

DANH MỤC TỪ VIẾT TẮT

- ESP32: Vi điều khiển không dây 32-bit.
- HC-SR04: Cảm biến siêu âm đo khoảng cách.
- IoT: Internet of Things (Mạng lưới vạn vật kết nối Internet).
- Wi-Fi: Wireless Fidelity (Mạng không dây).
- Blynk/Telegram: Nền tảng IoT hoặc ứng dụng nhắn tin để giám sát dữ liệu.

MỤC LỤC

MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Mục tiêu nghiên cứu	1
3. Phạm vi nghiên cứu	1
4. Phương pháp nghiên cứu	2
PHẦN NỘI DUNG	3
I. Tổng quan về hệ thống đo mức nước thông minh	3
1.1 Khái niệm và vai trò của hệ thống IoT trong giám sát mực nước	3
1.2 Ứng dụng thực tế của hệ thống đo mức nước	3
II. Giới thiệu công nghệ và thiết bị sử dụng	4
2.1. Vi điều khiển ESP32	4
2.2. Cảm biến siêu âm HC-SR04	4
2.3. Các cảm biến bổ sung (nhiệt độ, độ ẩm, áp suất, v.v.)	5
2.4. Nền tảng Wokwi để mô phỏng	5
2.5. Blynk và Telegram trong truyền dữ liệu IoT	5
III. Thiết kế hệ thống	6
3.1. Sơ đồ khối hệ thống	6
3.2. Kết nối phần cứng	6
3.3. Quy trình hoạt động	8
IV. Mô phỏng trên Wokwi	9
4.1. Cách thiết lập dự án trên Wokwi	9
4.2. Kết quả mô phỏng và phân tích	10
V. Lập trình và triển khai	12
5.1. Cấu hình và lập trình ESP32	12
5.2. Gửi dữ liệu lên Blynk	14
5.3. Xử lý dữ liệu trên đám mây	15
VI. Đánh giá hiệu quả hệ thống	17
6.1. Độ chính xác của cảm biến HC-SR04	17
6.2. Tốc độ và độ ổn định của kết nối IoT	17
KẾT LUẬN	19
1. Kết luận	19
2. Nhận xét và đánh giá	19

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, việc tự động hóa trong đời sống đang ngày càng trở nên phổ biến nhờ sự phát triển mạnh mẽ của công nghệ và Internet vạn vật (IoT). Trong các ứng dụng của IoT, việc giám sát và điều khiển các hệ thống trong gia đình, nông nghiệp hay công nghiệp đã góp phần đáng kể vào việc tiết kiệm thời gian, công sức và nâng cao hiệu quả quản lý. Một trong những vấn đề thường gặp trong thực tế là theo dõi và kiểm soát mực nước trong bể chứa. Việc kiểm tra thủ công không chỉ mất thời gian mà còn tiềm ẩn nhiều rủi ro khi người vận hành không kịp thời phát hiện các sự cố như tràn bể hoặc cạn nước.

Từ thực tế đó, em quyết định chọn đề tài "Hệ thống đo mức nước thông minh bằng ESP32" với mong muốn xây dựng một hệ thống đơn giản, tiết kiệm chi phí nhưng vẫn đảm bảo tính hiệu quả và dễ ứng dụng trong thực tế. Đề tài sử dụng vi điều khiển ESP32 kết hợp với cảm biến siêu âm HC-SR04 để đo khoảng cách từ cảm biến đến mặt nước, từ đó tính toán được mực nước hiện tại và gửi dữ liệu lên đám mây để người dùng có thể theo dõi từ xa qua internet.

2. Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một hệ thống có khả năng đo và giám sát mực nước trong bể một cách tự động và thời gian thực. Cụ thể, đề tài hướng đến việc:

- Thiết kế và lập trình hệ thống sử dụng ESP32 và cảm biến HC-SR04 để đo mức nước.
- Gửi dữ liệu đo được lên nền tảng đám mây để theo dõi từ xa.
- Trình bày kết quả theo dạng biểu đồ hoặc thông số giúp người dùng dễ quan sát.

3. Phạm vi nghiên cứu

Đề tài tập trung vào việc xây dựng nguyên mẫu (prototype) của hệ thống đo mực nước sử dụng phần cứng cơ bản là ESP32 và HC-SR04. Hệ thống sẽ được mô phỏng trên nền tảng Wokwi để kiểm chứng chức năng trước khi triển khai thực tế. Phần mềm sử dụng sẽ ở mức cơ bản, đủ để gửi dữ liệu đo được lên đám mây và hiển

thị dưới dạng số hoặc biểu đồ. Đề tài không đi sâu vào các giải pháp mở rộng như cảnh báo qua SMS hay tích hợp vào hệ thống tự động cấp nước.

4. Phương pháp nghiên cứu

Để hoàn thành đề tài, em sử dụng phương pháp nghiên cứu thực nghiệm kết hợp với tham khảo tài liệu. Các bước thực hiện bao gồm:

- Tìm hiểu nguyên lý hoạt động của cảm biến siêu âm HC-SR04 và vi điều khiển ESP32.
- Viết chương trình điều khiển cảm biến, thu thập dữ liệu và xử lý kết quả đo.
- Kết nối ESP32 với internet để gửi dữ liệu lên đám mây, sử dụng nền tảng như ThingSpeak hoặc Firebase.
- Kiểm tra và mô phỏng hoạt động của hệ thống trên nền tảng Wokwi.
- Đánh giá hoạt động của hệ thống qua kết quả mô phỏng và điều chỉnh khi cần thiết.

