A white rectangular frame with black border

Description automatically generated**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**

**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT**

**VÀ TRÍ TUỆ NHÂN TẠO**

**---oOo---**

****

**TIỂU LUẬN KẾT THÚC MÔN HỌC**

**CÔNG NGHỆ IOT**

**Tên đề tài: Đề 2**

**OPENREMOTE ĐO NHIỆT ĐỘ VÀ ĐỘ ẨM**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | **Lê Duy Hùng** |
| Nhóm sinh viên thực hiện: | **Bùi Hữu Phước\_22050020**  **Ngô Lê Thành Hải\_22050007** |
|  |  |

**Bình Dương 2024**

**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**

**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT**

**VÀ TRÍ TUỆ NHÂN TẠO**

**---oOo---**

****

**TIỂU LUẬN KẾT THÚC MÔN HỌC**

**CÔNG NGHỆ IOT**

**Tên đề tài: Đề 2**

**OPENREMOTE ĐO NHIỆT ĐỘ VÀ ĐỘ ẨM**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | **Lê Duy Hùng** |
| Nhóm sinh viên thực hiện: | **Bùi Hữu Phước\_22050020**  **Ngô Lê Thành Hải\_22050007** |

**Bình Dương 2024**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

**MỤC LỤC**

[LỜI CẢM ƠN v](#_Toc175639841)

[LỜI MỞ ĐẦU vi](#_Toc175639842)

[I.GIỚI THIỆU PLATFORM: OPENREMOTE 1](#_Toc175639843)

[1 Mục đích 1](#_Toc175639844)

[1.1 Các tính năng chính 1](#_Toc175639845)

[1.2 Ưu điểm 2](#_Toc175639846)

[2.PHÂN TÍCH KIẾN TRÚC CỦA HỆ THỐNG TRIỂN KHAI 3](#_Toc175639847)

[2.1. Phần cứng 3](#_Toc175639848)

[2.1.1. Board điều khiển: ESP8266 3](#_Toc175639849)

[2.1.2. Cảm biến DHT11 3](#_Toc175639850)

[2.2. Phần mềm 3](#_Toc175639851)

[2.2.1. Cài đặt và cấu hình OpenRemote 3](#_Toc175639852)

[2.2.2. Giao thức kết nối: HTTP hoặc MQTT 3](#_Toc175639853)

[2.3. Kiến trúc hệ thống 4](#_Toc175639854)

[2.3.1. Sơ đồ kiến trúc 4](#_Toc175639855)

[3 QUY TRÌNH HOẠT ĐỘNG 5](#_Toc175639856)

[3.1. Quy trình Kết nối 5](#_Toc175639857)

[3.2 Sơ đồ kết nối 7](#_Toc175639858)

[4 CÁC BƯỚC KẾT NỐI THIẾT BỊ VÀO HỆ THỐNG 8](#_Toc175639859)

[4.1. Cài đặt Phần mềm 8](#_Toc175639860)

[4.1.1. Cài đặt Docker 8](#_Toc175639861)

[4.1.2. Cài đặt OpenRemote trên Docker 8](#_Toc175639862)

[4.2. Kết nối Phần cứng 9](#_Toc175639863)

[4.2.1. Kết nối Cảm biến DHT11 với Board Điều khiển (ESP8266) 9](#_Toc175639864)

[4.2.2. Cấu hình Board Điều khiển để Truyền Dữ liệu qua HTTP/MQTT 10](#_Toc175639865)

[4.3. Tích hợp vào Hệ thống 10](#_Toc175639866)

[5 SO SÁNH HỆ THỐNG VỚI MỘT PLATFORM KHÁC 12](#_Toc175639867)

[5.1. So sánh với Platform Triển khai Docker 12](#_Toc175639868)

[5.1.1. Triển khai Docker 12](#_Toc175639869)

[5.1.2. Triển khai Serviceless 13](#_Toc175639870)

[5.1.3. Sử dụng Trực tiếp trên Phần Cứng 13](#_Toc175639871)

[6 KẾT LUẬN 15](#_Toc175639872)

[6.1. Tóm tắt Các Điểm Mạnh của Hệ thống Sử dụng OpenRemote 15](#_Toc175639873)

[6.2. Đánh Giá về Tiềm Năng Ứng Dụng Thực Tế và Cải Tiến Trong Tương Lai 15](#_Toc175639874)

[7 CODE 16](#_Toc175639875)

[TÀI LIỆU THAM KHẢO 25](#_Toc175639876)

# LỜI CẢM ƠN

Với lòng biết ơn sâu sắc và tình cảm chân thành nhất, nhóm em xin bày tỏ lòng biết ơn đến thầy, Lê Duy Hùng, người đã tận tình hướng dẫn và hỗ trợ nhóm em trong suốt quá trình nghiên cứu và hoàn thiện đề tài về Công nghệ IoT này.

