

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN CUNG CẤP ĐIỆN  
-----o0o-----



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**XÂY DỰNG HỆ THỐNG ĐO ĐẠC VÀ  
GIÁM SÁT NĂNG LƯỢNG TRONG HỆ THỐNG  
NĂNG LƯỢNG MẶT TRỜI**

**SVTH: NGUYỄN NGỌC THẮNG**

**MSSV: 1413669**

**GVHD: PGS. TS NGUYỄN ĐÌNH TUYÊN**

**TP. HỒ CHÍ MINH, 2018**

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: **Cung Cấp Điện**

## NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP

1. HỌ VÀ TÊN: NGUYỄN NGỌC THẮNG

MSSV: 1413669

2. NGÀNH: **KỸ THUẬT ĐIỆN – ĐIỆN TỬ**

LỚP : DD14KSKT

3. Đề tài:

4. 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: .....

7. Họ và tên người hướng dẫn:

Phản 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***

Chợt nhìn lại thì thời gian ở Bách Khoa trôi nhanh như một cơn mưa rào, mới ngày nào em còn là một cậu nhóc miền Trung một mình vào Sài Gòn nhập học, ngỡ ngác với mọi thứ về nơi phố thị này. Rồi cũng xong một năm học đại cương thật sự mông lung ở cơ sở hai, em được chuyển về học tập ở cơ sở một. Và chắc với em mọi thứ đời sinh viên bắt đầu tập nập hơn khi đặt chân vào trung tâm thành phố này. Bách Khoa cho em quá nhiều thứ: thầy cô, bạn bè, tri thức chuyên ngành cả kinh nghiệm sống... Để giờ lúc này khi hoàn thành xong cuốn luận văn này thì ngoài sự háo hức, cũng thật sự là tiếc nuối khi phải sắp phải bước chân ra khỏi quãng đời sinh viên.

Em xin chân thành cảm ơn tất cả quý thầy cô đã dạy em từ những ngày đầu còn là sinh viên nhất, cảm ơn các thầy cô ở bộ môn Cung Cấp Điện, ở ngành Kỹ Thuật Điện, Khoa Điện – Điện Tử đã truyền đạt cho em nhiều kiến thức để thực hiện luận văn này và cho em 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 gửi lời cảm ơn, tri ân chân thành nhất đến Thầy Nguyễn Chấn Việt, Thầy Nguyễn Đình Tuyên đã giao nhiệm vụ và hướng dẫn em tận tình trong đề tài. Cảm ơn bạn Lê Viết Hưng cũng đã cùng hợp tác để đi đến kết quả này. Do thời gian và kiến thức còn hạn chế nên luận văn sẽ không tránh những thiếu sót bất cập, 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 năng lượng mặt trời. Sử dụng (stm32+esp8266) để đo đạc, tính toán các thông số của hệ thống, và truyền dữ liệu về server. Dùng raspberry để làm webserver. Thiết kế web và app 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 vi điều khiển STM32F103C8T6, SoC esp8266, máy tính nhúng raspberry,... các vấn đề về chuẩn giao thức http, webserver và webservice.
3. Thiết kế và thực hiện phần cứng
4. Thiết kế và thực hiện phần mềm: Sử dụng phần mềm Keil uVision5 để viết code và debug trên kit STM32F103, phần mềm Xampp để tạo Localhost trên win, LAMP để chạy máy chủ web trên raspberry, và Sublime Text để viết code phần web( ngôn ngữ PHP, HTML, CSS, JAVASCRIPT)
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 .....	9
1.1 Tổng quan đề tài.....	9
1.2 Tình hình nguyên cứu trong và ngoài nước .....	9
1.2.1 Tình hình nguyên cứu ngoài nước:.....	9
1.2.2 Tình hình nguyên cứu trong nước: .....	10
1.3 Nhiệm vụ luận văn .....	10
2. LÝ THUYẾT .....	12
2.1 Tổng quan thiết kế.....	12
2.2 Tổng quan về năng lượng mặt trời .....	13
2.2.1 Ưu, nhược điểm của năng lượng mặt trời .....	13
2.2.2 Một số hệ thống năng lượng mặt trời .....	14
2.2.2.1 Off-Gird Solar PV System .....	14
2.2.2.2 On-Gird Solar PV System .....	16
2.2.2.3 Hybird Solar PV System .....	16
2.3 Giới thiệu vi điều khiển STM32F103C8T6 .....	17
2.3.1 Giới thiệu chung.....	17
2.3.2 Sơ lược các chức năng chính sử dụng trong luận văn .....	19
2.3.2.1 GPIO .....	19
2.3.2.2 USART.....	21
2.3.2.3 ADC .....	22
2.4 Giới thiệu SoC ESP8266.....	23
2.4.1 Giới thiệu.....	23
2.4.2 Các loại Module ESP8266 .....	27
2.4.2.1 ESP-WROOM-02: .....	27
2.4.2.2 ESP-01: .....	28
2.4.2.3 ESP-12E: .....	29
2.4.3 Kết nối với ESP8266.....	29

2.4.3.1	Giao tiếp UART .....	29
2.4.3.2	Kết nối Wifi.....	30
2.4.4	Lập trình cho ESP8266 .....	31
2.4.4.1	Lập trình trực tiếp.....	32
2.4.4.2	Sử dụng firmware thư viện từ cộng đồng phát triển .....	34
2.5	Máy tính nhúng Raspberry.....	37
2.5.1	Giới thiệu chung.....	37
2.5.2	Hệ điều hành và phần mềm .....	38
2.5.2.1	Hệ điều hành .....	38
2.5.2.2	Phần mềm.....	39
2.5.3	Một số lệnh cơ bản trên Raspberry.....	42
2.5.3.1	Các câu lệnh cơ bản .....	43
2.5.3.2	Các câu lệnh liên quan đến file, folder.....	43
2.5.3.3	Các câu lệnh liên quan đến mạng.....	44
2.5.3.4	Các lệnh liên quan đến hệ thống .....	45
2.6	Giới thiệu về Web .....	45
2.6.1	Cơ sở dữ liệu và Webserver .....	45
2.6.1.1	Cơ sở dữ liệu .....	45
2.6.1.2	Webserver .....	47
2.6.2	Lập trình Web.....	49
2.6.2.1	HTML .....	49
2.6.2.2	CSS.....	52
2.6.2.3	PHP .....	52
2.6.3	Giao thức HTTP.....	53
3.	THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG .....	55
3.1	Sơ đồ khối tổng quát hệ thống .....	55
3.2	Sơ đồ mạch chi tiết.....	57
3.2.1	khối CT.....	57

3.2.2 khối VT .....	58
3.2.3 khối MCU.....	60
3.2.4 Khối ESP8266.....	61
3.2.5 khối POWER.....	62
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM .....	62
4.1 Giải thuật vi điều khiển.....	62
4.1.1 Giải thuật tổng quát.....	62
4.1.2 Giải thuật measure_data .....	63
4.1.2.1 Giải thuật detect zero_crossing .....	64
4.1.2.2 Giải thuật tính toán RMS: .....	64
4.2 Giải thuật ESP8266 .....	65
4.3 Xây dựng Webserver và Website.....	65
4.3.1 Xamp trên laptop/pc .....	66
4.3.2 Xây dựng web .....	69
4.3.2.1 User .....	69
4.3.2.2 Get dữ liệu.....	70
4.3.2.3 Hiển thị Chart.....	71
5. KẾT QUẢ THỰC HIỆN.....	71
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	72
6.1 Kết luận.....	72
6.2 Hướng phát triển .....	72
7. TÀI LIỆU THAM KHẢO.....	73
8. PHỤ LỤC.....	74
8.1 Chương trình ở vi điều khiển: .....	74



## DANH SÁCH HÌNH MINH HỌA

Hình 2.1 Mô hình hệ thống năng lượng mặt trời.....	12
Hình 2.2 Mô hình hệ thống giám sát năng lượng mặt trời .....	13
Hình 2.3 Mô hình hệ thống năng lượng mặt trời Off-Gird.....	14
Hình 2.4 Mô hình hệ thống năng lượng mặt trời On-Gird .....	16
Hình 2.5 Mô hình hệ thống năng lượng mặt trời Hybird .....	16
Hình 2.6 Kit STM32F103C8T6 .....	18
Hình 2.7 Sơ đồ chân của vi điều khiển STM32F103C8T6 .....	19
Hình 2.8 Sơ đồ khối ADC .....	23
Hình 2.9 Chip ESP8266 .....	24
Hình 2.10 Sơ đồ khối chip ESP8266.....	26
Hình 2.11 ESP-WROOM-02.....	27
Hình 2.12 ESP-01 .....	28
Hình 2.13 ESP-12E .....	29
Hình 2.14 UART trên ESP8266 .....	30
Hình 2.15 Mô hình mạng Wifi .....	30
Hình 2.16 Bộ thư viện ESP8266 NONOS_SDK.....	33
Hình 2.17 Mô hình sử dụng IDE của Arduino lập trình cho ESP8266 .....	34
Hình 2.18 Máy tính Raspberry pi 2 .....	37
Hình 2.19 Remote raspberry bằng SSH thông qua phần mềm PUTTY .....	40
Hình 2.20 SSH raspberry bằng cmd trên win 10.....	40
Hình 2.21 Remote Desktop Connection .....	41
Hình 2.22 Màn hình đăng nhập .....	42
Hình 2.23 Giao diện GUI .....	42
Hình 2.24 Ví dụ về bảng cơ sở dữ liệu .....	46
Hình 2.25 Tiến trình truyền tải web .....	48
Hình 2.26 Giao diện khi chạy thẻ HTML.....	51
Hình 2.27 Mô hình Client – Server web.....	53
Hình 3.1 Sơ đồ khối hệ thống phần cứng .....	56
Hình 3.2 Schematic khối CT .....	57
Hình 3.3 Nguyên lí khối CT .....	57
Hình 3.4 Schematic khối VT .....	58

Hình 3.5 Nguyên lí khối VT .....	59
Hình 3.6 Schematic khối ESP8266.....	61
Hình 3.7 Schematic khối Power .....	62
Hình 4.1 Giải thuật tổng quát của vi điều khiển.....	62
Hình 4.2 Giải thuật measure_data .....	63
Hình 4.3 Giải thuật detect zero_crossing.....	64
Hình 4.4 Giải thuật chương trình ESP8266.....	65
Hình 4.5 Trang chủ của web tải phần mềm localhost-Xampp .....	67
Hình 4.6 Bảng điều khiển khi khởi động Xampp.....	67
Hình 4.7 Giao diện Xampp khi khởi động hoàn tất các ứng dụng Apache và MySQL .....	68
Hình 4.8 Giao diện của Editor Sublime Text .....	68
Hình 4.9 Giao diện trang đăng nhập.....	69
Hình 4.10 Thông tin Account của 1 user.....	70
Hình 4.11 Chart hiển thị thông số hệ thống.....	71

## 1. GIỚI THIỆU

### 1.1 Tổng quan đề tài

Ngày nay, sự phát triển mạnh mẽ của công nghệ thông tin đã làm thay đổi diện mạo của tất cả lĩnh vực đời sống. Trong những năm gần đây khái niệm Internet of Thing (IoT) tạm dịch là “Vạn vật kết nối” được nhắc đến rất nhiều, ngày càng nhiều các thiết bị, phần mềm giúp con người theo dõi, tương tác điều khiển từ xa các hệ thống thiết bị thông qua mạng internet

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 được hệ thống năng lượng mặt trời 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, em quyết định ứng dụng Internet of Things để thực hiện đề tài: **“Xây dựng hệ thống đo đạc và giám sát năng lượng trong hệ thống năng lượng mặt trời”**.

### 1.2 Tình hình nguyên cứu trong và ngoài nước

#### 1.2.1 Tình hình nguyên cứu ngoài nước:

Thế giới 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 tivi, đầu DVR, thiết bị đeo. 13 tỉ còn lại là điện thoại di động, máy tính xách tay 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 Things đang là chìa khóa của thành công trong tương lai.

### 1.2.2 Tình hình nguyên cứu trong nước:

Internet of Things không phải khái niệm xa lạ đối với nước ta những năm gần đây. Tuy nhiên, giải pháp này vẫn chưa thực sự phổ biến tại Việt Nam và vẫn được xem là cao cấp mỗi khi nhắc đến. Ứng dụng IoT vào phục vụ đời sống con người hiện đang được nghiên cứu mà mọi người thường biết là những công tơ điện tử smart để thu thập dữ liệu từ xa của điện lực, điều khiển đèn đường thông minh, hay những tủ điện GPRS, ... Trong hệ thống năng lượng mặt trời hiện nay thường rất ít những hệ thống nối lưới bởi vì nước ta chưa phổ biến những hệ thống giám sát năng lượng mặt trời tốt, đáng tin cậy. Vì vậy việc phát triển một hệ thống giám sát năng lượng mặt trời, ứng dụng IoT có khả năng đáp ứng được nhu cầu mà mọi người đang quan tâm.

### 1.3 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 theo dõi các thông số của hệ thống năng lượng mặt trời. Cụ thể là xây dựng được hệ thống đo các thông số của hệ thống như: I, U, P, ... Xây dựng được một website/App với cơ sở dữ liệu để lưu trữ các thông số thu thập được từ hệ thống đo đặc thông qua Internet. Người vận hành có thể theo dõi được các thông số của hệ thống này từ các thiết bị có thể kết nối mạng internet như Laptop, PC, điện thoại di động trên một giao diện website/app... Do điều kiện thực tế, luận văn chỉ thiết kế mô phỏng hệ thống giám sát năng lượng mặt trời Hybrid, với lưới điện 1 pha, công suất của pin mặt trời dưới 1kW.

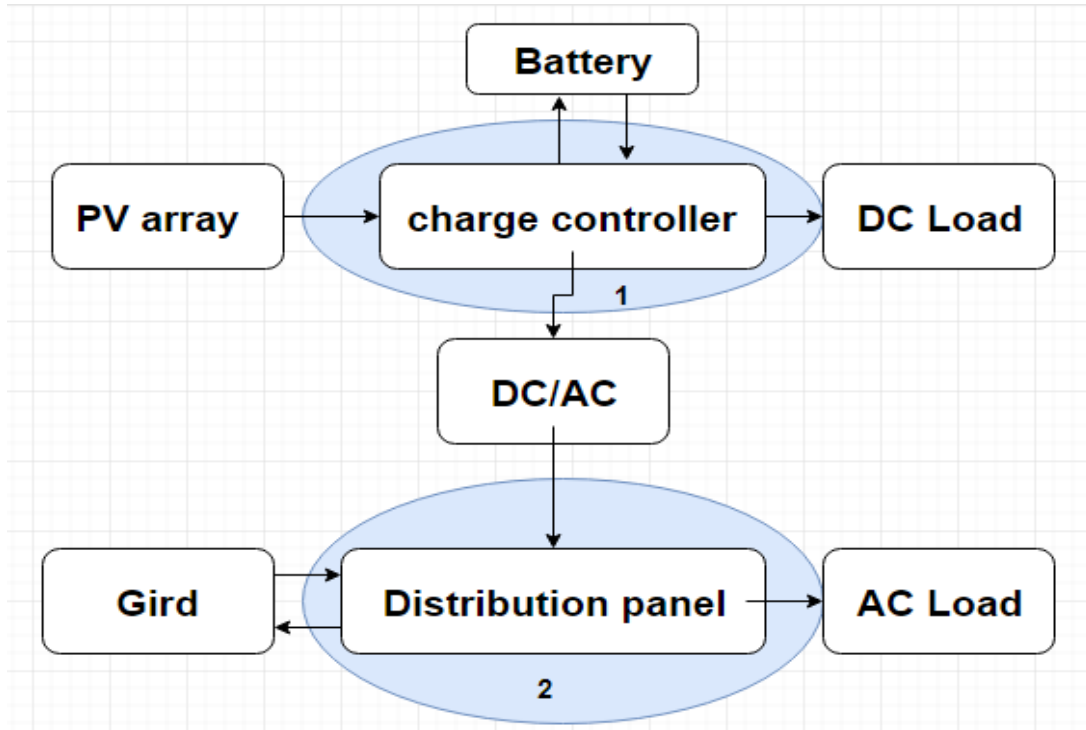
Luận văn chủ yếu tập trung nghiên cứu các phần sau:

- Tìm hiểu hệ thống mặt trời
- Đo đạc các thông số của hệ thống

- + Nghiên cứu sử dụng vi điều khiển STM32F103C8T6
  - + Nghiên cứu mạch đo dòng điện, điện áp để từ đó tính ra công suất
  - Truyền và hiển thị thông số trên web
    - + Nghiên cứu sử dụng ESP8266 để truyền nhận data đến server thông qua wifi
    - + Nguyên cứu sử dụng Raspberry để làm server
    - + Truyền nhận dữ liệu qua mạng Internet bằng giao thức HTTP
    - + Cơ sở dữ liệu MySql và xử lý trên Webserver thông qua ngôn ngữ PHP và hiển thị ra giao diện web bằng ngôn ngữ html và css
  - Xây dựng mô hình thực nghiệm
- Thiết kế, thi công sơ đồ mạch và giải thuật đọc các thông số và giao tiếp giữa vi xử lý, ESP8266 và server

