**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN VIỄN THÔNG**

---------------o0o---------------

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**HỆ THỐNG CẢM BIẾN**

**VÀ DỰ BÁO SỬ DỤNG MÁY HỌC**

**GVHD: PSG.TS HÀ HOÀNG KHA**

**SVTH: MSSV:**

**NGUYỄN TẤN TOÀN 1513543**

**PHẠM MINH QUÂN 1512688**

**TP. HỒ CHÍ MINH, THÁNG 06 NĂM 2019**

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC BÁCH KHOA Độc lập – Tự do – Hạnh phúc.

-----✩----- -----✩-----

Số: \_\_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Điện Tử**

N**HIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. HỌ VÀ TÊN : NGUYỄN TẤN TOÀN MSSV: 1513543

PHẠM MINH QUÂN 1512688

1. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG**
2. Đề tài: Hệ thống cảm biến và dự báo sử dụng máy học.
3. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):

- Tìm hiểu lý thuyết về Linear Regression, Network và Internet, Wifi, Webserver…

- Tìm hiểu về Kit TivaC LaunchPad MCU TM4C123G, ESP32, các cảm biến DHT22, cảm biến độ ẩm đất, cảm biến ánh sáng, module GPS NEO-6M.

- Sử dụng giải thuật Linear Regression để dự đoán nhiệt độ, độ ẩm, cường độ sáng.

- Thiết kế bộ điều khiển thiết bị thông qua các relay sử dụng Webserver và các nút nhấn.

1. Ngày giao nhiệm vụ luận văn: ...............................
2. Ngày hoàn thành nhiệm vụ: ...................................
3. Họ và tên người hướng dẫn: Phần hướng dẫn

................................................................. .....................................

................................................................. .....................................

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

*Tp.HCM, ngày…... tháng….. năm 20*

**CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH**

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ):.......................

Đơn vị:......................................................

Ngày bảo vệ : ...........................................

Điểm tổng kết: .........................................

Nơi lưu trữ luận văn: ...............................

***LỜI CẢM ƠN***

*Sau quá trình học tập và thực hành tại Khoa Điện – Điện tử, Trường Đại học Bách Khoa TP.HCM và 3 tháng làm luận văn tốt nghiệp nghiên cứu và thực hiện đề tài “Hệ thống cảm biến và dự báo sử dụng máy học”, chúng em xin chân thành cảm ơn các Thầy cô giảng viên, Cán bộ các phòng ban của trường đã giúp đỡ chúng em hoàn thành luận văn tốt nghiệp này.*

*Đồng thời, chúng em xin gửi lời cảm ơn đặc biệt đến PGS.TS. Hà Hoàng Kha – Giảng viên trực tiếp hướng dẫn Luận văn tốt nghiệp, đã chỉ bảo và giúp đỡ chúng em rất nhiều.*

*Chúng em cũng cảm ơn bạn bè đang học tập tại trường, cảm ơn gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ cho chúng em trong suốt quá trình thực hiện để có thể hoàn thành đề tài một cách tốt nhất.*

*Do kiến thức và kinh nghiệm còn hạn chế nên không tránh khỏi những thiếu sót về nội dung lẫn hình thức, rất mong quý Thầy cô nhận xét góp ý đề chúng em có thể hoàn thiện luận văn cũng như chính bản thận mình để tiếp tục phát triển trong tương lai.*

*Tp. Hồ Chí Minh, ngày tháng 06 năm 2019*

**Sinh viên Nguyễn Tấn Toàn Sinh viên Phạm Minh Quân**

**TÓM TẮT LUẬN VĂN**

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Trong khi đó, Internet kết nối vạn vật (Internet of Things – IoT) cũng là một chủ đề nghiên cứu rất được quan tâm và được đưa vào nhiều ứng dụng hiện nay. Sức mạnh thực sự của IoT được khai thác qua sự kết nối trực tiếp hoặc gián tiếp giữa “vạn vật”. Trong hệ thống kết nối, các đối tượng dịch vụ cung cấp các khả năng khác nhau, từ việc tập hợp dữ liệu đơn giản đến giám sát để phân tích các dữ liệu phức tạp.

Dựa vào hai xu hướng kể trên, luận văn này thực hiện đề tài về hệ thống cảm biến và dự báo sử dụng máy học. Trong đó sử dụng kit Tiva C LaunchPad thu thập và xử lý thông tin nhiệt độ - độ ẩm không khí, độ ẩm đất, cường độ ánh sáng từ các cảm biến và vị trí đo đạc thông qua thiết bị định vị GPS. Thông tin đó sẽ được truyền lên Webserver bằng Module Wifi ESP32 và lưu vào cơ sở dữ liệu. Với dữ liệu có được ta tiến hành dự báo bằng máy học sử dụng giải thuật Linear Regression và đưa thông số dự báo lên Website. Bên cạnh đó, đề tài này cũng xây dựng giao diện điều khiển thiết bị qua Internet trên Website và ứng dụng Smartphone.

Chúng em hi vọng luận văn này sẽ phục vụ phần nào đó nhu cầu của con người trong thời đại mới, đáp ứng những nhu cầu ngày một cao trong thời đại công nghệ phát triển nhanh và đa dạng, cung cấp những tiện ích thiết thực nhất đến với đời sống của con người. Đồng thời, qua đề tài này chúng em cũng thể hiện phần nhiều những kiến thức và kĩ năng có được trong suốt quãng thời gian học tập và rèn luyện tại trường Đại học Bách Khoa TP.HCM.

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc10489947)

[1.1 Tổng quan 1](#_Toc10489948)

[1.2 Tình hình nghiên cứu trong và ngoài nước 1](#_Toc10489949)

[1.3 Nhiệm vụ luận văn 2](#_Toc10489950)

[CHƯƠNG 2: LÝ THUYẾT 3](#_Toc10489951)

[2.1 Lý thuyết về Linear Regression (Hồi quy Tuyến tính) 3](#_Toc10489952)

[2.2 Tensorflow 5](#_Toc10489953)

[2.3 Tổng quan về Network và Internet 8](#_Toc10489954)

[2.3.1. Giao thức TCP/IP 8](#_Toc10489955)

[2.3.2. Giao thức HTTP 9](#_Toc10489956)

[2.3.3. Ngôn ngữ PHP 10](#_Toc10489957)

[2.3.4. Cơ sở dữ liệu Database 10](#_Toc10489958)

[CHƯƠNG 3: THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 12](#_Toc10489959)

[3.1 Yêu cầu thiết kế 12](#_Toc10489960)

[3.2 Phân tích thiết kế 12](#_Toc10489961)

[3.2.1. Tiva C Series TM4C123G LaunchPad 13](#_Toc10489962)

[3.2.2. Module Wifi ESP32 14](#_Toc10489963)

[3.2.3. Các cảm biến 17](#_Toc10489964)

[3.2.4. Các thiết bị ngõ ra 21](#_Toc10489965)

[3.3 Sơ đồ khối 23](#_Toc10489966)

[3.4 Sơ đồ mạch chi tiết 25](#_Toc10489967)

[CHƯƠNG 4: THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 28](#_Toc10489968)

[4.1 Thực hiện dự đoán bằng máy học 28](#_Toc10489969)

[4.1.1. Chuẩn bị dữ liệu 28](#_Toc10489970)

[4.1.2. Xử lý dữ liệu 28](#_Toc10489971)

[4.1.3. Tiến hành training 30](#_Toc10489972)

[4.2 Xây dựng chương trình cho phần cứng 34](#_Toc10489973)

[4.2.1. Chương trình nạp vào Tiva C Series TM4C123G LaunchPad 34](#_Toc10489974)

[4.2.2. Chương trình nạp vào Module Wifi ESP32 42](#_Toc10489975)

[4.3 Xây dựng Website 49](#_Toc10489976)

[4.4 Xây dựng ứng dụng điều khiển thiết bị trên Smartphone 66](#_Toc10489977)

[CHƯƠNG 5: KẾT QUẢ THỰC HIỆN 73](#_Toc10489978)

[5.1 Kết quả giải thuật dự báo 73](#_Toc10489979)

[5.2 Kết quả thực hiện phần cứng 79](#_Toc10489980)

[5.3 Kết quả xây dựng Website và ứng dụng Smartphone 82](#_Toc10489981)

[CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 86](#_Toc10489982)

[6.1 Kết luận 86](#_Toc10489983)

[6.2 Hướng phát triển 86](#_Toc10489984)

[TÀI LIỆU THAM KHẢO 87](#_Toc10489985)

[PHỤ LỤC 88](#_Toc10489986)

[1. Tìm hiểu về Wifi 88](#_Toc10489987)

[2. Tìm hiểu về GPS 89](#_Toc10489988)

DANH SÁCH HÌNH MINH HỌA

[Hình 1‑1. Mối quan hệ giữa AI, Machine Learning và Deep Learning 2](#_Toc10655777)

[Hình 2‑1. Ví dụ về một graph trong Tensorflow 6](#_Toc10655778)

[Hình 2‑2. Mô hình 4 tầng của TCP / IP 9](#_Toc10655779)

[Hình 2‑3. Mô tả cách hoạt động của Webserver dựa trên database MySQL và các tệp PHP 11](#_Toc10655780)

[Hình 3‑1. Tiva C Series TM4C123G LaunchPad 13](#_Toc10655781)

[Hình 3‑2. Ảnh thực tế của ESP32 14](#_Toc10655782)

[Hình 3‑3. Sơ đồ ra chân của ESP32 15](#_Toc10655783)

[Hình 3‑4. Cấu hình kết nối ESP32 trên Arduino IDE 17](#_Toc10655784)

[Hình 3‑5. Cảm biến DHT22 18](#_Toc10655785)

[Hình 3‑6. Cảm biến độ ẩm đất 18](#_Toc10655786)

[Hình 3‑7. Cảm biến ánh sáng photodiode 19](#_Toc10655787)

[Hình 3‑8. Module NEO-6M GPS 20](#_Toc10655788)

[Hình 3‑9. GPS Antenna 21](#_Toc10655789)

[Hình 3‑10. Module relay 2 kênh 22](#_Toc10655790)

[Hình 3‑11. Động Cơ Bơm P385 12VDC 3W 23](#_Toc10655791)

[Hình 3‑12. Đèn Compact 23](#_Toc10655792)

[Hình 3‑13. Đèn Led chiếu sáng 24](#_Toc10655793)

[Hình 3‑14. Sơ đồ khối phần cứng tổng quát 24](#_Toc10655794)

[Hình 3‑15. Sơ đồ khối phần cứng chi tiết 25](#_Toc10655795)

[Hình 3‑16. Sơ đồ nguyên lý của mạch vẽ bằng phần mềm Altium 26](#_Toc10655796)

[Hình 3‑17. Mặt trên của bản mạch 27](#_Toc10655797)

[Hình 3‑18. Mặt dưới của bản mạch 28](#_Toc10655798)

[Hình 4‑1. Mô tả mạng Neural Network 32](#_Toc10655799)

[Hình 4‑2. Mô tả mạng Recurrent Neural Network 32](#_Toc10655800)

[Hình 4‑3. Lưu đồ giải thuật điều khiển thiết bị ngõ ra 35](#_Toc10655801)

[Hình 4‑4. Sơ đồ giao tiếp giữa phần cứng và Website 50](#_Toc10655802)

[Hình 4‑5. Địa chỉ của trang web đã tạo 51](#_Toc10655803)

[Hình 4‑6. Quản lý cơ sở dữ liệu được tạo 51](#_Toc10655804)

[Hình 4‑7. Cấu trúc của database “id8287156\_tantoan” 52](#_Toc10655805)

[Hình 4‑8. Cấu trúc của bảng “student” 52](#_Toc10655806)

[Hình 4‑9. Cấu trúc của bảng “weather” 52](#_Toc10655807)

[Hình 4‑10. Cấu trúc của bảng “output” 53](#_Toc10655808)

[Hình 4‑11. Lưu đồ giải thuật đọc dữ liệu từ database lên website 56](#_Toc10655809)

[Hình 4‑12. Các tệp truy xuất và hiển thị dữ liệu 62](#_Toc10655810)

[Hình 4‑13. Cấu trúc giao tiếp giữa ứng dụng và cơ sở dữ liệu 67](#_Toc10655811)

[Hình 4‑14. Nhập thư viện “Volley” để xử lý dữ liệu dạng JSON từ Server 68](#_Toc10655812)

[Hình 4‑15. Cấu trúc ứng dụng Android Webservice 68](#_Toc10655813)

[Hình 4‑16. Layout hiển thị danh sách thiết bị 69](#_Toc10655814)

[Hình 4‑17. Nút nhấn chèn thêm thiết bị 70](#_Toc10655815)

[Hình 4‑18. Layout màn hình thêm thiết bị 71](#_Toc10655816)

[Hình 4‑19. Layout màn hình cập nhật thiết bị 72](#_Toc10655817)

[Hình 4‑20. Các tệp truy xuất thông tin thiết bị từ bảng “student” trên cơ sở dữ liệu 73](#_Toc10655818)

[Hình 4‑21. Trang web hiển thị trạng thái thiết bị 73](#_Toc10655819)

[Hình 5‑1. Đồ thị so sánh nhiệt độ dự báo và thực tế trong 48 giờ 75](#_Toc10655820)

[Hình 5‑2. Đồ thị so sánh độ ẩm tương đối dự báo và thực tế trong 48 giờ 76](#_Toc10655821)

[Hình 5‑3. Dữ liệu bảng “output” trên database 80](#_Toc10655822)

[Hình 5‑4. Mạch in PCB 81](#_Toc10655823)

[Hình 5‑5. Mạch thực tế 82](#_Toc10655824)

[Hình 5‑6. Sơ đồ nối thiết bị với relay điều khiển 83](#_Toc10655825)

[Hình 5‑7. Giao diện Website hiển thị giá trị đo đạc gần nhất 84](#_Toc10655826)

[Hình 5‑8. Giao diện Website hiển thị số liệu dự báo 85](#_Toc10655827)

[Hình 5‑9. Giao diện Website điều khiển thiết bị 85](#_Toc10655828)

[Hình 5‑10. Giao diện ứng dụng điều khiển thiết bị trên Smartphone 86](#_Toc10655829)

**DANH SÁCH BẢNG SỐ LIỆU**

[Bảng 3‑1. Phân tích các lựa chọn về phần cứng 12](#_Toc10651850)

[Bảng 4‑1. Mảng dữ liệu dùng để training 29](#_Toc10651851)

[Bảng 5‑1. Tính toán độ chính xác của dự báo nhiệt độ trong 48 giờ tới 75](#_Toc10651852)

[Bảng 5‑2. Đánh giá trọng số cho giá trị nhiệt độ 76](#_Toc10651853)

[Bảng 5‑3. Đánh giá trọng số cho giá trị độ ẩm 77](#_Toc10651854)

[Bảng 5‑4. Đánh giá trời nắng dự vào số liệu 77](#_Toc10651855)

[Bảng 5‑5. Tổng hợp dữ liệu và kết quả dự báo 78](#_Toc10651856)

# GIỚI THIỆU

## Tổng quan

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, … chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

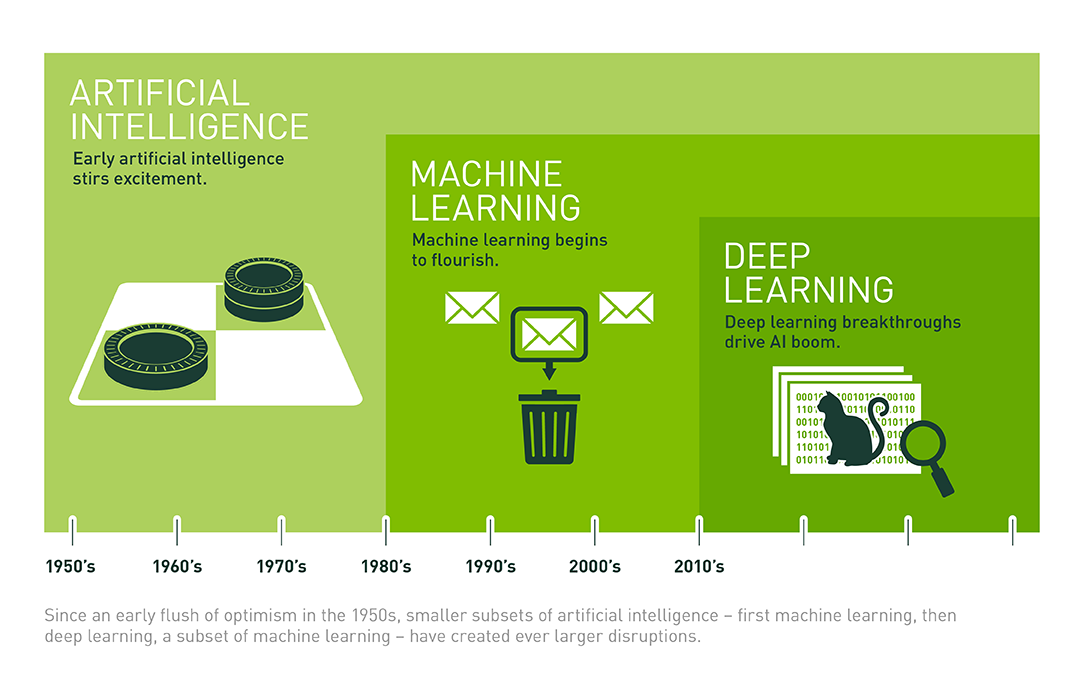
Trong khi đó, Internet kết nối vạn vật (Internet of Things – IoT) cũng là một chủ đề nghiên cứu rất được quan tâm và được đưa vào nhiều ứng dụng hiện nay. Sức mạnh thực sự của IoT được khai thác qua sự kết nối trực tiếp hoặc gián tiếp giữa “vạn vật” (“Things”). Trong hệ thống kết nối, các đối tượng dịch vụ (IoT Services) đóng vai trò như một thực thể vô hình, cung cấp các khả năng khác nhau, từ việc tập hợp dữ liệu đơn giản đến giám sát để phân tích các dữ liệu phức tạp.

Chúng ta biết rằng khi đời sống con người ngày càng nâng cao thì công nghệ càng phải bắt kịp với nhu cầu của con người. Xuất phát từ một nhu cầu trong thực tế đó là tự nuôi trồng thực phẩm sạch, ví dụ như trồng rau xà lách, trồng giá đậu xanh,… thì ta cần có được thông tin các yếu tố của môi trường như nhiệt độ, độ ẩm đất, cường độ ánh sáng,... Rồi từ các thông tin này thì ta mới đưa ra quyết định là nên tưới nước hay không, tăng hay giảm nguồn sáng và nhiều tình huống xử lý khác. Từ hai xu hướng phát triển của IoT và Machine Learning, luận văn này thực hiện đề tài về hệ thống thu thập các thông tin của môi trường như nhiệt độ, độ ẩm… thông qua các cảm biến, sử dụng máy học để đưa ra dự đoán và điều khiển các thiết bị phục vụ cho mô hình trồng cây thông qua Webserver.

## Tình hình nghiên cứu trong và ngoài nước

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia: “Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (Học Sâu). Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc [1].



Hình 1‑1. Mối quan hệ giữa AI, Machine Learning và Deep Learning

## Nhiệm vụ luận văn

Nhiệm vụ của luận văn là thực hiện hệ thống có khả năng thu thập thông tin về thời tiết, sử dụng máy học để đưa ra dự đoán, từ đó điều khiển các thiết bị.

Nội dung 1: Tìm hiểu lý thuyết về Linear Regression, Wifi, Network và Internet, Database, Webserver.

Nội dung 2: Tìm hiểu về Kit TivaC LaunchPad MCU TM4C123G, ESP32, các cảm biến DHT22, cảm biến độ ẩm đất, cảm biến ánh sáng, module GPS NEO-6M.

Nội dung 3: Tìm hiểu về Python và Tensorflow, sử dụng giải thuật Linear Regression để dự đoán nhiệt độ, độ ẩm, cường độ sáng.

Nội dung 4: Thiết kế bộ điều khiển thiết bị thông qua các relay sử dụng Webserver và các nút nhấn.

Nội dung 5: Vận hành hệ thống, thu thập dữ liệu từ cảm biến, tiến hành dự đoán bằng máy học và đưa thông tin lên Website; sử dụng kết quả có được để điều khiển thiết bị hoạt động đúng theo ý muốn.

# LÝ THUYẾT

## Lý thuyết về Linear Regression (Hồi quy Tuyến tính)

Ý tưởng chính của Linear Regression (Hồi quy Tuyến tính) là sử dụng chuỗi các thông tin. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào. Mạng được gọi là hồi quy (Regression) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, mạng có khả năng nhớ các thông tin được tính toán trước đó [2].

Hàm dự đoán của ta có dạng: với Ở đây  là một vector hàng chứa thông tin input, y là một số vô hướng (scalar) biểu diễn output.

Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa đầu ra và 3 đại lượng đầu vào là:



 (1)  
trong đó là các hằng số. Mối quan hệ  bên trên là một mối quan hệ tuyến tính (Linear). Bài toán ta đang làm là một bài toán thuộc loại hồi quy (Regression). Bài toán đi tìm các hệ số tối ưu chính vì vậy được gọi là bài toán Linear Regression.

Chú ý 1:  là giá trị thực của output (dựa trên số liệu thống kê ta có trong tập training data), trong khi  là giá trị mà mô hình Linear Regression dự đoán được. Nhìn chung,  và  là hai giá trị khác nhau do có sai số mô hình, tuy nhiên, ta mong muốn rằng sự khác nhau này rất nhỏ.

Chú ý 2: Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán.

***Phân tích toán học***

a) Dạng của Linear Regression

Trong phương trình (1) phía trên, nếu ta đặt là vector (cột) hệ số cần phải tối ưu và  là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:



b) Sai số dự đoán

Ta mong muốn rằng sự sai khác  giữa giá trị thực  và giá trị dự đoán  là nhỏ nhất. Nói cách khác, mục đích là làm cho giá trị sau đây càng nhỏ càng tốt:



trong đó hệ số là để thuận tiện cho việc tính toán (khi tính đạo hàm thì số sẽ bị triệt tiêu). ở đây ta cần  vì có thể là một số âm, việc nói  nhỏ nhất sẽ không đúng vì khi  là rất nhỏ nhưng sự sai lệch là rất lớn.

c) Hàm mất mát

Điều tương tự xảy ra với tất cả các cặp (input, outcome) , với  là số lượng dữ liệu quan sát được. Điều ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm  để hàm số sau đạt giá trị nhỏ nhất:

 (2)

Hàm số được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số  sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của  làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (Optimal point), ký hiệu:



Trước khi đi tìm lời giải, ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt  là một vector cột chứa tất cả các output của training data là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát  được viết dưới dạng ma trận đơn giản hơn:

 (3)

với là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác là tổng của bình phương mỗi phần tử của vector . Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình (3).

d) Nghiệm cho bài toán Linear Regression

**Cách phổ biến nhất để tìm nghiệm cho một bài toán tối là giải phương trình đạo hàm (gradient) bằng 0!** Tất nhiên đó là khi việc tính đạo hàm và việc giải phương trình đạo hàm bằng 0 không quá phức tạp. Thật may mắn, với các mô hình tuyến tính, hai việc này là khả thi.

Đạo hàm theo của hàm mất mát là:



Phương trình đạo hàm bằng 0 tương đương với:

   (4)

Đặt 

Nếu ma trận vuông  khả nghịch (non-singular hay invertible) thì phương trình (4) có nghiệm duy nhất: .

## Tensorflow

Nếu hiểu theo cách đơn giản thì Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lí tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Tensorflow được sử dụng khi ta cần giải quyết các bài toán Supervised Learning [3].

