**Mục tiêu:**

**Phần nghiên cứu:**

* **Tìm hiểu nắm vững các nên tảng, thư viện cũng như các công cụ để thực hiện dự án này.**
* **Áp dụng các kiến thức về thiết kế mềm, kĩ thuật lập trình vào dự án.**
* **Tổ chức dự án cho phép dễ dàng bảo trì, mở rộng.**

**Phần kết quả:**

* **Hiện thực mạch đo giá trị ppm (parts per million) trong môi trường dung dịch.**
* **Hệ thống board điều khiển đáp ứng các tiêu chí:**
  + **Vận hành trong thời gian dài, ổn định.**
  + **Chức năng thu thập dữ liệu các cảm biển phải hoạt động tốt, ổn định, sai số chấp nhận được.**
  + **Chức năng điều khiển các thiết bị như máy bơm nước, xục oxi, đèn chiếu sáng.. đáp ứng thời gian thực.**
  + **Các kết nối về sóng wifi ổn định, cung cấp chức năng cấu hình cho module wifi.**
  + **Protocol giao tiếp giữa board điều khiển và web-server hoạt động chính xác.**
  + **Thiết kế hiện thực board điều khiển về mặt phần cứng, phần khung, vỏ bảo vệ.**
* **Hiện thực mô hình thủy canh thủy thu nhỏ cho demo.**

1. **Công nghệ và công cụ.**
   1. **Arduino**

* **Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.**
* **Arduino đồng thời là nền tảng mã nguồn mở được phát triển rộng rãi, dễ tiếp cận cho người mới bắt đầu.**
  1. **Arm**
* **Cấu trúc ARM (viết tắt từ tên gốc là Advanced RISC Machine) là một loại cấu trúc vi xử lí kiểu RSIC được sử dụng rộng rãi trong các thiết kế nhúng. Do có đặc điểm tiết kiệm năng lượng, các bộ cpu ARM chiếm ưu thế trong các sản phẩm điện tử di động, mà với các sản phẩm này việc tiêu tán công suất thấp là một mục tiêu thiết kế quan trọng hàng đầu.**
* **Đi kèm với hệ thống vi xử lí là nền tảng lập trình đa dạng, các thư viện mã nguồn mở, hệ thống các board tham chiếu** 
  1. **Mbed**
* **Mbed là nền tảng và hệ điều hành được phát triển cho các thiết bị kết nối internet dữa trên các thiết bị 32-bit Arm Cortex-M.**
* **Mbed cung cấp nhiều phương pháp tiếp cận lập trình khác nhau:**
  + **Online**
    - **Hinh ảnh (DevicePhotos/mbed.PNG)**
    - **Chúng ta có thể lập trình online, biên dịch, tải file thực thi về rồi nạp vào thiết bị.**
  + **Offline**
    - **Lập trình trên nền tảng linux và make.**

1. Thiết kế
   1. Tổng quan mô hình
      1. **Các thành phần chức năng**

**Hệ thống board điều khiển chia ra làm 3 thành phần chức năng chính:**

* **Thu thập dữ liệu các cảm biến**
* **Định thời, điều khiển các actuator (máy bơm, xục oxi, đèn chiếu sáng...) theo thời gian thực.**
* **Tuyền nhận dữ liệu với web-server thông qua môi trường wifi.**
  + 1. **Các module, cảm biến**

1. **Nhiệt độ, độ ẩm**
   * + **Nhiệt đô, độ ẩm được đo bởi module DHT22**
     + **Hình ảnh (DevicePhotos/DHT22.jpg) Y**
     + **Thông số kĩ thuật**
       - **Nguồn 3-5 VDC.**
       - **Dòng sử dụng: max là 2.5mA.**
       - **Đo tốt ở độ ẩm 0x100RH%, sai số 2-5%**
       - **Tần số lấy mẫu tối đa 0.5HZ (2 giây 1 lần).**
       - **Kích thước 27mm x 58mm x 13.5mm.**
       - **3 chân, khoảng cách 0.1 inch.**
2. **Cường độ ánh sáng**
   * + **Cường độ ánh sáng được đo bởi module BH1750.**
     + **Hình ảnh (DevicePhotos/BH1750.PNG) Y**
     + **Thông sô kĩ thuật:**
       - **Nguồn: 3-5VDC.**
       - **Protocol giao tiếp: I2C**
       - **Khoảng đo: 1->65535 lux.**
       - **Kích cỡ: 21x16x3.3 mm.**
3. **Relay**
   * + **Các actuators (Máy bơm nước, xục Oxi, led...) được kích thông qua relay.**
     + **Hình ảnh (DevicePhotos/Relay.jpg) Y**
     + **Thông số kĩ thuật:**
       - **Nguồn: 5 VDC.**
       - **Điện áp kích: 5 VDC.**
4. **Thanh ghi dịch**
   * + **Thanh ghi dịch sử dụng chip 74HC595.**
     + **Hình ảnh (DevicePhotos/Register.jpg) Y**
     + **Thông số kĩ thuật:**
       - **Nguồn: 5 VDC.**
       - **Các chân Output: 5 VDC.**
       - **Protocol giao tiếp: SPI.**
5. **Thời gian thực**
   * + **Thời gian thực được đo sử dụng module Realtime clock DS1307.**
     + **Hình ảnh (DevicePhotos/DS1307.PNG) Y**
     + **Thông số kĩ thuật:**
       - **Nguồn: 5 VDC.**
       - **Khả năng lưu trữ: 32K (EEPROM AT24C32).**
       - **Protocol: I2C.**
       - **Có pin chạy độc lập.**
       - **Tần số ra 1 HZ.**
       - **Kích thước: 16x22x23 mm.**
6. **Board ARM STM32F03C8T6**
   * + **Board arm STM32F103C8T6 được sử dụng để làm board điều khiển các actuators.**
     + **Hình ảnh (DevicePhotos/smt32.jpg) Y**
     + **Thông số kĩ thuật:**
       - **Hình ảnh (DevicePhotos/stm32KT.PNG) Y**
7. **Board NodeMCU ESP8266**
   * + **Board NodeMCU được sử dụng để làm board truyền nhận dữ liệu qua wifi.**
     + **Hình ảnh (DevicePhotos/NodeMCU.jpg) Y**
     + **Thông số kĩ thuật:**
       - **Hình ảnh (DevicePhotos/NodeMCUKT.PNG) Y**
   1. **Hoạt động, chức năng**
      1. **Sơ đồ hoạt động.**
   * **Hình ảnh (DevicePhotos/mohinh.PNG) Y**
     1. **Thu thập dữ liệu cảm biến**