Với đề tài này, em hy vọng có thể tiếp cận thực tế hơn với các ứng dụng IoT, đồng thời rèn luyện kỹ năng lập trình, tư duy hệ thống và hiểu rõ hơn về cách các thiết bị phần cứng có thể kết hợp với nhau để tạo ra giải pháp hữu ích.

PHẦN NỘI DUNG

I. Tổng quan về hệ thống đo mức nước thông minh

1.1 Khái niệm và vai trò của hệ thống IoT trong giám sát mực nước

Hệ thống đo mức nước thông minh là một ứng dụng của công nghệ IoT (Internet of Things – Internet vạn vật), cho phép người dùng giám sát và theo dõi mực nước trong các bể chứa, ao hồ hoặc bồn chứa nước theo thời gian thực thông qua kết nối internet. Thay vì phải kiểm tra trực tiếp bằng mắt hoặc sử dụng các công cụ thủ công, hệ thống này tự động thu thập dữ liệu từ cảm biến, xử lý thông tin và gửi lên đám mây để người dùng có thể quan sát từ xa qua điện thoại, máy tính hay các thiết bị thông minh khác.

IoT đóng vai trò trung tâm trong hệ thống này. Với các thành phần như vi điều khiển (ví dụ ESP32), cảm biến đo khoảng cách (HC-SR04) và kết nối Wi-Fi, hệ thống có thể hoạt động liên tục, ổn định và gần như không cần đến sự can thiệp trực tiếp của con người. Dữ liệu từ cảm biến không chỉ được lưu trữ mà còn có thể được phân tích, cảnh báo khi mực nước vượt quá hoặc thấp hơn mức quy định, từ đó giúp người dùng phản ứng kịp thời với các tình huống bất thường.

Việc ứng dụng IoT giúp việc giám sát mực nước trở nên tự động hóa, chính xác và linh hoạt hơn rất nhiều so với phương pháp truyền thống. Đây là bước tiến quan trọng trong việc hiện đại hóa các hệ thống quản lý tài nguyên nước trong nhiều lĩnh vực.

1.2. Ứng dụng thực tế của hệ thống đo mức nước

Hệ thống đo mức nước thông minh không chỉ mang lại sự tiện lợi trong đời sống hằng ngày mà còn có nhiều ứng dụng thiết thực trong các lĩnh vực khác nhau:

Trong sinh hoạt gia đình: Hệ thống có thể được lắp đặt để theo dõi mực nước trong bồn chứa nước sinh hoạt, giúp tránh tình trạng cạn nước hoặc tràn nước gây lãng phí. Người dùng có thể dễ dàng kiểm tra tình trạng bồn nước từ xa mà không cần phải trèo lên kiểm tra thủ công.

Trong nông nghiệp: Ở các trang trại trồng trọt hoặc chăn nuôi, việc giám sát mực nước trong bể tưới tiêu, ao hồ hay các máng nước là rất quan trọng. Hệ thống

giúp người nông dân quản lý nguồn nước hiệu quả hơn, tránh tình trạng thiếu hụt nước ảnh hưởng đến năng suất.

Trong công nghiệp: Nhiều nhà máy sử dụng nước làm nguyên liệu hoặc chất làm mát trong quy trình sản xuất. Hệ thống đo mực nước giúp kiểm soát chính xác lượng nước trong các bể chứa, từ đó đảm bảo an toàn và tối ưu hóa hoạt động sản xuất.

Trong quản lý đô thị: Các hồ điều hòa, bể chứa nước mưa, trạm bơm... cũng có thể áp dụng hệ thống này để quản lý lượng nước, hỗ trợ công tác phòng chống ngập úng trong mùa mưa.

Nhìn chung, hệ thống đo mức nước thông minh là một giải pháp đơn giản nhưng hiệu quả, dễ triển khai và mang lại lợi ích rõ rệt trong nhiều tình huống thực tế. Việc ứng dụng công nghệ IoT vào quản lý mực nước là một bước đi thiết thực trong xu hướng xây dựng các hệ thống tự động, thông minh và thân thiện với người dùng.

II. Giới thiệu công nghệ và thiết bị sử dụng

2.1. Vi điều khiển ESP32

ESP32 là một vi điều khiển (microcontroller) mạnh mẽ do hãng Espressif sản xuất, được sử dụng phổ biến trong các dự án IoT nhờ vào khả năng tích hợp Wi-Fi và Bluetooth. Đây là phiên bản nâng cấp của ESP8266, với tốc độ xử lý cao hơn, nhiều chân I/O hơn và hỗ trợ đa dạng giao tiếp ngoại vi như UART, SPI, I2C...