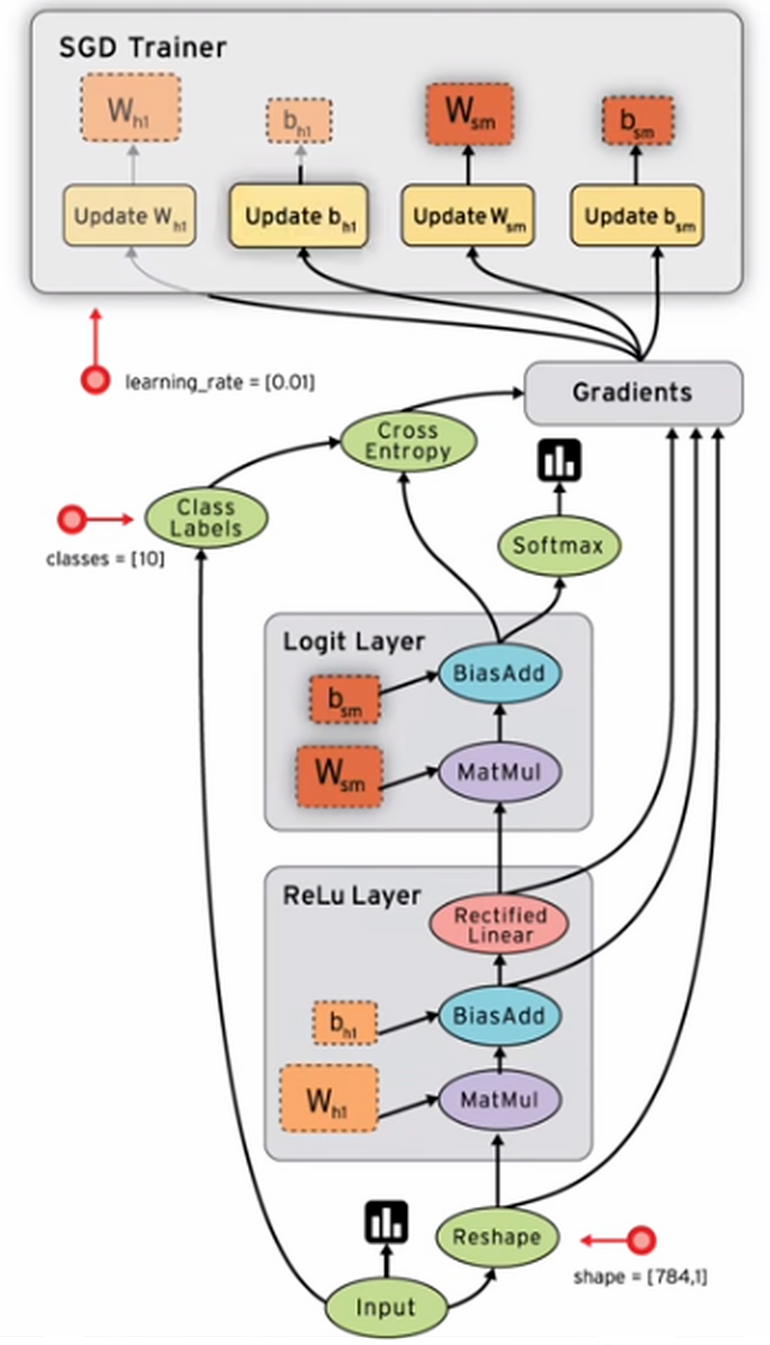
Khi thực hành với Tensorflow, sẽ có rất nhiều khái niệm phức tạp. Tuy nhiên chỉ ở mức cơ bản, ta sẽ đi đến khái niệm quan trọng nhất trong Tensorflow là Tensor.

1. Node

Vì Tensorflow mô tả lại dòng chảy của dữ liệu thông qua graph nên mỗi một điểm giao cắt trong graph thì được gọi là Node. Tại sao điều này quan trọng thì là vì các Node chính là điểm đại diện cho việc thay đổi của dữ liệu nên việc lưu trữ lại tham chiếu của các Node này là rất quan trọng.

1. Tensor

Để giải được các bài toán Machine Learning, cần phải làm cho máy tính có thể hiểu được dữ liệu của tập nguồn và dữ liệu của tập đích. Tensorflow cung cấp một loại dữ liệu mới được gọi là Tensor. Trong thế giới của Tensorflow, mọi kiểu dữ liệu đều được quy về một mối được gọi là Tensor hay trong Tensorflow, tất cả các loại dữ liệu đều là Tensor. Vậy nên có thể hiểu được phần nào cái tên Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor.



Hình 2‑1. Ví dụ về một graph trong Tensorflow

Tensor là một kiểu dữ liệu dạng mảng có nhiều chiều được mô tả dạng Tensor = [[[1,1,1],[178,62,74]],[[45,2,2],[19,0,17]],[[7,5,2],[0,11,4]],[[8,13,5],[1,6,7]]]. Mảng nhiều chiều này được đính kèm thêm một vài thuộc tính tham chiếu khác. Các thuộc tính của Tensor được mô tả trong tài liệu bao gồm:

• device: Tên của thiết bị mà Tensor hiện tại sẽ được xuất bản. Có thể None.

• graph: Đồ thị chứa Tensor hiện tại.

• name: Tên của Tensor hiện tại.

• shape: Trả về TensorShape mô tả lại Shape của Tensor hiện tại.

• op: operation - toán tử (phép toán) được sử dụng để xuất bản Tensor hiện tại.

• dtype: Kiểu của các phần tử (elements) trong Tensor hiện tại.

1. Rank

Rank là bậc hay độ sâu của một Tensor. Ví dụ như Tensor = [1] sẽ có rank là 1, Tensor = [[[1,1,1],[178,62,74]]] sẽ có rank bằng 3, Tensor = [[1,1,1],[178,62,74]] sẽ có rank bằng 2. Cách nhanh nhất để xác định rank của một Tensor là đếm số lần mở ngoặc vuông cho đến giá trị khác ngoặc vuông đầu tiên. Việc phân rank này khá quan trọng vì nó đồng thời cũng giúp phân loại dữ liệu của Tensor. Khi ở cách rank đặc biệt cụ thể, Tensor có những tên gọi riêng như sau:

• Scalar: Khi Tensor có rank bằng 0, Tensor đại diện cho một số hoặc một chuỗi cụ thể. Ví dụ: scalar=123.

• Vector: Vector là một Tensor rank 1. Trong python thì Vector là một list hay mảng một chiều chứa các số. Ví dụ: list=[123,345].

• Matrix: Đây là một Tensor rank 2 hay mảng hai chiều theo khái niệm của Python. Ví dụ: matrix=[[1,2],[2,1]].

• N-Tensor: Khi rank của Tensor tăng lên lớn hơn 2, chúng được gọi chung là N-Tensor.

1. Shape

Shape là một tuple có số chiều (dimention) bằng với rank của Tensor tương ứng dùng để mô tả lại cấu trúc của Tensor đó. Dưới đây là ví dụ về Shape.

• Tensor = 1 sẽ có Shape = ().

• Tensor = [1] sẽ có Shape = (1).

• Tensor = [[[1,1,1],[178,62,74]]] sẽ có Shape = (1,1,3).

• Tensor = [[1,1,1],[178,62,74]] sẽ có Shape = (1,3).

Dựa vào cấu trúc của Shape, ta dễ dàng thấy rằng ràng buộc cơ bản của Tensor là chiều của các elements trong Tensor tại mỗi bậc phải bằng nhau.

1. Op

Được viết tắt là op, khái niệm Operator là toán tử được dùng để thực thi Tensor tại node đó. Các toán tử này có thể là Const (hằng số), Variable (biến số), Add (phép cộng), Mul (phép nhân)… Có thể nói, khái niệm operator trong Tensorflow là khái niệm dùng để mô tả lại trạng thái của Node nói chung.

1. DType

Đây là kiểu dữ liệu của các elements trong Tensor. Vì một Tensor chỉ có duy nhất một thuộc tính DType nên từ đó cũng suy ra là chỉ có duy nhất một kiểu DType duy nhất cho toàn bộ các elements có trong Tensor hiện tại. Việc tạo ra hơn một DType khác nhau cho các elements của Tensor là không khả dụng.

1. Placeholder

Đối với thuật toán Linear Regression, đồ thị mong muốn của ta là một hàm bậc nhất một ẩn có dạng y = wx + b với x là dữ liệu tập nguồn, y là dữ liệu tập đích, w là trọng số (weight) của x và b là độ lệch (bias) của hàm mong muốn. Để xây dựng được hàm dưới dạng graph, ta không thể chỉ dùng các Node cố định mà Tensorflow mong muốn giá trị của x và y có thể thay đổi được trong quá trình Session yêu cầu Worker làm việc.

Khi một Node có Operator là placeholder, chúng không cần giá trị cụ thể mà chỉ yêu cầu một kiểu DType đặt trước. Giá trị của Node sẽ được xác định mỗi lần Worker chạy qua chúng thông qua một tập dữ liệu kiểu Dictionaries (từ điển) mà điển hình là Array (mảng).

## Tổng quan về Network và Internet

### Giao thức TCP/IP

Bộ giao thức TCP/IP (Internet Protocol Suite) là bộ giao thức mà Internet và các mạng máy tính đang sử dụng và chạy trên đó. Nó gồm 2 giao thức chính là TCP (Transmission Control Protocol - Giao thức điều khiển giao vận) và IP (Internet Protocol - Giao thức liên mạng).

Bộ giao thức TCP/IP được coi là một tập hợp các tầng. Mỗi tầng giải quyết một tập các vấn đề liên quan đến việc truyền tải dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Hay nói cách khác, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể truyền đi một cách vật lý.

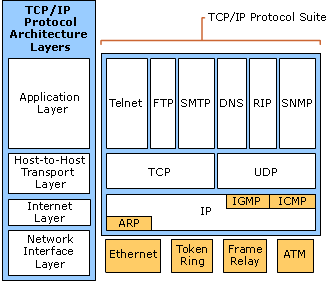
4 tầng của TCP/IP

• Tầng ứng dụng - Application Layer

• Tầng giao vận - Transport Layer

• Tầng mạng - Internet Layer

• Tầng liên kết - Link Layer



Hình 2‑2. Mô hình 4 tầng của TCP / IP

### Giao thức HTTP

HTTP là viết tắt của HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản. Giao thức này nằm trong tầng Application Layer, được sử dụng để truyền nội dung trang web từ Webserver đến trình duyệt Web ở Client. Là giao thức Client/Server dùng cho Internet - World Wide Web, HTTP thuộc tầng ứng dụng của bộ giao thức TCP/IP (Các giao thức nền tảng cho Internet). Cơ chế hoạt động chính của HTTP là Request-Response: Web Client sẽ gửi Request đến Web Server, Web Server xử lý và trả về Response cho Web Client.

Hai phương thức được sử dụng nhiều nhất trong HTTP request là GET và POST

Với GET, câu truy vấn sẽ được đính kèm vào đường dẫn của HTTP request. Ví dụ: /?username=“abc”&password=“def”

Một số đặc điểm của phương thức GET:

* GET request có thể được cached, bookmark và lưu trong lịch sử của trình duyệt.
* GET request bị giới hạn về chiều dài, do chiều dài của URL là có hạn.
* GET request không nên dùng với dữ liệu quan trọng, chỉ dùng để nhận dữ liệu.

Ngược lại, với POST thì câu truy vấn sẽ được gửi trong phần message body của HTTP request, một số đặc điểm của POST như:

* POST không thể cached, bookmark hay lưu trong lịch sử trình duyệt.
* POST không bị giới hạn về độ dài.

### Ngôn ngữ PHP

PHP là viết tắt của Hypertext Preprocessor là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

JSON (JavaScript Object Notation) là một kiểu dữ liệu mở trong JavaScript. Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị". Về cấu trúc, nó mô tả một vật thể bằng cách bọc những vật thể con trong vật thể lớn hơn trong dấu ngoặc nhọn. JSON là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình.

### Cơ sở dữ liệu Database

Cơ sở dữ liệu (database) là một tập hợp những thông tin được tổ chức để dễ dàng trong việc tạo lập, cập nhập và khai thác thông tin. Cơ sở dữ liệu được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành hay được lưu trữ trong các hệ quản trị cơ sở dữ liệu.

SQL (Structured Query Language) hay ngôn ngữ truy vấn mang tính cấu trúc, là một loại ngôn ngữ máy tính phổ biến để tạo, sửa, và lấy dữ liệu từ một hệ quản trị cơ sở dữ liệu quan hệ. Ngôn ngữ này phát triển là để phục vụ các hệ quản trị cơ sở dữ liệu đối tượng-quan hệ.

SQL là cần thiết để:

• Tạo cơ sở dữ liệu, bảng và view mới.

• Để chèn các bản ghi vào trong một cơ sở dữ liệu.

• Để xóa các bản ghi từ một cơ sở dữ liệu.

• Để lấy dữ liệu từ một cơ sở dữ liệu.

Chức năng của SQL:

• Với SQL, chúng ta có thể truy vấn Database theo nhiều cách khác nhau, bởi sử dụng các lệnh.

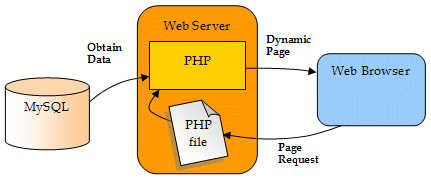
• SQL cho phép người dùng miêu tả dữ liệu.

• SQL cho phép người dùng định nghĩa dữ liệu trong một Database và thao tác nó khi cần thiết.

• Cho phép người dùng tạo, xóa Database và bảng.

• Cho phép người dùng tạo view, Procedure, hàm trong một Database.

• Cho phép người dùng thiết lập quyền truy cập vào bảng, thủ tục và view.



Hình 2‑3. Mô tả cách hoạt động của Webserver dựa trên database MySQL và các tệp PHP

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## Yêu cầu thiết kế

Thiết kế mô hình thu thập thông tin từ môi trường và điều khiển thiết bị gồm có:

Hệ thống các cảm biến gồm:

* DHT22 đo đạc các tín hiệu nhiệt độ - độ ẩm tương đối trong không khí
* Cảm biến độ ẩm đất đo đạc tín hiệu độ ẩm tương đối trong đất
* Cảm biến ánh sáng đo đạc tín hiệu về cường độ ánh sáng mặt trời
* Module GPS NEO-6M thu thập vị trí khối cảm biến

Các thiết bị được điều khiển thông qua relay gồm có:

* Máy bơm dùng để tưới nước cho cây trồng
* Đèn chiếu sáng dùng để cung cấp thêm ánh sáng cho cây trồng

Thông số đo đạc tức thời hiển thị qua màn hình LCD và dữ liệu được đưa lên và lưu trữ trên mạng. Các thiết bị được điều khiển bằng nút nhấn trực tiếp bằng tay hoặc thông qua Website hoặc ứng dụng trên Smartphone.

## Phân tích thiết kế

Bảng 3‑1. Phân tích các lựa chọn về phần cứng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Components** | Processor /  Microcontroller | Wireless transceiver circuit | Photosensor | Temperature and humidity sensor | Display |
| **Option 1** | PIC16Fxxx | Bluetooth  HC-05 | Photomultiplier tube | AM2301 | Graphic LCD  128x64 |
| **Mô tả** | Dễ sử dụng, phổ biến | Không thể giao tiếp với Webserver | Giá thành đắt | Giá thành đắt | Giá thành đắt |
| **Option 2** | Arduino Uno R3 | ESP8266 | Photoresistor (CDS) | DHT11 | Text LCD  LCD 2004 |
| **Mô tả** | Dễ sử dụng, phổ biến | Dể sử dụng, phổ biến | Giá thành rẻ | Giá thành rẻ | Giá thành rẻ |
| **Final Option** | ARM TM4C123G | ESP32 | Photodiode | DHT22 | Text LCD  LCD 1602 |
| **Mô tả** | Có sẵn | Cải tiến hơn ESP8266 | Có sẵn | Chính xác hơn DHT11 | Có sẵn |

### Tiva C Series TM4C123G LaunchPad

Kit vi tính LaunchPad TM4C123G là một nền tảng vi tính chi phí thấp cho các bộ vi điều khiển dựa trên ARM Cortex-M4F của Texas Cụ. Thiết kế của LaunchPad TM4C123G làm nổi bật bộ vi điều khiển TM4C123GH6PM với giao diện thiết bị USB 2.0 và mô-đun ngủ đông. EK-TM4C123GXL cũng có các nút người dùng có thể lập trình và đèn LED RGB cho các ứng dụng tùy chỉnh. Các tiêu đề có thể xếp chồng lên nhau của Giao diện LaunchPad BoosterPack ™ XL của TM4C123G giúp dễ dàng và đơn giản để mở rộng chức năng của LaunchPad TM4C123G khi giao tiếp với các thiết bị ngoại vi khác với TexasU MCU BoosterPack.



Hình 3‑1. Tiva C Series TM4C123G LaunchPad

Tính năng, đặc điểm: Bộ công cụ đánh giá LaunchPad MCU TM4C123G dựa trên ARM Cortex-M4F (EK-TM4C123GXL) cung cấp các tính năng sau:

- MCU hiệu suất cao TM4C123GH6PM:

• CPU vi điều khiển dựa trên ARM Cortex-M4 80-bit 32-bit

• Flash 256KB, SRAM 32KB, EEPROM 2KB

• Hai mô-đun Mạng điều khiển (CAN)

• Máy chủ / Thiết bị USB 2.0 / OTG + PHY

• ADC 2MSPS 12 bit kép, bộ điều khiển chuyển động PWM

• 8 UART, 6 I2C, 4 SPI

- Giao diện gỡ lỗi trên bo mạch (ICDI)

- USB Micro-B cắm vào cáp cắm USB-A

- Ứng dụng khởi động nhanh RGB được cài sẵn

- Hướng dẫn dễ sử dụng

- EK-TM4C123GXL TivaWare ™ cho phần mềm C Series có sẵn để tải xuống miễn phí

### Module Wifi ESP32



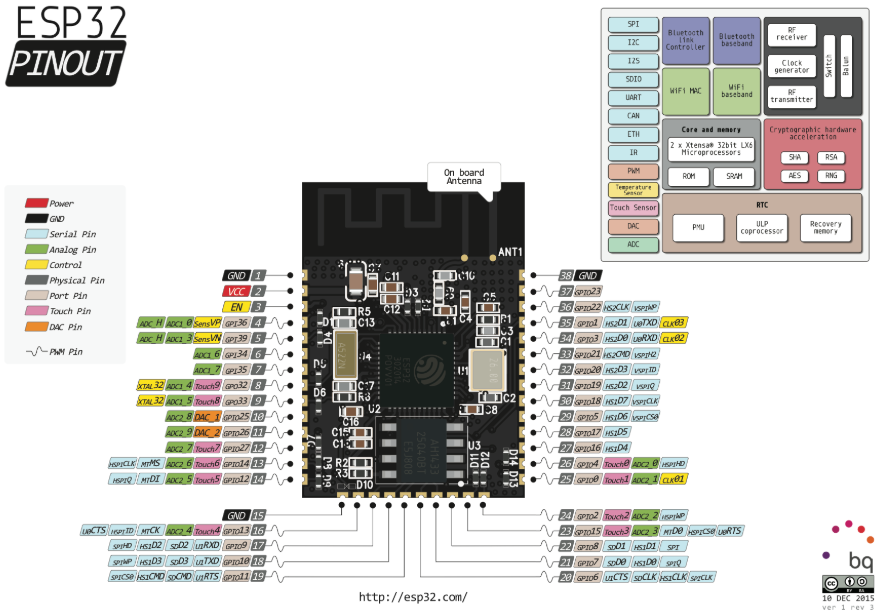
Hình 3‑2. Ảnh thực tế của ESP32

ESP32 là module wifi giá rẻ và được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác.

ESP32 là một chip tích hợp cao, được thiết kế cho nhu cầu của một thế giới kết nối mới, thế giới Internet of thing (IoT). Nó cung cấp một giải pháp kết nối mạng Wi-Fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

ESP32 có xử lý và khả năng lưu trữ mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với một chi phí tối thiểu và một PCB tối thiểu. Sử dụng giao thức nối tiếp với tốc độ Baud là 9600(mặc định). Tốc độ baud có thể lên đến 115200.

ESP32 là module sử dụng các tập lệnh AT. Tuy nhiên đã có thêm nhiều phiên bản firmware hỗ trợ nhiều ngôn ngữ, trong đó có LUA và Arduino. Chính vì thế có thể dùng bất kì board TivaC nào để điều khiển thu phát thông qua phương thức truyền nối tiếp.



Hình 3‑3. Sơ đồ ra chân của ESP32

Chức năng các chân cơ bản:

- Vcc: 3.3V.

- GND: 0V.

- TX: chân TX của giao thức UART, kết nối đến chân RX của vi điều khiển.

- RX: chân RX của giao thức UART, kết nối đến chân TX của vi điều khiển.

- RST: chân reset kéo xuống mass để bật reset.

- CH-PD: chân này kéo lên mức cao module bắt đầu thu phát, kéo xuống thấp module dừng. Vì ESP32 khởi động hút dòng lớn nên phải để mức thấp khi khởi động xong sẽ kéo lên mức cao.

- GPIO0: kéo xuống thấp cho chế độ upgrade firmwave.

Chức năng:

• WiFi ESP32, Bluetooth LE SoC - 240Mhz - Mô-đun ESP-WROOM-32

• Tự động chọn 3 nguồn năng lượng (DC6-28V, USB và Pin)

• Tự động tải xuống chế độ Flash

• Khe cắm SDCARD tích hợp (hỗ trợ chế độ 1 bit)

• Thiết kế phần cứng mở với giấy phép KiCad, CC-BY-SA.

• Tiêu đề hiển thị OLED I2C

• Bộ sạc pin lithium-Ion

• 1 nút đặt lại, 1 nút lập trình

• 1 đèn LED nguồn, 1 đèn LED lập trình, 1 đèn LED sạc

• Tương thích với Shields cho ESP32 trong tương lai (Cổng - GSM / GPRS / GPS và Lora Shield, Kết nối - CAN, RS485, Shield Shield, Audio Shield, ...)

Cấu hình kết nối: Sau khi cài đặt xong và kết nối, sẽ xuất hiện cổng COM ảo trên máy tính (Tùy từng loại hệ điều hành mà có những tên cổng như: COM1, COM2 ... đối với Windows, /dev/tty/wchusbserial1420 trên Mac OS), /dev/ttyUSB0 trên Ubuntu Linux)

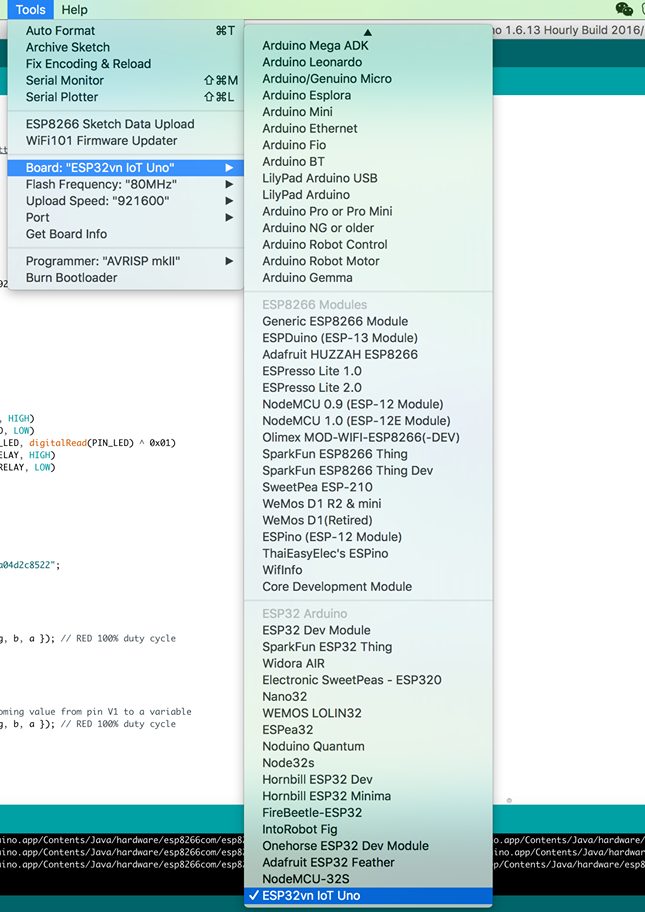
Mở Arduino IDE và lựa chọn board (tham khảo cấu hình kết nối như hình dưới):

Board: ESP32vn IoT Uno

Flash Frequency: 80MHz

Port: chọn cổng khi gắn thiết bị vào sẽ thấy xuất hiện

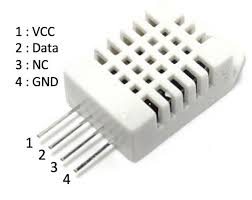
Upload speed: Chọn cao nhất, nếu nạp không được chọn thấp dần



Hình 3‑4. Cấu hình kết nối ESP32 trên Arduino IDE

### Các cảm biến

a) Cảm biến nhiệt độ - độ ẩm DHT22



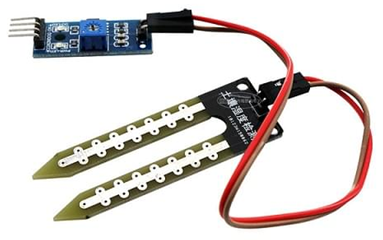
Hình 3‑5. Cảm biến DHT22

Cảm biến độ ẩm và nhiệt độ DHT22 sử dụng giao tiếp 1 Wire dễ dàng kết nối và giao tiếp với Vi điều khiển để thực hiện các ứng dụng đo nhiệt độ, độ ẩm môi trường, cảm biến có chất lượng tốt, kích thước nhỏ gọn, độ bền và độ ổn định cao.

