dùng bằng âm thanh khi mực nước vượt quá mức quy định, giúp phản ứng nhanh trong các tình huống có nguy cơ ngập úng.

1.6. Relay

Relay là một công tắc điện tử hoạt động như một thiết bị trung gian điều khiển mạch điện công suất lớn bằng tín hiệu điện áp nhỏ. Bên trong relay bao gồm một cuộn dây và một tiếp điểm đóng/ngắt mạch.

Khi dòng điện chạy qua cuộn dây, relay tạo ra từ trường hút tiếp điểm, từ đó đóng hoặc ngắt mạch điện phía ngoài. Relay thường được sử dụng trong các ứng dụng điều khiển tự động, nơi cần tách biệt mạch điều khiển và mạch tải.

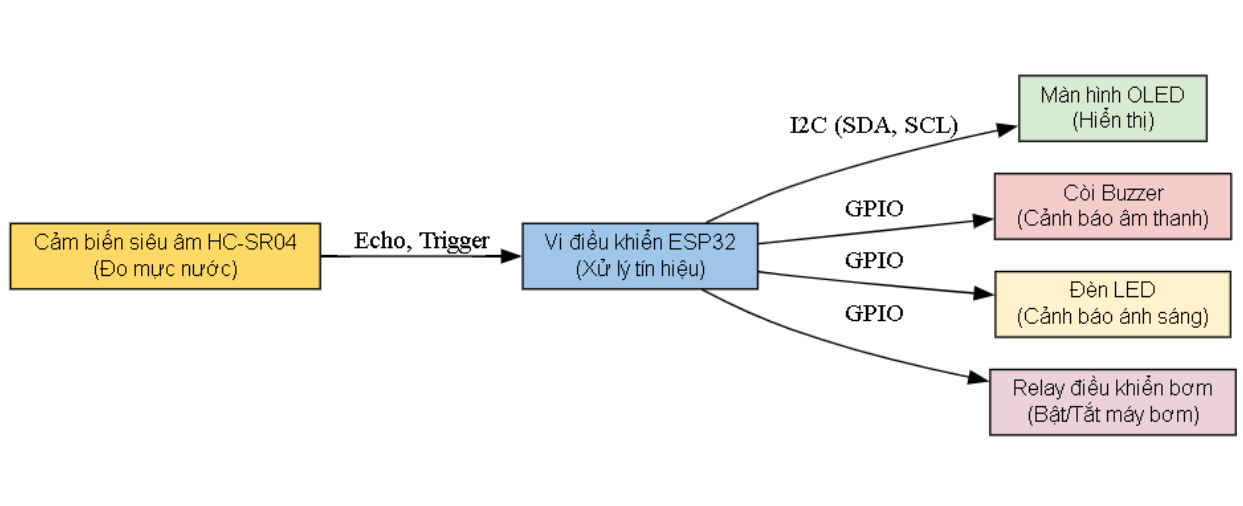
Trong hệ thống giám sát mực nước này, relay được sử dụng để điều khiển bơm nước:

- Khi mực nước vượt quá ngưỡng cho phép, ESP32 sẽ kích hoạt relay để tắt bơm, tránh tràn nước.
- Khi mực nước xuống thấp, relay được kích hoạt lại để bật bơm, duy trì mức nước ổn định.

Relay giúp hệ thống hoạt động an toàn và tự động hóa quá trình điều khiển bơm.

2. Thiết kế hệ thống

2.1. Sơ đồ khối hệ thống



2.2. Nguyên lý hoạt động

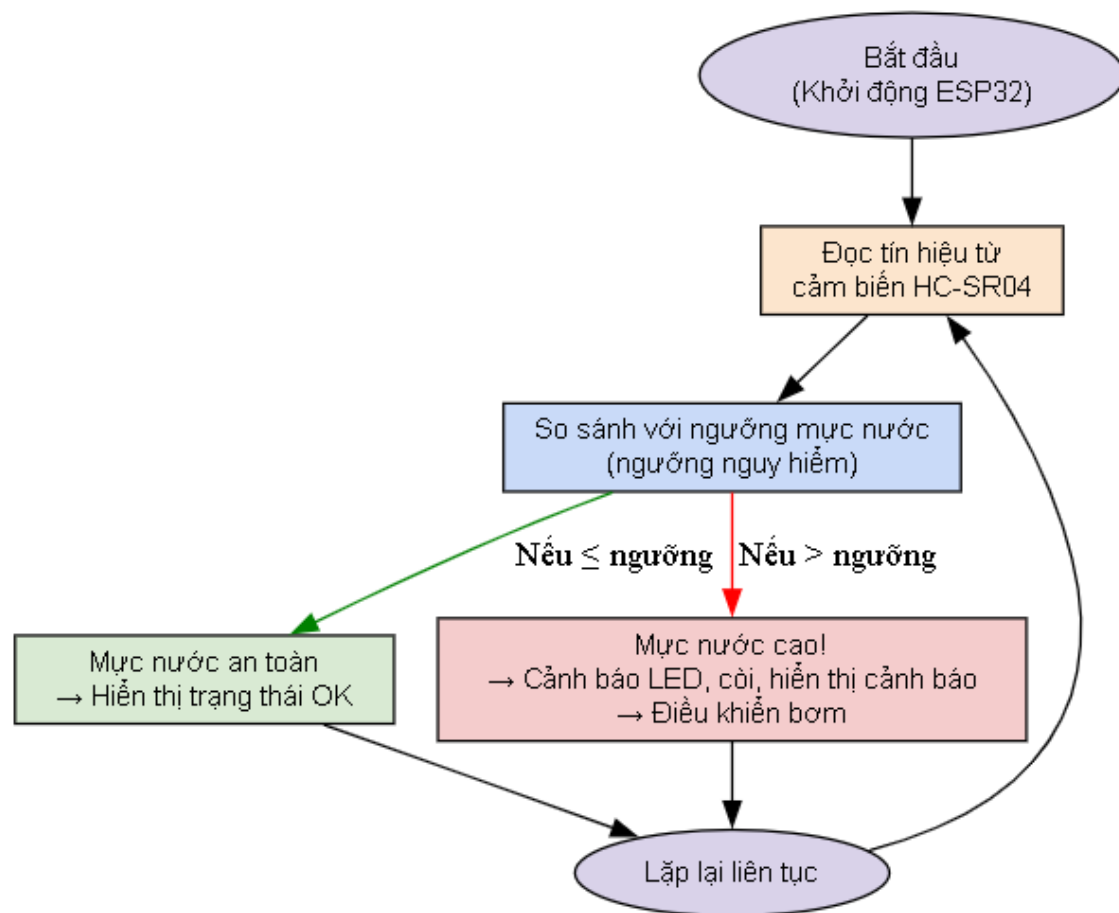
Cảm biến siêu âm HC-SR04 liên tục đo khoảng cách từ mặt cảm biến đến mặt nước để xác định mực nước hiện tại. Tín hiệu được gửi về ESP32 thông qua các chân digital (Trigger và Echo). Vi điều khiển ESP32 tính toán khoảng cách dựa trên thời gian phản hồi sóng siêu âm, từ đó xác định mực nước thực tế.

ESP32 so sánh mực nước với ngưỡng an toàn đã được thiết lập. Nếu mực nước vượt quá ngưỡng (nguy cơ ngập), hệ thống sẽ kích hoạt còi buzzer để phát âm thanh cảnh báo, đèn LED đỏ nhấp nháy để thu hút sự chú ý, đồng thời màn hình OLED hiển thị cảnh báo. Ngược lại, nếu mực nước ở mức an toàn, màn hình OLED hiển thị trạng thái "An toàn", các thiết bị cảnh báo sẽ tắt.

2.3. Chu trình xử lý của ESP32

ESP32 thực hiện chu trình xử lý chính theo các bước tuần hoàn liên tục, đảm bảo hệ thống giám sát mực nước theo thời gian thực:

1. Khởi tạo hệ thống: Khởi tạo các chân I/O, thiết bị ngoại vi như màn hình OLED, cảm biến HC-SR04, còi buzzer, LED và relay điều khiển bơm. Đồng thời thiết lập ngưỡng mực nước cảnh báo (ví dụ: 10 cm từ mặt cảm biến đến mặt nước).
2. Đọc tín hiệu cảm biến: ESP32 gửi xung đến chân Trigger của HC-SR04, sau đó đo thời gian phản hồi tại chân Echo để tính khoảng cách từ cảm biến đến mặt nước.
3. Xử lý và phân tích dữ liệu: Tính toán mực nước dựa trên khoảng cách đo được và chiều cao bể chứa. So sánh với ngưỡng cảnh báo đã thiết lập.
4. Cảnh báo người dùng:
 - Nếu mực nước vượt ngưỡng (nguy cơ ngập), hệ thống sẽ:
 - Bật còi buzzer, đèn LED, và relay để kích hoạt bơm thoát nước (nếu có).
 - Hiển thị cảnh báo trên màn hình OLED.
 - Nếu mực nước an toàn, hệ thống sẽ:
 - Hiển thị trạng thái "An toàn" trên OLED.
 - Tắt còi, LED và bơm.
5. Lặp lại: Chu trình quay lại bước 2 sau một khoảng trễ nhỏ (ví dụ: 500ms) để giám sát liên tục mực nước.



3. Mô hình thiết kế

3.1. Bảng sơ đồ kết nối phần cứng

Thiết bị	Chân Thiết bị	ESP32 GPIO	Chức năng
Cảm biến HC-SR04	VCC	5V	Cấp nguồn cho cảm biến
	GND	GND	Nối đất
	TRIG	GPIO5	Gửi xung siêu âm
	ECHO	GPIO18	Nhận xung phản hồi
Màn hình OLED SSD1306	VCC	5V	Cấp nguồn
	GND	GND	Nối đất
	SDA	GPIO21	Dữ liệu I2C
	SCL	GPIO22	Đồng hồ I2C
Relay điều khiển bơm	VCC	5V	Cấp nguồn
	GND	GND	Nối đất
	IN	GPIO16	Điều khiển bật/tắt bơm
Đèn LED chế độ	Cực dương (+)	GPIO12	Báo chế độ tự động
	Cực âm (-)	GND	Nối đất
Nút MODE (chuyển chế độ)	1 chân	GND	Nối đất (qua điện trở kéo lên nội bộ)
	Chân còn lại	GPIO32	Chuyển đổi chế độ
Nút PUMP (bật/tắt bơm thủ công)	1 chân	GND	Nối đất
	Chân còn lại	GPIO25	Điều khiển bơm khi ở chế độ thủ công
Buzzer cảnh báo	Cực dương (+)	GPIO14	Phát âm thanh khi cảnh báo
	Cực âm (-)	GND	Nối đất

3.2. Sơ đồ Wokwi

```
{
  "version": 1,
  "author": "Ngo Kim Nguyen",
  "editor": "wokwi",
  "parts": [
```

```

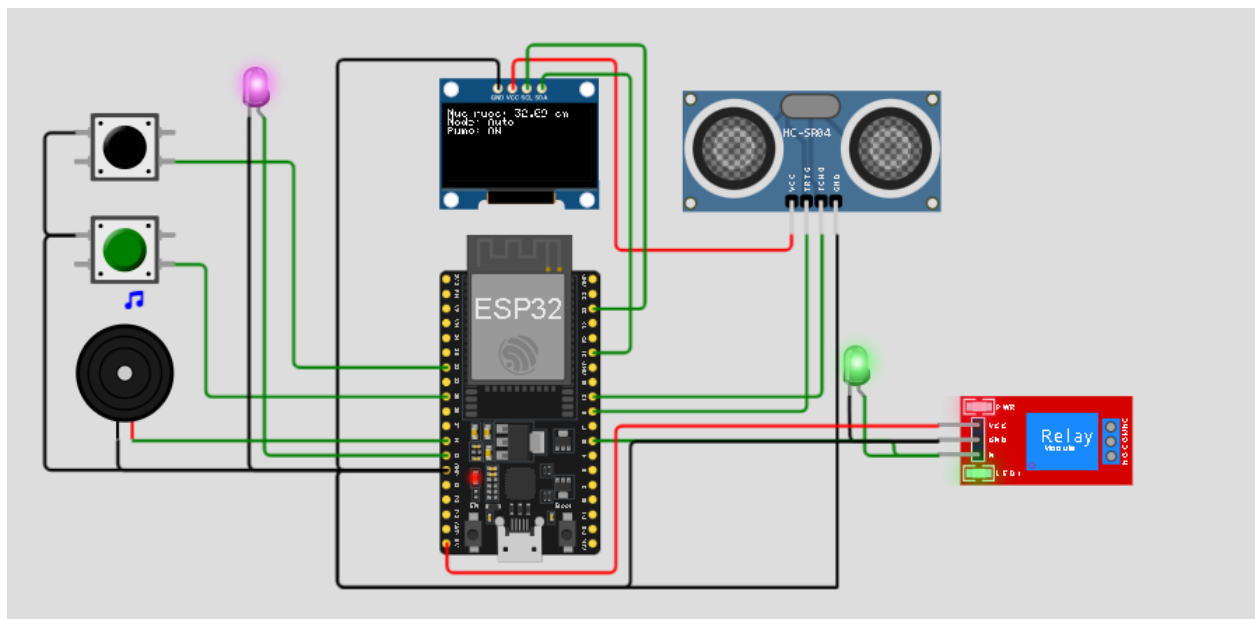
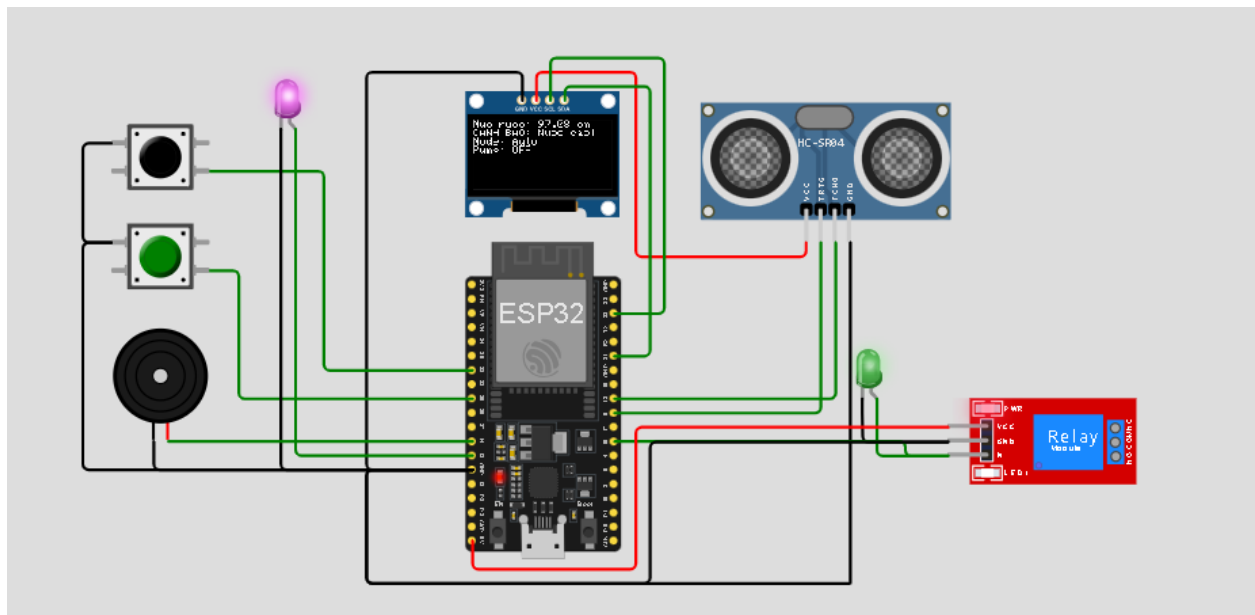
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0,
"attrs": {} },
    {
      "type": "board-ssd1306",
      "id": "oled1",
      "top": -102.46,
      "left": 0.23,
      "attrs": { "i2cAddress": "0x3c" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -94.5,
      "left": 159.1,
      "attrs": { "distance": "3" }
    },
    { "type": "wokwi-relay-module", "id": "relay1", "top": 105.8, "left":
326.4, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led4",
      "top": 63.6,
      "left": 253.4,
      "attrs": { "color": "green" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn4",
      "top": -13,
      "left": -240,
      "attrs": { "color": "green" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn5",
      "top": -80.2,
      "left": -240,
      "attrs": { "color": "black" }
    },
    {
      "type": "wokwi-led",
      "id": "led5",
      "top": -118.8,
      "left": -140.2,
      "attrs": { "color": "purple" }
    }

