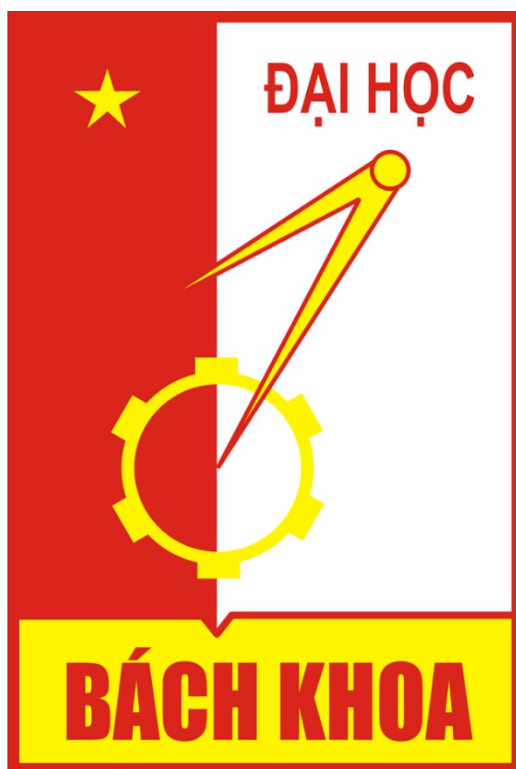


ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
-----o0o-----



BÁO CÁO BÀI TẬP LỚN
ĐỀ TÀI: HỆ THỐNG ĐO MỰC NƯỚC ĐẦU NGUỒN PHỤC
NÔNG NGHIỆP

GVHD: TS. Đặng Tuấn Linh

SVTH: Trần Ngọc Bảo - 20215529

Lý Văn Hiếu - 20204829

Hà Duy Long - 20204841

Nguyễn Cảnh Chi - 20172973

Hà Nội, tháng 1, năm 2024

Mục lục

Chương I. Giới thiệu đề tài.....	3
1. Ý tưởng đề tài.....	3
2. Mục tiêu.....	3
Chương II. Thông tin, phân chia, đánh giá công việc của nhóm.....	4
Chương III. Thiết kế hệ thống.....	5
1. Sơ đồ mô hình tổng quan hệ thống.....	5
2. Giới thiệu các công nghệ sử dụng.....	5
2.1. Cảm biến siêu âm HC-SR04.....	5
2.2. Vi điều khiển ESP32.....	7
2.3. Giao thức MQTT.....	8
2.4. NODE.JS.....	9
2.5. MongoDB.....	10
Chương IV. Triển khai hệ thống.....	12
1. Lắp đặt cảm biến HC-SR04 và chip điều khiển ESP32.....	12
2. Triển khai trang web.....	13
2.1. Giao diện mở đầu.....	13
2.2. Giao diện cho người dùng đăng kí tài khoản mới.....	13
2.3. Giao diện trang web gửi trả kết quả.....	14
2.4. Giao diện thông tin người dùng.....	14
2.5. Triển khai cơ sở dữ liệu.....	15
Chương V. Demo hệ thống.....	16
1. Kết quả khi chạy các đoạn code trong Arduino IDE.....	16
2. Kết quả dữ liệu quan sát được trên MQTT Explorer.....	16
3. Kết quả dữ liệu được web server subscribe về từ các topic device trên MQTT broker.....	17
4. Kết quả dữ liệu sau khi được subscribe được lưu vào MongoDB.....	17
5. Kết quả dữ liệu có giá trị vượt ngưỡng cảnh báo được web server phát hiện và gửi email cho người dùng.....	18
6. Kết quả dữ liệu cảnh báo được lưu vào MongoDB.....	18
Chương VI. Kết luận.....	20
1. Những khó khăn gặp phải trong quá trình xây dựng hệ thống.....	20
2. Hướng phát triển.....	20
3. Lời cảm ơn.....	20
Chương VII. Tài liệu tham khảo.....	21

Chương I. Giới thiệu đề tài

1. Ý tưởng đề tài

Việt Nam là một đất nước có nền kinh tế nông nghiệp. Tuy nhiên do tác động từ biến đổi khí hậu, việc sản xuất của nông dân gặp rất nhiều khó khăn và thiệt hại đáng kể do hạn hán kéo dài. Để nông dân có thể đối phó trước với hạn hán, luôn đảm bảo nước tưới cho cây trồng cần can thiệp rất nhiều kỹ thuật khoa học hiện đại vào nông nghiệp.

Trước thực trạng biến đổi khí hậu, thiên tai, hạn hán, lũ lụt ngày càng diễn biến khốc liệt, khó lường, việc dự báo trước mực nước tại các hồ chứa đang được quan tâm đặc biệt. Ứng dụng khoa học kỹ thuật cụ thể là IOT trong công tác đo đạc đang là một hướng đi đúng, nhằm phục vụ công tác chỉ đạo ứng phó, giúp tăng năng suất trong sản xuất nông nghiệp. Với những lý do trên nhóm chúng em quyết định phát triển hệ thống IOT giám sát đo mực nước trong các hồ chứa để phục vụ nông nghiệp.

2. Mục tiêu

Mục tiêu trước mắt: Giúp các vùng có khí hậu khắc nghiệt giám sát được mực nước trong các hồ chứa để chủ động, đảm bảo lượng nước có trong các hồ chứa. Giúp nông dân đảm bảo được sản lượng, năng suất cây trồng.