Thông số kỹ thuật:

* Nguồn sử dụng: 3~5VDC.
* Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
* Đo tốt ở độ ẩm 0100%RH với sai số 2-5%.
* Đo tốt ở nhiệt độ -40 to 80°C sai số ±0.5°C.
* Tần số lấy mẫu tối đa 0.5Hz (2 giây 1 lần)
* Kích thước 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
* 4 chân, khoảng cách chân 0.1''

b) Cảm biến độ ẩm đất



Hình 3‑6. Cảm biến độ ẩm đất

Cảm biến độ ẩm đất thường được sử dụng trong các mô hình tưới nước tự động, vườn thông minh… cảm biến giúp xác định độ ẩm của đất qua đầu dò và trả về giá trị Analog, Digital qua 2 chân tương ứng để giao tiếp với Vi điều khiển để thực hiện vô số các ứng dụng khác nhau.

Thông số kỹ thuật:

- Điện áp hoạt động: 3.3~5VDC

- Tín hiệu đầu ra:

* Analog: theo điện áp cấp nguồn tương ứng.
* Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trở thông qua mạch so sánh LM393 tích hợp.

- Kích thước: 3 x 1.6cm.

Sơ đồ chân:

* VCC: Nguồn 3V~5V
* GND: GND của nguồn ngoài
* DO: Ngõ ra tín hiệu số Digital (mức cao hoặc mức thấp)
* AO: Ngõ ra tín hiệu tương tự Analog

c) Cảm biến ánh sáng



Hình 3‑7. Cảm biến ánh sáng photodiode

Cảm biến ánh sáng Photodiod sử dụng Photodiod thay vì quang trở để cảm biến ánh sáng nên cho độ chính xác và độ ổn định cao, cảm biến có Opamp tích hợp cho cả hai ngõ ra tín hiệu là Digital và Analog rất dễ sử dụng, thích hợp cho các ứng dụng nhận biết sáng tối, cảm biến cường độ sáng…

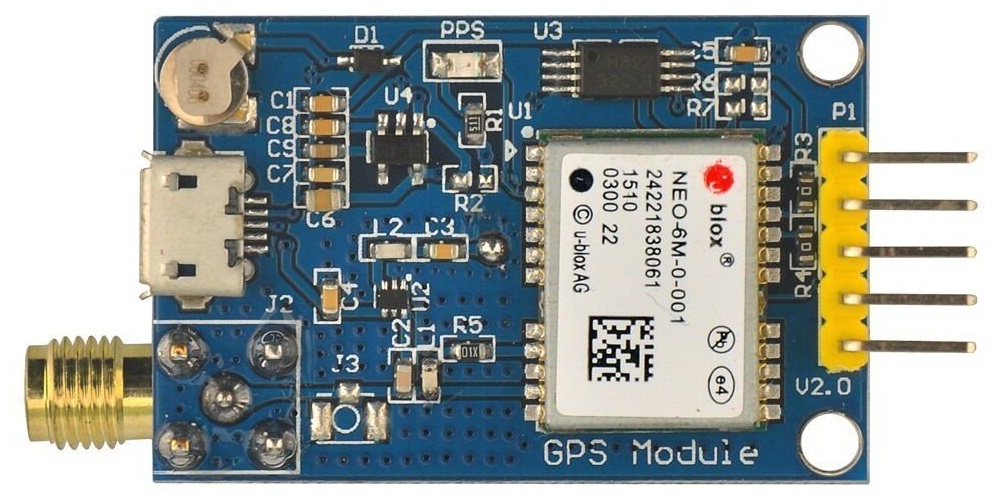
Thông số kỹ thuật:

* Nguồn: 3.3 -> 5VDC
* Sử dụng Photodiod cho độ chính xác cao.
* Xuất tín hiệu Digital tinh chỉnh bằng biến trở trên mạch hoặc Analog rất dễ sử dụng.
* Kích thước: 30 x 16mm

Sơ đồ chân:

* VCC: Nguồn 3V~5V
* GND: GND của nguồn ngoài
* DO: Ngõ ra tín hiệu số Digital (mức cao hoặc mức thấp)
* AO: Ngõ ra tín hiệu tương tự Analog

d) Module NEO-6M GPS



Hình 3‑8. Module NEO-6M GPS

Module NEO-6M GPS là module định vị toàn cầu sử dụng hệ thống vệ tinh GPS của Mỹ. Module GPS NEO-6M cho tốc độ xác định vị trí nhanh và chính xác, có nhiều mức năng lượng hoạt động, phù hợp với các ứng dụng chạy pin. Anten GPS trên mạch có 2 chuẩn kết nối là ipex và SMA, thích hợp với các ứng dụng định vị vị trí hoặc thời gian bằng GPS. Mạch GPS U-Blox NEO-6M có độ bền cao, phần cứng được thiết kế chất lượng tốt, sử dụng module chính hãng U-Blox cho độ bắt sóng mạnh và độ ổn định cao nhất.

Đặc điểm kĩ thuật:

* Nguồn hoạt động: 3.3-5.5V (Nên sử dụng ở mức 3.3V)
* Dòng hoạt động bình thường: 50 mA
* Dòng hoạt động ở trạng thái tiết kiệm: 30 mA
* Giao tiếp UART/TTL (Để sử dụng giao tiếp UART bạn có thể sử dụng module UART TTL CP2102)
* Baud rate: Gồm nhiều mức khác nhau 1200, 2400, 4800, 19200, 38400, 9600 (mặc định), 57600, 115200…
* Kích cỡ module: 39\*25.5mm

Sơ đồ chân:

* VCC: Nguồn 3V~5V
* GND: GND của nguồn ngoài
* TX: Pin truyền dữ liệu
* RX: Pin nhận dữ liệu
* PPS: Sau gần 1s sẽ xuất 1 xung ra (có thể kết nối với led)

Đi kèm là Anten GPS V1 có độ nhận sóng cao kể cả ở những điều kiện thời tiết không tốt. Anten có khả năng chống chịu nước, dây dẫn sóng jack nối SMA đực dài 3m giúp cho việc bắt nối trở nên linh động và dễ dàng, thích hợp để gắn lên xe và ở điều khiện ngoài trời.



Hình 3‑9. GPS Antenna

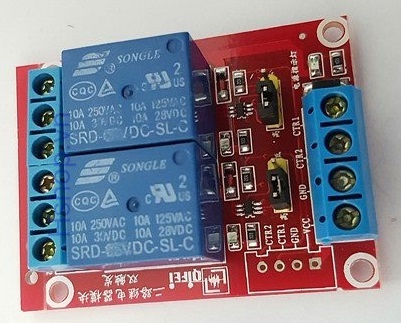
Thông số kỹ thuật:

* Tần số: 1575.42mhz±3mhz
* Tỉ số sóng đứng điện áp (VSWR): 1.5: 1
* Băng thông: ± 5 mhz
* Trở kháng: 50 Ohm
* Loại phân cực: RHCP
* Bộ khuếch đại: 28±2db Typical
* Tiếng ồn: 1.5 db
* Điện áp: 2.2~5 V
* Dòng tiêu thụ: 5~15ma
* Cáp: RG174 3M
* Đầu kết nối: SMA

Lưu ý: Để anten hoạt động tốt nhất không để anten trong các hộp kín bằng kim loại.

### Các thiết bị ngõ ra

a) Module relay



Hình 3‑10. Module relay 2 kênh

Module 2 Relay kích H/L (5VDC) sử dụng nguồn 5VDC để nuôi mạch, tín hiệu kích có thể tùy chọn kích mức cao (High - 5VDC) hoặc mức thấp (Low - 0VDC) qua Jumper trên mỗi relay. Thích hợp cho các thiết bị sử dụng mức tín hiệu 5VDC như Vi điều khiển. Ngoài ra có thể chọn các module có số lượng relay tùy ý phụ thuộc vào số lượng thiết bị sử dụng.

Thông số kỹ thuật sản phẩm:

* Điện áp nuôi mạch: 5VDC.
* Dòng tiêu thụ: khoảng 200mA/1Relay
* Tín hiệu kích: High (5VDC) hoặc Low (0VDC) chọn bằng Jumper.
* Relay trên mạch:

- Nguồn nuôi: 5VDC.

- Tiếp điểm đóng ngắt max: 100W-10A

* Kích thước: 52 (L) \* 41(W) \* 19 (H) mm.

b) Động cơ bơm nước

Động cơ bơm P385 12VDC 3W có kích thước nhỏ gọn, được sử dụng để bơm nước, dung dịch với khả năng bơm tối đa lên đến 1.8L / 1 phút, động cơ sử dụng điện áp 12VDC, thích hợp với các thiết kế sử dụng máy bơm nhỏ: bơm hồ cá, tưới nước cho cây,..., lưu ý không cấp ngược cực vì có thể làm hư cơ cấu bơm của động cơ (cực dương có đánh dấu màu đỏ).

Thông số kĩ thuật:

* Loại động cơ DC: 385
* Điện áp sử dụng: 12VDC
* Dòng điện sử dụng: 0.25A
* Công suất: 3W
* Lưu lượng bơm: 1,8 ± 0,1 L/phút
* Áp suất nước: 0.3Mpa
* Thời gian làm việc liên tục tối đa trong 1 ngày: không quá 8h
* Đường kính ngoài ống dẫn: 7mm
* Kích thước: 86 x 46 x 46 mm



Hình 3‑11. Động Cơ Bơm P385 12VDC 3W

c) Đèn chiếu sáng

Ở đây ta có thể sử dụng các loại đèn Compact hoặc đèn Led tùy vào nhu cầu chiếu sáng. Đồng thời những lựa chọn kể trên cũng hướng đến mục đích tiết kiệm điện năng tiêu thụ so với đèn dây tóc thông thường.



Hình 3‑12. Đèn Compact

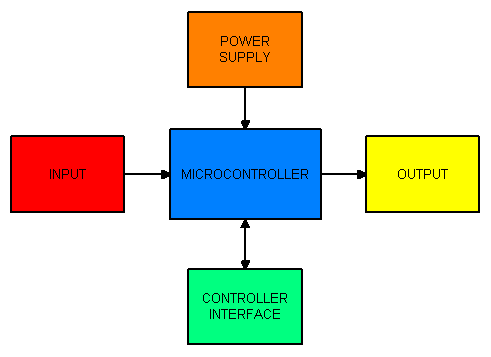


Hình 3‑13. Đèn Led chiếu sáng

## Sơ đồ khối

Xây dựng sơ đồ khối tổng quát gồm 5 khối chính:

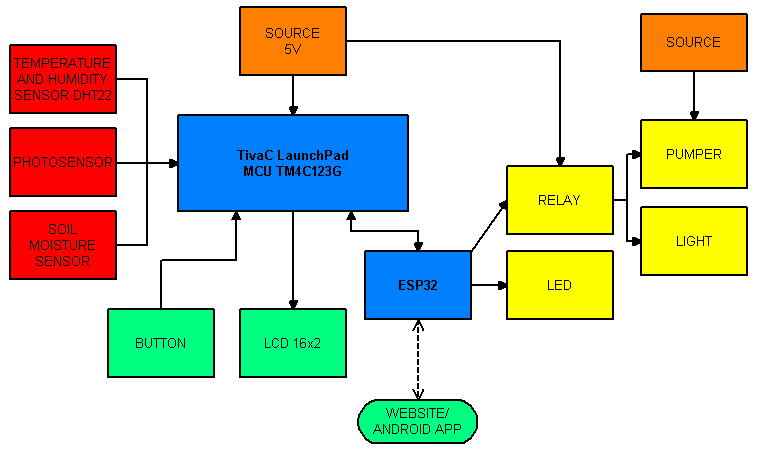
* Khối xử lý trung tâm
* POWER SUPPLY: Khối cấp nguồn
* INPUT: Khối các thiết bị ngõ vào
* OUTPUT: Khối các thiết bị ngõ ra
* CONTROLLER INTERFACE: Khối giao diện điều khiển



Hình 3‑14. Sơ đồ khối phần cứng tổng quát

Xây dựng sơ đồ khối chi tiết bao gồm các khối nhỏ hơn nằm trong 5 khối chính:

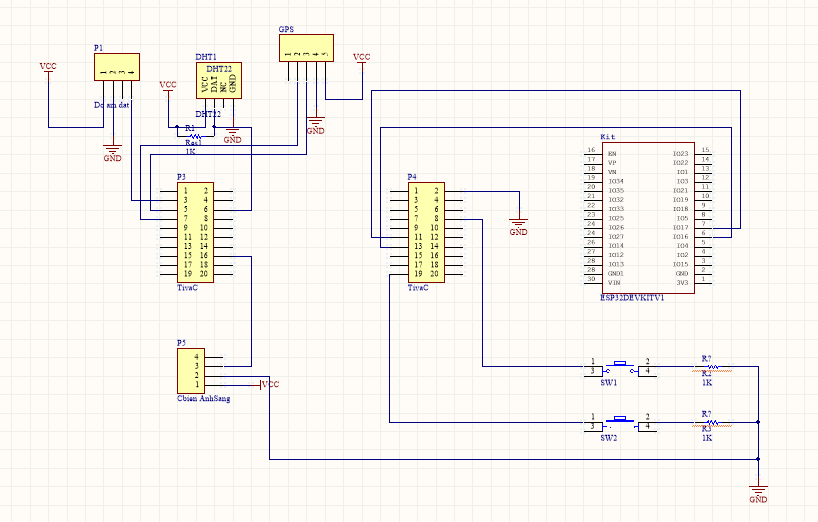
* Khối xử lý trung tâm gồm có Tiva C LaunchPad và Module Wifi ESP32.
* Khối cấp nguồn gồm các module cấp nguồn 3.3V, 5V, 9V.
* Khối các thiết bị ngõ vào gồm có các cảm biến: DHT22, cảm biến độ ẩm đất, cảm biến cường độ sáng và module GPS NEO-6M.
* Khối các thiết bị ngõ ra gồm có các đèn LED, relay 2 kênh điều khiển máy bơm và đèn chiếu sáng.
* Khối giao diện điều khiển gồm có các nút nhấn (thường mở) và màn hình LCD 1602.



Hình 3‑15. Sơ đồ khối phần cứng chi tiết

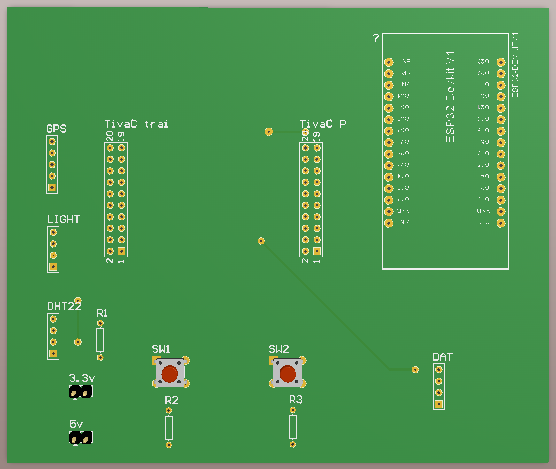
## Sơ đồ mạch chi tiết

Ta xây dựng sơ đồ mạch chi tiết bằng việc thực hiện vẽ sơ đồ nguyên lý bằng phần mềm Altium.

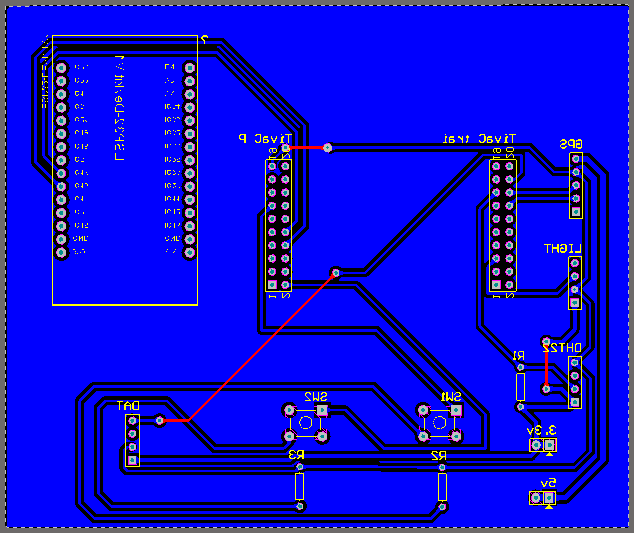


Hình 3‑16. Sơ đồ nguyên lý của mạch vẽ bằng phần mềm Altium

Kế đến ta thiết kế layout cho PCB cũng bằng phần mềm Altium. Vị trí kit Tiva C LaunchPad, Module Wifi ESP32, DHT22, các module cảm biến, các header, nút nhấn và đường mạch đồng được quyết định ở bản thiết kế này:



Hình 3‑17. Mặt trên của bản mạch



Hình 3‑18. Mặt dưới của bản mạch

# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

## Thực hiện dự đoán bằng máy học

### Chuẩn bị dữ liệu

Dữ liệu là một yếu tố bắt buộc trong tất cả các bài toán Machine Learning. Với bài toán dự báo thời tiết, ta có thể download dữ liệu thời tiết Thành phố Hồ Chí Minh tại website https://www.meteoblue.com/en/weather/forecast/week/ho-chi-minh-city\_vietnam\_1566083. Trang này cung cấp đầy đủ các thông tin về lượng mưa, độ ẩm, nhiệt độ, nắng gió... Ở đây ta sẽ thu thập dữ liệu về Temperature (Nhiệt độ), Humidity (Độ ẩm) và Solar Radiation (Bức xạ ánh sáng) được chứa trong một tệp tin đuôi .csv (có thể mở bằng chương trình Excel). Vì Linear Regression rất nhạy cảm với nhiễu (sensitive to noise) nên ta phải trải qua bước loại bỏ nhiễu (outlier) hay còn gọi là tiền xử lý (pre-processing). Với dữ liệu được tải về trên website thời tiết như trên thì ta sẽ sử dụng luôn mà không cần loại bỏ nhiễu.

### Xử lý dữ liệu

Để triển khai thuật toán Linear Regression bằng Tensorflow trên Python, ta nhập các thư viện sẽ được dùng [4]:

import tensorflow as tf

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from IPython import get\_ipython

ipy = get\_ipython()

if ipy is not None:

ipy.run\_line\_magic('matplotlib','inline')

Tải dữ liệu dùng để training:

file\_path = 'weather.csv'

//Tên file csv chứa dữ liệu

data=pd.read\_csv(file\_path, delimiter=',',header=0,skipinitialspace=True)

//header chỉ vị trí của hàng bắt đầu lấy dữ liệu

data.head(24)

Như vậy ta đã có được file dữ liệu training, bây giờ chỉ cần gọi data['Temperature'] hoặc data['Wind\_Speed'] ... để sử dụng:

temperature = np.array(data['Temperature'])

Ta có 1 chuỗi các giá trị array, các số tiếp theo sẽ phụ thuộc vào các số trước đó, cũng như vậy ta sẽ có các giá trị thời tiết tại thời điểm trước đó sẽ là X và giá trị thời tiết tại thời điểm trên sau 1 giờ sẽ là Y. Ví dụ như bảng dữ liệu nhiệt độ theo giờ trong 2 ngày 3/4/2019 và 4/4/2019:

Bảng 4‑1. Mảng dữ liệu dùng để training

|  |  |
| --- | --- |
| X | Y |
| 26.08 | 25.71 |
| 25.71 | 25.32 |
| 25.32 | 25.05 |
| 25.05 | 25.8 |
| 25.8 | 25.73 |
| 25.73 | 26.23 |
| 26.23 | 27.81 |
| 27.81 | 28.49 |
| 28.49 | 31.07 |
| 31.07 | 31.25 |
| 31.25 | 31.46 |
| 31.46 | 32.18 |
| 32.18 | 32.15 |
| 32.15 | 32.17 |
| 32.17 | 32.1 |
| 32.1 | 32.92 |
| 32.92 | 31.77 |
| 31.77 | 30.83 |
| 30.83 | 29.55 |
| 29.55 | 28.63 |
| 28.63 | 27.84 |
| 27.84 | 27.13 |
| 27.13 | 26.6 |

Bài toán sẽ sử dụng 48 giá trị thu thập được để so sánh. Đặt num\_periods = 24 giờ/ngày và ta lấy dữ liệu của 2 ngày tương ứng với “num\_periods\*2”, f\_horizon chỉ khoảng chênh lệch giữa dữ liệu ngõ vào X và dữ liệu ngõ ra Y tính theo giờ:

num\_periods = 24

f\_horizon = 1

x\_data = temperature[:(len(temperature)-(num\_periods\*2))]

x\_batches = x\_data.reshape(-1, num\_periods, 1)

y\_data = temperature[1:(len(temperature)-(num\_periods\*2))+f\_horizon]

y\_batches = y\_data.reshape(-1, num\_periods, 1)

print(y\_batches.shape)

Dữ liệu đầu vào:

def test\_data(series, forecast, num):

testX = temperature[-(num + forecast):][:num].reshape(-1, num\_periods, 1)

testY = temperature[-(num):].reshape(-1, num\_periods, 1)

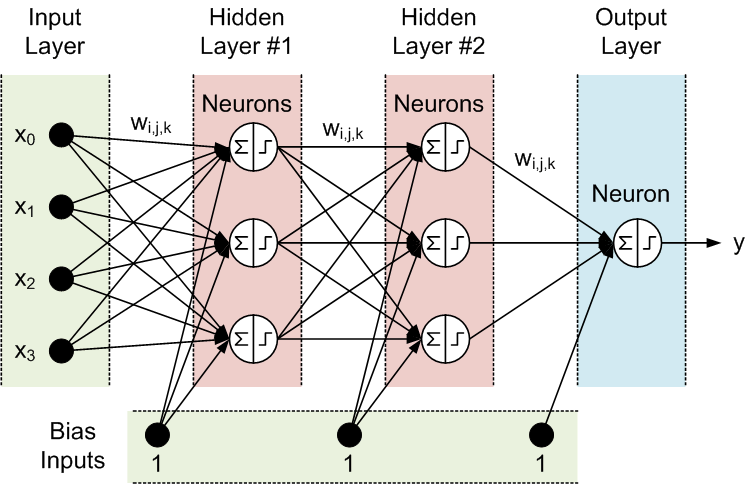
return testX, testY

X\_test, Y\_test = test\_data(temperature, f\_horizon, 24\*2)

print(X\_test.shape)

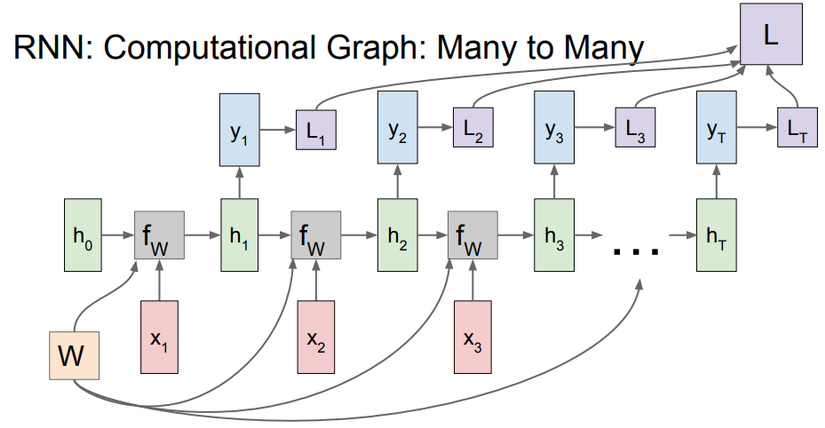
### Tiến hành training

Do đặc thù của bài toán này là dùng các thông tin trong quá khứ, để dự đoán tương lai, nên ở đây ta lựa chọn thuật toán Recurrent Neural Network (RNN) hay trong tiếng Việt gọi là Mạng Nơron hồi quy. Để có thể hiểu rõ về RNN, trước tiên ta nhìn lại mô hình Neural Network dưới đây:



Hình 4‑1. Mô tả mạng Neural Network

Neural Network bao gồm 3 phần chính là Input layer, Hidden layer và Output layer, ta có thể thấy là đầu vào và đầu ra của mạng neuron này là độc lập với nhau. Như vậy mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ... vì những dự đoán tiếp theo như thời tiết ngày hôm sau phụ thuộc vào thời tiết ngày hôm nay và những ngày trước đó. Và như vậy RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ những bước tính toán xử lý trước để dựa vào đó có thể đưa ra kết quả gần đúng nhất cho bước dự đoán hiện tại.