* **Các cảm biến được kết nối trực tiếp với board NodeMCU và lấy dữ liệu trực tiếp.**
  + **Nhiệt độ, độ ẩm: module DHT22 giao tiếp 1 wire với board NodeMCU. Dữ liệu nhiệt độ (C), độ ẩm (%) được đọc sau mỗi (1s).**
  + **Cường độ ánh sáng: module BH1750 giao tiếp I2C với board NodeMCU. Dữ liệu cường độ ánh sáng (lux) được đọc sau mỗi (1s).**
  + **Module ppm hoạt động độc lập, đọc dữ liệu từ môi trường dung dịch, xuất đầu ra analog được đọc bởi board NodeMCU sau mỗi (1s).**
  + **Định thời đọc dữ liệu cảm biến sử dụng timer, timeout là 1s.**
* **Hoạt động đọc , quản lí dữ liệu cảm biến sẽ được thực hiện bởi 1 đối tượng:**
  + **Quản lí các thông tin:**
    - **Nhiệt độ**
    - **Độ ẩm**
    - **PPM**
    - **Ánh sáng**
  + **Hoạt động độc lập, định thời đọc dữ liệu theo sơ đồ sau:**
    - **Hình ảnh (DevicePhotos/sensor.PNG). Y**
    1. **Điều khiển các actuators.**
* **Thông số các actuators:**
  + **Máy bơm nước:**
    - **Nguồn: 12 VDC.**
    - **Công suất:**
    - **Khả năng phun nước độ cao 2m phương thẳng đứng.**
    - **Hình ảnh (DevicePhotos/pump.jpg). Y**
  + **Máy xục oxi:**
    - **Nguồn: 12 VDC.**
    - **Công suất:**
    - **Hình ảnh (DevicePhotos/oxi.jpg) Y**
  + **Đèn chiếu sáng:**
    - **Nguồn: 12 VDC.**
    - **Công suất:**
    - **Hình ảnh (DevicePhotos/led.jpg) Y**
  + **Quạt**
    - **Nguồn: 12 VDC.**
    - **Công suất:**
    - **Hình ảnh (DevicePhotos/fan.jpg) N**
* **Cách điều khiển:**
  + **Hoạt động của thanh ghi dịch:**
  + **Hình ảnh (DevicePhotos/stm\_register.PNG) Y**
    - **Giao tiếp của thanh ghi dịch với board điều khiển là SPI, các chân điều khiển:**
      * **DATA: chân dữ liệu.**
      * **LATCH: đẩy dữ liệu từ thanh ghi bên trong chip xuống các pinout, cụ thể là 8 chân, điện áp ra là 5V.**
      * **CLOCK: tín hiệu clock lấy từ board điều khiển.**
    - **Chân DATA của thanh ghi đầu tiên sẽ nối với chân tín hiệu đầu ra của board điều khiển, DATA của các thanh ghi tiếp theo sẽ lấy từ pinout thứ 9 của thanh ghi liền trước nó.**
    - **Hình ảnh (DevicePhotos/register\_register.PNG) Y**
  + **Các actuators sẽ được kích hoạt trực tiếp thông qua relay:**
  + **Hình ảnh (DevicePhotos/actuator.PNG) Y**
  + **Tín hiệu điều khiển Relay sẽ lấy từ thanh ghi:**
  + **Hình ảnh (DevicePhotos/register\_relay.PNG) Y**
* **Thời gian thực:**
  + **Module Realtime Clock DS1307 sẽ hoạt động độc lập với pin, cụ thể, các thông số thời gian thực trong module không bị ảnh hưởng bởi nguồn nuôi của toàn mạch.**
  + **Thời gian thực (year, month, day, hour, min, sec) có thể được đọc và ghi bởi board điều khiển, cụ thể là NodeMCU.**
  + **Hình ảnh (DevicePhotos/Node\_Realtime.PNG) Y**
  + **Định thời đọc dữ liệu thời gian thực sử dụng timer, timeout là 1/3s.**
* **Tổ chức, quản lí hoạt động của các actuators.**
  + **Chân điều khiển:**
    - **Các actuator được điểu khiển (tắt, mở) thông qua tín hiệu các chân của thanh ghi dịch (74HC595) và relay nên không phụ thuộc vào các pinout của board điều khiển (STM32F103C8T6).**
    - **Có thể tăng số chân điều khiển (cũng như tăng số actuators) bằng cách thêm các thanh ghi dịch, số pinout sử dụng luôn là 3, cụ thể là các chân SPI (DATA, LATCH, CLOCK).**
  + **Thời gian biểu:**
    - **Mỗi actuator sẽ hoạt động theo một thời gian biểu, cụ thể trong thủy canh:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Start** | **Duration** | **Repeat interval** | **Stop** |