## 2. LÝ THUYẾT

### 2.1 Tổng quan thiết kế



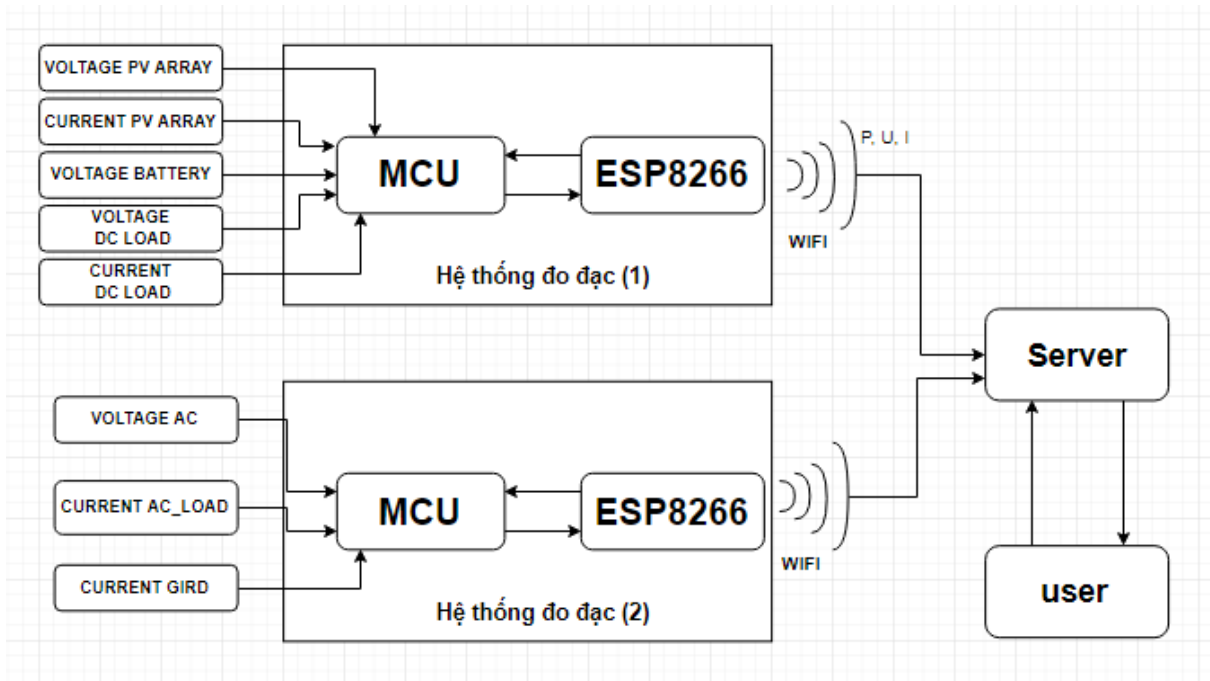
Hình 2.1 Mô hình hệ thống năng lượng mặt trời

Luận văn mô phỏng đo đạc đối với mô hình hệ thống Hybrid Solar PV System. Năng lượng từ PV array được đưa tới khối charge controller và được lưu trữ ở Battery, khối charge controller sẽ cung cấp điện cho DC Load và bộ chuyển đổi DC/AC. Bộ DC/AC có nhiệm vụ chuyển đổi điện áp DC thành AC đưa tới khối Distribution, và lưu trữ lên lưới điện. Khối Distribution sẽ cung cấp điện cho AC Load.

Để giám sát hệ thống này, ta xây dựng 2 điểm đo đạc ở (1) và (2).

- Điểm (1)
  - Đo Voltage của PV ARRAY, BATTERY, DC load
  - Đo current PV ARRAY, DC Load
- Điểm (2)

- Đo voltage AC
- Đo current AC Load, Gird



Hình 2.2 Mô hình hệ thống giám sát năng lượng mặt trời

Hệ thống đo đặc gồm 2 khối chính là MCU và ESP8266. Các dữ liệu đầu vào sẽ qua mạch cách ly để vào MCU, ở đây, MCU sẽ đo và tính toán ra các thông số U, I, P cần thiết của hệ thống, sau đó MCU đóng gói gửi cho ESP8266. ESP8266 có nhiệm vụ 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 hoặc App.

## 2.2 Tổng quan về năng lượng mặt trời

### 2.2.1 Ưu, nhược điểm của năng lượng mặt trời

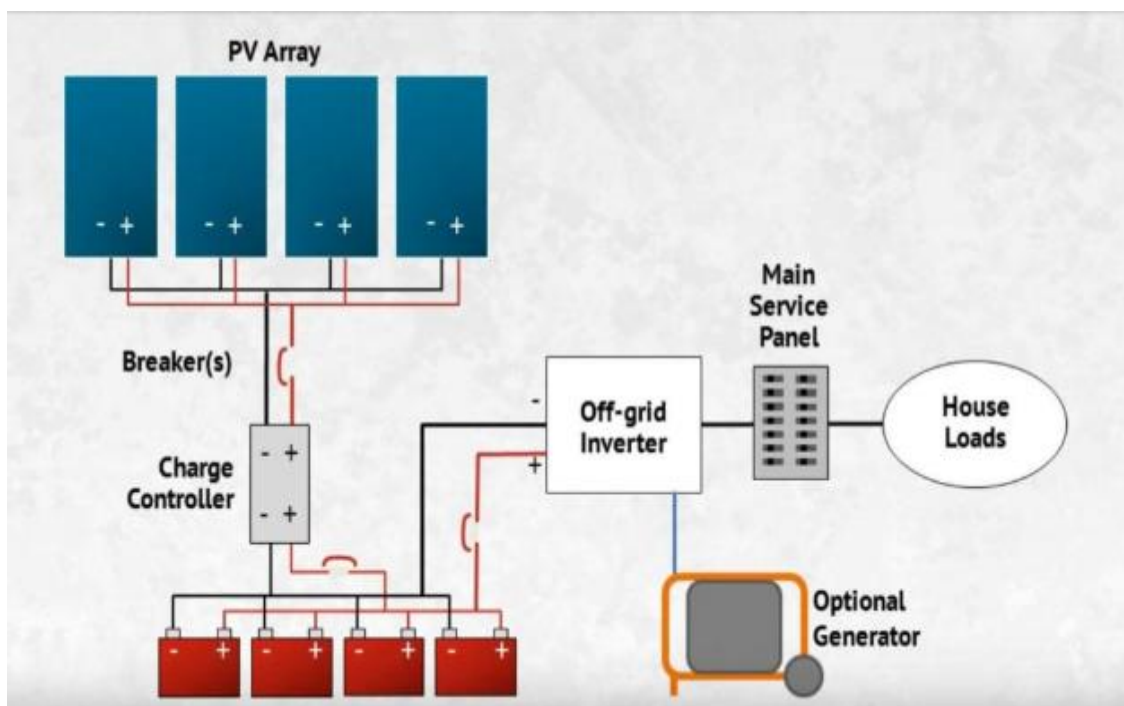
Năng lượng mặt trời là vô tận, dư thừa để đáp ứng nhu cầu về năng lượng của nhân loại, theo tính toán của NASA, mặt trời còn có thể cung cấp năng lượng cho chúng ta khoảng 6,5 tỉ năm nữa. Tiềm năng của năng lượng mặt trời là rất lớn, mỗi ngày bề mặt trái đất được hưởng 120.000 terawatts (TW) của ánh sáng 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.

### 2.2.2 Một số hệ thống năng lượng mặt trời

#### 2.2.2.1 Off-Grid Solar PV System



Hình 2.3 Mô hình hệ thống năng lượng mặt trời Off-Grid



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 lượng từ pin (DC) qua bộ inverter chuyển thành điện AC, cung cấp cho House Load.

Nhược điểm của 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, ...

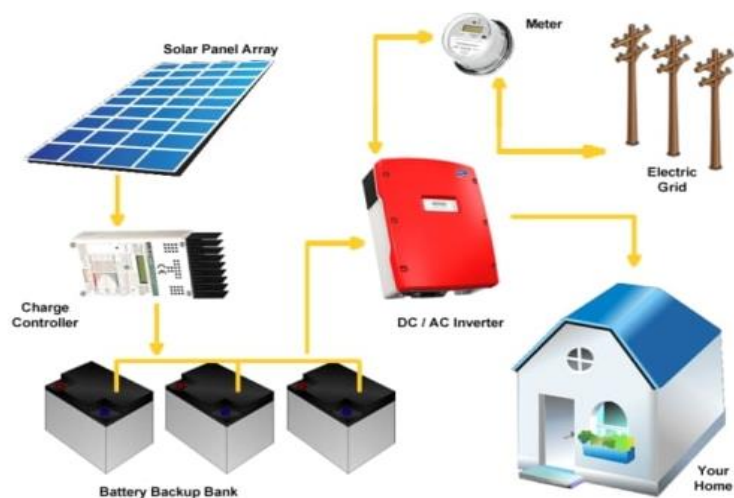
### 2.2.2.2 On-Grid Solar PV System



Hình 2.4 Mô hình hệ thống năng lượng mặt trời On-Grid

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 Array. Vào những ngày nắng năng lượng mặt trời dồi dào, thì ta có 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.

### 2.2.2.3 Hybrid Solar PV System



Hình 2.5 Mô hình hệ thống năng lượng mặt trời Hybrid

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

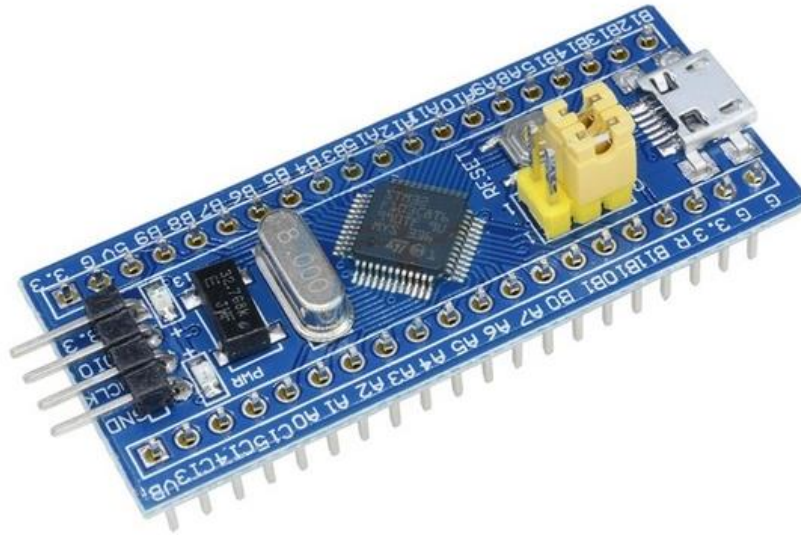
## **2.3 Giới thiệu vi điều khiển STM32F103C8T6**

### **2.3.1 Giới thiệu chung**

Trong vài năm trở lại đây, một trong những xu hướng chủ yếu trong các thiết kế với vi điều khiển là sử dụng các chip lõi ARM như một vi điều khiển đa dụng. Ngày nay các nhà sản xuất IC đưa ra thị trường rất nhiều dòng vi điều khiển sử dụng lõi ARM. Tập đoàn ST Microelectronic đã cho ra mắt dòng STM32, vi điều khiển đầu tiên dựa trên nền lõi ARM Cortex-M3 thế hệ mới do hãng ARM thiết kế. Dòng STM32 thiết lập các tiêu chuẩn mới về hiệu suất, chi phí, cũng như khả năng đáp ứng các ứng dụng tiêu thụ năng lượng thấp và tính điều khiển thời gian thực khắt khe. Với ứng dụng rộng rãi: từ điện tử dân dụng, xe hơi đời mới, game, mobile , laptop... Dòng STM32 tiêu thụ năng lượng thấp trong khi đó hiệu suất cực cao và đồng thời với sự đồ sộ về ngoại vi (GPIO, I2C, SPI, ADC, USB, Ethernet, CAN....), ST cung cấp cho chúng ta các thư viện trực tiếp cho mỗi dòng ARM (gọi là CMSIS - Cortex Microcontroller Software Interface Standard), giúp dễ dàng hơn trong công việc lập trình.

STM32 đầu tiên gồm 14 biến thể khác nhau, được phân thành hai nhóm: dòng Performance có tần số hoạt động của CPU lên tới 72Mhz và dòng Access có tần số hoạt động lên tới 36Mhz. Các biến thể STM32 trong hai nhóm này tương thích hoàn toàn về cách bố trí chân và phần mềm, đồng thời kích thước bộ nhớ FLASH ROM có thể lên tới 128K và 20K SRAM.

STM32F103C8T6 là vi điều khiển 32bit dựa trên nền lõi Cortex M3, dòng Performance với 64 Kb bộ nhớ Flash. Từ những phân tích ưu việt trên với ưu thế giá thành rẻ, có kit ra chân và mạch nạp nên luận văn chọn sử dụng con vi điều khiển này.



Hình 2.6 Kit STM32F103C8T6

#### Thông tin chung về STM32F103C8T6

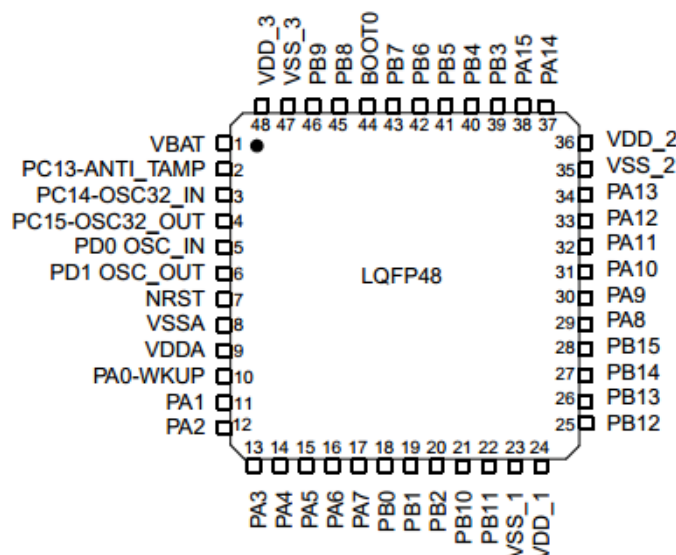
- Lõi : ARM 32 bit Cortex M3
- Tần số hoạt động lên tới 72 MHz
- Bộ nhớ : 64 Kb Flash , 20Kb SRAM
- Điện áp : 2~3.6 VDC
- Tổng số I/O : 32
- ADC : 2x12 bit, tần số lấy mẫu 1MHz
- DMA : Điều khiển 7 kênh DMA
- Timer : 4 bộ, 16 bit ( IC, OC, PWM )
- Giao diện kết nối : 2xI2C, 3xUSART, 2xSPI, CAN, 1xUSB 2.0 full- speed, 1xCAN

Những tính năng này làm cho vi điều khiển STM32F103C8T6 thích hợp cho một loạt các ứng dụng như điều khiển động cơ, kiểm soát các ứng dụng nâng cao, thiết bị y tế và thiết bị cầm tay, máy tính và thiết bị ngoại vi chơi game, GPS, ứng dụng công nghiệp, PLC, biến tần, máy in, máy quét , hệ thống báo động, hệ thống liên lạc video.....

### 2.3.2 Sơ lược các chức năng chính sử dụng trong luận văn

#### 2.3.2.1 GPIO

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.



Hình 2.7 Sơ đồ chân của vi điều khiển STM32F103C8T6

CU STM32F103C8T6 có 48 chân, trong đó có 32 chân IO. Mỗi cổng GPIO đều có 2 thanh ghi 32-bit điều khiển. Như vậy ta có 64-bit để cấu hình 16 chân của một cổng GPIO. Vậy mỗi chân của cổng GPIO sẽ có 4 bit để điều khiển: 2 bit sẽ quy định hướng ra vào dữ liệu: input hay output, 2 bit còn lại sẽ quy định đặc tính dữ liệu.

Có 8 chế độ IO có thể lập trình cho từng pin: Input floating, Input pull-up, Input pull-down, Analog input, Output open-drain, Output push-pull, Alternate function push-pull, Alternate function open-drain.

**Mode info:** Các bit mode[1:0] cấu hình chế độ input hoặc output

- 00 input( mặc định khi reset)
- 01 output max 10MHz
- 10 output Max 2Mhz
- 11 output Max 50 MHz

**Input Mode :** CNF[1:0] info qui định đặt tính dữ liệu

- 00 analog input
- 01 floating input(digital)
- 10 input với pullup/pulldown.
- 11 reserver

**Output Mode :** CNF[1:0] info qui định đặt tính dữ liệu

- 00 output push/pull
- 01 output open drain
- 10 alternate output push/pull
- 11 alternate output open drain

Chế độ input pullup/pulldown sẽ do giá trị bit tương ứng trên thanh ghi ODR quyết định. Các pin IO đều có dạng 5V tolerant (ngoài 2 pin chung chức năng với thạch anh đồng hồ thời gian thực) tức là có thể nối với các thiết bị dùng chuẩn 5V.