Hình 4‑2. Mô tả mạng Recurrent Neural Network

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào. RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trong nhiều mạng neural truyền thống khác, dữ liệu đầu vào và đầu ra hoàn toàn độc lập với nhau, tức là chúng không có liên kết thành chuỗi. Do đó khi áp dụng vào bài toán dự báo thời tiết này sẽ rất khó để đưa ra kết quả dự đoán.

Khai báo thuật toán:

tf.reset\_default\_graph()

inputs = 1

#số 1 được thêm vào dữ liệu đầu vào mở rộng để tiện cho việc tính toán

rnn\_size = 100

#kích thước của mạng RNN (số lượng các đơn vị trong mạng)

output = 1

learning\_rate=0.001

dropout\_keep\_prob = tf.placeholder(tf.float32)

X = tf.placeholder(tf.float32, [None, num\_periods, 1])

Y = tf.placeholder(tf.float32, [None, num\_periods, 1])

rnn\_cells=tf.contrib.rnn.BasicRNNCell(num\_units=rnn\_size, activation=tf.nn.relu)

rnn\_output, states = tf.nn.dynamic\_rnn(rnn\_cells, X, dtype=tf.float32)

#mô hình RNN truyền thống với hàm activation là relu

output=tf.reshape(rnn\_output, [-1, rnn\_size])

logit=tf.layers.dense(output, 1, name="softmax")

outputs=tf.reshape(logit, [-1, num\_periods, 1])

print(logit)

loss = tf.reduce\_sum(tf.square(outputs - Y))

#hàm tính toán mất mát “loss”

accuracy = tf.reduce\_mean(tf.cast(tf.equal(tf.argmax(logit, 1), tf.cast(Y, tf.int64)), tf.float32))

#hàm tính toán độ chính xác “accuracy”

optimizer = tf.train.AdamOptimizer(learning\_rate=learning\_rate)

train\_step=optimizer.minimize(loss)

#tối ưu hóa (optimizer) bằng cách tối thiểu mất mát

init=tf.global\_variables\_initializer()

Bắt đầu training:

epochs = 1000

sess = tf.Session()

init = tf.global\_variables\_initializer()

sess.run(init)

saver = tf.train.Saver()

for epoch in range(epochs):

train\_dict = {X: x\_batches, Y: y\_batches, dropout\_keep\_prob:0.5}

sess.run(train\_step, feed\_dict=train\_dict)

y\_pred=sess.run(outputs, feed\_dict={X: X\_test})

save\_path = saver.save(sess, "models/model.ckpt")

#lưu lại model để sử dụng cho lần sau

Vậy là quá trình training đã hoàn thành. Khi test dữ liệu, ta cần restore lại model vừa training để sử dụng:

with tf.Session() as sess:

# Restore variables from disk.

saver.restore(sess, "models/model.ckpt")

predict=sess.run(outputs, feed\_dict={X: X\_test})

Lưu kết quả ra file “predict.csv”

import csv

csvData = y\_predict

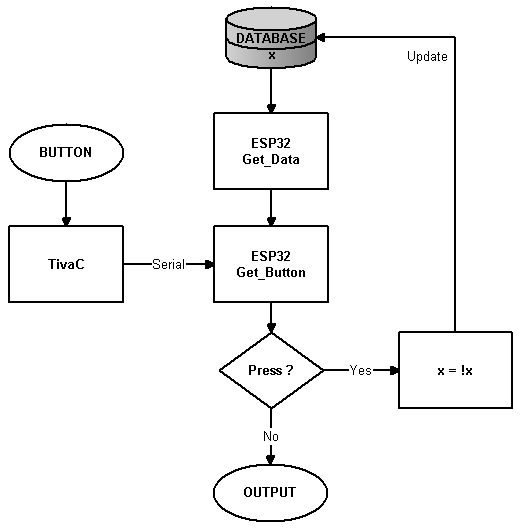
with open('predict.csv', 'w') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(csvData)

csvFile.close()

## Xây dựng chương trình cho phần cứng



Hình 4‑3. Lưu đồ giải thuật điều khiển thiết bị ngõ ra

### Chương trình nạp vào Tiva C Series TM4C123G LaunchPad

Để viết chương trình, biên dịch và nạp vào Tiva C LaunchPad, ta sử dụng phần mềm Energia. Trước tiên ta nhập vào các thư viện cần thiết, gán tên cho các chân sẽ sử dụng, xác định ngõ vào tín hiệu nút nhấn, ngõ vào các cảm biến DHT22, cảm biến độ ẩm đất, cảm biến cường độ sáng, GPS và chân xuất ra LCD:

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#include <TinyGPS.h>

#define DHTPIN PD\_0

#define PUMPW\_ON PF\_4

#define PUMPW\_PIN PF\_2

#define PUMPS\_ON PF\_0

#define PUMPS\_PIN PF\_3

#define BT1 PF\_0

#define LED1 PF\_3

#define DHTTYPE DHT22

Kết nối DHT22:

DHT | Tiva C

VCC(1) | 3.3V

DATA(2) | PD0

NC(3) | Để trống

GND(4) | GND

Nối trở 10k giữa chân (1) và chân (2)

Khởi tạo cảm biến:

int tong, anhsang,as;

String temper;

int sensorPin = PE\_2;

float flat,flon;

int pre\_dt,pre\_dt\_1;

int state,state\_1;

int pump\_stt,pump\_stt\_1;

int phannguyen\_flat,bit1\_flat,bit2\_flat,bit3\_flat;

int phannguyen\_flon,bit1\_flon,bit2\_flon,bit3\_flon;

int phannguyen\_temp,bit1\_temp,bit2\_temp,bit3\_temp;

int phannguyen\_hum,bit1\_hum,bit2\_hum,bit3\_hum;

unsigned long lastTime = 0;

bool flag = false;

DHT dht(DHTPIN, DHTTYPE);

TinyGPS gps;

LiquidCrystal\_I2C lcd(0x27, 16, 2); //Khởi tạo LCD

Tiếp theo là hàm “setup()” để cấu hình các chân input/output, các chân ngắt, tốc độ baud giao tiếp cổng COM, tốc độ baud giao tiếp Serial3 với module ESP32, kiểm tra DHT22 và LCD:

void setup() {

pinMode(PUMPW\_PIN, OUTPUT);

pinMode(PUMPS\_PIN, OUTPUT);

pinMode(LED1, OUTPUT);

pinMode(PUMPW\_ON, INPUT\_PULLUP);

pinMode(PUMPS\_ON, INPUT\_PULLUP);

pinMode(BT1, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(PF\_4),change\_status,RISING);

//Ngắt khi có xung lên

attachInterrupt(digitalPinToInterrupt(BT1),change\_status\_1,RISING);

Serial3.begin(9600);

Serial.begin(9600);

Serial1.begin(9600);

Serial.println("DHTxx test!");

dht.begin();

lcd.begin(); //Khởi tạo LCD và in thông tin ban đầu

lcd.print(" ===LUAN\_VAN===");

delay(1000);

lcd.clear();} //In xong thông tin intro thì xóa màn hình

Để xử lý dữ liệu GPS truyền từ Module GPS NEO-6M, ta viết hàm “read\_gps()“

void read\_gps(){

while(Serial1.available()){ //Kiểm tra dữ liệu GPS

if(gps.encode(Serial1.read())){ //Mã hóa dữ liệu GPS

gps.f\_get\_position(&flat,&flon);}

}

Serial.print("Position: ");

Serial.print("Latitude:");

Serial.print(flat,6);

Serial.print(";");

Serial.print("Longitude:");

Serial.println(flon,6);}

Tiếp theo ta viết hàm “xu\_ly\_data” để xử lý các giá trị kiểu dữ liệu dấu chấm động float trước khi truyền đến ESP3. Ta tách số float thành phần nguyên và phần thập phân, chuyển đổi chúng thành dạng nhị phân (binary) rồi truyền qua Serial3.

void xu\_ly\_data(int phannguyen,int bit\_1,int bit\_2, int bit\_3, float data){

phannguyen = int(data);

float phantp = data - phannguyen;

int i\_phantp = phantp \* 1000000;

Serial.println(phannguyen);

Serial.println(i\_phantp);

Serial.println(i\_phantp,BIN);

bit\_1 = i\_phantp & 0xFF0000;

bit\_1 = bit\_1 >> 16;

Serial.println(bit\_1,BIN);

bit\_2 = i\_phantp & 0x00FF00;

bit\_2 = bit\_2 >> 8;

Serial.println(bit\_2,BIN);

bit\_3 = i\_phantp & 0x0000FF;

Serial.println(bit\_3,BIN);

int ghep\_pn = ((bit\_1 << 16) + (bit\_2 << 8) + bit\_3);

float ghep\_pn\_f = (float)ghep\_pn/1000000;

Serial.println(ghep\_pn\_f);

Serial3.write(phannguyen);

Serial3.write(bit\_1);

Serial3.write(bit\_2);

Serial3.write(bit\_3);}

Khi nút nhấn được bấm và nhả, cạnh lên sẽ xuất hiện ở chân ngắt, chương trình sẽ chạy hàm “change\_status()”. Hàm sẽ cập nhật biến “state” ứng với trạng thái của nút nhấn và xuất ra tín hiệu trên chân PF\_2. Tương tự với hàm “change\_status\_1()” xác định trạng thái của một nút nhấn khác:

void change\_status(){

state++ ;

Serial.println("trang thai");

Serial.println(state);

if(state%2==1){

digitalWrite(PF\_2,HIGH);}

else{

digitalWrite(PF\_2,LOW);}

Hàm “blinkled()” dưới đây xác định giá trị “pump\_stt” dùng để truyền tín hiệu bật/tắt đến ESP32. Tương tự với hàm “blinkled\_1()” khi ta muốn xác định giá trị “pump\_stt\_1” của một nút nhấn khác:

void blinkled(){

if(state%2==1){

pump\_stt = 2;

digitalWrite(PF\_2,HIGH);}

else{

pump\_stt = 3;

digitalWrite(PF\_2,LOW);}

Serial.println(pump\_stt);}

Tiếp theo là hàm “recive\_data” dùng để xác định trạng thái ngõ ra khi nhận được tín hiệu từ ESP và gán giá trị cho trạng thái đó. Ta cũng sử dụng hàm “recive\_data\_1” tương tự như trên:

void recive\_data(int rc\_data\_s1){

if(rc\_data\_s1==2){

state = 1;}

else if(rc\_data\_s1==3){

state = 0;}

}

Kế đến ta xây dựng hàm “loop()” để thực hiện các tiến trình trong vòng lặp. Đầu tiên là khối nhận tín hiệu điều khiển thiết bị từ ESP qua Serial3. Ta gọi các hàm “recive\_data” để nhận và “blinkled” để truyền tín hiệu trạng thái:

void loop() {

int rc\_data\_s,rc\_data\_s\_1;

int rc\_data,rc\_data\_1;

while(Serial3.available()){

rc\_data = Serial3.read();

Serial.println("dddddddddddddd545454");

Serial.println(rc\_data);

rc\_data\_1 = Serial3.read();

Serial.println(rc\_data\_1);

Serial.println("dddddddddddddd545454");

if(pre\_dt != rc\_data){

rc\_data\_s = rc\_data;

pre\_dt = rc\_data;}

recive\_data(rc\_data\_s);

blinkled();

if(pre\_dt\_1 != rc\_data\_1){

rc\_data\_s\_1 = rc\_data\_1;

pre\_dt\_1 = rc\_data\_1;}

recive\_data\_1(rc\_data\_s\_1);

blinkled\_1();}

Khối tiếp theo dùng ta lập trình Tiva C đọc thông tin từ các cảm biến DHT, cảm biến độ ẩm đất, cảm biến cường độ sáng và vị trí GPS:

if((unsigned long)millis() - lastTime >= 2000){

//Bắt đầu đọc dữ liệu từ cảm biến

read\_gps();

pinMode(sensorPin, INPUT);

anhsang = digitalRead(sensorPin);

if (anhsang==0){as=1;}

else{as=0;}

float h = dht.readHumidity();

float t = dht.readTemperature();

//Kiểm tra quá trình đọc thành công hay không

if (isnan(h) || isnan(t) ) {

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

h=0;t=0;

return;}

float hic = dht.computeHeatIndex(t, h, false);

int soilValue = analogRead(PB\_5);

soilValue =((int)soilValue\*100)/4096;

soilValue=99-(int)soilValue;

Ngay sau đó ta truyền dữ liệu đọc được đến ESP qua Serial3:

Serial3.write(0x23);

xu\_ly\_data(phannguyen\_flat,bit1\_flat,bit2\_flat,bit3\_flat,flat);

xu\_ly\_data(phannguyen\_flon,bit1\_flon,bit2\_flon,bit3\_flon,flon);

xu\_ly\_data(phannguyen\_temp,bit1\_temp,bit2\_temp,bit3\_temp,t);

xu\_ly\_data(phannguyen\_hum,bit1\_hum,bit2\_hum,bit3\_hum,h);

Serial3.write(soilValue);

Serial3.write(as);

Đồng thời in thông tin ra màn hình debug (cổng COM):

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" \*C ");

Serial.print("sun: ");

Serial.print(as);

Serial.print(" Soil Value:");

Serial.print(soilValue);

Serial3.write(0x25);

Serial3.write(pump\_stt);

Serial3.write(pump\_stt\_1);

Serial.println("--------------------------------------");

Serial.print(" button1:");

Serial.println(String(rc\_data));

Và in thông tin ra LCD:

if(flag){

lcd.clear();

lcd.setCursor(0, 1);

lcd.print("HUMID% = "); //Thông tin về độ ẩm

lcd.print(h);

lcd.println(" %" );

lcd.setCursor(1, 0);

lcd.print(" TEMP = "); //Thông tin về nhiệt độ

lcd.print(t);

lcd.println("\*C");

flag = !flag;

}else{

lcd.clear();

lcd.setCursor(1, 0);

lcd.print(" Solar = "); //Thông tin về cường độ sáng

lcd.print(as);

lcd.setCursor(0, 1);

lcd.print("Soil Value: "); //Thông tin về độ ẩm đất

lcd.print(soilValue);

lcd.println(" % " );

flag = !flag;}

lastTime = millis();}}

### Chương trình nạp vào Module Wifi ESP32

Để viết chương trình, biên dịch và nạp vào ESP32, ta sử dụng phần mềm Arduino IDE. Trước tiên ta nhập vào các thư viện cần thiết bao gồm <WiFi.h> là mạng sử dụng để kết nối không dây, <HTTPClient.h> xác định giao thức kết nối và định dạng kiểu dữ liệu cho các giá trị sử dụng trong chương trình:

#include <WiFi.h>

#include <HTTPClient.h>

float t,h,flat,flon,sun;

int soil;

String gui,pre\_gui,gui\_1,pre\_gui\_1;

int ledStatus,lstt,ledStatus\_1,lstt\_1,bien,bien\_1,abc,abc\_1;

boolean xyz,xyz\_1;

const char\* ssid = "MinhQuan"; //tên wifi mà module cần kết nối tới

const char\* password = "12345678"; //password wifi

String server = "http://tantoanbk.tk"; //server để nhận dữ liệu lên

const char\* host = "tantoanbk.000webhostapp.com"; //host để đưa dữ liệu lên

Tiếp theo sẽ là hàm “setup()” để cài đặt các thông số cần thiết cho module như tốc độ baud giao tiếp cổng COM, tốc độ baud giao tiếp Serial2 với kit Tiva C, đồng thời cho phép module kết nối đến mạng wifi như đã cài đặt ở phần trên:

void setup(){

Serial.begin(115200);

Serial2.begin(9600);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());}

Kế đến, ta viết hàm “xu\_ly\_dulieu()” để xử lý dữ liệu nhận được từ kit Tiva C qua Serial2. Nói cách khác, từ một con số nhận được ở kiểu dữ liệu dấu chấm động float, hàm này là công đoạn tách chuỗi các bit, sau đó ghép lại thành một con số với “phannguyen” nằm trước dấu chấm thập phân và 6 chữ số phía sau dấu chấm thập phân.

float xu\_ly\_dulieu(){

int phannguyen,bit1,bit2,bit3;

phannguyen = Serial2.read();

bit1 = Serial2.read();

bit2 = Serial2.read();

bit3 = Serial2.read();

int ghep\_pn = ((bit1 << 16) + (bit2 << 8) + bit3);

float ghep\_pn\_f = (float)ghep\_pn/1000000;

float tong = float(phannguyen)+ghep\_pn\_f;

Serial.println(tong,6);

return tong;}

Để nhận tín hiệu điều khiển thiết bị từ server, ta viết hàm “Get\_Data()”. Hàm sẽ đọc dữ liệu từ trang web http://tantoanbk.tk/getdata.php dưới dạng chuỗi, sau đó phân tích chuỗi. Nếu có chuỗi “RL12” xuất hiện thì số “2” sẽ là tín hiệu bật, số “3” sẽ là tín hiệu tắt. Tương tự, nếu có chuỗi “RL13” xuất hiện thì số “4” sẽ là tín hiệu bật, số “5” sẽ là tín hiệu tắt.

void Get\_Data(){

HTTPClient GetStatus;

String Link = server + "/getdata.php";

GetStatus.begin(Link);

int Code = GetStatus.GET();

String Content = GetStatus.getString();

GetStatus.end();

Serial.println(Content);

int RL12 = Content.indexOf("RL12");

int RL13 = Content.indexOf("RL13");

int bat12 = Content.indexOf("2",RL12 + 1);

int tat12 = Content.indexOf("3",RL12 + 1);

int tt12 = Content.indexOf("RL",RL12 + 1);

int bat13 = Content.indexOf("4",RL13 + 1);

int tat13 = Content.indexOf("5",RL13 + 1);

int tt13 = Content.indexOf("RL",RL13 + 1);

if (RL12 != -1) {

if (bat12 != -1 && bat12 < tt12){

bien = 2;

abc = 1;}

if (tat12 != -1 && tat12 < tt12){

bien = 3;

abc = 0;}

}

Serial.println("--------------");

if (RL13 != -1) {

if (bat13 != -1 && bat13 < tt13){

bien\_1 = 4;

abc\_1 = 1;}

if (tat13 != -1 && tat13 < tt13){

bien\_1 = 5;

abc\_1 = 0;}}}

Khi nhấn nút nhấn trên board mạch, để cập nhật trạng thái thiết bị lên server, ta sử dụng hàm “Get\_Button()”. Hàm này khởi động bằng thủ tục kết nối lên host, sau đó thực hiện việc cập nhật bằng phương thức GET của HTML và kết thúc bằng việc đóng kết nối. Giá trị “id” trong hàm tương ứng với thiết bị cần cập nhật trạng thái. Ta viết các hàm tương tự ví dụ như “Get\_Button\_1()” với một giá trị “id” khác để hướng đến thiết bị mong muốn.

void Get\_Button(){

Serial.print("connecting to ");

Serial.println(host);

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;}

int id = 1; //id của thiết bị

String urls = "/api/capnhat.php?id="+String(id)+"&trangthai="+String(gui);

Serial.print("Requesting URL: ");

Serial.println(urls);

client.print(String("GET ") + urls + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

while(client.available()){

String line = client.readStringUntil('\r');

Serial.print(line);}

Serial.println();

Serial.println("closing connection");}

Tiếp theo, ta xây dựng hàm “loop()” để thực hiện các tiến trình trong vòng lặp. Khi nhận tín hiệu “0x23” từ Serial2, ESP đọc dữ liệu được truyền từ Tiva C gồm kinh độ (flat), vĩ độ (flon), nhiệt độ (t), độ ẩm (h), độ ẩm đất (soil), cường độ sáng (sun). Và khi nhận tín hiệu “0x25” từ Serial2 thì ESP đọc tín hiệu điều khiển thiết bị từ nút nhấn nối với Tiva C. Ở đây ta cũng gọi hàm “Get\_Data()” để liên tục nhận thông tin điều khiển từ server.

void loop(){

if(Serial2.read()==0x23){

flat = xu\_ly\_dulieu();

flon = xu\_ly\_dulieu();

t = xu\_ly\_dulieu();

h = xu\_ly\_dulieu();

soil = Serial2.read();

sun = Serial2.read();

if(Serial2.read() == 0x25){

lstt = Serial2.read();

Serial.println(lstt);

lstt\_1 = Serial2.read();

Serial.println(lstt\_1);

ledStatus = lstt;

ledStatus\_1 = lstt\_1;

if(ledStatus==2){

gui = "1";}

if(ledStatus==3){

gui = "0";}

if(ledStatus\_1==4){

gui\_1 = "1";}

if(ledStatus\_1==5){

gui\_1 = "0";}

}

Serial.println("LED Status");

Serial.println(ledStatus);

Serial.println(gui);}

Get\_Data();

Điều kiện để chạy hàm “Get\_Button()” để cập nhật trạng thái thiết bị là sự kiện nhấn nút xảy ra. Hàm cập nhật sẽ lặp lại liên tục cho đến khi giá trị “gui” từ ESP trùng với giá trị “bien” mà ESP nhận được từ server. Kết thúc tiến trình bằng câu lệnh đóng kết nối. Lưu ý khối này vẫn nằm trong vòng lặp của hàm “loop()”.

if(gui != pre\_gui){

xyz = 1;

while(xyz==1){

Get\_Button();

Get\_Data();

if(String(abc)==gui){

xyz = 0;

break;}}}

pre\_gui = gui;

if(gui\_1 != pre\_gui\_1){

xyz\_1 = 1;

while(xyz\_1==1){

Serial.println(bien\_1);

Serial.println(gui\_1);

Get\_Button\_1();

Get\_Data();

if(String(abc\_1)==gui\_1){

xyz\_1 = 0;

break;}}}

pre\_gui\_1 = gui\_1;

Serial2.write(bien);

Serial.println(bien);

Serial2.write(bien\_1);

Serial.println(bien\_1);

Serial.print("connecting to ");

Serial.println(host);

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;}

Sau cùng là khối chương trình dùng để chèn dữ liệu thu thập được từ cảm biến lên cơ sở dữ liệu. Ở đây ta tiếp tục sử dụng phương thức GET của HTML để đưa dữ liệu lên dưới dạng chuỗi. Tiến trình này nằm trong hàm “loop()” để lặp lại liên tục. Kết thúc tiến trình bằng câu lệnh đóng kết nối.