* + - * **Start: thời điểm bắt đầu hoạt động.**
      * **Duration: thời lượng actuator được mở.**
      * **Reapeat interval: thời lượng tắt actuator giữa hai lần mở.**
      * **Stop: thời điểm tắt actuator cho đến Start kế tiếp.**
      * **Định dạng thời gian cho mỗi kiểu như trên là hhmmss.**
      * **Mỗi thời gian biểu cho mỗi ngày sẽ bao gồm nhiều mẫu như trên, ví dụ với máy bơm:**
        + **Máy bơm sẽ bắt đầu bơm nước từ 08:00:00 (start), mỗi lần tưới sẽ kéo dài trong 15s (duration), sau đó sẽ tắt trong 10s (repeat interval), lại bật tiếp 15s, cứ thế cho đến 09:00:00 (stop).**
  + **Lifetime của mỗi actuator (phần mềm).**
    - **Hình ảnh (DevicePhotos/actuator\_lifetime.PNG) Y**
    - **Mỗi actuator sẽ là một đối tượng (phần mềm), giải thích lifetime:**
      * **Ban đầu, sẽ không có actuator nào, các actuator không được sét tĩnh.**
      * **Khi nhận gói tin (Add) từ web-server, một đối tượng actuator sẽ được tạo ra, là một node trong danh sách liên kết (linked-list) các actuators. Mỗi đối tượng chứa các thông tin:**
        + **Actuator ID: định danh của actuator.**
        + **Primary: là actuator chính hay phụ.**
        + **Schedule: thời gian biểu hoạt động.**
        + **Started: điều kiện để hoạt động.**
      * **Sau khi được add (Added), đối tượng sẽ chờ nhận thêm hai gói tin (thứ tự bất kì) là (Start) và (Schedule) để chuyển sang trạng thái (Actived).**
        + **Nhận gói (Schedule): đối tượng sẽ lưu lại thông tin schedule.**
        + **Nhận gói (Start): đối tượng sẽ cập nhập thông tin (Started).**
      * **Ở trạng thái (Added) và (Actived) khi nhận được gói (Remove) , đối tượng sẽ bị xóa.**
      * **Ở trạng thái (Actived), đối tượng actuator sẽ hoạt động với thông tin (Schedule) theo thời gian thực đọc được từ module Realtime Clock DS1307.**
  + **Lưu trữ thông tin actuator.**
    - **Để đáp ứng tính độc lập của board điều khiển, các thông tin của mỗi actuator sẽ được lưu lại trong bộ nhớ ROM, tránh trường hợp reset của board ( có thể do nhiều nguyên nhân khác nhau: mất điện, crash..).**
    - **Thông tin actuator được thêm trong trong lúc board đang hoạt động ở thời điểm ngẫu nhiên (runtime), không sét tĩnh, đây là ưu điểm của giải pháp này, cho phép mở rộng, thay đổi các actuator.**
    - **Bộ nhớ lưu trữ là EEPROM, phân vùng lưu trữ sẽ nêu cụ thể trong phần (Lưu trữ dữ liệu) nằm trên board NodeMCU.**
    - **Cấu trúc dữ liệu lưu trữ được nêu cụ thể trong phần hiện thực.**
    - **Dưới đây là cách hoạt động của việc lưu trữ (Save) và tải lại (Load).**
      * **Qui ước: Các sự kiện (Add), (Start), (Schedule), (Update) (Remove) sẽ được gọi chung là (Update). Các trạng thái (Added), (Actived) được gọi chung là (Running).**
      * **Hình ảnh (DevicePhotos/persistent\_actuator.PNG). Y**
    1. Truyền nhận dữ liệu với giao thức MQTT.
* Đối tượng thực hiện công việc này hoạt động trên board NodeMCU với vai trò là một MQTT Client, bao gồm:
  + Lưu trữ các thông tin:
    - MQTT broker.
    - MQTT port.
    - MQTT subcribe channel.
    - MQTT publish channel.
  + Các chức năng:
    - Publish: truyền dữ liệu tới kênh web-server đăng kí( cũng đóng vai trò là một MQTT Client).
    - Subcribe: đăng kí kênh nhận dữ liệu từ web-server.
* Các thông tin của đối tượng MQTT Client cũng được lưu trữ trong ROM, nêu trong phần (**Lưu trữ dữ liệu**), và cho phép cấu hình (**Cấu hình WiFi**).
* Đối tượng hoạt động độc lập theo sơ đồ dưới:
  + Hình ảnh (DevicePhotos/mqtt\_client.PNG). Y
    1. Cấu hình, hoạt động của các chế độ WiFi.
* Board NodeMCU cho phép hoạt động ở 3 chế độ:
  + **Station**: đóng vai trò là một thiết bị kết nối vào mạng wifi tương tự như điện thoại.
  + **Access Point**: Phát wifi, cho phép thiết bị khác kết nối vào.
  + **Both**: cả hai chức năng trên.
* Ở dự án này, 2 chế độ được sử dụng là (**Station**) và (**Access Point**), cụ thể:
  + Station: Ở chế độ này, board sẽ thực hiện chức năng của MQTT Client.
  + Access Point: cung cấp chức năng của một Web-server nhỏ, cho phép cấu hình:
    - Các thông số wifi (tên wifi, mật khẩu) khi board hoạt động ở (**Station)**.
    - Thông tin của đối tượng MQTT Client (broker, port...).
    - Các thông tin sau khi cấu hình sẽ được lưu trữ trong EEPROM.
* Chức năng chuyển chế độ sẽ được thực hiện bởi một nút nhấn.
* Các chế độ hoạt động như sau:
  + Hình ảnh (DevicePhotos/wifi\_station.PNG). Y
  + Hình ảnh (DevicePhotos/wifi\_ap.PNG). Y
  + Hình ảnh (DevicePhotos/wifi\_application.PNG). Y
    1. Đệm dữ liệu (FIFO)
* Chức năng:
  + Để đảm bảo tính ổn định, đồng bộ giữa quá trình **Nhận**-**Xử lí** các gói tin.
  + Giảm tải thời gian xử lí **interrupt** (có khả năng gây xung đột các task).
* Các vị trí cần sử dụng bộ đệm:
  + Khi nhận các gói tin **Serial (uart).**
  + Khi gửi các gói tin MQTT: việc gửi các gói tin MQTT là thông qua môi trường wifi, nên không phải lúc nào cũng đảm bảo tính kết nối giữa client với broker, vì vậy cần bộ đệm để lưu trữ các gói tin trước khi gửi.
* Cấu trúc bộ đệm: các gói tin xếp theo thứ tự FIFO.
  + 1. Lưu trữ dữ liệu.
* Các thông tin cần được lưu trữ trong EEPROM:
  + Thông tin wifi station: ssid, password.
  + Thông tin MQTT: broker, port, subcribe channel, publish channel.
  + Actuators.
* Việc lưu trữ và quản lí sẽ do một đối tượng thực hiện.
* EEPROM đặt trên board NodeMCU, đối tượng Actuators lại hoạt động trên board STM nên sẽ cần một giao thức để tải dữ liệu từ NodeMCU sang STM.
  + 1. Giao tiếp giữa NodeMCU và STM32F103C8T6.
* Giao thức lớp dưới được sử dụng là **UART.**
* Một giao thức đơn giản được xây dựng trên lớp **UART** nhằm:
  + Truyền, nhận các gói tin lớn.
  + Đảm bảo tính toàn vẹn của gói tin.
* Cấu trúc giao thức:
  + Hình ảnh (DevicePhotos/uart\_protocol.PNG). Y
  + Các thành phần chính:
    - Timer: định thời đọc dữ liệu có sẵn trong UART ở lớp dưới.
    - Buffer: bộ đệm tạm thời cho dữ liệu rời rạc.
    - Privacy: chính sách truyền nhận.
    - Data Queue: gói tin toàn vẹn được đưa vào queue (FIFO), nêu cụ thể trong phần (**Đệm dữ liệu**).
    - UART: giao thức truyền nhận lớp dưới.
  + Cách hoạt động:
    - Hình ảnh (DevicePhotos/uart\_flow.PNG). Y
  1. **Kiến trúc phần mềm**
     1. **Sequence diagram**
* **Cấu hình wifi:** 
  + **Hình ảnh (DevicePhotos/sq\_switch\_wifi.PNG). Y**
* **Hoạt động của web-server hoạt động trên board Node MCU.**
  + **Hình ảnh (DevicePhotos/sq\_web.PNG). Y**
* **Truyền nhận dữ liệu:**
  + **Hình ảnh (DevicePhotos/sq\_mqtt.PNG) Y**
    1. **Class diagram**