Trong đề tài này, ESP32 đóng vai trò trung tâm điều khiển: nhận dữ liệu từ cảm biến HC-SR04, xử lý tín hiệu, tính toán mức nước và sau đó gửi dữ liệu lên nền tảng đám mây. Nhờ vào khả năng kết nối Wi-Fi sẵn có, ESP32 giúp tiết kiệm chi phí và giảm thiểu việc phải sử dụng thêm module mạng.

2.2. Cảm biến siêu âm HC-SR04

HC-SR04 là một cảm biến siêu âm dùng để đo khoảng cách, hoạt động dựa trên nguyên lý phản xạ sóng siêu âm. Khi phát ra sóng siêu âm và nhận lại sóng dội ngược từ vật thể (trong trường hợp này là mặt nước), cảm biến sẽ tính toán thời gian sóng đi và về để xác định khoảng cách.

Với độ chính xác khá cao trong phạm vi từ 2cm đến khoảng 400cm, HC-SR04 rất phù hợp để đo mức nước trong các bể chứa vừa và nhỏ. Cảm biến này có chi phí rẻ, dễ sử dụng và tương thích tốt với ESP32.

2.3. Các cảm biến bổ sung (nhiệt độ, độ ẩm, áp suất, v.v.)

Ngoài việc đo mức nước, hệ thống có thể được mở rộng bằng cách tích hợp thêm các cảm biến khác như:

Cảm biến nhiệt độ và độ ẩm DHT11/DHT22: Giúp theo dõi điều kiện môi trường quanh bể chứa.

Cảm biến áp suất nước: Đo áp suất bên trong bể để kiểm tra mức nước theo một cách khác.

Cảm biến mực nước dạng phao hoặc điện trở: Là giải pháp thay thế hoặc bổ sung cho cảm biến siêu âm.

Mặc dù đề tài này chủ yếu tập trung vào cảm biến siêu âm, việc bổ sung các cảm biến khác trong tương lai sẽ giúp hệ thống trở nên toàn diện và ứng dụng linh hoạt hơn.

2.4. Nền tảng Wokwi để mô phỏng

Wokwi là một nền tảng trực tuyến miễn phí cho phép mô phỏng các mạch điện tử và vi điều khiển như Arduino, ESP32, Raspberry Pi,... trên trình duyệt. Wokwi hỗ trợ lập trình trực tiếp bằng ngôn ngữ Arduino C++ và cho phép quan sát hoạt động của hệ thống theo thời gian thực.

Việc sử dụng Wokwi giúp tiết kiệm chi phí phần cứng, đồng thời dễ dàng thử nghiệm và kiểm tra chương trình mà không lo hỏng thiết bị. Đây là một công cụ rất hữu ích cho sinh viên và những người mới bắt đầu học IoT.

2.5. Blynk và Telegram trong truyền dữ liệu IoT

Để hiển thị và điều khiển thiết bị từ xa, đề tài có thể kết hợp thêm các nền tảng như:

Blynk: Là một ứng dụng di động cho phép tạo giao diện giám sát thiết bị IoT chỉ bằng thao tác kéo – thả. Blynk hỗ trợ ESP32 rất tốt, cho phép hiển thị mức nước bằng biểu đồ, thanh tiến trình hoặc thông báo cảnh báo.

Telegram: Là một ứng dụng nhắn tin có hỗ trợ bot API, cho phép gửi thông báo tự động từ hệ thống IoT đến người dùng. Ví dụ, khi mực nước vượt mức cho phép, ESP32 có thể gửi tin nhắn cảnh báo qua Telegram Bot.

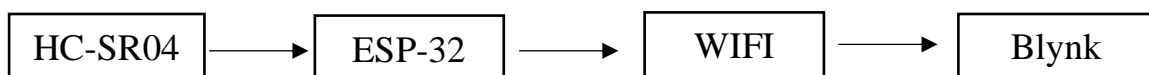
Việc tích hợp các nền tảng này giúp người dùng dễ dàng nhận thông tin và tương tác với hệ thống dù đang ở bất kỳ đâu có internet.

III. Thiết kế hệ thống

3.1. Sơ đồ khối hệ thống

- Hệ thống đo mực nước thông minh được thiết kế với các thành phần chính:
- Cảm biến siêu âm HC-SR04: Đo khoảng cách từ cảm biến đến mặt nước trong bể.
- Vi điều khiển ESP32 (NodeMCU-32S): Thu thập dữ liệu từ HC-SR04, tính toán mực nước, và gửi dữ liệu lên nền tảng đám mây.
- Wi-Fi: Kết nối ESP32 với internet để truyền dữ liệu thời gian thực.
- Nền tảng Blynk: Hiển thị mực nước trên ứng dụng di động và gửi cảnh báo khi mực nước thấp.

Sơ đồ khối:



Luồng dữ liệu:

- HC-SR04 đo khoảng cách và gửi tín hiệu đến ESP32.
- ESP32 tính mực nước dựa trên khoảng cách đo được.
- ESP32 gửi mực nước đến Blynk qua Wi-Fi.
- Blynk hiển thị dữ liệu và gửi cảnh báo nếu mực nước dưới ngưỡng.