String url = "/api/insertweather.php?temp="+String(t)+"&hum="+String(h)+"&soil="+String(soil)+"&solar="+String(sun)+"&flat="+String(flat)+"&flon="+String(flon);

Serial.print("Requesting URL: ");

Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

while(client.available()){

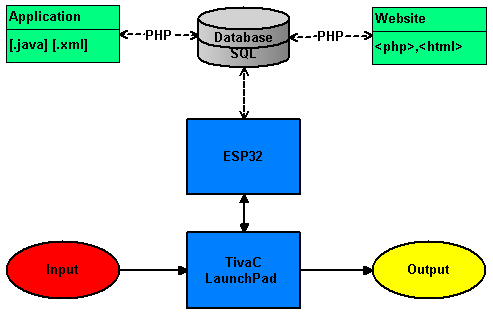
String line = client.readStringUntil('\r');

Serial.print(line);}

Serial.println();

Serial.println("closing connection");}

## Xây dựng Website

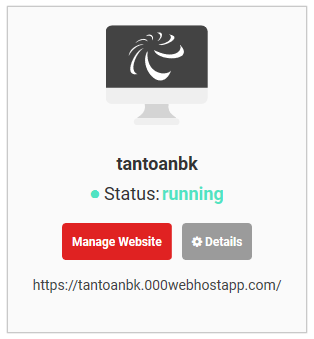


Hình 4‑4. Sơ đồ giao tiếp giữa phần cứng và Website

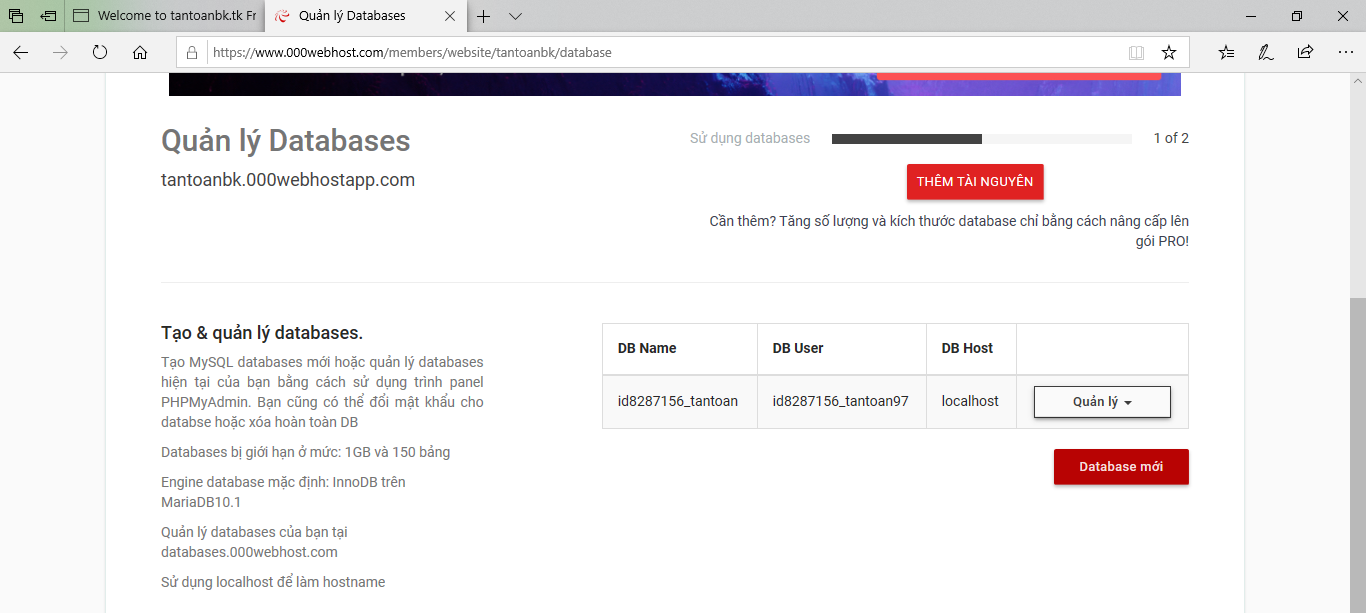
Luận văn này sẽ sử dụng dịch vụ của [www.000webhost.com](http://www.000webhost.com) để xây dựng trang web. “000webhost” cung cấp bởi Hostinger miễn phí web hosting với PHP, MySQL, cPanel và không quảng cáo. Với khả năng kích hoạt tài khoản ngay lập tức ta có thể bắt đầu dự án nhanh chóng, bất kể là một trang blog WordPress hay một website HTML, đều có thể dễ dàng tạo được chỉ trong vài bước. Để cung cấp một môi trường tối ưu cho người dùng, “000webhost” đã phát triển riêng một control panel (trang điều khiển) với các tính năng quan trọng nhất được tích hợp vào một chỗ. Server không gặp tình trạng nghẽn mạng, có thể xử lý truy vấn ở cường độ cao với tài nguyên server giới hạn. Nền tảng web hosting miễn phí được giám sát định kì và cập nhật công nghệ mới nhất của PHP và MySQL. Server cũng thường xuyên được kiểm thử để đảm bảo môi trường web hosting an toàn, đáng tin cậy và nhanh chóng cho tất cả những ai đang sử dụng nó trên khắp thế giới.

Khuyết điểm của “000webhost” là cơ sở dữ liệu thiếu ổn định, các tác vụ truy cập và xử lý dữ liệu trên database hay bị báo lỗi khiến cho việc truy cập và tải trang web gặp nhiều khó khăn.

Sau khi đăng kí hosting 000webhost và trỏ domain đã tạo sẵn trước đó là tantoanbk.tk về 000webhost, ta tiến hành tạo cơ sở dữ liệu (database).

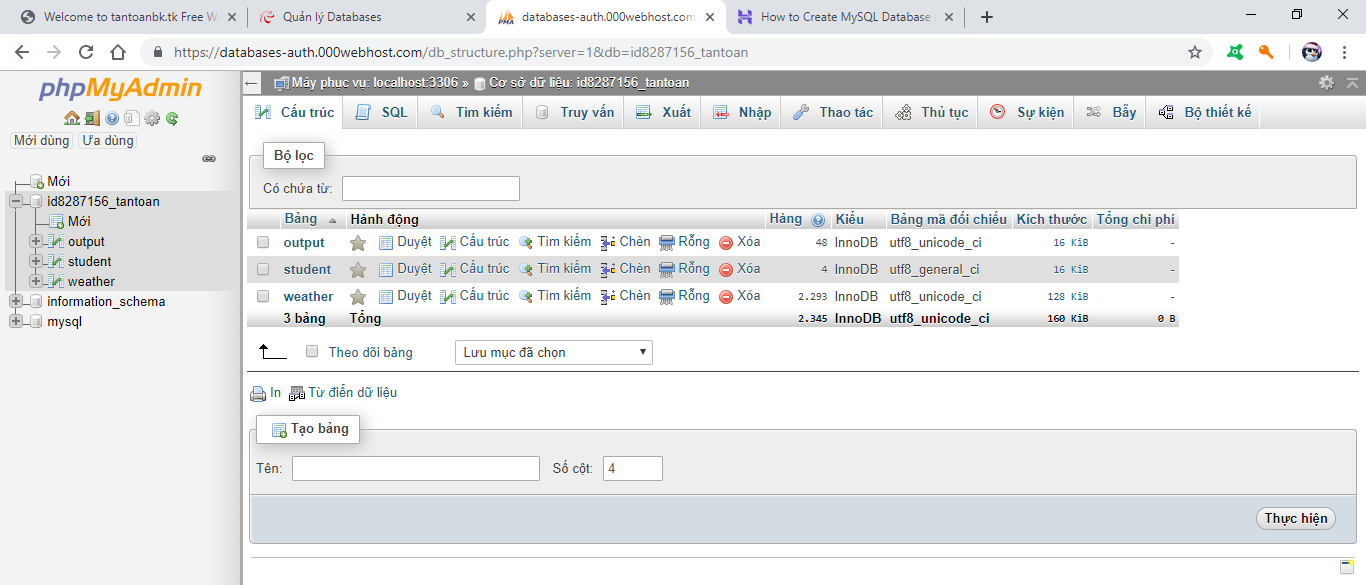


Hình 4‑5. Địa chỉ của trang web đã tạo

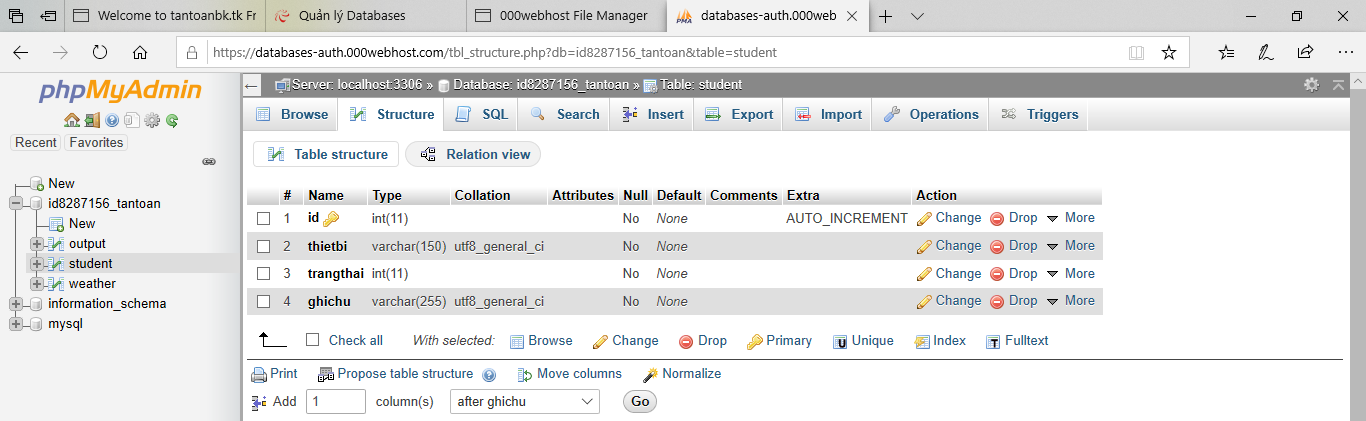


Hình 4‑6. Quản lý cơ sở dữ liệu được tạo

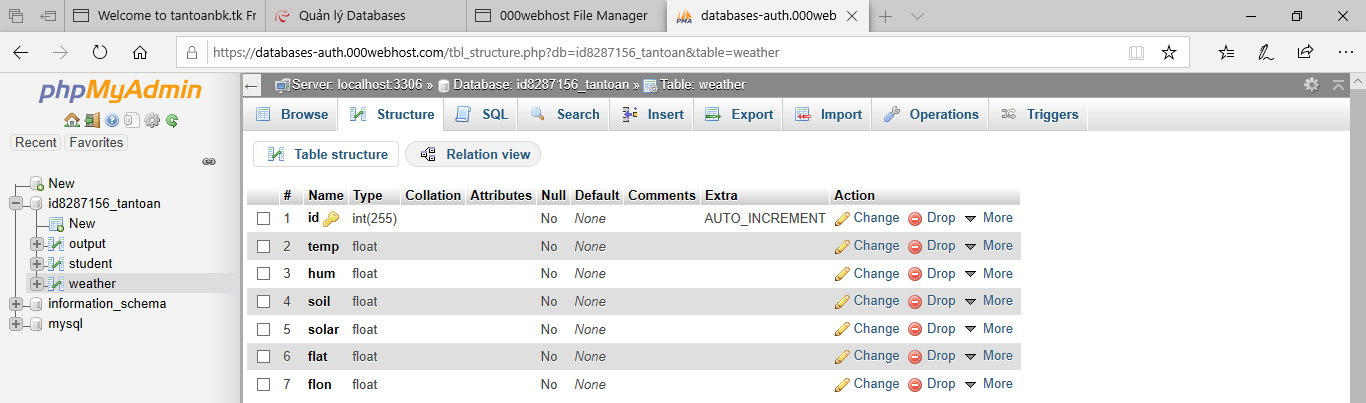
Cơ sở dữ liệu của ta ở đây có tên là “id8287156\_tantoan”, user là “id8287156\_tantoan97”, host mặc định là “localhost”. Tiếp tục tạo các bảng “output”, “student”, “weather” để phục vụ cho việc lưu trữ dữ liệu. Bảng “output” chứa kết quả dự báo, bảng “student” chứa thông tin điều khiển thiết bị, bảng “weather” chứa dữ liệu cảm biến thu thập được.



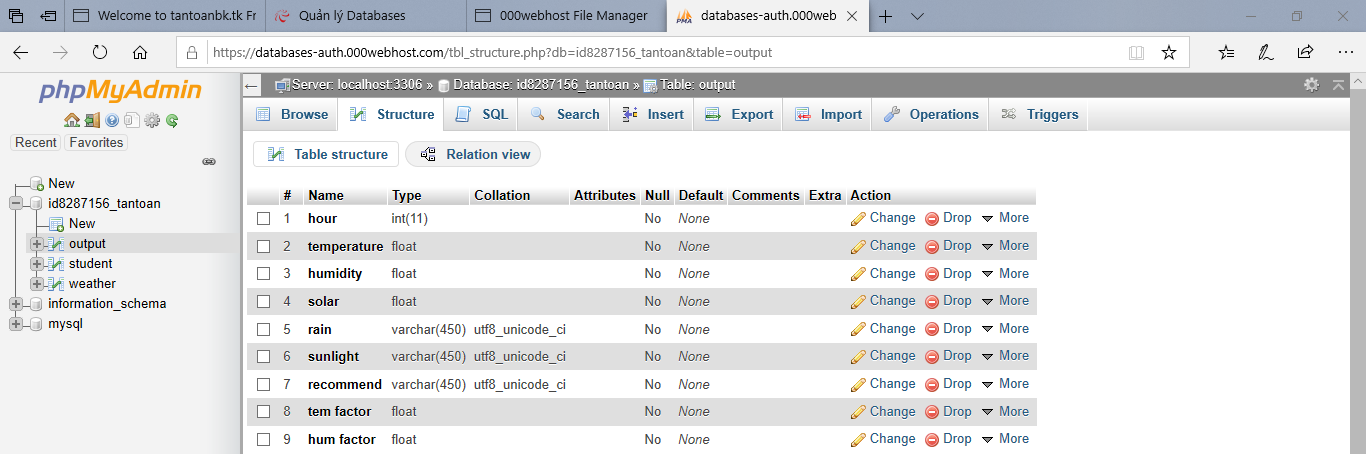
Hình 4‑7. Cấu trúc của database “id8287156\_tantoan”



Hình 4‑8. Cấu trúc của bảng “student”



Hình 4‑9. Cấu trúc của bảng “weather”



Hình 4‑10. Cấu trúc của bảng “output”

Bước tiếp theo ta sẽ xây dựng Website thông qua các tệp mã nguồn được viết bằng ngôn ngữ PHP trên phần mềm Sublime Text 3. Đầu tiên là tạo kết nối giữa Website với database bằng 2 tệp tin “dbconfig.php”:

<?php

define('DB\_USER', "id8287156\_tantoan97"); //Nhập user name

define('DB\_PASSWORD', "tantoan97"); //Nhập database password

define('DB\_DATABASE', "id8287156\_tantoan"); //Nhập database name

define('DB\_SERVER', "localhost"); //Nhập database server

?>

và “db\_connect.php”:

<?php

class DB\_CONNECT {

//Tạo hàm kết nối

function \_\_construct() {

$this->connect();} //Kết nối đến database

//Tạo hàm ngắt kết nối

function \_\_destruct() {

$this->close();} //Ngắt kết nối database

//Hàm kết nối đến database

function connect() {

//nhập tệp dbconfig.php chứa thông tin xác thực cơ sở dữ liệu

$filepath = realpath (dirname(\_\_FILE\_\_));

require\_once($filepath."/dbconfig.php");

//Kết nối với mysql

$con = mysql\_connect(DB\_SERVER, DB\_USER, DB\_PASSWORD) or die(mysql\_error());

//Chọn cơ sở dữ liệu

$db = mysql\_select\_db(DB\_DATABASE) or die(mysql\_error()) or die(mysql\_error());

return $con;}

//Hàm đóng kết nối đến mysql

function close() {

mysql\_close();}

}?>

Kế tiếp ta viết tệp “insertweather.php” dùng để chèn thêm dữ liệu cảm biến thu thập được vào bảng “weather” có trong database.

<?php

header("Access-Control-Allow-Origin: \*");

header("Content-Type: application/json; charset=UTF-8");

//Tạo mảng cho phản hồi JSON

$response = array();

//Kiểm tra trường nhận được từ user

if (isset($\_GET['temp']) && isset($\_GET['hum']) && isset($\_GET['soil']) && isset($\_GET['solar']) && isset($\_GET['flat']) && isset($\_GET['flon'])) {

$temp = $\_GET['temp'];

$hum = $\_GET['hum'];

$soil = $\_GET['soil'];

$solar = $\_GET['solar'];

$flat = $\_GET['flat'];

$flon = $\_GET['flon'];

$filepath = realpath (dirname(\_\_FILE\_\_)); //Gọi lớp kết nối cơ sở dữ liệu

require\_once($filepath."/db\_connect.php");

$db = new DB\_CONNECT(); //Kết nối đến cơ sở dữ liệu

Truy vấn SQL để chèn dữ liệu vào bảng “weather”:

$result = mysql\_query("INSERT INTO weather(temp,hum,soil,solar,flat,flon) VALUES('$temp','$hum','$soil','$solar','$flat','$flon')");

//Kiểm tra thực hiện thành công truy vấn

if ($result) {

$response["success"] = 1; //chèn thành công

$response["message"] = "Weather successfully created.";

echo json\_encode($response);} //Hiển thị phản hồi JSON

else {

$response["success"] = 0; //Không thể chèn dữ liệu vào cơ sở dữ liệu

$response["message"] = "Something has been wrong";

echo json\_encode($response);} //Hiển thị phản hồi JSON

} else {

//Nếu thiếu tham số yêu cầu

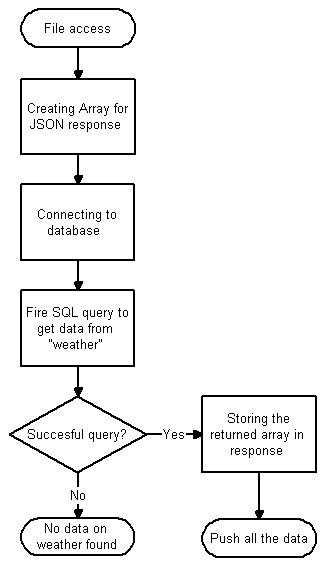
$response["success"] = 0;

$response["message"] = "Parameter(s) are missing. Please check the request";

echo json\_encode($response);} //Hiển thị phản hồi JSON

?>

Viết tệp “read\_all.php” dùng để đọc bảng “weather” trên Website dưới dạng chuỗi.



Hình 4‑11. Lưu đồ giải thuật đọc dữ liệu từ database lên website

<?php

header("Access-Control-Allow-Origin: \*");

header("Content-Type: application/json; charset=UTF-8");

$response = array(); //Tạo mảng cho phản hồi JSON

//Gọi lớp kết nối cơ sở dữ liệu từ tệp “db\_connect.php”

$filepath = realpath (dirname(\_\_FILE\_\_));

require\_once($filepath."/db\_connect.php");

$db = new DB\_CONNECT(); //Kết nối đến cơ sở dữ liệu

//Truy vấn SQL để lấy tất cả dữ liệu từ bảng “weather”

$result = mysql\_query("SELECT \*FROM weather") or die(mysql\_error());

//Kiểm tra thực hiện truy vấn thành công hoặc không tìm thấy kết quả

if (mysql\_num\_rows($result) > 0) {

$response["weather"] = array(); //Lưu trữ mảng trả về trong phản hồi

//Vòng lặp while để lưu trữ tất cả các phản hồi trả về trong biến

while ($row = mysql\_fetch\_array($result)) {

$weather = array(); //Gán các giá trị của mảng

$weather["id"] = $row["id"];

$weather["temp"] = $row["temp"];

$weather["hum"] = $row["hum"];

$weather["soil"] = $row["soil"];

$weather["solar"] = $row["solar"];

$weather["flat"] = $row["flat"];

$weather["flon"] = $row["flon"];

array\_push($response["weather"], $weather);} //Đẩy tất cả các mục

$response["success"] = 1; //Thành công

echo json\_encode($response);} //Hiển thị phản hồi JSON

else {//Nếu không có dữ liệu

$response["success"] = 0;

$response["message"] = "No data on weather found";

echo json\_encode($response);} //Hiển thị phản hồi JSON

?>

Viết tệp “index.html” là giao diện hiển thị giá trị cuối cùng của bảng “weather” lên Website, gồm có: nhiệt độ, độ ẩm không khí, độ ẩm đất, cường độ sáng, tọa độ đo đạc [5].

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<meta http-equiv="Access-Control-Allow-Origin" content="\*">

<script type="text/javascript">

document.write(["\<script src='",

("https:" == document.location.protocol) ? "https://" : "http://",

"ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js' type='text/javascript'>\<\/script>"

].join(''));

</script>

Trang web được viết bằng ngôn ngữ HTML, sử dụng phương thức “ajax” trong thư viện “jquery” để tải một phần dữ liệu mới vào tài liệu HTML mà không cần phải tải lại toàn bộ tài liệu, giúp giảm thiểu thời gian tải trang cũng như băng thông của Website. Tiếp theo là viết tiêu đề cũng như xây dựng bố cục cho Website:

<title>Sensor From PHP API</title>

</head>

<body bgcolor="#ffffff">

<center style=" height: 350px;

width: 800px;

background: #c0c5ce;

position: absolute;

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

top: 30%;

left: 50%;

margin-top: -160px;

margin-left: -400px;">

Khối tiếp theo sẽ xây dựng phần hiển thị của dữ liệu dưới dạng bảng (không có viền bảng). Trong đó ta cài đặt các thông số về kích thước (chiều rộng), màu sắc, đổ bóng, kiểu chữ, cỡ chữ, canh lề cho bảng:

//Hiển thị tựa đề bảng

<h1 style="font-family:Helvetica; font-size:20px; color: black;">Sensor From PHP API</h1>

<br><br>//Xây dựng bảng hiển thị

<table style="border-collapse: collapse;width: 100%;color: #588c7e;font-family: Times New Roman; font-size: 25px;text-align: center;">

<tr><td style = 'line-height: 60px;font-family: Helvetica;color: #fff;font-size: 50px;' id="temperature">

<img src = 'temperature.png' height="60px" width="60px" style='vertical-align: middle' /> 00.00\*C</td>

<td style = 'line-height: 60px;font-family: Helvetica;color: #fff;font-size: 50px;' id="humidity">

<img src = 'humidity.png' height="60px" width="60px" style='vertical-align: middle' /> 00.00%</td></tr>

<tr><td style = 'line-height: 100px;font-family: Helvetica;color: #fff;font-size: 50px;' id="soilsss">

<img src = 'soil2.png' height="60px" width="60px" style='vertical-align: middle' /> 00.00%</td>

<td style = 'line-height: 100px;font-family: Helvetica;color: #fff;font-size: 50px;' id="solarsss">

<img src = 'sun.png' height="65px" width="60px" style='vertical-align: middle' /> 00.00</td></tr>

<tr><td style = 'line-height: 100px;font-family: Helvetica;color: #fff;font-size: 50px;' id="latitude"> 00.00</td>

<td style = 'line-height: 100px;font-family: Helvetica;color: #fff;font-size: 50px;' id="longitude"> 00.00

</td></tr></table></center>

Bên dưới ta sẽ cho Website hiển thị bản đồ cho biết vị trí của cảm biến thu thập dữ liệu. Dịch vụ định vị của Google Map rất mạnh và chính xác nên để tiện lợi và đơn giản ta sẽ nhúng bản đồ từ Google Map vào Website.