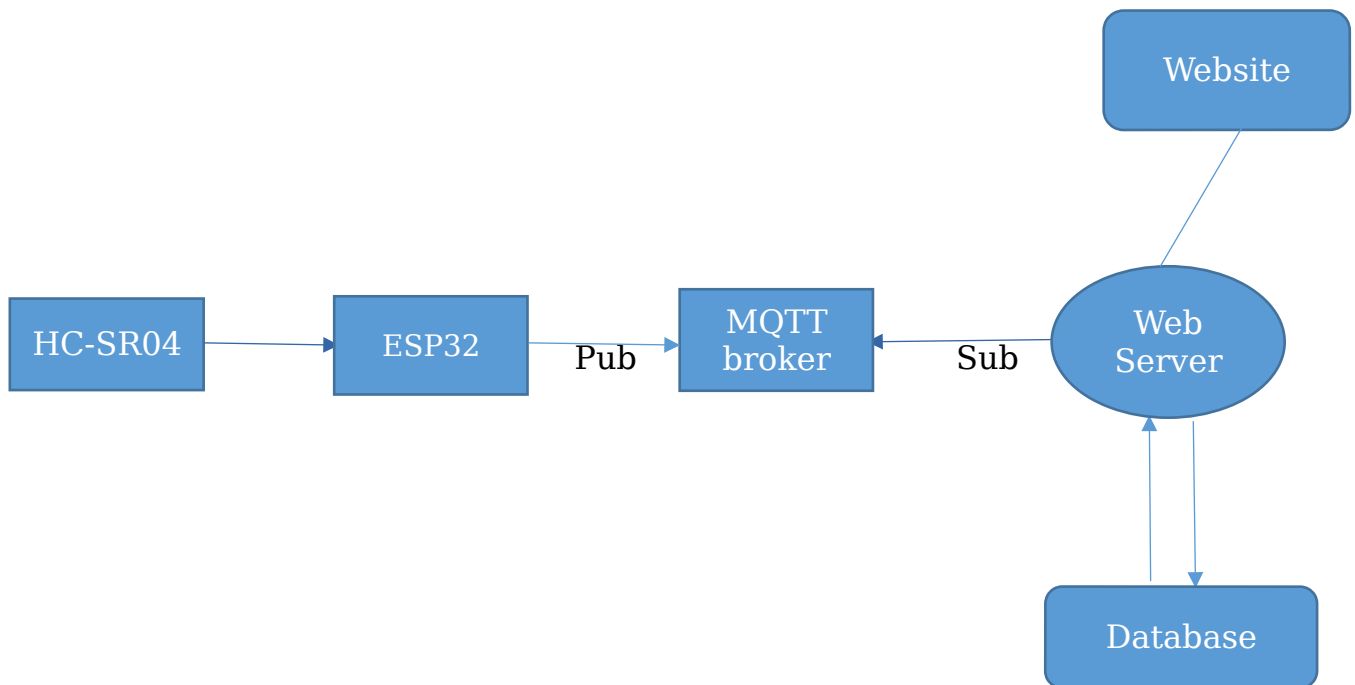
Mục tiêu tương lai: Mở rộng quy mô ra rộng lớn ở nhiều nơi trên đất nước không chỉ trong trồng trọt mà cả trong chăn nuôi động vật thủy sinh.

Chương II. Thông tin, phân chia, đánh giá công việc của nhóm

Họ tên	MSSV	Nhiệm vụ	Đánh giá
Trần Ngọc Bảo (Trưởng nhóm)	20215529	Hardware	100%
Lý Văn Hiếu	20204829	Front-end	100%
Hà Duy Long	20204841	Back-end	100%
Nguyễn Cảnh Chi	20172973	Reports & slides	100%

Chương III. Thiết kế hệ thống

1. Sơ đồ mô hình tổng quan hệ thống



2. Giới thiệu các công nghệ sử dụng

2.1. Cảm biến siêu âm HC-SR04

Cảm biến siêu âm HC-SR04 là một dạng cảm biến module. Cảm biến này thường chỉ là một bản mạch, hoạt động theo nguyên lý thu phát sóng siêu âm bởi 2 chiếc loa cao tần.

Cảm biến siêu âm HC-SR04 thường được kết hợp với các bộ arduino, PIC, AVR,... để chạy một số ứng dụng như : phát hiện vật cản trên xe robot, đo khoảng cách vật,...



Module cảm biến siêu âm HC-SR04

Thông số kỹ thuật HC-SR04:

- Điện áp làm việc: 5VDC
- Dòng điện: 15mA
- Tần số: 40 KHZ
- Khoảng cách phát hiện: 2cm – 4m
- Tín hiệu đầu ra: Xung mức cao 5V, mức thấp 0V
- Góc cảm biến: Không quá 15 độ.
- Độ chính xác cao: Lên đến 3mm
- Chế độ kết nối: VCC / Trig (T-Trigger) / Echo (R-Receive) / GND

Ưu điểm

- Nhỏ gọn
- Giá thành rẻ
- Dễ tìm mua
- Dễ dàng lắp đặt, kết hợp với các bộ kit arduino
- Phát sóng siêu âm, có độ chính xác cao
- Phạm vi đo rộng 2-400cm
- Được hỗ trợ phần code nhiều từ cộng đồng arduino

Nhược điểm

- Hiệu suất làm việc liên tục không cao
- Không thích hợp sử dụng trong môi trường công nghiệp
- Dễ bị nhiễu tác động
- Độ nhạy và dải đo phụ thuộc vào code nạp

Từ các đặc điểm này ta thấy HC-SR04 rất phù hợp với quy mô của project. Tuy nhiên để phát triển ra quy mô siêu lớn chúng ta cần các loại cảm biến có hiệu suất tốt hơn, chất lượng tốt hơn. Ở các mức độ phát triển lớn, quy mô lớn hơn chúng ta có các lựa chọn phù hợp hơn như: dinel (của cộng hòa Séc), Omron (của Nhật Bản), Carlo Gavazzi (của Italya), HAWK (của Hoa Kỳ)..

2.2. Vi điều khiển ESP32

ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.