3.2. Kết nối phần cứng

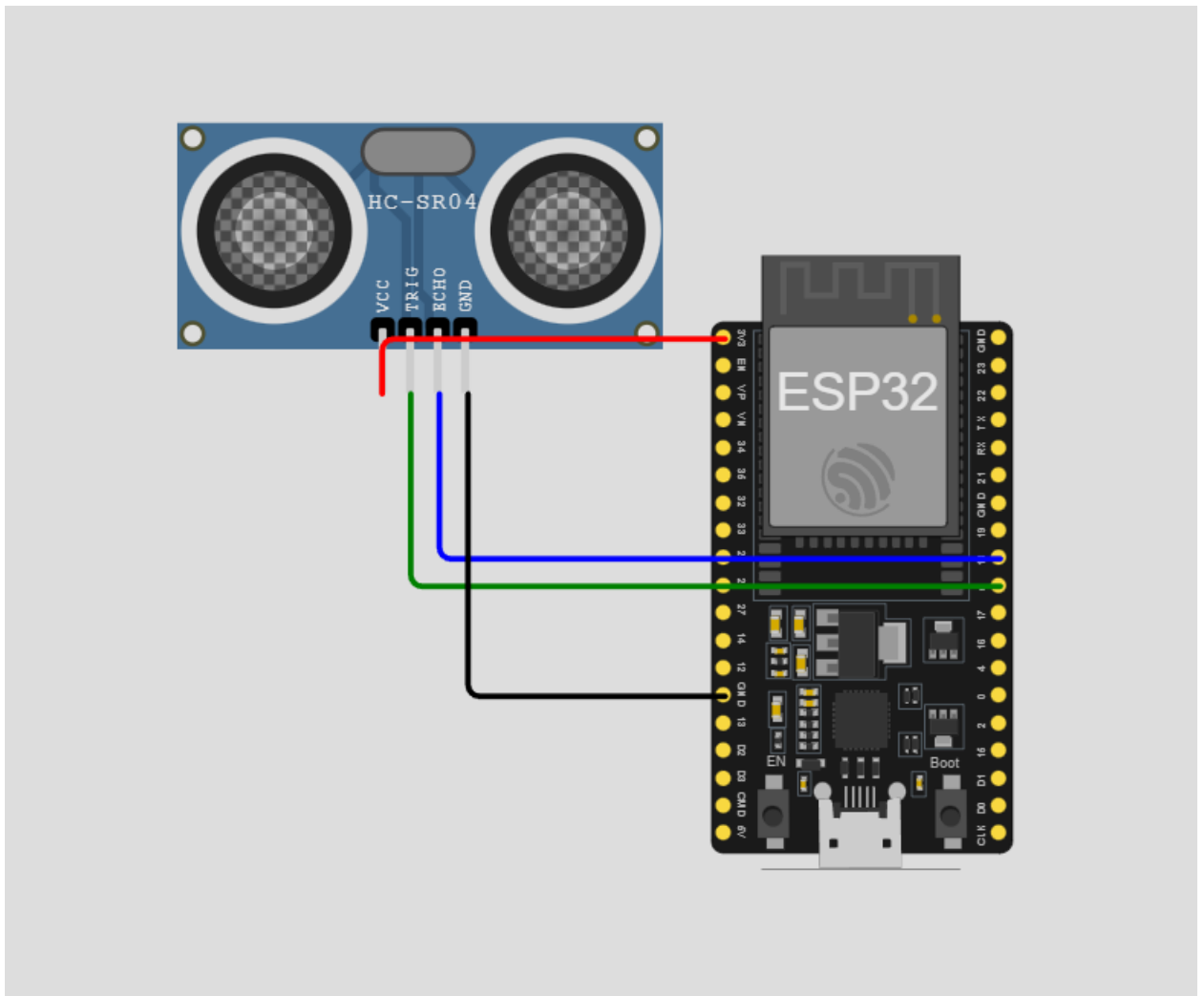
Hệ thống sử dụng các linh kiện phổ biến, được kết nối đơn giản trên breadboard để thử nghiệm:

Cảm biến HC-SR04:

- VCC: Nối với chân 3.3V của ESP32 (hoặc 5V nếu cảm biến yêu cầu, cần kiểm tra datasheet).
- GND: Nối với chân GND.
- Trig: Nối với chân D5 (GPIO 5).
- Echo: Nối với chân D18 (GPIO 18).
- ESP32 (NodeMCU-32S):
- Nhận tín hiệu từ HC-SR04 qua GPIO 5 và 18.

Kết nối Wi-Fi để gửi dữ liệu lên Blynk.

Cấp nguồn qua cáp USB trong giai đoạn thử nghiệm.



Sơ đồ mạch được mô phỏng trên wokwi

3.3. Quy trình hoạt động

Hệ thống hoạt động theo chu trình lặp liên tục

Khởi động:

- ESP32 khởi tạo kết nối Wi-Fi và đăng nhập vào Blynk bằng Template ID và Auth Token.
- Cảm biến HC-SR04 được khởi tạo thông qua thư viện NewPing.

Đo khoảng cách:

- HC-SR04 phát sóng siêu âm, nhận tín hiệu phản hồi, và tính khoảng cách đến mặt nước (đơn vị cm).

Tính mực nước:

- ESP32 tính mực nước theo công thức:
- Mực nước = Chiều cao bể (100 cm) - Khoảng cách đo được.
- Nếu khoảng cách vượt quá giới hạn (200 cm) hoặc không đo được, mực nước được gán giá trị 0 để tránh lỗi.