Các thanh ghi quan trọng:

- Input data register **GPIOx\_IDR**
- Output data register **GPIOx\_ODR**
- Bit Set/Reset register **GPIOx\_BSRR**
- Bit Reset register **GPIOx\_BRR**
- Lock mechanism register **GPIOx\_LCKR**

Trong thư viện “*stm32f10x\_gpio.h*” các pin tương ứng đã được định nghĩa sẵn để người dùng dễ sử dụng: GPIO\_Pin\_x. Các port được định nghĩa bằng tên GPIOx trong đó x: A,B,D... Thực chất GPIOx có dạng con trỏ trỏ tới địa gốc của port tương ứng. Ví dụ: để

cấu hình GPIO PINB.5 là cổng ra

Enable clock GPIOB :

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE);
```

Cấu hình cổng ra :

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
```

```
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
```

```
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

```
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

### 2.3.2.2 USART

Là một trong những giao tiếp phổ biến và dễ sử dụng nhất, dùng để truyền nhận dữ liệu giữa hai hay nhiều thiết bị khác nhau. UART có hai đường truyền và nhận dữ liệu riêng biệt do đó nó có thể đồng thời vừa truyền, vừa nhận dữ liệu một lúc (được gọi là truyền song công). UART là chuẩn truyền thông không đồng bộ, nghĩa là 2 thiết bị giao tiếp với nhau cần phải biết được các thông số của khung truyền như tốc độ, độ dài khung, số bit stop, bit parity... Khi muốn dùng giao tiếp UART thì chúng ta phải khai báo hàm khởi tạo giá trị cho các thanh ghi UART.

STM32 có đến 3 khối USART, mỗi khối có khả năng hoạt động đến tốc độ 4.5Mbps, có khả năng hỗ trợ giao tiếp không đồng bộ UARTS, modem cũng như giao tiếp hồng ngoại và Smartcard. Trong đó khối USART3 nằm trên APB2 ở PortB10- Rx và PortB11- Tx với xung nhịp hoạt động 72MHz, các khối còn lại nằm trên APB1, khối USART1 ở port PA9-Tx, PA10-Rx và khối USART2 nằm ở PA2-Tx, PA3-Rx hoạt động ở xung nhịp 36MHz.

Ví dụ để cấu hình khối USART1 ở port PA9-Tx, PA10-Rx hoạt động, ta cần thực hiện: Cấu hình GPIO các chân PA9, PA10 là chức năng Uart, sau đó cấu hình thông số truyền Uart : Tốc độ Baud, chiều dài khung truyền, số lượng bit stop, có hay không kiểm tra chẵn lẻ...

```
huart1.Instance = USART1;
```

```
huart1.Init.BaudRate = 9600;
```

```
huart1.Init.WordLength = UART_WORDLENGTH_8B;  
huart1.Init.StopBits = UART_STOPBITS_1;  
huart1.Init.Parity = UART_PARITY_NONE;  
huart1.Init.Mode = UART_MODE_TX_RX;  
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;  
huart1.Init.OverSampling = UART_OVERSAMPLING_16;  
if (HAL_UART_Init(&huart1) != HAL_OK)
```

### 2.3.2.3 ADC

ADC – Analog to digital Converter là bộ chuyển đổi tín hiệu tương tự sang tín hiệu số. ADC được ứng dụng rất nhiều như đo nhiệt độ, đọc giá trị điện áp, cường độ dòng điện, đọc phím nhấn, đọc giá trị biến trở, bảo vệ động cơ, ...

ADC trong STM32F103 là bộ ADC có 12 bit tức là giá trị đọc về nằm trong khoảng 0  $\rightarrow 2^{12} = 4096$ . Giá trị điện áp đầu vào bộ ADC được cung cấp trên chân VDDA và thường lấy bằng giá trị cấp nguồn cho vi điều khiển VDD(+3V3). STM32F103C8 có 2 kênh ADC đó là ADC1 và ADC2, mỗi kênh có tối đa là 9 channel với nhiều mode hoạt động như: single, continuous, scan hoặc discontinuous. Kết quả chuyển đổi được lưu trữ trong thanh ghi 16 bit.

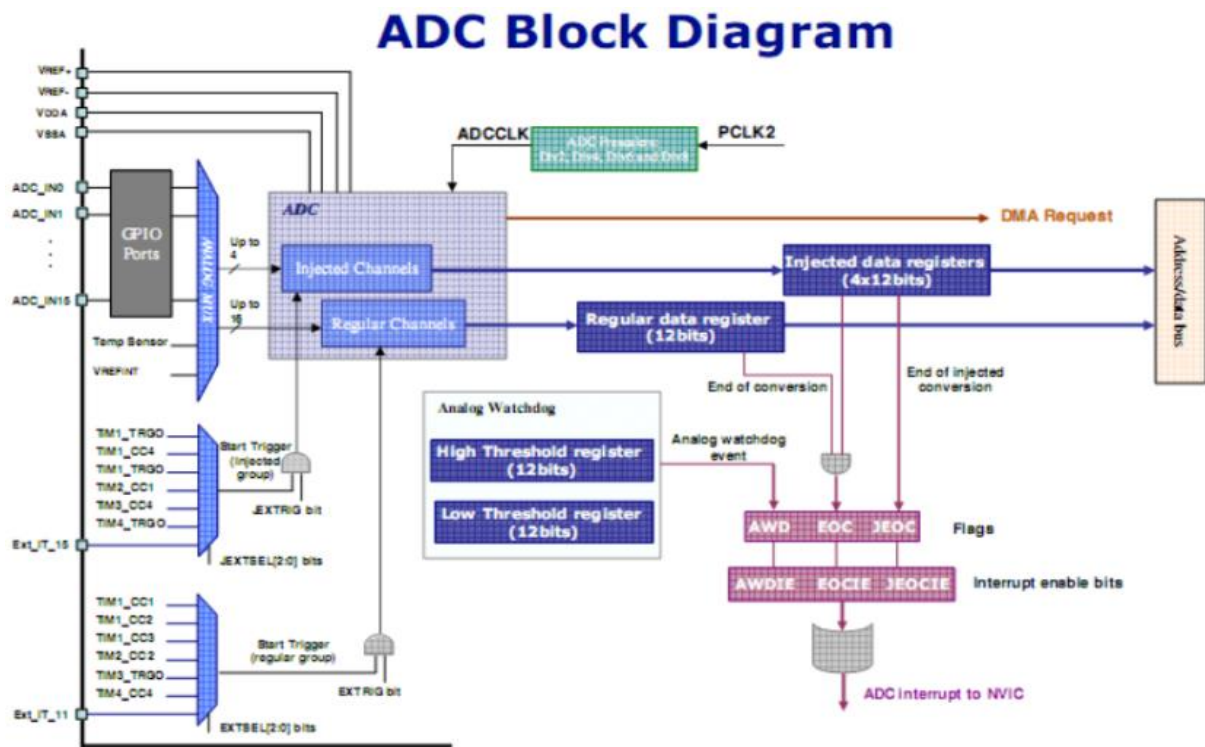
Một số tính năng chính được liệt kê như sau:

- Độ phân giải 12 bit tương ứng với giá trị maximum là 4095.
- Có các ngắt hỗ trợ như End conversion, End of Injected Conversion and Analog Watchdog Event.
- Single mode hay Continuous mode.
- Tự động calib và có thể điều khiển hoạt động ADC bằng xung Trigger.
- Thời gian chuyển đổi nhanh : 1us tại tần số 65Mhz.



- Điện áp cung cấp cho bộ ADC là 2.4V -> 3.6V. Nên điện áp Input của thiết bị có ADC  $2.4V \leq V_{IN} \leq 3.6V$ .
- Có bộ DMA giúp tăng tốc độ xử lí.

### Sơ đồ khối của bộ ADC trong STM32F103:



### Hình 2.8 Sơ đồ khối ADC

## 2.4 Giới thiệu SoC ESP8266

### 2.4.1 Giới thiệu

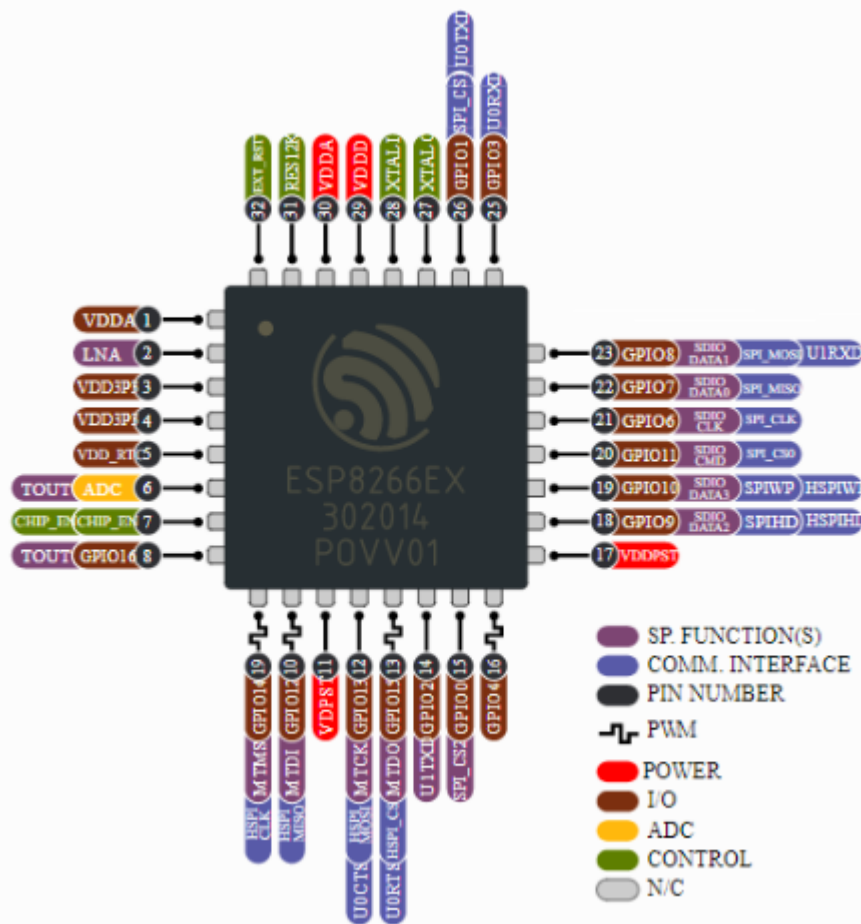
ESP8266 là dòng chip tích hợp Wifi 2.4Ghz, có thể lập trình được, giá thành rẻ được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems

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.

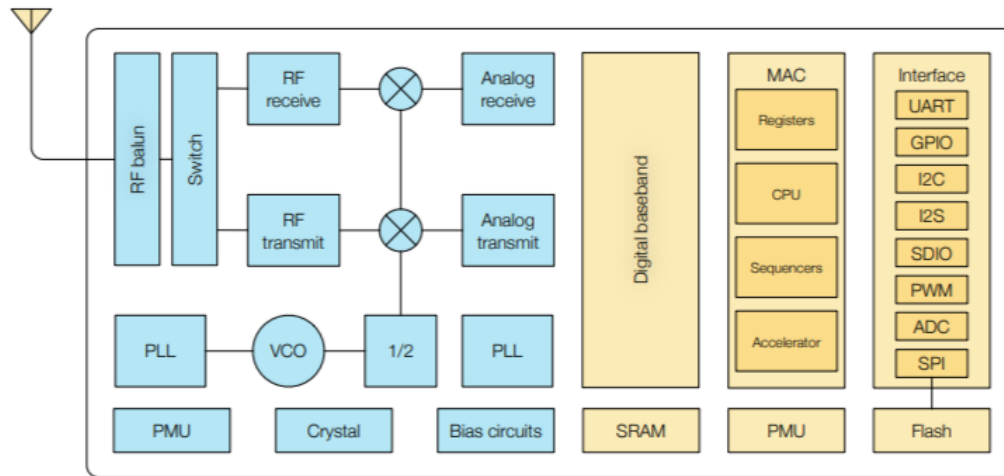


Hình 2.9 Chip ESP8266

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 IEEE 802.11 b/g/n, Wi-Fi 2.4 GHz
  - + Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network
  - + Hỗ trợ WEP, WPA/WPA2, Open network
- Tích hợp giao thức TCP/IP
- Hỗ trợ nhiều loại anten
- 16 chân GPIO
- Hỗ trợ SDIO 2.0, UART, SPI, I2C, PWM, I<sup>2</sup>S 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 tiêu 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 dải nhiệt khá rộng  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  (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.

### 2.4.2 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:

#### 2.4.2.1 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.

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

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

Hỗ trợ Wi-Fi Alliance, SRRC, FCC, CE, TELEC, IC & KCC Certified, RoHS,

Halogen Free, REACH & CFSI Compliant, HTOL, ESD-HM, MSL,  $\mu$ HAST, HTSL

Hỗ trợ Cloud Server Development

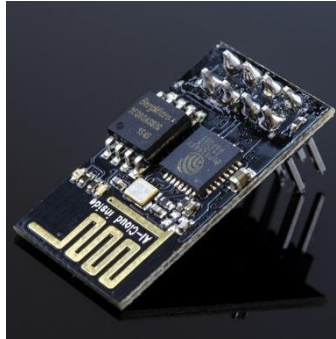
Custom firmware development qua SDK

Cấu hình người dùng qua AT Instruction Set, Cloud Server và ứng dụng Android/iOS

Khoảng cách giữa các chân 2.54mm

#### 2.4.2.2 ESP-01:

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



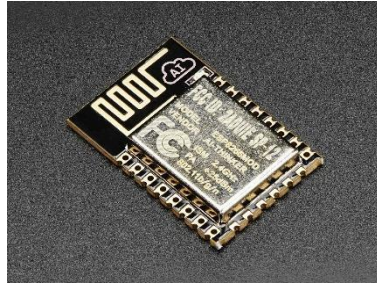
Hình 2.12 ESP-01

Tích hợp sẵn 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

### 2.4.2.3 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 hỗ trợ bảo mật WPA/WPA2

Khoảng cách giữa các chân 2mm

### 2.4.3 Kết nối với ESP8266

#### 2.4.3.1 Giao tiếp UART

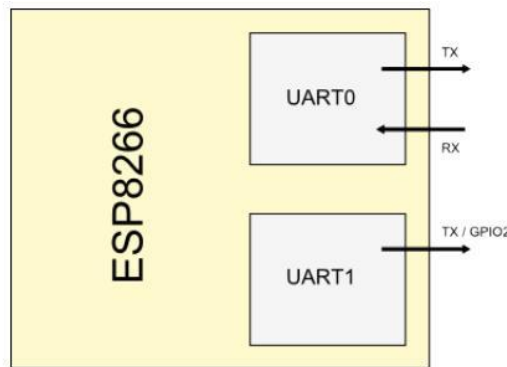
UART (Universal Asynchronous Receiver – Transmitter) thực hiện chuyển đổi giữa dữ liệu nối tiếp và song song. Một chiều, UART chuyển đổi dữ liệu song song bus hệ thống ra dữ liệu nối tiếp để truyền đi. Một chiều khác, UART chuyển đổi dữ liệu nhận được dạng dữ liệu nối tiếp thành dạng dữ liệu song song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

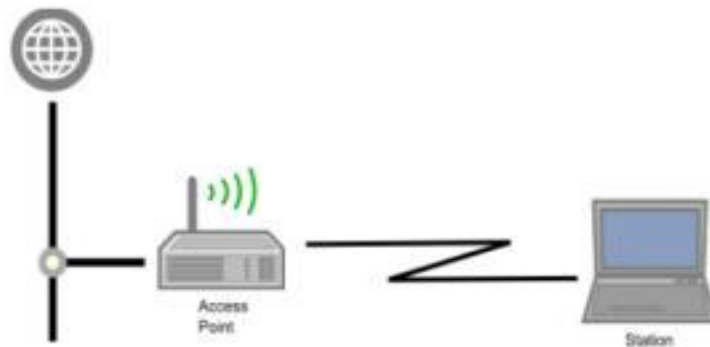
Thông qua UART, chúng ta có thể gửi các tập lệnh và quan sát các dữ liệu nhận được trên màn hình thông qua terminal. Mục đích quan trọng nhất của UART là để ghi các chương trình lên bộ nhớ flash của ESP8266. Có rất nhiều cách để thực hiện việc này thông qua các platform như IDE Arduino, Eclipse, PlatformIO... 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.

ESP8266 còn có một 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



Hình 2.14 UART trên ESP8266



Hình 2.15 Mô hình mạng Wifi

#### 2.4.3.2 Kết nối 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 nhanh 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.

