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**

**XÂY DỰNG HỆ THỐNG ĐO ĐẠC, GIÁM SÁT VÀ ĐIỀU HƯỚNG TRONG HỆ THỐNG NĂNG LƯỢNG MẶT TRỜI ĐỘC LẬP**

|  |  |
| --- | --- |
| **SVTH** | **MSSV** |
| **NGUYỄN HỮU SỸ** | **1413345** |
| **PHẠM TRUNG VINH** | **41204560** |
| **GVHD: PGS. TS HÀ HOÀNG KHA** | |

**TP. HỒ CHÍ MINH, 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: **VIỄN THÔNG**

N**HIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. HỌ VÀ TÊN: NGUYỄN HỮU SỸ MSSV: 1413345

PHẠM TRUNG VINH MSSV: 41204560

1. NGÀNH: LỚP :
2. Đề tài:
3. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):

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

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

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

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

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

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

5. Ngày giao nhiệm vụ luận văn: ...............................

6. Ngày hoàn thành nhiệm vụ: ...................................

1. 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***

Chúng em xin chân thành cảm ơn tất cả quý thầy cô đã dạy chúng em từ những ngày đầu còn là sinh viên năm nhất, cảm ơn các thầy cô ở ngành Điện Tử Viễn Thông, Khoa Điện – Điện Tử đã truyền đạt cho chúng em nhiều kiến thức để thực hiện luận văn này và cũng như một hành trang vững chãi hơn, để sẵn sàng hơn khi bước chân ra khỏi Bách Khoa đối mặt với bộn bề cuộc sống .

Và đặc biệt em xin gởi lời cảm ơn, tri ân chân thành nhất đến Thầy Hà Hoàng Kha, vì đã tận tình hướng dẫn cho chúng em hoàn thành đề tài này. Trong suốt quá trình làm đề tài, Thầy đã giúp đỡ và hỗ trợ chúng em hết mình, bên cạnh đó Thầy còn truyền đạt thêm cho chúng em nhiều kiến thức chuyên sâu về đề tài mà chúng em còn thiếu sót. Chúng em cảm thấy thật sự may mắn vì đã được một giảng viên có tâm huyết với nghề, với sinh viên như thầy hướng dẫn. Hi vọng trong tương lai nếu có cơ hội, sẽ được quay về báo đáp công ơn của thầy cũng như của các giảng viên trong khoa Điện - Điện Tử.

Do thời gian và kiến thức còn hạn chế nên luận văn sẽ không tránh khỏi những thiếu sót, bất cập, chúng em rất mong nhận được sự đóng góp từ quý Thầy Cô để hoàn thiện hơn.

Em xin chân thành cảm ơn!

*Tp. Hồ Chí Minh, ngày tháng năm .*

Sinh viên

***TÓM TẮT LUẬN VĂN***

Luận văn này trình bày về thiết kế và thi công hệ thống giám sát và điều hướng năng lượng mặt trời. Sử dụng Arduino Uno để điều hướng kép hệ thống năng lượng mặt trời, tự động xoay theo ánh nắng. Sử dụng Node MCU ESP8266 để đo đạc, tính toán các thông số của hệ thống: điện áp, công suất pin, nhiệt độ pin, vị trí GPS và truyền dữ liệu về server, xử lí dữ liệu và cung cấp thông tin đến máy khách (laptop, smartphone,..), theo dõi các thông số kỹ thuật và gửi email thông báo cho Quản lý hệ thống (Admin) khi có những thây đổi bất thường; trên môi trường internet qua giao thức HTTP, điều hành và lưu trữ toàn bộ dữ liệu hoạt động trong mạng. Thiết kế web để hiển thị trực quan liên tục lên một website, smartphone.

Luận văn gồm 8 chương:

* 1. Giới thiệu: Giới thiệu tổng quát về hệ thống năng lượng mặt trời, tình hình nghiên cứu hiện nay và các ứng dụng của đề tài.
  2. Lý Thuyết: Trình bày các vấn đề lý thuyết liên quan đến luận văn như mạch phân áp, cảm biến nhiệt độ, module GPS, động cơ servo, các vấn đề về chuẩn giao thức http, webserver và web service.
  3. Thiết kế và thực hiện phần cứng
  4. Thiết kế và thực hiện phần mềm
  5. Kết quả thực hiện
  6. Kết luận và hướng phát triển
  7. Tài liệu tham khảo
  8. Phụ lục

**MỤC LỤC**

[1. GIỚI THIỆU 11](#_Toc10904779)

[1.1 Tổng quan đề tài 11](#_Toc10904780)

[1.2 Tình hình nguyên cứu trong và ngoài nước 12](#_Toc10904781)

[1.2.1 Tình hình nguyên cứu ngoài nước: 12](#_Toc10904782)

[1.2.2 Tình hình nguyên cứu trong nước: 13](#_Toc10904783)

[1.3 Nhiệm vụ luận văn 14](#_Toc10904784)

[2. LÝ THUYẾT 17](#_Toc10904785)

[2.1 Tổng quan thiết kế 17](#_Toc10904786)

[2.2 Tổng quan về năng lượng mặt trời 19](#_Toc10904787)

[2.2.1 Ưu, nhược điểm của năng lượng mặt trời 19](#_Toc10904788)

[2.2.2 Một số hệ thống năng lượng mặt trời 20](#_Toc10904789)

[2.3 Giới thiệu về SoC ESP8266 22](#_Toc10904790)

[2.3.1 Giới thiệu 22](#_Toc10904791)

[2.3.2 Các loại Module ESP8266 25](#_Toc10904792)

[2.3.2.1 ESP-WROOM-02: 25](#_Toc10904793)

[2.3.2.2 ESP-01: 26](#_Toc10904794)

[2.3.2.3 ESP-12E: 27](#_Toc10904795)

[2.3.1 Sơ lược các chức năng trong luận văn 28](#_Toc10904796)

[2.3.1.1 GPIO 29](#_Toc10904797)

[2.3.1.2 UART 31](#_Toc10904798)

[2.3.1.3 SERIAL 32](#_Toc10904799)

[2.3.1.4 Analog/ADC trong ESP8266 34](#_Toc10904800)

[2.3.1.5 Kết nối Wifi 35](#_Toc10904801)

[2.3.1.6 Lập trình cho ESP8266 36](#_Toc10904802)

[2.4 Giới thiệu về ARDUINO UNO 42](#_Toc10904803)

[2.4.1 Giới thiệu chung 42](#_Toc10904804)

[2.4.2 Sơ lược các chức năng trong luận văn 45](#_Toc10904805)

[2.4.2.1. PPM 45](#_Toc10904806)

[2.4.2.2. Analog/ADC 46](#_Toc10904807)

[2.5 Giới thiệu về Website (Web) 47](#_Toc10904808)

[2.5.1 Cơ sở dữ liệu 47](#_Toc10904809)

[2.5.2 Web Server 47](#_Toc10904810)

[2.5.3 Lập trình Website 48](#_Toc10904811)

[2.5.4 Tìm hiểu về HTML 48](#_Toc10904812)

[2.5.5 Tìm hiểu về CSS (LESS, SASS, …) 49](#_Toc10904813)

[2.5.6 Tìm hiểu framework Angular 2+ 49](#_Toc10904814)

[2.5.6.1 Angular 2+ là gì ? 49](#_Toc10904815)

[2.5.6.2 Angular 2 + có những tính năng gì ? 49](#_Toc10904816)

[2.5.7 Tìm hiểu về NodeJS 49](#_Toc10904817)

[2.5.8 Tìm hiểu về ExpressJS 50](#_Toc10904818)

[2.5.8.1 ExpressJS là gì ? 50](#_Toc10904819)

[2.5.8.2 Cấu trúc của ExpressJS 51](#_Toc10904820)

[2.5.9 Giao thức HTTP và HTTPs: 51](#_Toc10904821)

[2.5.9.1 Giao thức HTTP: 51](#_Toc10904822)

[2.5.9.2 Giao thức HTTPs: 52](#_Toc10904823)

[2.5.10 Tìm hiểu về Socket Server và Socket Client 52](#_Toc10904824)

[2.5.10.1 Socket là gì ? 52](#_Toc10904825)

[2.5.10.2 Có bao nhiêu loại socket ? 52](#_Toc10904826)

[2.5.10.3 Lý do chọn mô hình Socket 52](#_Toc10904827)

[2.6 Giới thiệu về hệ thống điều hướng năng lượng mặt trời 54](#_Toc10904828)

[2.6.1 Cơ cấu xoay 54](#_Toc10904829)

[2.6.1.1 Hệ thống điều hướng một trục 54](#_Toc10904830)

[2.6.1.2 Hệ thống điều hướng kép (2 trục) 55](#_Toc10904831)

[2.6.2 Động cơ lái 55](#_Toc10904832)

[2.6.3 Bộ điều khiển theo dõi năng lượng mặt trời 56](#_Toc10904833)

[2.6.3.1 Bộ điều khiển phản hồi 57](#_Toc10904834)

[2.6.3.2 Bộ điều khiển vòng hở 57](#_Toc10904835)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 58](#_Toc10904836)

[3.1 Sơ đồ khối tổng quát hệ thống 58](#_Toc10904837)

[3.2 Sơ đồ mạch chi tiết 59](#_Toc10904838)

[3.2.2 Khối đo áp 59](#_Toc10904839)

[3.2.3 Khối Node MCU ESP8266 60](#_Toc10904840)

[3.2.4 Khối Power 60](#_Toc10904841)

[3.2.5 Khối đo nhiệt độ 61](#_Toc10904842)

[3.2.6 Khối GPS 61](#_Toc10904843)

[3.2.7 Khối điều hướng kép 61](#_Toc10904844)

[3.2.7.1 Lựa chọn khung vật liệu 61](#_Toc10904845)

[3.2.7.2 Thiết kế kích thước khung 62](#_Toc10904846)

[3.2.7.3 Cơ cấu trục xoay 63](#_Toc10904847)

[3.2.7.4 Động cơ 65](#_Toc10904848)

[3.3 Khối Arduino Uno 72](#_Toc10904849)

[3.4 Thực hiện phần cứng trên phần mềm Altinum 72](#_Toc10904850)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 74](#_Toc10904851)

[4.1 Giải thuật tổng quát 74](#_Toc10904852)

[4.2 Giải thuật tính toán áp DC 74](#_Toc10904853)

[4.3 Giải thuật đo nhiệt độ 74](#_Toc10904854)

[4.4 Giải thuật GPS 75](#_Toc10904855)

[4.5 Giải thuật điều hướng pin mặt trời 75](#_Toc10904856)

[4.6 Xây dựng Web Server và Website 76](#_Toc10904857)

[4.6.1 Tổng quát về ứng dụng Web 76](#_Toc10904858)

[4.6.2 Mô hình Socket Server và Socket Client trong Luận văn 77](#_Toc10904859)

[4.6.3 Xử lý dữ liệu trên Arduino ESP8266 78](#_Toc10904860)

[4.6.4 Xử lý dữ liệu phía Server 81](#_Toc10904861)

[4.6.4.1 Khởi tạo Server: 81](#_Toc10904862)

[4.6.4.2 Khởi tạo Socket Server: 82](#_Toc10904863)

[4.6.4.3 Tạo chức năng gửi mail tự động: 84](#_Toc10904864)

[5. KẾT QUẢ THỰC HIỆN 86](#_Toc10904865)

[5.1 Kết quả thực hiện phần cứng 86](#_Toc10904866)

[5.2 Kết quả thực hiện phần mềm 87](#_Toc10904867)

[5.3 Đánh giá độ chính xác của hệ thống 92](#_Toc10904868)

[6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 93](#_Toc10904869)

[6.1 Kết luận 93](#_Toc10904870)

[6.2 Hướng phát triển 93](#_Toc10904871)

[7. TÀI LIỆU THAM KHẢO 94](#_Toc10904872)

[8. PHỤ LỤC 95](#_Toc10904873)

**DANH SÁCH HÌNH MINH HỌA**

[Hình 2.1 Mô hình hệ thống năng lượng mặt trời độc lập 17](#_Toc10904880)

[Hình 2.2 Mô hình hệ thống đo đạc và giám sát năng lượng mặt trời 18](#_Toc10904881)

[Hình 2.3 Mô hình hệ thống điều hướng năng lượng mặt trời 18](#_Toc10904882)

[Hình 2.4 Mô hình hệ thống năng lượng mặt trời Off-Gird 20](#_Toc10904883)

[Hình 2.5 Mô hình hệ thống năng lượng mặt trời On-Gird 21](#_Toc10904884)

[Hình 2.6 Mô hình hệ thống năng lượng mặt trời Hybird 22](#_Toc10904885)

[Hình 2.9 Chip ESP8266 23](#_Toc10904886)

[Hình 2.10 Sơ đồ khối chip ESP8266 24](#_Toc10904887)

[Hình 2.11 ESP-WROOM-02 25](#_Toc10904888)

[Hình 2.12 ESP-01 26](#_Toc10904889)

[Hình 2.13 ESP-12E 27](#_Toc10904890)

[Hình 2.14 Kit RF thu phát Wifi ESP8266 NodeMCU Lua 27](#_Toc10904891)

[Hình 2.15 Sơ đồ chân NODE MCU ESP8266 29](#_Toc10904892)

[Hình 2.16 UART trên ESP8266 32](#_Toc10904893)

[Hình 2.17 Mô hình mạng Wifi 35](#_Toc10904894)

[Hình 2.18 Bộ thư viện ESP8266 NONOS\_SDK 38](#_Toc10904895)

[Hình 2.19 Mô hình sử dụng IDE của Arduino lập trình cho ESP8266 39](#_Toc10904896)

[Hình 2.20 Mạch Arduino Uno 42](#_Toc10904897)

[Hình 2.21 Thời gian xung ở mức cao quy định góc quay của RC servo 46](#_Toc10904898)

[Hình 2.22 Cách mắc đơn cho từng quang trở 47](#_Toc10904899)

[Hình 2.23 Cấu trúc ExpressJS 51](#_Toc10904900)

[Hình 2.24 Giao thức HTTP 52](#_Toc10904901)

[Hình 2.25 Trụ đặt tấm pin mặt trời 54](#_Toc10904902)

[Hình 2.26 Hệ thống điều hướng một trục 55](#_Toc10904903)

[Hình 2.27 Hệ thống điều hướng kép 55](#_Toc10904904)

[Hình 3.1 Sơ đồ khối hệ thống phần cứng 58](#_Toc10904905)

[Hình 3.2 Schematic khối đo áp DC 59](#_Toc10904906)

[Hình 3.3 Schematic khối Node MCU ESP8266 60](#_Toc10904907)

[Hình 3.4 Schematic khối Node MCU ESP8266 60](#_Toc10904908)

[Hình 3.5 Schematic khối nhiệt độ 61](#_Toc10904909)

[Hình 3.6 Schematic khối GPS 61](#_Toc10904910)

[Hình 3.7 Tuốc năng trong quạt công nghiệp 64](#_Toc10904911)

[Hình 3.8 Tín hiệu điều khiển động cơ servo 68](#_Toc10904912)

[Hình 3.9 Cấu tạo của động cơ servo 69](#_Toc10904913)

[Hình 3.10 Thiết kế mạch cảm biến ánh sáng 69](#_Toc10904914)

[Hình 3.11 Schematic khối điều khiển Servo 70](#_Toc10904915)

[Hình 3.12 gia cố bằng ke 71](#_Toc10904916)

[Hình 3.13 Gia công khung sắt 72](#_Toc10904917)

[Hình 3.14 Ổ bi 72](#_Toc10904918)

[Hình 3.15 Khối Arduino Uno 72](#_Toc10904919)

[Hình 3.16 layout PCB 73](#_Toc10904920)

[Hình 3.17 Hình 3D của board 73](#_Toc10904921)

[Hình 4.1 Sơ đồ khối giải thuật tổng quát 74](#_Toc10904922)

[Hình 4.2 Sơ đồ khối giải thuật tính toán áp DC 74](#_Toc10904923)

[Hình 4.3 Sơ đồ khối giải thuật đo nhiệt độ 75](#_Toc10904924)

[Hình 4.4 Sơ đồ khối giải thuật GPS 75](#_Toc10904925)

[Hình 4.5 Sơ đồ khối giải thuật điều hướng pin mặt trời 76](#_Toc10904926)

[Hình 4.6 Sơ đồ khối user flow 77](#_Toc10904927)

[Hình 4.7 Mô hình Socket Server 78](#_Toc10904928)

[Hình 4.8 Lưu đồ giải thuật xử lý dữ liệu trên Node MCU ESP8266 79](#_Toc10904929)

[Hình 4.9 Sơ đồ tổng quát giải thuật Socket 83](#_Toc10904930)

[Hình 5.1 board thực tế 86](#_Toc10904931)

[Hình 5.2 Mạch cảm biến ánh sáng 87](#_Toc10904932)

[Hình 5.3 Kết cấu trục quay tuốc năng trong quạt công nghiệp 87](#_Toc10904933)

[Hình 5.4 Mô hình thực tế 87](#_Toc10904934)

[Hình 5.5 Đồ thị điện áp 88](#_Toc10904935)

[Hình 5.6 Số liệu điện áp 88](#_Toc10904936)

[Hình 5.7 Đồ thị nhiệt độ 89](#_Toc10904937)

[Hình 5.8 Số liệu nhiệt độ 89](#_Toc10904938)

[Hình 5.9 Sơ đồ (map) thể hiện các trạm 89](#_Toc10904939)

# **GIỚI THIỆU**

Chương này sẽ giới thiệu lý do em làm đề tài, tình hình trong và ngoài nước hiện nay của hệ thống pin NLMT.

## Tổng quan đề tài

Hiện tại bạn Sỹ đang điều hành một tổ chức NPO tên là Frogsleap Foundation, với sứ  
mệnh lan tỏa những công nghệ xanh đến vùng sâu vùng xa, vùng gặp nhiều khó khăn, một  
trong những hoạt động nền móng và quan trọng nhất là lắp những tấm pin năng lượng mặt  
trời cho người dân. Tính đến thời điểm hiện tại đã là 3 năm, với một hành trình dài như thế,  
pin năng lượng mặt trời đối với bạn không chỉ là một sản phẩm giúp đỡ dành cho những người  
dân chưa có điện lưới quốc gia, mà nó còn mang một giá trị tương lai sâu sắc mà loài người  
sẽ hướng đến. Theo những gì bạn kể và em tìm hiểu, chúng em có niềm tin mãnh liệt rằng  
không sớm thì muộn, các loại năng lượng tái tạo sẽ chiếm tỷ trọng lớn trong nền công nghiệp  
năng lượng trên thế giới, vì vậy để đứng trên cơn sóng đó, để trở thành một bánh răng trong  
guồng quay đó, chúng em mong muốn hơn nữa việc phát triển kỹ năng, tìm hiểu thông tin và  
làm những đề tài liên quan tới năng lượng mặt trời.

Với việc kinh tế phát triển nhanh, Việt Nam đang đứng trước sức ép về nhu cầu năng lượng. Tuy đang sử dụng các nguồn năng lượng truyền thống như than, thủy điện và khí để sản xuất điện, đảm bảo an ninh năng lượng quốc gia, nhưng về lâu dài, Việt Nam đang phải hướng tới xây dựng cơ cấu năng lượng cân bằng thông qua phát triển năng lượng tái tạo. Năng lượng mặt trời là dạng năng lượng tái tạo, và được xem như là dạng năng lượng ưu việt trong tương lai, nó là nguồn năng lượng sẵn có, sạch và miễn phí. Điện năng lượng mặt trời gần đây được xem như là “năng lượng toàn dân”, phản ánh sự đơn giản của việc tích hợp điện mặt trời vào hệ thống cung cấp điện nhà, song song với điện lưới hoặc điện từ nguồn cung cấp khác. Do đó, việc phải giám sát hệ thống năng lượng mặt trời và tối ưu lượng ánh sáng thu được là một điều hết sức cần thiết. Xuất phát từ nhu cầu thực tiễn đó, cũng như xu hướng phát triển công nghệ hiện nay, chúng em quyết định thực hiện đề tài: “**Xây dựng hệ thống đo đạc, giám sát năng lượng và điều hướng trong hệ thống năng lượng mặt trời**”.

## Tình hình nguyên cứu trong và ngoài nước

* + 1. Tình hình nguyên cứu ngoài nước:

Trong các năm gần đây, các công nghệ NLTT, trong đó có các công nghệ NLMT có tốc độ tăng trưởng cao và liên tục. Lý do của xu hướng trên là: (1) Công nghệ ngày càng hoàn thiện, dẫn đến giá NLTT càng ngày càng giảm sâu; (2) Vấn đề an ninh năng lượng. NLTT là nguồn năng lượng (NL) địa phương nên không phụ thuộc vào nguồn nhập khẩu, và do đó không phụ thuộc vào các biến đổi chính trị và các tác động khác; (3) Các nguồn NL hóa thạch đã dần cạn kiệt, trong lúc nhu cầu NL không ngừng tăng; (4) Ô nhiễm môi trường do khai thác sử dụng NL hóa thạch đã đến mức báo động, dẫn đến các hiện tượng biến đổi khí hậu trên toàn cầu. Việc cắt giảm phát thải, sử dụng các nguồn NL sạch - các nguồn NLTT, vì vậy trở nên cấp bách và càng ngày càng có tính nghĩa vụ đối với các quốc gia.

Đến 2013, NLTT đã chiếm tỷ lệ 22,1% trong tổng sản xuất điện năng trên toàn cầu. Nếu kể thêm cả sản xuất nhiệt thì tỷ lệ NLTT trong tổng sản xuất NL trên toàn cầu còn có tỷ lệ cao hơn nhiều. Đặc biệt, trong các năm gần đây, giai đoạn 2008-2013, tốc độ tăng trưởng NLTT nói chung và NLMT nói riêng đạt giá trị khá cao. Trừ 2 nguồn thủy điện và địa nhiệt có tốc độ dưới 4%/năm thì các nguồn NLTT khác có tốc độ tăng trưởng trên 10%/năm. Ấn tượng nhất là tốc độ tăng trưởng của các công nghệ NLMT: điện PMT tăng 55%; nhiệt điện mặt trời (CSP) - 48% và nhiệt mặt trời (chủ yếu để đun nước nóng) - 14%/năm.

Xu thế chung ngày càng rõ nét của tất cả các nước trên thế giới hiện nay là tăng tỷ phần NLTT và giảm NL hóa thạch. Ví dụ, năm 2013, ở Đan Mạch và Tây Ban Nha, điện NL gió đáp ứng lần lượt là 33,2% và 21% tổng nhu cầu điện; nhiều cộng đồng và vùng lãnh thổ đặt mục tiêu sử dụng 100% điện NLTT vào năm 2020 như Dijibouti, Scotland và các quốc gia đảo vùng Tuvalu; nước Đức đặt ra mục tiêu đến năm 2020, khoảng 20 triệu dân (trên tổng số 65 triệu) sống ở các vùng sử dụng 100% NLTT (REN21-2014).

Bên cạnh đó thế giới cũng đang chuyển mình mạnh mẽ trước xu thế Internet of Things (IoT). Internet of Things sẽ là mạng khổng lồ kết nối tất cả mọi thứ, bao gồm cả con người và sẽ tồn tại các mối quan hệ giữa người và người, người và thiết bị, thiết bị và thiết bị. Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng.

Theo báo cáo Ericsson Mobility Report, tới năm 2021, dự kiến sẽ có 28 tỉ thiết bị kết nối trong đó có 15 tỉ thiết bị kết nối IoT bao gồm thiết bị M2M như đồng hồ đo thông minh, cảm biến trên đường, địa điểm bán lẻ, các thiết bị điện tử tiêu dùng như ti vi, đầu DVR, thiết bị đeo. 13 tỉ còn lại là điện thoại di động, laptop, PC, máy tính bảng. Tới năm 2022, theo dự đoán của Gartner thì giá trị gia tăng do IoT mang lại sẽ là 1.900 tỉ đô la Mỹ. Và theo McKinsey, tới năm 2025 IoT sẽ đóng góp vào nền kinh tế toàn cầu là 11.000 tỉ đô la Mỹ. Vì thế, Internet of Thing đang là chìa khóa của thành công trong tương lai.

Những xu thế trên như trở thành một trào lưu trong làng công nghệ, khiến những dự án mà có thể tích hợp cả năng lượng mặt trời và IoT càng trở nên hấp dẫn hơn bao giờ hết.