Những kiến thức quý báu và sự giúp đỡ của thầy đã giúp nhóm em có thể hoàn thành đề tài tiểu luận một cách tốt nhất. Nhóm em xin chân thành cảm ơn thầy vì đã dành thời gian và tâm huyết để giúp nhóm em phát triển ý tưởng và giải quyết các khó khăn gặp phải trong quá trình nghiên cứu.

Trong quá trình thực hiện đề tài, nhóm em đã nỗ lực hết sức mình, nhưng không thể tránh khỏi những thiếu sót và hạn chế. Vì vậy, nhóm em rất mong nhận được sự góp ý và đánh giá công tâm từ thầy để có thể hoàn thiện và nâng cao chất lượng của đề tài này.

# LỜI MỞ ĐẦU

Trong thời đại công nghiệp 4.0, Công nghệ Internet vạn vật (IoT) đã trở thành một xu hướng không thể thiếu, đem lại những đột phá trong việc giám sát, quản lý và điều khiển từ xa các hệ thống và thiết bị. Một trong những ứng dụng quan trọng của IoT là việc đo lường và giám sát các thông số môi trường, như nhiệt độ và độ ẩm, giúp các doanh nghiệp và cá nhân tối ưu hóa quy trình sản xuất, bảo quản, và đảm bảo chất lượng sản phẩm.

Bài tiểu luận này sẽ tập trung phân tích và khám phá việc ứng dụng nền tảng OpenRemote – một hệ thống mã nguồn mở mạnh mẽ – trong việc đo nhiệt độ và độ ẩm từ xa. Thông qua việc triển khai các cảm biến IoT và tích hợp dữ liệu vào hệ thống OpenRemote, chúng ta sẽ thấy được cách công nghệ này có thể cải thiện hiệu quả quản lý môi trường và đưa ra các quyết định dựa trên dữ liệu thời gian thực.

Mục tiêu của bài viết là không chỉ trình bày những lợi ích và khả năng của OpenRemote trong việc giám sát nhiệt độ và độ ẩm, mà còn đánh giá những thách thức và tiềm năng phát triển của công nghệ này trong tương lai. Từ đó, bài viết hy vọng sẽ góp phần mở ra những hướng đi mới cho việc ứng dụng IoT trong các lĩnh vực quản lý môi trường và điều khiển từ xa.

**I.GIỚI THIỆU PLATFORM: OPENREMOTE**

## 1 Mục đích

OpenRemote là một nền tảng mã nguồn mở được thiết kế để quản lý và giám sát các hệ thống IoT (Internet of Things). Mục tiêu của OpenRemote là cung cấp một công cụ linh hoạt và mạnh mẽ, cho phép người dùng dễ dàng kết nối, quản lý, và điều khiển các thiết bị IoT từ xa. Nền tảng này hỗ trợ việc tích hợp và tương tác giữa các thiết bị thông minh, hệ thống cảm biến, và cơ sở hạ tầng dữ liệu, tạo ra một hệ sinh thái IoT liền mạch.

## 1.1 Các tính năng chính

* Kết nối với thiết bị IoT: OpenRemote hỗ trợ kết nối với nhiều loại thiết bị và cảm biến khác nhau, bao gồm cảm biến nhiệt độ, độ ẩm, ánh sáng, và nhiều loại cảm biến môi trường khác. Người dùng có thể dễ dàng cấu hình để các thiết bị này gửi dữ liệu về hệ thống.
* Quản lý dữ liệu: OpenRemote cho phép thu thập, lưu trữ và quản lý dữ liệu từ các thiết bị IoT. Dữ liệu này có thể được xử lý, phân tích để đưa ra những thông tin hữu ích cho người dùng, như biểu đồ, báo cáo, hoặc cảnh báo.
* Hiển thị dữ liệu qua Dashboard: Nền tảng cung cấp một giao diện người dùng trực quan, nơi dữ liệu từ các thiết bị được hiển thị thông qua các biểu đồ, bảng, và các widget khác. Dashboard của OpenRemote có thể tùy chỉnh để phù hợp với nhu cầu của người dùng, giúp họ dễ dàng theo dõi và quản lý các chỉ số quan trọng từ xa.
* Tự động hóa và điều khiển: OpenRemote cho phép thiết lập các kịch bản tự động hóa dựa trên các điều kiện cụ thể, chẳng hạn như điều khiển thiết bị dựa trên dữ liệu từ cảm biến, hoặc gửi cảnh báo khi dữ liệu vượt quá ngưỡng cho phép.
* Khả năng mở rộng: OpenRemote được xây dựng với khả năng mở rộng cao, cho phép tích hợp thêm nhiều loại cảm biến, giao thức kết nối và ứng dụng bên thứ ba khi cần thiết. Điều này làm cho nền tảng phù hợp với nhiều quy mô hệ thống, từ những ứng dụng đơn lẻ đến các dự án IoT phức tạp.