Thông số kỹ thuật ESP32:

- Bộ vi xử lý LX6 32-bit lõi đơn hoặc lõi kép với xung nhịp lên đến 240 MHz.
- 520 KB SRAM, 448 KB ROM và 16 KB SRAM RTC.
- Hỗ trợ kết nối Wi-Fi 802.11 b / g / n với tốc độ lên đến 150 Mbps.
- Hỗ trợ cho cả thông số kỹ thuật Bluetooth v4.2 và BLE cổ điển.
- 34 GPIO có thể lập trình.
- 18 kênh SAR ADC 12 bit và 2 kênh DAC 8 bit
- Kết nối nối tiếp bao gồm 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.

- Ethernet MAC cho giao tiếp mạng LAN vật lý (yêu cầu PHY bên ngoài).
- 1 bộ điều khiển host cho SD / SDIO / MMC và 1 bộ điều khiển slave cho SDIO / SPI.
- Động cơ PWM và 16 kênh LED PWM.
- Khởi động an toàn và mã hóa Flash.
- Tăng tốc phần cứng mật mã cho AES, Hash (SHA-2), RSA, ECC và RNG.

Ưu điểm:

- Quen thuộc với người sử dụng Arduino
- Code nhanh dễ, có nhiều nguồn hỗ trợ

Nhược điểm:

- Không tối ưu code
- Thiếu một số chức năng

Với những đặc điểm trên modul ESP32 có thể đảm nhiệm vai trò phù hợp với các dự án có các quy mô lớn hoặc nhỏ.

2.3. Giao thức MQTT

MQTT (Message Queue Telemetry Transport) là một giao thức nhắn tin gọn nhẹ được thiết kế để liên lạc nhẹ giữa các thiết bị và hệ thống máy tính. MQTT được thiết kế ban đầu cho các mạng SCADA, các kịch bản sản xuất và băng thông thấp, MQTT đã trở nên phổ biến gần đây do sự phát triển của Internet-of-Things (IoT).

Ưu điểm của MQTT

Giao thức MQTT cho phép hệ thống SCADA của bạn truy cập dữ liệu IoT. Giao thức MQTT giúp đảm bảo truyền dữ liệu diễn ra trơn tru với băng thông thấp, giảm tải cho CPU và RAM. MQTT mang lại nhiều lợi ích mạnh mẽ cho quy trình :

- Dữ liệu được truyền tải ngay lập tức, không phụ thuộc nội dung được truyền.
- Sử dụng TCP/IP làm giao thức nền.
- Tăng khả năng mở rộng và giảm đáng kể tiêu thụ băng thông mạng.
- Tiết kiệm thời gian xây dựng và phát triển.
- Chi phí thấp nhưng vẫn đảm bảo an toàn, bảo mật.
- Được sử dụng trong các ngành công nghiệp và các công ty lớn như Amazon, Facebook,...

Hạn chế của MQTT

- MQTT có chu kỳ truyền dữ liệu chậm hơn so với một số giao thức khác như CoAP.
- MQTT không được mã hóa trực tiếp, tên người dùng và mật khẩu được gửi dưới dạng văn bản rõ ràng. Thay vào đó, giao thức sử dụng TLS/SSL để bảo mật. SSL/TLS không phải là giao thức nhẹ.
- Rất khó để tạo ra một mạng MQTT mở rộng toàn cầu.
- Các thách thức MQTT khác liên quan đến khả năng tương tác và xác thực.

Kết lại, sự tồn tại của giao thức MQTT đóng vai trò rất quan trọng trong quá trình phát triển của IoT nói riêng và thế giới công nghệ nói chung.

2.4. NODE.JS

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Đây là một nền tảng (platform) phát triển độc lập dựa trên V8 JavaScript engine. Đây là một trình thông dịch thực thi mã JavaScript cho phép tạo các ứng dụng web như video clip và diễn đàn, đặc biệt có thể mở rộng nhanh chóng và dễ dàng cũng như thu hẹp phạm vi hoạt động của các trang mạng xã hội.

Đây được xem là một lợi thế khi NodeJS có thể hoạt động trên nhiều nền tảng hệ điều hành khác nhau, từ Windows, Linux đến OS X. NodeJS cung cấp một thư viện phong phú dưới dạng các mô-đun Javascript khác nhau giúp đơn giản hóa việc lập trình và giảm thiểu thời gian cần thiết.

Ưu Điểm

- IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
- Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
- Chia sẻ cùng code ở cả phía client và server.
- NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
- Cộng đồng hỗ trợ tích cực.
- Cho phép stream các file có kích thước lớn

Nhược điểm:

- Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
- Khó thao tác với cơ sở dữ liệu quan hệ.
- Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
- Cần có kiến thức tốt về JavaScript.

- Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

IoT đã trở nên phổ biến mạnh mẽ trong thập kỷ gần đây và kể từ đó Node.js là một trong những giải pháp được ưa thích nhất cho các tổ chức muốn xây dựng các hệ thống giải pháp IoT công cộng và riêng tư.

Lợi thế cốt lõi của Node.js là khả năng xử lý các yêu cầu đồng thời liên mạch với hàng nghìn sự kiện được phát hành bởi hàng tỷ thiết bị trên mạng của nó. Hơn nữa, Node.js hoạt động trên các kênh và luồng có thể ghi và đọc được, đó là lý do tại sao nó là nền tảng phù hợp nhất để phát triển ứng dụng IoT.

2.5. MongoDB

MongoDB hay Mongo Database là phần mềm cơ sở dữ liệu opensource (mã nguồn mở) dạng NoSQL hỗ trợ nhiều nền tảng lập trình và được thiết kế theo kiểu hướng đối tượng. Những bảng dữ liệu trong MongoDB (các Collection) được thiết kế với cấu trúc linh hoạt cho phép dữ liệu được lưu trữ không cần phải tuân theo định dạng cây trúc nào.