<center>

<div id="map" style="width:700px;height:700px;"></div>

//Nhúng bản đồ từ Google Map

<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3919.5372732263427!2d106.65795211433674!3d10.770100662256551!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x31752ec272ddbf69%3A0xf98ffdac134090f9!2zN0IvMTA1IFRow6BuaCBUaMOhaSwgUGjGsOG7nW5nIDE0LCBRdeG6rW4gMTAsIEjhu5MgQ2jDrSBNaW5oLCBWaeG7h3QgTmFt!5e0!3m2!1svi!2s!4v1555647659132!5m2!1svi!2s" width="600" height="450" frameborder="0" style="border:0" allowfullscreen></iframe>

</center>

</body>

<footer style="background:#64B5F6;

width:100%;

height:100px;

bottom:0;

left:0;">

Phía dưới cùng của trang web là tên tác giả và nguồn tham khảo

<center>

//Tên tác giả cuối website

<br><h4 style="font-family: Helvetica;color: white;">&copy; 2019 | Tan Toan Nguyen | hcmut<br>

Reference: <a href="http://vsgupta.in/">Vivek Gupta</a> | <a href="http://www.iotmonk.com/">IoTMonk.com</a></h4>

</center></footer>

Công việc tiếp theo là đưa dữ liệu vào Website. Đầu tiên là đọc dữ liệu dưới dạng mảng (array) từ trang web PHP đã tạo trước đó là read\_all.php. Ta tiến hành gán các giá trị: nhiệt độ = “temper”, độ ẩm = “humid”, cường độ sáng = “as”, độ ẩm đất = “soi”, kinh độ = “longi”, vĩ độ = “lati” thành các biến “var”, từ đó dùng hàm Object.keys để đọc dữ liệu:

//Đưa dữ liệu vào website bằng hàm loaddata()

<script>

window.onload = function() {

loaddata();};

function loaddata(){

var url = "https://tantoanbk.000webhostapp.com/api/read\_all.php";

//Nhúng mảng từ trang web “read\_all.php” đã tạo

$.getJSON(url, function(data) {

var val= data; //Gán các giá trị đọc được

var humid=(data['weather'][(Object.keys(data['weather']).length)-1]['hum']);

var temper=(data['weather'][(Object.keys(data['weather']).length)-1]['temp']);

var soi=(data['weather'][(Object.keys(data['weather']).length)-1]['soil']);

var as=(data['weather'][(Object.keys(data['weather']).length)-1]['solar']);

var lati=(data['weather'][(Object.keys(data['weather']).length)-1]['flat']);

var longi=(data['weather'][(Object.keys(data['weather']).length)-1]['flon']);

Cuối cùng ta nhúng các giá trị vào trang web HTML hiện tại bằng hàm document.getElementById theo thứ tự đính sẵn bởi giá trị “id”:

//Nhúng các giá trị vào trang web mới

document.getElementById("temperature").innerHTML = "<img src = 'temperature.png' height=\"60px\" width=\"60px\" style='vertical-align: middle' />" +' ' +temper +'\*C';

document.getElementById("humidity").innerHTML = "<img src = 'humidity.png' height=\"60px\" width=\"60px\" style='vertical-align: middle' />" +' ' +humid +'%';

document.getElementById("soilsss").innerHTML = "<img src = 'soil2.png' height=\"65px\" width=\"60px\" style='vertical-align: middle' />" +' ' +soi +'%';

document.getElementById("solarsss").innerHTML = "<img src = 'sun.png' height=\"60px\" width=\"60px\" style='vertical-align: middle' />" +' ' +as;

document.getElementById("latitude").innerHTML = 'lat=' +lati;

document.getElementById("longitude").innerHTML = 'long=' +longi;

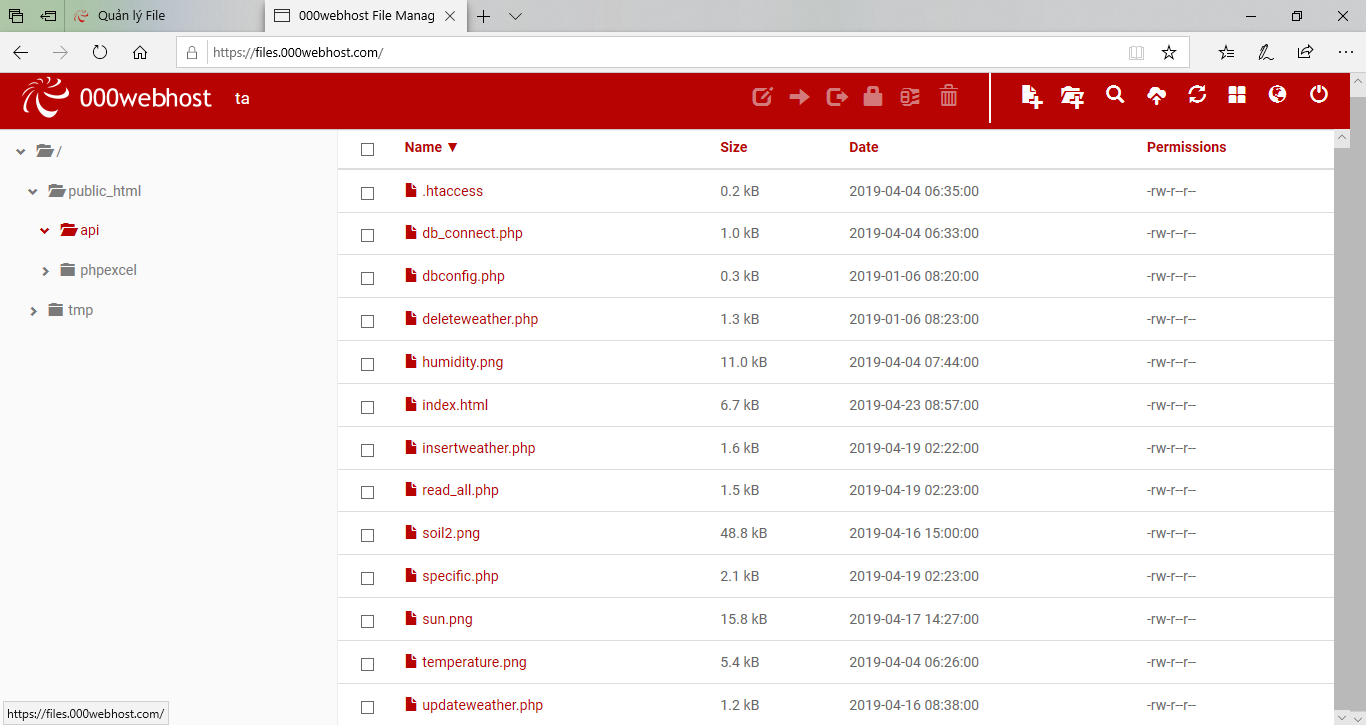
console.log(data['weather'][(Object.keys(data['weather']).length)-1]['temp']);});}

window.setInterval(function(){

loaddata();}, 5000);

</script></html>

Upload các tệp mã nguồn lên hosting:



Hình 4‑12. Các tệp truy xuất và hiển thị dữ liệu

Viết tệp “hienthi.php” dùng để hiển thị bảng dữ liệu dự báo từ bảng “output” lên Website:

<!DOCTYPE html>

<html><head>

<meta charset="UTF-8">

<title>Bảng dự đoán</title></head>

<body><table><tr>

<th>Hour</th>

<th>Temperature</th>

<th>Humidity</th>

<th>Solar</th>

<th>Rain</th>

<th>Sunlight</th></tr>

<?php

$conn = mysqli\_connect("localhost", "id8287156\_tantoan97", "tantoan97", "id8287156\_tantoan");

if ($conn->connect\_error) { //Kiểm tra kết nối

die("Connection failed: " . $conn->connect\_error);}

$sql = "SELECT hour, temperature, humidity, solar, rain, sunlight FROM output";

$result = $conn->query($sql);

if ($result->num\_rows > 0) { //Xuất dữ liệu theo hàng

while($row = $result->fetch\_assoc()) {

echo"<tr><td>".$row["hour"]."</td><td>".$row["temperature"]."</td><td>".$row["humidity"]."</td><td>".$row["solar"]."</td><td>".$row["rain"]."</td><td>".$row["sunlight"]."</td></tr>";}

echo "</table>";}

else { echo "0 results"; }

$conn->close();?>

</table></body></html>

Cuối cùng là Website điều khiển thiết bị gồm các tệp:

Tệp “view.php” hiển thị giao diện danh sách cách thiết bị:

<!DOCTYPE html>

<html><head>

<meta charset="UTF-8">

<title>Bảng điều khiển</title></head>

<body><table><tr>

<th>ID</th>

<th>Ngõ ra</th>

<th>Trạng thái</th>

<th>Thiết bị</th>

<th>Sửa</th></tr>

<?php

$connect = mysqli\_connect("localhost", "id8287156\_tantoan97", "tantoan97", "id8287156\_tantoan");

if ($connect->connect\_error) { //Kiểm tra kết nối

die("Connection failed: " . $connect->connect\_error);}

$sql = "SELECT id, thietbi, trangthai, ghichu FROM student";

$result = $connect->query($sql);

if ($result->num\_rows > 0) { //Xuất dữ liệu theo hàng

while($row = $result->fetch\_assoc()) {

echo "<tr><td>" . $row['id'] . "</td>";

echo "<td>" . $row['thietbi'] . "</td>";

echo "<td>" . $row['trangthai'] . "</td>";

echo "<td>" . $row['ghichu'] . "</td>";

//Truyền tham số id tới trang update.php

echo "<td><a href='update.php?id=" . $row['id'] . "'>Update</a></td>";

echo"</tr>";}

echo "</table>";}

else { echo "0 results"; }

$connect->close();?>

</table></body></html>

Tệp “update.php” hiển thị form chỉnh sửa thiết bị:

<?php

$username = "id8287156\_tantoan97"; //Khai báo username

$password = "tantoan97"; //Khai báo password

$server = "localhost"; //Khai báo server

$dbname = "id8287156\_tantoan"; //Khai báo database

//Kết nối database “id8287156\_tantoan”

$connect = mysqli\_connect($server, $username, $password, $dbname);

//Nếu kết nối bị lỗi thì xuất báo lỗi và thoát.

if (!$connect) {

die("Không kết nối :" . mysqli\_connect\_error());

exit();

}

Lấy dữ liệu từ “view.php” bằng phương thức GET:

if(isset($\_GET['id'])){

$id = $\_GET['id'];

$sql = "SELECT \* FROM student WHERE id='$id'";

$ket\_qua = $connect->query($sql);

while ($row = $ket\_qua->fetch\_array(MYSQLI\_ASSOC)) {

$thietbi = $row['thietbi'];

$trangthai = $row['trangthai'];

$ghichu = $row['ghichu'];?>

Truyền dữ liệu vào form:

<form action="process.php" method="post">

<meta charset="UTF-8" lang="vi">

<table><tr>

<th>ID:</th>

<td><input type="hidden" name="id" value="<?php echo $id; ?>">

<?php echo $id; ?>

</td></tr><tr>

<th>Ngõ ra:</th>

<td><input type="text" name="thietbi" value="<?php echo $thietbi; ?>">

</td></tr><tr>

<th>Trạng thái:</th>

<td><input type="binary" name="trangthai" value="<?php echo $trangthai; ?>">

</td></tr><tr>

<th>Thiết bị:</th>

<td><input type="text" name="ghichu" value="<?php echo $ghichu; ?>">

</td></tr></table>

<button type="submit">Xác nhận</button></button></form>

<?php }}

$connect->close();?> //Đóng kết nối database

Tệp “process.php” xử lý dữ liệu cập nhật:

<?php

$username = "id8287156\_tantoan97"; //Khai báo username

$password = "tantoan97"; //Khai báo password

$server = "localhost"; //Khai báo server

$dbname = "id8287156\_tantoan"; //Khai báo database

//Kết nối đến database

$connect = mysqli\_connect($server, $username, $password, $dbname);

//Nếu kết nối bị lỗi thì xuất báo lỗi và thoát.

if (!$connect) {

die("Không kết nối :" . mysqli\_connect\_error());

exit();}

//Đưa dữ liệu lên “update.php” bằng phương thức POST.

$id = $\_POST['id'];

$thietbi = $\_POST['thietbi'];

$trangthai = $\_POST['trangthai'];

$ghichu = $\_POST['ghichu'];

Xử lý, update dữ liệu vào table dựa theo điều kiện ‘WHERE’ tại ‘id’ tương ứng

$sql = "UPDATE student SET thietbi='$thietbi', trangthai='$trangthai', ghichu='$ghichu' WHERE id=$id";

if ($connect->query($sql) === TRUE) {

//Nếu kết quả kết nối thành công, trở về trang view.

header('Location: view.php');}

else {

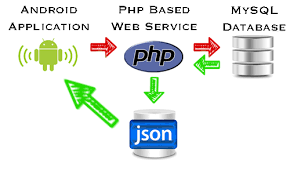
//Nếu kết quả kết nối không được thì trở về update.php đồng thời gán giá trị error=1, dựa theo giá trị này trang update.php có thể thông báo lỗi cần thiết.

header('Location: update.php?error=1');}

$connect->close(); //Đóng kết nối database

## Xây dựng ứng dụng điều khiển thiết bị trên Smartphone

Webservice khác biệt hơn so với localhost. Localhost chỉ sử dụng được bởi một máy, thiết bị cục bộ; các thiết bị khác không thể tương tác hay cập nhật. Còn đối với Webservice, ta viết bằng PHP, database là MySQL, chạy trên một Webserver, dữ liệu trả về App. Khi ta cập nhật cập nhật dữ liệu thì trên Server cũng cập nhật theo. Để App đọc được dữ liệu trên Server phpMySQL phải thông qua định dạng JSON. Thông qua JSON, ta sẽ viết các lệnh php trên Server truy vấn dữ liệu từ database MySQL trả ra định dạng JSON, và từ JSON đưa vào App Android. Ngược lại từ App Android có thể truy vấn lên Server, có thể là thêm, xóa, cập nhật dữ liệu (chuỗi, số…). Dữ liệu dạng online Webservice có thể linh động hơn localhost, khi cần cập nhật dữ liệu thì đều có thể thực hiện thông qua Internet.



Hình 4‑13. Cấu trúc giao tiếp giữa ứng dụng và cơ sở dữ liệu

Tương tự với phần xây dựng Website, ta sử dụng cơ sở dữ liệu “id8287156\_tantoan” đã tạo và bảng “student” để lưu trữ thông tin thiết bị. Sử dụng phần mềm Sublime Text 3 tạo tệp “getdata.php” để lấy dữ liệu JSON trả về [6].

<?php

class SinhVien{ //Tạo class

function SinhVien($thietbi,$trangthai,$ghichu){

$this->ThietBi=$thietbi;

$this->TrangThai=$trangthai;

$this->GhiChu=$ghichu;}

}

$mangSV=array(); //Tạo mảng

//Thêm phần tử vào mảng

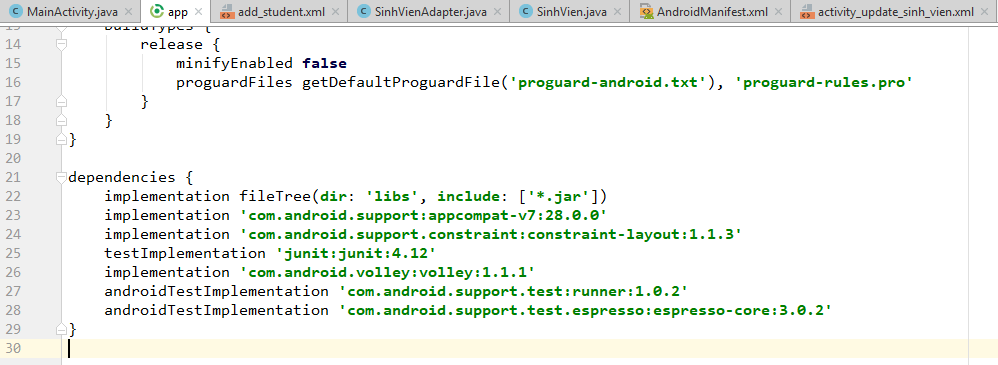
array\_push($mangSV,new SinhVien("RLA",1,"may bom"));

array\_push($mangSV,new SinhVien("RLB",0,"den"));

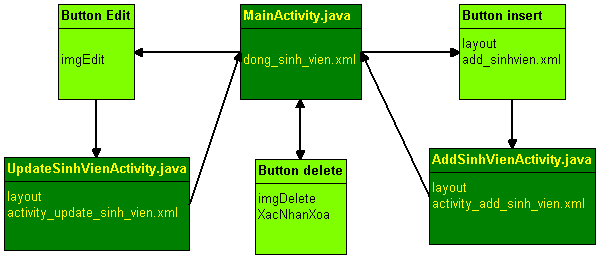
//Chuyển định dạng của mảng thành json

echo json\_encode($mangSV);?>

Ta tiến hành viết ứng dụng bằng ngôn ngữ Java, tạo project “web service” trên phần mềm Android Studio. Sử dụng dòng code: “implementation 'com.android.volley:volley:1.1.1' ” để đọc mảng dữ liệu trả về dạng JSON.

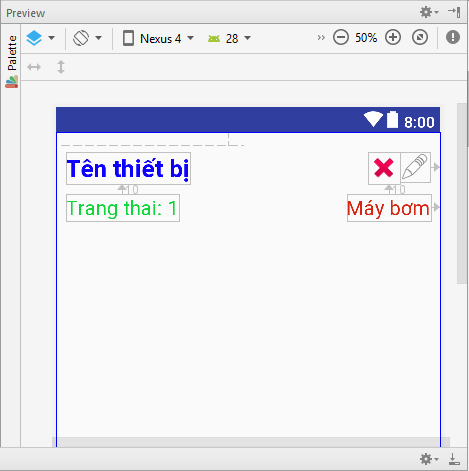


Hình 4‑14. Nhập thư viện “Volley” để xử lý dữ liệu dạng JSON từ Server



Hình 4‑15. Cấu trúc ứng dụng Android Webservice

Tạo layout “dong\_sinh\_vien.xml” là màn hình chính hiển thị danh sách thiết bị:



Hình 4‑16. Layout hiển thị danh sách thiết bị

Viết tệp “insert.php” dùng để chèn thêm thiết bị vào bảng, gửi dữ liệu lên bằng hàm $\_POST.

<?php

require "dbCon.php";

$thietbi = $\_POST['thietbiSV'];

$trangthai = $\_POST['trangthaiSV'];

$ghichu = $\_POST['ghichuSV'];

$query = "INSERT INTO student VALUES(null, '$thietbi','$trangthai','$ghichu')";

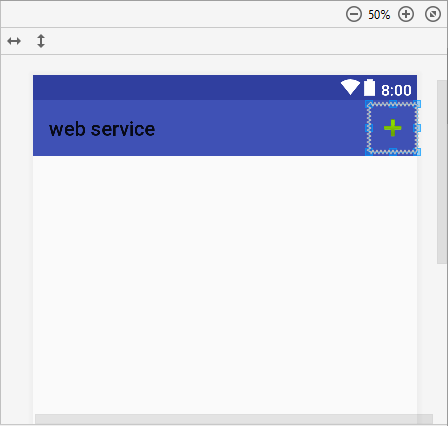
if(mysqli\_query($connect, $query)){

echo "success";} //thành công

else{ //lỗi

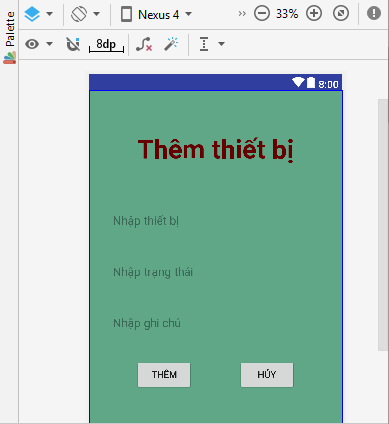
echo "error";}?>

Tạo menu “add\_student” để tạo nút nhấn chèn thêm thiết bị:



Hình 4‑17. Nút nhấn chèn thêm thiết bị

Tạo layout “activity\_add\_sinh\_vien.xml” hiển thị màn hình thêm thiết bị:



Hình 4‑18. Layout màn hình thêm thiết bị

Viết tệp “update.php” dùng để cập nhật thiết bị:

<?php

require "dbCon.php";

$id = $\_POST['idSV'];

$thietbi = $\_POST['thietbiSV'];

$trangthai = $\_POST['trangthaiSV'];

$ghichu = $\_POST['ghichuSV'];

$query = "UPDATE student SET thietbi = '$thietbi', trangthai = '$trangthai', ghichu = '$ghichu' WHERE id = '$id'";

if(mysqli\_query($connect, $query)){

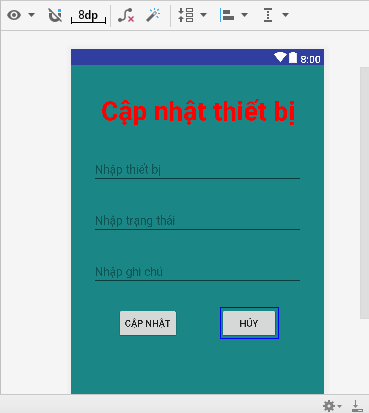
//thành công

echo "success";}

else{//lỗi

echo "error";}?>

Tạo layout “activity\_update\_sinh\_vien.xml” hiển thị màn hình cập nhật thiết bị:



Hình 4‑19. Layout màn hình cập nhật thiết bị

Viết tệp “delete.php” dùng để xóa thiết bị:

<?php

require "dbCon.php";

$idsv = $\_POST['idCuaSinhVien'];

$query = "DELETE FROM student WHERE id = '$idsv'";

if(mysqli\_query($connect, $query)){

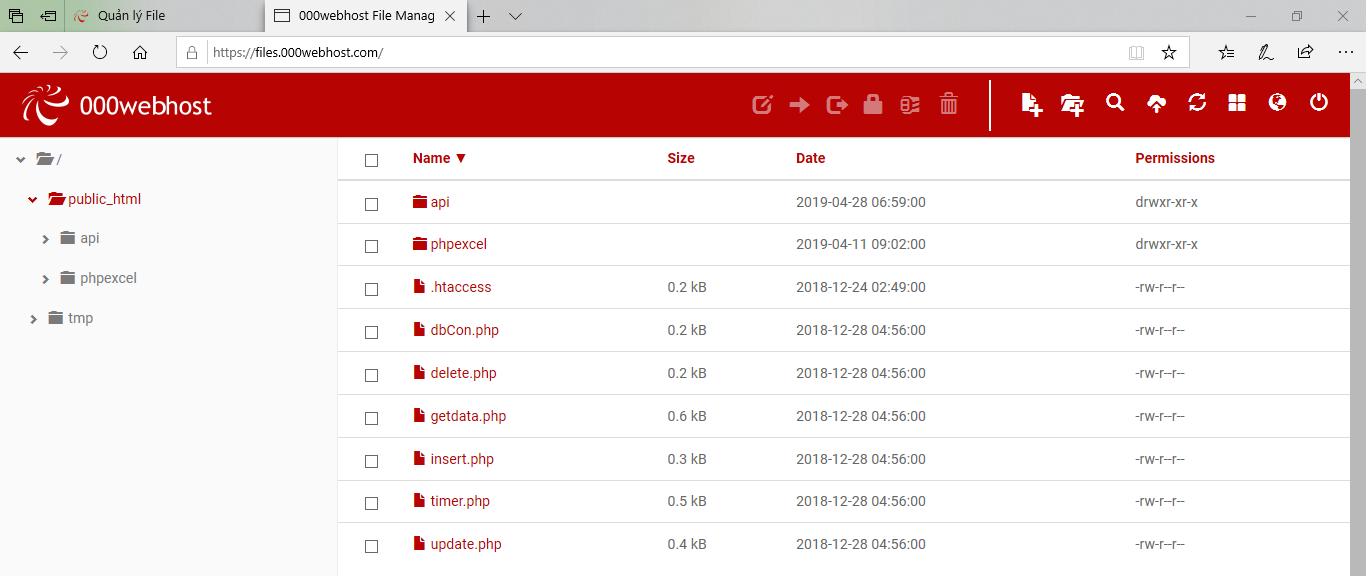
//thành công

echo "success";}

else{//lỗi

echo "error";}?>

Cuối cùng ta upload các tệp mã nguồn lên hosting:



Hình 4‑20. Các tệp truy xuất thông tin thiết bị từ bảng “student” trên cơ sở dữ liệu

****

Hình 4‑21. Trang web hiển thị trạng thái thiết bị

# KẾT QUẢ THỰC HIỆN

## Kết quả giải thuật dự báo

Sau khi triển khai thuật toán Linear Regression bằng Tensorflow trên Python với đầu vào là số liệu về nhiệt độ (cột “Temperature” trong tệp tin csv) màn hình xuất ra kết quả là:

[26.635757 26.93783 27.205732 27.298222 27.330328 26.826542 28.372198

27.943539 30.013775 32.1727 33.785038 33.33924 32.510853 33.169624

32.00715 31.931244 30.936794 29.56342 29.963161 29.379179 27.046267

25.87078 26.401134 26.679441 26.055984 25.77284 25.440844 24.99861

24.881916 25.628038 28.326578 28.239164 29.302927 29.70434 31.870382

32.757465 32.47599 32.42788 31.593191 31.471987 31.044271 30.604467

29.728916 28.979906 28.322004 27.719961 26.978312 26.42832 ]

Nếu chỉ nhìn vào dữ liệu đầu ra như thế kia thôi thì cũng không biết liệu kết quả dự đoán của ta sẽ ra sao. Do đó để dễ dàng so sánh, việc vẽ đồ thị sẽ rất cần thiết để giúp ta có cái nhìn chính fig=plt.figure().

plt.title("Temperature", fontsize=14)

plt.plot(pd.Series(np.ravel(Y\_test)), "bo", markersize=10, label="Actual")

plt.plot(pd.Series(np.ravel(y\_pred)), "r.", markersize=10, label="Forecast")

plt.legend(loc="upper left")

plt.xlabel("Time Periods")

plt.show()

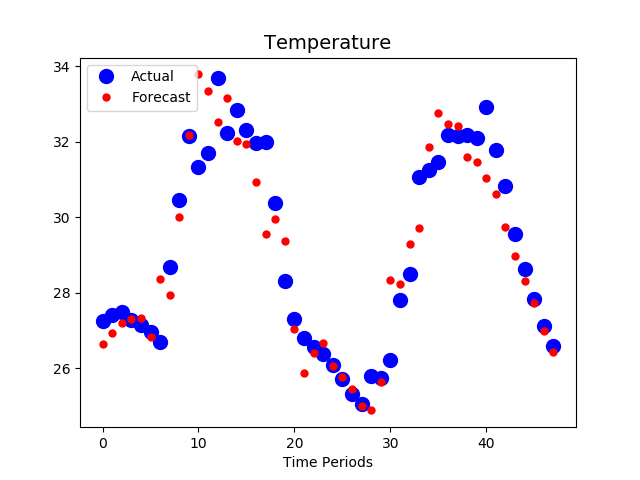
fig.savefig('TemSS.png',

facecolor=fig.get\_facecolor(),

edgecolor=fig.get\_edgecolor(),

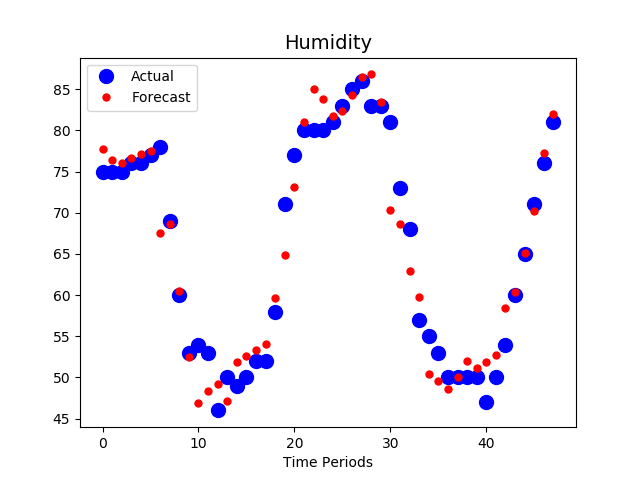
dpi = fig.get\_dpi())

#lưu lại đồ thị dưới tên TemSS.png xác để đánh giá mô hình dự đoán này.