## 1.2 Ưu điểm

* Tính linh hoạt: OpenRemote có khả năng tùy biến cao, cho phép người dùng dễ dàng điều chỉnh và mở rộng hệ thống theo nhu cầu thực tế. Người dùng có thể thêm, bớt hoặc thay đổi các thành phần trong hệ thống một cách dễ dàng.
* Khả năng mở rộng: Nền tảng này có thể mở rộng theo quy mô, từ các dự án nhỏ lẻ đến các hệ thống IoT phức tạp với hàng ngàn thiết bị. Điều này giúp OpenRemote phù hợp với cả cá nhân, tổ chức nhỏ, và các doanh nghiệp lớn.
* Tích hợp dễ dàng: OpenRemote hỗ trợ nhiều giao thức kết nối và có khả năng tương thích với nhiều loại cảm biến và thiết bị khác nhau. Người dùng có thể tích hợp các thiết bị hiện có hoặc thêm các thiết bị mới mà không gặp nhiều khó khăn.
* Hỗ trợ cộng đồng: Là một nền tảng mã nguồn mở, OpenRemote có một cộng đồng người dùng và phát triển rộng lớn, cung cấp tài liệu, hỗ trợ và các tài nguyên hữu ích khác cho người mới bắt đầu cũng như các chuyên gia.
* Bảo mật: OpenRemote cung cấp các công cụ bảo mật giúp bảo vệ dữ liệu và đảm bảo an toàn cho hệ thống, từ việc mã hóa dữ liệu đến việc quản lý quyền truy cập cho người dùng.

**2.PHÂN TÍCH KIẾN TRÚC CỦA HỆ THỐNG TRIỂN KHAI**

**2.1. Phần cứng**

**2.1.1. Board điều khiển: ESP8266**

**ESP8266:** Đây là các board điều khiển vi xử lý phổ biến với khả năng kết nối Wi-Fi, phù hợp cho các ứng dụng IoT. ESP8266 được ưa chuộng nhờ chi phí thấp, tiêu thụ năng lượng ít, và hỗ trợ các giao thức truyền thông như HTTP và MQTT.

**2.1.2. Cảm biến DHT11**

**DHT11:** Đây là cảm biến đo nhiệt độ và độ ẩm phổ biến, dễ dàng tích hợp với các board điều khiển như ESP32, ESP8266, hoặc Raspberry Pi. Cảm biến DHT11 có ưu điểm là giá rẻ, kích thước nhỏ gọn, và dễ dàng kết nối với các board điều khiển thông qua giao diện kỹ thuật số. Tuy nhiên, độ chính xác của DHT11 có thể thấp hơn so với các cảm biến khác như DHT22.

**2.2. Phần mềm**

**2.2.1. Cài đặt và cấu hình OpenRemote**

* **Cài đặt OpenRemote:** OpenRemote có thể được cài đặt trên nhiều nền tảng, bao gồm máy tính cá nhân, máy chủ ảo, hoặc dưới dạng Docker container. Việc sử dụng Docker giúp đơn giản hóa quá trình cài đặt và triển khai hệ thống, đồng thời đảm bảo tính linh hoạt và khả năng mở rộng.
* **Cấu hình OpenRemote:** Sau khi cài đặt, cần cấu hình OpenRemote để kết nối với các thiết bị IoT và cảm biến. Cấu hình này bao gồm việc thiết lập các endpoint cho các cảm biến, thiết lập các giao thức truyền thông (HTTP hoặc MQTT), và cấu hình giao diện dashboard để hiển thị dữ liệu.

**2.2.2. Giao thức kết nối: HTTP hoặc MQTT**

* **HTTP:** Đây là giao thức truyền tải phổ biến và dễ sử dụng. HTTP thường được sử dụng cho các ứng dụng web và API. Trong hệ thống này, HTTP có thể được sử dụng để gửi dữ liệu từ cảm biến đến server OpenRemote. Tuy nhiên, HTTP không phải là giao thức tối ưu cho các ứng dụng IoT do không có khả năng duy trì kết nối liên tục và tiêu thụ nhiều băng thông.
* **MQTT:** Đây là giao thức nhẹ và tối ưu cho các ứng dụng IoT, đặc biệt là trong các hệ thống có yêu cầu về hiệu năng và băng thông. MQTT hoạt động dựa trên mô hình publish-subscribe, giúp giảm thiểu lưu lượng dữ liệu và cho phép cập nhật dữ liệu gần như tức thời từ các cảm biến đến hệ thống OpenRemote. MQTT cũng có khả năng duy trì kết nối liên tục và đáng tin cậy trong môi trường mạng không ổn định.