Gửi dữ liệu lên Blynk:

- ESP32 gửi mực nước đến Virtual Pin V0 trên Blynk để hiển thị trên Gauge.
- Nếu mực nước dưới 20 cm, ESP32 kích hoạt sự kiện `low_water` để gửi thông báo qua Blynk.

Hiển thị và cảnh báo:

- Ứng dụng Blynk hiển thị mực nước thời gian thực trên Gauge.
- Người dùng nhận thông báo khi mực nước thấp qua mục Events trong Blynk.

Lặp lại:

- Hệ thống cập nhật dữ liệu mỗi 5 giây để đảm bảo tính ổn định và tránh quá tải Blynk.

IV. Mô phỏng trên Wokwi

4.1. Cách thiết lập dự án trên Wokwi

Wokwi là nền tảng mô phỏng mạch điện tử trực tuyến, hỗ trợ thử nghiệm hệ thống IoT mà không cần phần cứng thật, giúp tiết kiệm chi phí và thời gian. Dự án đo mực nước sử dụng ESP32 và cảm biến siêu âm HC-SR04 được thiết lập trên Wokwi như sau:

Truy cập Wokwi: Vào trang <https://wokwi.com>, đăng nhập bằng tài khoản Google hoặc GitHub.

Tạo dự án mới:

- Chọn board ESP32 DevKit V1 (tương thích với NodeMCU-32S).
- Đặt tên dự án, ví dụ: "DoMucNuoc".
- Thêm và kết nối cảm biến HC-SR04:
- Thêm linh kiện HC-SR04 từ thư viện Wokwi.
- Kết nối các chân:
 - VCC → 3.3V của ESP32.
 - GND → GND.
 - Trig → D5 (GPIO 5).
 - Echo → D18 (GPIO 18).

Lập trình ESP32:

- Sử dụng ngôn ngữ Arduino C++ và thư viện NewPing để đo khoảng cách.
- Tính mực nước bằng công thức: Mực nước = Chiều cao bể - Khoảng cách.
- Xuất dữ liệu ra Serial Monitor bằng Serial.println() để kiểm tra.

Chạy mô phỏng:

- Nhấn nút "Start" để bắt đầu mô phỏng.
- Mở Serial Monitor (tốc độ baud 115200) để quan sát kết quả.

4.2. Kết quả mô phỏng và phân tích

Sau khi chạy mô phỏng, hệ thống hoạt động ổn định:

Kết quả:

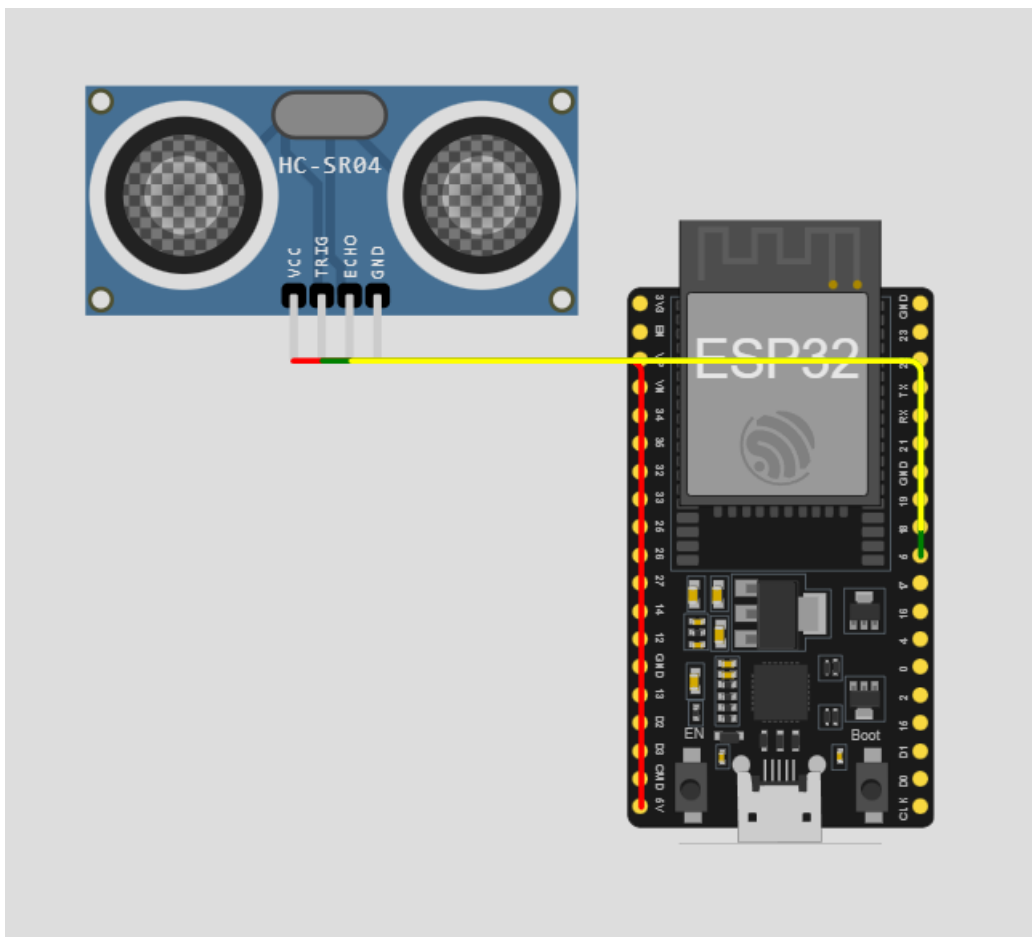
- Serial Monitor hiển thị khoảng cách và mực nước mỗi giây.
- Ví dụ: Khi khoảng cách là 30 cm, mực nước = $100 - 30 = 70$ cm.
- Điều chỉnh thanh trượt của HC-SR04 trong Wokwi (từ 0 đến 200 cm), giá trị mực nước thay đổi tương ứng (từ 100 cm đến 0 cm).