1. **Các class hoạt động trên board NodeMCU.**

* **WiFiStation**
  + **Hình ảnh (DevicePhotos/class\_wifistation.PNG) Y**
* **MqttClient**
  + **Hình ảnh (DevicePhotos/class\_mqttclient.PNG) Y**
* **UserEEPROM**
  + **Hình ảnh (DevicePhotos/class\_usereeprom.PNG) Y**
* **Sensor**
  + **Hình ảnh (DevicePhotos/class\_sensor.PNG) Y**
* **SerialTransport**
  + **Hình ảnh (DevicePhotos/class\_serialtransport.PNG) Y**
* **Webserver**
  + **Hình ảnh (DevicePhotos/class\_webserver.PNG) Y**
* **SwitchAP**
  + **Hình ảnh (DevicePhotos/class\_switchap.PNG) Y**
* **Application**
  + **Hình ảnh (DevicePhotos/class\_nodeapplication.PNG) Y**

1. **Các class hoạt động trên board STM**

* **SerialTransport**
  + **Hình ảnh (DevicePhotos/class\_serialtransport.PNG) Y**
* **Actuator**
  + **Hình ảnh (DevicePhotos/class\_actuator.PNG) Y**
* **Application**
  + **Hình ảnh (DevicePhotos/class\_stmapplication.PNG) Y**

1. **Hiện thực**
   1. **Mạch chức năng đo ppm.**
      1. **Lí thuyết**
2. **Độ dẫn điện (EC)**

* **Chỉ số EC (electro-conductivity) (tương đối) là chỉ số diễn tả tổng nồng độ ion hòa tan trong dung dịch. Độ dẫn điện có thể được thể hiện bằng một số đơn vị khác nhau, nhưng đơn vị tiêu biểu được dùng để đo lường EC là millisiemens trên centimet (mS / cm). Chỉ số EC không diễn tả nồng độ của từng chất trong dung dịch đồng thời cũng không thể hiện mức độ cân bằng của các chất dinh dưỡng trong dung dịch. EC là thước đo độ dẫn điện từ hai đầu dò 1cm.**
* **Trong suốt quá trình tăng trưởng, cây hấp thu khoáng chất mà chúng cần, do vậy duy trì EC ở một mức ổn định là rất quan trọng. Nếu dung dịch có chỉ số EC cao thì sự hấp thu nước của cây diễn ra nhanh hơn sự hấp thu khoáng chất. Điều này làm nồng độ dung dịch tăng cao và gây ngộ độc cho cây. Khi đó ta phải bổ sung thêm nước vào môi trường. Ngược lại, nếu EC thấp, cây sẽ hấp thu khoáng chất nhanh hơn hấp thu nước. Khi đó, nồng độ dung dịch giảm mạnh, cây sẽ không được cung cấp đầy đủ khoáng chất, chậm lớn và phát triển kém.**

1. **Chỉ số TDS**

* **Chỉ số TDS (Total Dissolved Solids) là chỉ số đo tổng lượng chất rắn hoà tan, tổng số các ion mang điện tích bao gồm khoáng chất, muối hoặc kim loại tồn tại trong một khối lượng nước nhất định. TDS thường được biểu thị bằng hàm số ml/L hoặc ppm (Parts Per Million). 1 ppm tương ứng với 1mg chất rắn hòa tan trong một lít nước. Hầu hết nước máy sẽ có chỉ số PPM rơi vào khoảng từ 200 – 400ppm.**
* **Chỉ số TDS cũng ảnh hưởng lớn đến sự phát triển của cây: Nếu TDS lên quá cao, nồng độ dung dịch vượt mức cho phép sẽ gây ra hiện tượng ngộ độc cho cây. Ngược lại, khi chỉ số TDS xuống thấp, dung dịch thủy canh sẽ không đảm bảo cung cấp đủ chất dinh dưỡng cho cây trồng.**

1. **Tương quan giữa EC và TDS**

* **Mặc dù có một mối tuơng quan giữa EC và TDS nhưng chúng không giống nhau. TDS và EC là 2 tham số riêng biệt. TDS là tổng lượng chất rắn hoà tan trong nước. EC là khả năng của các chất co thể gây ra dòng điện. Lượng chất rắn như muối trong phân bón tỉ lệ trực tiếp với độ dẫn điện của nó, vì vậy lượng chất rắn cao gây độ dẫn cao. Vì khi phân bón hoà tan trong nước chúng trở thành các "ion", có mang điện tích âm hoặc dương, nên chúng sinh ra dòng điện.**
* **Mối quan hệ của TDS và độ dẫn đặc hiệu của nước ngầm có thể được ước lượng bằng phương trình sau:**

**TDS = ke\*EC**

* + **Trong đó: TDS có đơn vị mg / L và EC là độ dẫn điện ở microsiemens trên mỗi centimet ở 25 °C. Yếu tố tương quan ke dao động từ 0,55 đến 0,8.**
* **Giới hạn với cây trồng:**
  + **Hình ảnh (DevicePhotos/PPM/thamkhao.PNG)**