```

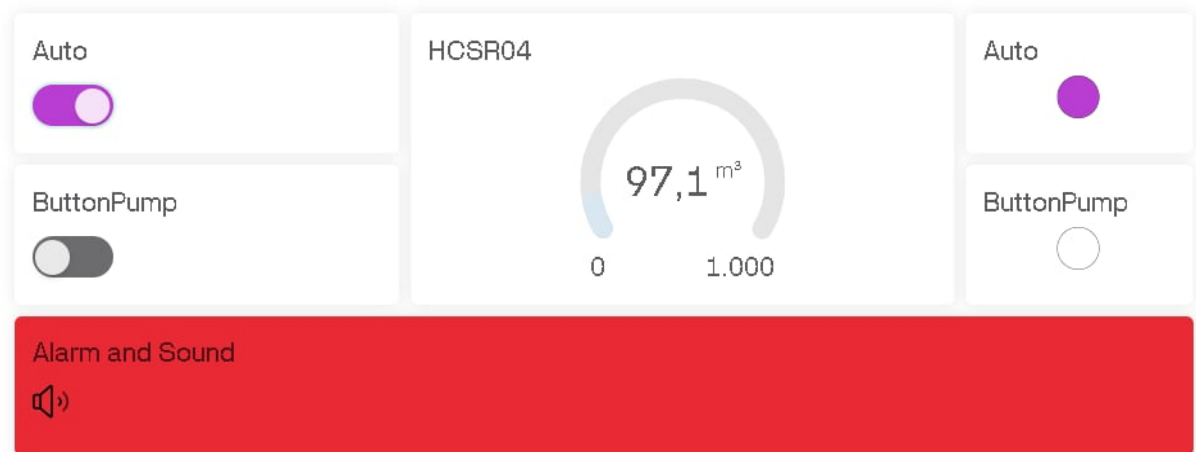
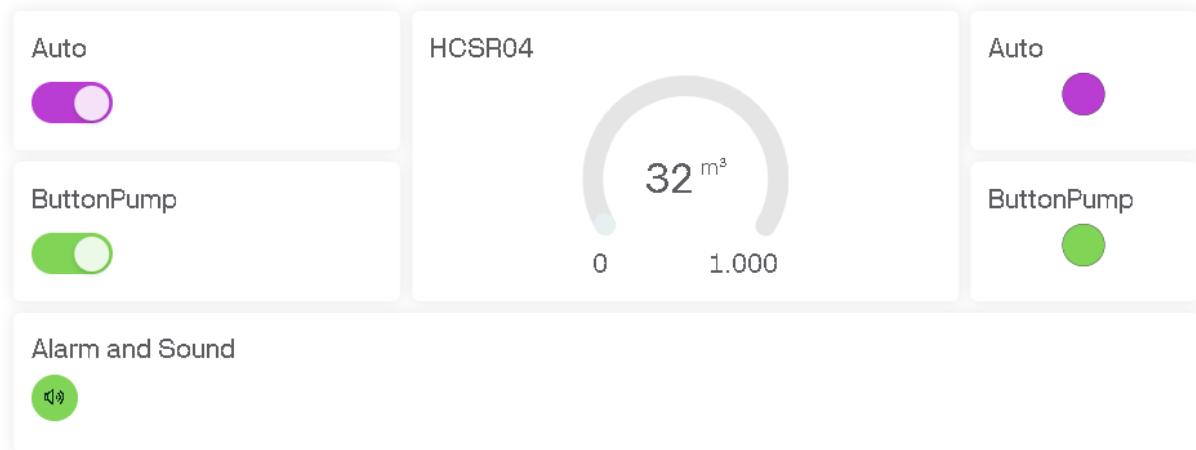
```

    },
    {
      "type": "wokwi-buzzer",
      "id": "bz1",
      "top": 50.4,
      "left": -238.2,
      "attrs": { "volume": "0.1" }
    }
  ],
  "connections": [
    [ "esp:TX", "$serialMonitor:RX", "", [] ],
    [ "esp:RX", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:TRIG", "esp:5", "green", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:18", "green", [ "v0" ] ],
    [ "oled1:VCC", "ultrasonic1:VCC", "red", [ "v-19.2", "h67.35", "v124.8",
    "h96" ] ],
    [ "oled1:SCL", "esp:22", "green", [ "v-28.8", "h77.1", "v172.8" ] ],
    [ "oled1:SDA", "esp:21", "green", [ "v-9.6", "h57.67", "v182.4" ] ],
    [ "oled1:GND", "esp:GND.1", "black", [ "v-19.2", "h-105.6", "v268.8" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v230.4", "h-1.2", "v0", "h-
    326.4", "v-76.8" ] ],
    [ "led4:C", "relay1:GND", "black", [ "v9.6", "h0.4", "v19.6" ] ],
    [ "led4:A", "relay1:IN", "green", [ "v0" ] ],
    [ "btn4:2.r", "esp:25", "green", [ "h19.4", "v86.6" ] ],
    [ "btn5:1.1", "btn4:1.1", "black", [ "h-19.2", "v67.2" ] ],
    [ "btn5:2.r", "esp:32", "green", [ "h77", "v125" ] ],
    [ "led5:C", "esp:GND.1", "black", [ "v230.4", "h10" ] ],
    [ "led5:A", "esp:12", "green", [ "v0" ] ],
    [ "relay1:IN", "esp:16", "green", [ "h-28.8", "v-230.6" ] ],
    [ "btn4:1.1", "esp:GND.1", "black", [ "h-19.2", "v153.6" ] ],