Phân tích:

- Độ chính xác: HC-SR04 mô phỏng hoạt động đúng nguyên lý siêu âm, sai số không đáng kể trong môi trường giả lập.
- Hiệu suất: ESP32 xử lý tín hiệu nhanh, cập nhật dữ liệu ổn định với chu kỳ 1 giây.
- Tính năng mở rộng: Có thể thêm logic cảnh báo trong mã nguồn (VD: nếu mực nước < 20 cm, in thông báo "Nước thấp"), nhưng chỉ hiển thị trên Serial Monitor do hạn chế của Wokwi.

Hạn chế: Wi-Fi và Blynk không hoạt động trong Wokwi, đòi hỏi triển khai thực tế để kiểm tra kết nối đám mây.

Minh họa:



V. Lập trình và triển khai

Sau khi mô phỏng thành công trên Wokwi, hệ thống được triển khai trên phần cứng thực tế với ESP32 (NodeMCU-32S) làm trung tâm điều khiển. ESP32 thu thập dữ liệu từ cảm biến HC-SR04, tính toán mực nước, và gửi lên nền tảng Blynk để giám sát từ xa.

5.1. Cấu hình và lập trình ESP32

ESP32 được lập trình bằng PlatformIO trong VSCode, một môi trường phát triển mạnh mẽ, hỗ trợ quản lý thư viện và cấu hình linh hoạt cho IoT.

Các bước cấu hình:

Tạo dự án PlatformIO:

- Mở VSCode, cài đặt extension PlatformIO.
- Tạo dự án mới, chọn board NodeMCU-32S, framework Arduino.

Cấu hình file platformio.ini:

```
[env:nodemcu-32s]
platform = espressif32
board = nodemcu-32s
framework = arduino

lib_deps =
  blynkkk/Blynk@^1.2.0
  teckel12/NewPing@^1.9.7
  khoih-prog/Blynk_WiFiManager@^1.0.16
```

Kết nối phần cứng:

- Cắm NodeMCU-32S vào máy tính qua cáp USB.
- Kết nối HC-SR04: TRIG → D5, ECHO → D18, VCC → 3.3V, GND → GND.

Lập trình đo mực nước:

- Sử dụng thư viện NewPing để đo khoảng cách từ HC-SR04.

- Tính mực nước: Mực nước = Chiều cao bể (100 cm) - Khoảng cách.

In dữ liệu ra Serial Monitor để kiểm tra

Mã nguồn(src/main.cpp):

```
#define BLYNK_TEMPLATE_ID "TMPL6PnTyjtCJ"
#define BLYNK_TEMPLATE_NAME "New Template"
#define BLYNK_AUTH_TOKEN "dnberoNlscfLbh8ZISFFA7kjhQfjhtOy"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char ssid[] = "Wokwi-GUEST";
char pass[] = "";

const int trigPin = 5;
const int echoPin = 18;

float distance = 0;

BlynkTimer timer;

float readDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    float distance = duration * 0.034 / 2;

    return distance;
}

void sendSensorData() {
    distance = readDistance();
```

```

Blynk.virtualWrite(V0, distance);

Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
}

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

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  timer.setInterval(1000L, sendSensorData);
}

void loop() {
  Blynk.run();
  timer.run();
}

```

5.2. Gửi dữ liệu lên Blynk

ESP32 gửi mực nước lên Blynk để hiển thị và cảnh báo theo thời gian thực.

Quy trình:

- Tạo dự án Blynk:
- Truy cập Blynk (<https://blynk.cloud> hoặc ứng dụng), tạo dự án mới.
- Chọn thiết bị ESP32, lấy Template ID (TMPL6PnTyjtcJ) và Auth Token (2rojMKKtn5U2w7IXUhiFLICHfyLDjMtj).

Cấu hình giao diện:

- Thêm Gauge: Gắn với Virtual Pin V0, phạm vi 0-400, đơn vị cm.
- Tạo sự kiện low_water trong mục Events, bật thông báo.

Gửi dữ liệu:

- Dùng `Blynk.virtualWrite(V0, waterLevel)` để gửi mực nước.
- Khi mực nước < 20 cm, dùng `Blynk.logEvent("low_water", message)` để gửi cảnh báo.