* + 1. Tình hình nguyên cứu trong nước:

Việt nam được xem là một quốc gia có tiềm năng rất lớn về năng lượng mặt trời, đặc biệt ở các vùng miền trung và miền nam của đất nước, với cường độ bức xạ mặt trời trung bình khoảng 5 kWh/m2. Trong khi đó cường độ bức xạ mặt trời lại thấp hơn ở các vùng phía Bắc, ước tính khoảng 4 kWh/m2do điều kiện thời tiết với trời nhiều mây và mưa phùn vào mùa đông và mùa xuân (Tô Quốc Trụ, 2010). Ở Việt nam, bức xạ mặt trời trung bình 150 kcal/m2 chiếm khoảng 2.000 – 5.000 giờ trên năm, với ước tính tiềm năng lý thuyết khoảng 43,9 tỷ TOE.

Năng lượng mặt trời ở Việt nam có sẵn quanh năm, khá ổn định và phân bố rộng rãi trên các vùng miền khác nhau của đất nước. Đặc biệt, số ngày nắng trung bình trên các tỉnh của miền trung và miền nam là khoảng 300 ngày/năm. Năng lượng mặt trời có thể được khai thác cho hai nhu cầu sử dụng: sản xuất điện và cung cấp nhiệt.

Theo Quyết định số 11/2017/QĐ-TTg của Thủ tướng Chính phủ được ban hành ngày 11/4/2017 về cơ chế khuyến khích phát triển các dự án điện mặt trời tại Việt Nam, dự án điện mặt trời được miễn thuế nhập khẩu đối với hàng hóa nhập khẩu để tạo tài sản cố định cho dự án; thực hiện theo quy định của pháp luật hiện hành về thuế xuất khẩu, thuế nhập khẩu đối với hàng hóa nhập khẩu phục vụ sản xuất của dự án là nguyên liệu, vật tư, bán thành phẩm trong nước chưa sản xuất được.

Các dự án điện mặt trời, công trình đường dây và trạm biến áp để đấu nối với lưới điện được miễn, giảm tiền sử dụng đất, tiền thuê đất, tiền thuê mặt nước theo quy định của pháp luật hiện hành áp dụng cho dự án thuộc lĩnh vực ưu đãi đầu tư.

Bên mua điện có trách nhiệm mua toàn bộ sản lượng điện từ các dự án nối lưới với giá mua điện tại điểm giao nhận điện là 2.086 đồng/kWh (chưa bao gồm thuế giá trị gia tăng, đương đương 9,35 Uscents/kWh. Giá bán điện được điều chỉnh theo biến động của tỷ giá đồng/USD. Giá điện này chỉ áp dụng cho các dự án nối lưới có hiệu suất của tế bào quang điện (Solar cell) lớn hơn 16% hoặc module lớn hơn 15%.

Các dự án trên mái nhà được thực hiện cơ chế bù trừ điện năng sử dụng hệ thống công tơ hai chiều. Trong một chu kỳ thanh toán, lượng điện phát ra từ các dự án trên mái nhà lớn hơn lượng điện tiêu thụ sẽ được chuyển sang chu kỳ thanh toán kế tiếp. Khi kết thúc năm hoặc khi kết thúc hợp đồng mua điện, lượng điện phát dư sẽ được bán cho bên mua điện với giá theo quy định.

Chi phí mua điện từ các dự án điện mặt trời được tính toán và đưa đầy đủ trong thông số đầu vào của phương án giá bán điện hàng năm của Tập đoàn Điện lực Việt nam.

## Nhiệm vụ luận văn

Với tất cả những gì đã phân tích ở trên, luận văn đặt ra yêu cầu là thiết kế một hệ thống giám sát, theo dõi các thông số từ hệ thống pin năng lượng mặt trời độc lập bao gồm U, P, nhiệt độ. Nắm được vị trí của từng hệ thống thông qua GPS. Điều hướng tự động cho các tấm pin sao cho NLMT thu thập được là tối ưu. Bên cạnh đó, với sự phát triển nhanh của nền công nghiệp 4.0 với việc quản lý mọi thứ qua ứng dụng website, … nên chúng em đã tích hợp thêm hệ thống server để liên kết với website giúp cho người quản lý dễ dàng hơn trong việc xem lại các báo cáo số liệu thu thập được từ hệ thống, hoặc xử lý sự cố kịp thời nếu có những thay đổi bất thường do thời tiết hoặc các tác động bên ngoài. Luận văn chủ yếu tập trung nghiên cứu các phần sau:

Tìm hiểu hệ thống mặt trời

* Đo đạc các thông số của hệ thống: U, nhiệt độ
  + Nghiên cứu kit MCU ESP8266 và Arduino Uno
  + Nghiên cứu mạch đo dòng điện, điện áp.
  + Nghiên cứu đo nhiệt độ từ cảm biến nhiệt độ DS18B20
  + Nghiên cứu GPS thông qua module Neo 8
  + Nghiên cứu động cơ Servo
  + Nghiên cứu sử dụng quang trở trong điều hướng
  + Nghiên cứu hệ thống cơ khí trục quay điều hướng kép.
* Truyền và hiển thị thông số trên web
  + Tìm hiểu cách tạo một server để lưu trữ dữ liệu và thực hiện các request từ user.
  + Nghiên cứu sử dụng kit MCU ESP8266 để truyền nhận data đến server thông qua wifi.
  + Nghiên cứu cách truyền data realtime giữa ESP8266 và server và cách truyền data realtime giữa server và client để hệ thống được cập nhật sớm nhất có thể các thông số.
  + Phía web (client) sẽ lấy dữ liệu từ server và hiển thị lên phía web. Bao gồm cách tác vụ như thống kê số liệu, vẽ đồ thị số liệu, thể hiện vị trí các trạm pin mặt trời trên map (Google Maps).
* Xây dựng mô hinh thực nghiệm
* Thiết kế, thi công sơ đồ mạch và giải thuật đọc các thông số và giao tiếp giữa các cảm biến, kit MCU ESP8266, Arduino uno.
* Lập trình server để giao tiếp với Arduino và lưu trữ dữ liệu dưới Database (Dạng NoSQL).
* Lập trình client để giao tiếp và thể hiện những số liệu thu thập được. Thống kê những số liệu đó để tiện cho việc xử lý nếu cần, cũng như sử dụng cho việc nghiên cứu, dự đoán các thông số khác.

# LÝ THUYẾT

Chương này sẽ giới thiệu thông tin về các loại hình lắp đặt năng lượng mặt trời, các công nghệ kèm theo nó như mạch cầu phân áp, cảm biến nhiệt độ, GPS, điều hướng kép,...

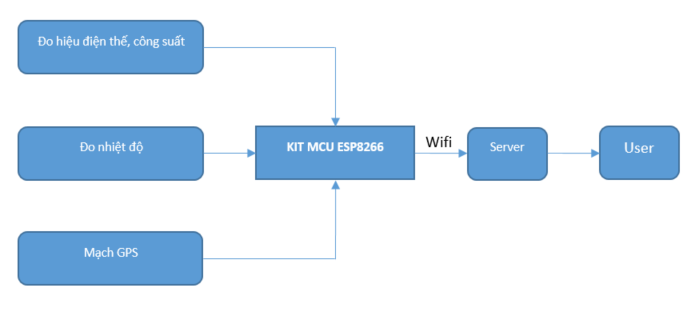
## Tổng quan thiết kế



##### Hình 2.1 Mô hình hệ thống năng lượng mặt trời độc lập

Luận văn xây dựng hệ thống đo đạc và điều hướng đối với mô hình hệ thống Hybird Solar PV System. Năng lượng từ PV array được đưa tới khối charge controller và được lưu trữ ở Baterry, khối charge controller sẽ cung cấp điện cho DC Load.

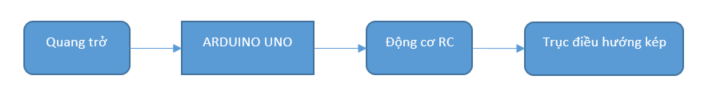
Về hệ thống giám sát và đo đạc, ta xây dựng như sau:



##### Hình 2.2 Mô hình hệ thống đo đạc và giám sát năng lượng mặt trời

Hệ thống đo đạc có khối chính là MCU ESP8266. Các dữ liệu về hiệu điện thế sẽ qua mạch giảm áp để vào MCU, ở đây MCU sẽ đo và tính toán ra các thông số U, I, P cần thiết của hệ thống, nhiệt độ và GSP được kết nối trực tiếp. Sau đó MCU ESP8266 có nhiêm vụ đóng gói và chuyển dữ liệu đó tới server bằng giao thức HTTP thông qua kết nối WIFI. User (người dùng) sẽ lấy thông tin từ server thông qua giao diện Web.

Về hệ thống điều hướng, ta xây dựng như sau:



##### Hình 2.3 Mô hình hệ thống điều hướng năng lượng mặt trời

Hệ thống điều hướng kép năng lượng mặt trời có khối chính là mạch Arduino Uno. Thông tin từ quang trở được thu thập và tính toán, chuyển đổi thành góc quay cho động cơ RC. Có 2 động cơ RC điều hướng phương dọc và phương ngang cho khung cơ khí, từ đó xoay pin mặt trời cho phù hợp với hướng của ánh sáng.

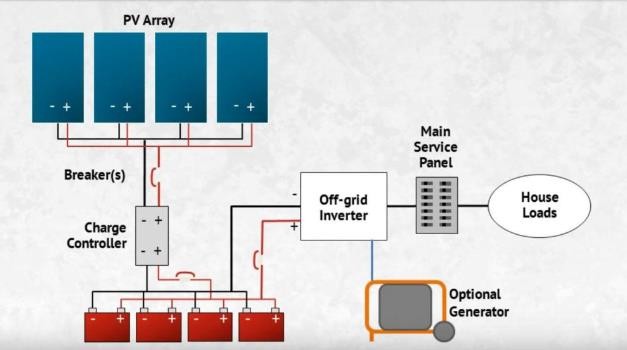
## Tổng quan về năng lượng mặt trời

* + 1. Ưu, nhược điểm của năng lượng mặt trời

Năng lượng mặt trời là vô tận, dư thừa để đáp ứng nhu cầu về năng lượng của nhân loại, theo tính toán của NASA, mặt trời còn có thể cung cấp năng lượng cho chúng ta khoảng 6,5 tỉ năm nữa. Tiềm năng của năng lượng mặt trời là rất lớn, mỗi ngày bề mặt trái đất được hưởng 120.000 terawatts (TW) của ánh sang mặt trời, cao gấp 20.000 lần so với nhu cầu của con người trên toàn thế giới (1 TW = 1.000 tỉ W). Không chỉ thế, việc sản xuất năng lượng mặt trời không sử dụng các loại động cơ như trong nhà máy phát điện nên việc tạo ra điện không gây tiếng ồn. Do vậy, năng lượng mặt trời ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực ở Việt Nam và các nước trên thế giới. Điện năng lượng mặt trời gần đây được xem như là “năng lượng toàn dân”, phản ánh sự đơn giản của việc tích hợp điện mặt trời vào hệ thống cung cấp điện nhà, song song với điện lưới hoặc điện từ nguồn cung cấp khác.

Tuy nhiên, có một thực tế bất khả kháng: vào ban đêm hay trong những ngày nhiều mây và mưa thì cường độ ánh sáng mặt trời rất yếu, vì thế, năng lượng mặt trời thu được không ổn định, thu được dồi dào vào những ngày nắng, và rất ít vào những ngày tối. Ngoài ra, chi phí lưu trữ điện mặt trời để sử dụng cho ban đêm hay trời không có nắng còn khá cao so với túi tiền của đại đa số người dân.

* + 1. Một số hệ thống năng lượng mặt trời
       1. Off-Gird Solar PV System



##### Hình 2.4 Mô hình hệ thống năng lượng mặt trời Off-Gird

Đây là hệ thống năng lượng mặt trời độc lập, không hòa lưới. khối PV Array nhận ánh sáng từ mặt trời, chuyển hóa thành năng lượng điện, đẩy vào battery thông qua Charge Controller, năng lương từ pin (DC) qua bộ inverter chuyển thành điện AC, cung cấp cho House Load.

Nhược điểm ủa hệ thống này là chi phí pin lưu trữ năng lượng mặt trời rất lớn, nên hệ thống này chủ yếu được sử dụng ở những nơi có lưới điện không ổn định, như vùng sâu vùng xa, hải đảo, ...

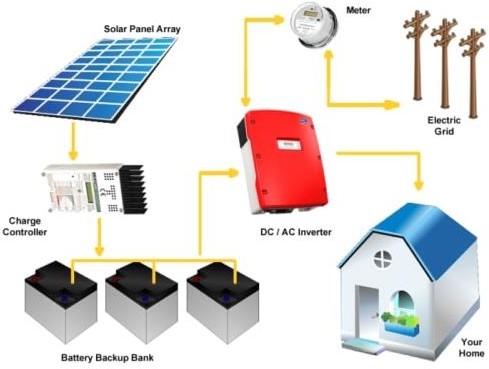
* + - 1. On-Gird Solar PV System



##### Hình 2.5 Mô hình hệ thống năng lượng mặt trời On-Gird

Hệ thống này thường được sử dụng ở nơi có lưới điện ổn định. Your Home sẽ sử dụng 2 nguồn cung cấp, một nguồn từ lưới điện, 1 nguồn từ Solar Panel Arry. Vào những ngày nắng năng lượng mặt trời dồi dào, thì ta có thể thể bán điện cho điện lực, còn những ngày tối, năng lượng mặt trời không nhiều, thì Your Home sẽ sử dụng điện từ lưới điện và 1 phần nhỏ của năng lượng mặt trời. Ưu điểm của nó là đơn giản, dễ lắp đặt, ít tốn chi phí.

* + - 1. Hybird Solar PV System



##### Hình 2.6 Mô hình hệ thống năng lượng mặt trời Hybird

Đây là hệ thống kết hợp cả 2 hệ Off-Gird Solar PV System và On-Gird Solar PV System.

## Giới thiệu về SoC ESP8266

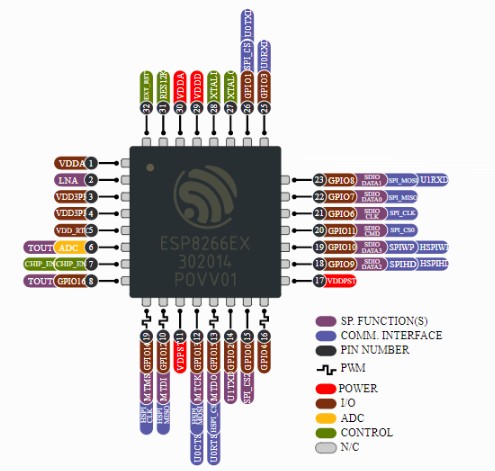
* + 1. Giới thiệu

ESP8266 là dòng chip tích hợp Wifi 2.4Ghz, có thể lập trình được, giá thành rẻ được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems

Được phát hành đầu tiên vào tháng 8 năm 2014, đóng gói đưa ra thị trường dạng Module ESP-01, được sản xuất bởi bên thứ 3: AI-Thinker. Có khả năng kết nối internet qua mạng Wifi một cách nhanh chóng và sử dụng rất ít linh kiện đi kèm. Với giá cả có thể nói là rất rẻ so với tính năng và khả năng ESP8266 có thể làm được.

ESP 8266 có một cộng đồng các nhà phát triển trên thế giới rất lớn, cung cấp nhiều module lập trình mã nguồn mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng rất nhanh.

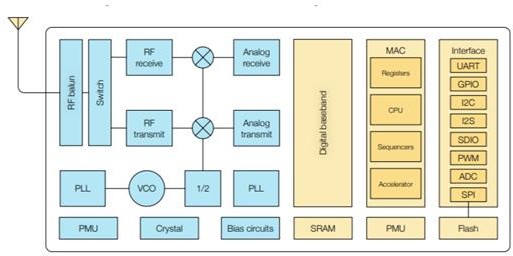
Hiện nay các dòng chip ESP8266 trên thị trường đều mang nhãn ESP8266EX, là phiên bản nâng cấp của ESP8266 với kích thước chỉ 5mmx5mm.



##### Hình 2.9 Chip ESP8266

Thông số chung của chip ESP8266.

* 32 bit RISC CPU: Tensilica Xtensa LX106 hoạt động ở xung nhịp 80 MHz
* Hỗ trợ Flash ngoài từ 512KB đến 4MB
* 64 Kbytes RAM thực thi lệnh
* 96 Kbytes RAM dữ liệu
* 64 Kbytes boot ROM
* Chuẩn wifi EEE 802.11 b/g/n, Wi-Fi 2.4 GHz
  + Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network
  + Hổ trợ WEP, WPA/WPA2, Open network
* Tích hợp giao thức TCP/IP
* Hổ trợ nhiều loại anten
* 16 chân GPIO
* Hổ trợ SDIO 2.0, UART, SPI, I2C, PWM,I²S với DMA
* 1 ADC 10-bit
* Dải nhiệt độ hoạt động rộng: -40C ~ 125C



##### Hình 2.10 Sơ đồ khối chip ESP8266

Ưu điểm của chip ESP8266

*32-bit Tensilica MCU*: ESP8266EX tích hợp vi lý Tensilica L106 32-bit (MCU) là dòng chip low-power, 16-bit RSIC, tốc độ clock cao nhất là 160 MHz. Nếu hệ thống hoạt động với Real Time Operation System (RTOS) và Wi-Fi stack thì ta có khoảng 80% khả năng xử lý cho ứng dụng người dùng.

*Low Power Management*: Với những ứng dụng cho mobile, thiết bị điện tử cầm tay và Internet of Things (IoT), ESP8266EX hoạt động với mức tiệu thụ năng lượng rất thấp với nhưng công nghệ độc quyền. Tính năng tiết kiệm năng lượng với 3 chế độ hoạt động – active mode, sleep mode và deep sleep mode, vì vậy cho phép hiện thực những thiết bị với thời lượng Pin rất lớn.

*Là thiết kế bền vững:* Hoạt động với dãi nhiệt khá rộng -40°C to +125°C (trong công nghiệp), ESP8266EX có thể hoạt động tốt trong môi trường công nghiệp. Với sự tích hợp cao, dòng chip này hoạt động với rất ít linh kiện ngoài làm tăng độ tin cậy, chặt chẽ và ổn định cao.

* + 1. Các loại Module ESP8266

ESP8266 cần ít nhất thêm 7 linh kiện nữa mới có thể hoạt động, trong đó phần khó nhất là Antena. Đòi hỏi phải được sản xuất, kiểm tra với các thiết bị hiện đại. Do đó, trên thị trường xuất hiện nhiều Module phát triển đảm đương hết để người dùng đơn giản nhất trong việc phát triển ứng dụng. Một số Module phổ biến trên thị trường hiện nay:

* + - 1. ESP-WROOM-02:



##### Hình 2.11 ESP-WROOM-02

ESP-WROOM-02 là một module MCU Wifi 32-bit tiết kiệm năng lượng dựa trên chip ESP8266.

Hổ trợ các chuẩn mạng không dây 802.11 b/g/n

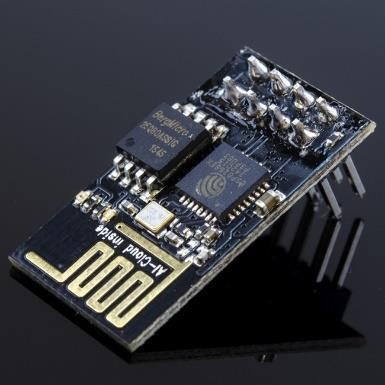
Tích hợp sẳn giao thức TCP/IP, 10-bit ADC, TR switch, balun, LNA, và các chuẩn giao tiếp HSPI/UART/PWM/I2C/I2S.

Hổ trợ Wi-Fi Alliance, SRRC, FCC, CE, TELEC, IC & KCC Certified, RoHS, Halogen Free, REACH & CFSI Compliant, HTOL, ESD-HM, MSL, μHAST, HTSL Hổ trợ Cloud Server Development

Custom firmware development qua SDK

Cấu hình người dùng qua AT Instruction Set, Cloud Server và ứng dụng Android/iOS Khoảng cách giữa các chân 2.54mm

* + - 1. ESP-01:



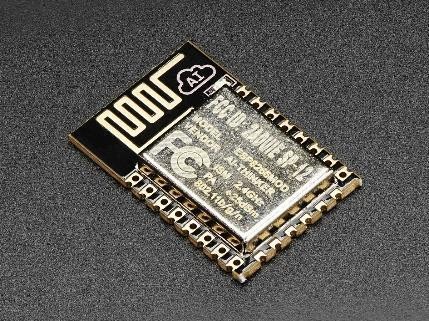
##### Hình 2.12 ESP-01

Mạch nhỏ, gọn (24.75mm x 14.5mm) Điện áp làm việc 3.3v

Tích hợp sẳn anten PCB trace trên module Có hai led báo hiệu : led nguồn, led TXD Có các chế độ: AP, STA, AT + STA

Lệnh AT rất đơn giản, dễ dàng sử dụng Khoảng cách giữa các chân 2.54 mm

* + - 1. ESP-12E:



##### Hình 2.13 ESP-12E

Sử dụng nguồn 3.3v

Tích hợp anten PCB trace trên module 4MB flash

Tiêu chuẩn wifi: 802.11b/g/n, với tần số 2.4GHz và hổ trợ bảo mật WPA/WPA2 Khoảng cách giữa các chân 2mm

Hiện nay ESP-12E đã được tích hợp luôn thành một bộ kit hoàn chỉnh, được sử dụng rộng rãi, được gọi là kit RF thu phát Wifi ESP8266 NodeMCU Lua.



##### Hình 2.14 Kit RF thu phát Wifi ESP8266 NodeMCU Lua

Mô tả: Kit RF thu phát Wifi ESP8266 NodeMCU Lua là kit phát triển dựa trên nền module ESP-12E, với thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua trong đề tài sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window và Linux, đây là phiên bản nâng cấp từ các phiên bản sử dụng IC nạp giá rẻ CH340.

Thông số kỹ thuật:

- IC chính: ESP8266 Wifi SoC.

- Phiên bản firmware: NodeMCU Lua

- Chip nạp và giao tiếp UART: CP2102.

- GPIO tương thích hoàn toàn với firmware Node MCU.

- Cấp nguồn: 5VDC MicroUSB hoặc Vin.

- GIPO giao tiếp mức 3.3VDC

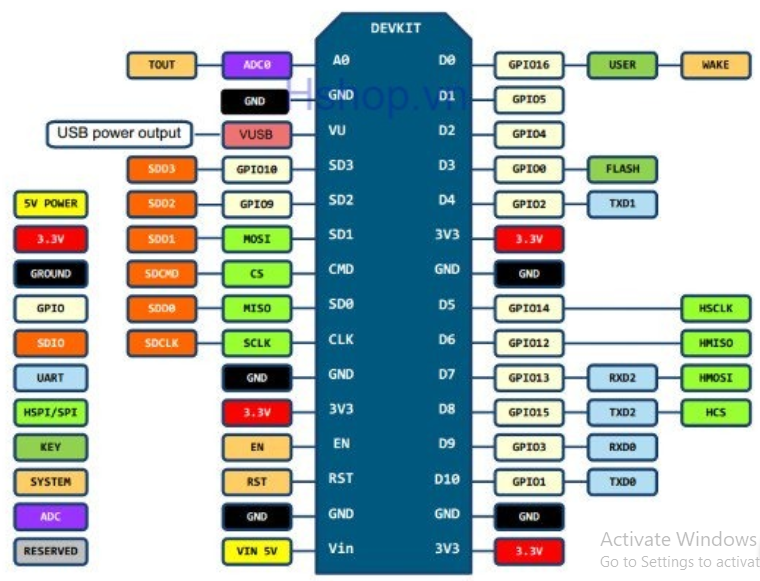
- Tích hợp Led báo trạng thái, nút Reset, Flash.

- Tương thích hoàn toàn với trình biên dịch Arduino.

- Kích thước: 25 x 50 mm

Vì những tính năng và sự thuận tiện trên, đề tài quyết định sử dụng kit RF thu phát Wifi ESP8266 NodeMCU Lua làm một trong 2 mạch chính để hoạt động. Mọi chức năng và lập trình về sau có liên quan đến ESP8266 đều được làm trên bộ kit này.

* + 1. Sơ lược các chức năng trong luận văn
    2. GPIO



##### Hình 2.15 Sơ đồ chân NODE MCU ESP8266

GPIO là viết tắt General Purpose Input Output ( Cổng đầu vào và ra với mục đích cơ bản) thực tế nó là các chân đầu ra, đầu vào đa chức năng. Nhìn vào cấu trúc bất kì vi điều khiển nào chúng ta dễ dàng nhận thấy các hàng chân của vi điều khiển, chúng có chức năng kiểm soát và giao tiếp với các thiết bị bên ngoài.

Tên của GPIO trong Node MCU ESP8266 được sử sụng trực tiếp tương tự như Arduino với các hàm sau: pinMode, digitalRead và digitalWrite:

- PinMode: Mặc định khi esp8266 khi khởi động sẽ set các GPIO ở trạng thái INPUT, vì vậy tùy theo mục đích sử dụng GPIO các bạn nên set chức năng cho chúng trong hàm setup hoặc các hàm muốn sử dụng đến GPIO theo cú pháp: pinMode(pin number, mode);

VD: pinMode(1, INPUT);

Các GPIO 0 – 15 có các mode: INPUT, OUTPUT, INPUT\_PULLUP

GPIO6 có các mode: INPUT, OUTPUT, INPUT\_PULLDOWN

Chú ý: khi sử dụng mode INPUT\_PULLUP, hoặc INPUT\_PULLDOWN để chọn điện trở kéo lên hoặc kéo xuống tránh nhầm lẫn giữa các GPIO.

- DigitalRead: Hàm này có chức năng đọc về trạng thái của GPIO và trả về HIGH / 1 nếu GPIO đó ở mức logic 1 và LOW / 0 nếu nó ở mức 0. Chú ý là nó sử dụng được cho GPIO ở cả trạng thái INPUT và OUTPUT. Sử dụng theo cú pháp: digitalRead(pin number);

VD: digitalRead(1);

- DigitalWrite: Hàm này có chức năng set trạng thái của GPIO. Chú ý là nó sử dụng được cho GPIO ở trạng thái OUTPUT, hãy chắc chắn đã sử dụng pinMode OUTPUT trước khi sử dụng nó. Sử dụng theo cú pháp: digitalWrite(pin number, value); // value = HIGH or LOW.

VD: digitalWrite(1, HIGH);

- Analog input: ESP8266 chỉ có duy nhất 1 GPIO để chuyển đổi ADC, nó có 2 chức năng( chú ý chỉ dùng 1 trong 2):

+ Đọc giá tri điện áp nguồn (VCC) sử dụng qua hàm setup: ADC\_MODE(ADC\_VCC) và đọc giá trị thông qua hàm ESP.getVcc(), chú ý GPIO ADC không được kết nối tới bất kỳ đâu.

+ Đọc giá trị điện áp ngoài đặt lên ADC GPIO ( => ADC chính hiệu), đọc giá trị bằng hàm: analogRead(A0). Chú ý quan trọng: điện áp để đọc ADC nằm trong khoảng 0 – 1V

- Analog output: Thực ra Tearm sáng lập ra Arduino đặt tên chức năng là “Analog output” có vẻ không hay cho lắm, vì nó dễ gây ra hiểu lầm là chức năng DAC tuy nhiên thực tế nó là bộ băm xung PWM như các dòng vi điều khiển khác. Để sử dụng hết sức đơn giản, dùng hàm analogWrite(pin, value) với:

+ Pin: 0 – 16: tương ứng với GPIO0 – 16

+ Value: 0 – PWMRANGE, mặc định PWMRANGE = 1023, có thể thay đổi PWMRANGE thông qua hàm analogWriteRange(new\_range)

+ Tần số xung mặc định là 1kHz, sử dụng hàm analogWriteFreq(new\_frequency) nếu muốn set tần số khác.

- Các chuẩn giao tiếp ngoại vi: Ngoài tính năng logic tại các GPIO, chúng còn được tích hợp các chuẩn giao tiếp phổ biến như Serial, I2C, SPI, và các GPIO sử dụng cho các giao tiếp đó liệt kê trong datasheet.

Ở một số board hoặc module( như: ESP-12ED, NodeMCU 1.0) chip Flash hoạt động ở mode DIO nên được ra chân sẵn 2 GPIO 9 và 11, bạn có thể sử dụng chúng bình thường.

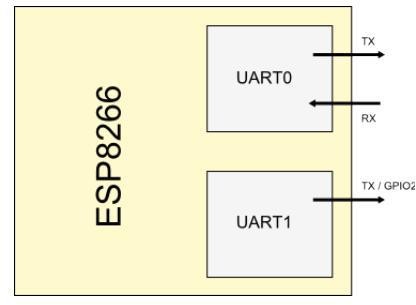
Ngắt ngoài: ESP8266 hỗ trợ ngắt ngoài ở hầu hết các GPIO ngoại trừ GPIO16, hỗ trợ các sườn ngắt: CHANGE, RISING, FALLING. Dễ dàng sử dụng thông qua các hàm attachInterrupt, detachInterrupt của Arduino.

* + 1. UART

UART (Universal Asynchronous Receiver – Transmitter) thực hiện chuyển đổi giữa dữ liệu nối tiếp và song song. Một chiều, UART chuyển đổi dữ liệu song song bus hệ thống ra dữ liệu nối tiếp để truyền đi. Một chiều khác, UART chuyển đổi dữ liệu nhận được dạng dữ liệu nối tiếp thành dạng dữ liệu song song cho CPU có thể đọc vào bus hệ thống.

ESP8266 là một thiết bị Wifi và do đó ta sẽ kết nối với nó bằng cách sử dụng giao thức Wifi, nhưng một số thiết lập ban đầu là điều cần thiết. Vì vậy ESP8266 có một giao tiếp UART bằng cách sử dụng chân TX (truyền dữ liệu đi) và RX (nhận dữ liệu tới). Để đơn giản và thuận tiện thì ta có thể dùng bộ chuyển đổi USB→UART để giao tiếp với ESP8266 với máy tính.

Thông qua UART, chúng ta có thể gửi các tập lệnh và quan sát các dữ liệu nhận được trên màn hình thông qua terminal. Mục đích quan trọng nhất của UART là để ghi các chương trình lên bộ nhớ flash của ESP8266. Có rất nhiều cách để thực hiện việc này thông qua các platform như IDE Arduino, Eclipse, PlatformIO... Khi sử dụng UART, chúng ta cần quan tâm đến tốc độ baud. Đây là tốc độ truyền thông dữ liệu giữa ESP8266 và đối tác của nó. Tốc độ mặc định của ESP8266 là 115200, tại thời điểm này công cụ nạp cho ESP8266 có thể đạt tới 921600 baud.



##### Hình 2.16 UART trên ESP8266

ESP8266 có 2 UART, trong đó UART0 có đầy đủ 2 tính năng TX, RX, và dùng làm ngõ ra mặc định cho các thông tin debug cho SDK, cũng như là cổng nạp chương trình.

UART0 của ESP8266 hỗ trợ 128 Byte FIFO và tính năng Ngắt Timeout, ngắt khi đầy FIFO. Có thể hiểu nếu cấu hình ngắt 128 bytes FIFO, và ngắt Timeout, khi UART nhận đầy FIFO sẽ kích hoạt ngắt đầy FIFO, hoặc nếu trường hợp chưa đầy FIFO nhưng qua 1 khoảng thời gian (có thể cấu hình được) mà không nhận được dữ liệu nữa thì cũng phát sinh ngắt TOUT.

UART thứ hai với chỉ có đầu ra. Mục đích của UART này là đưa ra các chuẩn đoán và gỡ lỗi thông tin, điều này cực kì hữu ích trong quá trình lập trình. UART này có thể được sử dụng bằng cách ghép với GPIO

* + 1. SERIAL

Có thể nói Serial là một trong những phương thức giao tiếp đơn giản nhất. Tuy nhiên ESP8266 chỉ có duy nhất một cổng Serial được phần cứng hỗ trợ sẵn. Vì vậy, nếu muốn giao tiếp với nhiều module Serial thì đó là một chuyện không thể. Và đó chính là lý do vì sao thư viện Software Serial ra đời.

Trong đề tài, vì ESP8266 phải vừa giao tiếp với máy tính, vừa giao tiếp với thiết bị ngoại vi nên việc sử dụng Software Serial là điều bắt buộc.

Ví dụ cho việc sử dụng thư viện đó như sau:

#include <SoftwareSerial.h>

SoftwareSerial ss(4,5);

Trong đoạn code này, ss là tên mới cho serial thứ 2, chân pin 4 tương đương với RX, pin 5 tương đương với TX. Serial cũ vẫn hoạt động trên chân mặc định của nó.

void setup()

{ Serial.begin(9600);

ss.begin(9600); }

Tại phần setup, baudrate của 2 serial phải giống nhau thì mới có thể giao tiếp được.

Đối tượng Serial của ESP8266 làm việc giống nhiều với Arduino bình thường. Ngoại trừ việc phần cứng của ESP8266 có thêm 128 bytes RAM FIFO và 256 bytes RX-TX Buffer. Cả TX và RX đều truyền nhận dựa vào interrupt. Việc đọc và ghi dữ liệu chỉ bị block lại khi FIFO/buffer đầy/rỗng.

Serial sử dụng là UART0, được map thằng vào chân GPIO1 (TX) và GPIO3 (RX). Serial có thể REMAP lại vào GPIO15 (TX) và GPIO13 (RX) bởi việc gọi hàm Serial.swap() sau khi gọi Serial.begin. Gọi swap lại sẽ MAP UART0 trở lại GPIO1 và GPIO3.

Serial1 sử dụng UART1, TX pin là chân GPIO2. UART1 không thể sử dụng để nhận dữ liệu bởi vì bình thường nó được sử dụng để kết nối với Flash. Để sử dụng Serial1, gọi Serial1.begin(baudrate).

Nếu Serial1 không được sử dụng và Serial không bị REMAP - TX cho UART0 có thể MAP sang GPIO2 bởi gọi hàm Serial.set\_tx(2) sau Serial.begin hay trực tiếp với Serial.begin(baud, config, mode, 2).

Mặc định, tất cả thông tin chuẩn đoán hệ thống và thư viện sẽ bị bỏ qua nếu gọi hàm Serial.begin. Để cho phép những thông tin đó, có thể gọi Serial.setDebugOutput(true). Để chuyển thông tin đó ra Serial1, gọi hàm Serial1.setDebugOutput(true).

Ta cũng cần phải gọi Serial.setDebugOutput(true) để cho phép hàm printf().

Cả đối tượng Serial và Serial1 hỗ trợ 5, 6, 7, 8 bits dữ liệu, odd (O), even (E), và no (N) parity, và 1 hay 2 bits stop. Để cấu hình các mode trên, gọi Serial.begin(baudrate, SERIAL\_8N1), Serial.begin(baudrate, SERIAL\_6E2), v.v..

Một phương thức được hiện thực trên cả 2 Serial và Serial1 để lấy baud rate hiện tại như sau: Serial.baudRate(), Serial1.baudRate() trả về một số int của tốc độ Baud hiện tại.

Ví dụ

// Set Baud rate 57600

Serial.begin(57600);

// Kiểm tra baud rate hiện tại

int br = Serial.baudRate();

// sẽ xuất ra "Serial is 57600 bps"

Serial.printf("Serial is %d bps", br);

Lưu ý thư viện Software Serial cho ESP8266 được thực hiện chỉ duy nhất cho ESP8266 boards, và sẽ không làm việc với các thư viện Arduino khác.

* + 1. Analog/ADC trong ESP8266

ESP8266 có duy nhất 1 chân ADC, chúng ta có thể dùng để đọc điện áp bên ngoài, hay đọc điện áp VCC. Để đọc điện áp bên ngoài bằng chân ADC, sử dụng hàm analogRead(A0). Điện áp chấp nhận ở mức 0 — 1.0V trên chân CHIP và 0-3.3V trên board IoT WiFi Uno sau khi được chia áp. Để đọc điện áp cấp (VCC) cho module, sử dụng ESP.getVcc() và chân ADC bên ngoài phải để hở. Đồng thời phải cấu hình dòng lệnh sau trong sketch:

ADC\_MODE(ADC\_VCC);

Dòng lệnh này có thể bất kỳ đâu, ở phía ngoài một hàm - có thể coi như ngang ngửa với dòng #include trong sketch.

Ngõ ra Analog: analogWrite(pin, value) cho phép sử dụng Software PWM trên bất kỳ GPIO nào từ 0..16. Gọi analogWrite(pin, 0) sẽ ngừng cho phép PWM trên chân đó. value có thể giới hạn từ 0 đến PWMRANGE, mặc định là 1023. và có thể thay đổi bởi hàm analogWriteRange(new\_range). Tần số PWM mặc định là 1kHz. Gọi analogWriteFreq(new\_frequency) để thay đổi tần số.

Vd: Đọc giá trị analog ghi ra PWM

int ledPin = 16; // LED connected to digital pin 9

int val = 0; // variable to store the read value

void setup()

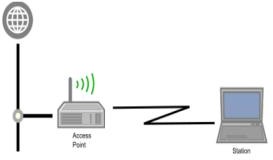
{ pinMode(ledPin, OUTPUT); // sets the pin as output }

void loop()

{ val = analogRead(A0); // read the input pin

analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023, analogWrite values from 0 to 255 }

* + 1. Kết nối Wifi



##### Hình 2.17 Mô hình mạng Wifi

WiFi là từ viết tắt của Wireless Fidelity, sử dụng sóng vô tuyến để truyền tín hiệu. Loại sóng vô tuyến này tương tự như sóng điện thoại, truyền hình và radio. Và trên hầu hết các thiết bị điện tử ngày nay như máy tính, laptop, điện thoại, máy tính bảng... đều có thể kết nối Wifi.

Để có được sóng Wifi thì chúng ta cần phải có bộ phát Wifi - chính là các thiết bị như modem, router. Thiết bị modem, router sẽ lấy tín hiệu Internet qua kết nối hữu tuyến rồi chuyển thành tín hiệu vô tuyến, và gửi đến các thiết bị sử dụng như điện thoại smartphone, máy tính bảng, laptop... Đây là quá trình nhận tín hiệu không dây (hay còn gọi là adapter) chính là card wifi trên laptop, điện thoại... và chuyển hóa thành tín hiệu Internet. Và quá trình này hoàn toàn có thể thực hiện ngược lại, nghĩa là router, modem nhận tín hiệu vô tuyến từ adapter và giải mã chúng, gửi qua Internet.

Sóng Wifi sử dụng chuẩn kết nối 802.11 trong thư viện IEEE (Institute of Electrical and Electronics Engineers), chuẩn này bao gồm 4 chuẩn nhỏ hơn là a/b/g/n. (chúng thường thấy trên modem, router có các ký hiệu này).

* Chuẩn 802.11b là phiên bản yếu nhất, hoạt động ở mức 2.4GHz và có thể xử lý đến 11 megabit/giây.
* Chuẩn 802.11g nhỉnh hơn đôi chút so với chuẩn b, tuy nó cũng hoạt động ở tần số 2.4GHz nhưng nó có thể xử lý 54 megabit/giây.
* Chuẩn 802.11a phát ở tần số cao hơn là 5GHz và tốc độ xử lý đạt 54 megabit/giây.
* Cuối cùng là chuẩn 802.11n, nó hoạt động ở tần số 2.4GHz nhưng tốc độ xử lý lên đến 300 megabit/giây.

ESP8266 có chuẩn kết nối IEEE 802.11b/g/n, có thể đóng vai trò là điểm truy cập (Access Point), một trạm (Static) hay cả hai cùng lúc. Đây chính là điểm mạnh của chip ESP8266 so với các dòng chip wifi khác trên thị trường.

* + 1. Lập trình cho ESP8266

Espressif hiện đã hỗ trợ 3 nền tảng SDK (Software Development Kit - Gói phát triển phần mềm) độc lập là: NONOS SDK, RTOS SDK và Arduino. Cả 3 đều có những ưu điểm riêng phù hợp với từng ứng dụng nhất định, và sử dụng chung nhiều các hàm điều khiển phần cứng. Hiện nay Arduino đang được sử dụng rộng rãi bởi tính dễ sử dụng, kiến trúc phần mềm tốt và tận dụng được nhiều thư viện cộng đồng.

* + - 1. Lập trình trực tiếp

Trong mô hình này, chúng ta sẽ lập trình ra một firmware cho ESP8266 sử dụng bộ thư viện SDK được cung cấp bởi Espressif. Chúng ta sẽ theo mô hình này cho các ứng dụng muốn dùng ES8266 để điều khiển thiết bị trực tiếp mà không cần một external MCU điều khiển; hoặc chúng ta cần các tính năng mà firmware chuẩn chưa cung cấp.

Bộ thư viện SDK của ESP8266 bao gồm các thành phần sau:

Các file code thư viện cung cấp API để điều khiển phần cứng (HAL driver), định nghĩa các địa chỉ của thanh ghi điều khiển, cũng như các module software như RTOS, Memory Managment...

Các chương trình biên dịch (compiler and linker) để biên dịch source code và tạo ra file \*.bin firmware cho ESP8266

Software để truyền file \*.bin xuống ESP8266 qua đường UART

Lưu ý Espressif khuyên chúng ta nên biên dịch firmware cho ESP8266 trên nền Linux OS sử dụng các compiler tool được khuyên dùng để đảm bảo firmware được biên dịch đúng. Với môi trường Windows, Espressif cung cấp một file máy ảo (dùng trong Virtual Box) có cài đặt sẵn các trình biên dịch có thể sử dụng ngay. Ngoài ra cộng đồng mạng cũng hỗ trợ 1 bộ toolkit khác dùng trên Windows biên dịch bằng Python.

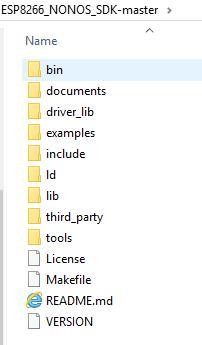
Với các file code thư viện, Espressif hỗ trợ 2 bộ thư viện: Non-OS SDK và RTOS SDK. Chúng ta sẽ dùng bộ thư viện Non-OS SDK để lập trình cho ESP8266 không sử dụng hệ thống RTOS và RTOS SDK để lập trình ứng dụng có sử dụng các giải pháp đồng bộ các tác vụ ứng dụng của hệ thống RTOS.

Bản chất ứng dụng sử dụng Non-OS SDK và RTOS SDK có thể được mô tả như

sau:

Non-OS SDK: Do không có sử dụng hệ điều hành quản lý (OS) nên các hoạt động trong chương trình sẽ được quản lý thông qua timer hay các hàm call- back. Các tác vụ sẽ được gọi khi có 1 điều kiện tương ứng xảy ra và thông thường được xử lý theo phương thức polling để kiểm tra tất cả các điều kiện của các tính năng tương ứng. Đây là mô hình lập trình truyền thống mà chúng ta thường sử dụng khi bắt đầu quá trình học lập trình ứng dụng. Ưu điểm của nó là đơn giản nên ổn định và được nhà sản xuất cập nhật các tính năng mới đầu tiên. Hiện nay, phiên bản mới nhất của Non-OS SDK là V2.0.0.

RTOS SDK: Sử dụng FreeRTOS mã nguồn mở giúp hệ thống hoạt động đa nhiệm. Do đó, hệ thống này cho phép người dùng sử dụng các phương thức chuẩn cho việc quản lý tài nguyên, chu kỳ hoạt động theo chu kỳ, quản lý thời gian trễ trong tác vụ cung cấp thông tin cũng như đồng bộ hóa giữa các tác vụ. Ưu điểm của nó có thể xử lý đa nhiệm trong các yêu cầu dự án lớn tuy nhiên phụ thuộc vào chất lượng SDK mà hệ thống hoạt động ổn định hay không. Do đó thông thường các API mới được cập nhật chậm hơn so với Non-OS. Điểm đặc biệt cần lưu ý là RTOS không hỗ trợ AT command. Do đó nếu bạn muốn viết 1 firmware mà cần hỗ trợ AT command để giao tiếp với MCU bạn phải sử dụng Non-OS SDK. Hiện nay, phiên bản mới nhất của RTOS SDK là V1.4.0 cung cấp hầu hết tất cả các Api của Non-OS SDK ngoại trừ việc không hỗ trợ AT command.[5]



##### Hình 2.18 Bộ thư viện ESP8266 NONOS\_SDK

Cấu trúc thư mục của Non-OS SDK như sau:

**app**: Thư mục làm việc chính chứa tất cả source code sẽ được compile của ứng dụng.

**bin**: Chứa firmware đã được compile và nạp trực tiếp xuống flash

**documents**: các tài liệu liên quan đến SDK

**driver\_lib**: Các thư viện hỗ trợ các thiết bị ngoại vi như UART, I2C và GPIO.

**examples**: Chứa các ví dụ mẫu để tham khảo.

**include**: Chứa các header file sử dụng trong SDK. Các tập tin này cũng định nghĩa các API cũng như các macro mà SDK hỗ trợ. Người sử dụng không nên sửa các file này

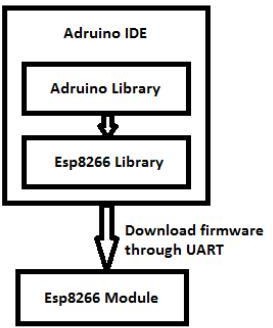
**ld**: Chứa các file tương tác với các phần mềm chạy SDK. Nhà sản xuất khuyến cáo không nên sửa đổi nó trừ khi có những yêu cầu đặc biệt.

**lib**: Chứa các thư viện hỗ trợ bởi SDK như cấu hình chip, Wifi config, ... đến các tính năng cao cấp như Mesh, SmartConfig, mDNS, Sniffer, …

**tools**: Chứa các script cần thiết cho quá trình compile firmware: script tạo file binary, script tạo certificate, … Nhà sản xuất khuyến cáo người sử dụng không nên sửa các file này.

* + - 1. Sử dụng firmware thư viện từ cộng đồng phát triển

Ngoài 2 thư viện SDK chuẩn do Espressif cung cấp, cộng đồng phát triển của ESP8266 cũng góp phần tạo ra các thư viện cũng như môi trường lập trình khác thân thiện hơn giúp cho chúng ta có thể tiếp cận và sử dụng module ESP8266 một cách dễ dàng hơn.



##### Hình 2.19 Mô hình sử dụng IDE của Arduino lập trình cho ESP8266

Bộ thư viện Arduino cho ESP8266 hỗ trợ các hàm chuẩn để làm việc với Digital IO, Analog IO, Software I2C, UART, timer. Ngoài ra, bộ thư viện còn hỗ trợ các hàm API của riêng ESP8266 như các hàm Wifi, các hàm để lấy thông tin phần cứng trên ESP, các hàm để làm việc với files/folders, và các hàm để thực hiện cả OTA.

Một số lệnh cơ bản trong Arduino IDE

Các hàm chuẩn cho digital IO:

*pinMode(pin, Mode) //pin = 0 cho GPIO0, 1 cho GPIO01 và 2 cho GPIO02*

*digitalRead(pin) digitalWrite(pin, value)*

*attachInterrupt(digitalPinToInterrupt(pin), callback\_func, mode); //Lưu ý không có*

*interrupt cho GPIO16 deattachInterrupt(digitalPinToInterrupt(pin)*

Các hàm chuẩn cho Analog IO:

*analogRead(A0) //trả về 0-1023 tương ứng với 0-1V*

*analogWrite(pin, value) // tạo điện áp analog tại chân GPIO, value từ 0- 1023 analogWriteFreq(new\_frequency) //thay đổi tần số 1KHz mặc định của PWM để tạo analog điện áp*

***Các hàm chuẩn cho timer:*** *delay(ms) delayMicroseconds(us)*

Các hàm chuẩn cho UART:

*Serial.begin(baudrate) // và Serial1.begin(baudrate)*

*Serial.print(), Serial.println(), Serial.readByte(), Serial.readString(), v.v…*

Các hàm chuẩn cho I2C:

*Wire.begin(sda\_pin, scl\_pin) //thay đổi chân mặc định chân GPIO4, GPIO5 cho I2C*

*Wire.beginTransaction(),Wire.write(), Wire.read(), Wire.endTransaction(), v.v….*

***Các hàm đọc thông tin phần cứng trên ESP8266:*** *ESP.restart() //reset ESP ESP.getResetReason()*

*ESP.getChipId() ESP.getFlashChipSize() ESP.getFlashSpeed()*

Các hàm API cho Wifi:

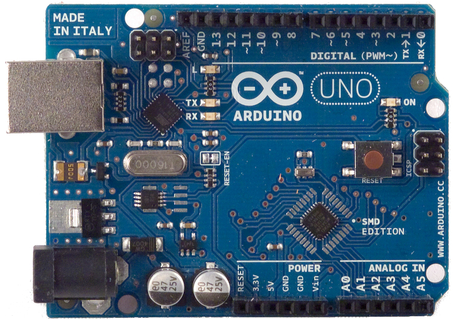
*WiFi.mode(mode) //chọn mode là thiết bị Wifi, Access Point, hoặc Soft Access Point WiFi.softAP(ssid, password) //thiết lập thông tin mạng Wifi WiFi.macAddress(mac) //thay đổi MAC address khi làm thiết bị Wifi WiFi.softAPmacAddress(mac) //thay đổi MAC address khi làm Access Point*

*Wifi.localIP() //đọc địa chỉ IP khi hoạt động như 1 thiết bị Wifi*

*Wifi.softAPIP() //đọc địa chỉ IP khi hoạt động như 1 Access Point WifiUDP.xxxx() // hỗ trợ truyền nhận dữ liệu qua UDP*

## Giới thiệu về ARDUINO UNO

* + 1. Giới thiệu chung



##### Hình 2.20 Mạch Arduino Uno

Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3).

Một vài thông số của Arduino UNO R3

* Vi điều khiển ATmega328 họ 8bit
* Điện áp hoạt động 5V DC (chỉ được cấp qua cổng USB)
* Tần số hoạt động 16 MHz
* Dòng tiêu thụ khoảng 30mA
* Điện áp vào khuyên dùng 7-12V DC
* Điện áp vào giới hạn 6-20V DC
* Số chân Digital I/O 14 (6 chân hardware PWM)
* Số chân Analog 6 (độ phân giải 10bit)
* Dòng tối đa trên mỗi chân I/O 30 mA
* Dòng ra tối đa (5V) 500 mA
* Dòng ra tối đa (3.3V) 50 mA
* Bộ nhớ flash 32 KB (ATmega328) với 0.5KB dùng bởi bootloader
* SRAM 2 KB (ATmega328)
* EEPROM 1 KB (ATmega328)

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lí những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lí tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,… hay những ứng dụng khác mà bạn đã được xem ở đây.

Thiết kế tiêu chuẩn của Arduino UNO sử dụng vi điều khiển ATmega328 với giá khoảng 90.000đ. Tuy nhiên nếu yêu cầu phần cứng của bạn không cao hoặc túi tiền không cho phép, bạn có thể sử dụng các loại vi điều khiển khác có chức năng tương đương nhưng rẻ hơn như ATmega8 (bộ nhớ flash 8KB) với giá khoảng 45.000đ hoặc ATmega168 (bộ nhớ flash 16KB) với giá khoảng 65.000đ.

Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lí nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

Các chân năng lượng:

* GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
* 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
* 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
* Vin (Voltage Input): để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
* IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
* RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

Lưu ý: Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó bạn phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO.

Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích. Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board. Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328. Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển. Cấp điệp áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển. Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

Vi điều khiển Atmega328 của Arduino Uno cung cấp cho người dùng:

* 32KB bộ nhớ Flash: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
* 2KB cho SRAM (Static Random Access Memory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
* 1KB cho EEPROM (Electrically Eraseble Programmable Read Only Memory): đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

Các cổng vào/ra: Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

* chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
* Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
* Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
* LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.
* Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10bit.
* Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

Lập trình cho Arduino: các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “ngôn ngữ Arduino”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ thở đối với bạn.

Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cũng cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (Intergrated Development Environment).

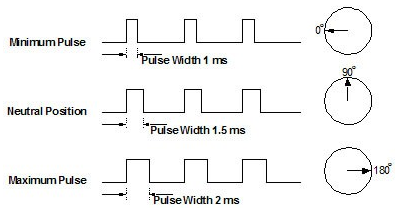
* + 1. Sơ lược các chức năng trong luận văn
    2. PPM

Xung PPM (Pulse Position Modulation) được sử dụng để điều khiển servo. Về bản chất PPM cũng là một xung..

Xung PPM khác với PWM ở chỗ:

* Tần số thông thường có giá trị trong khoảng 50Hz (20 mili giây), không quan trọng
* Thời gian xung ở mức cao chỉ từ 1ms đến 2ms, rất quan trọng.
* Có thể có nhiều hơn 1 sự thay đổi trạng thái điện cao/thấp

Nắm bắt được 2 ý trên ta đã có thể phân biệt được xung PPM và xung PWM giống nhau và khác nhau như thế nào.



##### Hình 2.21 Thời gian xung ở mức cao quy định góc quay của RC servo

Với thời gian 1ms mức cao, góc quay của servo là 0, 1.5ms góc quay 90 và 2ms góc quay là 180. Các góc khác từ 0-180 được xác định trong khoảng thời gian 1-2ms.

Lưu ý: có thể ghép nhiều xung trong cùng 1 thời gian là 20ms để xác định vị trí góc của nhiều servo cùng 1 lúc. Tối đa là 10 servo.

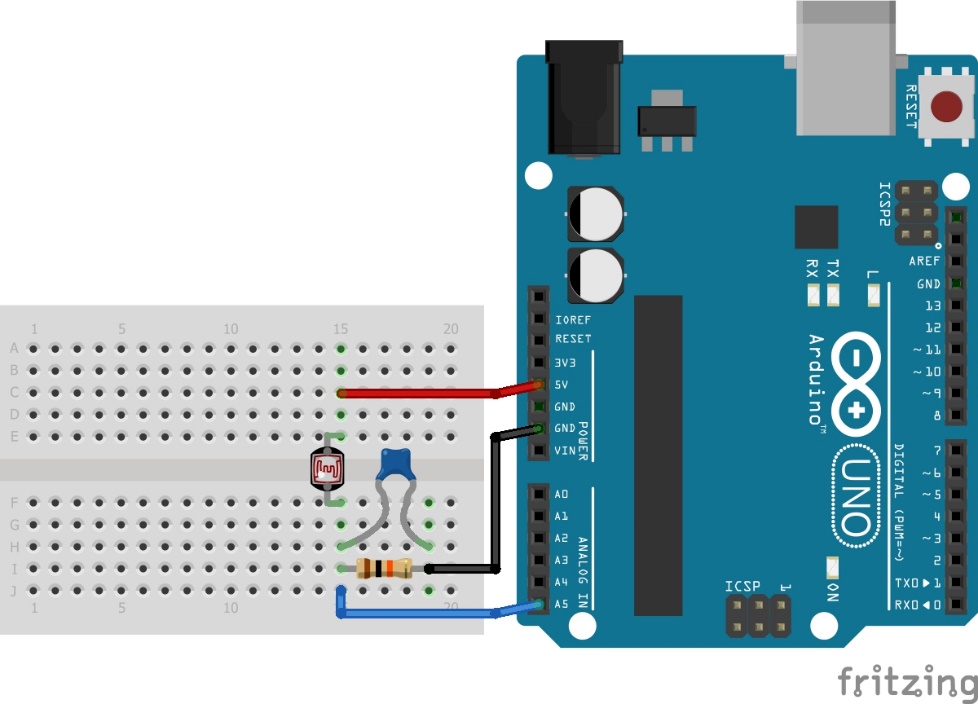
* + 1. Analog/ADC

Trên mạch Arduino UNO có 6 chân Analog In, được kí hiệu từ A0 đến A5. Trên các mạch khác cũng có những chân tương tự như vậy với tiền tố "A" đứng đầu, sau đó là số hiệu của chân. **A**nalogRead() luôn trả về 1 số nguyên nằm trong khoảng từ 0 đến 1023 tương ứng với thang điện áp (mặc định) từ 0 đến 5V. Ta có thể điều chỉnh thang điện áp này bằng hàm analogReference().Hàm analogRead() cần 100 micro giây để thực hiện.

Trong đề tài, các chân analog của Arduino Uno được dùng để kết nối với quang trở. Có tổng cộng 4 quang trở được nối từ A0 đến A3, với mục tiêu tạo thành một mạch 4 cảm biến ánh sáng liên tiếp để tiếp nhận ánh sáng năng lượng mặt trời.

Để tạo thành cảm biến ánh sáng, ta sử dụng phương pháp có tên gọi là đọc hiệu điện thế từ cầu phân áp, xem kỹ hơn về phần mạch cầu phân áp này ở phần mạch đo áp DC.

Ta xem quang trở là điện trở R1, vào phía trước mỗi chân Analog của Arduino có một điện trở, và xem đó là điện trở R2 = 10k). Ta có cách mắc đơn cho từng quang trở để nó biến thành cảm biến ánh sáng như sau.



##### Hình 2.22 Cách mắc đơn cho từng quang trở

Tương tự cách mắc đối với những quang trở còn lại. Mạch 4 cảm biến ánh sáng sẽ được trình bày rõ hơn trong phần cứng.

## 2.5 Giới thiệu về Website (Web)

### Cơ sở dữ liệu

Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thõa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau. Các thông tin có thể được lưu trữ dưới dạng bảng hoặc dạng chuỗi JSON. Về cơ bản cấu trúc của một cơ sở dữ liệu có thể bao gồm các thành phần chính:

* Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thõa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau. Các thông tin có thể được lưu trữ dưới dạng bảng hoặc dạng chuỗi JSON. Về cơ bản cấu trúc của một cơ sở dữ liệu có thể bao gồm các thành phần chính:
* Database Name: Là tên của dữ liệu đã tạo trên máy chủ lưu trữ nó.
* Database Username và Database Password: là những thông tin cần thiết để kiểm tra quyền truy cập vào database của người dùng

### Web Server

**Web Server** nghĩa là máy chủ. Web server là máy tính lớn được kết nối với tập hợp mạng máy tính mở rộng. Đây là một dạng máy chủ trên internet mỗi máy chủ là một IP khác nhau và có thể đọc các ngôn ngữ như file \*.htm và \*.html… Tóm lại máy chủ là kho để chứa toàn bộ dữ liệu hoạt động trên internet mà nó được giao quyền quản lý.

**Web server** phải là một máy tính có dung lượng lớn, tốc độ rất cao để có thể lưu trữ vận hành tốt một kho dữ liệu trên internet. Nó sẽ điều hành trơn tru cho một hệ thống máy tính hoạt động trên internet, thông qua các cổng giao tiếp riêng biệt của mỗi máy chủ. Các web server này phải đảm bảo hoạt động liên tục không ngừng nghỉ để duy trì cung cấp dữ liệu cho mạng lưới máy tính của mình.

Đôi nét về Web Server:

* Web server có thể xử lý dữ liệu và cung cấp thông tin đến máy khách thông qua các máy tính cá nhân trên môi trường Internet qua giao thức HTTP, giao thức được thiết kế để gửi các file đến trình duyệt Web, và các giao thức khác. (Ví dụ: khi các bạn truy cập vào trang web vinahost.vn máy chủ sẽ cung cấp đến các bạn tất cả dữ liệu về trang web đó thông qua lệnh giao tiếp).
* Máy tính nào cũng có thể là một máy chủ nếu cài đặt lên nó một chương trình phần mềm Server Software và sau đó kết nối vào Internet
* Phần mềm Web Server Software cũng giống như các phần mềm khác, nó dùng để cài đặt và chạy trên bất kì máy tính nào đáp ứng đủ yêu cầu về bộ nhớ. Nhờ có chương trình này mà người sử dụng có thể truy cập đến các thông tin của trang Web từ một máy tính khác ở trên mạng
* Khi là SEO chúng ta thường gặp các máy chủ nhỏ, [máy chủ ảo](https://vinahost.vn/vps.html) và thông thường chúng ta hay thuê máy chủ dạng VPS hay Hosting để lưu giữ liệu trang web của mình.

Tóm lại, web server có thể là phần cứng hoặc phần mềm hoặc cả hai với những đặc điểm sau:

* **Ở khía cạnh phần cứng**: một web server là một máy tính lưu trữ các file thành phần của một website (ví dụ: các tài liệu HTML, các file ảnh, CSS và các file JavaScript) và có thể phân phát chúng tới thiết bị của người dùng cuối (end-user). Nó kết nối tới mạng Internet và có thể truy cập tới thông qua một tên miền giống như mozilla.org.
* **Ở khía cạnh phần mềm**: một web server bao gồm một số phần để điều khiển cách người sử dụng web truy cập tới các file được lưu trữ trên một HTTP server (máy chủ HTTP). Một *HTTP server* là một phần mềm hiểu được các URL (các địa chỉ web) và HTTP (giao thức trình duyệt của bạn sử dụng để xem các trang web).
* Ở mức cơ bản nhất, bất cứ khi nào một trình duyệt cần một file được lưu trữ trên một web server, trình duyệt request (yêu cầu) file đó thông qua HTTP. Khi một request tới đúng web server (phần cứng), HTTP server (phần mềm) gửi tài liệu được yêu cầu trở lại, cũng thông qua HTTP.

### Lập trình Website

Lập trình web là công việc có nhiệm vụ nhận toàn bộ dữ liệu từ bộ phận thiết kế web để chuyển thành một hệ thống website hoàn chỉnh có tương tác với CSDL và tương tác với người dùng dựa trên ngôn ngữ máy tính. Sau khi xây dựng trang web xong thì có thể lập trình viên sẽ được phân công quản trị website, khi đó bạn cần trang bị thêm một vài [công cụ quản trị web](http://www.ezwebsitemonitoring.com/mot-so-cong-cu-quan-ly-website-tot-nhat-hien-nay/), nhằm giúp cho việc quản trị trở nên dễ dàng hơn. Các công cụ đó sẽ hỗ trợ bạn kiểm tra những lần uptime, downtime, tỷ lệ thoát trang web, nguồn traffic đổ vào website, hoặc tình trạng quá tải băng thông. Các ngôn ngữ lập trình phổ biến nhất là PHP, ASP.NET, Ruby on Rails, Perl, Java, Javascripts, … Trong phạm vi nghiên cứu. luận văn đề cập tới ngôn ngữ Javascripts hay đúng hơn là NodeJS cho cả phía client và server. Ngoài ra còn kết hợp thêm ngôn ngữ đánh dấu mã kịch bản HTML (*Hyper Text Markup Language)* và CSS (Cascading Style Sheet) dùng để trình bày bổ cục của một Website dựa vào các thẻ HTML.

Ở phía client sẽ dùng framework Angular 2 – một trong 3 framework phổ biến nhất hiện tại ở phía client để xử lý các tác vụ logic phía client và giao diện cho user. Ở phía server sẽ dùng framework ExpressJS, một framework hỗ trợ việc tạo một server với ngôn ngữ Javascript (hoặc Typescript) dựa trên nền tảng của NodeJS.

### Tìm hiểu về HTML

HTML là chữ viết tắt của cụm từ **H**yper**T**ext **M**arkup **L**anguage((Xem thêm tại <http://vi.wikipedia.org/wiki/HTML>)) (dịch là Ngôn ngữ đánh dấu [siêu văn bản](http://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n)) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thi thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là **Tim Berners-Lee**, cũng là người khai sinh ra World Wide Web và chủ tịch của **World Wide Web Consortium** (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc ngọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> dụ </strong>). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong>Đây là chữ in đậm</strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>). Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là **.html** hoặc **.htm**.

### Tìm hiểu về CSS (LESS, SASS, …)

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng, thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc, rất nhiều.

### Tìm hiểu framework Angular 2+

#### Angular 2+ là gì ?

Angular 2 là 1 framework UI để xây dựng ứng dụng web trên desktop và mobile.

Nó được xây dựng dựa trên Javascript. Chúng ta có thể dùng nó để xây dựng 1 ứng dụng client side thú vị dùng HTML, CSS và Javascript.

Angular 2 có rất nhiều cải tiến so với Angular 1 (AngularJS) để dễ dàng học và phát triển các ứng dụng quy mô doanh nghiệp.

Với Angular 2 thì chúng ta dễ dàng xây dựng được 1 ứng dụng có thể dễ dàng mở rộng, bảo trì, kiểm nghiệm và chuẩn hóa ứng dụng của mình.

#### Angular 2 + có những tính năng gì ?

***Two-way data binding:*** Đây là 1 trong những tính năng tuyệt với nhất trong angular 2. Dữ liệu được binding một cách tự động và nhanh chóng, những thay đổi trong view sẽ được tự động cập nhật vào trong các component class.

***Powerful routing support:*** Angular 2 hỗ trợ mạnh mẽ các routing thông qua cách tải trang không đồng bộ trên cùng 1 trang cho phép chúng ta tạo ra 1 single page application.

***Expressive HTML:*** Angular 2 cho phép chúng ta dùng các cấu trúc lập trình như câu lệnh if, vòng lặp for, ... để render và kiểm soát các trang HTML.

***Modular by design:*** Angular 2 được thiết kế theo hướng modul hóa để tổ chức và quản lý code 1 cách tốt hơn.

***Built in back end support:***Angular 2 được xây dựng để hỗ trợ việc giao tiếp với back-end servers và thực thi bất kỳ business logic hoặc lấy dữ liệu.

***Active community:*** Angular 2 được hỗ trợ bởi google và có 1 cộng đồng đông đảo sẵn sàng hỗ trợ và giải đáp bất cứ câu hỏi nào của bạn.

### Tìm hiểu về NodeJS

**Node.js** là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì bạn cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

Trong một môi trường server điển hình LAMP (Linux-Apache-MySQL-PHP), bạn có một web server là Apache hoặc NGINX nằm dưới, cùng với PHP chạy trên nó. Mỗi một kết nối tới server sẽ sinh ra một thread mới, và điều này khiến ứng dụng nhanh chóng trở nên chậm chạp hoặc quá tải - cách duy nhất để hỗ trợ nhiều người dùng hơn là bằng cách bổ sung thêm nhiều máy chủ. Đơn giản là nó không có khả năng mở rộng tốt. Nhưng với Node.js thì điều này không phải là vấn đề. Không có một máy chủ Apache lắng nghe các kết nối tới và trả về mã trạng thái HTTP - bạn sẽ phải tự quản lý kiến trúc lõi của máy chủ đó. May mắn thay, có một số module giúp thực hiện điều này được dễ dàng hơn, nhưng công việc này vẫn gây cho bạn một chút khó khăn khi mới bắt đầu. Tuy nhiên, kết quả thu được là một ứng dụng web có tốc độ thực thi cao.

**JavaScript** là một ngôn ngữ dựa trên sự kiện, vì vậy bất cứ thứ gì xảy ra trên server đều tạo ra một sự kiện non-blocking. Mỗi kết nối mới sinh ra một sự kiện; dữ liệu nhận được từ một upload form sinh ra một sự kiện data-received; việc truy vấn dữ liệu từ database cũng sinh ra một sự kiện. Trong thực tế, điều này có nghĩa là một trang web Node.js sẽ chẳng bao giờ bị khóa (lock up) và có thể hỗ trợ cho hàng chục nghìn user truy cập cùng lúc. Node.js đóng vai trò của server - Apache - và thông dịch mã ứng dụng chạy trên nó. Giống như Apache, có rất nhiều module (thư viện) có thể được cài đặt để bổ sung thêm các đặc trưng và chức năng - như lưu trữ dữ liệu, hỗ trợ file Zip, đăng nhập bằng Facebook, hoặc các cổng thanh toán. Dĩ nhiên, nó không có nhiều thư viện như PHP, nhưng Node.js vẫn đang ở trong giai đoạn ban đầu và có một cộng đồng rất mạnh mẽ ở đằng sau nó.

### Tìm hiểu về ExpressJS

#### ExpressJS là gì ?

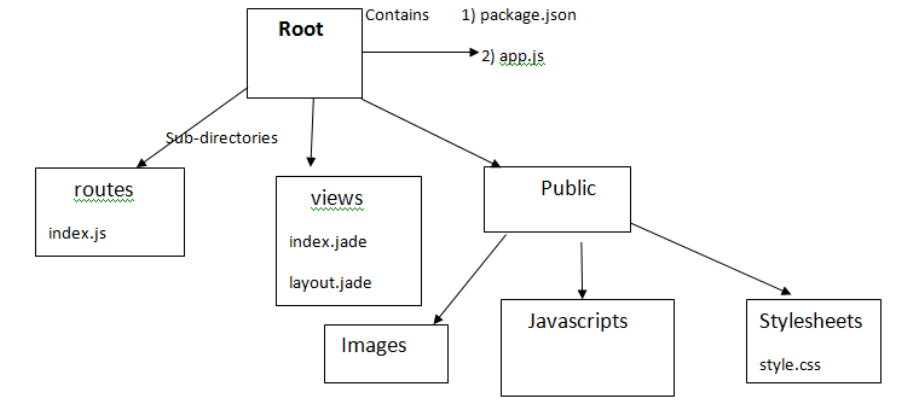
Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile

Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.

Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.

Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN, ...

#### Cấu trúc của ExpressJS



##### Hình 2.23 Cấu trúc ExpressJS

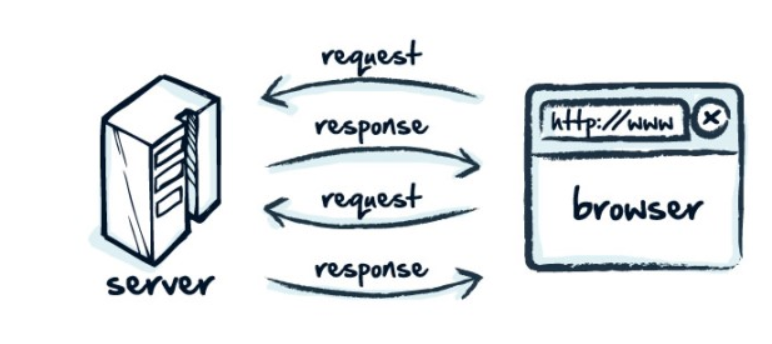
Cấu trúc của express js vô cùng đơn giản:

* Root:
* app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa,... để ứng dụng của chúng ta có thể chạy một cách tốt nhất.
* package.json chứa các package cho ứng dụng chạy. Trong PHP hoặc RoR thì file này có chức năng tương tự như composer.json hoặc Gemfile
* Folder routes: chứa các route có trong ứng dụng.
* Folder view: chứa view/template cho ứng dụng. Riêng phần này chúng ta sẽ xử dụng Angular 2 cho phần view (cùng với các tác vụ logic phía client).
* Folder public chứa các file css, js, images, ...cho ứng dụng

### Giao thức HTTP và HTTPs:

#### Giao thức HTTP:

**HTTP** là chữ viết tắt của HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Đây là một giao thức ứng dụng trong bộ các giao thức TCP/IP (gồm một nhóm các giao thức nền tảng cho internet).



##### Hình 2.24 Giao thức HTTP

**HTTP** hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này. Để có thể nói chuyện được với nhau, các máy chủ và máy khách phải thực hiện việc trao đổi thông qua các giao thức. Một trong những giao thức được sử dụng thường xuyên nhất chính là HTTP.

Khi bạn gõ một địa chỉ Web URL vào trình duyệt Web, một lệnh HTTP sẽ được gửi tới Web server để ra lệnh và hướng dẫn nó tìm đúng trang Web được yêu cầu. Trang Web này sau đó sẽ được kéo về và mở trên trình duyệt Web. Nói đơn giản hơn, HTTP là giao thức giúp cho việc truyền tải file từ một Web server vào một trình duyệt Web để người dùng có thể xem một trang Web đang hiện diện trên trình duyệt.

#### Giao thức HTTPs:

**HTTPS** viết tắt của Hyper Text Transfer Protocol Secure (giao thức truyền tải siêu văn bản bảo mật) là phiên bản an toàn của HTTP, giao thức mà nhờ đó dữ liệu được gửi giữa trình duyệt và trang web bạn đang kết nối. Chữ 'S' ở cuối HTTPS là viết tắt của "Secure" (Bảo mật). Nó có nghĩa là tất cả các giao tiếp giữa trình duyệt và trang web đều được mã hóa. HTTPS thường được sử dụng để bảo vệ các giao dịch trực tuyến có tính bảo mật cao như giao dịch ngân hàng và đặt hàng mua sắm trực tuyến.

Các trình duyệt web như Internet Explorer, Firefox và Chrome cũng hiển thị biểu tượng ổ khóa trên thanh địa chỉ để cho thấy một kết nối HTTPS có hiệu lực.

### Tìm hiểu về Socket Server và Socket Client

#### Socket là gì ?

Có 1 vài cách định nghĩa như sau :

* Cách sử dụng standard Unix file descriptors để giao tiếp với các chương trình khác. Hoặc như sau:
* Socket is method for accomplishing interprocess communcation(IPC). Nghĩa là một socket được sử dụng để cho phép 1 process nói chuyện với 1 process khác. Rất giống với định nghĩa 1 chiếc điện thoại - cho phép 1 người nói chuyện với 1 người khác.

=> Socket là một cổng logic mà một chương trình sử dụng để kết nối với một chương trình khác chạy trên một máy tính khác trên Internet. Chương trình mạng có thể sử dụng nhiều. Socket cùng một lúc, nhờ đó nhiều chương trình có thể sử dụng Internet cùng một lúc.

#### Có bao nhiêu loại socket ?

**Stream Socket:***Dựa trên giao thức TCP( Tranmission Control Protocol) việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.*

**Datagram Socket:***Dựa trên giao thức UDP( User Datagram Protocol) việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 quá trình. Ngược lại với giao thức TCP thì dữ liệu được truyền theo giao thức UDP không được tin cậy, có thế không đúng trình tự và lặp lại. Tuy nhiên vì nó không yêu cầu thiết lập kết nối không phải có những cơ chế phức tạp nên tốc độ nhanh…ứng dụng cho các ứng dụng truyền dữ liệu nhanh như chat, game…..*

#### Lý do chọn mô hình Socket

* Một lệnh HTTP truyền thống cần phải có thời gian thiết lập kết nối, điều đó là bắt buộc. Sau đó, tuần tự các phép chuyển dữ liệu mới diễn ra. Trong khi đó, websocket chỉ cần thiết đặt một lần duy nhất và khi kết nối còn giữ thì mọi việc diễn ra rất nhanh, dữ liệu được gửi đi ngay tức khắc với độ trễ chỉ khoảng 1-2 mili giây.
* Việc trao đổi diễn ra hai chiều: Nghĩa là server có thể tạo ra một sự kiện ở client.

Vậy chỉ cần chuẩn bị một Socket Server ở ngoài Internet và cài đặt ESP8266 trở thành một Socket Client ở tại một địa điểm trong nhà và chúng ta đã có một hệ thống Internet of Things phù hợp trong việc nghiên cứu và giải trí. Ngoài việc sử dụng Socket, chúng ta có thể sử dụng phương thức MQTT để truyền tín hiệu. Bản chất MQTT vẫn sử dụng Socket và có thêm các tính năng phù hợp trong công nghiệp như khả năng phân quyền, dễ dàng mở rộng và độ chịu lỗi lớn. Tuy nhiên, chúng ta sẽ không bàn đến MQTT trong dự án này.

## Giới thiệu về hệ thống điều hướng năng lượng mặt trời

Hệ thống điều hướng năng lượng mặt trời là một thiết bị cụ thể nhằm di chuyển các mô đun năng lượng theo cách chúng liên tục đối mặt mặt trời với mục đích tối đa hóa sự chiếu xạ mà mảng PV nhận được. Một hệ thống theo dõi năng lượng mặt trời bao gồm ba thành phần khác biệt: cơ cấu xoay, động cơ lái và bộ điều khiển theo dõi.

### 2.6.1 Cơ cấu xoay

Cơ cấu này là một phần của hệ thống theo dõi chịu trách nhiệm đảm bảo độ chính xác trong việc theo dõi hướng đi của mặt trời. Phần này phải được thiết kế để chịu được các điều kiện thời tiết khắc nghiệt và nó cũng phải có tuổi thọ hữu ích ước tính bằng hoặc tương đương với tấm pin năng lượng mặt trời.

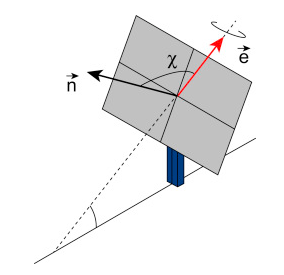


##### Hình 2.25 Trụ đặt tấm pin mặt trời

Khi phân loại các cơ cấu xoay, hệ thống điều hướng pin mặt trời có thể được phân thành 2 loại, hệ thống điều hướng đơn và hệ thống điều hướng kép

#### 2.6.1.1 Hệ thống điều hướng một trục

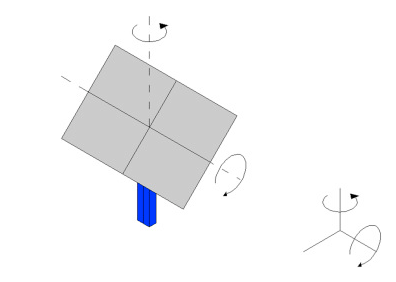
Đối với hệ thống điều hướng năng lượng mặt trời, các bộ phận chuyển động và hệ thống điều khiển thường đắt tiền, vì vậy hệ thống theo dõi trục đơn dường như là giải pháp tốt nhất cho các nhà máy điện năng lượng mặt trời nhỏ. Hệ thống theo dõi năng lượng mặt trời một trục sử dụng giá treo bảng Photovoltaic nghiêng và một động cơ điện để di chuyển bảng trên quỹ đạo gần đúng so với vị trí của Mặt trời. Trục xoay có thể là ngang, dọc hoặc xiên. Hình dưới đây cho thấy sơ đồ tổng quát của bộ điều hướng một trục hiển thị cả trục xoay (vectơ đơn vị e) và mặt phẳng collector (vectơ đơn vị bình thường đối với mặt phẳng collector). Góc giữa hai vectơ đơn vị này thường được giữ không đổi trong loại trình theo dõi này.



##### Hình 2.26 Hệ thống điều hướng một trục

#### 2.6.1.2 Hệ thống điều hướng kép (2 trục)

Sử dụng hệ thống điều hướng kép có thể đạt được mức thu năng lượng tối đa bởi vì tấm pin mặt trời hoàn toàn tự do di chuyển (phía bắc Nam và phía đông tây), các tấm photovoltaic có thể đối mặt với tia nắng mặt trời suốt cả ngày.



##### Hình 2.27 Hệ thống điều hướng kép

### 2.6.2 Động cơ lái

Tùy thuộc vào cơ chế lái mà hệ thống điều hướng năng lượng mặt trời có thể được phân loại thành điều hướng chủ động hoặc thụ động. Cơ chế lái chủ động có thể dựa trên động cơ điện (DC hoặc AC) hoặc hệ thống thủy lực, trong khi đó hệ thống thụ động dựa trên lực hấp dẫn.

|  |  |  |
| --- | --- | --- |
|  | Điều hướng chủ động | Điều hướng thụ động |
| Cơ chế | Cho phép điều khiển chuyển động đơn giản và chính xác thông qua động cơ điện xoay chiều hoặc một chiều. Đối với máy theo dõi một trục, chỉ cần một động cơ, trong khi đối với máy theo dõi hai trục, cần có hai động cơ. | Hoạt động nhờ mực di chuyển chất lỏng từ bên này sang bên kia. Cho phép trọng lực xoay bộ khung. |
| Giá thành | Đắt | Rẻ hơn |
| Ưu điểm | - Tận dụng được tối đa nguồn sáng  - Nhanh chóng và chính xác | - Giá thành rẻ  - Không có bánh răng, không có điều khiển lỗi  - Ít phải bảo trì |
| Khuyết điểm | - Giá thành cao  - Phải bảo trì, sửa chữa thường xuyên | - Phụ thuộc nhiều vào nhiệt độ  - Độ linh hoạt không cao, hiệu suất thấp |
| Hình ảnh |  |  |

Cơ chế lái được ưa chuộng nhất hiện này vẫn là sử dụng động cơ điện vì tính linh hoạt và hiệu suất của nó.

### 2.6.3 Bộ điều khiển theo dõi năng lượng mặt trời

Tùy thuộc vào đầu vào của qua trình theo dõi, các bộ điều khiển theo dõi năng lượng mặt trời có thể được phân loại thành các hệ thống theo dõi năng lượng mặt trời định hướng các tấm PV dựa trên các quỹ đạo mặt trời được tính toán trước đó (điều khiển vòng hở) và các máy theo dõi năng lượng mặt trời sử dụng cảm biến bức xạ mặt trời để điều khiển hướng của hệ thống (đóng- điều khiển vòng lặp).

#### 2.6.3.1 Bộ điều khiển phản hồi

Các bộ điều khiển này dựa trên việc sử dụng các cảm biến năng lượng mặt trời trực tiếp để phát hiện vị trí của Mặt trời. Đối với mục đích này, các cảm biến bao gồm các cảm biến ánh sáng được gắn trên bảng điều khiển. Cả lúc bình minh và trong điều kiện phơi nắng thấp (ngày nhiều mây), các máy theo dõi được điều khiển bởi cảm biến ánh sáng có thể bị mất phương hướng. Do đó, điều cần thiết là sử dụng một hệ thống theo dõi phụ trợ, điều khiển vòng quay của động cơ cho đến khi định hướng được khôi phục lại.

#### 2.6.3.2 Bộ điều khiển vòng hở

Kỹ thuật điều khiển này cũng sử dụng bộ vi xử lý, nhưng nó không cần bất kỳ cảm biến nào để xác định vị trí của Mặt trời. Chuyển động của Mặt trời có thể được dự đoán bằng các mối quan hệ thiên văn, được lập trình trong bộ vi xử lý để bản thân bộ vi xử lý có thể tính toán vị trí của Mặt trời bất cứ lúc nào. Loại điều khiển này không bị ảnh hưởng bởi mây hoặc các trường hợp bức xạ thấp khác có thể dẫn đến sai sót về độ chính xác.

# **THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG**

Chương này dùng để giới thiệu về phần cứng của đề tài, sơ đồ khối và cách hoạt động phần cứng.

## Sơ đồ khối tổng quát hệ thống

Yêu cầu đặt ra:

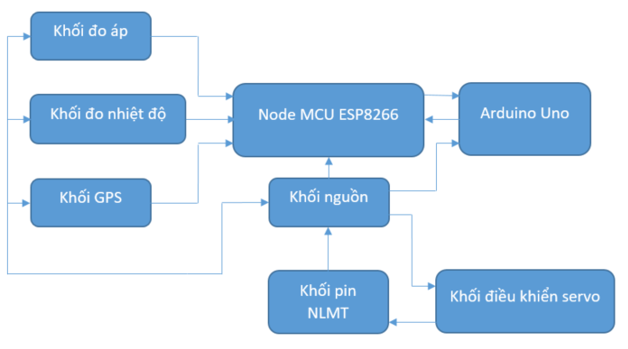
- Thực hiện mô hình hoạt động ổn định liên tục 24/24

- Thực hiện việc nhận dữ liệu từ hệ thống năng lượng mặt trời an toàn, chính xác, liên tục.

- Thực hiện điều hướng pin mặt trời nhanh và chính xác.

- Tính tương tác người dùng cao

- Giá thành rẻ để có thể phổ biến rộng rãi



##### Hình 3.1 Sơ đồ khối hệ thống phần cứng

Phân tích: hệ thống gồm tám phần

Khối đo áp: khối này có nhiệm vụ chuyển tín hiệu điện áp thực sang tín hiệu vi điều khiển.

Khối đo nhiệt độ: khối này có nhiệm vụ chuyển tín hiệu nhiệt độ sang tín hiệu vi điều khiển.

Khối GPS: Khối này có nhiệm vụ nhận thông tin về tọa độ GPS từ vệ tinh và chuyển sang vi điều khiển.

Khối pin năng lượng mặt trời (pin NLMT): Nhận ánh sáng mặt trời, chuyển thành điện DC và lưu trữ vào bình ắc quy.

Khối điều khiển servo: khối này có nhiệm vụ chuyển thông tin quang thông qua vi điều khiển, trở thành góc quay cho 2 động cơ servo.

Node MCU ESP8266: đọc input, xử lý, tính toán và chuyển data đến server.

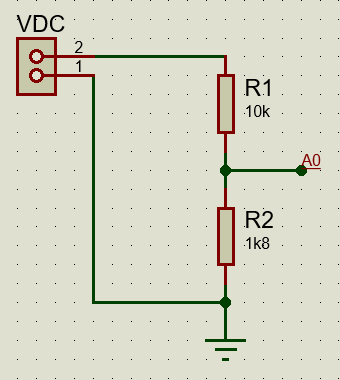
Arduno Uno: Chuyển thông tin từ quang trở thành góc quay, gửi tín hiệu điều khiển đến động cơ servo.

Khối Power: cung cấp điện cho toàn bộ các khối.

## Sơ đồ mạch chi tiết

### Khối đo áp

Khối đo điện áp DC thì ta chỉ cần qua cầu phân áp, làm giảm điện áp, để trong tầm vi điều khiển từ 0V - 3.3V là mức điện áp của chân analog Node MCU ESP8266.



##### Hình 3.2 Schematic khối đo áp DC

Cường độ dòng điện: I=

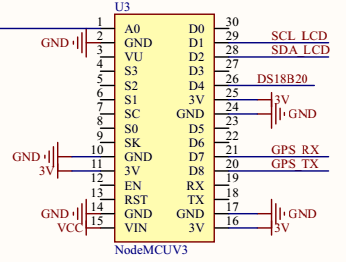
Điện áp ra tối đa tại chân A0:

=

Chọn R1 = 10k𝞨, R2 = 1k8𝞨

= = 2.6V

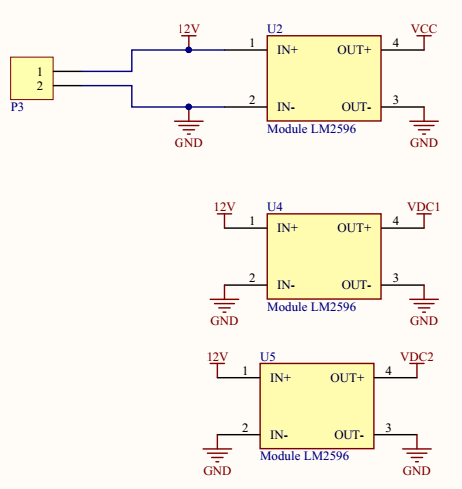
### Khối Node MCU ESP8266



##### Hình 3.3 Schematic khối Node MCU ESP8266

Khối Node MCU ESP8266 nhận thông tin từ khối tính áp, nhiệt độ và GPS. Một mặt chuyển thông tin lên LCD, mặt khác thông qua wifi để chuyển đến Server.

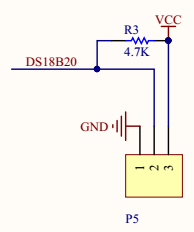
### Khối Power

****

##### Hình 3.4 Schematic khối Node MCU ESP8266

Từ pin mặt trời sẽ chuyển điện DC vào ắc quy. Ắc quy tích điện DC, đồng thời cung cấp nguồn 12VDC cho 3 module hạ áp LM2596. 3 module LM2596 hạ áp xuống 5V-1A, lần lượt cấp điện áp cho 2 động cơ servo, Node MCU ESP8266 và Arduino Uno.

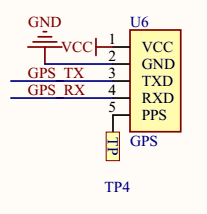
### Khối đo nhiệt độ

****

##### Hình 3.5 Schematic khối nhiệt độ

DS18B20 là cảm biến nhiệt độ, chuyển đổi trực tiếp sang tín hiệu digital cho Node MCU ESP8266.

### Khối GPS



##### Hình 3.6 Schematic khối GPS

Khối GPS gồm một angten GPS, module NEO 7. Khối này chuyển tín hiệu GPS Từ vệ tinh đến ESP8266 thông qua UART.

### Khối điều hướng kép

Khối này chịu trách nhiệm gửi thông tin đến Arduino Uno. Arduino sẽ xử lý và tính toán, sau đó chuyển thành số đo góc cho 2 động cơ thực hiện xoay ngang và dọc bằng cơ cấu trục xoay.

#### 3.2.7.1 Lựa chọn khung vật liệu

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nhôm | Sắt | Thép |
| Ưu điểm | - Khối lượng riêng: 2601 kg/  - Giá thành: 40.000VND/m  - Dễ khoan, dễ gia công, độ linh hoạt cao. | - Khối lượng riêng: 7850 kg/  - Giá thành: 32000VND/m  - Độ chắc và độ bền cao. | - Khối lượng riêng: 7850 kg/  - Giá thành: 74000VND/m  - Độ chắc và độ bền cao. |
| Khuyết điểm | - Chỉ có thể bắt vít. Không hàn được.  - Mối nối dễ lỏng | - Khó khoan, khó bắt vít.  - Độ linh động đối với sinh viên không cao, phụ thuộc nhiều vào thợ hàn. | - Khó khoan, khó bắt vít.  - Độ linh động đối với sinh viên không cao, phụ thuộc nhiều vào thợ hàn. |

Xét thấy hệ thống điều hướng nên ưu tiên vào sự linh hoạt, bộ khung nhỏ nhẹ. Vì vậy đề tài chọn nhôm làm bộ khung chính, các chi tiết gia cố thêm cho khung nhôm sẽ được nói ở phần sau.

#### 3.2.7.2 Thiết kế kích thước khung

- Kích thước pin NLMT:

48.7 cm

**Tấm pin NLMT**

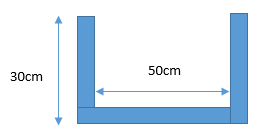
35,7 cm

- Thiết kế khung xoay dọc

Để đảm bảo tấm pin xoay dễ dàng và cân bằng lực ở mặt trước và sau. Ta thiết kế khung hình chữ U.

* Độ dài > (1/2) Độ dài pin NLMT= (1/2)\*48.7=24.35 (cm)
* Độ rộng trong > độ rộng pinNLMT=35.7cm

Cộng trừ >= 5cm cho cánh tay đòn của các motor. Ta có

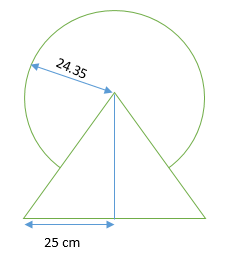
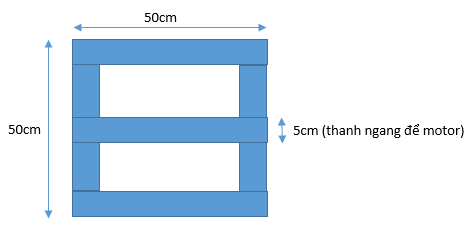


- Thiết kế đế

Chiều dài vươn ra tối đa của pin NLMT= 24.35\*2 = 48.7 (cm).

Để đế đứng vững thì ta phải có:

Kích thước đế >= kích thước vươn dài tối đa của pin NLMT.



Kích thước chân đế thực tế:

#### Cơ cấu trục xoay

Khi sử dụng động cơ servo, ta không nên để động cơ xoay trực tiếp cho tải bởi vì các nguyên nhân sau:

* Khó bảo trì và sửa chữa.
* Dễ hư động cơ do bánh răng chịu áp lực trực tiếp từ tải.
* Không thuật tiện khi lắp đặt vị trí xoay.
* Không giảm được lực xoay.

Chính vì những nguyên nhân trên, yêu cầu đặt ra phải có hệ thống trục xoay giảm lực cho động cơ servo, đồng thời chống trượt góc do sức nặng của tải, chống lại các tác nhân môi trường khác.

Thông qua tìm hiểu, đề tài quyết định chọn cơ cấu tuốc năng trong quạt công nghiệp để làm trục xoay của mình.



##### Hình 3.7 Tuốc năng trong quạt công nghiệp

Thông tin về tuốc năng:

* Tuốc năng là bộ phận làm cho quạt quay sang trái và phải một góc dưới 1800 gúp cho quạt thổi gió rộng hơn.
* Được cấu tạo từ bộ bánh răng lắp trong khung vỏ đúc bằng nhôm, dễ dàng thay thế khi bị hư hỏng.

Đặc biệt cơ cấu tuốc năng là cơ cấu xoay một chiều, chỉ cho phép tác động xoay từ động cơ servo chứ không bị xoay ngược lại bởi sức nặng của tải hay ngoại lực. Hoàn toàn phù hợp với yêu cầu đề tài đặt ra.

#### Động cơ

3.2.7.4.1 Lựa chọn động cơ

Khi lựa chọn động cơ, những yếu tố ta cần cân nhắc bao gồm góc quay động cơ, mức độ tải mà động cơ có thể chịu được.

|  |  |
| --- | --- |
| THÔNG SỐ TẢI | |
| Cân nặng tấm pin | 2 kg |
| Cân nặng động cơ xoay dọc phải chịu | 2kg (Sai số 0.2kg) |
| Cân nặng động cơ xoay ngang phải chịu | 3kg (Sai số 0.2kg) |

Công thức tính Momen lực:

M=F\*d với F là lực, d là độ dài cánh tay đòn.

Hiện nay các động cơ đều có thông số về lực kéo moment với cánh tay đòn là 1cm, Đề tài sử dụng cánh tay đòn 5cm. Vì vậy thông số động cơ cần là:

|  |  |
| --- | --- |
| THÔNG SỐ ĐỘNG CƠ | |
| Moment lực kéo cho động cơ xoay dọc | 2\*5= 10 kg.cm |
| Moment lực kéo cho động cơ xoay ngang | 3\*5= 15 kg.cm |
| Điện áp làm việc | 5 - 6 V |
| Dòng | 1 A |

- Khi làm đề tài, 2 loại động cơ được cân nhắc sử dụng là động cơ servo và động cơ bước (STEP). Lý do là vì 2 loại động cơ này rất phổ biến trong các hệ thống máy phay, máy CNC, máy chuyên dụng. Sau đây là bảng so sánh.

|  |  |  |
| --- | --- | --- |
|  | Động cơ Servo | Động cơ bước |
| Số lượng điểm cực | 4-12 | 50-100 |
| Cách vận hành | Đọc dữ liệu từ các tín hiệu đầu vào, thông qua chiết áp mà xác định vị trí góc. | Di chuyển từng bước bằng cách cấp xung điện vào các cuộn dây. |
| Độ chính xác | Động cơ Servo nhờ sự phản hồi của encoder, điều khiển vòng kín có điều kiện nên có độ chính xác cao hơn, lấy ví dụ với encoder 1000 xung thì động cơ servo có thể chính xác tới 360/1000 = 0,36 độ . Do sử dụng bằng cách đếm xung để điều khiển vị trí nên gặp trường hợp bị trượt lực ( moto bị trượt ) sẽ vẫn có kết quả chính xác ở vị trí cần thiết | Motor bước khi bị trượt lực sẽ cho vị trí không chính xác ( do bị mất bước ) ( VD : cấp 200 xung để quay hết 1 vòng nhưng do tải quả nặng, bị kẹt nên cấp 200 xung mà vẫn chưa quay hết 1 vòng ). |

Đề tài yêu cầu độ chính xác về mặt điều hướng, cũng như về sức nặng của tải, vì vậy động cơ servo được ưu tiên sử dụng.

Tuy nhiên trong thực tế thông số của các động cơ servo ghi trên vỏ khá không chính xác. Trong quá trình thử nghiệm các loại động cơ. Đầu tiên đề tài thử nghiệm sử dụng động cơ RC FUTABA S3003 với các thông số như sau.

Động cơ RC FUTABA S3003

* Model: Futaba S3003
* Điện áp hoạt động 4.8 ~ 6.0VDC
* Xung yêu cầu: xung vuông điện áp đỉnh từ 3 ~ 5VDC
* Dòng điện tiêu thụ (4.8V): 50mA khi đứng yên, không tải.
* Dòng điện tiêu thụ (6.0V): 50mA khi đứng yên, không tải.
* Kiểu bánh răng: bánh răng nhựa.
* Tốc độ hoạt động (4.8V): 0.23s / 60 độ khi không tải.
* Tốc độ hoạt động (6.0V): 0.19s / 60 độ khi không tải
* Torque (4.8V-1A): 3.2kg.cm
* Torque (6.0V-1A): 4.1kg.cm
* Kích thước: 41 x 20 x 40.5mm
* Khối lượng: 38g.

Thực tế động cơ này không thể kéo được tải 2kg với cánh tay đòn 1cm. Hơn nữa kiểu bánh răng bằng nhựa gây nên cảm giác thiếu chắc chắn khi phải kéo tải nặng với cánh tay đòn dài hơn. Vì vậy ta nên lựa chọn loại động cơ servo với thông số gấp khoảng 1.3 lần giá trị ta cần đo (gợi ý của nhà sản xuất) với cách tay đòn 1cm và bánh răng kim loại.

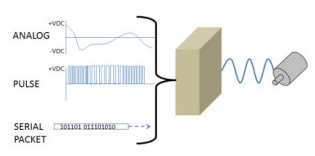
Hai loại động cơ servo hiện tại được sử dụng trong đề tài là:

|  |  |
| --- | --- |
| Tên | Thông số |
| Động cơ xoay dọc: servo LDX335MG | Điện áp làm việc: 6~7.2VDC (Kiểm nghiệm thực tế)  Dòng tiêu thụ không tải: 100mA, có tải >1500mA  Tốc độ: 0.16sec / 60 ° tại 7VDC  Lực kéo moment: 13Kg.cm 6VDC; 15KG.cm 6.5VDC; 17KG.cm 7VDC.  Trọng lượng: 60g  Kích thước: 40 x 20 x 40.5mm  Chiều dài cáp: 30cm. |
| Động cơ xoay ngang: servo LD20MG | Điện áp làm việc: 6~7.2VDC (Kiểm nghiệm thực tế)  Dòng tiêu thụ không tải: 100mA, có tải >1500mA  Tốc độ: 0.16sec / 60 ° tại 7VDC  Lực kéo moment: 20Kg.cm 6.6VDC - 2.5A.  Trọng lượng: 60g  Kích thước: 40 x 20 x 40.5mm  Chiều dài cáp: 30cm. |

3.2.7.4.1 Điều chỉnh góc động cơ và trục xoay

Thực tế khi động cơ servo kết nối với trục xoay, góc của động cơ và góc của tuốc năng là hoàn toàn khác nhau. Vậy yêu cầu đặt ra là làm sao để đưa góc của động cơ trở thành góc của tuốc năng. Để giải quyết vấn đề này, ta cần tham khảo kỹ hơn về tín hiệu điều khiển của động cơ servo và sự liên quan của phần cứng động cơ tới tín hiệu này.

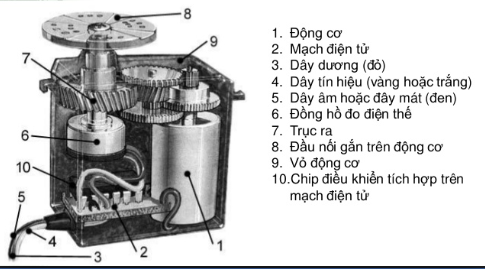
Tín hiệu điều khiển động cơ Servo



##### Hình 3.8 Tín hiệu điều khiển động cơ servo

Một động cơ servo được điều khiển bằng cách gửi một loạt các xung qua đường tín hiệu. Tần số của tín hiệu điều khiển phải là 50Hz hoặc một chu kỳ xung là 20ms. Độ rộng của xung xác định vị trí góc của servo và các loại servo này thường có thể xoay 180 độ (chúng có giới hạn vật lý khi di chuyển).

Trong động cơ servo, độ rộng xung được xác định bằng đồng hồ đo điện thế, ngày nay thường được thay bằng chiết áp hoặc biến trở. Chiết áp được gắn trên bánh răng cuối cùng hoặc trục đầu ra, do đó động cơ cũng quay chiết áp, tạo ra một điện áp liên quan đến góc tuyệt đối của trục đầu ra. Trong mạch điều khiển, điện áp chiết áp này được so sánh với điện áp đến từ đường tín hiệu. Nếu cần, bộ điều khiển kích hoạt mạch cầu H tích hợp cho phép động cơ quay theo hai hướng cho đến khi hai tín hiệu đạt mức chênh lệch bằng không. Cấu tạo của động cơ với đồng hồ đo điện áp như hình bên dưới.



##### Hình 3.9 Cấu tạo của động cơ servo

Vậy cách giải quyết của ta là tháo con chiết áp từ trong động cơ servo ra và gắn vào trục của bánh răng mà nối với tấm pin. Bằng cách đó, khi tấm pin quay, chiết áp cũng sẽ quay theo và mặc nhiên, góc của pin sẽ trở thành góc của servo.

3.2.7.4.2 Mạch cảm biến ánh sáng

Mạch cảm biến ánh sáng được thiết kế với yêu cầu xác định được hướng của ánh nắng mặt trời. Vì vậy mạch quang trở được thiết kế như sau:



##### Hình 3.10 Thiết kế mạch cảm biến ánh sáng

Bốn quang trở được để 4 góc, ở giữa là một vách ngăn hình chữ thập cao hơn chiều cao quang trở từ 1.5 - 2cm. Vách ngăn dùng nhựa dẻo, đế là bảng mạch, kích thước đế là 3cmx3cm. Đảm bảo độ nhẹ và bền chắc cho phần cứng này.

Dưới đây là hình schematic của khối này.

****

##### Hình 3.11 Schematic khối điều khiển Servo

3.2.7.4.2 Các chi tiết gia cố thêm

Vì phần khung sử dụng là nhôm nên việc hàn là bất khả thi, chỉ có bắt vịt từ cấu kiện này sang cấu kiện khác. Điều này không tránh khỏi việc phải gia cố thêm bằng các chi tiết phụ để đảm bảo độ bền chắc cho khung điều hướng. Các chi tiết cơ khi gia cố thêm trong luận văn là

- Ke: ke là vật liệu được sử dụng nhiều để định hình các kết cấu mà không thể hàn được, ví dụ như nhôm, gỗ. Đề tài sử dụng chủ yếu là ke tam giác (được dùng nhiều trong khung nhôm định hình) và ke chữ L (dài, ngắn), được dùng trong đóng khung gỗ, khung nhôm.



##### Hình 3.12 gia cố bằng ke

Để sử dụng ke, ta áp các mặt phẳng vào của ke vào 2 chi tiết cần cố định, sau đó bắt ốc vít vào các lỗ có sẵn. Hiện tại trên thị trường chỉ có loại ke chữ L và ke tam giác, trong đề tài có một vài thanh ke thằng là do đập thẳng lại từ ke chữ L ra.

- Khung sắt: Xét yếu tố ở dưới đế nên gia cố thêm sức nặng và độ chắc chắn để đảm bảo nâng đỡ được toàn bộ dàn khung phía trên. Đề tài thay khung nhôm ở giữa bằng một khung sắt có khả năng chịu lực tốt.



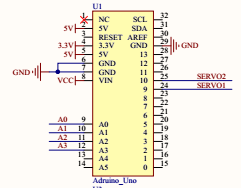
##### Hình 3.13 Gia công khung sắt

- Ổ bi: nhằm phụ trợ thêm cho quá trình quay của động cơ, giảm áp lực lên cánh tay đòn. Ổ bi là một giải pháp hữu hiệu dành cho các kết cấu trục quay cần độ linh hoạt cao.



##### Hình 3.14 Ổ bi

## 3.3 Khối Arduino Uno

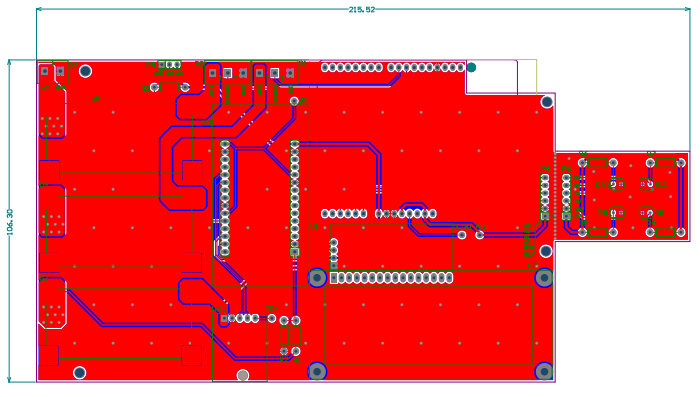
****

##### Hình 3.15 Khối Arduino Uno

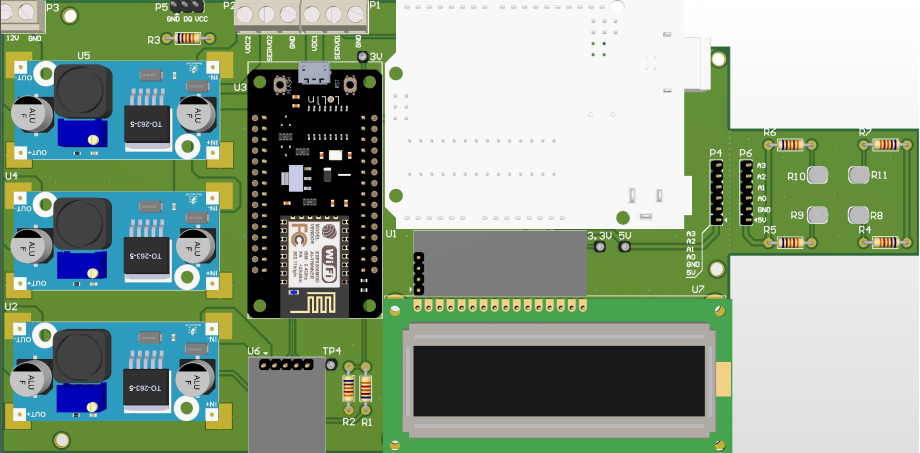
Arduino Uno nhận thông tin từ quang trở, tính toán và chuyển thành góc quay cho 2 động cơ servo.

## Thực hiện phần cứng trên phần mềm Altinum

Dưới đây là hình layout PCB và mô phỏng 3D board bằng phần mềm altium.



##### Hình 3.16 layout PCB

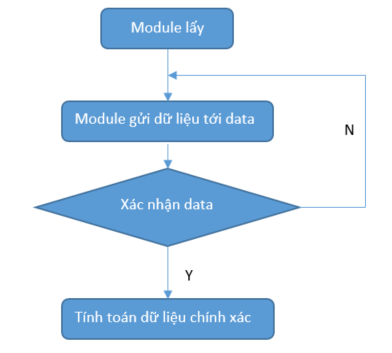


##### Hình 3.17 Hình 3D của board

# **THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM**

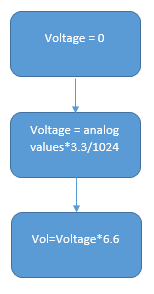
Chương này dùng để giới thiệu về phần mềm của đề tài, sơ đồ khối, cách hoạt động,...

## Giải thuật tổng quát



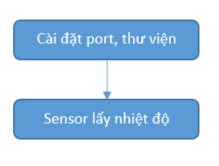
##### Hình 4.1 Sơ đồ khối giải thuật tổng quát

## Giải thuật tính toán áp DC

****

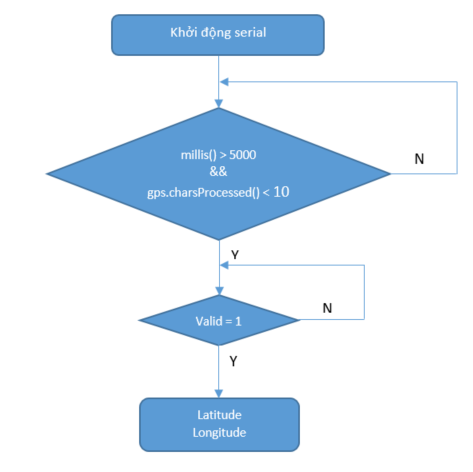
##### Hình 4.2 Sơ đồ khối giải thuật tính toán áp DC

## Giải thuật đo nhiệt độ



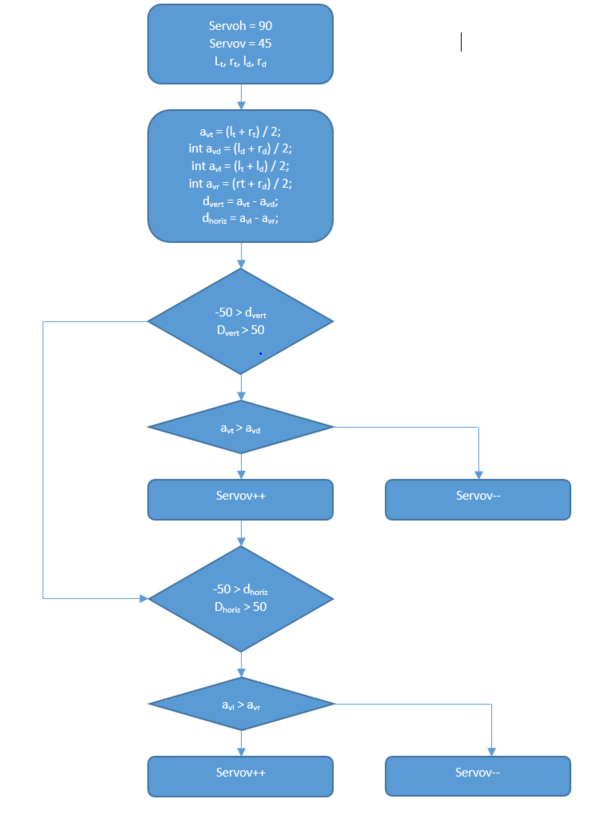
##### Hình 4.3 Sơ đồ khối giải thuật đo nhiệt độ

## Giải thuật GPS



##### Hình 4.4 Sơ đồ khối giải thuật GPS

## Giải thuật điều hướng pin mặt trời



##### Hình 4.5 Sơ đồ khối giải thuật điều hướng pin mặt trời

## Xây dựng Web Server và Website

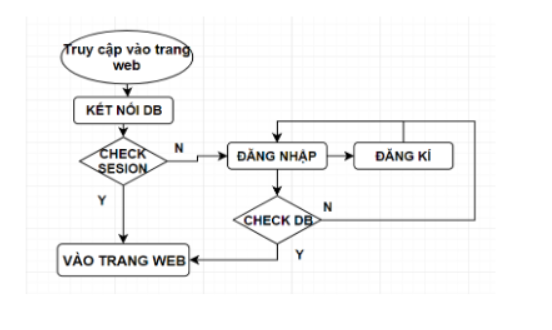
### 4.6.1 Tổng quát về ứng dụng Web

Để xây dựng 1 website thì ta cần phải có:

* Host
* Các file chứa mã nguồn (HTML, CSS, JS, …)
* Cơ sở dữ liệu (Database)

Khi truy cập tới tên miền, host sẽ chạy các file chứa mã nguồn tương ứng, các fiel này truy xuất vào cơ sở dữ liệu để tương tác và thực hiện những yêu cầu (request) từ người dung (user).

Để bảo mật thông tin, khi sử dụng giao diện Website thì người dung phải đăng nhập mới có thể xem các thông số kỹ thuật được. User Flow của người dùng như sau:



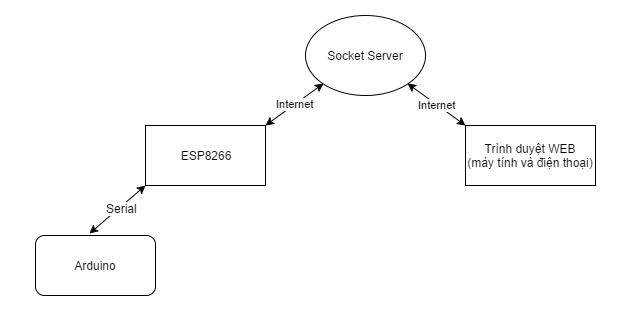
##### Hình 4.6 Sơ đồ khối user flow

Sau khi đăng nhập user có thể xem các thông số chi tiết của một trạm (ví dụ nhiệt độ, điện áp thu được, …). Bên cạnh đó cũng có thể kiểm soát vị trí của các trạm thông qua định vị GPS được gửi về từ mỗi trạm và thể hiện trên bản đồ để có thể biết được vị trí thực tế của các trạm.

Dữ liệu thực tế sẽ được lưu trên 1 database server riêng, tiện cho việc người user cần backup data về để sử dụng cho các mục đích khác.

### 4.6.2 Mô hình Socket Server và Socket Client trong Luận văn

Luận văn sẽ sử dụng NodeJS để lập trình cho một server được cài NodeJS nằm ngoài Internet. Máy chủ này sẽ tiếp nhận mọi yêu cầu từ Socket Client và sẽ là môi giới liên lạc giữa các Socket Client. Các Socket Client có thể là mạch ESP8266 hay là người dùng sử dụng trình duyệt web (trên điện thoại thông minh hoặc trên máy tính điện tử).



##### Hình 4.7 Mô hình Socket Server

Theo như mô hình trên:

* Socket Server nằm ở tầng cao nhất gọi là tầng Server.
* ESP8266 và Trình duyệt Web nằm ở tầng thứ 2 gọi là tầng Client
* Arduino ở tầng thử 3 gọi là Application.
* Mỗi tầng chỉ có thể liên lạc với tầng kế tiếp hoặc bên dưới nó.

Như vậy, chỉ cần xây dựng các phần mô hình giao tiếp ổn định giữa các tầng là chúng ta đã có ngay mô hình. Cụ thể, chúng ta sẽ xây dựng 2 phương thức giao tiếp giữa các tầng.

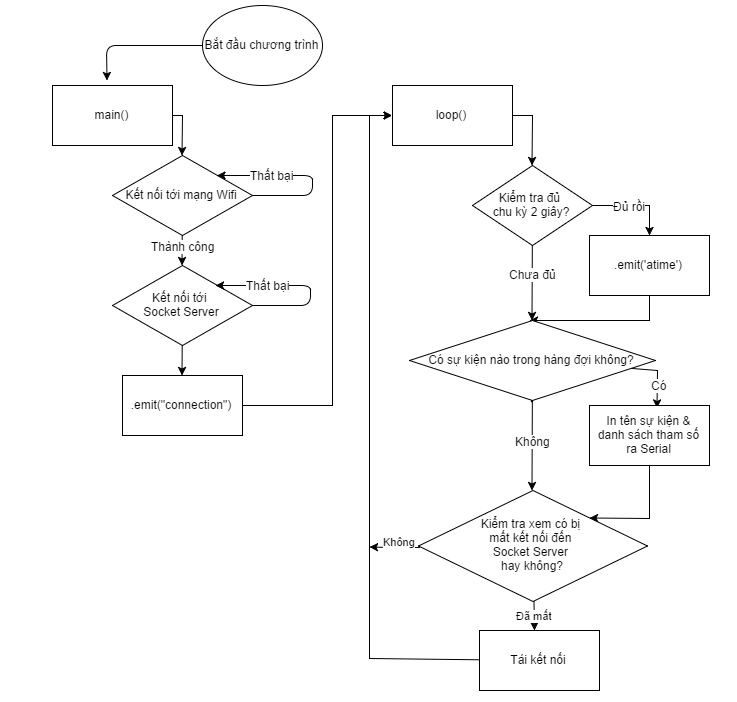
* Để giao tiếp giữa tầng 1 và tầng 2, chúng ta sử dụng kết nối Internet thông qua các gói dữ liệu được đóng gói theo chuẩn JSON.
* JSON là gì mà lại chọn nó? JSON là một kiểu dữ liệu đặc biệt, nó sẽ giúp ta đóng gói **tập hợp các dữ liệu**thành một chuỗi đơn giản theo quy chuẩn nhất định. Chuỗi này sẽ dễ dàng được vận chuyển thông qua môi trường Internet.
* Để giao tiếp giữa tầng 2 và tầng 3, chúng ta sử dụng cổng Serial truyền thống mà các bạn đã có thể làm quen thông qua việc làm xe điều khiển từ xa thông qua bluetooth.
* Nói chung Serial cũng đã là một giao tiếp chuẩn và tất cả các mạch ESP8266 và Arduino đều hỗ trợ ít nhất một cổng Serial trong hệ thống IO của mình. Trong dự án này, mình sẽ sử dụng Serial để việc giao tiếp giữa Arduino và ESP8266 trở nên dễ dàng.

### 4.6.3 Xử lý dữ liệu trên Arduino ESP8266

Nhiệm vụ của socket trên Arduino:

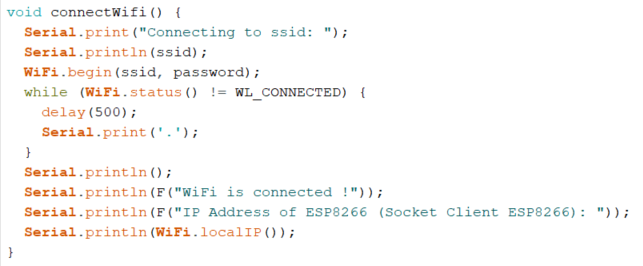
* Kết nối tới Socket Server.
* Nếu nó bị ngắt kết nối thì tự động kết nối lại.
* Cứ sau mỗi 2 giây sẽ gửi lệnh atime lên Socket Server để phá.
* Nếu nhận bất cứ lệnh gì từ socket server thì in ra console để xem.

Lưu đổ giải thuật:

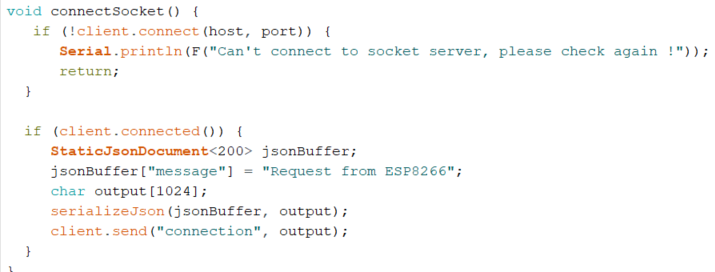


##### Hình 4.8 Lưu đồ giải thuật xử lý dữ liệu trên Node MCU ESP8266

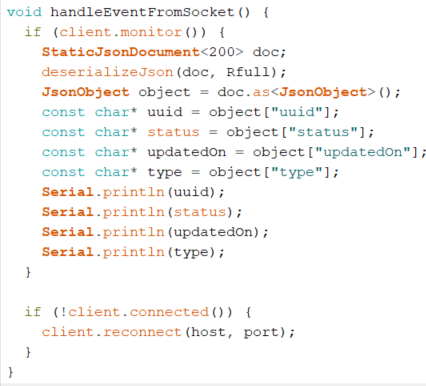
Tiếp theo việc đầu tiên cần làm là kết nối wifi cho ESP8266:



Trong đoạn code trên, chúng ta sẽ connect wifi cho ESP8266 và sẽ thông báo nếu kết nối thành công. Ngược lại sẽ delay khoảng 500 ms và tiếp tục thực hiện lại kết nói cho đến khi thành công.



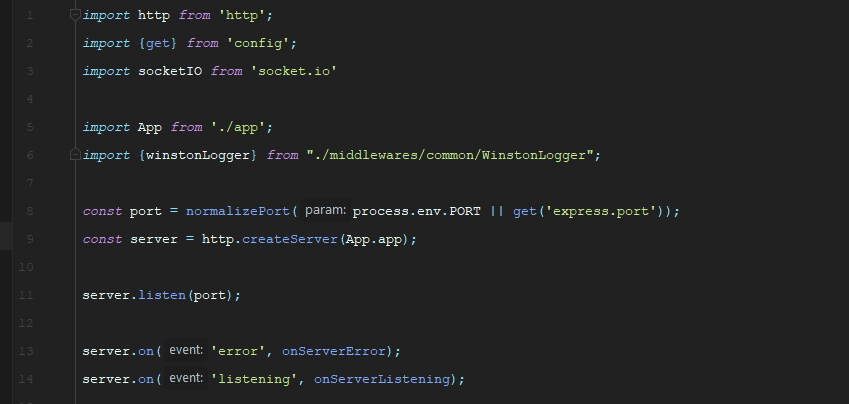
Tương tự sau khi kết nối wifi thành công, chúng ta tiếp tục connect đến socket phía server để thực hiện giao tiếp, gửi/nhận dữ liệu. Ở lệnh If đầu tiên sẽ kiểm tra xem socket có kết nối được không và nếu đã kết nói thì sẽ send 1 event về cho server để yêu cầu kết nối.



### 4.6.4 Xử lý dữ liệu phía Server

#### 4.6.4.1 Khởi tạo Server:

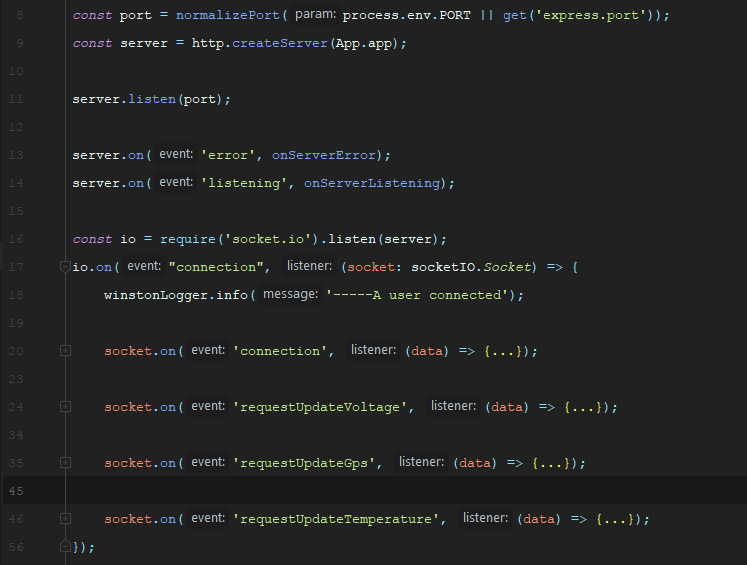
Đầu tiên, chúng ta sẽ khởi tạo 1 server đơn giản như sau:



Đây là đoạn code khởi tạo 1 server. Mặc định khi ở local, server sẽ chạy ở port 3000 được set up trong config. Khi deploy lên 1 server thật thì sẽ chạy trên port của server đó yêu cầu. 

Đoạn code trên sẽ theo dõi và in ra những thông báo để chúng ta có thể biết server đã được tạo thành công hay chưa và nếu có lỗi sẽ xảy ra lỗi gì.

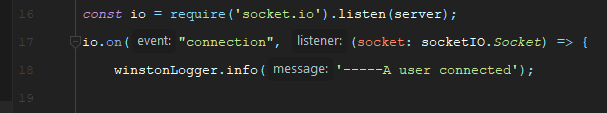
#### 4.6.4.2 Khởi tạo Socket Server:



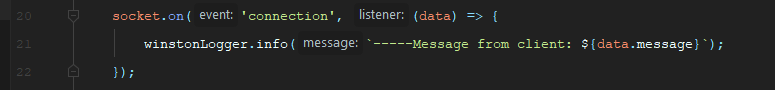
Chúng ta đã tạo một dịch vụ mạng (trong trường hợp này là socket server), dịch vụ này lắng nghe các kết nối tại địa chỉ 3000 (default port). Tất cả các chương trình mạng khác (socket client,...) muốn kết nối đến socket server thì phải biết được địa chỉ IP hoặc tên miền (trong trường hợp ra Internet) và địa chỉ port (cổng kết nối) của dịch vụ socket server.

Như vậy, nếu trong một địa chỉ IP mạng LAN nếu máy tính của chúng ta bị thay đổi IP (router hoặc modem wifi cấp DHCP lại một địa chỉ IP mới sau khi bị mất điện) thì các chương trình Socket Client **tin tưởng** vào địa chỉ IP trong mạng Local kia sẽ không hoạt động được vì không tìm thấy dịch vụ mạng socket server tại địa chỉ ip và port nói trên. Bạn có thể sét ip tĩnh cho máy server của bạn, tuy nhiên, đó chỉ là giải pháp tạm thời và không mang tính chuyên nghiệp. Ở phần cuối mình sẽ hướng dẫn bạn đưa dịch vụ socket server ra internet hoặc tạo một socket server ở một server ngoài Internet.

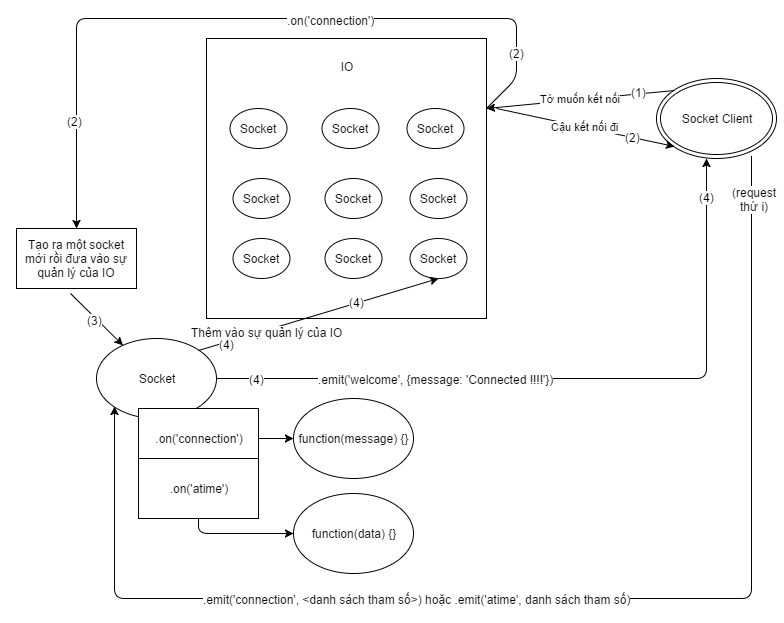
Trong đoạn code trên, line 9 sẽ khởi tạo server cho toàn bộ ứng dụng phía Server và ở line 16, Socket Server cũng sẽ lắng nghe server vừa được khởi tạo ở trên để handle những event từ client.



Ở đây là một API của thư viện socket.io (xem các dòng khai báo thư viện), sự kiện này sẽ xảy ra khi một dịch vụ socket client kết nối vào socket server! Ở đây, khi một socket client kết nối đến server, thì thư viện socket.io thông qua đại diện là biến io sẽ tạo ra một sự kiện tên là 'connection' với một biến là tên là socket. Biến socket này sẽ giúp chúng ta **giao tiếp**với các socket client từ chương trình socket server.



Ở các dòng bên trong sự kiện connection trên, chúng ta để ý rằng, socket có 2 hàm chính đó là "on" và "emit" . Trong đó, hàm on là hàm bắt sự kiện và hàm emit là hàm các sự kiện. Sơ đồ tổng quát giải thuật:

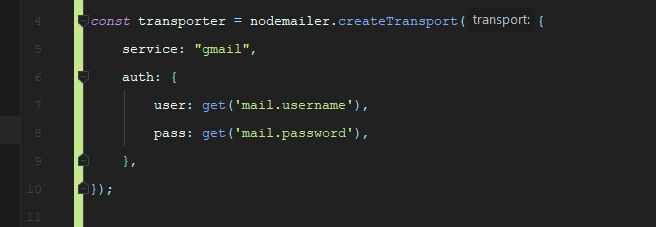


##### Hình 4.9 Sơ đồ tổng quát giải thuật Socket

Ở bước này, chúng ta đã có thể tạo một server hoàn chỉnh cho việc giao tiếp realtime giữa server và các thiết bị khác. Data sẽ được truyền/nhận liên tiếp giữa các thiết bị để tạo một hệ thống truyền nhận dữ liệu theo thời gian thực như đã đề ra ở yêu cầu của Đề tài. Công việc cuối cùng là viết một hàm làm nhiệm vụ tự gửi email để thông báo khi có sự cố xảy ra.

#### 4.6.4.3 Tạo chức năng gửi mail tự động:

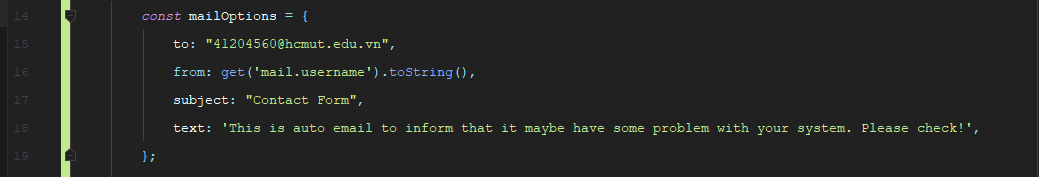
Để có thể gửi email, ta cần cấu hình 1 số thứ như sau:



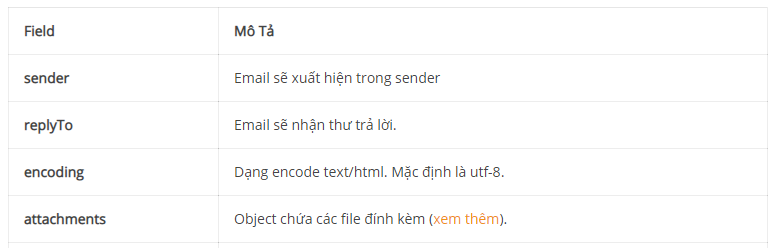
Với username và password là email của hệ thống. Để kiểm tra xem config và kết nối có thành công hay không chúng ta có cú pháp như sau:



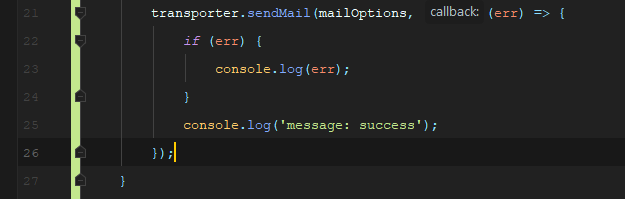
Sau khi đã kiểm tra kết nối thành công, chúng ta sẽ phải tiếp tục cấu hình cho email nhận với cú pháp như sau:



Ngoài ra, chúng ta cũng có thể cấu hình thêm các thông số sau vào email nhận thư:



Và công việc cuối cùng chỉ là gọi hàm sendMail từ library:



Tiếp đến, chúng ta sẽ xem xét các điều kiện khi nào có thể xảy ra sự cố để gửi email thông báo cho người quản lý hệ thống biết. Cụ thể trong đề tài, chúng em sẽ gửi email cảnh báo về sự cố khi điện áp thu được nhỏ hơn 2V:



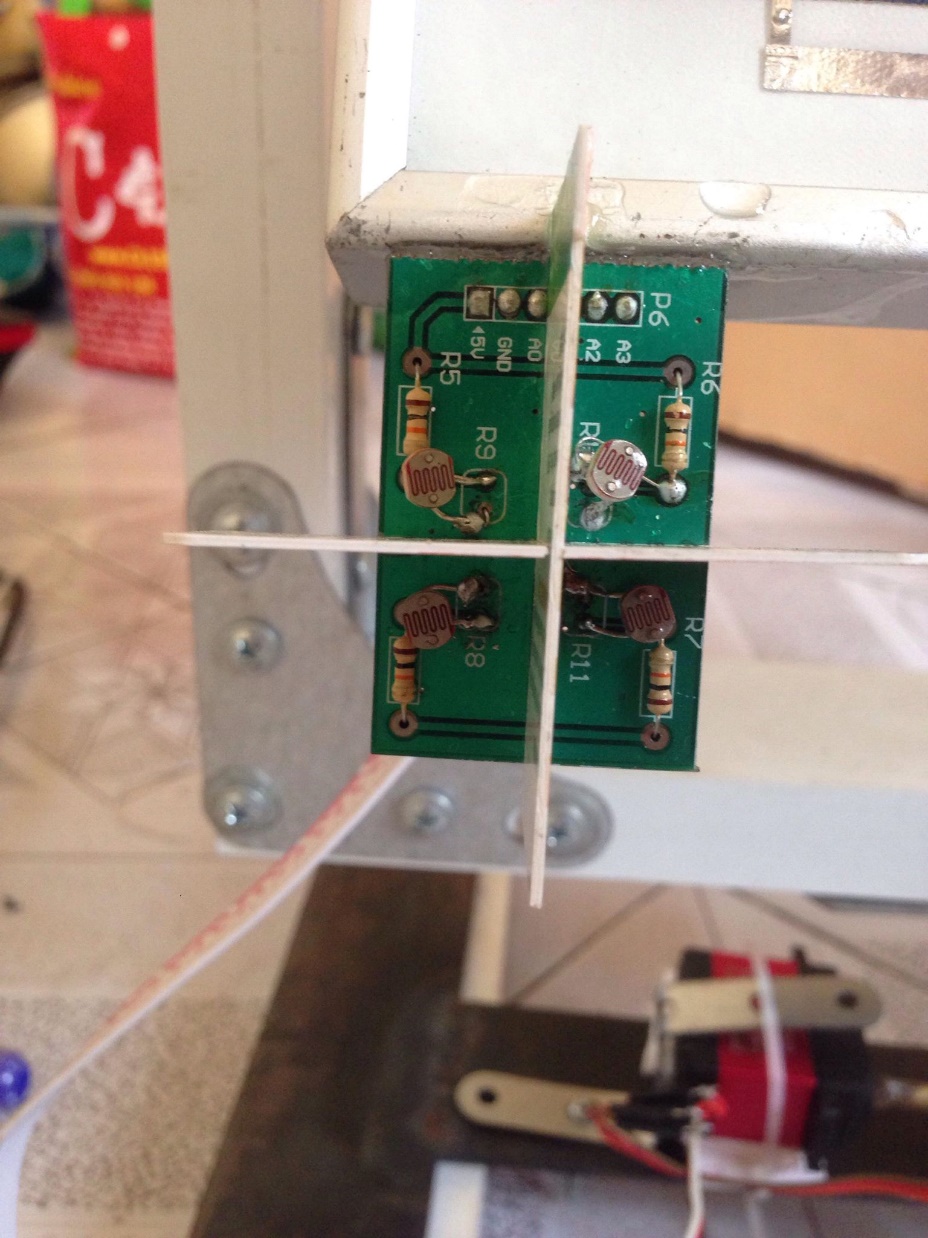
# **KẾT QUẢ THỰC HIỆN**

Chương này dùng để đưa ra những hình ảnh, số liệu thực tế từ đề tài.

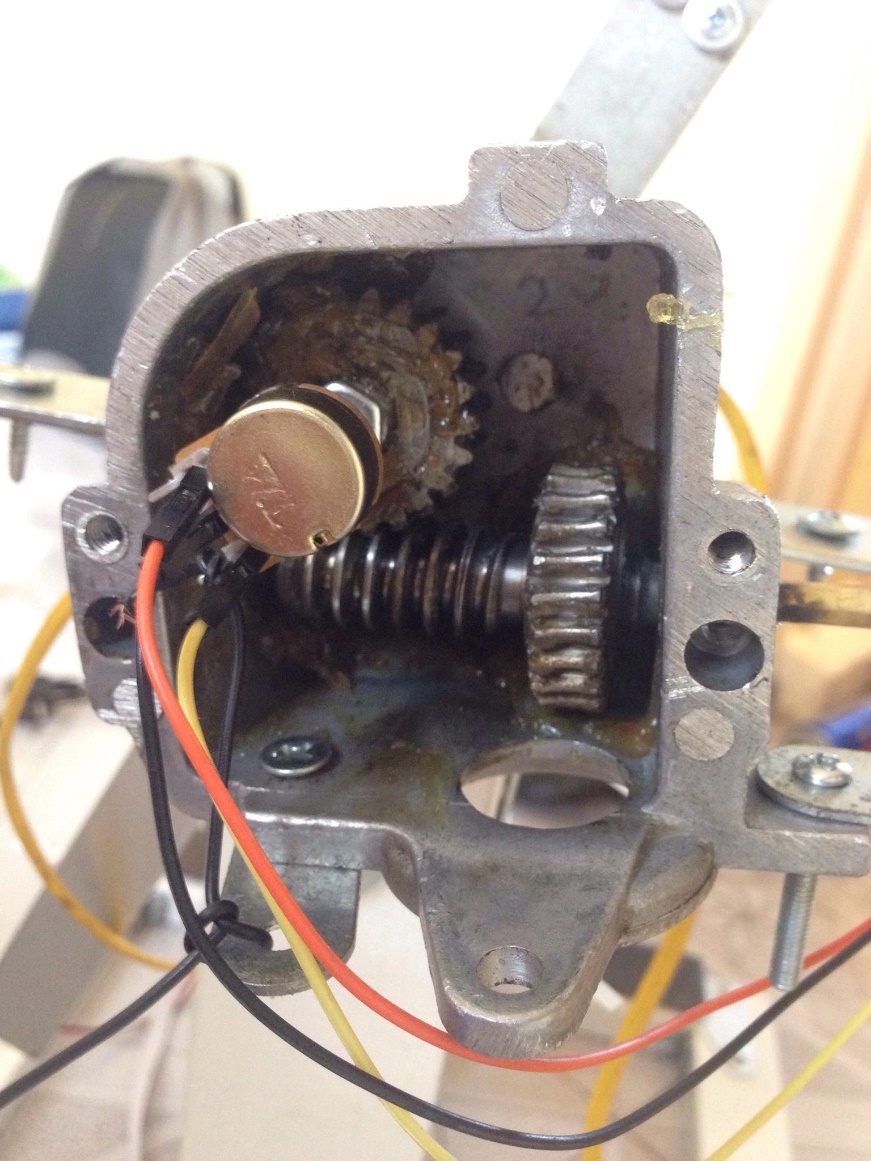
* 1. Kết quả thực hiện phần cứng



##### Hình 5.1 board thực tế



##### Hình 5.2 Mạch cảm biến ánh sáng



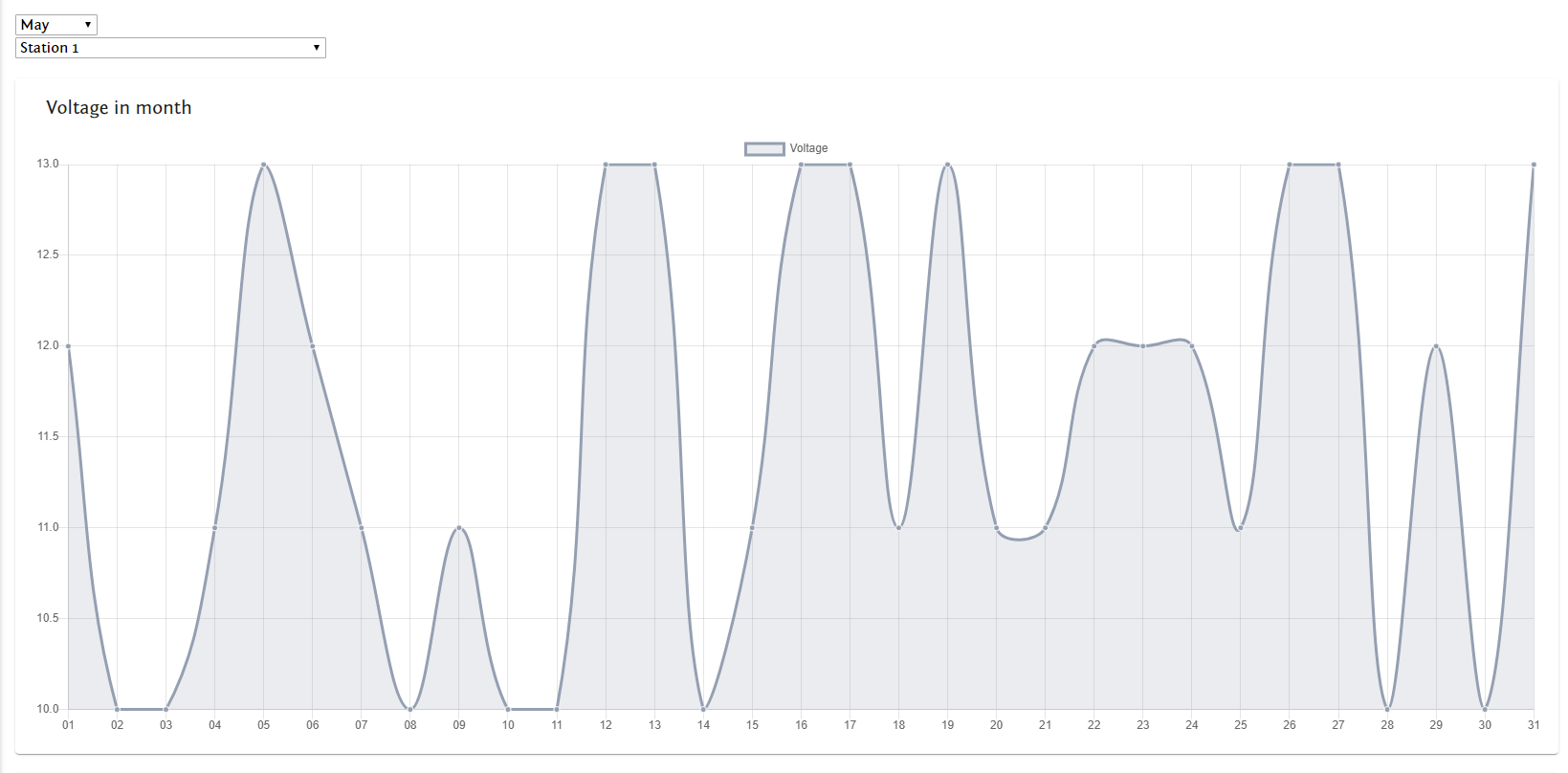
##### Hình 5.3 Kết cấu trục quay tuốc năng trong quạt công nghiệp



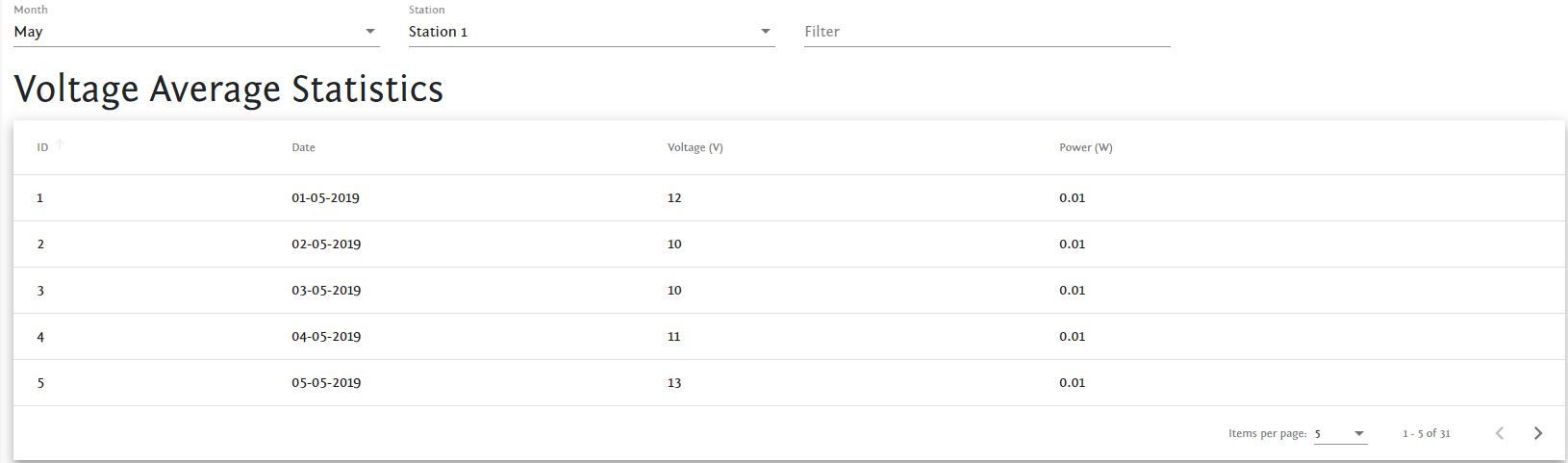
##### Hình 5.4 Mô hình thực tế

* 1. Kết quả thực hiện phần mềm

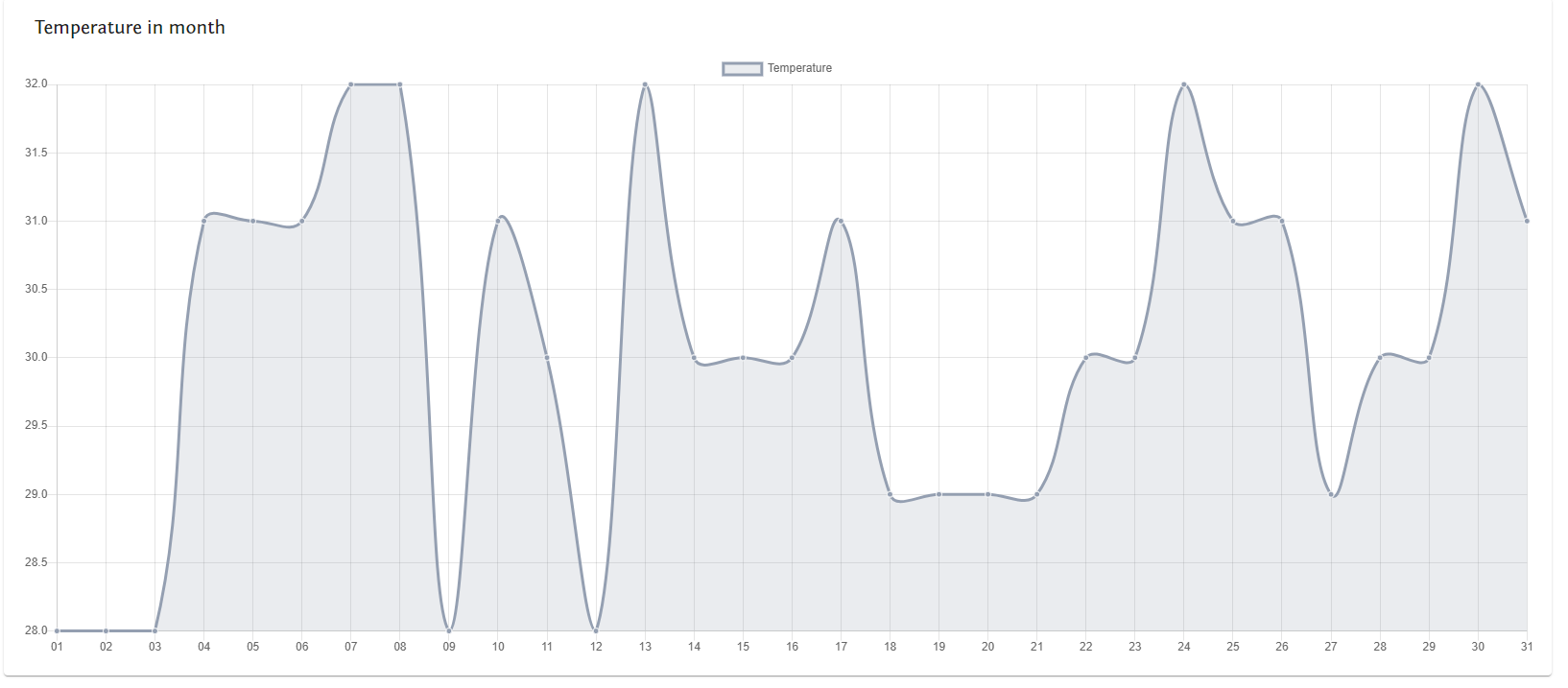
Số liệu server thu được:



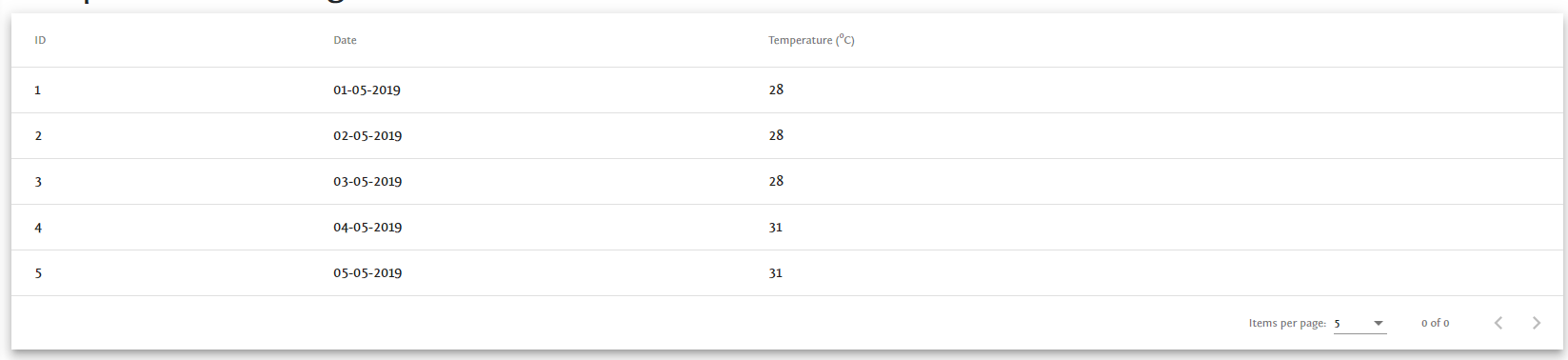
##### Hình 5.5 Đồ thị điện áp



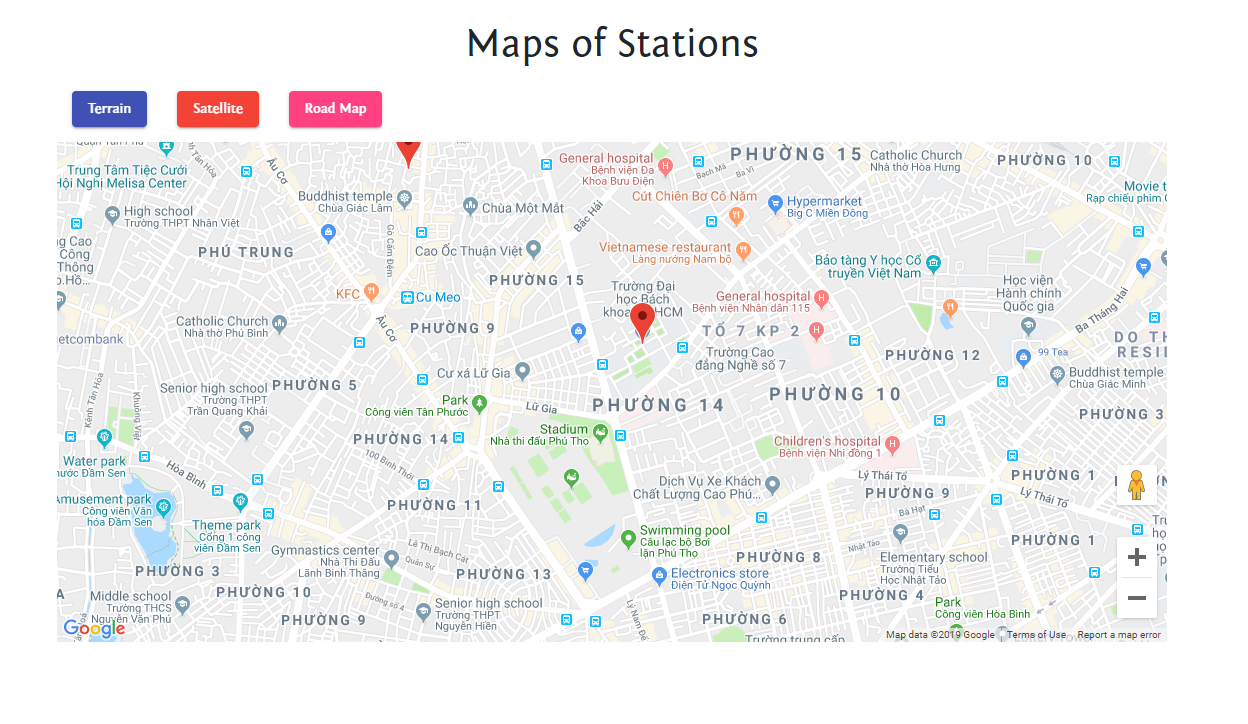
##### Hình 5.6 Số liệu điện áp



##### Hình 5.7 Đồ thị nhiệt độ



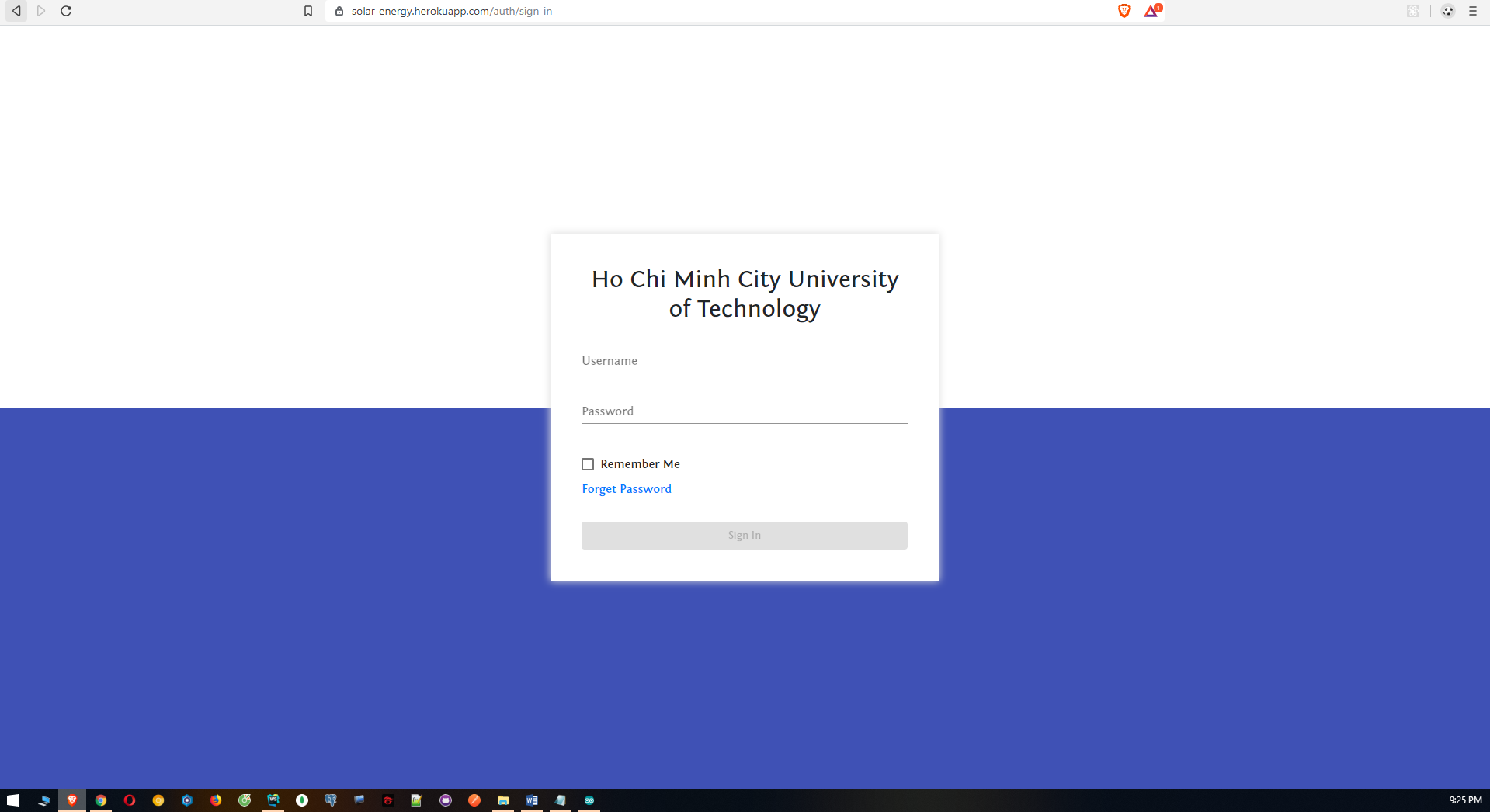
##### Hình 5.8 Số liệu nhiệt độ



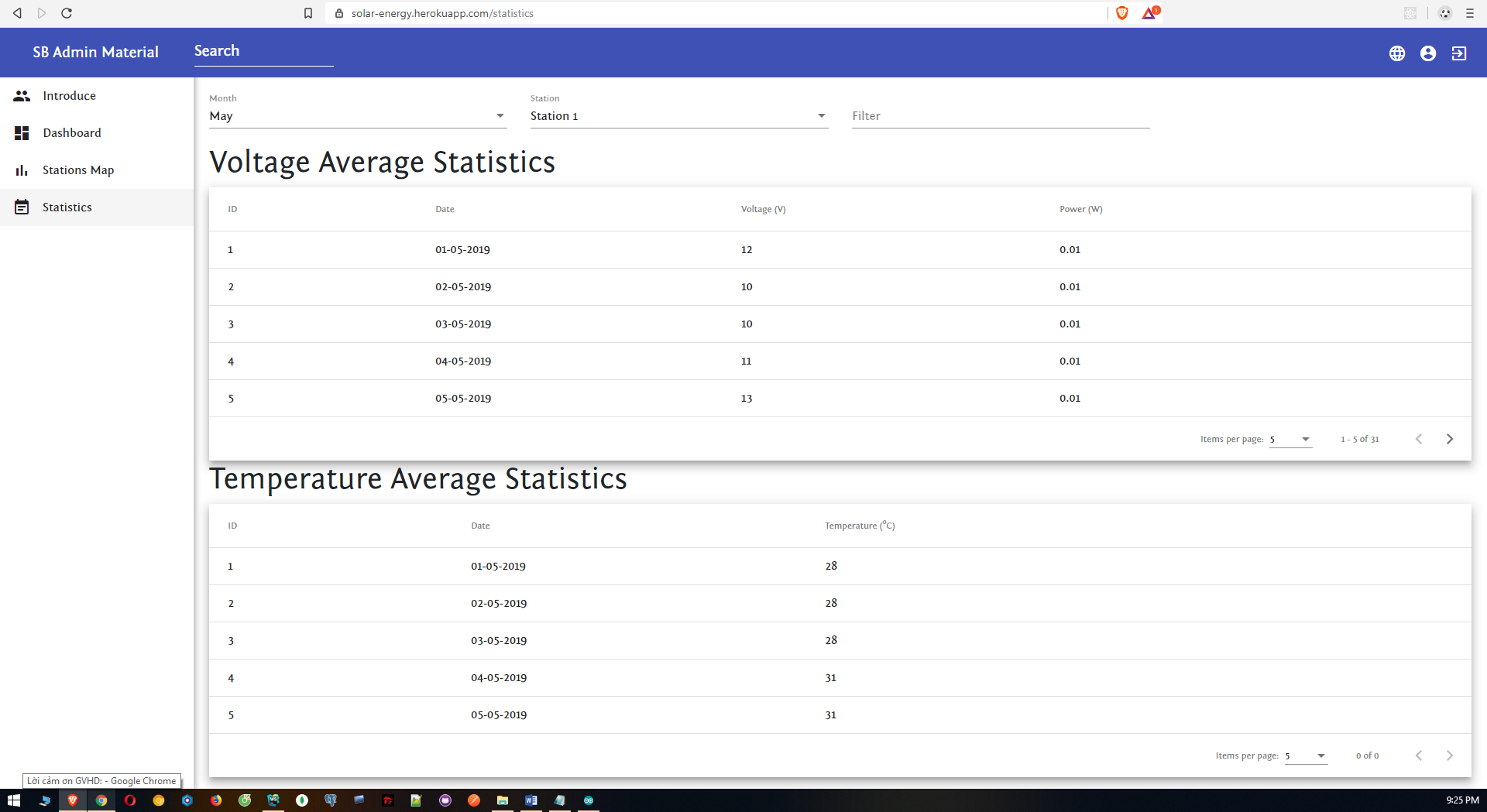
##### Hình 5.9 Sơ đồ (map) thể hiện các trạm

Giao diện ứng dụng web:

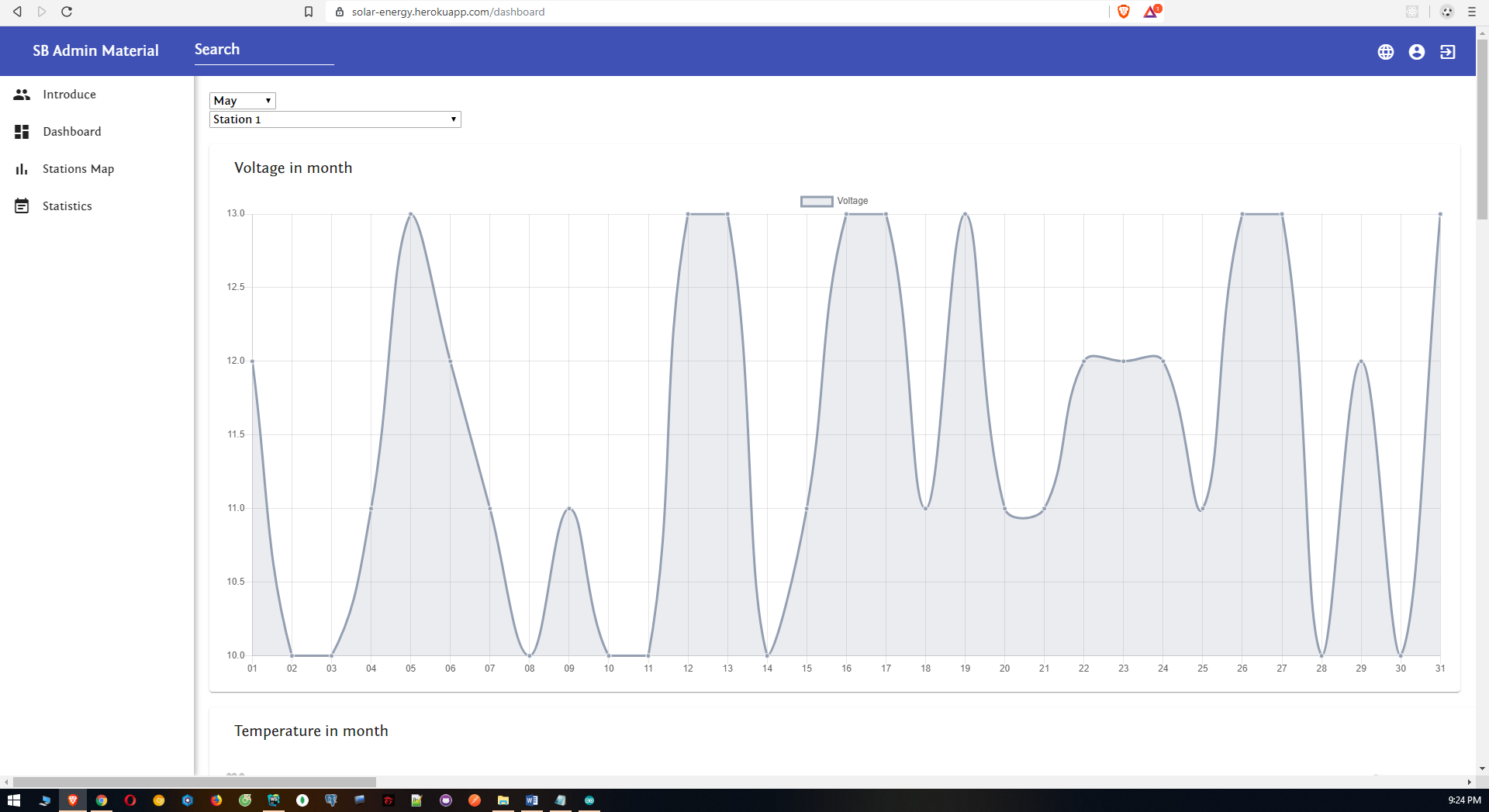
\* Trang đăng nhập, đăng ký:



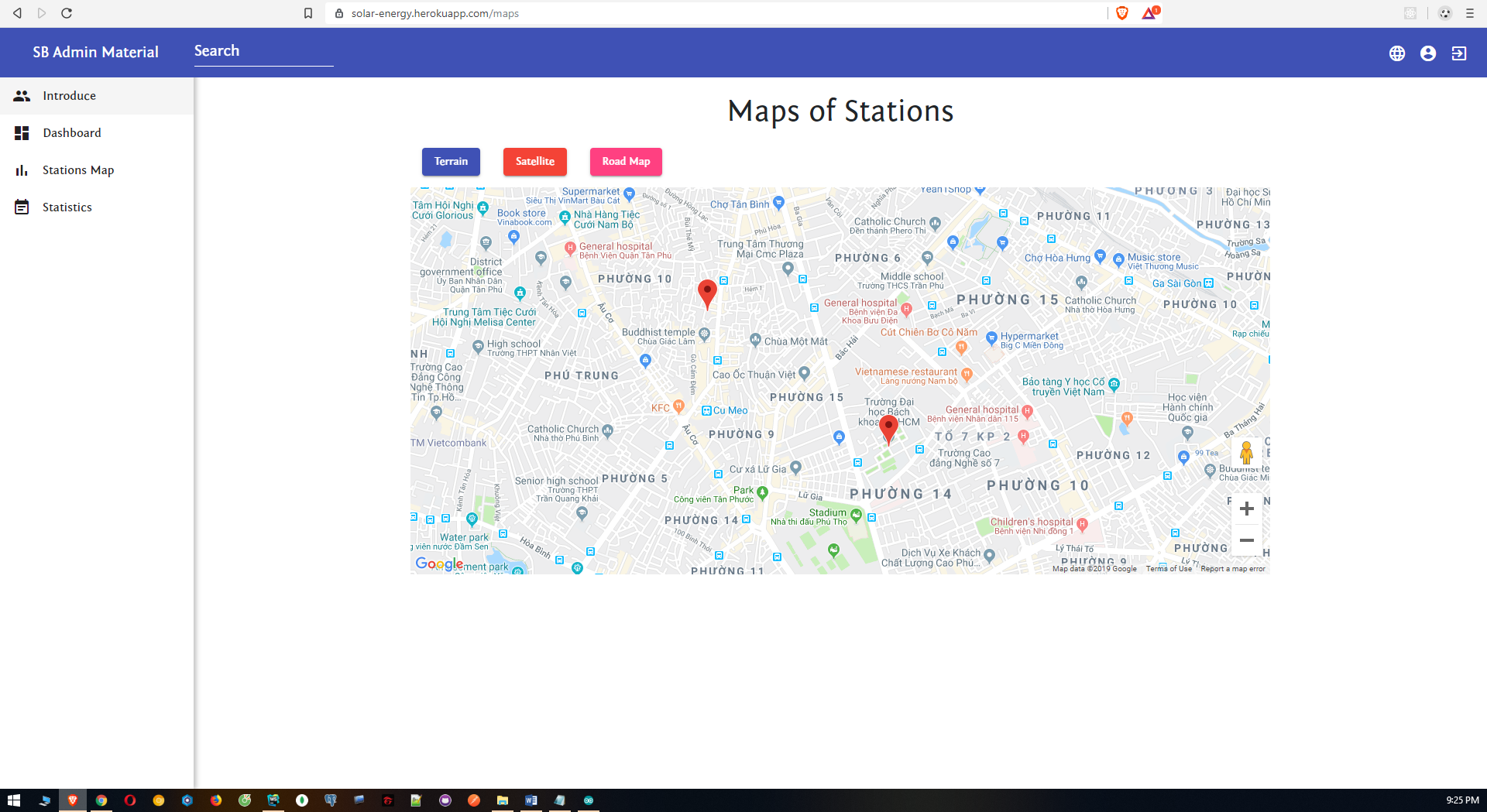
\* Trang thống kê số liệu:



\* Trang thống kê số liệu qua đồ thị:



\* Trang thể hiện vị trí các trạm trên bản đồ (Google Maps):



* 1. Đánh giá độ chính xác của hệ thống

Góc quay của ngang và góc quay dọc tối đa: 170 độ

Sai số điện áp: 0.2V

Sai số nhiệt độ: 0.3

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. Kết luận

Kết quả đạt được cơ bản các yêu cầu đặt ra ban đầu của luận văn:

* Hoàn thành mạch đo các thông số hệ thống (điện áp, nhiệt độ, tọa độ GPS) và truyền dữ liệu đến server.
* Thiết kế được server có database cập nhật dữ liệu liên tục, và website/app hiển thị trực quan cho người sử dụng dể dàng giám sát.
* Hệ thống điều hướng kép cho năng lượng mặt trời.

Nhờ đề tài luận văn này, bản thân em có cơ hội ôn tập lại và thực hành thực tế các kiến thức đã học, đặc biệt là các giao thức truyền thông mạng đã học trong môn Truyền số liệu và mạng, và ứng dụng của điện tử công suất trong việc thiết kế mạch đo thông số, tự bản thân tìm hiểu thêm lĩnh vực webserver và webservice, thực hành tìm hiểu và lắp đặt các kết cấu cơ khí phục vụ điều hướng.

Ngoài ưu điểm luận văn đã cơ bản đạt được những mục tiêu để ra, luận văn còn một số mặt hạn chế:

* Giám sát một chiều từ thiết bị gởi về server, chưa có từ server điều khiển ngược lại hệ thống => Chưa tích hợp được tính năng điều khiển Tấm pin mặt trời xoay từ hệ thống Website được.
* Hệ thống điều hướng sử dụng pin mặt trời nhỏ, công suất không cao, cân nặng chỉ tầm 2kg.
* Tính thẩm mỹ hệ thống chưa cao.
  1. Hướng phát triển
* Tích hợp thêm hệ thống pin mặt trời nối lưới.
* Thiết kế điều khiển ngược lại từ server tới hệ thống, định hướng xoay cho hệ thống từ giao diện website.
* Thực hiện điều hướng với hệ thống lớn hơn (tăng kích thước, số lượng tấm pin NLMT), tăng sức mạnh của động cơ cũng như độ linh động của hệ thốn

# **TÀI LIỆU THAM KHẢO**

1. <https://hshop.vn/products/kit-rf-thu-phat-wifi-esp8266-nodemcu><https://solarenergypanels.in/solar-power-plants>
2. <http://arduino.vn/bai-viet/1257-gps-va-arduino-phan-2-lay-du-lieu-tu-ve-tinh-gps><https://arduino.esp8266.vn/>
3. <https://arduino.esp8266.vn/>
4. <https://esp8266.vn/>
5. <https://iotmaker.vn/esp8266-iot-wifi-uno.html>
6. <https://www.youtube.com/watch?v=cbEQFaViI6U>
7. <https://iotmaker.vn/cam-bien-nhiet-do-probe-ds18b20.html>
8. <https://hshop.vn/collections/dong-co-rc-servo>
9. <http://daotaodientu.com/cau-phan-ap-mach-chia-ap/>
10. <https://iotmaker.vn/mach-chuyen-doi-i2c-cho-lcd.html>
11. <https://www.youtube.com/watch?v=_6QIutZfsFs>
12. <http://arduino.vn/bai-viet/1496-esp8266-ket-noi-internet-phan-1-cai-dat-esp8266-lam-mot-socket-client-ket-noi-toi>
13. <http://arduino.vn/bai-viet/1497-esp8266-ket-noi-internet-phan-2-arduino-gap-esp8266-hai-dua-noi-chuyen-bang-json>
14. <http://arduino.vn/bai-viet/1511-esp8266-ket-noi-internet-phan-3-arduino-gap-smartphone-hai-dua-noi-chuyen-bang-json>

# **PHỤ LỤC**

Source của luận văn đã được upload trên open source <https://github.com/Jorkendy/solar-energy>