Ưu Điểm

- Do sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi Collection đều có thiết kế kích thước và thuộc những document khác nhau. Tuy nhiên chúng lại khá linh hoạt khi tiến hành lưu trữ bởi vậy nếu người dùng muốn lưu thêm dữ liệu chỉ cần insert là xong.
- Những dữ liệu lưu trong hệ thống của MongoDB không bị ràng buộc nhau, không bị phụ thuộc bởi khóa chính hay khóa phụ như RDBMS nên khi thực hiện các thao tác thêm, sửa, xóa thì sẽ đơn giản hơn việc kiểm tra ràng buộc như trong RDBMS.
- Khả năng mở rộng tốt của MongoDB được đánh giá cao bởi nó sử dụng cụm các node chứa những dữ liệu giao tiếp được với nhau được gọi là Cluster. Từ đó để mở rộng bạn chỉ cần thêm một node vào hệ thống.
- Những index cho từng dữ liệu sẽ là tự động để hỗ trợ truy vấn thông tin nhanh và đạt hiệu suất cao.
- Tốc độ truy vấn đáng kể của MongoDB là một lợi thế so với những hệ quản trị cơ sở dữ liệu khác. Thử nghiệm cho thấy ở một lượng dữ liệu chung, MongoDB có khả năng insert nhanh gấp 100 lần so với HTQ SQL.

Nhược điểm:

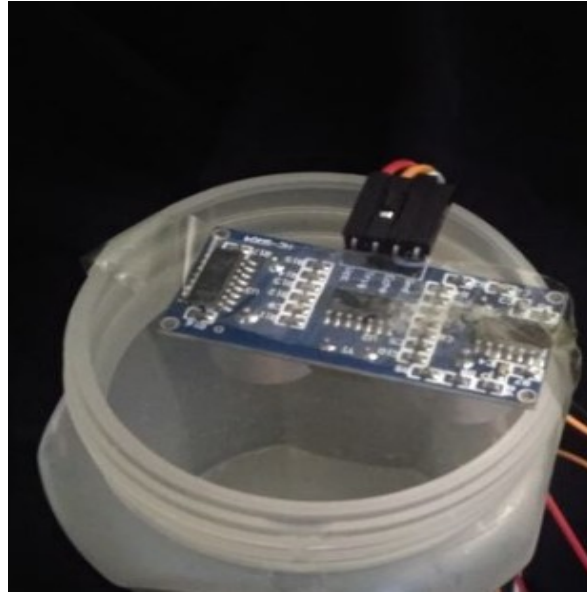
- Chính vì không bị ràng buộc các trường dữ liệu như trong RDBMS nên bạn cần phải hết sức cẩn thận khi thao tác trên những dữ liệu lưu trữ để tránh những kết quả ngoài ý muốn tác động xấu đến dữ liệu.
- Đôi khi sẽ tốn bộ nhớ do dữ liệu lưu ở dạng key-value nên những Collection chỉ khác về value mà có phần giống nhau về key. Và vì không hỗ trợ join như RDBMS nên sẽ có tình trạng dư thừa dữ liệu.

- Khi thực hiện các tác vụ thêm, sửa, xóa thì MongoDB chưa thể cập nhật tức khắc vào ổ cứng mà cần 60 giây sau để ghi toàn bộ những thay đổi dữ liệu từ RAM xuống ổ cứng. Điều này khá rủi ro nếu như trong 60 giây đó có sự cố xảy ra như mất điện sẽ gây mất dữ liệu.

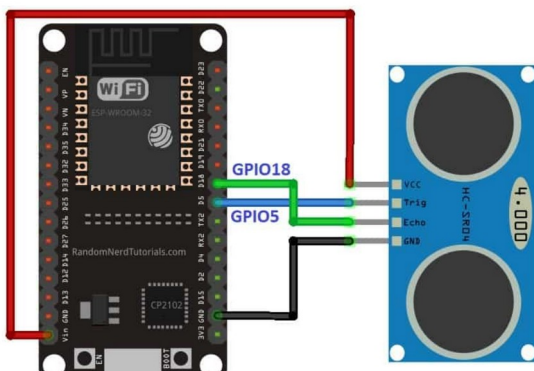
Chương IV. Triển khai hệ thống

1. Lắp đặt cảm biến HC-SR04 và chip điều khiển ESP32

Cảm biến HC-SR04 là loại cảm biến siêu âm hoạt động với nguyên lý phát sóng siêu âm và thu sóng phản xạ. Vì thế cần lắp đặt hướng 2 chiều loa vuông góc với bề mặt chất lỏng như hình dưới:

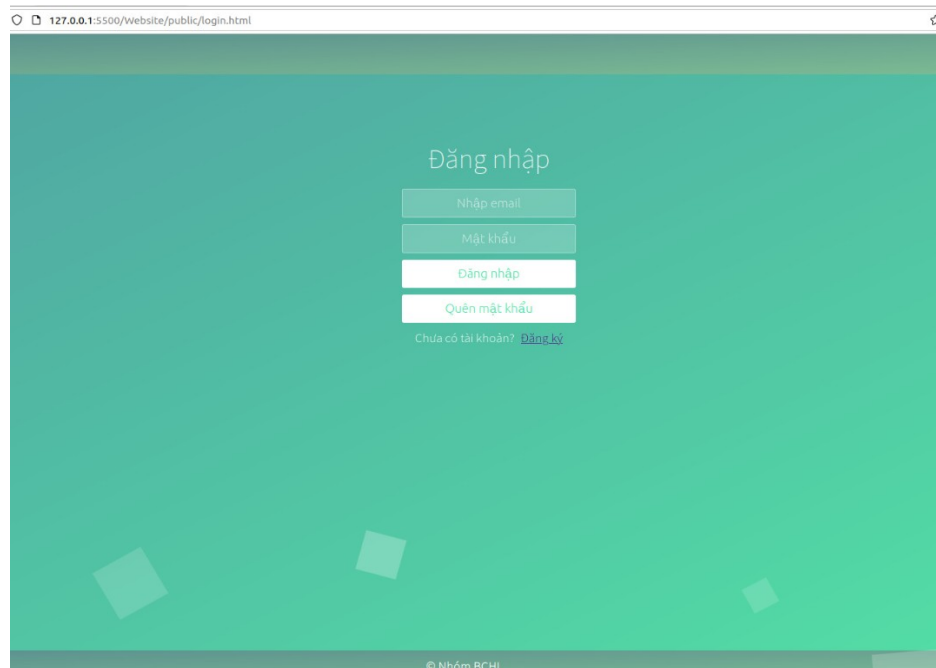


Sau đó thực hiện kết nối các chân của cảm biến với chip ESP32 thông qua dây cáp:



2. Triển khai trang web

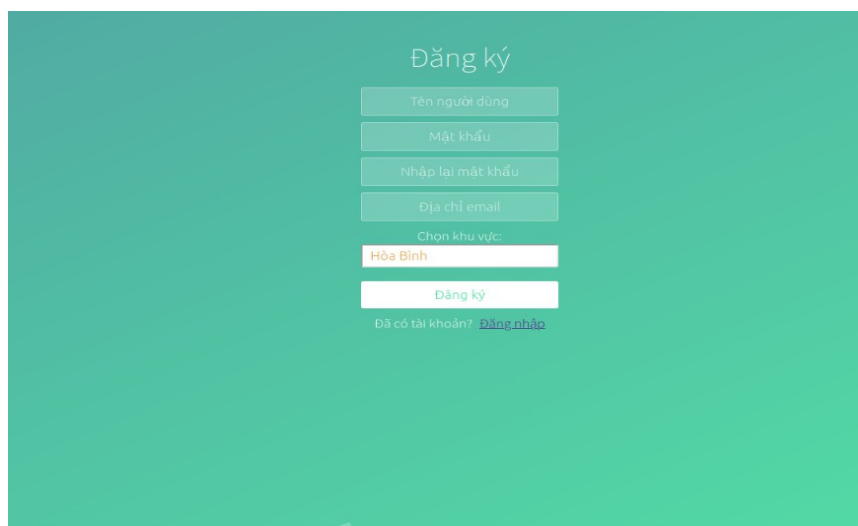
2.1. Giao diện mở đầu



Giao diện mở đầu gồm các chức năng:

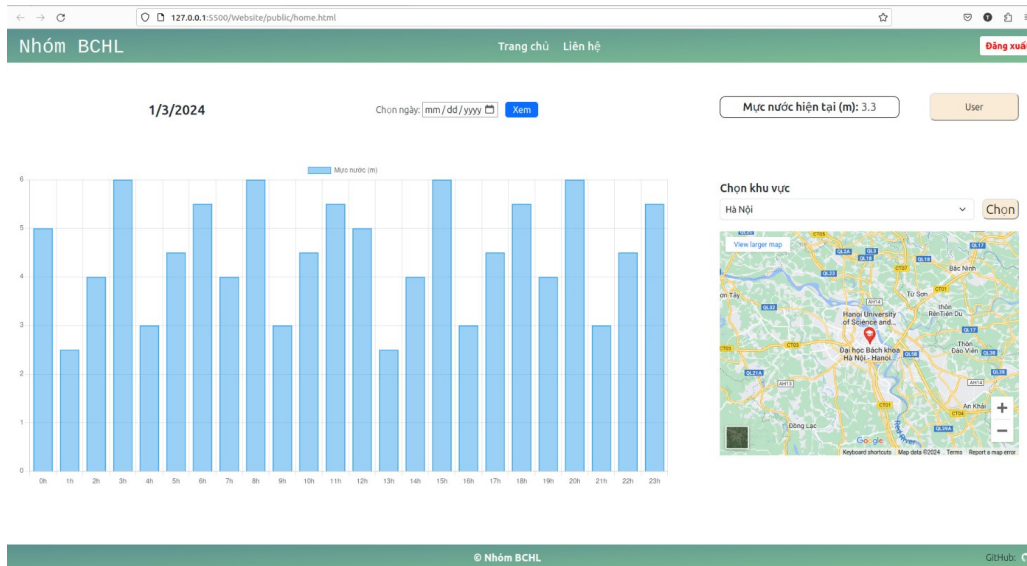
- Nhập email và mật khẩu để đăng nhập
- Quên mật khẩu: Chức năng dùng để lấy lại mật khẩu nếu như đã đăng kí tài khoản trước đó
- Đăng ký : Đăng kí tài khoản cho người dùng mới

2.2. Giao diện cho người dùng đăng kí tài khoản mới



Giao diện yêu cầu người dùng nhập email, mật khẩu, xác nhận lại mật khẩu, chọn tên người dùng, khu vực để đăng kí. Nếu như đã có tài khoản có thể chuyển tiếp qua giao diện đăng nhập thông qua chức năng “*Đăng nhập*” ở góc dưới màn hình

2.3. Giao diện trang web gửi trả kết quả



Giao diện gồm chức năng chọn khu vực, chọn thời điểm điểm để xem mực nước. Và hệ thống sẽ trả kết quả về kết quả mực nước về bằng biểu đồ cột.