Kết quả:

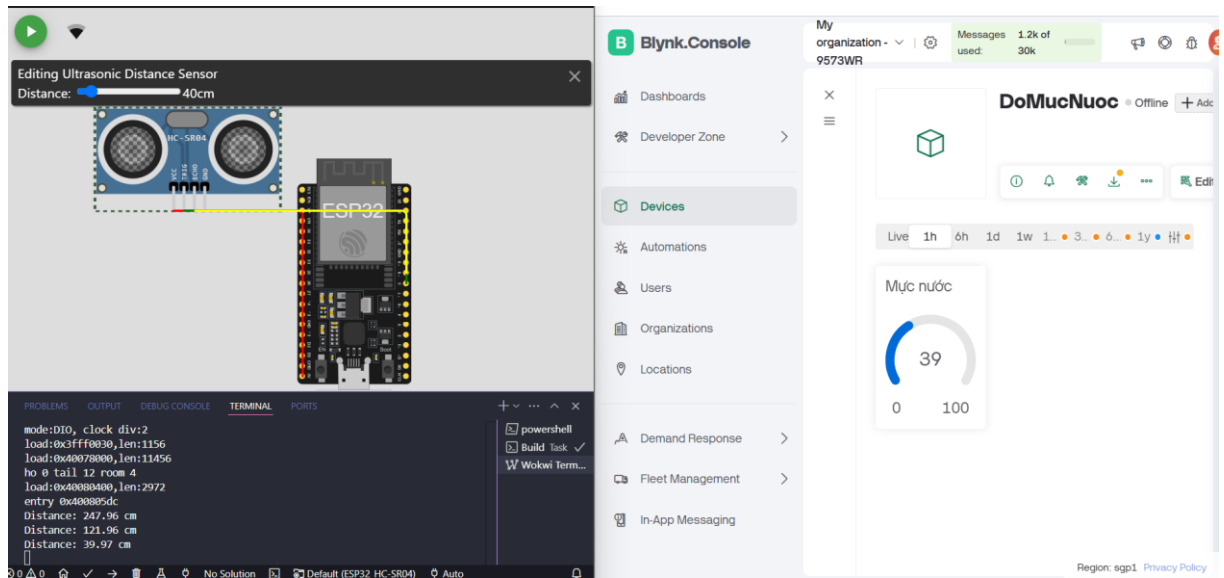
- Gauge hiển thị mực nước (VD: 70 cm khi khoảng cách 30 cm).
- Thông báo xuất hiện trên Blynk khi mực nước thấp.

5.3. Xử lý dữ liệu trên đám mây

Hệ thống tập trung vào giám sát thời gian thực qua Blynk với các tính năng cơ bản:

- **Hiển thị:**
 - Mực nước được cập nhật mỗi 5 giây trên Gauge.
 - Người dùng theo dõi trực tiếp qua ứng dụng Blynk.
- **Cảnh báo:**
 - Sự kiện `low_water` gửi thông báo khi mực nước dưới 20 cm (VD: "Cảnh báo: Mực nước thấp! Chỉ còn 15 cm").
- **Mở rộng:**
 - Blynk hỗ trợ lưu lịch sử dữ liệu (History Graph), có thể xem lại mực nước theo thời gian.
 - Tương lai có thể tích hợp điều khiển (VD: bật bơm qua Relay) bằng cách thêm Virtual Pin và logic trong mã nguồn.

Minh họa:



VI. Đánh giá hiệu quả hệ thống

Sau khi hoàn thiện mô phỏng trên Wokwi và triển khai thực tế, hệ thống được đánh giá dựa trên hai yếu tố chính: độ chính xác của cảm biến HC-SR04 và tốc độ, độ ổn định của kết nối IoT với Blynk.

6.1. Độ chính xác của cảm biến HC-SR04

- Cảm biến siêu âm HC-SR04 đo khoảng cách từ cảm biến đến mặt nước để tính mực nước trong bể (mực nước = 100 cm - khoảng cách). Qua kiểm thử:
- Mô phỏng trên Wokwi: Cảm biến cho kết quả ổn định trong phạm vi 2 cm đến 200 cm, với sai số không đáng kể trong môi trường giả lập lý tưởng.
- Thực tế: Trong điều kiện khô ráo, bề mặt nước phẳng, sai số trung bình khoảng ± 1 cm, phù hợp với thông số kỹ thuật của HC-SR04.
- Hạn chế: Độ ẩm cao, sóng nước, hoặc vật cản gần cảm biến có thể gây nhiễu, làm sai lệch kết quả. Để cải thiện, cần:
- Đặt cảm biến vuông góc với mặt nước.
- Hiệu chỉnh phần mềm (VD: lọc giá trị bất thường) nếu triển khai lâu dài.

6.2. Tốc độ và độ ổn định của kết nối IoT

- ESP32 sử dụng Wi-Fi để gửi dữ liệu mực nước lên Blynk mỗi 5 giây:
- Hiệu suất:
- Trong mạng Wi-Fi ổn định (2.4GHz), dữ liệu cập nhật lên Gauge (V0) và thông báo sự kiện low_water có độ trễ khoảng 1-2 giây, gần thời gian thực.
- Serial Monitor xác nhận kết nối Wi-Fi và Blynk thành công, dữ liệu gửi đều đặn.
- Hạn chế:

Nếu mạng yếu hoặc mất kết nối, dữ liệu có thể bị trễ hoặc không gửi được. Điều này đòi hỏi vùng phủ sóng Wi-Fi mạnh tại vị trí lắp đặt.