#### 2.4.4 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.

#### 2.4.4.1 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 ESP8266 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 Management, ...

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.

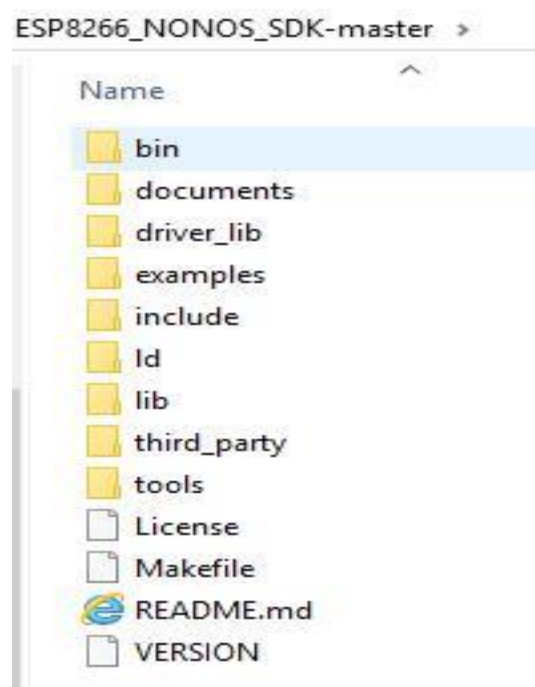
Đây là mô hình lập trình truyền thống mà chúng ta thường sử dụng khi bắt đầu quá trình học lập trình ứng dụng. Ưu điểm của nó là đơn giản nên ổn định và được nhà sản xuất cập nhật các tính năng mới đầu tiên. Hiện nay, phiên bản mới nhất của Non-OS SDK là V2.0.0.

**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]

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



Hình 2.16 Bộ thư viện ESP8266 NONOS\_SDK

**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

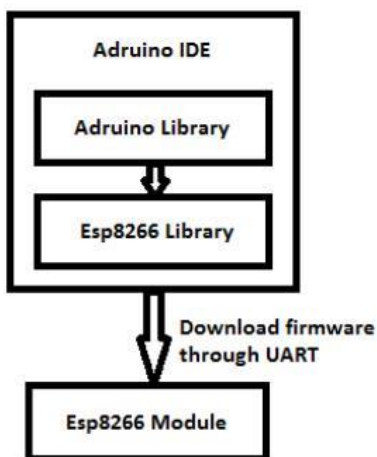
**ld:** Chứa các file tương tác với các phần mềm chạy SDK. Nhà sản xuất khuyến cáo không nên sửa đổi nó trừ khi có những yêu cầu đặc biệt.

**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.

#### 2.4.4.2 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 2.17 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
```

```
detachInterrupt(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

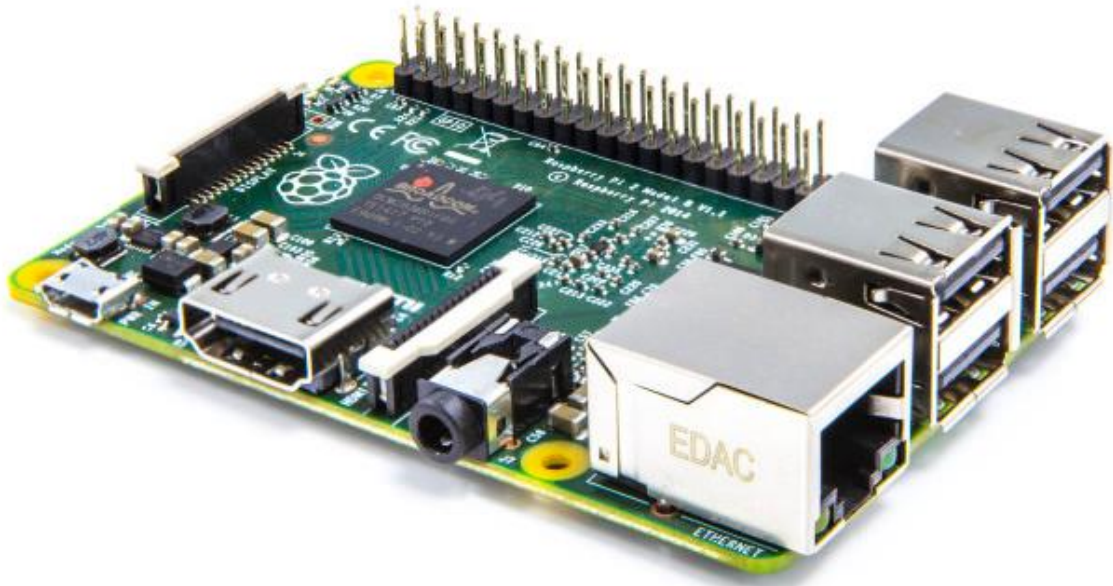
Wifi.localIP() //đọc địa chỉ IP khi hoạt động như 1 thiết bị Wifi

Wifi.softAPIP() //đọc địa chỉ IP khi hoạt động như 1 Access Point

WifiUDP.xxxx() // hỗ trợ truyền nhận dữ liệu qua UDP

## 2.5 Máy tính nhúng Raspberry

### 2.5.1 Giới thiệu chung



Hình 2.18 Máy tính Raspberry pi 2

Raspberry Pi là một máy vi tính rất nhỏ gọn, kích thước hai cạnh chỉ cỡ một cái thẻ ATM. Người ta đã tích hợp mọi thứ cần thiết trong đó để bạn sử dụng như một cái máy vi tính. Trên bo mạch của Pi có CPU, GPU, RAM, khe cắm thẻ microSD, Wi-Fi, Bluetooth và 4 cổng USB 2.0. Khi mua Pi về, chúng ta chỉ việc cài hệ điều hành (thực ra là copy/paste cái thư mục vô thẻ nhớ), gắn chuột, bàn phím và màn hình là bắt đầu sử dụng được rồi (hoặc cao cấp hơn xịu là remote desktop từ một máy khác qua, hoặc SSH)

Nhiệm vụ ban đầu của dự án Raspberry Pi là tạo ra máy tính rẻ tiền có khả năng

lập trình cho những sinh viên, nhưng Pi đã được sự quan tâm từ nhiều đối tượng khác nhau. Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lý SoC Broadcom BCM2835 (là chip xử lý mobile mạnh mẽ có kích thước nhỏ hay được dùng trong điện thoại di động) bao gồm CPU, GPU, bộ xử lý âm thanh/video, và các tính năng khác, ... tất cả được tích hợp bên trong chip có điện năng thấp này.

Dưới đây là thông số kỹ thuật của Raspberry Pi 2:

- CPU: Lõi tứ ARM Cortex A7, tốc độ 900MHz
- GPU: Broadcom VideoCore IV – 250MHz
- Bộ nhớ RAM: 1GB, tuy nhiên bị chia sẻ 250MHz cho GPU
- Cổng USB: 4
- Video output: HDMI, composite video thông qua jack 3,5 mm
- Audio outputs: Analog qua jack 3,5 mm; kỹ thuật số thông qua cổng HDM
- Bộ nhớ trong: thẻ nhớ microSD
- Kết nối mạng: Ethernet 10/100Mbps
- Nguồn điện: 5 V qua MicroUSB hoặc GPIO
- Kích thước: 85.60mm × 56.5mm
- Trọng lượng: 45g

Ban đầu, chiếc máy tính này được cài sẵn hệ điều hành Raspbian được thiết kế riêng cho bộ vi xử lý của Raspberry Pi 2. Tuy nhiên theo như người tạo ra chiếc máy tính này là Eben Upton cho biết thì nó có thể chạy các nền tảng hệ điều hành khác như Ubuntu hay thậm chí là Win 10, Raspberry Pi 2 hiện tại cũng đủ sức mạnh để thay thế những chiếc PC cơ bản.

## 2.5.2 Hệ điều hành và phần mềm

### 2.5.2.1 Hệ điều hành

Raspberry Pi có rất nhiều hệ điều hành hỗ trợ, trong đó có Raspbian là hệ điều hành chính thức của Raspberry Pi Foundation, ngoài ra có 7 hệ điều hành khác được xác nhận hỗ trợ và khá khá hệ điều hành do các nhà phát triển tự tối ưu.



Một số hệ điều hành thông dụng cho Raspberry Pi

- Raspbian: <https://www.raspberrypi.org/downloads/raspbian/>
- Ubuntu Mate: <https://ubuntu-mate.org/raspberry-pi/>
- Snappy Core Ubuntu:  
<https://developer.ubuntu.com/en/snappy/start/#snappy-raspi2>
- Windows 10 IoT Core: <http://ms-iot.github.io/content/en--US/Downloads.htm>
- OSMC: <https://osmc.tv/download/>
- OpenELEC: <http://openelec.tv/get-openelec>
- PiNet: <http://pinet.org.uk/>
- RiscOS: <https://www.riscosopen.org/content/downloads/raspberry-pi>

#### 2.5.2.2 Phần mềm

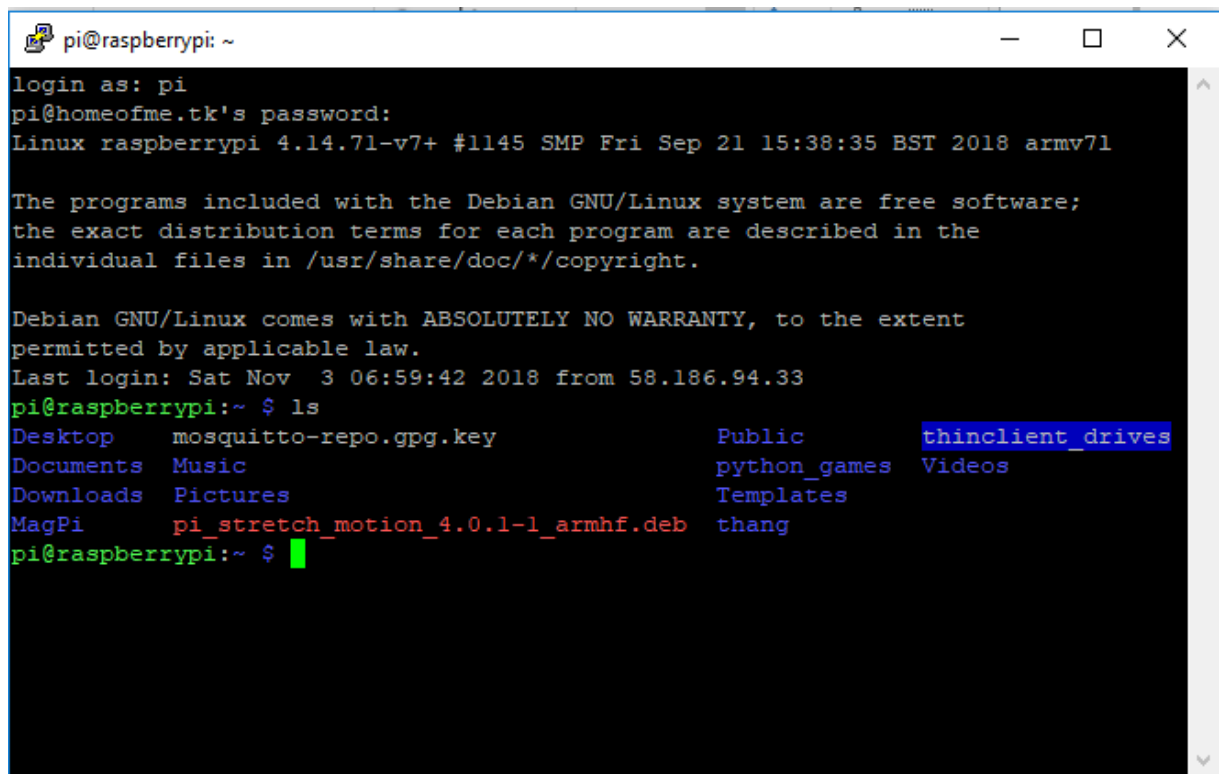
##### a. SSH

SSH, hoặc được gọi là Secure Shell, là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet. Dịch vụ được tạo ra nhằm thay thế cho trình Telnet vốn không có mã hóa và sử dụng kỹ thuật cryptographic để đảm bảo tất cả giao tiếp gửi tới và gửi từ server từ xa diễn ra trong tình trạng mã hóa. Nó cung cấp thuật toán để chứng thực người dùng từ xa, chuyển input từ client tới host, và relay kết quả trả về tới khách hàng.

Lệnh của SSH có ba phần:

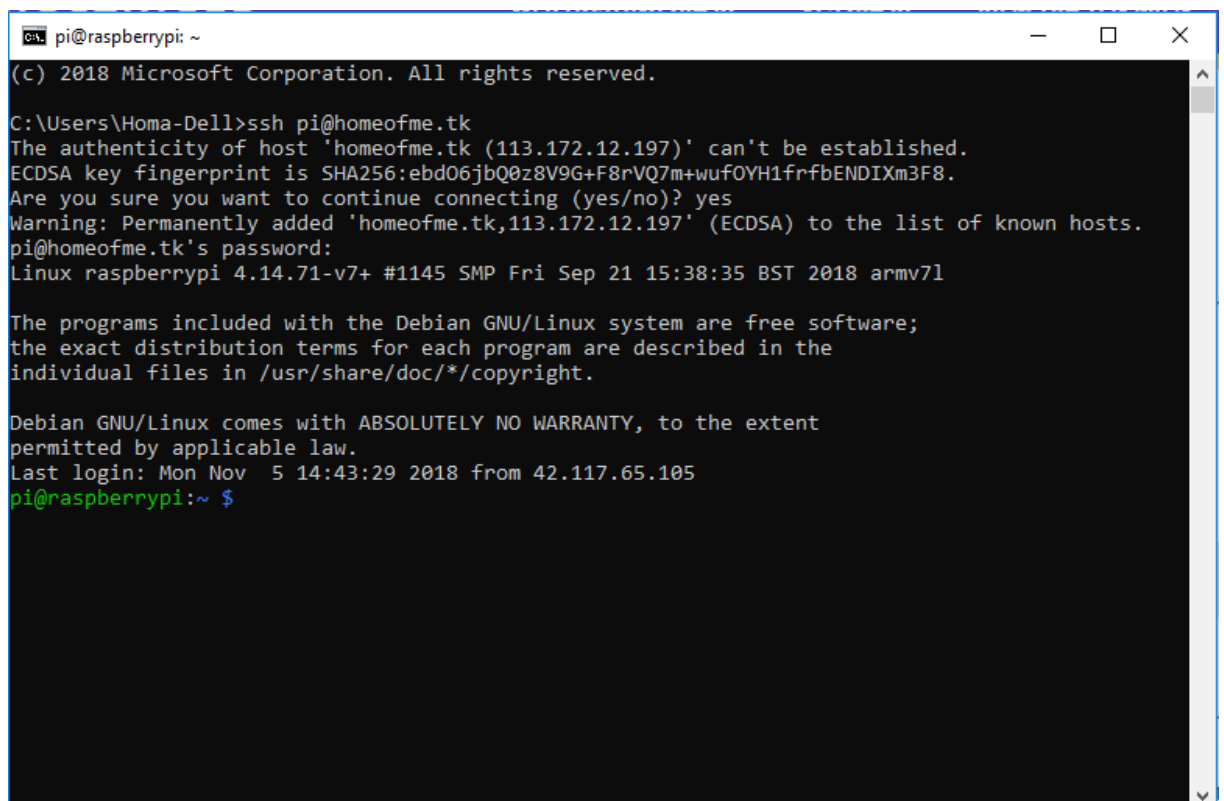
```
ssh {user}@{host}
```

SSH key command cho hệ thống biết là bạn muốn mở một kết nối được mã hóa Secure Shell Connection. **{user}** đại diện cho tài khoản người dùng bạn muốn dùng để truy cập. Ví dụ, bạn muốn truy cập user **root**, thì thay root tại đây. User root là user quản trị hệ thống với toàn quyền để chỉnh sửa bất kỳ điều gì trên hệ thống. **{host}** đại diện cho máy tính bạn muốn dùng để truy cập. Nó có thể là một địa chỉ IP (ví dụ **113.172.12.197**) hoặc một tên miền (ví dụ: **homeofme.tk**).



```
pi@raspberrypi: ~  
login as: pi  
pi@homeofme.tk's password:  
Linux raspberrypi 4.14.71-v7+ #1145 SMP Fri Sep 21 15:38:35 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat Nov 3 06:59:42 2018 from 58.186.94.33  
pi@raspberrypi:~ $ ls  
Desktop      mosquitto-repo.gpg.key      Public      thinclient_drives  
Documents    Music                       python_games Videos  
Downloads    Pictures                   Templates  
MagPi        pi_stretch_motion_4.0.1-l_armhf.deb  thang  
pi@raspberrypi:~ $
```