**2.3. Kiến trúc hệ thống**

**2.3.1. Sơ đồ kiến trúc**

**Sơ đồ kiến trúc của hệ thống gồm các thành phần chính:**

1. **Thiết bị cảm biến (DHT11):** Đo nhiệt độ và độ ẩm, sau đó gửi dữ liệu này đến board điều khiển.
2. **Board điều khiển (ESP8266):** Nhận dữ liệu từ cảm biến DHT11 và xử lý trước khi truyền dữ liệu qua giao thức HTTP hoặc MQTT đến hệ thống OpenRemote.
3. **OpenRemote Server:** Hệ thống quản lý trung tâm nhận dữ liệu từ các board điều khiển thông qua HTTP hoặc MQTT, lưu trữ và phân tích dữ liệu. Hệ thống này có thể triển khai trên Docker để dễ dàng quản lý và mở rộng.
4. **Dashboard:** Giao diện người dùng của OpenRemote, nơi dữ liệu nhiệt độ và độ ẩm được hiển thị dưới dạng biểu đồ, bảng số liệu, hoặc các widget tùy chỉnh. Người dùng có thể truy cập dashboard này từ bất kỳ thiết bị nào có kết nối internet để theo dõi và giám sát các thông số môi trường.

**3 QUY TRÌNH HOẠT ĐỘNG**

**3.1. Quy trình Kết nối**

**Bước 1: Đo lường dữ liệu từ cảm biến DHT11**

* **Mô tả:** Cảm biến DHT11 là một thiết bị phổ biến dùng để đo nhiệt độ và độ ẩm trong môi trường. Cảm biến này liên tục thu thập dữ liệu về nhiệt độ và độ ẩm từ môi trường xung quanh.
* **Hoạt động:**
  + Khi có sự thay đổi về nhiệt độ hoặc độ ẩm trong môi trường, cảm biến DHT11 sẽ cập nhật giá trị mới.
  + Dữ liệu từ cảm biến bao gồm nhiệt độ (°C) và độ ẩm (%), thường được đọc dưới dạng tín hiệu số.

**Bước 2: Truyền dữ liệu đến board điều khiển**

* **Mô tả:** Board điều khiển (ví dụ: ESP8266 hoặc ESP32) là thiết bị chịu trách nhiệm thu thập dữ liệu từ cảm biến DHT11. Nó được lập trình để đọc dữ liệu từ cảm biến và xử lý sơ bộ trước khi truyền đi.
* **Hoạt động:**
  + Board điều khiển được kết nối với cảm biến DHT11 thông qua các chân GPIO.
  + Một đoạn mã (thường viết bằng ngôn ngữ lập trình như C++ hoặc MicroPython) sẽ liên tục đọc dữ liệu từ cảm biến theo một chu kỳ nhất định.
  + Sau khi dữ liệu được đọc, board điều khiển có thể thực hiện một số xử lý cơ bản, như lọc dữ liệu hoặc tính toán giá trị trung bình nếu cần thiết.

**Bước 3: Truyền dữ liệu đến OpenRemote thông qua HTTP/MQTT**

* **Mô tả:** Sau khi xử lý dữ liệu, board điều khiển sẽ gửi dữ liệu này đến hệ thống OpenRemote bằng cách sử dụng giao thức truyền thông như HTTP hoặc MQTT.
* **Hoạt động:**
  + **Giao thức HTTP:** Board điều khiển gửi yêu cầu POST chứa dữ liệu (JSON/XML) về nhiệt độ và độ ẩm đến máy chủ OpenRemote.
  + **Giao thức MQTT:** Board điều khiển xuất bản dữ liệu đến một topic cụ thể trên broker MQTT, nơi OpenRemote đã đăng ký lắng nghe và nhận dữ liệu.
  + Cả hai giao thức đều đảm bảo dữ liệu được truyền một cách an toàn và chính xác từ thiết bị đến hệ thống trung tâm.

**Bước 4: Nhận và lưu trữ dữ liệu tại OpenRemote**

* **Mô tả:** Khi OpenRemote nhận được dữ liệu từ board điều khiển, hệ thống sẽ tiến hành lưu trữ và xử lý dữ liệu để chuẩn bị cho việc hiển thị trên dashboard.
* **Hoạt động:**
  + Dữ liệu được lưu trữ trong cơ sở dữ liệu của OpenRemote, có thể là SQL, NoSQL hoặc các dịch vụ lưu trữ đám mây tùy thuộc vào cấu hình hệ thống.
  + Hệ thống có thể thực hiện các thao tác như làm sạch dữ liệu, tính toán các chỉ số bổ sung (ví dụ: chỉ số độ ẩm tương đối), và chuẩn bị dữ liệu cho hiển thị trực quan.