    [ "relay1:VCC", "esp:5V", "red", [ "h-211.2", "v96", "h-115.2" ] ],
    [ "relay1:GND", "esp:GND.1", "black", [ "h-201.6", "v95.6", "h-192", "v-
    76.8" ] ],
    [ "bz1:1", "esp:GND.1", "black", [ "v19.2" ] ],
    [ "bz1:2", "esp:14", "green", [ "v0" ] ]
  ],
  "dependencies": {}
}

```



3.3. Giao diện Blynk



4. Lập trình cho hệ thống

4.1. Chương trình tổng quát

Phần mềm được lập trình bằng Visual Studio Code với ngôn ngữ C/C++ sử dụng nền tảng Arduino. Chương trình có nhiệm vụ:

- Khởi tạo các thiết bị (màn hình OLED, cảm biến siêu âm HC-SR04, buzzer, LED, relay).
- Đo khoảng cách mực nước bằng cảm biến HC-SR04.
- Tính toán mực nước từ khoảng cách đo được.
- Hiển thị mực nước và trạng thái hoạt động lên màn hình OLED.
- So sánh mực nước với ngưỡng cài đặt để tự động điều khiển bơm và cảnh báo.
- Cho phép người dùng chuyển đổi chế độ Tự động/Thủ công bằng nút nhấn hoặc từ xa qua app Blynk.
- Gửi dữ liệu mực nước và trạng thái thiết bị lên ứng dụng Blynk để theo dõi từ xa.

4.1.1. Mục tiêu của chương trình

Xây dựng một hệ thống giám sát và điều khiển mực nước thông minh có khả năng:

- Đo khoảng cách từ mặt nước đến cảm biến và tính toán mực nước thực tế.
- Hiển thị mực nước, chế độ hoạt động, và trạng thái bơm trên màn hình OLED.
- Phát cảnh báo bằng còi buzzer khi mực nước vượt ngưỡng an toàn.
- Tự động điều khiển bơm trong chế độ Auto hoặc cho phép điều khiển bằng tay (Manual).
- Gửi dữ liệu thời gian thực lên app Blynk để người dùng giám sát và điều khiển từ xa.

4.1.2. Cấu trúc chương trình tổng quát

STT	Thành phần	Mô tả
1	Khai báo thư viện	Sử dụng thư viện cho OLED, WiFi, Blynk, cảm biến siêu âm, v.v.
2	Biến cấu hình	Khai báo thông tin WiFi, Blynk token, chân kết nối thiết bị, ngưỡng mực nước, chiều cao bể, trạng thái thiết bị.
3	Hàm setup()	Khởi tạo thiết bị: màn hình OLED, WiFi, Blynk, cấu hình chân vào/ra, thiết lập ngắt ngoài cho nút nhấn.

4	Hàm loop()	Đọc cảm biến, điều khiển bơm, xử lý cảnh báo, hiển thị dữ liệu, gửi dữ liệu lên Blynk.
5	Hàm BLYNK_WRITE()	Nhận lệnh điều khiển từ app Blynk: bật/tắt bơm, chuyển chế độ, bật/tắt còi cảnh báo.

4.1.3. Luồng hoạt động của hệ thống

1. Khởi động

- Kết nối WiFi.
- Kết nối ứng dụng Blynk.
- Khởi tạo màn hình OLED và hiển thị thông báo khởi động.

2. Đo mực nước

- Sử dụng cảm biến siêu âm HC-SR04 đo khoảng cách từ cảm biến đến mặt nước.
- Tính toán mực nước dựa trên chiều cao bể trừ đi khoảng cách đo được.

3. Hiển thị lên OLED

- Hiển thị mực nước hiện tại (đơn vị cm).
- Hiển thị chế độ hoạt động (Auto/Hand).
- Hiển thị trạng thái của bơm (ON/OFF).
- Hiển thị cảnh báo nếu mực nước vượt ngưỡng.

4. Xử lý cảnh báo và điều khiển bơm

- Nếu ở chế độ Auto:
 - Mực nước < ngưỡng: tắt bơm
 - Mực nước > ngưỡng: bật bơm + còi cảnh báo
- Nếu ở chế độ Manual:
 - Người dùng có thể bật/tắt bơm bằng nút nhấn hoặc app.
 - Cảnh báo vẫn có thể được bật/tắt thủ công từ xa.

5. Gửi dữ liệu lên Blynk

- Gửi mực nước thực tế lên pin ảo V1.
- Gửi trạng thái bơm (ON/OFF) lên V0.
- Gửi chế độ hoạt động (Auto/Hand) lên V2.

- Nhận điều khiển còi cảnh báo từ V3.

4.1.4. Mô hình dữ liệu kết nối Blynk

Tên thiết bị	Pin ảo	Kiểu	Vai trò
Mức nước	V0	Double	Gửi dữ liệu mức nước về Blynk
Trạng thái bơm	V1	Integer	Gửi trạng thái ON/OFF của bơm
Chế độ hoạt động	V2	Integer	Nhận điều khiển chế độ từ người dùng
Buzzer	V3	Integer	Nhận điều khiển còi cảnh báo từ app

4.1.5. Ưu điểm của mô hình

- Chế độ linh hoạt: Hệ thống có thể hoạt động tự động hoặc thủ công tùy ý người dùng.
- Giám sát từ xa: Người dùng có thể theo dõi mức nước, điều khiển bơm/cảnh báo qua Internet.
- Cảnh báo kịp thời: Còi và đèn LED giúp cảnh báo tại chỗ khi mức nước vượt ngưỡng.
- Hiển thị trực quan: Màn hình OLED cung cấp thông tin rõ ràng ngay tại thiết bị.
- Tiết kiệm năng lượng: Bơm chỉ hoạt động khi thực sự cần thiết.

4.2. Khai báo thông tin Blynk

```
#define BLYNK_TEMPLATE_ID "TMPL6ZNbU8xp0"
#define BLYNK_TEMPLATE_NAME "ESP32HCSR04"
#define BLYNK_AUTH_TOKEN "2wIx8xYYy0kcz--RtnqNI5TAIT58rPm"
```

Mục đích: Liên kết với nền tảng Blynk IoT.

Chi tiết:

- BLYNK_TEMPLATE_ID: ID mẫu từ Blynk.
- BLYNK_TEMPLATE_NAME: Tên thiết bị hiển thị trên ứng dụng.
- BLYNK_AUTH_TOKEN: Mã xác thực để kết nối thiết bị với tài khoản Blynk.

4.3. Thư viện và cấu hình OLED

```
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
```

Mục đích:

- Kết nối WiFi + Blynk.
- Điều khiển màn hình OLED SSD1306.