2.4. Giao diện thông tin người dùng

The screenshot shows a web interface for user information. At the top, there is a navigation bar with "Nhóm BCHL", "Trang chủ", "Liên hệ", and a "Đăng xuất" button. Below the navigation bar, the date "1/3/2024" is shown, along with a date selection dropdown and a "Xem" button. A "Mức nước hiện tại (m): 3.3" is displayed next to a "User" button. The main content area features a form titled "Thông tin tài khoản" with the following fields and buttons:

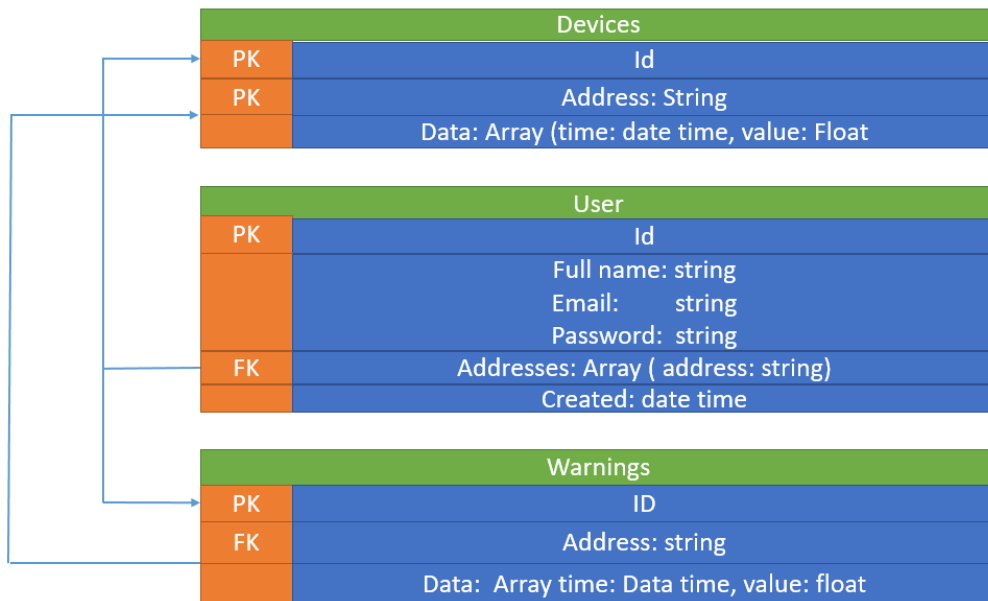
- Tài khoản: User
- Email: user@gmail.com (with a "Đổi email" button)
- Mật khẩu: *** (with a "Đổi mật khẩu" button)
- Khu vực:
- Thêm khu vực: Hòa Bình (with a "Thêm" button)
- Xóa khu vực: Hòa Bình (with a "Xóa" button)

The footer contains "© Nhóm BCHL" and "GitHub" links.

Giao diện này giúp người dùng đổi thông tin cá nhân (email, mật khẩu), thêm hoặc xóa khu vực, có thể đăng xuất (chọn chức đăng xuất) cũng như quay lại trang kết quả (click vào trang chủ).

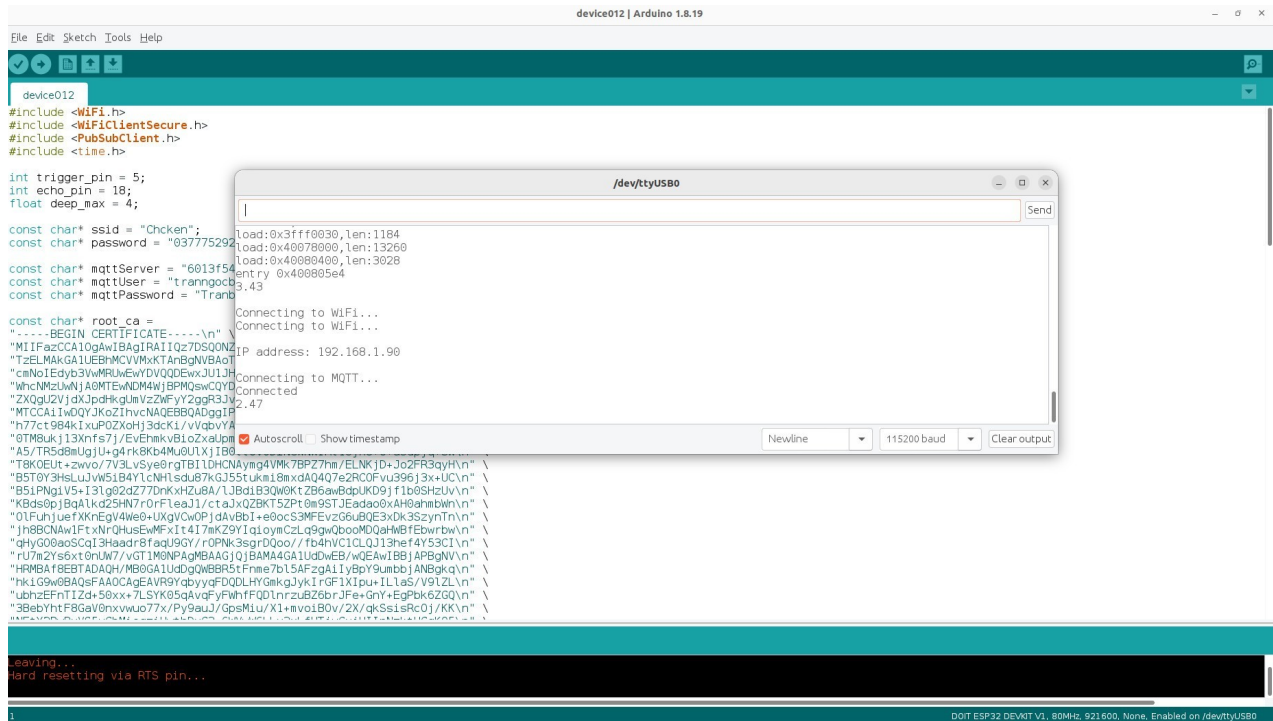
2.5. Triển khai cơ sở dữ liệu

Sơ đồ triển khai cơ sở dữ liệu:



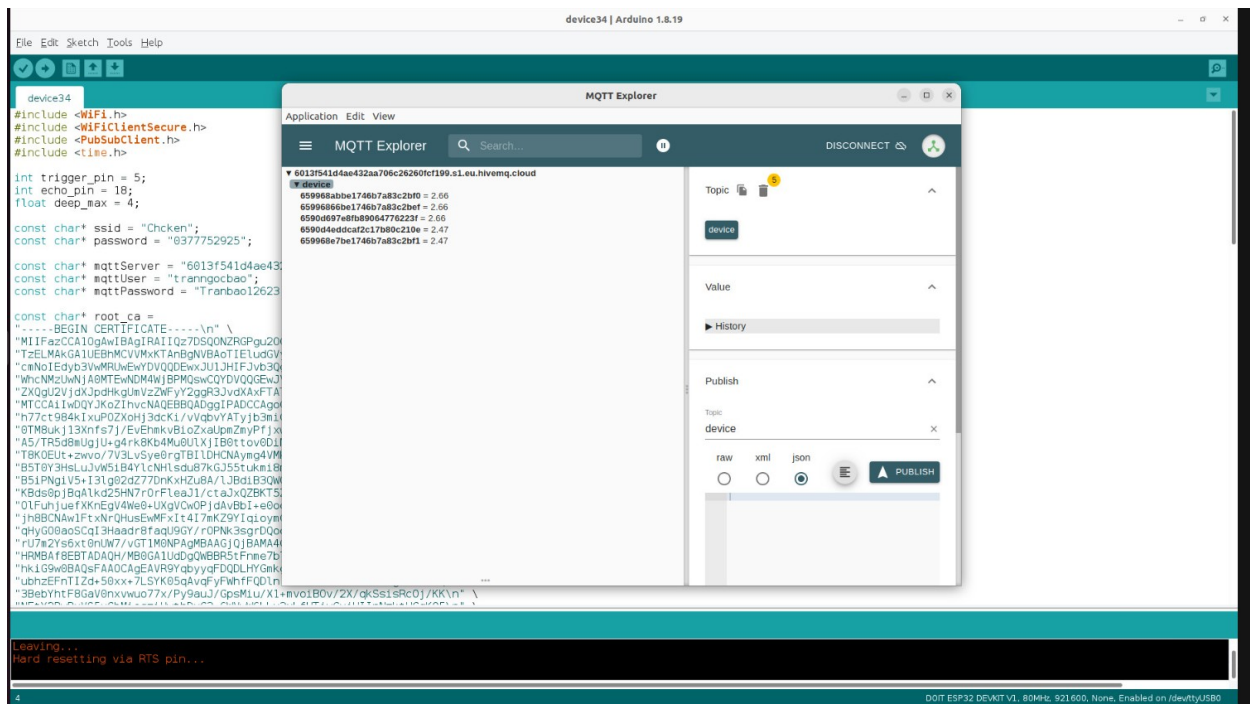
Chương V. Demo hệ thống

1. Kết quả khi chạy các đoạn code trong Arduino IDE



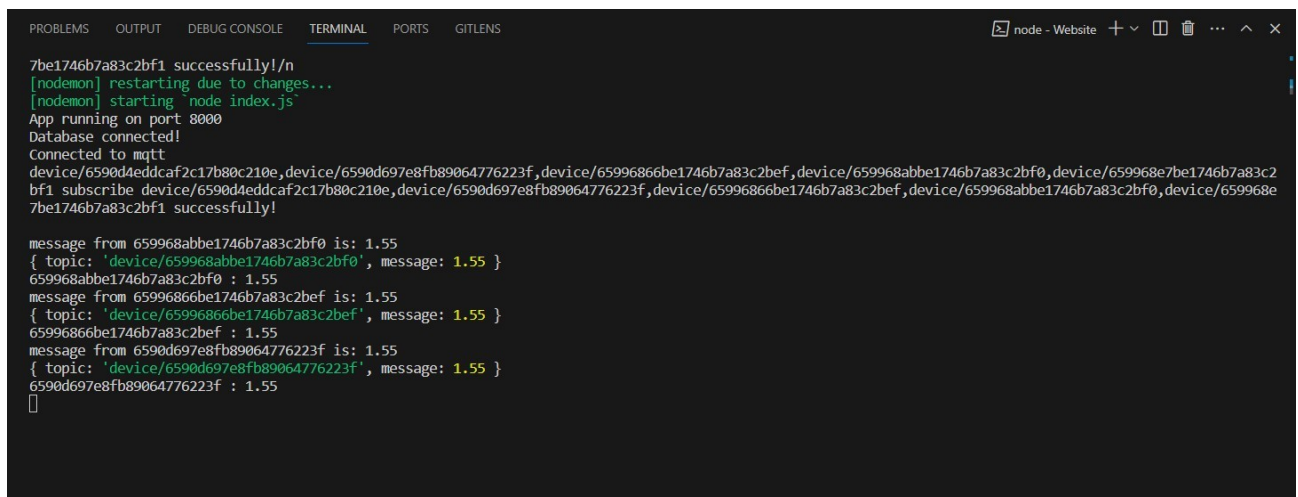
Tiến hành kết nối với Wifi và MQTT Broker (Được cung cấp miễn phí bởi HiveMQ). Đo dữ liệu 10 phút 1 lần, 5 lần 1 giờ và gửi dữ liệu trung bình đồng thời gửi cảnh báo nếu có về MQTT broker

2. Kết quả dữ liệu quan sát được trên MQTT Explorer



Sử dụng MQTT Explorer để quan sát các dữ liệu được ESP32 gửi lên MQTT Broker vào các topic.