**Bước 5: Hiển thị dữ liệu trên Dashboard**

* **Mô tả:** Dữ liệu nhiệt độ và độ ẩm được hiển thị theo thời gian thực trên dashboard của OpenRemote, nơi người dùng có thể dễ dàng quan sát và phân tích.
* **Hoạt động:**
  + Dashboard của OpenRemote cung cấp các biểu đồ, bảng dữ liệu, và các widget tương tác để hiển thị thông tin thu thập từ cảm biến.
  + Người dùng có thể tùy chỉnh giao diện để hiển thị dữ liệu theo ý muốn, chẳng hạn như biểu đồ đường, biểu đồ cột, hoặc biểu đồ nhiệt.

### 3.2 Sơ đồ kết nối

A circuit board with wires and a black square

Description automatically generated

# 4 CÁC BƯỚC KẾT NỐI THIẾT BỊ VÀO HỆ THỐNG

## 4.1. Cài đặt Phần mềm

### 4.1.1. Cài đặt Docker

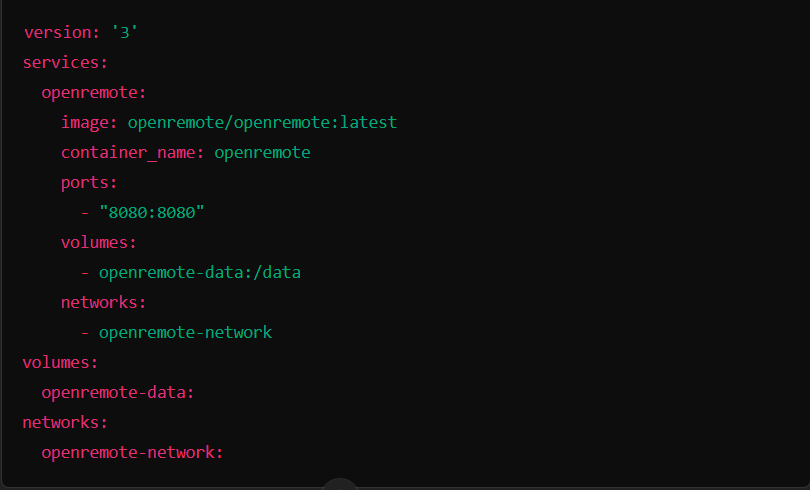
* **Mô tả:** Docker là một nền tảng giúp triển khai ứng dụng trong các container, đảm bảo tính nhất quán và dễ dàng quản lý. Để cài đặt Docker, cần tải và cài đặt Docker Desktop từ trang web chính thức của Docker.
* **Hướng dẫn cài đặt:**
  1. **Tải Docker Desktop:** Truy cập trang Docker Desktop và tải phiên bản phù hợp với hệ điều hành.
  2. **Cài đặt Docker Desktop:** Mở tệp cài đặt và làm theo hướng dẫn để hoàn tất quá trình cài đặt.
  3. **Khởi động Docker Desktop:** Sau khi cài đặt, mở Docker Desktop và đảm bảo rằng Docker Engine đang chạy.

### 4.1.2. Cài đặt OpenRemote trên Docker

* **Mô tả:** OpenRemote có thể được triển khai trên Docker để dễ dàng quản lý và bảo trì. Sử dụng các tệp Docker Compose để cấu hình và triển khai OpenRemote.
* **Hướng dẫn cài đặt:**
  1. **Tạo thư mục dự án:** Tạo một thư mục để chứa tệp cấu hình Docker Compose.



* 1. **Tạo tệp Docker Compose:** Tạo một tệp có tên docker-compose.yml với nội dung cấu hình cho OpenRemote.



* 1. **Khởi động OpenRemote:** Chạy lệnh Docker Compose để bắt đầu triển khai OpenRemote.



* 1. **Kiểm tra trạng thái:** Truy cập http://localhost:80 để đảm bảo rằng OpenRemote đang chạy.

## 4.2. Kết nối Phần cứng

### 4.2.1. Kết nối Cảm biến DHT11 với Board Điều khiển (ESP8266)

* **Mô tả:** Cảm biến DHT11 cần được kết nối với board ESP8266 để truyền dữ liệu về nhiệt độ và độ ẩm.
* **Hướng dẫn kết nối:**
  1. **Kết nối chân cảm biến:**
     + Chân VCC của DHT11 kết nối với chân 3V3 trên ESP8266.
     + Chân GND của DHT11 kết nối với chân GND trên ESP8266.
     + Chân DATA của DHT11 kết nối với một chân GPIO trên ESP8266 (ví dụ, GPIO2).
  2. **Cung cấp nguồn:** Đảm bảo rằng board ESP8266 được cấp nguồn qua cổng USB hoặc qua một nguồn cấp phù hợp.

### 4.2.2. Cấu hình Board Điều khiển để Truyền Dữ liệu qua HTTP/MQTT

* **Mô tả:** ESP8266 cần được lập trình để đọc dữ liệu từ DHT11 và truyền dữ liệu này đến OpenRemote qua HTTP hoặc MQTT.
* **Hướng dẫn cấu hình:**
  1. **Cài đặt Arduino IDE:** Nếu chưa có, tải và cài đặt Arduino IDE từ Arduino.
  2. **Cài đặt thư viện:** Cài đặt các thư viện cần thiết cho ESP8266 và DHT11 trong Arduino IDE (ví dụ: ESP8266 và DHT).
  3. **Viết mã nguồn:**

## 4.3. Tích hợp vào Hệ thống

**1. Thiết lập OpenRemote để Nhận và Xử lý Dữ liệu từ Board Điều khiển**

* **Mô tả:** OpenRemote cần được cấu hình để nhận dữ liệu từ ESP8266 qua giao thức MQTT và hiển thị trên dashboard.
* **Hướng dẫn thiết lập:**
  1. **Truy cập OpenRemote:** Mở trình duyệt và truy cập vào địa chỉ của OpenRemote (http://localhost:8080).
  2. **Tạo kết nối MQTT:** Trong giao diện quản lý của OpenRemote, cấu hình kết nối đến broker MQTT.
     + Cung cấp thông tin về broker MQTT, bao gồm địa chỉ IP và port (thường là 1883).
  3. **Thiết lập topic:** Đảm bảo rằng OpenRemote đăng ký lắng nghe topic mà ESP8266 xuất bản dữ liệu (ví dụ: "sensor/data").
  4. **Cấu hình dữ liệu:** Tạo các cấu hình dữ liệu và các widget trên dashboard để hiển thị dữ liệu nhiệt độ và độ ẩm nhận được từ cảm biến.
  5. **Kiểm tra và giám sát:** Đảm bảo rằng dữ liệu từ ESP8266 được hiển thị chính xác trên dashboard và thực hiện các điều chỉnh nếu cần thiết.

# 5 SO SÁNH HỆ THỐNG VỚI MỘT PLATFORM KHÁC

## 5.1. So sánh với Platform Triển khai Docker

### 5.1.1. Triển khai Docker

* **Ưu điểm:**
  + **Nhất quán và Di động:** Docker cung cấp một môi trường nhất quán cho ứng dụng trên bất kỳ hệ điều hành nào, giúp dễ dàng di chuyển và triển khai ứng dụng giữa các môi trường phát triển, thử nghiệm và sản xuất.
  + **Cô lập và Bảo mật:** Các container Docker hoạt động độc lập, giúp ngăn chặn xung đột phần mềm và tăng cường bảo mật.
  + **Quản lý dễ dàng:** Docker Compose cho phép cấu hình nhiều dịch vụ trong một tệp đơn giản, giúp quản lý và triển khai các ứng dụng phức tạp dễ dàng hơn.
  + **Khả năng mở rộng:** Docker dễ dàng mở rộng và cân bằng tải, cho phép triển khai các dịch vụ bổ sung hoặc mở rộng quy mô hệ thống mà không làm ảnh hưởng đến các dịch vụ khác.
* **Nhược điểm:**
  + **Độ phức tạp:** Đối với người mới, việc thiết lập và cấu hình Docker có thể đòi hỏi thời gian học hỏi và làm quen.
  + **Hiệu suất:** Một số ứng dụng có thể gặp vấn đề về hiệu suất khi chạy trong container so với khi chạy trên phần cứng trực tiếp, đặc biệt là với các ứng dụng yêu cầu hiệu suất cao.

### 5.1.2. Triển khai Serviceless

* **Ưu điểm:**
  + **Chi phí hiệu quả:** Với dịch vụ serverless, chỉ trả tiền cho lượng tài nguyên sử dụng, không cần quản lý máy chủ hoặc cơ sở hạ tầng.
  + **Khả năng mở rộng tự động:** Các dịch vụ serverless tự động mở rộng theo nhu cầu, giúp xử lý tải cao mà không cần cấu hình thêm.
  + **Đơn giản hóa quản lý:** Không cần quản lý máy chủ hoặc cấu hình phức tạp, giúp tập trung vào phát triển ứng dụng.
* **Nhược điểm:**
  + **Hạn chế tính tùy chỉnh:** Các dịch vụ serverless có thể hạn chế về mặt tùy chỉnh và cấu hình so với Docker.
  + **Cold Start:** Một số dịch vụ serverless có thể gặp độ trễ khi khởi động lần đầu (cold start), ảnh hưởng đến hiệu suất trong một số trường hợp.

### 5.1.3. Sử dụng Trực tiếp trên Phần Cứng

* **Ưu điểm:**
  + **Hiệu suất cao:** Chạy trực tiếp trên phần cứng có thể cung cấp hiệu suất tốt nhất mà không bị ảnh hưởng bởi lớp ảo hóa.
  + **Tùy chỉnh đầy đủ:** Có khả năng kiểm soát hoàn toàn môi trường và cấu hình phần mềm.
* **Nhược điểm:**
  + **Quản lý phức tạp:** Cần phải quản lý phần cứng và hệ điều hành, điều này có thể tốn thời gian và công sức.
  + **Khả năng mở rộng hạn chế:** Khả năng mở rộng không linh hoạt như Docker hoặc serverless. Mở rộng hệ thống thường yêu cầu nâng cấp phần cứng hoặc thay đổi cấu hình phức tạp.

# 6 KẾT LUẬN

## 6.1. Tóm tắt Các Điểm Mạnh của Hệ thống Sử dụng OpenRemote

* **Tính Linh Hoạt:** OpenRemote cung cấp nền tảng linh hoạt để quản lý và giám sát các thiết bị IoT như cảm biến DHT11. Hệ thống có khả năng tích hợp với nhiều loại cảm biến và giao thức truyền thông.
* **Giao Diện Dashboard:** Dashboard của OpenRemote cho phép người dùng dễ dàng theo dõi và phân tích dữ liệu từ cảm biến, cung cấp các công cụ tương tác và tùy chỉnh mạnh mẽ.
* **Tích Hợp MQTT:** Khả năng tích hợp với MQTT giúp hệ thống truyền dữ liệu hiệu quả và đáng tin cậy, phù hợp cho các ứng dụng IoT cần cập nhật dữ liệu thời gian thực.

## 6.2. Đánh Giá về Tiềm Năng Ứng Dụng Thực Tế và Cải Tiến Trong Tương Lai

* **Tiềm Năng Ứng Dụng:** OpenRemote có tiềm năng cao trong các ứng dụng IoT cần giám sát môi trường, quản lý thiết bị từ xa, và tự động hóa quy trình dựa trên dữ liệu cảm biến. Nó có thể được áp dụng trong các lĩnh vực như quản lý môi trường, hệ thống thông minh trong nhà, và giám sát công nghiệp.
* **Cải Tiến:**
* **Tăng Cường Tính Tương Tác:** Phát triển các tính năng tương tác bổ sung trên dashboard để cung cấp thêm thông tin và kiểm soát cho người dùng.
* **Tối Ưu Hóa Hiệu Suất:** Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa hiệu suất để giảm độ trễ trong việc xử lý và hiển thị dữ liệu.
* **Mở Rộng Tính Năng:** Cập nhật và mở rộng khả năng của hệ thống để hỗ trợ các cảm biến và thiết bị mới, cùng với các giao thức truyền thông khác.

**7 CODE**

**Code trong docker-compose.yml**

version: '2.4'

volumes:

proxy-data:

manager-data:

postgresql-data:

services:

proxy:

image: openremote/proxy:${PROXY\_VERSION:-latest}

restart: always

depends\_on:

manager:

condition: service\_healthy

ports:

- "80:80" # Needed for SSL generation using letsencrypt

- "${OR\_SSL\_PORT:-443}:443"

- "8883:8883"

# - "127.0.0.1:8404:8404" # Localhost metrics access

volumes:

- proxy-data:/deployment

environment:

LE\_EMAIL: ${OR\_EMAIL\_ADMIN:-}

DOMAINNAME: ${OR\_HOSTNAME:-localhost}

DOMAINNAMES: ${OR\_ADDITIONAL\_HOSTNAMES:-}

# USE A CUSTOM PROXY CONFIG - COPY FROM https://raw.githubusercontent.com/openremote/proxy/main/haproxy.cfg

#HAPROXY\_CONFIG: '/data/proxy/haproxy.cfg'

postgresql:

restart: always

image: openremote/postgresql:${POSTGRESQL\_VERSION:-latest}

shm\_size: 128mb

volumes:

- postgresql-data:/var/lib/postgresql/data

- manager-data:/storage

keycloak:

restart: always

image: openremote/keycloak:${KEYCLOAK\_VERSION:-latest}

depends\_on:

postgresql:

condition: service\_healthy

volumes:

- ./deployment:/deployment

environment:

KEYCLOAK\_ADMIN\_PASSWORD: ${OR\_ADMIN\_PASSWORD:-secret}

KC\_HOSTNAME: ${OR\_HOSTNAME:-localhost}

KC\_HOSTNAME\_PORT: ${OR\_SSL\_PORT:--1}

manager:

# privileged: true

restart: always

image: openremote/manager:${MANAGER\_VERSION:-latest}

depends\_on:

keycloak:

condition: service\_healthy

ports:

- "1883:1883"

# - "127.0.0.1:8405:8405" # Localhost metrics access

environment:

OR\_SETUP\_TYPE:

OR\_ADMIN\_PASSWORD:

OR\_SETUP\_RUN\_ON\_RESTART:

OR\_EMAIL\_HOST:

OR\_EMAIL\_USER:

OR\_EMAIL\_PASSWORD:

OR\_EMAIL\_X\_HEADERS:

OR\_EMAIL\_FROM:

OR\_EMAIL\_ADMIN:

OR\_METRICS\_ENABLED: ${OR\_METRICS\_ENABLED:-true}

OR\_HOSTNAME: ${OR\_HOSTNAME:-localhost}

OR\_ADDITIONAL\_HOSTNAMES:

OR\_SSL\_PORT: ${OR\_SSL\_PORT:--1}

OR\_DEV\_MODE: ${OR\_DEV\_MODE:-false}

# The following variables will configure the demo

OR\_FORECAST\_SOLAR\_API\_KEY:

OR\_OPEN\_WEATHER\_API\_APP\_ID:

OR\_SETUP\_IMPORT\_DEMO\_AGENT\_KNX:

OR\_SETUP\_IMPORT\_DEMO\_AGENT\_VELBUS:

volumes:

- manager-data:/storage

**Code trong arduino**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>  // Thêm thư viện DHT

// WiFi

const char\* ssid = "CuongDZ"; // Tên wifi

const char\* password = "0901926570"; // Password

// MQTT Broker

const char\* mqtt\_server = "192.168.1.82";

unsigned int mqtt\_port = 1883; //SSL 8883 NoneSSL 1883

const char\* username = "master:mqtt"; // Service User Realm:Serviceuser

const char\* mqttpass = "hCFzb4XBXv4qvK5P2jZAIgiGh35HpJDP"; // Service User Secret

const char\* ClientID = "client123";

// LastWill

const char\* lastwill = "master/client123/writeattributevalue/writeAttribute/28Bt1qsMpKCx9qvWEDLaLn";

const char\* lastwillmsg = "0";

// Subscribing Topic

const char \*topic = "master/client123/attribute/writeAttribute/28Bt1qsMpKCx9qvWEDLaLn";



// DHT Sensor Settings

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// Objects

WiFiClient askClient; // WiFi Client

PubSubClient client(askClient);

void setup() {

  Serial.begin(115200);

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

  }

  Serial.println(WiFi.localIP());

  client.setServer(mqtt\_server, mqtt\_port);

  client.setCallback(callback);

  dht.begin();

  delay(10000); // Delay 10 giây trước khi gửi dữ liệu tiếp theo

}

void loop() {

  if (!client.connected()) {

    reconnect();

  }

  client.loop();

  // Đọc dữ liệu từ cảm biến DHT11

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  // Kiểm tra xem dữ liệu có hợp lệ không

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  // Chuyển đổi dữ liệu thành chuỗi để publish lên MQTT

  String temp = String(t);

  String hum = String(h);

  // Publish dữ liệu lên các topic MQTT

  client.publish("master/client123/writeattributevalue/temp/28Bt1qsMpKCx9qvWEDLaLn", temp.c\_str());

  client.publish("master/client123/writeattributevalue/hum/28Bt1qsMpKCx9qvWEDLaLn", hum.c\_str());

  Serial.print("Temp: ");

  Serial.println(temp.c\_str());

  Serial.print("Hum: ");

  Serial.println(hum.c\_str());

  Serial.println("- - - - -");

  delay(10000); // Delay 10 giây trước khi gửi dữ liệu tiếp theo

}

// MQTT callback

void callback(char\* topic, byte\* payload, unsigned int length) {

  Serial.print("Message arrived in topic: ");

  Serial.println(topic);

  Serial.print("Message: ");

  for (int i = 0; i < length; i++) {

    Serial.print((char)payload[i]);

  }

  Serial.println();

}

// MQTT reconnect

void reconnect() {

  while (!client.connected()) {

    Serial.print("Attempting MQTT connection...");

    if (client.connect(ClientID, username, mqttpass, lastwill, 1, 1, lastwillmsg)) {

      Serial.println("connected");

      client.subscribe(topic);

    } else {

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

      delay(5000);

    }

  }

}

# TÀI LIỆU THAM KHẢO

<https://forum.openremote.io/t/while-connecting-esp8266-with-or-using-mqtt-broker-fails-with-state-2/1934>

<https://forum.openremote.io/t/connecting-esp8266-to-openremote-mqtt-broker/2008>

<https://forum.openremote.io/t/openremote-and-esp8266/124>

<https://docs.openremote.io/docs/user-guide/gateways-and-devices/connect-esp32-or-esp8266-using-mqtt/>

<https://docs.openremote.io/docs/user-guide/manager-ui/>

<https://github.com/openremote/openremote>

<https://forum.openremote.io/t/failed-with-state-2the-client-esp8266-client-84f3eb52fb33-connects-to-the-public-mqtt-broker/2411>

<https://forum.openremote.io/t/connect-esp32-to-open-remote-with-mqtt/1776>