Hình 5‑1. Đồ thị so sánh nhiệt độ dự báo và thực tế trong 48 giờ

Tương tự, khi đưa dữ liệu đầu vào là độ ẩm (cột “Humidity”) ta được kết quả:



Hình 5‑2. Đồ thị so sánh độ ẩm tương đối dự báo và thực tế trong 48 giờ

Trên đồ thị, chấm tròn nhỏ màu đỏ biểu thị giá trị thời tiết mà mô hình đã dự đoán (Forecast), còn chấm tròn ta hơn màu xanh biểu thị giá trị thời tiết thực tế (Actual). Chi tiết hơn, ta thực hiện một bảng tính toán độ chính xác của phép dự đoán nhiệt độ:

Bảng 5‑1. Tính toán độ chính xác của dự báo nhiệt độ trong 48 giờ tới

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hour | Actual | Forecast | Accuracy | Hour | Actual | Forecast | Accuracy |
| 1 | 27.25 | 26.63576 | 97.75 | 25 | 26.08 | 26.05598 | 99.91 |
| 2 | 27.41 | 26.93783 | 98.28 | 26 | 25.71 | 25.77284 | 99.76 |
| 3 | 27.48 | 27.20573 | 99.00 | 27 | 25.32 | 25.44084 | 99.52 |
| 4 | 27.29 | 27.29822 | 99.97 | 28 | 25.05 | 24.99861 | 99.79 |
| 5 | 27.14 | 27.33033 | 99.30 | 29 | 25.8 | 24.88192 | 96.44 |
| 6 | 26.95 | 26.82654 | 99.54 | 30 | 25.73 | 25.62804 | 99.60 |
| 7 | 26.7 | 28.3722 | 93.74 | 31 | 26.23 | 28.32658 | 92.01 |
| 8 | 28.68 | 27.94354 | 97.43 | 32 | 27.81 | 28.23916 | 98.46 |
| 9 | 30.45 | 30.01377 | 98.57 | 33 | 28.49 | 29.30293 | 97.15 |
| 10 | 32.16 | 32.1727 | 99.96 | 34 | 31.07 | 29.70434 | 95.60 |
| 11 | 31.34 | 33.78504 | 92.20 | 35 | 31.25 | 31.87038 | 98.01 |
| 12 | 31.71 | 33.33924 | 94.86 | 36 | 31.46 | 32.75747 | 95.88 |
| 13 | 33.7 | 32.51085 | 96.47 | 37 | 32.18 | 32.47599 | 99.08 |
| 14 | 32.23 | 33.16962 | 97.08 | 38 | 32.15 | 32.42788 | 99.14 |
| 15 | 32.84 | 32.00715 | 97.46 | 39 | 32.17 | 31.59319 | 98.21 |
| 16 | 32.31 | 31.93124 | 98.83 | 40 | 32.1 | 31.47199 | 98.04 |
| 17 | 31.98 | 30.93679 | 96.74 | 41 | 32.92 | 31.04427 | 94.30 |
| 18 | 32 | 29.56342 | 92.39 | 42 | 31.77 | 30.60447 | 96.33 |
| 19 | 30.38 | 29.96316 | 98.63 | 43 | 30.83 | 29.72892 | 96.43 |
| 20 | 28.32 | 29.37918 | 96.26 | 44 | 29.55 | 28.97991 | 98.07 |
| 21 | 27.3 | 27.04627 | 99.07 | 45 | 28.63 | 28.322 | 98.92 |
| 22 | 26.8 | 25.87078 | 96.53 | 46 | 27.84 | 27.71996 | 99.57 |
| 23 | 26.55 | 26.40113 | 99.44 | 47 | 27.13 | 26.97831 | 99.44 |
| 24 | 26.38 | 26.67944 | 98.86 | 48 | 26.6 | 26.42832 | 99.35 |

Với tỉ lệ chính xác vào khoảng 96.75% chứng tỏ giải thuật ta sử dụng đáng tin cậy và không có sự sai lệch quá nhiều. Tuy nhiên, ở nhưng thời điểm có biên độ dao động lớn của nhiệt độ và độ ẩm, thì dữ liệu dự báo kém chính xác hơn. Nếu có thể thu thập được dữ liệu trong khoảng thời gian lâu hơn, lên đến trên 10 năm, thì kết quả dự báo sẽ được tối ưu hóa. Bên cạnh việc giảm thiểu chênh lệch giữa dự đoán và thực tế, thì trong thực tế hoàn toàn có thể xảy ra những dữ liệu bất thường, gây nhiễu đầu vào của bài toán nếu không trải qua bước tiền xử lý và làm ảnh hưởng lớn đến tính chính xác của kết quả. Đó cũng là một nhược điểm của mô hình Linear Regression.

**Tính toán dự trên dự báo và đưa ra khuyến nghị**

Xây dựng giải thuật dự đoán mưa như sau: từ giá trị nhiệt độ tại một thời điểm ta đánh giá trọng số cho nhiệt độ này theo bảng:

Bảng 5‑2. Đánh giá trọng số cho giá trị nhiệt độ

|  |  |
| --- | --- |
| Nhiệt độ (oC) | Trọng số |
| 40<t | 2 |
| 35<t≤40 | 2,5 |
| 32<t≤35 | 3 |
| 30<t≤32 | 4 |
| 26<t≤30 | 5 |
| 24<t≤26 | 4 |
| 18<t≤24 | 5 |
| 13<t≤18 | 3 |
| 8<t≤13 | 2 |
| 2<t≤8 | 2 |
| t≤2 | 1 |

Tương tự với giá trị độ ẩm tương đối, ta cũng xây dựng bảng trọng số như sau:

Bảng 5‑3. Đánh giá trọng số cho giá trị độ ẩm

|  |  |
| --- | --- |
| Độ ẩm (%) | Trọng số |
| h<40 | 0.5 |
| 40<h≤50 | 1 |
| 50<h≤55 | 1.5 |
| 55<h≤60 | 2 |
| 60<h≤65 | 2.5 |
| 65<h≤70 | 3 |
| 70<h≤75 | 3.5 |
| 75<h≤80 | 4 |
| 80<h≤85 | 5 |
| 85<h≤90 | 6 |
| h<95 | 7 |

Sau đó, chúng ta dự đoán mưa hay không mưa dựa theo công thức sau: Nếu 0.4 lần trọng số nhiệt độ cộng với 0.6 trọng số độ ẩm lớn hơn 6.0 thì đưa ra dự báo “mưa”, ngược lại dự báo “không mưa”. Tiếp đến, ta gán giá trị cho “dự báo trời nắng” theo bảng sau:

Bảng 5‑4. Đánh giá trời nắng dự vào số liệu

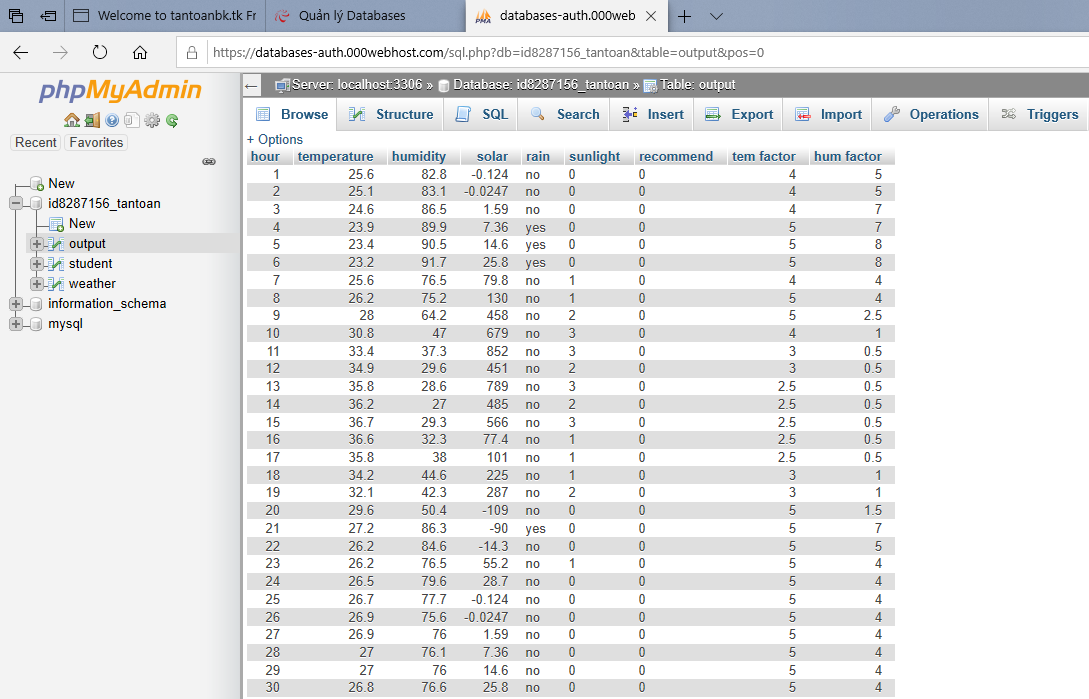
|  |  |
| --- | --- |
| Ánh sáng (W/m2) | Dự đoán |
| s≤50 | Trời tối |
| 50<s≤250 | Nắng nhẹ |
| 250<s≤500 | Trời nắng |
| 500<s | Nắng gắt |

Cuối cùng, ta tổng hợp số liệu và dự báo của 2 ngày 3/4/2019 và 4/4/2019:

Bảng 5‑5. Tổng hợp dữ liệu và kết quả dự báo

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Giờ | Nhiệt độ dự  đoán (oC) | Độ ẩm dự  đoán (%) | Ánh sáng dự  đoán (W/m2) | Dự đoán  mưa | Dự đoán  nắng | Trọng số  nhiệt độ | Trọng số  độ ẩm |
| 1 | 25.6 | 82.8 | 0 | no | trời tối | 4 | 5 |
| 2 | 25.1 | 83.1 | 0 | no | trời tối | 4 | 5 |
| 3 | 24.6 | 86.5 | 1.59482491 | no | trời tối | 4 | 6 |
| 4 | 23.9 | 89.9 | 7.355950356 | no | trời tối | 5 | 6 |
| 5 | 23.4 | 90.5 | 14.64730263 | yes | trời tối | 5 | 7 |
| 6 | 23.2 | 91.7 | 25.78609085 | yes | trời tối | 5 | 7 |
| 7 | 25.6 | 76.5 | 79.7671051 | no | nắng nhẹ | 4 | 4 |
| 8 | 26.2 | 75.2 | 130.3968964 | no | nắng nhẹ | 5 | 4 |
| 9 | 28 | 64.2 | 458.3119507 | no | trời nắng | 5 | 2.5 |
| 10 | 30.8 | 47 | 679.1079712 | no | nắng gắt | 4 | 1 |
| 11 | 33.4 | 37.3 | 852.4689941 | no | nắng gắt | 3 | 0.5 |
| 12 | 34.9 | 29.6 | 450.8434448 | no | trời nắng | 3 | 0.5 |
| 13 | 35.8 | 28.6 | 788.7572021 | no | nắng gắt | 2.5 | 0.5 |
| 14 | 36.2 | 27 | 484.7685852 | no | trời nắng | 2.5 | 0.5 |
| 15 | 36.7 | 29.3 | 565.8621826 | no | nắng gắt | 2.5 | 0.5 |
| 16 | 36.6 | 32.3 | 77.42417908 | no | nắng nhẹ | 2.5 | 0.5 |
| 17 | 35.8 | 38 | 100.7989502 | no | nắng nhẹ | 2.5 | 0.5 |
| 18 | 34.2 | 44.6 | 225.3500977 | no | nắng nhẹ | 3 | 1 |
| 19 | 32.1 | 42.3 | 287.2237244 | no | trời nắng | 3 | 1 |
| 20 | 29.6 | 50.4 | 0 | no | trời tối | 5 | 1.5 |
| 21 | 27.2 | 86.3 | 0 | no | trời tối | 5 | 6 |
| 22 | 26.2 | 84.6 | 0 | no | trời tối | 5 | 5 |
| 23 | 26.2 | 76.5 | 55.23558426 | no | nắng nhẹ | 5 | 4 |
| 24 | 26.5 | 79.6 | 28.67375374 | no | trời tối | 5 | 4 |
| 25 | 26.7 | 77.7 | 0 | no | trời tối | 5 | 4 |
| 26 | 26.9 | 75.6 | 0 | no | trời tối | 5 | 4 |
| 27 | 26.9 | 76 | 1.59482491 | no | trời tối | 5 | 4 |
| 28 | 27 | 76.1 | 7.355950356 | no | trời tối | 5 | 4 |
| 29 | 27 | 76 | 14.64730263 | no | trời tối | 5 | 4 |
| 30 | 26.8 | 76.6 | 25.78609085 | no | trời tối | 5 | 4 |
| 31 | 29.2 | 67.5 | 79.7671051 | no | nắng nhẹ | 5 | 3 |
| 32 | 29.6 | 66.4 | 130.3968964 | no | nắng nhẹ | 5 | 3 |
| 33 | 30.5 | 61.7 | 355.2525635 | no | trời nắng | 4 | 2.5 |
| 34 | 32.1 | 53.5 | 504.1925049 | no | nắng gắt | 3 | 1.5 |
| 35 | 33.9 | 46.9 | 1056.568359 | no | nắng gắt | 3 | 1 |
| 36 | 33.6 | 45.9 | 900.5276489 | no | nắng gắt | 3 | 1 |
| 37 | 32.8 | 48.4 | 376.4900208 | no | trời nắng | 3 | 1 |
| 38 | 33.4 | 45.1 | 792.3029785 | no | nắng gắt | 3 | 1 |
| 39 | 33 | 51.4 | 306.3743591 | no | trời nắng | 3 | 1.5 |
| 40 | 32.1 | 50.9 | 447.4032288 | no | trời nắng | 3 | 1.5 |
| 41 | 31.2 | 54.8 | 84.39355469 | no | nắng nhẹ | 4 | 1.5 |
| 42 | 29.9 | 57 | 102.3521194 | no | nắng nhẹ | 5 | 2 |
| 43 | 29.3 | 60.3 | 116.6161499 | no | nắng nhẹ | 5 | 2.5 |
| 44 | 28.7 | 65.2 | 119.4229507 | no | nắng nhẹ | 5 | 3 |
| 45 | 28.1 | 68.3 | 0 | no | trời tối | 5 | 3 |
| 46 | 27.4 | 74.1 | 0 | no | trời tối | 5 | 3.5 |
| 47 | 26.8 | 78.2 | 65.45771027 | no | nắng nhẹ | 5 | 4 |
| 48 | 26.3 | 80.5 | 56.33896255 | no | nắng nhẹ | 5 | 5 |

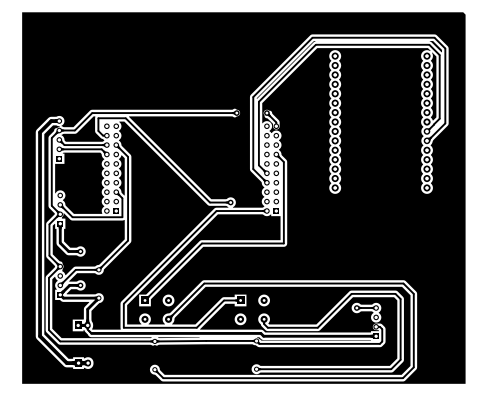
Khi có được bảng tổng hợp, ta tiến hành import vào cơ sở dữ liệu trên hosting



Hình 5‑3. Dữ liệu bảng “output” trên database

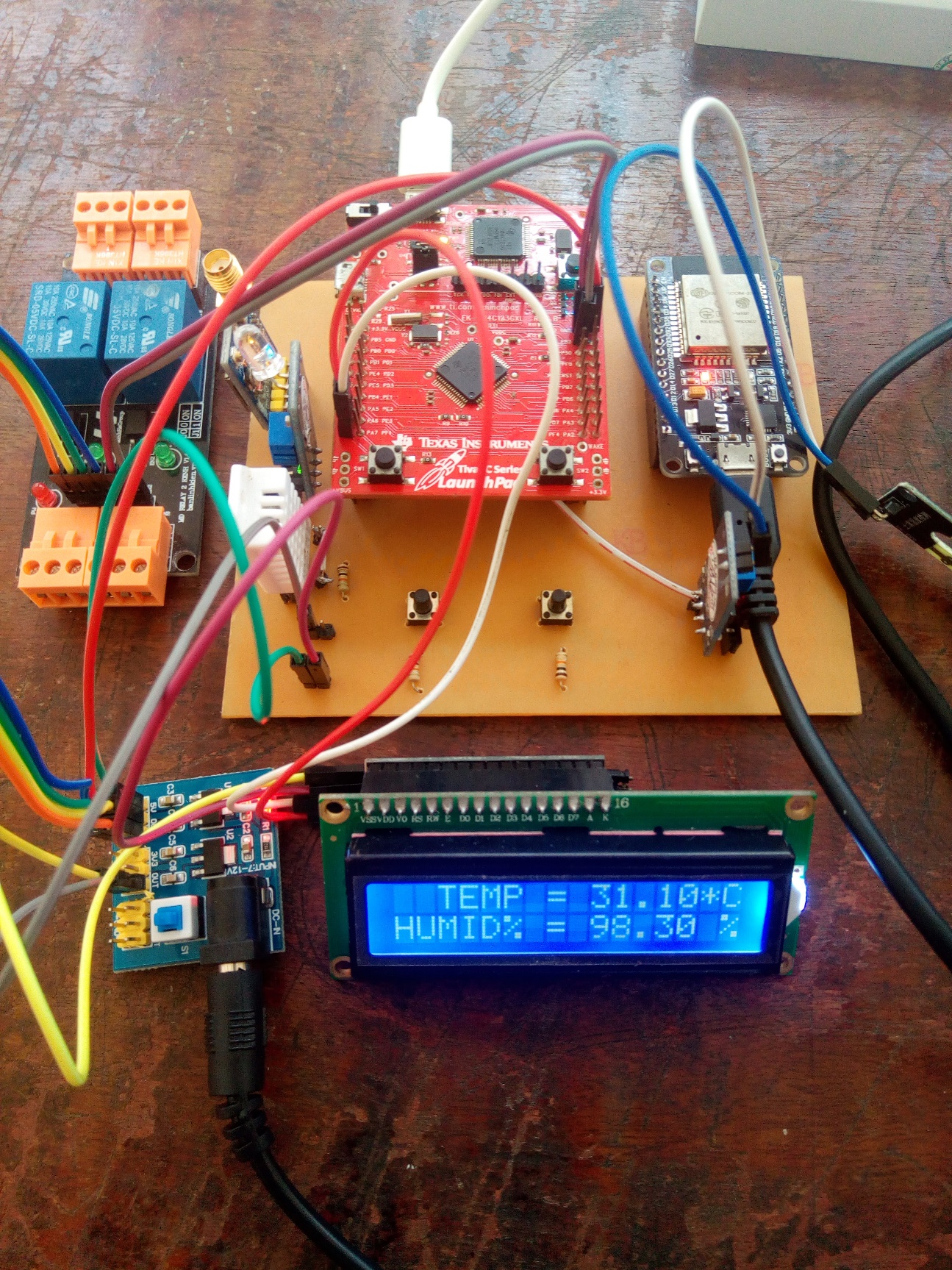
## Kết quả thực hiện phần cứng

Sau khi đã thiết kế PCB bằng Altium, ta xuất file PDF và tiến hành in mạch:



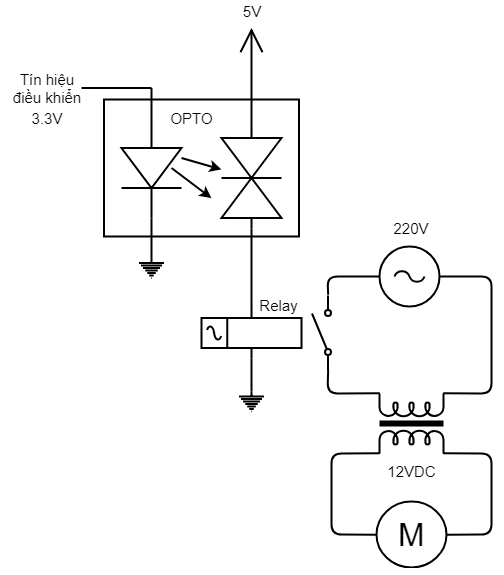
Hình 5‑4. Mạch in PCB

Sau các công đoạn ủi mạch, ngâm mạch, khoan lỗ chân pin, hàn mạch và phủ nhựa thông ta được bảng mạch có kích thước 10x12cm. Tiến hành lắp các linh kiện: Tiva C LaunchPad, module Wifi ESP32, DHT22, cảm biến độ ẩm đất, cảm biến cường độ sáng, module GPS NEO-6M, module relay 2 kênh, LCD 1602, module cấp nguồn cùng với dây nối, sản phẩm sau cùng có hình thực tế như bên dưới:



Hình 5‑5. Mạch thực tế

Mạch thành phẩm đầu tiên còn rất nhiều khuyết điểm như: mối hàn khá vụng, nút nhấn thiếu ổn định (cần nghiên cứu lại điện trở và tụ điện nối GND), phải sử dụng nhiều dây jump cũng như các dây ghim module. Cộng với việc các thiết bị lắp sau relay cũng cần thêm một số dây cấp nguồn khiến cho mạch trông khá rối rắm do sử dụng rất nhiều dây nối. Một vấn đề nữa mà chúng em gặp phải khi lắp thiết bị là khi nối động cơ bơm nước 12V vào relay, có sự biến thiên lớn giữa dòng khởi động và dòng hoạt động và sự xuất hiện của dòng điện do quán tính của rotor tạo ra sau khi ngắt điện. Các dòng điện đó gây ảnh hưởng đến mạch điều khiển. Chúng em xử lý bằng cách nối nguồn điện với ngõ vào của adapter 220VAC/12VDC thông qua relay, nguồn cấp sau khi biến áp mới được đưa vào động cơ bơm nước 12V.

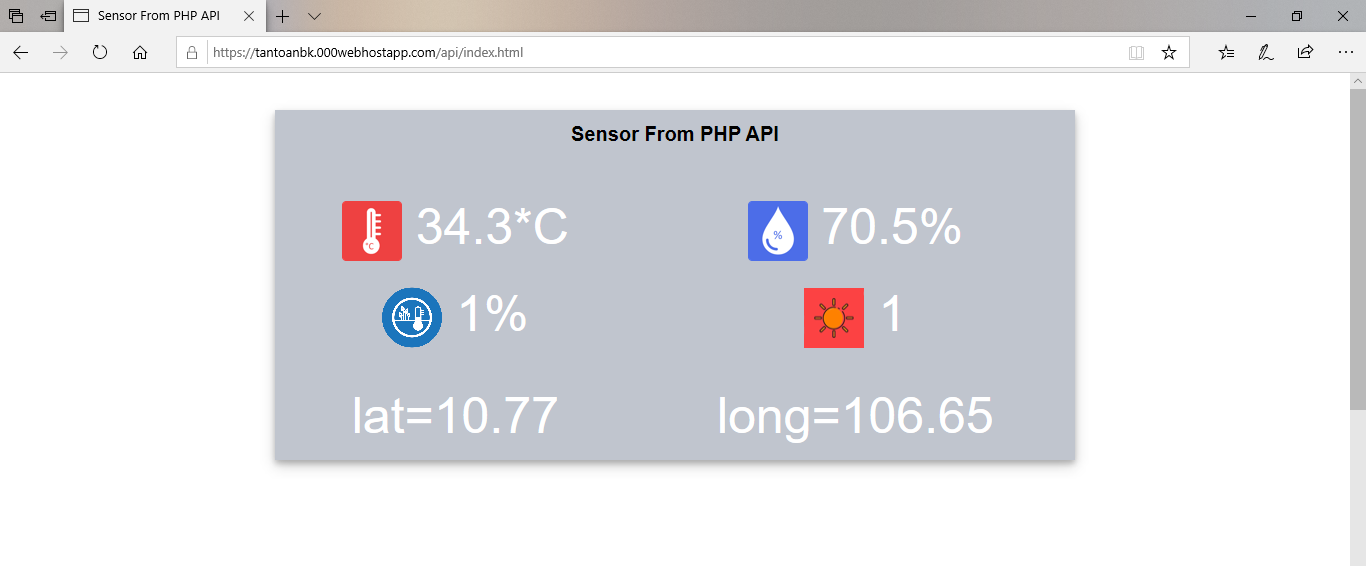


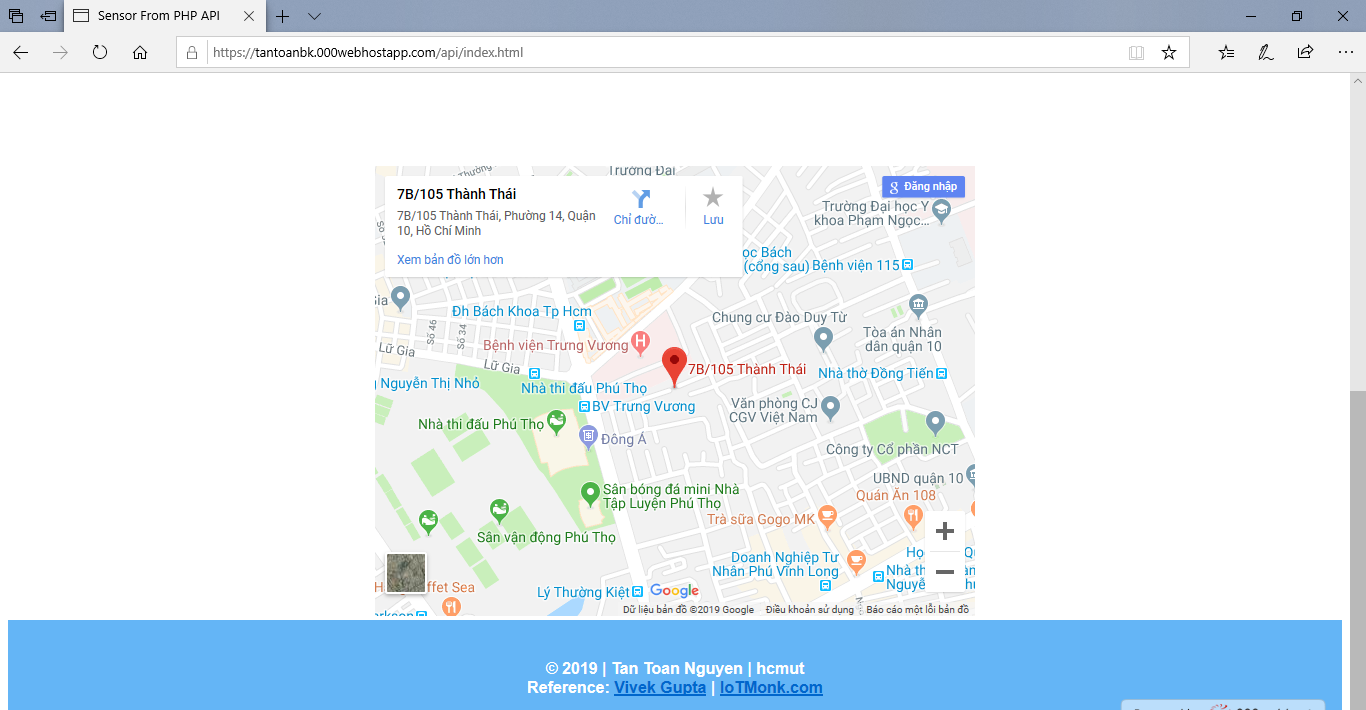
Hình 5‑6. Sơ đồ nối thiết bị với relay điều khiển

Các thiết bị khác cũng được nối tương tự như sơ đồ trên.

## Kết quả xây dựng Website và ứng dụng Smartphone

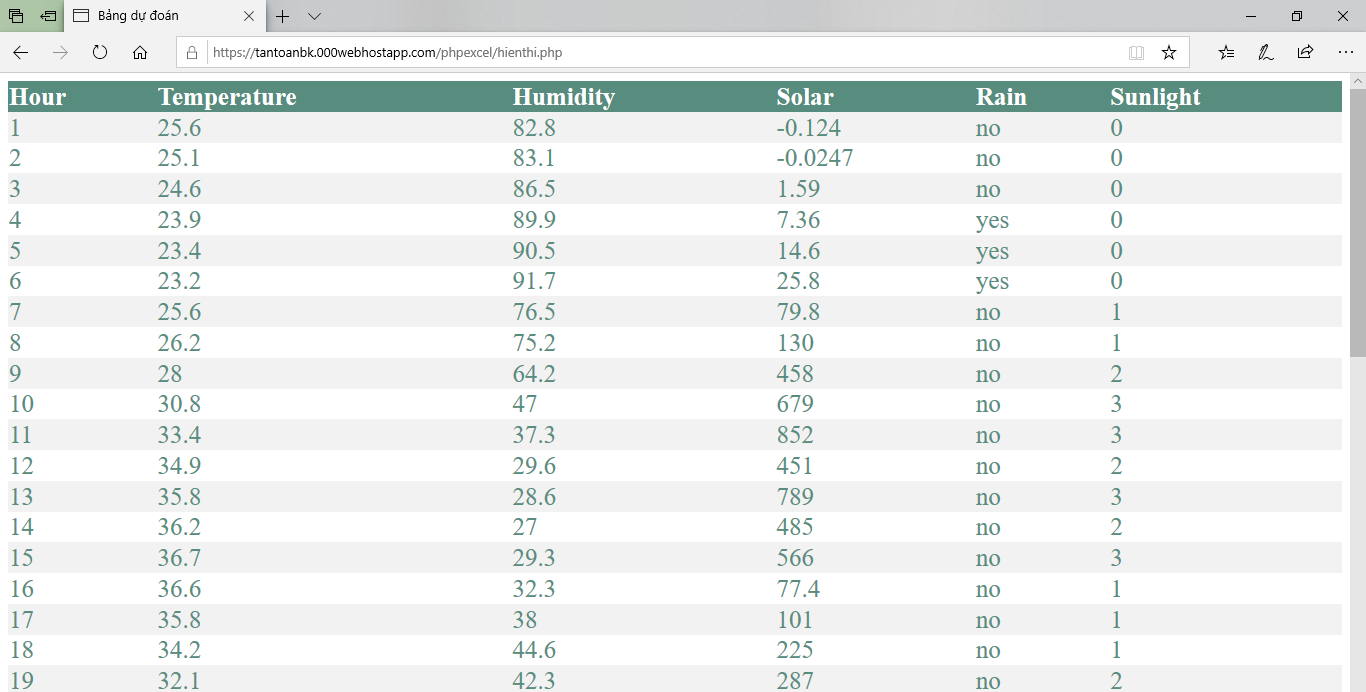
Giao diện hiển thị giá trị cuối cùng của bảng “weather” lên Website, gồm có: nhiệt độ, độ ẩm không khí, độ ẩm đất, ánh nắng mặt trời, tọa độ đo đạc.





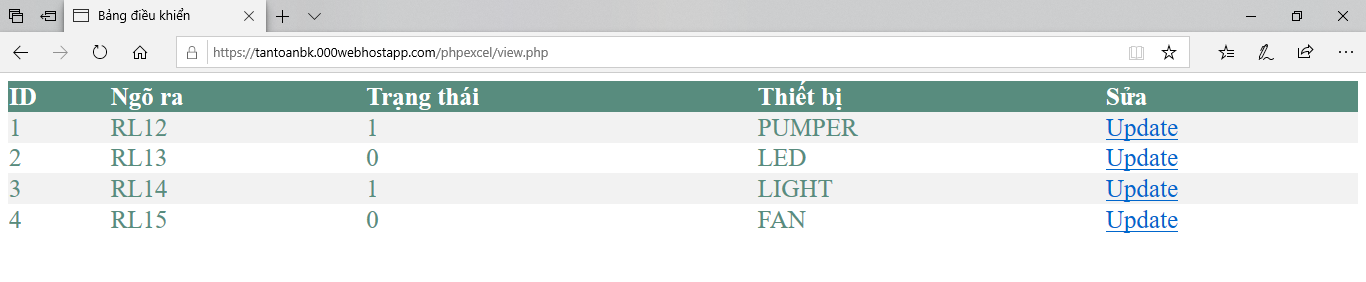
Hình 5‑7. Giao diện Website hiển thị giá trị đo đạc gần nhất

Website “Bảng dự đoán” https://tantoanbk.000webhostapp.com/phpexcel/hienthi.php hiển thị bảng “output” trong cơ sở dữ liệu.



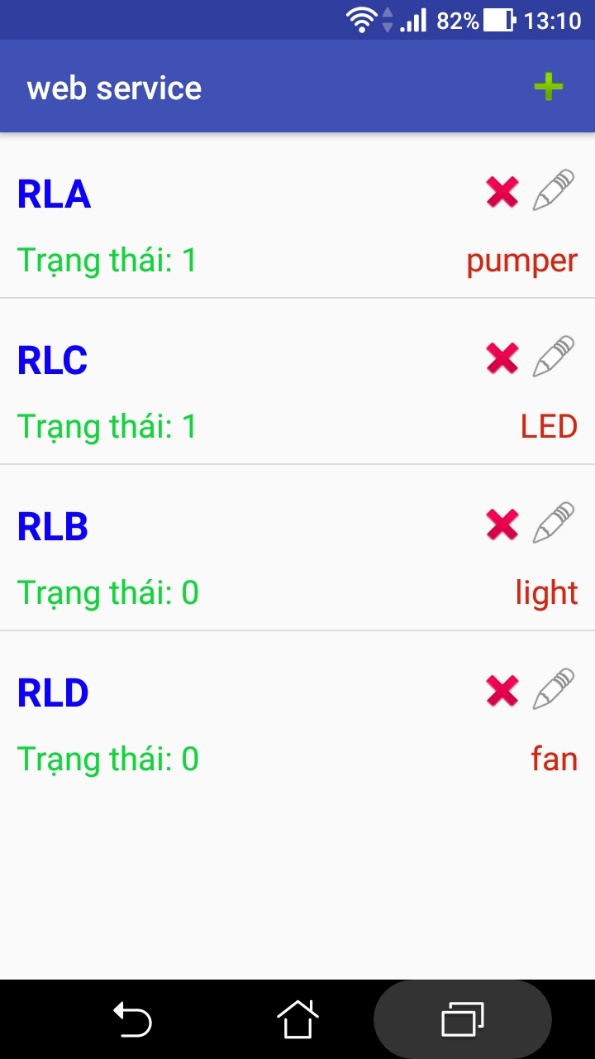
Hình 5‑8. Giao diện Website hiển thị số liệu dự báo

Website “Bảng điều khiển” https://tantoanbk.000webhostapp.com/phpexcel/view.phphiển thị thông tin điều khiển thiết bị.



Hình 5‑9. Giao diện Website điều khiển thiết bị

Cài đặt file “web service.apk” vào điện thoại di động có hệ điều hành Android 4.0 trở lên để có thể chạy ứng dụng điều khiển thiết bị qua điện thoại đó.

****

Hình 5‑10. Giao diện ứng dụng điều khiển thiết bị trên Smartphone

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Dựa vào kết quả đạt được, chúng em đã hoàn thành những công việc:

* Thu thập dữ liệu từ các cảm biến gồm có: nhiệt độ - độ ẩm không khí, độ ẩm đất, cường độ ánh sáng, vị trí đo đạc.
* Xử lý thông tin thu thập được, đưa lên cơ sở dữ liệu mạng và xây dựng giao diện Website hiển thị các thông tin đó.
* Tiến hành dự báo bằng máy học, sử dụng giải thuật Linear Regression và hiển thị số liệu dự báo trên Website.
* Xây dựng giao diện điều khiển thiết bị qua mạng Internet bằng trang web và ứng dụng Smartphone.

Ưu điểm của hệ thống:

* Đề tài có tính ứng dụng cao, có thể điều khiển từ xa nhiều loại thiết bị thông qua các relay đóng/ngắt.
* Đáp ứng được nhu cầu về cung cấp thông tin và dự báo một số yếu tố của thời tiết tại một địa điểm cụ thể.
* Sản phẩm nhỏ gọn, giao diện dễ nhìn, phù hợp với các mục đích nuôi trồng với quy mô nhỏ trong nông nghiệp.

Khuyết điểm của hệ thống:

* Do hệ thống sử dụng kết nối mạng Internet nên rất phụ thuộc vào điều kiện chất lượng của mạng Wifi mà hệ thống kết nối đến.
* Cơ sở dữ liệu của host thiếu ổn định, các tác vụ truy cập vào Website, xử lý dữ liệu trên server chưa nhanh, giao diện điều khiển chưa tiện dụng gây khó khăn cho thao tác điều khiển thiết bị.
* Việc thu thập dữ liệu thường hay gặp lỗi làm cho quá trình ghi nhận và hiển thị thông tin môi trường không được liên tục, dẫn đến kết quả vẫn chưa thể sử dụng dữ liệu do chính hệ thống thu thập mà phải tải dữ liệu từ nguồn khác.
* Giải thuật dự báo chưa hoàn chỉnh: chưa kết hợp được nhiều loại dữ liệu vào trong một tiến trình training, chưa thực hiện thao tác lọc dữ liệu ngõ vào và ngõ ra, gây sai lệch thông tin và ảnh hưởng đến kết quả dự báo.

## Hướng phát triển

* Kết hợp nhiều loại dữ liệu vào trong một tiến trình training nhằm tăng tính liên hệ giữa các yếu tố của môi trường để cho ra kết quả chính xác hơn; thực hiện thao tác lọc dữ liệu ngõ vào và ngõ ra, loại bỏ dữ liệu dây nhiễu, tiến tới tối ưu hóa kết quả dự báo.
* Sử dụng một cơ sở dữ liệu khác tốt hơn (chẳng hạn như Google Firebase) nhằm đơn giản hóa các thao tác với cơ sở dữ liệu, từ đó phát triển Website và ứng dụng nhanh chóng, ổn định.
* Xây dựng chức năng hẹn giờ tự động bật/tắt thiết bị, cho phép thiết bị hoạt động trong thời gian phù hợp với mục đích sử dụng dựa trên dữ liệu dự báo và thông tin thu thập từ môi trường. Cài đặt chế độ cho phép thiết lập tần số thu thập dữ liệu, có thể điều chỉnh trong khoảng từ 10 phút đến 1 giờ thì thu thập thông tin môi trường một lần.
* Thiết kế Website và ứng dụng Smartphone trực quan hơn để dễ thao tác điều khiển thiết bị. Xây dựng hệ thống quản lý người dùng, cấp quyền sử dụng theo các tài khoản lưu trên server nhằm gói gọn quyền hạn cập nhật và điều khiển, tăng tính bảo mật.
* Ngoài ra ta có thể thực hiện việc gửi trả trạng thái của thiết bị ngược về giao diện quản lý để từ đó dễ dàng nắm bắt được tình trạng hoạt động của thiết bị mà không cần trực tiếp quan sát. Đồng thời cũng đánh giá được khả năng làm việc, đơn giản hơn trong việc chẩn đoán và khắc phục sự cố nếu hệ thống gặp vấn đề trong hoạt động.

# TÀI LIỆU THAM KHẢO

[1] V. H. Tiệp, "Bài 1: Giới thiệu về Machine Learning," in *Machine Learning cơ bản*, ed, Dec 26, 2016

[2] V. H. Tiệp, "Bài 3: Linear Regression," in *Machine Learning cơ bản*, ed, Dec 28, 2016

[3] T. Đ. Tâm, "Bắt đầu với Machine Learning thông qua Tensorflow," ed, 17/07/2018.

[4] Lavender, "Machine Learning - Thử làm Nhà Thiên Văn Dự Báo Thời Tiết," in *viblo.asia*, ed, Jan 30th, 2018.

[5] V. Gupta, "Creating your own IoT Cloud from scratch using php, Mysql, ESP12E," in *iotmonk.com*, ed, 28 Oct, 2017.

[6] T. h. K. Phạm, "Lập trình Android: Webservice Giới thiệu," in *Lập trình Android*, ed, 2017.

# PHỤ LỤC

## Tìm hiểu về Wifi

Wi-Fi viết tắt từ Wireless Fidelity hay mạng 802.11 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống như điện thoại di động, truyền hình và radio. Hệ thống này đã hoạt động ở một số sân bay, quán café, thư viện hoặc khách sạn. Hệ thống cho phép truy cập Internet tại những khu vực có sóng của hệ thống này, hoàn toàn không cần đến cáp nối. Ngoài các điểm kết nối công cộng (hotspots), WiFi có thể được thiết lập ngay tại nhà riêng.

Tên gọi 802.11 bắt nguồn từ viện IEEE (Institute of Electrical and Electronics Engineers). Viện này tạo ra nhiều chuẩn cho nhiều giao thức kỹ thuật khác nhau, và nó sử dụng một hệ thống số nhằm phân loại chúng; 6 chuẩn thông dụng của WiFi hiện nay là 802.11a/b/g/n/ac/ad.

Sóng Wifi:

Các sóng vô tuyến sử dụng cho WiFi gần giống với các sóng vô tuyến sử dụng cho thiết bị cầm tay, điện thoại di động và các thiết bị khác. Nó có thể chuyển và nhận sóng vô tuyến, chuyển đổi các mã nhị phân 1 và 0 sang sóng vô tuyến và ngược lại. Tuy nhiên, sóng WiFi có một số khác biệt so với các sóng vô tuyến khác ở chỗ: Chúng truyền và phát tín hiệu ở tần số 2.4 GHz hoặc 5 GHz. Tần số này cao hơn so với các tần số sử dụng cho điện thoại di động, các thiết bị cầm tay và truyền hình. Tần số cao hơn cho phép tín hiệu mang theo nhiều dữ liệu hơn. Chúng dùng chuẩn 802.11:

- Chuẩn 802.11b là phiên bản đầu tiên trên thị trường. Đây là chuẩn chậm nhất và rẻ tiền nhất, và nó trở nên ít phổ biến hơn so với các chuẩn khác. 802.11b phát tín hiệu ở tần số 2.4 GHz, nó có thể xử lý đến 11 megabit/giây, và nó sử dụng mã CCK (complimentary code keying).

- Chuẩn 802.11g cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn 802.11b, tốc độ xử lý đạt 54 megabit/giây. Chuẩn 802.11g nhanh hơn vì nó sử dụng mã OFDM (orthogonal frequency-division multiplexing), một công nghệ mã hóa hiệu quả hơn.

- Chuẩn 802.11a phát ở tần số 5 GHz và có thể đạt đến 54 megabit/ giây. Nó cũng sử dụng mã OFDM. Những chuẩn mới hơn sau này như 802.11n còn nhanh hơn chuẩn 802.11a, nhưng 802.11n vẫn chưa phải là chuẩn cuối cùng.

- Chuẩn 802.11n cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn 802.11a, tốc độ xử lý đạt 300 megabit/giây.

- Chuẩn 802.11ac phát ở tần số 5 GHz.

- Chuẩn 802.11ad phát ở tần số 60 GHz.

WiFi có thể hoạt động trên cả ba tần số và có thể nhảy qua lại giữa các tần số khác nhau một cách nhanh chóng. Việc nhảy qua lại giữa các tần số giúp giảm thiểu sự nhiễu sóng và cho phép nhiều thiết bị kết nối không dây cùng một lúc. Tuy nhiên mạng không dây luôn vấp phải những vấn đề tồn tại khiến nó chưa thể phát triển và thay thế cho toàn bộ cách kết nối hiện nay. Nhưng vấn đề tiện lợi cho người dùng sẽ luôn đẩy công nghệ không dây ngày càng phát triển và chúng ta có thể khẳng định rằng mạng không dây sẽ ngày càng phát triển và thay thế dần dần mạng sử dụng cáp hiện nay.

Phạm vi Wifi:

Phạm vi tín hiệu Wifi phụ thuộc vào bằng tần, công suất đầu ra, độ lợi và loại anten cũng như các kỹ thuật điều chế. Line-of-sight là hướng dẫn thu nhỏ nhưng sự khúc xạ và phản xạ gây ra ảnh hưởng đáng kể.

Một điểm truy cập phù hợp với một trong hai chuẩn 802.11b hay 802.11g , sử dụng ăng-ten chứng khoán có thể có một phạm vi 100 m (330 ft). Các đài phát thanh cùng với một ăng-ten parabol bán bên ngoài (15 dB đạt được) có thể có một phạm vi hơn 20 dặm.

## Tìm hiểu về GPS

Hệ thống Định vị Toàn cầu (tiếng Anh: Global Positioning System - GPS) là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh. Tuy được quản lý bởi Bộ Quốc phòng Hoa Kỳ, chính phủ Hoa Kỳ cho phép mọi người trên thế giới sử dụng một số chức năng của GPS miễn phí, bất kể quốc tịch nào.

Phân loại:

Hệ thống định vị toàn cầu của Mỹ là hệ dẫn đường dựa trên một mạng lưới 26 quả vệ tinh được Bộ Quốc phòng Hoa Kỳ đặt trên quỹ đạo không gian. GPS ban đầu chỉ dành cho các mục đích quân sự, nhưng từ năm 1980 chính phủ Mỹ cho phép sử dụng trong dân sự. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên Trái Đất, 24 giờ một ngày. Không mất phí thuê bao hoặc mất tiền trả cho việc thiết lập sử dụng GPS nhưng phải tốn tiền không rẻ để mua thiết bị thu tín hiệu và phần mềm nhúng hỗ trợ.

Sự hoạt động của GPS:

Các vệ tinh GPS bay vòng quanh Trái Đất hai lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận thông tin này và bằng phép tính lượng giác tính được chính xác vị trí của người dùng. Về bản chất máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian nhận được chúng. Sai lệch về thời gian cho biết máy thu GPS ở cách vệ tinh bao xa. Rồi với nhiều quãng cách đo được tới nhiều vệ tinh máy thu có thể tính được vị trí của người dùng và hiển thị lên bản đồ điện tử của máy.

Máy thu phải nhận được tín hiệu của ít nhất ba vệ tinh để tính ra vị trí hai chiều (kinh độ và vĩ độ) và để theo dõi được chuyển động. Khi nhận được tín hiệu của ít nhất 4 vệ tinh thì máy thu có thể tính được vị trí ba chiều (kinh độ, vĩ độ và độ cao). Một khi vị trí người dùng đã tính được thì máy thu GPS có thể tính các thông tin khác, như tốc độ, hướng chuyển động, bám sát di chuyển, khoảng hành trình, quãng cách tới điểm đến, thời gian Mặt Trời mọc, lặn và nhiều thứ khác nữa.

Tín hiệu GPS:

Các vệ tinh GPS phát hai tín hiệu vô tuyến công suất thấp dải L1 và L2. (dải L là phần sóng cực ngắn của phổ điện từ trải rộng từ 0,39 tới 1,55 GHz). GPS dân sự dùng tần số L1 1575.42 MHz trong dải UHF. Tín hiệu truyền trực thị, có nghĩa là chúng sẽ xuyên qua mây, thuỷ tinh và nhựa nhưng không qua phần lớn các đối tượng cứng như núi và nhà.

L1 chứa hai mã "giả ngẫu nhiên" (pseudo random), đó là mã Protected (P) và mã Coarse/Acquisition (C/A). Mỗi một vệ tinh có một mã truyền dẫn nhất định, cho phép máy thu GPS nhận dạng được tín hiệu. Mục đích của các mã tín hiệu này là để tính toán khoảng cách từ vệ tinh đến máy thu GPS.

Tín hiệu GPS chứa ba mẩu thông tin khác nhau – mã giả ngẫu nhiên, dữ liệu thiên văn và dữ liệu lịch. Mã giả ngẫu nhiên đơn giản chỉ là mã định danh để xác định được quả vệ tinh nào là phát thông tin nào. Có thể nhìn số hiệu của các quả vệ tinh trên trang vệ tinh của máy thu Garmin để biết nó nhận được tín hiệu của quả nào.

Dữ liệu thiên văn cho máy thu GPS biết quả vệ tinh ở đâu trên quỹ đạo ở mỗi thời điểm trong ngày. Mỗi quả vệ tinh phát dữ liệu thiên văn chỉ ra thông tin quỹ đạo cho vệ tinh đó và mỗi vệ tinh khác trong hệ thống.

Dữ liệu lịch được phát đều đặn bởi mỗi quả vệ tinh, chứa thông tin quan trọng về trạng thái của vệ tinh (hoạt động tốt hay không), ngày giờ hiện tại. Phần này của tín hiệu là cốt lõi để phát hiện ra vị trí.