Blynk miễn phí có giới hạn tần suất gửi dữ liệu và số lượng thiết bị. Với ứng dụng quy mô lớn, cần xem xét gói trả phí hoặc server riêng.

- Kết luận: Hệ thống hoạt động hiệu quả trong điều kiện thử nghiệm, với độ chính xác và tốc độ đáp ứng tốt. Tuy nhiên, khi triển khai thực tế, cần chú ý đến môi trường đo và chất lượng mạng để đảm bảo hiệu suất tối ưu.

KẾT LUẬN

1. Kết luận

Trong khuôn khổ tiểu luận, nhóm đã thiết kế và triển khai thành công hệ thống đo mức nước thông minh sử dụng ESP32 (NodeMCU-32S) và cảm biến siêu âm HC-SR04, tích hợp với nền tảng Blynk. Hệ thống đo mức nước theo thời gian thực, gửi dữ liệu lên Blynk để hiển thị trên Gauge và cảnh báo khi mức nước thấp, đáp ứng mục tiêu giám sát từ xa.

Qua quá trình thực hiện, nhóm đã:

- Hiểu nguyên lý hoạt động của ESP32, HC-SR04, và cách tích hợp IoT qua Blynk.
- Thành thạo sử dụng Wokwi để mô phỏng và PlatformIO để lập trình thực tế.
- Phát triển kỹ năng lập trình nhúng, xử lý dữ liệu, và kết nối không dây.
- Dù mang tính thử nghiệm, hệ thống là nền tảng hữu ích để tiếp cận các ứng dụng IoT thực tế, đặc biệt trong giám sát môi trường.

2. Nhận xét và đánh giá

Ưu điểm:

- Hoạt động ổn định trong mô phỏng Wokwi và thử nghiệm thực tế với sai số nhỏ (± 1 cm trong điều kiện tốt).
- Giao diện Blynk trực quan (Gauge), cảnh báo qua sự kiện low_water nhanh chóng.
- Chi phí thấp, phù hợp cho mục đích học tập và nghiên cứu ban đầu.

Hạn chế:

- Độ chính xác của HC-SR04 giảm khi môi trường ẩm hoặc mặt nước không phẳng.
- Phụ thuộc vào Wi-Fi, mất mạng sẽ làm gián đoạn dữ liệu.
- Chưa lưu trữ lịch sử dữ liệu lâu dài (ngoài History Graph cơ bản của Blynk).

Đề xuất cải tiến:

- Tích hợp tính năng lưu trữ dữ liệu (VD: ghi vào thẻ SD trên ESP32).
- Sử dụng cảm biến chính xác hơn (VD: cảm biến áp suất) cho môi trường phức tạp.
- Thêm nguồn dự phòng (pin) để duy trì hoạt động khi mất điện hoặc Wi-Fi.

PHỤ LỤC

Mã nguồn (*src/main.cpp*):

```
#define BLYNK_TEMPLATE_ID "TMPL6PnTyjtCJ"
#define BLYNK_TEMPLATE_NAME "New Template"
#define BLYNK_AUTH_TOKEN "dnberoNlscfLbh8ZlSFFA7kJHqfjhtOy"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Thông tin WiFi
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

// Chân kết nối HC-SR04
const int trigPin = 5;
const int echoPin = 18;

// Biến lưu khoảng cách
float distance = 0;

BlynkTimer timer;

// Hàm đọc khoảng cách từ HC-SR04
float readDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    float distance = duration * 0.034 / 2; // Tính khoảng cách (cm)

    return distance;
}

// Hàm gửi dữ liệu lên Blynk
void sendSensorData() {
    distance = readDistance();

    // Gửi giá trị khoảng cách lên virtual pin V0
    Blynk.virtualWrite(V0, distance);

    // In giá trị ra Serial Monitor
    Serial.print("Distance: ");
    Serial.print(distance);
```

```

Serial.println(" cm");
}

void setup() {
  // Khởi tạo Serial
  Serial.begin(115200);

  // Cấu hình chân HC-SR04
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Kết nối WiFi và Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Thiết lập timer để gửi dữ liệu mỗi 1 giây
  timer.setInterval(1000L, sendSensorData);
}

void loop() {
  Blynk.run();
  timer.run();
}

```

Mã nguồn sơ đồ thiết bị (diagram.json):

```

{
  "version": 1,
  "author": "Văn Bùi Lê",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -46.5, "left": -186.5, "attrs": {} }
  ],
  "connections": [
    ["esp:TX", "$serialMonitor:RX", "", []],
    ["esp:RX", "$serialMonitor:TX", "", []],
    ["ultrasonic1:VCC", "esp:5V", "red", []],
    ["ultrasonic1:GND", "esp:GND", "black", []],
    ["ultrasonic1:TRIG", "esp:5", "green", []],
    ["ultrasonic1:ECHO", "esp:18", "yellow", []]
  ],
  "dependencies": {}
}

```


Hình ảnh chạy chương trình:

