

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



ĐỒ ÁN MÔN HỌC THIẾT KẾ LUẬN LÝ (CO3091)

HỆ THỐNG IOT GIÁM SÁT CHẤT LƯỢNG MÔI TRƯỜNG

INSTRUCTOR: Phan Đình Thế Duy

STUDENTS:	Huỳnh Hữu Hạnh	1910161
	Võ Minh Chánh	1910057

Ho Chi Minh City, January 2022

Contents

1	BẢNG PHÂN VIỆC	2
2	GIỚI THIỆU ĐỒ ÁN	3
3	THIẾT KẾ CHƯƠNG TRÌNH	3
3.1	Tổng quan	3
3.2	Các chức năng	4
3.2.1	Giao tiếp qua UART	4
3.2.2	Giao tiếp trên Adafuit IO	4
3.2.3	Giao tiếp trên thiết bị di động Android thông qua giao thức MTQQ	5
3.2.4	Auto-control mode	6
3.2.5	Manual-control mode	6
3.3	Hệ thống phần cứng	7
3.3.1	Công tắc điện từ	7
3.3.2	Biến trở vạn	8
3.3.3	Board KIT PIC	8
3.4	Sequence diagram	9
4	HIỆN THỰC SẢN PHẨM	9
4.1	Thiết kế schematic trên Proteus	9
4.2	Các file chương trình	10
4.2.1	File UART hiện thực trên board KIT PIC	10
4.2.2	File giao tiếp giữa terminal và server	10
4.2.3	File giao tiếp giữa server và điện thoại	10
5	HÌNH ẢNH SẢN PHẨM	11
5.1	Tổng thể mạch	11
5.2	Giao diện trên website	11
5.3	Giao diện trên điện thoại	12
6	ĐÁNH GIÁ	12
6.1	Ưu điểm	12
6.2	Khuyết điểm	12
7	KẾT LUẬN	12



1 BẢNG PHÂN VIỆC

No.	Full name	Student ID	Problems	Percentage of work
1	Huỳnh Hữu Hạnh	1910161	Giao tiếp giữa UART và Web Viết báo cáo	50 %
2	Võ Minh Chánh	1910057	Giao tiếp giữa server và điện thoại Hiện thực giao tiếp UART Hàn mạch	50%

2 GIỚI THIỆU ĐỒ ÁN

Trong quá trình phát triển của con người, những cuộc cách mạng về công nghệ đóng một vai trò rất quan trọng, chúng làm thay đổi từng ngày từng giờ cuộc sống của con người, theo hướng hiện đại hơn. Đi đôi với quá trình phát triển của con người, những thay đổi do chính tác động của con người trong tự nhiên, trong môi trường sống cũng đang diễn ra, tác động trở lại chúng ta, như ô nhiễm môi trường, khí hậu thay đổi, v.v... Dân số càng tăng, nhu cầu cũng tăng theo, các dịch vụ, các tiện ích từ đó cũng được hình thành và phát triển theo. Đặc biệt là áp dụng các công nghệ của các ngành điện tử, công nghệ thông tin và truyền thông vào trong thực tiễn cuộc sống con người. Công nghệ Internet of Things (IoT) nói chung và công nghệ cảm biến không dây (Wireless Sensor) nói riêng được tích hợp từ các kỹ thuật điện tử, tin học và viễn thông tiên tiến vào trong mục đích nghiên cứu, giải trí, sản xuất, kinh doanh, v.v..., phạm vi này ngày càng được mở rộng, để tạo ra các ứng dụng đáp ứng cho các nhu cầu trên các lĩnh vực khác nhau.

Hiện nay, mặc dù khái niệm IoT và công nghệ cảm biến không dây đã trở nên khá quen thuộc và được ứng dụng khá nhiều trong các lĩnh vực của đời sống con người, đặc biệt ở các nước phát triển có nền khoa học công nghệ tiên tiến. Tuy nhiên, những công nghệ này chưa được áp dụng một cách rộng rãi ở nước ta, do những điều kiện về kỹ thuật, kinh tế, nhu cầu sử dụng. Song nó vẫn hứa hẹn là một đích đến tiêu biểu cho các nhà nghiên cứu, cho những mục đích phát triển đầy tiềm năng.

Được sự định hướng và chỉ dẫn của thầy Phan Đình Thế Duy chúng em đã chọn đề tài đồ án "Hệ thống IoT cho giám sát chất lượng môi trường". Trên cơ sở tìm hiểu về IoT nói chung và giao thức giao tiếp Gateway và UART nói riêng, đồ án còn thực hiện một thực nghiệm trên các thiết bị đa dạng (website, điện thoại) để giám sát các thông số môi trường tiêu biểu (COD, pH, NH₄) đối với các trang trại, mô hình kinh doanh.

3 THIẾT KẾ CHƯƠNG TRÌNH

3.1 Tổng quan

Đề tài "Hệ thống IoT cho giám sát chất lượng môi trường" được thiết kế cho hệ thống trồng trọt giám sát môi trường đất. Hệ thống đảm bảo hoạt động đúng được các chức năng của mô hình giám sát và đảm bảo chất lượng trồng trọt.

Các thông số nồng độ các chất cơ bản của đất mà hệ thống giám sát:

- COD (Chemical Oxygen Demand): là lượng oxy cần thiết để oxy hoá các hợp chất hoá học trong nước bao gồm cả vô cơ và hữu cơ.
- SS (Chất rắn lơ lửng): là những chất rắn hầu như không tan ở trong nước và tạo nên độ đục của nước như axit sunphat đồng, chất hữu cơ photpho, oxit nhôm, oxit đồng, oxit sắt...
- NH₄ (Amoni): là chất dễ dàng hòa tan trong nước giúp cân bằng tỉ lệ NH₃ trong môi trường nước.
- NO₃ (Nitrate): là chất mà cây cối hấp thụ NO₃ trong đất để lấy dưỡng chất và có thể sẽ tạo một dư lượng nhỏ trong lá và quả.
- Độ pH: là thước đo độ axit hoặc tính base (độ kiềm) của đất.

Mô hình được trình bày dưới đây là mô hình gồm có:

- Có 5 biến trở con vặn dùng để giám sát và thay đổi thủ công các thông số đã nêu ở trên, cùng với màn LCD để hiển thị sự thay đổi của các thông số nồng độ.
- Một module 2 relay 5V dùng để đảm bảo các thông số nồng độ ở ngưỡng cho phép.
- Các phương thức điều khiển giúp người dùng dễ dàng tương tác với phần cứng hệ thống:
 1. Giao tiếp qua UART
 2. Giao tiếp trên Adafuit IO
 3. Giao tiếp trên thiết bị di động Android

3.2 Các chức năng

3.2.1 Giao tiếp qua UART

Đầu tiên, ta cắm dây UART - USB vào cổng USB của máy tính, trên máy tính sẽ hiện COM mới. Sau đó, kết nối mạch để giao tiếp UART:

1. Tạo Port COM cách phía trên.
2. Trên Proteus, bật SWITCH của UART sang ON (màu đen ở vị trí ON).
3. Mở phần mềm Terminal trong thư mục UART-software, COM port chọn COM2 và sau đó nhấn Connect.
4. Muốn gửi dữ liệu xuống board thì gõ vào khung trắng bên trái nút Send.

3.2.2 Giao tiếp trên Adafruit IO

Chúng ta sẽ phân ứng dụng thành 4 thành phần cơ bản, bao gồm nút cảm biến, gateway trung tâm, server và các thiết bị để theo dõi dữ liệu và điều khiển từ xa, như trình bày ở hình bên dưới:

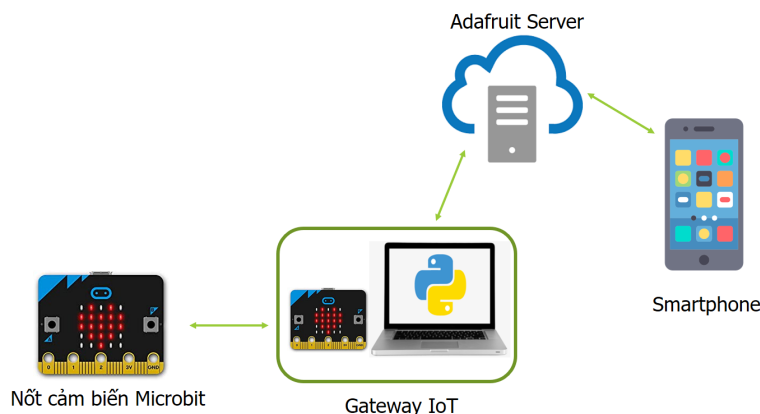


Figure 1: Kiến trúc 4 thành phần trong ứng dụng IoT

Tại đây, chương trình viết bằng ngôn ngữ Python sẽ gửi dữ liệu này lên server Adafruit IO. Từ đó, các thiết bị theo dõi từ xa như điện thoại di động hay thậm chí là một máy tính khác có thể theo dõi được dữ liệu của hệ thống. Trong trường hợp muốn điều khiển một thiết bị, luồng dữ liệu sẽ đi ngược từ phía thiết bị đầu cuối cho đến nút cảm biến, để thi hành lệnh điều khiển.

Trình tự hiện thực:

- Tạo kênh dữ liệu (Feed). Để có thể lưu dữ liệu trên server, chúng ta cần phải phân loại cho nó. Thông thường, chúng ta gọi là một kênh dữ liệu, hay feed. Mỗi đối tượng trong hệ thống cũng sẽ thường có 1 kênh dữ liệu cho riêng nó. Ở đồ án này, tên là **TKLL**.
- Thiết kế Dashboard trên Adafruit IO:



Figure 2: Giao diện Dashboard

- Hiện thực IoT Gateway bằng Python. Hiện thực chương trình đã nêu trong file code phần 4.2.2 File giao tiếp giữa terminal và server.
- Gửi dữ liệu lên Adafruit IO
- Lập trình cho Gateway

3.2.3 Giao tiếp trên thiết bị di động Android thông qua giao thức MTQQ

1. Giao thức MTQQ

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp / thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.

2. Vị trí của MQTT trong mô hình IoT

Một số ưu điểm nổi bật của MQTT như: băng thông thấp, độ tin cậy cao và có thể sử dụng ngay cả khi hệ thống mạng không ổn định, tốn rất ít byte cho việc kết nối với server và connection có thể giữ trạng thái open xuyên suốt, có thể kết nối nhiều thiết bị (MQTT client) thông qua một MQTT server (broker). Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng IoT.

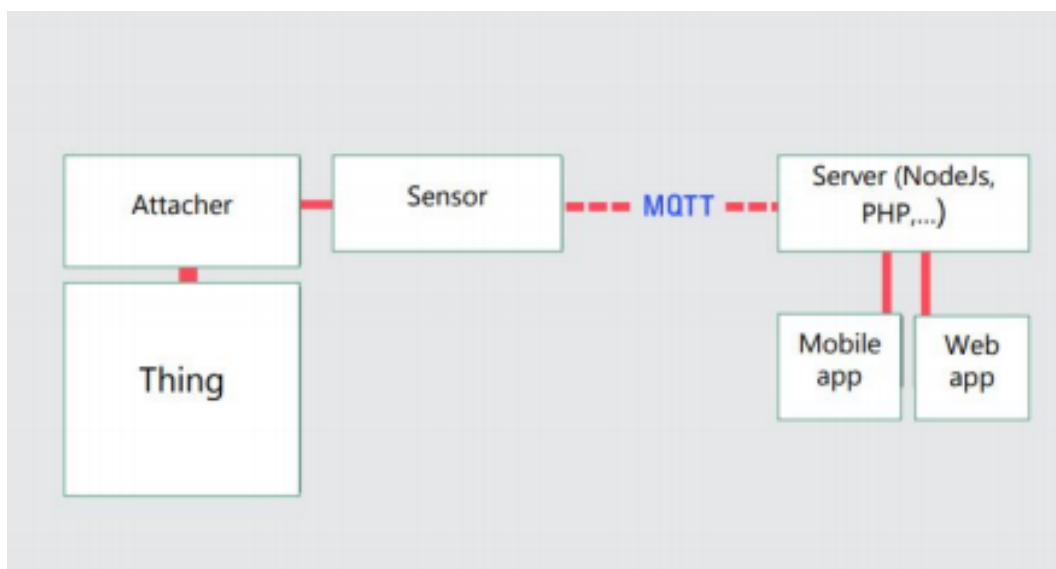


Figure 3: MQTT trong mô hình IoT

3. Cơ chế tổng quan

MQTT hoạt động theo cơ chế client/server, nơi mà mỗi cảm biến là một khách hàng (client) và kết nối đến một máy chủ, có thể hiểu như một Máy chủ môi giới (broker), thông qua giao thức TCP (Transmission Control Protocol). Broker chịu trách nhiệm điều phối tất cả các thông điệp giữa phía gửi đến đúng phía nhận.

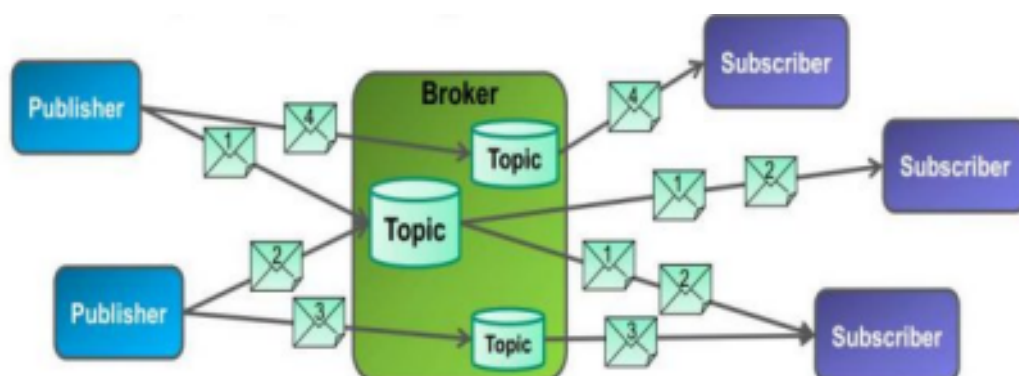


Figure 4: Cơ chế hoạt động của MTQQ

3.2.4 Auto-control mode

Trong chế độ auto, nồng độ COD trên 80% thì relay sẽ tự động đóng và để tránh nhiều thì nồng độ COD nhỏ hơn 70% thì relay sẽ mở. Tất cả các nồng độ khác đều hoạt động theo cơ chế trên. Màn LCD trên board KIT PIC hiển thị 5 thông số nồng độ từ 0% đến 100%. Góc bên phải dưới, hiển thị kí từ chế độ hệ thống. Dữ liệu sẽ được đồng bộ trên cả 3 thiết bị.

Khi hệ thống đang hoạt động, người dùng bấm phím “N” trên cửa sổ terminal để chuyển sang chế độ Auto mode. Khi muốn thoát khỏi chế độ Auto mode, người điều khiển có thể bấm các nút sau: “A”, “B”, “C”, “D” trên cửa sổ terminal. Hệ thống sẽ giữ nguyên trạng thái nếu người điều khiển bấm ngoài các nút kể trên.

3.2.5 Manual-control mode

Đây là chế độ người điều khiển tự điều chỉnh các thông số nồng độ theo ý muốn. Ở chế độ này các switch Manual sẽ ở trạng thái ON. Cơ chế về nồng độ giống nhau cả hai chế độ.

Thao tác trên terminal:

- Khi gõ phím "N" từ bàn phím lên terminal thì hệ thống sẽ vào chế độ auto.
- Khi gõ một trong các nút sau: "A", "B", "C", "D" từ bàn phím lên terminal thì hệ thống sẽ vào chế độ thủ công. Cụ thể:
 1. Khi gõ phím "A" thì relay 1 và relay 2 sẽ cùng bật.
 2. Khi gõ phím "B" thì relay 1 sẽ tắt và relay 2 sẽ bật.
 3. Khi gõ phím "C" thì relay 1 sẽ bật và relay 2 sẽ tắt.
 4. Khi gõ phím "D" thì relay 1 và relay 2 sẽ cùng tắt.

Giao diện trên web và điện thoại có 2 phần:

- Phần để giám sát các thông số mà sensor cung cấp.
- Phần là để người dùng tương tác với hệ thống là các switch. Trong đó, khi switch RLM tắt thì hệ thống chuyển sang chế độ tự động. Lúc này RL1 và RL2 chuyển trạng thái theo trạng thái relay trên mạch. Khi switch RLM bật thì hệ thống chuyển sang chế độ thủ công, người dùng có thể tương tác trực tiếp với relay trên mạch bằng cách bật tắt các switch relay.

Dữ liệu sẽ được đồng bộ trên cả 3 thiết bị.

3.3 Hệ thống phần cứng

3.3.1 Công tắc điện từ

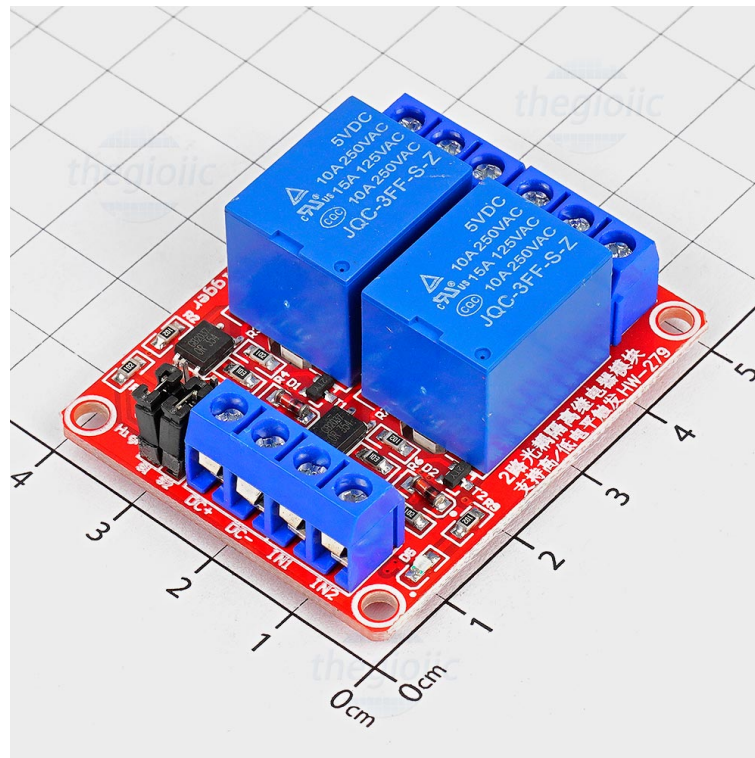


Figure 5: Module 2 Relay 5V Kích Mức Cao/Thấp V1

Thông số kỹ thuật:

- Nguồn cấp: 5 VDC.
- Có opto cách ly.
- Có jumper set mức Cao/Thấp.

- Dòng: 50mA (standby 10mA).
- Ngõ ra relay: 250 VAC 10A or 30 VDC 10A.
- Size: 51 × 41.8mm.

Nguyên lí hoạt động:

- Module relay có 4 chân, 1 chân GND, 1 chân VCC và 2 chân input.
- VCC và GND là nguồn cấp vào cho relay hoạt động.
- Khi kích mức thấp, tín hiệu đầu vào digital là 0 thì relay đóng lại. Ngược lại, khi kích mức cao, tín hiệu đầu vào digital là 1 thì relay mở ra.

3.3.2 Biến trở vạn

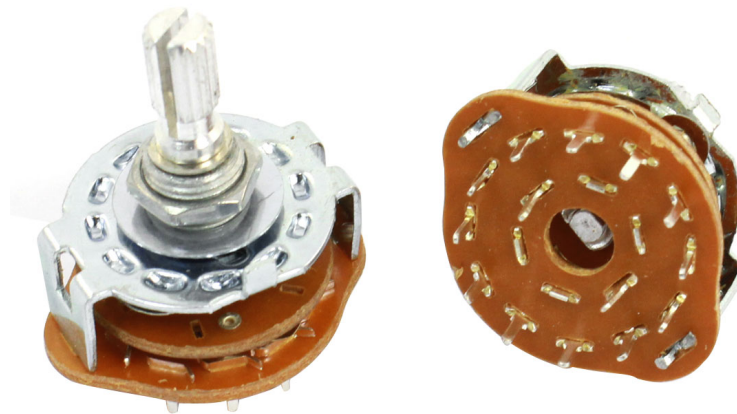


Figure 6: Chiết áp kép WH148 1M kép RK-104

Nguyên lí hoạt động:

- Biến trở có 3 chân: Chân 1 và chân 3 có điện trở cố định.
- Khi xoay núm vặn thì điện trở giữa chân 1 và chân 2, chân 2 và chân 3 thay đổi. Khi nối chân 2 vào đầu đọc analog của vi điều khiển thì giá trị đọc được sẽ từ 0 đến 5V chuyển sang tín hiệu analog có giá trị từ 0 đến 1023, tương trưng cho các sensor.

3.3.3 Board KIT PIC

Board kit PIC được thiết kế và phát triển bởi câu lạc bộ BKIT4U, sử dụng vi điều khiển PIC18F4620. Ngoài ra, trên KIT PIC này còn có các khối nhập xuất, ma trận phím 4x4, màn hình LCD 16x2, 8 LED đơn hiển thị, các port vào / ra, ...

Trong đề tài đồ án môn học lần này, em sử dụng board kit PIC này để thiết kế, lập trình sản phẩm.

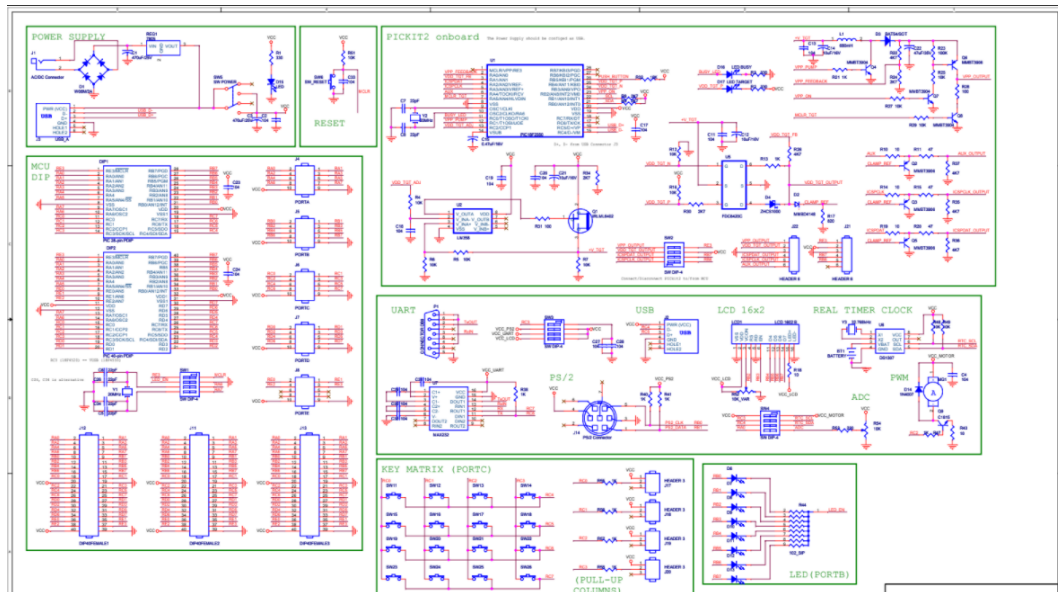


Figure 7: PIC KIT Schematic

3.4 Sequence diagram

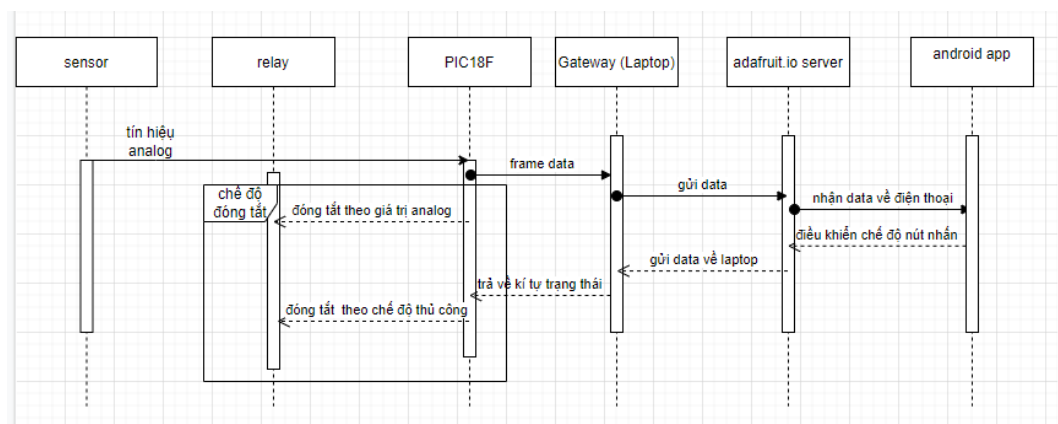


Figure 8: Sequence diagram của hệ thống

4 HIỆN THỰC SẢN PHẨM

4.1 Thiết kế schematic trên Proteus

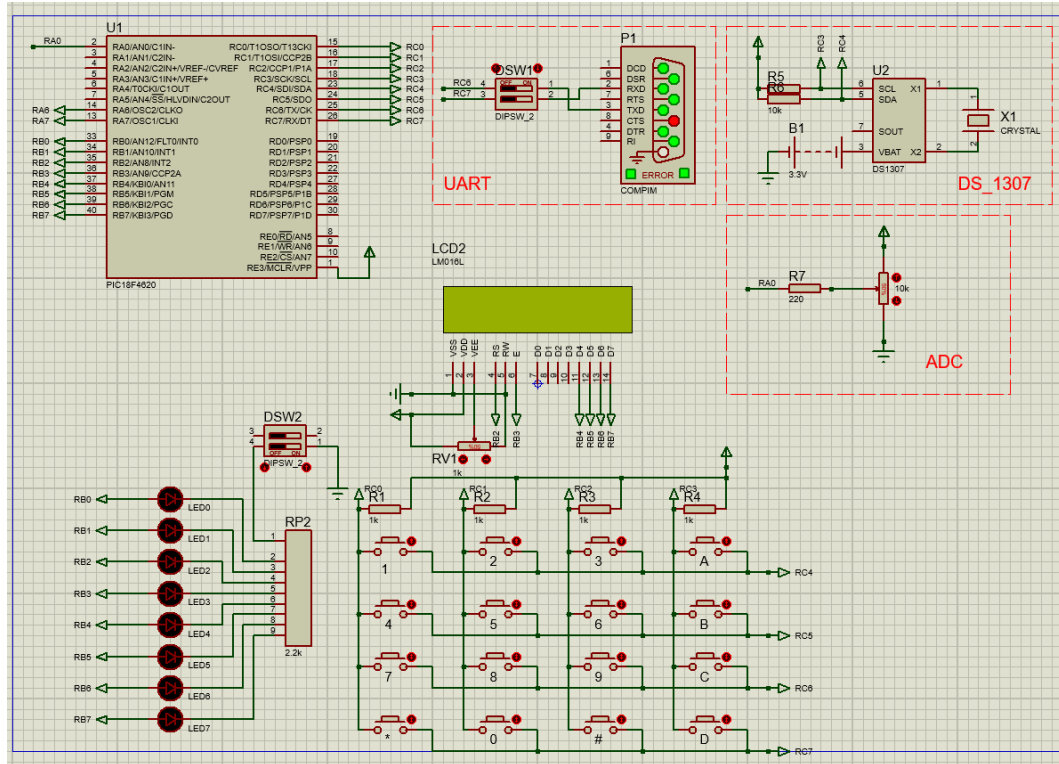


Figure 9: Hiện thực schematic trên proteus

4.2 Các file chương trình

4.2.1 File UART hiện thực trên board KIT PIC

- interrupt.c, timer.c: giúp tạo ra flag-timer3 với chu kỳ 50 ms. Hệ thống sử dụng flag-timer3 như một công cụ để thiết lập thời gian 1 giây của bộ đếm ngược. Khi flag-timer3 lặp lại 20 lần, lúc này thời gian thực hiện vòng lặp là $50\text{ms} \times 20 = 1000\text{ ms} = 1\text{s}$. Tức là ta cho vòng lặp flag-timer 3 đủ 20 lần, cờ flag-1s sẽ được bật lên và ta sử dụng cờ này như chu kỳ 1 giây cho hệ thống.
- adc.c: đọc tín hiệu của biến trở.
- lcd.c: chứa các hàm hoặc thủ tục làm việc và hiển thị trên LCD 16x2.
- timer.c: hiện thực trạng thái của relay vì nó có độ ưu tiên hiện thực cao hơn main.c.
- main.c: hiển thị giá trị lên màn hình LCD và gửi chuỗi dữ liệu lên terminal.

4.2.2 File giao tiếp giữa terminal và server

Chức năng giao tiếp được thực hiện bằng ngôn ngữ Python:

- main.py: giao tiếp thông qua USERNAME và KEY do Adafruit IO cung cấp. Nhận dữ liệu từ terminal, xử lý dữ liệu nhận được lên server và ngược lại.

4.2.3 File giao tiếp giữa server và điện thoại

Chức năng giao tiếp được thực hiện bằng ngôn ngữ Java và giao thức MQTT:

- MainActivity.java: dùng để hiển thị thông số và trạng thái nút nhấn trên màn hình điện thoại.
- MQTTHelper.java: gửi nhận dữ liệu từ server về điện thoại.

5 HÌNH ẢNH SẢN PHẨM

5.1 Tổng thể mạch

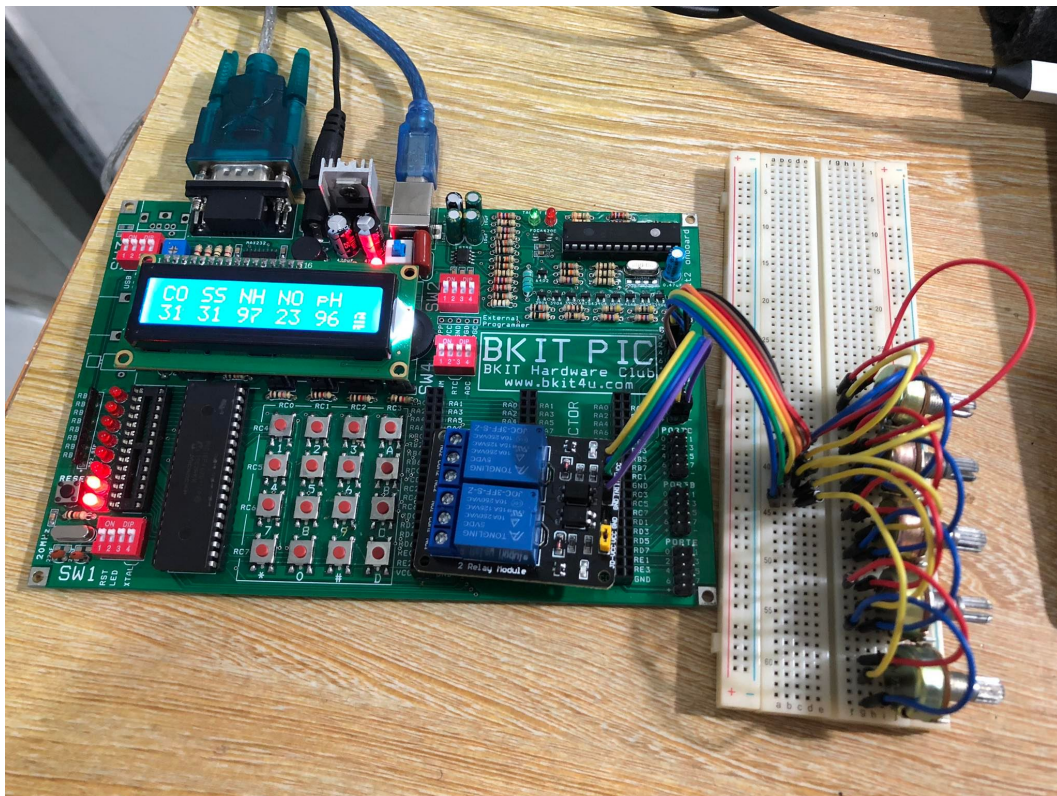


Figure 10: Hiện thực trên board KIT PIC

5.2 Giao diện trên website

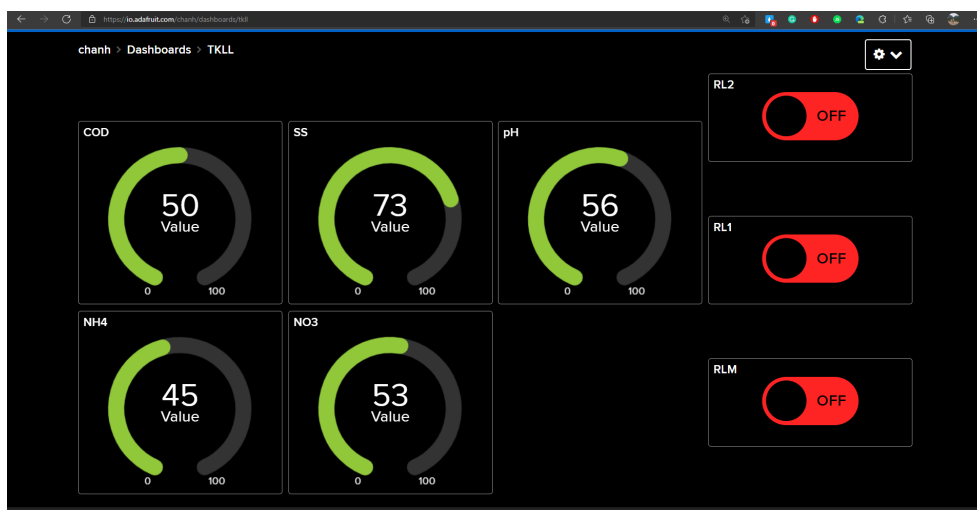


Figure 11: Hiện thực trên Adafruit IO

5.3 Giao diện trên điện thoại

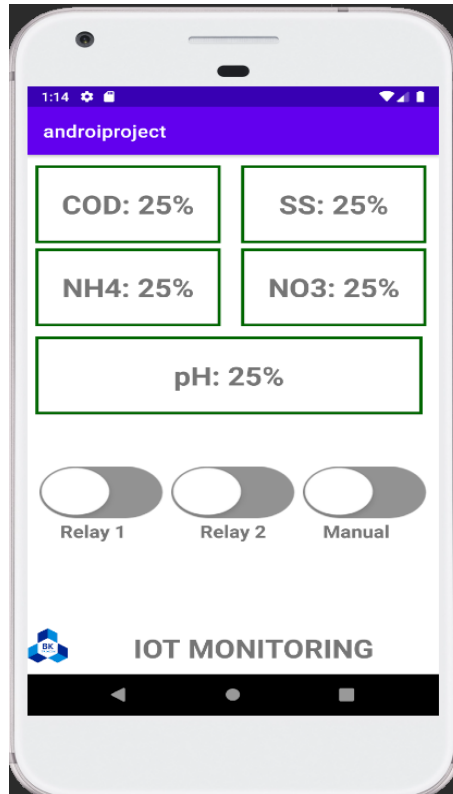


Figure 12: Hiện thực trên thiết bị android

6 ĐÁNH GIÁ

6.1 Ưu điểm

- Các trạng thái của hệ thống chuyển chính xác theo biểu đồ.
- Thiết kế được theo mô hình của một hệ thống IoT.
- Các cơ chế điều chỉnh rõ ràng, dễ thực thao tác và thực hiện.
- Giao diện trực quan, hợp lí đa nền tảng.

6.2 Khuyết điểm

- Độ trễ cập nhật dữ liệu còn cao vì sử dụng dịch vụ server. Có thể cải thiện bằng các sử dụng local server để tăng tốc độ cập nhật dữ liệu.
- Khi vận biến trở, độ nhiễu còn cao.
- Chưa thể vận hành vào thực tế do chỉ có những tính năng cơ bản của một hệ thống IoT.

7 KẾT LUẬN

Đầu tiên, nhóm chúng em xin cảm ơn sự hướng dẫn tận tình của thầy Phan Đình Thế Duy trong suốt quá trình thực hiện đồ án thiết kế luận lí.

Về đề tài, công nghệ IoT hứa hẹn tạo ra những ứng dụng đầy tiềm năng, có thể áp dụng trong nhiều lĩnh vực khác nhau, mà đối với các công nghệ khác còn nhiều hạn chế. Nhờ quá trình mô phỏng người thiết kế hệ thống có thể đánh giá được chất lượng dịch vụ của web server cung cấp, để từ đó có thể thiết kế hệ thống theo cách tối ưu nhất.

Trên cơ sở nghiên cứu tổng quan về IoT, công nghệ cảm biến không dây và các ứng dụng trong thực tiễn, em đã xây dựng thành công một chương trình thực nghiệm có tính khả thi cao (như đã trình bày trong phần đánh giá kết quả chạy thử nghiệm). Tuy nhiên, do thời gian và số lượng các cảm biến hạn chế, chương trình chưa chạy thử nghiệm với một số lượng lớn các thiết bị cảm biến, vì vậy chưa đánh giá hết được một số vấn đề như: việc truyền nhận dữ liệu từ thiết bị cảm biến lên server còn hạn chế, vấn đề xung đột dữ liệu... Đây cũng là một trong những hướng nghiên cứu, phát triển tiếp theo trong các đồ án tương lai.

References

- [1] **BKIT HARDWARE CLUB**: MPLAB 8.36 và MPLAB C18 trên BKIT PIC
- [2] **THE DARIU FOUNDATION**: Phát triển Gateway IoT bằng Python , by *Lê Trọng Nhân, Nguyễn Trần Hữu Nguyên , Võ Tấn Phương, Nguyễn Thanh Hải và Phạm Văn Vinh*
- [3] **Module Relay - Cách sử dụng relay và những ứng dụng**. Nguồn: <https://bitly.com.vn/qycdnh>
- [4] **Tìm hiểu biến trở vạn**. Nguồn: <https://bitly.com.vn/jhhel3>