Thiết lập độ phân giải màn hình OLED:

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
```

Tạo đối tượng display để thao tác với màn hình:

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

4.4. Khai báo chân phân cứng

```
const int trigPin = 5;
const int echoPin = 18;
const int Pump = 16;
const int LED = 12;
const int Buzzer = 14;
const int buttonPumpPin = 25;
const int buttonModePin = 32;
```

TRIG_PIN Chân phát tín hiệu của cảm biến siêu âm HC-SR04.

ECHO_PIN Chân nhận tín hiệu phản hồi từ cảm biến siêu âm HC-SR04.

PUMP_PIN Điều khiển bơm nước thông qua relay.

LED_PIN Đèn LED hiển thị trạng thái chế độ tự động.

BUZZER_PIN Buzzer dùng để cảnh báo khi mực nước vượt ngưỡng cho phép.

BUTTON_PUMP Nút nhấn điều khiển bật/tắt bơm khi ở chế độ thủ công.

BUTTON_MODE Nút nhấn chuyển đổi giữa chế độ tự động và thủ công.

4.4 Khai báo ngưỡng nước

```
float nguongnuoc = 50;  
float chieucaobe = 100.04;
```

nguongnuoc Ngưỡng mực nước tối đa cho phép (tính bằng cm). Khi vượt ngưỡng này, hệ thống sẽ kích hoạt cảnh báo và bật bơm (ở chế độ tự động).

chieucaobe Chiều cao tối đa của bể chứa (tính bằng cm), dùng để tính toán mực nước hiện tại dựa trên khoảng cách đo được.

4.5. Thông tin Wi-Fi

```
const char* ssid = "Wokwi-GUEST";  
const char* password = "";
```

4.6. Hàm setup()

Khởi tạo Serial để in thông tin lên terminal (debug):

```
void setup() {  
  Serial.begin(115200);
```

Kết nối Blynk + Wi-Fi:

```
connectWiFi();  
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);  
timer.setInterval(1000L, sendSensorData);
```

Cài đặt chế độ hoạt động của các chân: xuất tín hiệu trigger, đọc echo, điều khiển relay, LED cảnh báo, buzzer cảnh báo, nút bấm:

```
pinMode(trigPin, OUTPUT);  
pinMode(echoPin, INPUT);  
pinMode(Pump, OUTPUT);  
pinMode(LED, OUTPUT);  
pinMode(Buzzer, OUTPUT);  
digitalWrite(Buzzer, LOW);  
pinMode(buttonModePin, INPUT_PULLUP);  
pinMode(buttonPumpPin, INPUT_PULLUP);
```

Khởi động màn hình OLED, địa chỉ 0x3C là địa chỉ mặc định I2C, nếu không tìm thấy OLED, dừng chương trình tại while(true):

```
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {  
  Serial.println(F("Không tìm thấy màn hình OLED!"));  
  while (true);  
}
```

Hiển thị thông báo khởi động hệ thống trong 1 giây:

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 0);
display.println("Khoi dong he thong...");
display.display();
delay(1000);
```

Cài đặt ngắt ngoài cho nút bấm:

```
attachInterrupt(digitalPinToInterrupt(buttonPumpPin),    buttonPumpInterrupt,
FALLING);
    attachInterrupt(digitalPinToInterrupt(buttonModePin), buttonModeInterrupt,
FALLING);
```

Thiết lập gửi dữ liệu cảm biến định kỳ về Blynk:

```
timer.setInterval(1000L, sendSensorData);
```

4.7. Hàm loop()

Đọc cảm biến và điều khiển LED:

```
readDistanceAndControlPump(distanceCm);
```

Hiển thị dữ liệu lên màn hình OLED:

```
displayData(distanceCm);
```

4.8. Hiển thị lên màn hình OLED

- Hiển thị các thông tin:
 - Mức nước hiện tại.
 - Cảnh báo nếu mực nước vượt ngưỡng.
 - Trạng thái của hệ thống:
 - Chế độ hoạt động: "Auto" hoặc "Hand"
 - Trạng thái bơm: "ON" hoặc "OFF"

```
• void displayData(float distanceCm) {
•   display.clearDisplay();
•   display.setTextSize(1);
•   display.setTextColor(WHITE);
•   display.setCursor(0, 0);
•   int buzzerState = digitalRead(Buzzer);
•   Blynk.virtualWrite(V3, buzzerState);
•
•   if (distanceCm == -1) {
```

```

•   display.println("Loi cam bien!");
•   } else {
•       float mucNuoc = chieucaobe - distanceCm;
•       if (mucNuoc < 0) mucNuoc = 0;
•
•       display.print("Muc nuoc: ");
•       display.print(mucNuoc);
•       display.println(" cm");
•
•       if (mucNuoc > nguongnuoc) {
•           display.println("CANH BAO: Nuoc cao!");
•           display.setTextColor(WHITE); // Trở lại màu bình thường
•       }
•   }
•
•   display.print("Mode: ");
•   display.println(ModeState ? "Auto" : "Hand");
•   display.print("Pump: ");
•   display.println(PumpState ? "ON" : "OFF");
•
•   display.display();
• }

```

4.9. Điều kiện cảnh báo

Hệ thống sẽ kích hoạt cảnh báo bằng còi Buzzer khi mực nước vượt quá ngưỡng nguy hiểm đã cài đặt trước đó (nguongnuoc). Điều kiện cảnh báo được kiểm tra trong chế độ tự động (Auto).