1. **Tại sao lại dùng thiết bị này ?**

* **Nếu EC / PPM chỉ là đo điện dẫn (hoặc kháng) thì tại sao không sử dụng một đồng hồ volt / ohm trực tiếp? Bởi chúng đi qua dòng điện DC thông qua các đầu dò và bạn không thể đo độ dẫn của muối với dòng điện DC vì nó sẽ tách các phân tử ra ngoài, và vì các phân tử là điện dẫn điện,các phần tử điện sẽ thay đổi liên tục và sẽ không thu được kết quả gì. Bằng cách sử dụng một tín hiệu AC,với tần số đủ cao (> 1khz) các phân tử không có thời gian để di chuyển ra ngoài trước khi chúng được kéo theo hướng ngược lại.**
* **Các thành phần ion:**
  + **Hình ảnh (DevicePhotos/PPM/ion.png)**
* **Trường hợp sử dụng dòng AC**
  + **Hình ảnh (DevicePhotos/PPM/AC.png)**
* **Trường hợp sử dụng dòng DC**
  + **Hình ảnh (DevicePhotos/PPM/DC.png)**

1. **Hoạt động:**

* **Hai điện cực với một điện áp xoay chiều được đặt trong dung dịch. Điều này tạo ra một dòng điện phù thuộc vào bản chất dẫn điện của dung dịch. Thiết bị đọc dòng diện này và hiển thị theo đơn vị EC hoặc ppm.**
  + 1. **Linh kiện, thiết kế**

1. **Schematic**
   * **Hình ảnh (DevicePhotos/PPM/ppmschematic.png)**

* **Thành phần chính trong mạch:**
  + **Oscillator cầu Wein: là mạch điện tử tạo ra tín hiệu điện tử dao động, thường là một sóng sin hay sóng vuông. Bộ tạo dao động chuyển đổi dòng điện một chiều (DC) từ nguồn điện sang tín hiệu dòng xoay chiều (AC). Ưu điểm của mạch là chỉ có một vài thành phần và sự ổn định tần số tốt. Phần nhược điểm của mạch là biên độ đầu ra biến dạng cao gây khó khăn trong việc thu thập. Có một vài cách để giảm thiểu tác động này.** 
    - **Hình ảnh (DevicePhotos/PPM/wein.png)**
  + **Các giá trị được cung cấp để tạo ra mạch dao động có tần số là 10Khz**
  + **Ta có:**

**R = 1k Ohms, C = 0.015uF, suy ra F = 1/(2\*pi\*R\*C)**

* + **Chúng được sử dụng rộng rãi trong nhiều thiết bị điện tử. Ví dụ phổ biến của tín hiệu được tạo ra bởi dao động bao gồm các tín hiệu phát sóng của đài phát thanh và truyền hình , tín hiệu đồng hồ mà điều chỉnh máy tính và đồng hồ thạch anh ,âm thanh được tạo ra bởi beepers điện tử và trò chơi điện tử... Ta sử dụng một tín hiệu AC để đo độ dẫn của muối, có thể thay đổi giá trị R, C trong oscillator để đảm bảo rằng tần số đầu ra là trên 1kHz, tần số thấp hơn sẽ cung cấp các giá trị không ổn định.**
  + **Gain loop: mạch khuếch đại nguồn tín hiệu nhận được từ đầu dò 1cm đặt trong dung dịch**
  + **Bộ chuyển đổi AC thành DC: Một mạch chỉnh lưu, một mạch điện bao gồm các linh kiện điện - điện tử dùng để biến đổi dòng điện xoay chiều thành dòng điện một chiều. Mạch chỉnh lưu cả chu kỳ thường dùng 4 Diode mắc theo hình cầu (còn gọi là mạch chỉnh lưu cầu). Mạch chỉnh lưu có thể được sử dụng trong các bộ nguồn cung cấp dòng điện một chiều, hoặc trong các mạch tách sóng tín hiệu vô tuyến điện trong các thiết bị vô tuyến, sử dụng mạch chỉnh lưu nhiều điốt (4 điốt) để có thể biến đổi từ xoay chiều thành một chiều.**
    - **Hình ảnh (DevicePhotos/PPM/4diode.png)**
  + **Nửa chu kì đầu, khi điểm 1 dương so với điểm 2, dòng chạy từ 1 qua D3 qua tải R1 qua D1 về đầu âm**
  + **Nửa chu kì sau, khi điểm 2 dương so với điểm 1, dòng chạy từ 2 qua D4 qua tải R1 qua D2 về đầu âm.**
  + **Cả chu kì đều có dòng điện chạy qua tải.**