3. Kết quả dữ liệu được web server subscribe về từ các topic device trên MQTT broker



Web server được tạo nên bởi Node.js luôn bật để Subscribe dữ liệu từ các địa điểm đầu nguồn được Publish hàng giờ trên MQTT Broker

4. Kết quả dữ liệu sau khi được subscribe được lưu vào MongoDB

```

    _id: ObjectId('659968abbe1746b7a83c2bf0')
    address: "Sơn La"
  ▾ data: Array (26)
    ▾ 0: Object
      value: 1.1
      _id: ObjectId('659971a1f5a18b3ba78f1c5b')
      time: 2024-01-06T15:28:33.904+00:00
    ▾ 1: Object
      value: 1.2
      _id: ObjectId('659971abf5a18b3ba78f1c5f')
      time: 2024-01-06T15:28:43.601+00:00
    ▾ 2: Object
      value: 1.3
      _id: ObjectId('659971b8f5a18b3ba78f1c64')
      time: 2024-01-06T15:28:56.622+00:00
    ▾ 3: Object
      value: 1.48
      _id: ObjectId('659ab42b6899b1563d0aaf4b')
      time: 2024-01-07T14:24:43.350+00:00
    ▾ 4: Object
      value: 1.49

```

Dữ liệu sau khi nhận được sẽ được từ các topic device sẽ được Web server lưu luôn vào MongoDB theo các address tương ứng.

5. Kết quả dữ liệu có giá trị vượt ngưỡng cảnh báo được web server phát hiện và gửi email cho người dùng

```

{ topic: 'device/65996866be1746b7a83c2bef', message: 3.54 }
65996866be1746b7a83c2bef : 3.54
[SEND_MESSAGE_ERROR] Error: Missing credentials for "PLAIN"
    at SMTPConnection.formatError (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:790:19)
    at SMTPConnection.login (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:444:38)
    at D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-transport\index.js:272:32
    at SMTPConnection.<anonymous> (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:213:17)
    at Object.onceWrapper (node:events:631:28)
    at SMTPConnection.emit (node:events:517:28)
    at SMTPConnection.actionEHL0 (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:1347:14)
    at SMTPConnection.processResponse (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:969:20)
    at SMTPConnection._onData (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:755:14)
    at SMTPConnection._onSocketData (D:\HUST\2023.1\IoT\ProjectIoT\Website\node_modules\nodemailer\lib\smtp-connection\index.js:193:44) {
  code: 'EAUTH',
  command: 'API'
}

```

Khi Subscribe được dữ liệu cảnh báo, Web server lập tức gửi email cảnh báo cho người dùng đã đăng ký ở các địa điểm có cảnh báo.

6. Kết quả dữ liệu cảnh báo được lưu vào MongoDB

```
_id: ObjectId('659ab58ea3da16ce18551b53')
address: "Sơn La"
▼ data: Array (3)
  ▼ 0: Object
    value: 3.98
    _id: ObjectId('659ab76c3f76c62364e29863')
    time: 2024-01-07T14:38:36.225+00:00
  ▼ 1: Object
    value: 3.54
    _id: ObjectId('659c02fc20cd5f492c0eeaed')
    time: 2024-01-08T14:13:16.343+00:00
  ▼ 2: Object
    value: 3.97
    _id: ObjectId('659c033820cd5f492c0eeb87')
    time: 2024-01-08T14:14:16.362+00:00
  __v: 3
```

Dữ liệu sau khi nhận được sẽ được từ các topic warning cũng được Web server lưu luôn vào MongoDB theo các address tương ứng.

Chương VI. Kết luận

1. Những khó khăn gặp phải trong quá trình xây dựng hệ thống

- Do các thành viên trong nhóm còn chưa có kinh nghiệm trong việc xây dựng một hệ thống IOT nên còn gặp nhiều khó khăn, đặc biệt là do chưa sử dụng, phần lắp đặt và chạy thử nghiệm phần cứng (ESP32 và cảm biến).
- Khi tạo dựng database do chưa nắm bắt được dữ liệu thực tế nên xảy ra việc database chỉ phù hợp cho hệ thống nhỏ, chưa xử lý việc dữ liệu lớn.
- Các thành phần được phát triển bởi các thành viên khác nhau nên đôi khi có sự mất đồng bộ dẫn đến kết quả thử nghiệm bị sai lệch, đòi hỏi cần có sự thống nhất tỉ mỉ hơn về mặt thành phần hệ thống và luồng hoạt động

2. Hướng phát triển

Trong tương lai nếu tiếp tục có cơ hội hợp tác, chúng em dự định sẽ hoàn thiện thêm các chức năng hiện có, đồng thời sẽ bổ sung thêm một số tính năng mới như bổ sung, thay thế các thiết bị điều khiển, cảm biến khác có hiệu suất làm việc tốt hơn cho hoặc mở rộng phạm vi hoạt động cho phép nhiều cảm biến cùng hoạt động và chia sẻ dữ liệu giữa các nguồn nước.

3. Lời cảm ơn

Bản báo cáo này đánh dấu việc hoàn thành và kết thúc học phần. Chúng em xin cảm ơn thầy TS. Đặng Tuấn Linh đã giảng dạy, chỉ bảo chúng em rất nhiệt tình và giúp chúng em trang bị thêm nhiều kiến thức về IOT.

Chương VII. Tài liệu tham khảo

1. Tài liệu bài giảng bộ môn IoT và ứng dụng – Phạm Ngọc Hưng, Nguyễn Đình Thuận, Đặng Tuấn Linh
2. Tài liệu hướng dẫn về MongoDB: <https://www.mongodb.com/docs>
3. Tài liệu tham khảo về hệ thống:
<https://mecsuvn.vn/ho-tro-ky-thuat/giam-sat-muc-nuoc-khong-tiep-xuc-dua-tren-iot-voi-esp32-va-hcsr04.yLD>
4. Tài liệu tham khảo về Nodejs:
<https://www.w3schools.com/nodejs/default.asp>