Cụ thể như sau:

- Khi mực nước > ngưỡng:
 - Buzzer kêu (HIGH)
 - Bơm hoạt động (Pump ON)
- Khi mực nước < 20 cm:
 - Bơm tắt (Pump OFF) để tránh cạn bể
- Khi mực nước an toàn:
 - Tắt buzzer để tránh gây nhiễu

Điều kiện này được kiểm tra định kỳ trong hàm readDistanceAndControlPump() và chỉ được thực hiện khi hệ thống ở chế độ Auto.

```

if (ModeState && distanceCm != -1) {
    float mucNuoc = chieucaobe - distanceCm;

    if (mucNuoc > nguongnuoc) {
        digitalWrite(Pump, HIGH);
        PumpState = true;
        digitalWrite(Buzzer, HIGH);
        Serial.println(" ⚠ CẢNH BÁO: Mức nước vượt ngưỡng!");
    } else if (mucNuoc < 20) {
        digitalWrite(Pump, LOW);
        PumpState = false;
    } else {
        digitalWrite(Buzzer, LOW);
    }
}

```

4.10. Chu kỳ lặp

Cập nhật nội dung lên màn hình OLED và lặp lại mỗi 1 giây:

```

display.display();
delay(1000);

```

4.11. Tổng kết chức năng chính

Chức năng	Mô tả
Đọc giá trị mực nước	Đo khoảng cách bằng cảm biến siêu âm HC-SR04 (dùng chân trig 5, echo 18)
Hiển thị lên OLED	Sử dụng thư viện Adafruit_SSD1306 để hiển thị mực nước, chế độ, bơm
Cảnh báo LED + Buzzer	Khi mực nước vượt ngưỡng 50 cm, kích hoạt LED và còi báo
Gửi dữ liệu đến ứng dụng Blynk	Sử dụng V0 (trạng thái bơm), V1 (mực nước), V2 (chế độ), V3 (buzzer)

KẾT LUẬN

Qua quá trình nghiên cứu và thực hiện, đề tài “Phát triển hệ thống cảnh báo ngập nước với ESP32” đã đạt được các mục tiêu đề ra. Hệ thống có khả năng đo đặc mực nước bằng cảm biến siêu âm HC-SR04, xử lý và phân tích dữ liệu thông qua vi điều khiển ESP32, từ đó đưa ra cảnh báo tức thì bằng còi và đèn LED khi mực nước vượt quá giới hạn. Ngoài ra, việc tích hợp màn hình OLED giúp người dùng dễ dàng quan sát trực tiếp thông tin tại chỗ, trong khi nền tảng Blynk hỗ trợ giám sát và điều khiển thiết bị từ xa một cách linh hoạt.

Một điểm nổi bật của hệ thống là khả năng chuyển đổi giữa hai chế độ hoạt động: thủ công và tự động. Ở chế độ tự động, bơm được kích hoạt khi phát hiện tình trạng mực nước cao, đảm bảo phản ứng kịp thời với các tình huống ngập úng tiềm ẩn. Còn với chế độ thủ công, người dùng có thể trực tiếp điều khiển bơm thông qua nút nhấn hoặc ứng dụng trên điện thoại.

Hệ thống được đánh giá là thân thiện, dễ triển khai với chi phí thấp và hiệu quả cao, phù hợp để áp dụng trong nhiều môi trường như nhà dân, khu vực trũng thấp, hầm xe, hoặc các vùng hay xảy ra mưa lớn. Trong tương lai, hệ thống có thể mở rộng thêm các tính năng như tự động gửi cảnh báo qua mạng di động, đồng bộ dữ liệu lên nền tảng đám mây, hoặc tích hợp cùng các hệ thống thoát nước thông minh.

Tóm lại, đây là một mô hình ứng dụng công nghệ IoT có tính thực tiễn cao, góp phần vào việc chủ động phòng chống ngập úng, giảm thiểu thiệt hại và nâng cao chất lượng cuộc sống cho cộng đồng.

TÀI LIỆU THAM KHẢO

1. Tài liệu học tập, source code tham khảo của thầy Võ Việt Dũng, <https://github.com/vvdung-husc/2024-2025.2.TIN4024.004>
2. Internet of Things – Những nền tảng cơ bản và ứng dụng, Nguyễn Việt Hùng (2019), NXB Bách Khoa Hà Nội.
3. Tìm hiểu cảm biến siêu âm HC-SR04, link từ <https://byvn.net/qO9C>
4. Màn hình OLED là gì? Cấu tạo và đặc điểm nổi bật của màn hình OLED, <https://www.dienmayxanh.com/kinh-nghiem-hay/man-hinh-oled-la-gi-1060968>
5. Tìm hiểu về Buzzer, <https://dientutuonglai.com/tim-hieu-buzzer.html>
6. Hướng dẫn lập trình vi điều khiển ESP32 với Arduino IDE, Võ Quốc Hưng (2021), NXB Đại học Quốc gia TP.HCM.
5. Giáo trình Kỹ thuật cảm biến, Đào Văn Tường (2020), NXB Khoa học và Kỹ thuật.
10. Blynk Documentation – Quickstart, Widgets and Templates, Blynk Inc. (2021), Truy cập từ: <https://docs.blynk.io>