1. **Linh kiện**

* **Nguồn đôi 12V: Mạch tạo nguồn đôi 12V có chức năng đổi từ nguồn đơn DC sang nguồn đôi +-12V. Với điện áp ngỏ vào từ 2.8V-5VDC.**
  + **Hình ảnh (DevicePhotos/PPM/12vtruoc.png)**
  + **Hình ảnh (DevicePhotos/PPM/12vsau.png)**
* **Jack DC: kết nối với nguồn DC để cấp nguồn cho mạch**
  + **Hình ảnh (DevicePhotos/PPM/jackdc.jpg)**
* **IC TL074: gồm 4 opamp trong 1 IC, dùng chung 1 nguồn đôi** 
  + **Hình ảnh (DevicePhotos/PPM/tl074.jpg)**
  + **Hình ảnh (DevicePhotos/PPM/tl074\_pin.jpg)**
* **Diode Zener: diode ổn áp, nó được chế tạo sao cho khi phân cực ngược thì diode Zener sẽ ghim một mức điện áp gần cố định bằng giá trị ghi trên diode, làm ổn áp cho mạch điện**
  + **Hình ảnh (DevicePhotos/PPM/zener.jpg)**
* **Tụ điện:**
  + **0.015uF Hình ảnh (DevicePhotos/PPM/c015.jpg)**
  + **0.22uF Hình ảnh (DevicePhotos/PPM/c22.jpg)**
* **Cầu Diode: gồm 4 diode đơn mắc thành cầu chỉnh lưu được đóng gói trong một vỏ duy nhất, dùng để chỉnh lưu dòng điện xoay chiều thành một chiều.**
  + **Điện áp tối đa : 600V**
  + **Dòng điện định mức : 4A**
  + **Nhiệt độ hoạt động : -55 oC đến 150oC**
  + **Hình ảnh (DevicePhotos/PPM/caudiode.jpg)**
  1. Sử dụng, thu thập dữ liệu sensor
     1. Nhiệt độ, độ ẩm
* Nối dây:

|  |  |
| --- | --- |
| DHT | Node MCU |
| VCC | 3.3V |
| OUT | D3 |
| GND | GND |

Table. Nối chân DHT-NodeMCU

* Thư viện:
  + Hình ảnh (DevicePhotos/lib\_DHT.PNG) Y
  + Đường dẫn:
    - Hình ảnh (DevicePhotos/path\_DHT.PNG) Y
* Kết quả đo:
  + Hình ảnh (DevicePhotos/result\_DHT.PNG) Y
    1. Ánh sáng
* Nối dây:

|  |  |
| --- | --- |
| BH1750 | Node MCU |
| VCC | 3.3V |
| GND | GND |
| SCL | D1 |
| SDA | D2 |
| ADDR | D0 |

Table. Nối chân BH1750 – Node MCU

* Thư viện:
  + Hình ảnh (DevicePhotos/lib\_light.PNG) Y
  + Đường dẫn:
    - Hình ảnh (DevicePhotos/path\_light.PNG) Y
* Kết quả đo:
  + Hình ảnh (DevicePhotos/result\_light.PNG) Y
    1. PPM
* Nối dây:

|  |  |
| --- | --- |
| PPM | Node MCU |
| Analog Out | A0 |

Table. Nối chân PPM – Node MCU

* Kết quả đo:
  + Hình ảnh (DevicePhotos/result\_ppm.PNG) N
  1. Thời gian thực
* Nối dây:

|  |  |
| --- | --- |
| DS1307 | SMT32F103C8T6 |
| GND | GND |
| VCC | 5V |
| SDA | PB\_9 |
| SCL | PB\_8 |

Table. Nối chân DS1307 – STM32F103C8T6

* Thư viện:
  + Hình ảnh (DevicePhotos/lib\_realtime.PNG) Y
  + Đường dẫn
    - Hình ảnh (DevicePhotos/path\_realtime.PNG) Y
* Kết quả đo:
  + Hình ảnh (DevicePhotos/result\_realtime.PNG) N
  1. Điều khiển các actuator
     1. Thanh ghi dịch
* Nối dây:

|  |  |
| --- | --- |
| 74HC595 | STM32F103C8T6 |
| DATA | PA\_7 |
| LATCH | PA\_6 |
| CLOCK | PA\_5 |

Table. Nối chân 74HC595 – STM32F103C8T6

* Thư viện:
  + Hình ảnh (DevicePhotos/lib\_register.PNG) Y
    - Hình ảnh (DevicePhotos/path\_register.PNG) Y
    1. Nguồn cho các actuators.
* Các actuators sử dụng nguồn 12 VDC lấy từ adapter chuyển đổi AC 220V sang DC 12V.
  + Hình ảnh (DevicePhotos/adapter.jpg) Y
  1. Giao tiếp với web-server.
     1. Giao thức tầng dưới – thư viện MQTT
* Như đã đề cập giao thức truyền nhận giữa web-server là MQTT, cả thiết bị lần web-server đều đóng vai trò là một MQTT Client:
  + Thiết bị gửi tới broker (publish) các gói tin vào các kênh (channel) đã được cấu hình sẵn và thống nhất giữa thiết bị - webserver.
  + Thiết bị nhận các gói tin từ broker qua các kênh đã đăng kí (subcribe), khi webserver gửi gói tin vào các kênh đó.
* Thư viện:
  + Hình ảnh (DevicePhotos/lib\_mqtt.PNG) Y
  + Đường dẫn:
    - Hình ảnh (DevicePhotos/path\_mqtt.PNG) Y
    1. Giao thức truyền nhận lớp trên
* Gói tin trên đường truyền có định dạng là (**string**), ví dụ:

|  |  |  |
| --- | --- | --- |
| STT | Gói tin thật | Gói tin khi gửi (nhận) |
| 1 | [1][2][3] | “123” |
| 2 | [0x0A][0x0B] | “0A0B” |
| 3 | “MXYZ” | Không hợp lệ |

Table. Các trường hợp giá trị dữ liệu.

* + Ở dự án này toàn bộ dữ liệu có giá trị (value) nằm trong 3 trường hợp:
    - TH1: 0<= value <= 9.
    - TH2: ‘A’ <= value <= ‘F’, dùng để gửi địa chỉ MAC.
    - TH3: các trường hợp còn lại không hợp lệ.
* Các cấu trúc dữ liệu và phương thức cho giao thức được định nghĩa ở:
  + Hình ảnh (DevicePhotos/lib\_datastructure.PNG) Y
  + Đường dẫn:
    - Hình ảnh (DevicePhotos/path\_datastructure.PNG) Y
  1. Webserver
* Webserver hoạt động trên board NodeMCU, ở chế độ Access Point.
* Board chuyển từ chế độ Station sang Access Point khi nhận tín hiệu nhấn liên tục nút nhấn trong vòng 3s.
  + Hình ảnh (DevicePhotos/node\_button.PNG) Y
* Giao diện và chức năng:
  + Khi chuyển sang chế độ Access Point và hoạt động chức năng webserver, Node MCU sẽ phát ra wifi
    - Hình ảnh (DevicePhotos/webserver\_1.PNG) Y
  + Sau khi kết nối vào Access Point của Node MCU, mở trình duyệt, truy cập vào địa chỉ (**192.168.4.1**) ta sẽ kết nối với webserver.
    - Hình ảnh (DevicePhotos/webserver\_2.PNG) Y
  + Sau khi chọn (**wifi name**) và nhập (**password**), trình duyệt sẽ gửi 2 giá trị này về board và được lưu lại trong bộ nhớ eeprom.
    - Hình ảnh (DevicePhotos/webserver\_4.PNG) Y
    - Hình ảnh (DevicePhotos/webserver\_3.PNG) Y
  + Ngoài ra webserver còn cung cấp chức năng cấu hình các thông tin của giao thức MQTT, truy cập vào (192.168.4.1/config):
    - Hình ảnh (DevicePhotos/webserver\_5.PNG) Y
  1. Đèn tín hiệu.
* Đèn tín hiệu sử dụng led đơn, tiêu thụ dòng nhỏ.
* Hoạt động của đèn tín hiệu có ba trạng thái chính:
  + Tắt: 0V
  + Sáng: 3.3V.
  + Nháy: kết hợp timer.
* Các chức năng bao gồm:
  + Đèn báo tín hiệu wifi:
    - Kết nối: đèn tắt.
    - Mất kết nối: đèn nháy.
    - Mất kết nối trong hơn 30s đèn sáng.
  + Đèn báo tín hiệu MQTT:
    - Kết nối vơi broker : đèn tắt.
    - Mất kết nối: đèn sáng.
  + Đèn báo tín hiệu của các actuator.
    - Actuator chưa được kích hoạt: đèn tắt.
    - Actuator được kích hoạt: đèn sáng.
    - Actuator đang ở trong thời gian chạy thời gian biểu: đèn nháy.
  1. Thiết kế mạch.
* Schematic
  + Hình ảnh (DevicePhotos/schematic.PNG) Y
* PCB
  + Hình ảnh (DevicePhotos/pcb.PNG) N
  1. Mô hình thử nghiệm.
* Hình ảnh...

1. Kết quả và kiểm thử
   1. Thiết bị.
   2. Hoạt động của các chức năng.
      1. Mạch đo chức năng PPM.
      2. Thu thập dữ liệu cảm biến

|  |
| --- |
| Đọc dữ liệu nhiệt độ, độ ẩm |
| Đọc dữ liệu ánh sáng |
| Đọc dữ liệu ppm |
| Định thời |

* + 1. Điều khiển các actuators.

|  |
| --- |
| Đọc thời gian thực |
| Hoạt động theo schedule |
| Lưu trữ trạng thái các actuator vào EEPROM |
| Tải lại trạng thái các actuator từ EEPROM |
| Đèn báo tín hiệu các actuator |

* + 1. Giao tiếp với webserver.

|  |
| --- |
| Protocol hoạt động đúng |
| Protocol Mqtt truyền nhận đúng |

* + 1. Cấu hình wifi, mqtt.

|  |
| --- |
| Chuyển chế độ từ Station sang Access Point |
| Webserver hoạt động ổn định. |
| Lưu trữ các thông số sau khi cấu hình vào EEPROM |
| Tải lại các thông số sau khi reset. |