Hình 2.19 Remote raspberry bằng SSH thông qua phần mềm PUTTY



```
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\Users\Homa-Dell>ssh pi@homeofme.tk  
The authenticity of host 'homeofme.tk (113.172.12.197)' can't be established.  
ECDSA key fingerprint is SHA256:ebd06jbQ0z8V9G+F8rVQ7m+wufOYH1frfbENDIXm3F8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'homeofme.tk,113.172.12.197' (ECDSA) to the list of known hosts.  
pi@homeofme.tk's password:  
Linux raspberrypi 4.14.71-v7+ #1145 SMP Fri Sep 21 15:38:35 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Nov 5 14:43:29 2018 from 42.117.65.105  
pi@raspberrypi:~ $
```

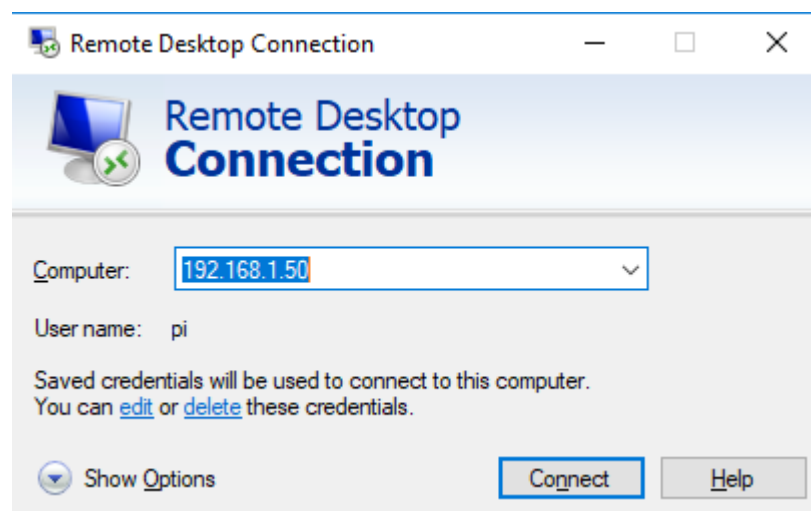
Hình 2.20 SSH raspberry bằng cmd trên win 10

(homeofme.tk là địa chỉ ip của raspberry)

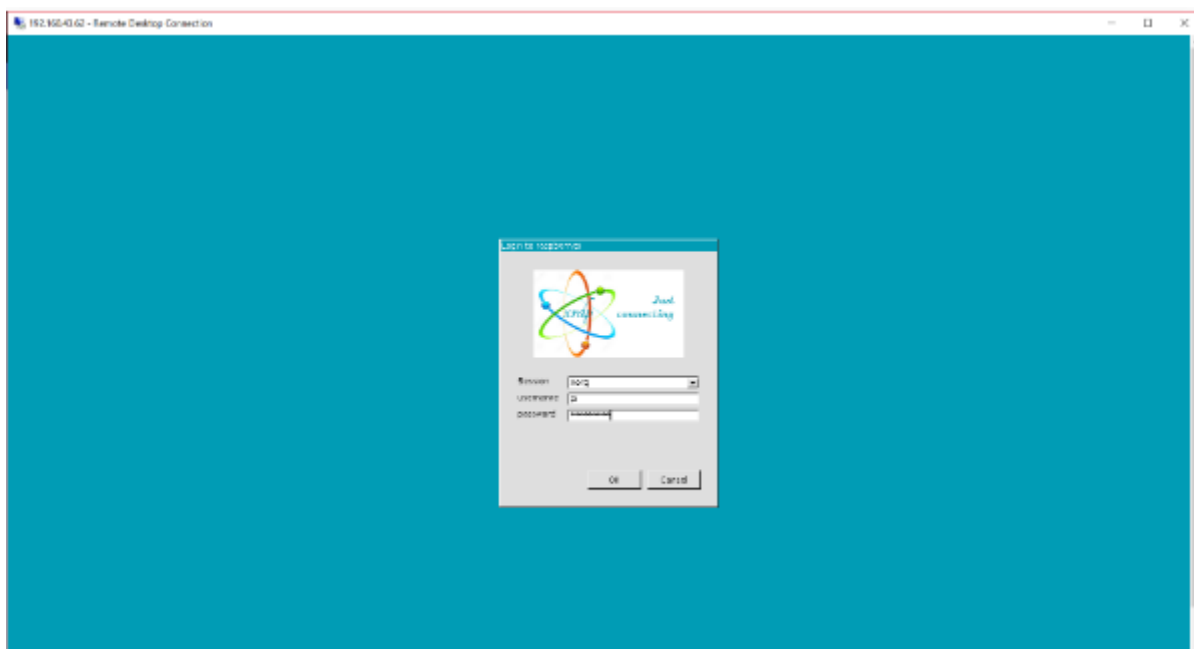
## b. REMOTE DESKTOP CONNECTION

Remote Desktop Services ( RDS ), được gọi là Terminal Services trong Windows Server 2008 và trước đó, là một trong các thành phần của Microsoft Windows cho phép người dùng kiểm soát máy tính từ xa hoặc máy ảo qua kết nối mạng. RDS là việc Microsoft triển khai máy khách mỏng, nơi phần mềm Windows và toàn bộ máy tính để bàn của máy tính chạy RDS, có thể truy cập được với máy khách từ xa hỗ trợ Giao thức bàn làm việc từ xa (RDP). Với RDS, chỉ giao diện người dùng phần mềm mới được chuyển sang hệ thống khách. Tất cả đầu vào từ hệ thống máy khách được truyền đến máy chủ, nơi thực hiện phần mềm. Điều này trái ngược với các hệ thống phát trực tuyến ứng dụng, như Microsoft App-V, trong đó các chương trình máy tính được truyền đến máy khách theo yêu cầu và được thực hiện trên máy khách.

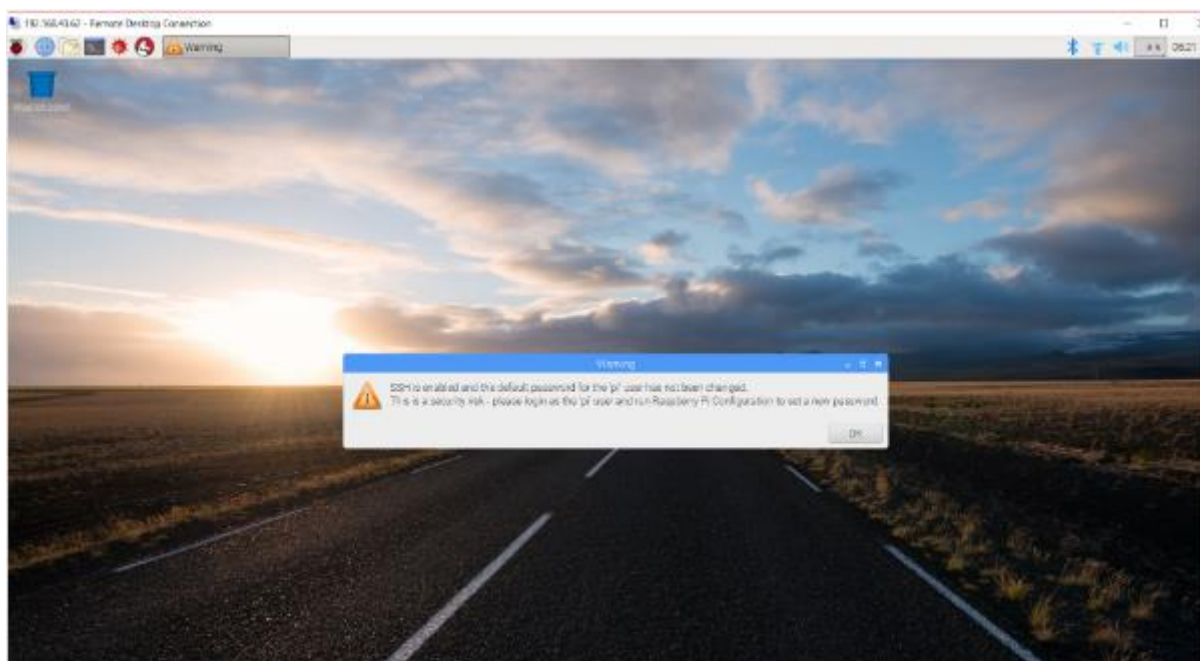
Remote Desktop Connection (RDC, còn được gọi là Remote Desktop, trước đây là Microsoft Terminal Services Client, mstsc hoặc tsclient) là ứng dụng máy khách cho RDS. Nó cho phép người dùng đăng nhập từ xa vào một máy tính nối mạng đang chạy máy chủ dịch vụ đầu cuối. RDC trình bày giao diện desktop (hoặc GUI ứng dụng) của hệ thống từ xa, như thể nó được truy cập cục bộ. Ngoài tên người dùng/ mật khẩu thông thường để cấp phép cho phiên từ xa, RDC cũng hỗ trợ sử dụng thẻ thông minh để ủy quyền. Với RDC 6.0, độ phân giải của phiên từ xa có thể được đặt độc lập với các cài đặt ở máy tính từ xa.



Hình 2.21 Remote Desktop Connection



Hình 2.22 Màn hình đăng nhập



Hình 2.23 Giao diện GUI

### 2.5.3 Một số lệnh cơ bản trên Raspberry

Người sử dụng Linux nói chung hay hệ điều hành Raspbian nói riêng thường có thói quen cài đặt/chỉnh sửa/thiết lập mọi thứ thông qua các câu lệnh vì nó giúp chúng ta không cần phải cài hay mở giao diện đồ họa vừa tốn tài nguyên vừa có cảm giác chậm chạp, bên cạnh đó cảm giác gõ lệnh trên Linux cũng khá là thú vị.

Có 2 chế độ người sử dụng trên Linux nói chung, một là chế độ người dùng với quyền truy cập cơ bản, và một chế độ với quyền truy cập quản trị viên (còn được gọi là root). Một số lệnh hoặc hành động không thể thực hiện với quyền cơ bản và ta sẽ cần phải đăng nhập vào chế độ root để thực hiện chúng. Do đó ta sẽ thường xuyên thấy tiền tố “*sudo*” trước câu lệnh, có nghĩa là mình đang nói với Raspbian rằng tôi cần thực thi các lệnh với quyền của root.

#### 2.5.3.1 Các câu lệnh cơ bản

- *apt-get update*: cập nhật phiên bản Raspbian. Lệnh này còn được dùng nếu bạn đổi Repository của Raspbian.
- *apt-get upgrade*: Nâng cấp toàn bộ các gói phần mềm đã cài đặt trên Raspbian.
- *clear*: Xóa màn hình terminal và các lệnh bạn đã thực thi trước đó. Lưu ý, nó chỉ xóa màn hình đi cho gọn, sạch mà thôi.
- *date*: Hiển thị ngày giờ hiện tại.
- *find / -name example.txt*: Tìm kiếm các file có tên là example.txt trên toàn hệ thống.
- *nano example.txt*: Mở file example.txt trong trình biên tập text **Nano** – Một trình biên tập text (text editor) quen thuộc của Linux
- *poweroff*: Tắt Raspberry Pi
- *raspi-config*: Mở menu thiết lập cấu hình Raspberry Pi hay Raspbian.
- *reboot*: Khởi động lại Raspberry Pi.
- *shutdown -h now*: Tắt Raspberry Pi ngay lập tức
- *shutdown -h 01:22*: Tắt Raspberry Pi vào lúc 1:22 AM.
- *startx*: Mở giao diện đồ họa của Raspbian (nếu bạn đang ở giao diện console)

#### 2.5.3.2 Các câu lệnh liên quan đến file, folder

- *cat example.txt*: Hiển thị nội dung của file example.txt.

- *cd /abc/xyz*: Di chuyển từ thư mục hiện tại sang thư mục /abc/xyz.
- *cp XXX*: Copy file hoặc thư mục XXX và paste vào vị trí mới.  
Ví dụ: *cp examplefile.txt /home/pi/office/* sẽ copy file *examplefile.txt* trong folder hiện tại đến /home/pi/office.  
Nếu file cần copy không ở folder hiện tại thì ghi rõ vị trí file cần copy (ví dụ *cp /home/pi/documents/examplefile.txt /home/pi/office/* sẽ copy file *examplefile.txt* trong folder documents sang folder office).
- *ls -la*: Hiển thị danh sách các file trong folder với các thông tin file size, ngày sửa, quyền hạn của file.
- *mkdir example\_directory*: Tạo folder tên *example\_directory* bên trong folder hiện tại.
- *mv XXX*: Nội dung cú pháp lệnh như lệnh *cp* nhưng mục đích của lệnh *mv* là move file thay vì copy..
- *rm example.txt*: Xóa file *example.txt*.
- *rmdir example\_directory*: Xóa thư mục *example\_directory* (nếu thư mục này đang trống).
- *touch*: Tạo file trắng mới trong folder hiện tại. Có thể sử dụng các lệnh khác như *nano*, v.v.

#### 2.5.3.3 Các câu lệnh liên quan đến mạng

- *ifconfig*: Kiểm tra tình trạng mạng hiện tại trên Raspberry Pi. Bạn có thể biết IP của Raspberry Pi nếu đang kết nối.
- *iwconfig*: Kiểm tra adapter không dây nào đang chạy.
- *iwlist wlan0 scan*: Hiển thị danh sách các mạng wifi trong khu vực.
- *iwlist wlan0 scan | grep ESSID*: Hiển thị danh sách các mạng wifi trong khu vực có tên theo yêu cầu.
- *nmap*: Quét mạng và hiển thị các thiết bị đang kết nối, cổng, giao thức, trạng thái của hệ thống, địa chỉ MAC và các thông tin khác.

- *ping*: Kiểm tra kết nối giữa các thiết bị trong mạng.
- *wget http://www.website.com/example.txt*: Tải file *example.txt* từ trên mạng về và lưu vào folder hiện tại.

#### 2.5.3.4 Các lệnh liên quan đến hệ thống

- *cat /proc/meminfo*: Hiển thị thông tin chi tiết về RAM của Raspberry Pi.
- *cat /proc/partitions*: Hiển thị thông tin các phân vùng của thẻ nhớ hoặc ổ cứng hoặc USB cắm trên Raspberry Pi.
- *cat /proc/version*: Hiển thị phiên bản Raspberry Pi đang sử dụng.
- *df -h*: Hiển thị thông tin dung lượng lưu trữ còn trống.
- *dpkg --get-selections | grep XXX*: Hiển thị các gói phần mềm đã cài đặt có liên quan đến từ khóa XXX.
- *dpkg --get-selections*: Hiển thị toàn bộ các phần mềm đã cài trên Raspberry Pi.
- *free*: Hiển thị lượng RAM còn trống.
- *hostname -I*: Hiển thị IP của Raspberry Pi.
- *lsusb*: Liệt kê các thiết bị USB đang cắm vào Raspberry Pi.
- *UP key*: Bấm phím mũi tên lên sẽ hiển thị các lệnh đã từng chạy trước đây.
- *vcgencmd measure\_temp*: Hiển thị thông tin nhiệt độ GPU.
- *vcgencmd get\_mem arm && vcgencmd get\_mem gpu*: Hiển thị thông tin RAM của CPU và GPU.

## 2.6 Giới thiệu về Web

### 2.6.1 Cơ sở dữ liệu và Webserver

#### 2.6.1.1 Cơ sở dữ liệu

Cơ sở dữ liệu viết tắt là CSDL, tên tiếng anh là Database, là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thỏa 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 chủ yếu được lưu trữ



dưới dạng bảng.

ID	NAME	CLASS	MARK	SEX
1	John Deo	Four	75	female
2	Max Ruin	Three	85	male
3	Arnold	Three	55	male
4	Krish Star	Four	60	female
5	John Mike	Four	60	female
6	Alex John	Four	55	male
7	My John Rob	Fifth	78	male
8	Asruid	Five	85	male
9	Tes Qry	Six	78	male
10	Big John	Four	55	female

Hình 2.24 Ví dụ về bảng cơ sở dữ liệu

Sự khác biệt cơ bản giữa cơ sở dữ liệu với dữ liệu thông thường đó là tính cấu trúc có hệ thống. Một tập hợp dữ liệu không có tính cấu trúc hệ thống nhất định không được gọi là cơ sở dữ liệu. Bất cứ cơ sở dữ liệu nào khi được tạo ra cũng cần phải được lưu trữ, quá trình lưu trữ này được gọi là hệ quản trị cơ sở dữ liệu. Hệ quản trị cơ sở dữ liệu là chương trình phần mềm giúp thực hiện việc lưu trữ cơ sở dữ liệu.

Hệ quản trị cơ sở dữ liệu khi lưu trữ cơ sở dữ liệu cần được đảm bảo tính cấu trúc có hệ thống trong cơ sở dữ liệu và đồng thời cần phải hỗ trợ việc đọc, chỉnh sửa, thêm và xóa dữ liệu một cách dễ dàng. Các hệ quản trị cơ sở dữ liệu được sử dụng phổ biến hiện nay bao gồm: Microsoft Access, MySQL, Oracle, PostgreSQL, SQL Server. Việc sử dụng các phần mềm hay các hệ quản trị cơ sở dữ liệu này giúp các nhà quản trị dễ dàng tạo mới, truy cập, tìm kiếm, chỉnh sửa... trên cơ sở dữ liệu. Để thực hiện các công việc này trên hệ quản trị cơ sở dữ liệu các nhà quản trị thường sử dụng các ngôn ngữ truy vấn mang tính chất cấu trúc: Structured Query Language hay còn gọi là SQL.

Do cơ sở dữ liệu (Database) là kiểu dữ liệu được lưu trữ riêng và được bảo vệ nên để sử dụng được database này ta cần phải có một số thông tin:



- Database Host: là địa chỉ của máy chủ sẽ lưu trữ các dữ liệu trong database mà ta cần sử dụng. Nếu sử dụng chính host chạy website để lưu trữ database thì thường ta sẽ khai báo Database Host là Localhost. Mặc định khi kết nối, nó sẽ kết nối qua cổng 3306 vì MySQL sử dụng cổng 3306. Một số nhà cung cấp đặc biệt hay các giải pháp Remote Database mới có thiết lập Database host khác ngoài Localhost.
- Database Name: Là tên của dữ liệu đã tạo trên máy chủ lưu trữ nó. Trong môi trường Localhost tên này sẽ được đặt là “databasename” nếu dữ liệu được tạo bằng công cụ phpMyAdmin. Còn trên các dịch vụ Shared Host thông thường thì Database Name sẽ có dạng là “username\_databasename” trong đó username là tên truy cập vào Host
- Database Username: là tên của người dùng đã được cấp quyền để truy và làm việc với database. Bởi khi ta làm việc với website, các tác động vào website sẽ được kết nối vào database thông qua database username và database password.
- Database Password: Là mật khẩu truy cập cho Database username.

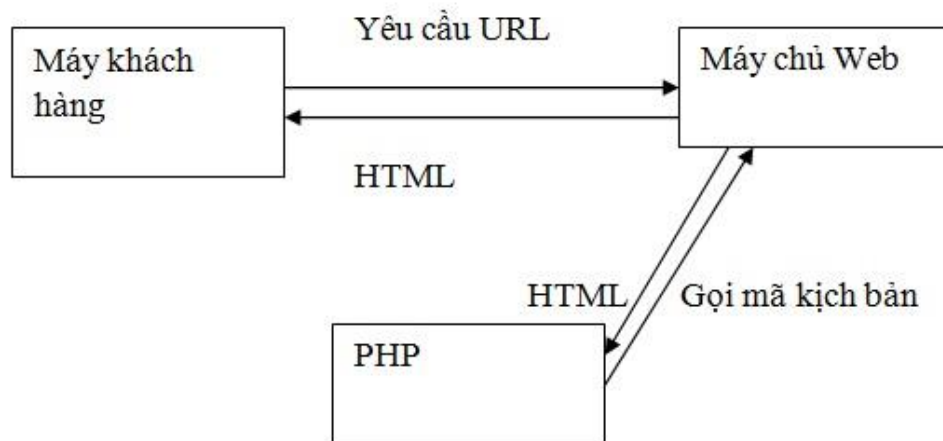
#### 2.6.1.2 Webserver

Web Server là máy chủ có dung lượng lớn, tốc độ cao, được dùng để lưu trữ thông tin như một ngân hàng dữ liệu, chứa những website đã được thiết kế cùng với những thông tin liên quan khác. Web Server có khả năng gửi đến máy khách những trang Web thông qua môi trường Internet qua giao thức HTTP, giao thức được thiết kế để gửi các file đến trình duyệt Web (Web Browser), và các giao thức khác. Bất kì máy tính nào cũng có thể đóng vai trò như một Web Server bởi việc cài đặt lên nó một chương trình phần mềm Server Software và sau đó kết nối vào Internet. Web Server Software được cài đặt, và chạy trên máy tính dùng làm Web Server, 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 máy tính kết nối đến một Web Server và gửi đến yêu cầu truy cập các thông tin từ một trang Web nào đó, Web Server sẽ nhận yêu cầu và gửi lại những thông tin mà người sử dụng mong muốn. Web Server còn có thể được tích hợp với CSDL, hay điều khiển việc kết nối vào CSDL để có thể truy cập và kết xuất thông tin từ CSDL lên các trang Web và truyền tải chúng đến người dùng.

Tất cả các Web Server đều có một địa chỉ IP (IP Address) hoặc cũng có thể có một tên miền Domain Name. Ví dụ khi ta gõ địa chỉ trang web có URL “http://www.abc.com/” vào trình duyệt web và nhấn Enter, trang web hiển thị ngay trên màn hình. Các bước cơ bản trong tiến trình truyền tải trang web đến màn hình người dùng bằng cơ chế hoạt động của Web server được thể hiện theo mô hình sau:

Sơ đồ hoạt động:



Hình 2.25 Tiến trình truyền tải web

Theo mô hình trên, trình duyệt web thực hiện một kết nối tới máy chủ web, yêu cầu một trang web và nhận lại nó. Sau đây, là thứ tự từng bước cơ bản xảy đến đằng sau màn hình của bạn:

Trình duyệt web tách địa chỉ website làm 3 phần:

- Phần giao thức: (“http”)
- Máy chủ tên miền: (www.abc.com)

- Tên tệp: ("index.htm")

Trình duyệt liên hệ với máy chủ tên miền để chuyển đổi tên miền "www.abc.com" ra địa chỉ IP (Internet Protocol). Sau đó, trình duyệt sẽ gửi tiếp một kết nối tới máy chủ có địa chỉ IP tương ứng qua cổng 80.

Dựa trên giao thức HTTP, trình duyệt gửi yêu cầu GET đến máy chủ, yêu cầu tệp "http://www.abc.com/index.htm" (cookies cũng sẽ được gửi kèm theo từ trình duyệt web đến máy chủ).

Tiếp đến, máy chủ sẽ gửi đoạn text dạng HTML đến trình duyệt web máy đã gửi yêu cầu (cookies cũng được gửi kèm theo từ máy chủ tới trình duyệt web, cookies được ghi trên đầu trang của mỗi trang web).

Trình duyệt web đọc các thẻ HTML, định dạng trang web và kết xuất ra màn hình.

### 2.6.2 Lập trình Web

Tất cả các công việc lập trình web đều được thực hiện với ngôn ngữ lập trình web. Hầu hết lập trình web đều được thực hiện bằng ngôn ngữ lập trình phía server (server – side). Mã này chạy trên máy chủ và sau đó cung cấp thông tin tính trở lại trình duyệt web. Các ngôn ngữ lập trình phổ biến nhất là: HTML, PHP, ASP.NET, Ruby on Rails, Perl, ASP cổ điển và JSP... Trong phạm vi nghiên cứu, luận văn đề cập tới ngôn ngữ HTML, CSS và PHP.

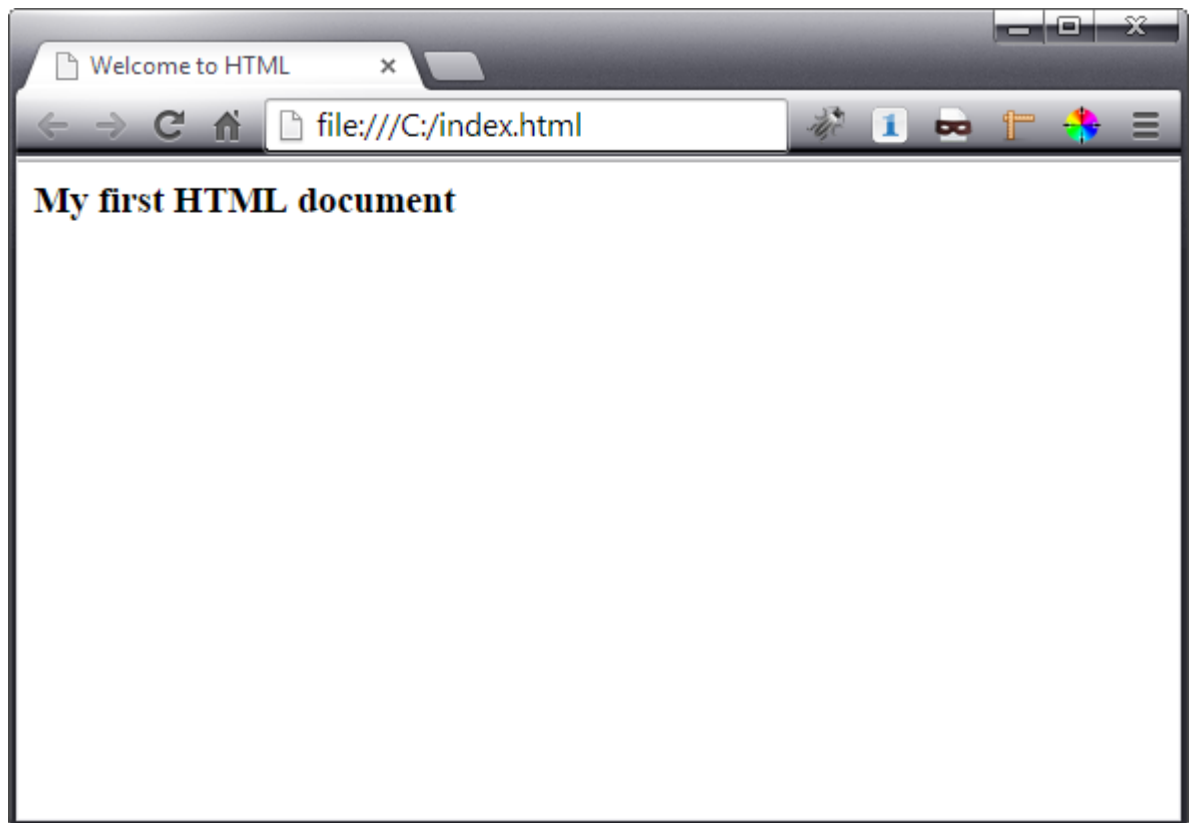
#### 2.6.2.1 HTML

HTML là viết tắt của Hyper Text Markup Language, nghĩa là ngôn ngữ đánh dấu siêu văn bản. Trong đó Hyper Text (siêu văn bản), là một đoạn text bất kì nhưng có chứa các link đến một nguồn thông tin khác (như một đoạn văn bản khác, một địa chỉ website, hình ảnh, âm thanh...). Ngôn ngữ đánh dấu này sẽ là cầu nối của sự giao tiếp giữa người dùng và trình duyệt, giúp trình duyệt hiểu được cách hiển thị trang web.

Tài liệu HTML tạo thành mã nguồn của trang Web. Khi được xem trên trình soạn thảo, tài liệu này là một chuỗi các thẻ và các phân tử, mà chúng xác định trang web

hiển thị như thế nào. Trình duyệt đọc các file có đuôi .htm hay .html và hiển thị trang web đó theo các lệnh có trong đó. HTML là kiến thức cơ bản mà bất cứ ai muốn học lập trình web hoặc học thiết kế web đều phải biết. Tất cả các trang web dù xử lý phức tạp đến đâu đều phải trả về dưới dạng mã nguồn HTML để trình duyệt có thể hiểu và hiển thị lên được. Ví dụ đoạn mã sau đây sẽ hiển thị thông điệp “My first html document” và trình duyệt sẽ hiển thị như hình:

```
<html>
  <head>
    <title>Welcome to html</title>
  </head>
  <body>
    <H3>My first html document</H3>
  </body>
</html>
```



Hình 2.26 Giao diện khi chạy thẻ HTML

Thông qua ví dụ trên, ta thấy được cấu trúc của HTML rất đơn giản và logic., với bố cục từ trên xuống dưới, từ trái qua phải, với 2 phần chính là HEAD và BODY. Các trang viết bằng ngôn ngữ HTML đều tuân theo cấu trúc cơ bản như sau:

- Mọi trang HTML đều phải khai báo DOCTYPE( định nghĩa chuẩn văn bản) ngay ở dòng đầu tiên.
- Phần <html>: Mọi tài liệu html phải bắt đầu bằng thẻ mở <html> và kết thúc bằng thẻ đóng </html>, chức năng cho trình duyệt biết mở đầu và kết thúc của trang HTML.
- Phần <head>: Phần tiêu đề bắt đầu bằng thẻ <head> và kết thúc bởi thẻ </head>. Phần này chứa tiêu đề mà được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ title, bắt đầu bằng thẻ <title> và kết thúc là thẻ </title>. Tiêu đề là phần khá quan trọng. Khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khóa chính yếu cho việc

tìm kiếm.

- Phần thân <body>: phần này nằm sau phần tiêu đề, bao gồm văn bản, hình ảnh và các liên kết mà bạn muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>.

#### 2.6.2.2 CSS

CSS là viết tắt của Cascade Style Sheet, là ngôn ngữ giúp trình duyệt hiểu được các thiết lập định dạng và bố cục cho trang web. CSS cho phép điều khiển thiết kế của nhiều thành phần HTML chỉ với duy nhất vùng chọn CSS. Điều này giúp giảm thiểu thời gian thiết kế và chỉnh sửa khi có thể tách biệt được cấu trúc HTML và định dạng CSS. CSS cho phép đưa các thông tin định nghĩa thẻ theo nhiều con đường khác nhau. Style có thể chỉ được quy định chỉ ở một thẻ HTML, được quy định trong một trang web, hoặc ở trong một file CSS bên ngoài. Nếu ta xem HTML là một bộ khung của một trang web, thì CSS chính là tất cả các chi tiết phụ họa được phủ lên khung với mục đích để trang trí làm cho website trở nên đẹp hơn, chuyên nghiệp hơn.

#### 2.6.2.3 PHP

PHP là một ngôn ngữ lập trình kịch bản trên môi trường website giúp nó giao tiếp dữ liệu với máy chủ (các ngôn ngữ giao tiếp với máy chủ gọi chung là Server-side Language). Một website tĩnh được làm hoàn toàn bằng các tập tin HTML hay CSS chỉ có thể hiển thị các văn bản đã khai báo trong tập tin HTML, nó không thể tự thân thay đổi được. Nhưng nếu người lập trình nhúng thêm PHP vào thì nó sẽ trở thành một website động, từ đó chúng ta có thể lưu các dữ liệu về trong cơ sở dữ liệu database trên máy chủ, cũng sử dụng PHP để lấy các dữ liệu từ database để hiển thị lên website tự động một cách liên tục mà không cần chỉnh sửa tập tin HTML. Một mã nguồn PHP có thể đặt bất cứ đâu trong tài liệu, mã nguồn PHP được bắt đầu bằng thẻ <?php và kết thúc bằng thẻ ?>

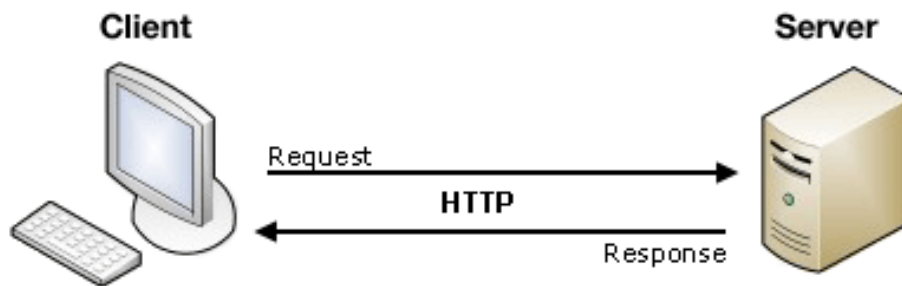
```
1 <?php
2 // mã nguồn PHP đặt tại đây
3 ?>
```

Tập tin mặc định của ngôn ngữ PHP có đuôi mở rộng là “.php”. Một file PHP thông thường chứa các thẻ HTML, và các mã kịch bản trong PHP. Trong phạm vi luận

văn, ngôn ngữ PHP được sử dụng để giao tiếp với cơ sở dữ liệu: cập nhật dữ liệu vào database và lấy dữ liệu từ database để hiển thị lên web.

### 2.6.3 Giao thức HTTP

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



Hình 2.27 Mô hình Client – Server web

Sở dĩ phần này, giao thức HTTP được đề cập đến bởi vì luận văn sử dụng 2 phương thức request HTTP là GET và POST là truyền và nhận dữ liệu giao tiếp với server. Hai phương thức này đều là gửi dữ liệu về server xử lý sau khi người dùng nhập thông tin vào form và thực hiện. Trước khi gửi thông tin, nó sẽ được mã hóa bằng cách sử dụng một giản đồ gọi là URL encoding. Giản đồ này là các cặp name/value được kết hợp với các kí hiệu = và các kí hiệu khác nhau được ngăn cách bởi dấu &.

Phương thức GET gửi thông tin người dùng đã được mã hóa thêm vào trên yêu cầu trang. Ví dụ `http://www.luanvan.com/index.htm?name=value1&name1=value1`. Ở đây, ta dùng phương thức HTTP GET để gửi dữ liệu vào database. Chúng ta thấy rằng GET lộ thông tin trên đường dẫn URL. Bằng thông của nó chỉ khoảng 1024 kí tự vì vậy GET hạn chế về số kí tự được gửi đi. GET không thể gửi dữ liệu nhị phân, hình

ảnh..

Một yêu GET lấy dữ liệu từ server bởi việc xác định các tham số trong đoạn URL của yêu cầu. Ví dụ:

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Server sẽ phản hồi lại yêu cầu như sau

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization, Accept
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

Phương thức POST truyền thông tin thông qua HTTP header, thông tin này được mã hóa như phương thức GET. Dữ liệu được gửi bởi phương thức POST rất bảo mật vì dữ liệu được gửi ngầm, không đưa lên URL. POST không có bất kì hạn chế nào về kích thước dữ liệu sẽ gửi, có thể gửi dữ liệu nhị phân, hình ảnh.

Ví dụ sau đây chỉ cách sử dụng của phương thức POST để gửi một dữ liệu mẫu tới Server, mà sẽ được xử lý bởi một process.cgi và cuối cùng một phản hồi sẽ được trả lại:

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: 88
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```



Bên Server, script process.cgi xử lý dữ liệu đã truyền và gửi phản hồi như sau:

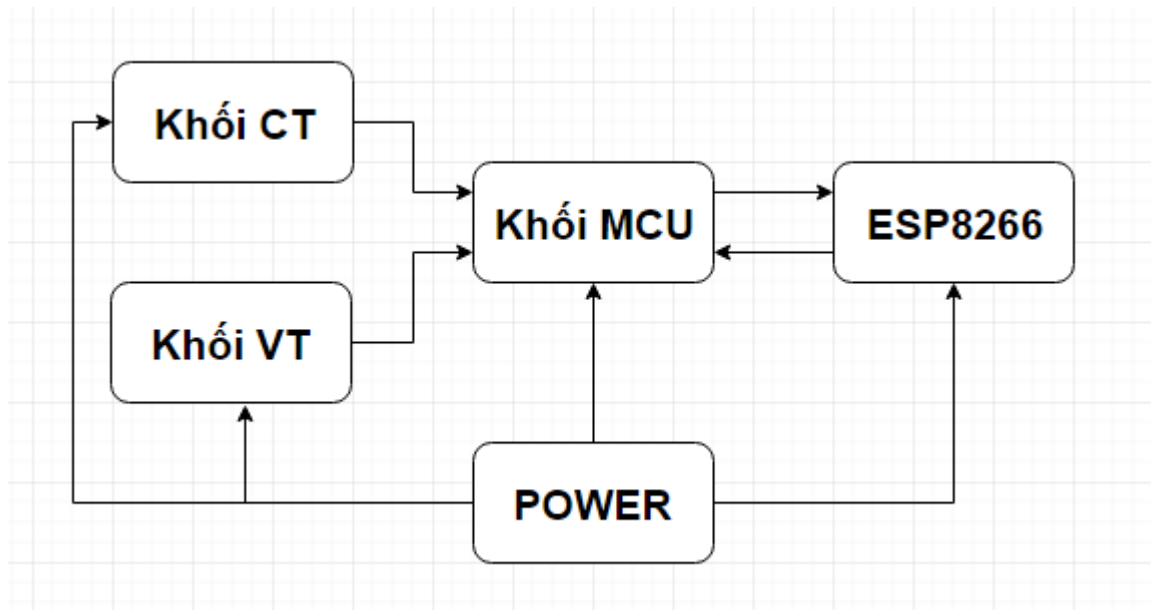
```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization, Accept
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text/html
Connection: Closed
<html>
<body>
<h1>Request Processed Successfully</h1>
</body>
</html>
```

### 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

#### 3.1 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.
- Tính tương tác người dùng cao
- Giá thành rẻ để có thể phổ biến rộng rãi



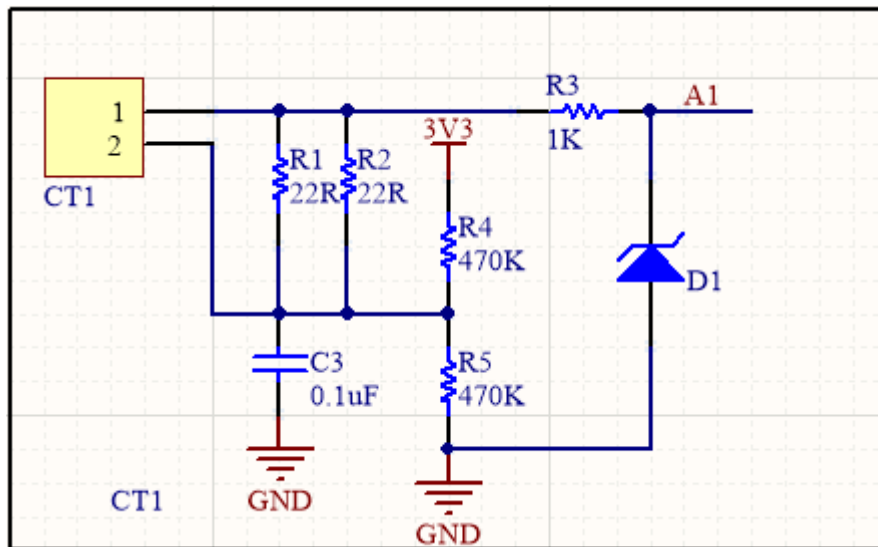
Hình 3.1 Sơ đồ khối hệ thống phần cứng

Phân tích: hệ thống gồm bốn phần chính:

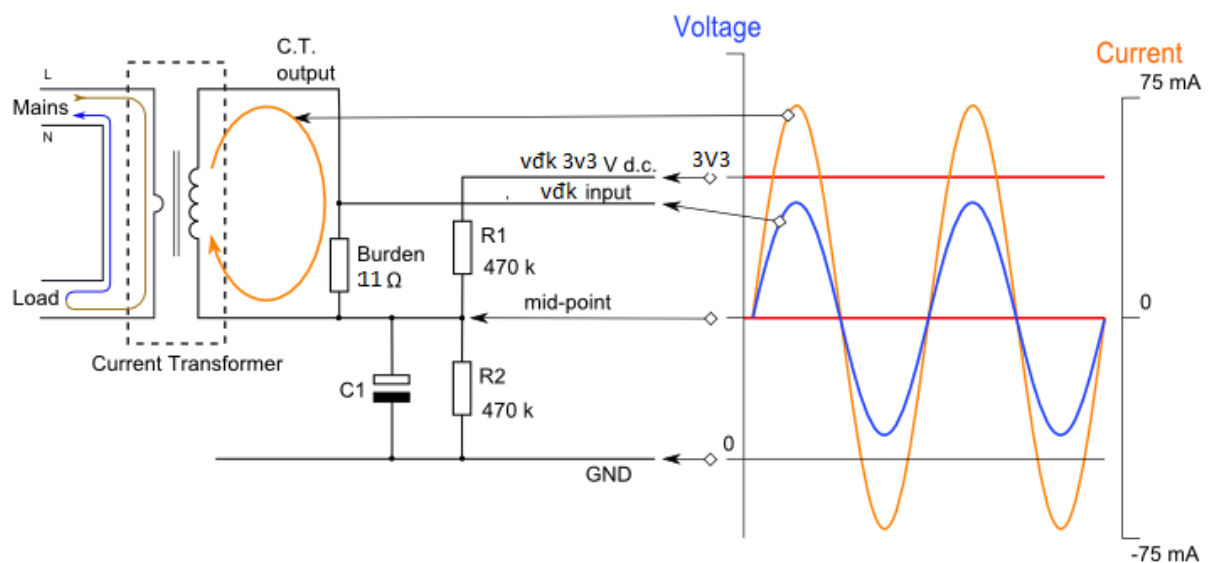
- **Khối CT:** khối này có nhiệm vụ cách ly và chuyển tín hiệu dòng điện thực sang tín hiệu vi điều khiển
- **Khối VT:** khối này có nhiệm vụ cách ly và chuyển tín hiệu điện áp thực sang tín hiệu vi điều khiển đọc được.
- **Khối MCU Vi điều khiển:** đọc input, xử lý, tính toán và truyền dữ liệu cho ESP8266
- **Khối ESP8266:** là khối trung gian, để trung chuyển data của MCU đến server
- **Khối Power:** cung cấp cung cấp cho toàn bộ các khối

### 3.2 Sơ đồ mạch chi tiết

#### 3.2.1 khối CT



Hình 3.2 Schematic khối CT



Hình 3.3 Nguyên lí khối CT

Current transformer output : 100mA rms

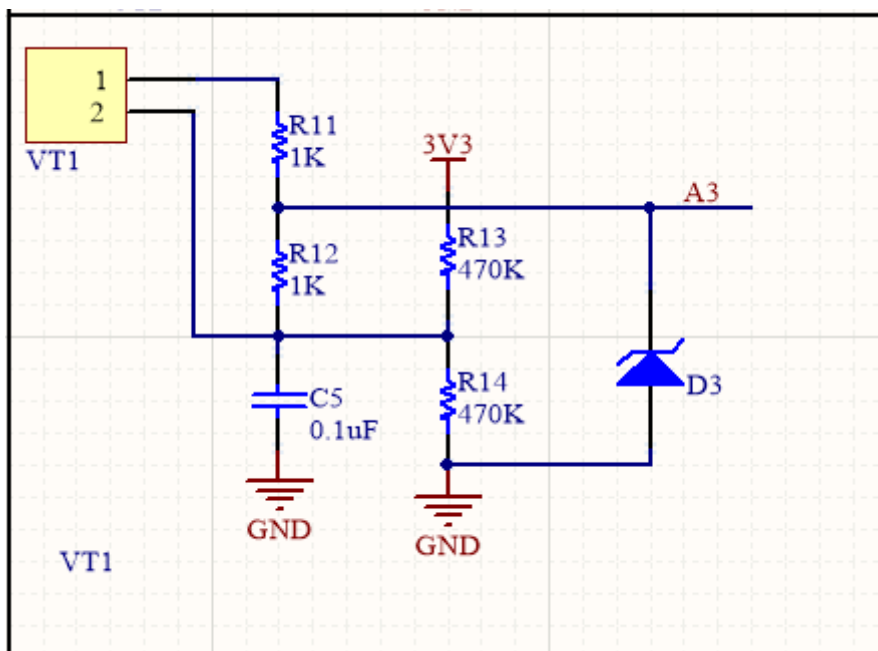
= 141.4 mA peak

= 282.8 mA peak- peak

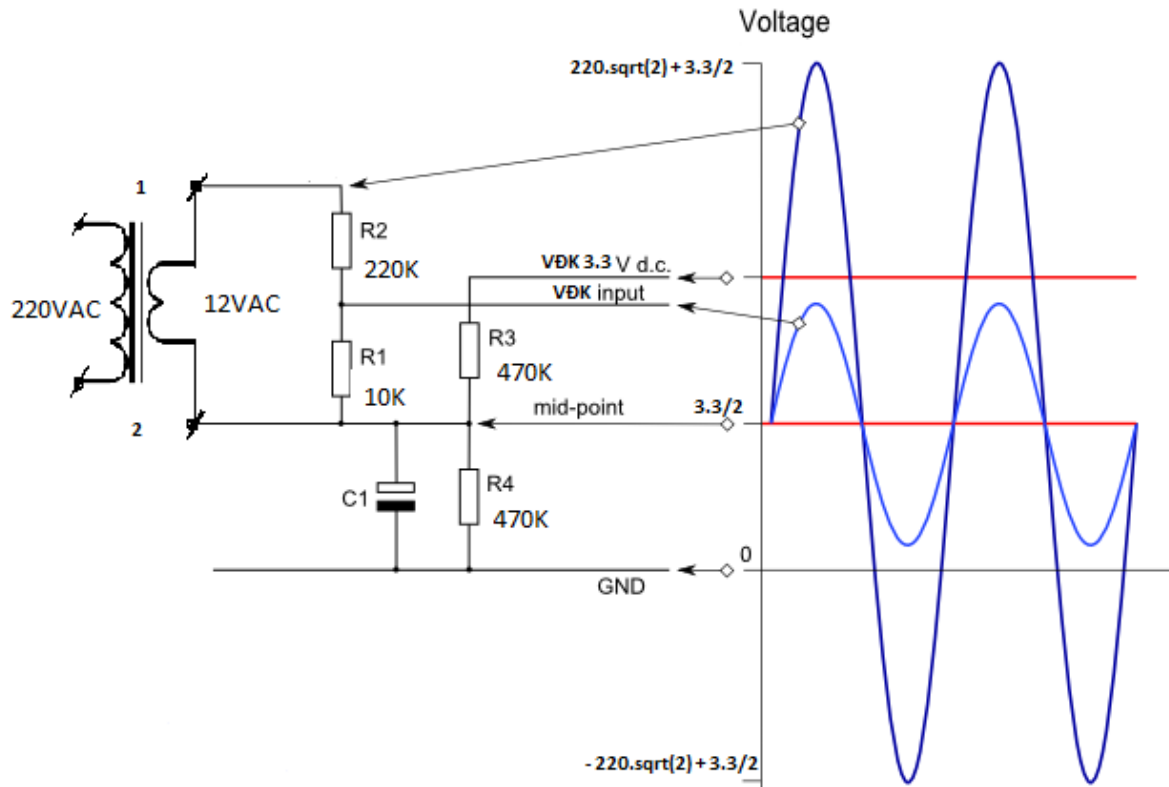
Voltage ở vdk input:

$$282.8 \times 11\Omega = 3.11 \text{ V peak-peak to vdk}$$

### 3.2.2 khối VT



Hình 3.4 Schematic khối VT



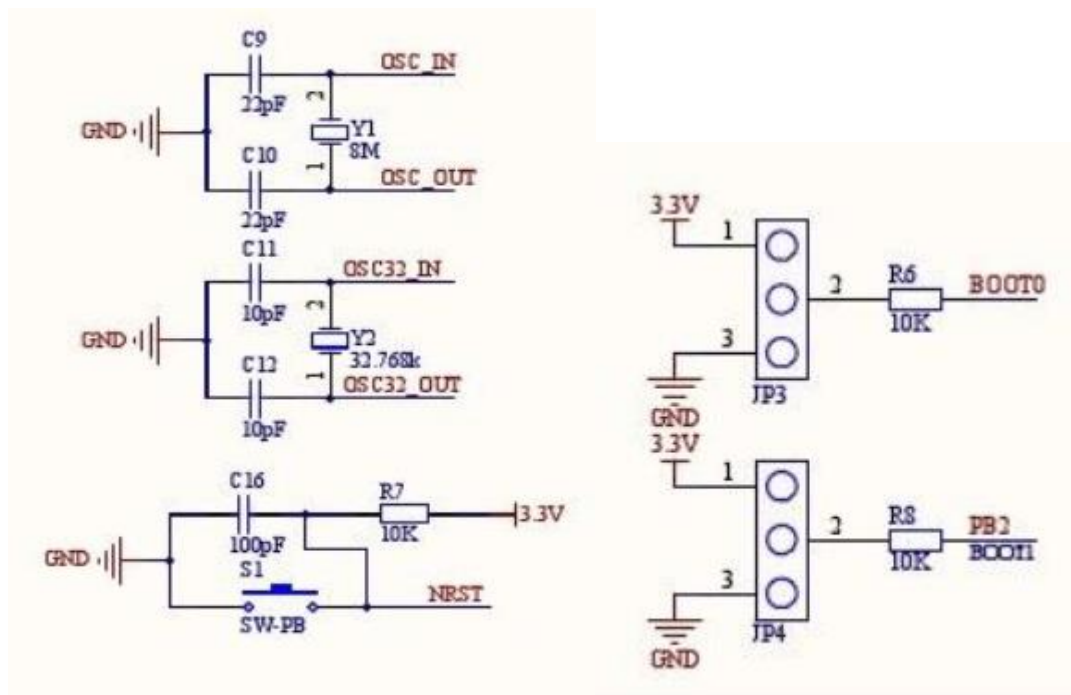
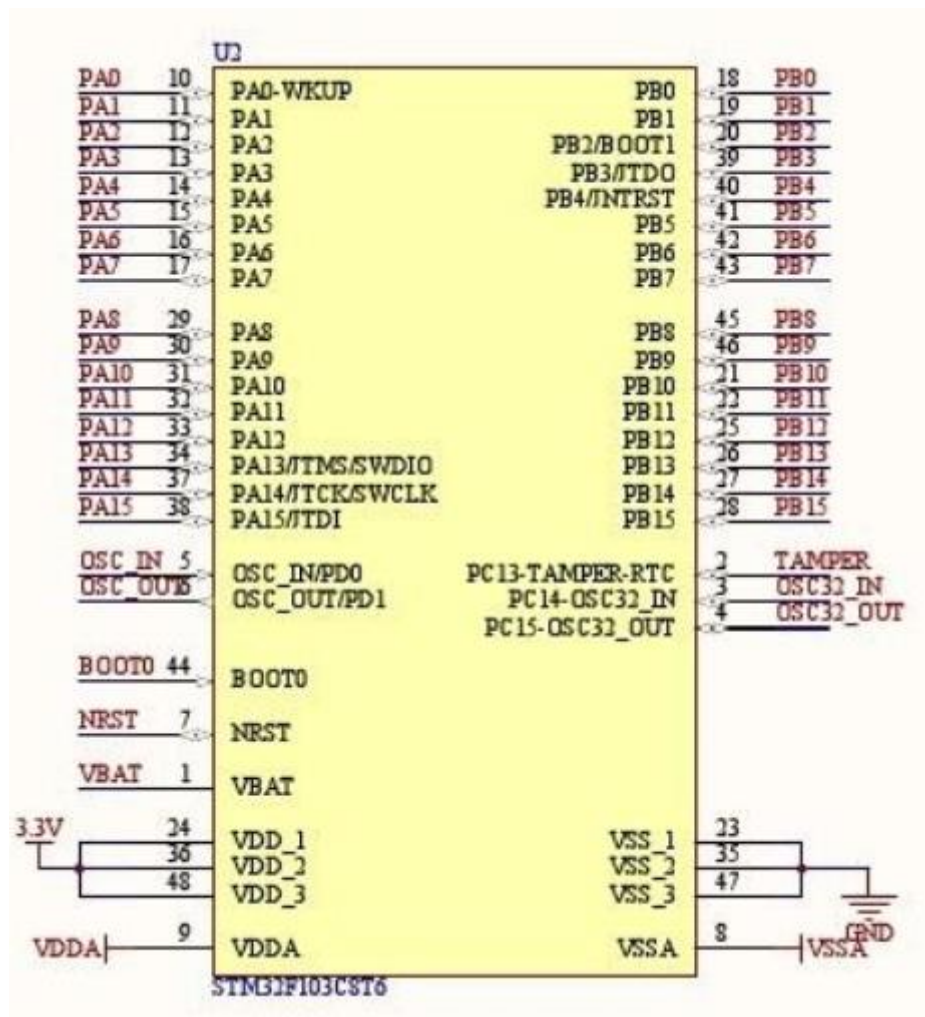
Hình 3.5 Nguyên lí khối VT

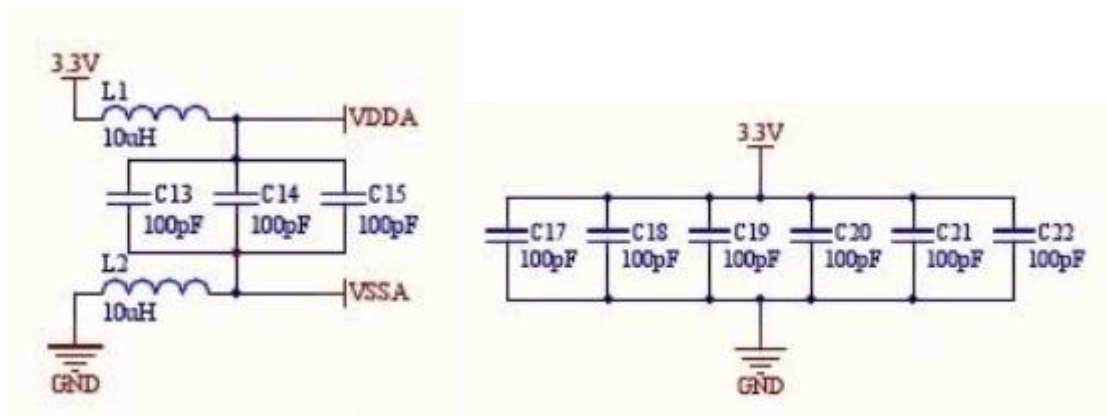
Nhiệm vụ của khối VT là chuyển tín hiệu 220VAC đầu vào xuống tín hiệu để vi điều khiển đọc được. Input 220VAC sẽ qua biến áp, hạ xuống 12VAC. Điện trở R1, R2 tạo thành cầu phân áp, làm giảm điện áp 12VAC. Điện trở R3, R4 cung cấp độ lệch điện áp. Tụ điện C1 cung cấp 1 trở kháng thấp để nối đất cho tín hiệu AC.

R1, R2 cần phải được lựa chọn để cung cấp một điện áp đỉnh đầu ra của ~1V. Đối với biến áp AC với output RMS 12V, kết hợp điện trở 220k cho R2 và 10K cho R1 sẽ phù hợp:

$$\text{peak\_voltage\_output} = R1 / (R1 + R2) \times \text{peak\_voltage\_input} = 10k / (10k + 220k) \times 16.97V = 0.738V$$

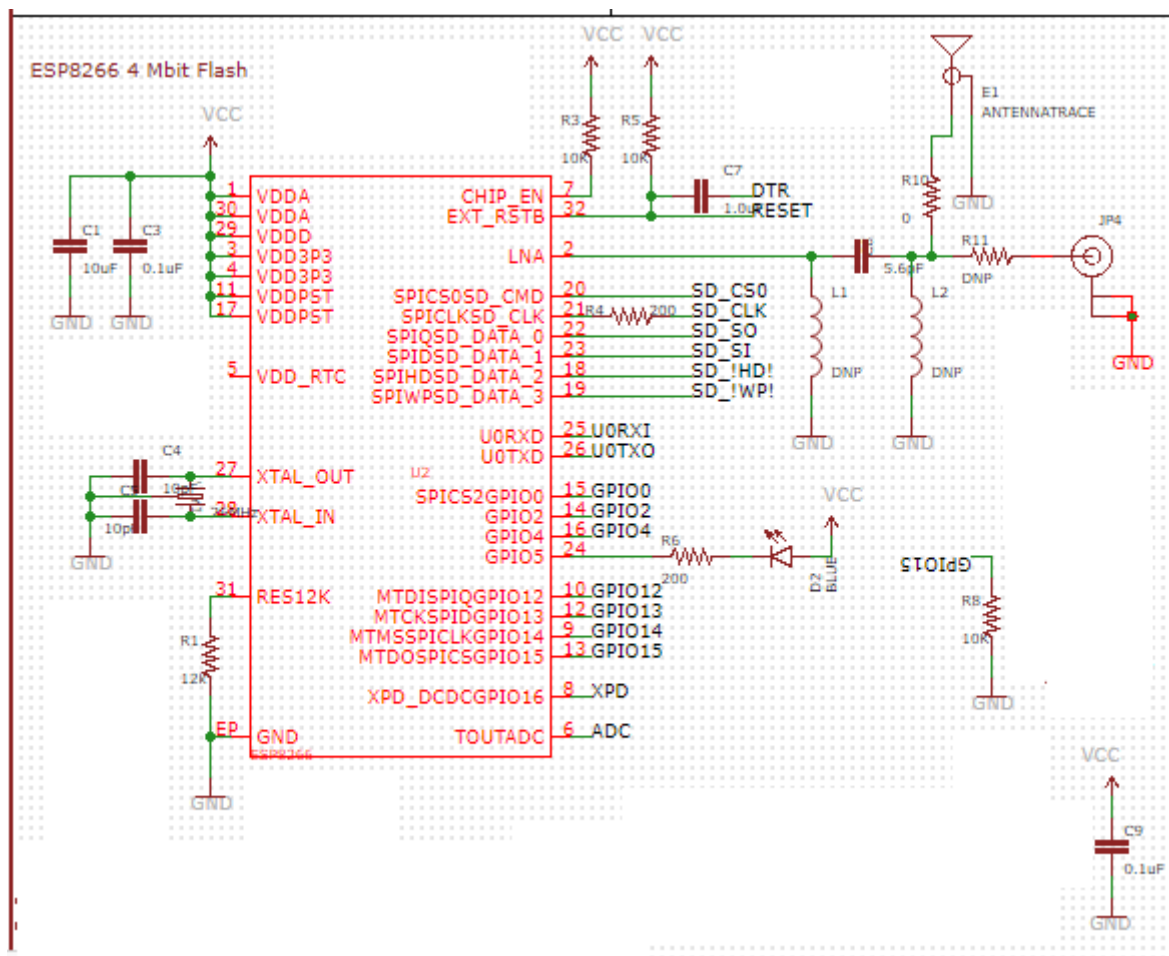
### 3.2.3 khối MCU





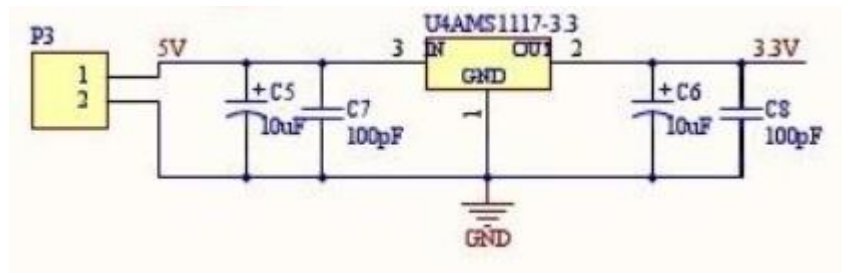
Khối MCU sử dụng vi điều khiển stm32f103c8t6, hoạt động ở thạch anh 8Mhz. MCU đọc ADC từ khối VT và CT sau đó xử lý tính toán để truyền uart cho esp8266

### 3.2.4 Khối ESP8266



Hình 3.6 Schematic khối ESP8266

### 3.2.5 khối POWER



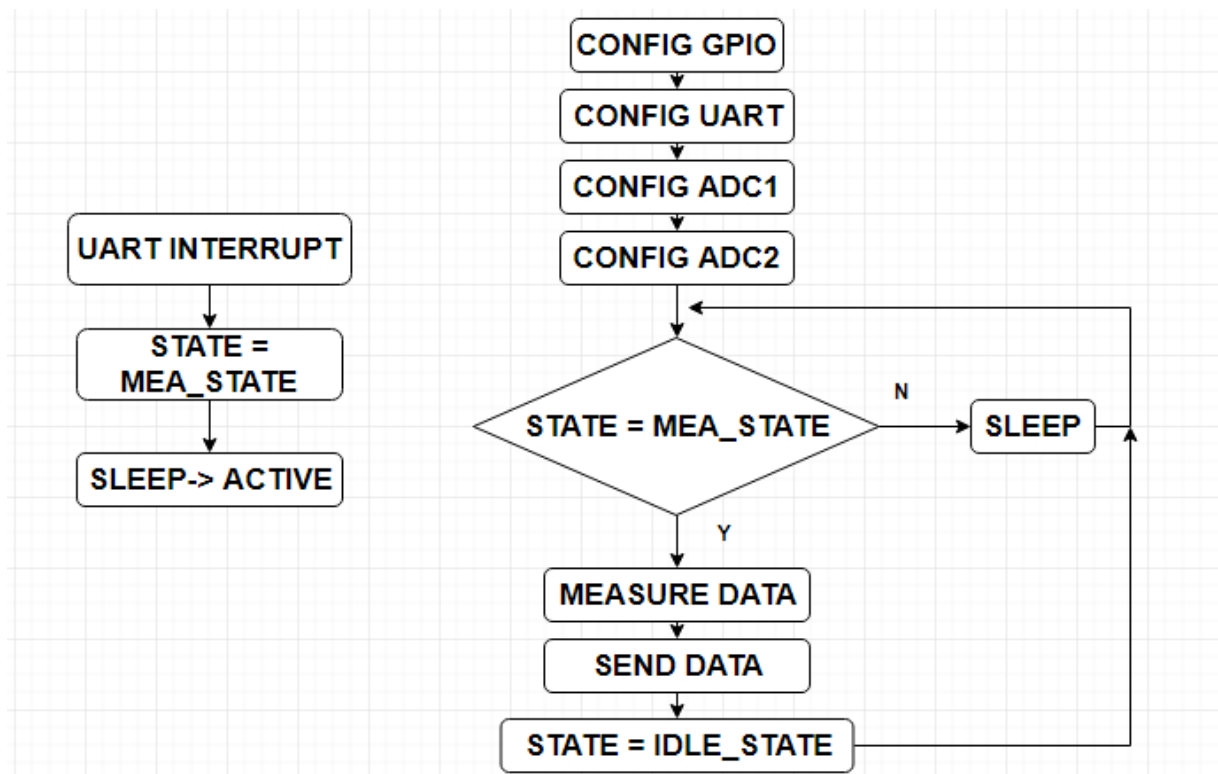
Hình 3.7 Schematic khối Power

Khối Power nhận nguồn 5V từ Adapter, giảm xuống 3V3 để nuôi vi điều khiển, ESP8266, CT và VT.

## 4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

### 4.1 Giải thuật vi điều khiển

#### 4.1.1 Giải thuật tổng quát



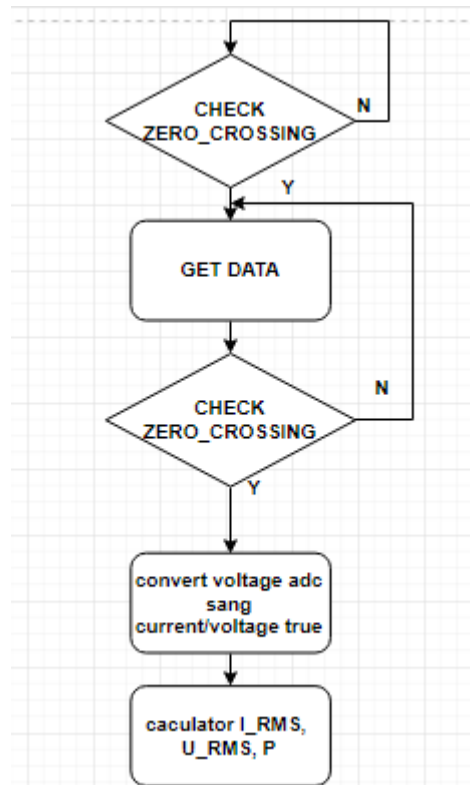
Hình 4.1 Giải thuật tổng quát của vi điều khiển

Sau khi config xong các chức năng cần thiết, vi điều khiển sẽ vào trạng thái ngủ để tiết kiệm năng lượng, khi có yêu cầu data từ ESP8266 (ngắt UART) thì vi điều khiển sẽ



vào chương trình ngắt, đổi biến trạng thái, chuyển chế độ sleep sang active và thoát ra khỏi chương ngắt, trở lại chương trình chính. Tiếp theo ở chương trình chính, vi điều khiển sẽ vào trạng thái đo dữ liệu, đọc input từ khối CT và VT, tính toán, chuyển sang dữ liệu U, I, P cần thiết của hệ thống rồi gửi dữ liệu cho ESP8266 qua UART.

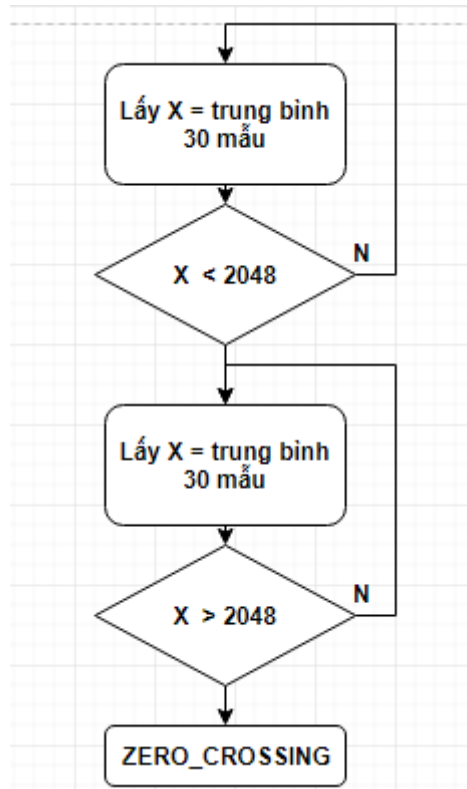
#### 4.1.2 Giải thuật measure\_data



Hình 4.2 Giải thuật measure\_data

Ban đầu vi điều khiển sẽ check zerocrossing, sau khi phát hiện zerocrossing, vi điều khiển sẽ bắt đầu lấy data từ CT, VT lưu vào mảng, nếu phát hiện zerocrossing thì vi điều khiển sẽ dừng việc lấy data. Sau đó, vi điều khiển sẽ chuyển dữ liệu đọc ADC sang dữ liệu dòng và áp thật, rồi tính toán các giá trị hiệu dụng, công suất.

#### 4.1.2.1 Giải thuật detect zero\_crossing



Hình 4.3 Giải thuật detect zero\_crossing

Ta lấy x bằng trung bình của 30 mẫu ADC để giảm nhiễu, (tốc độ lấy mẫu 0.1 us/mẫu)

#### 4.1.2.2 Giải thuật tính toán RMS:

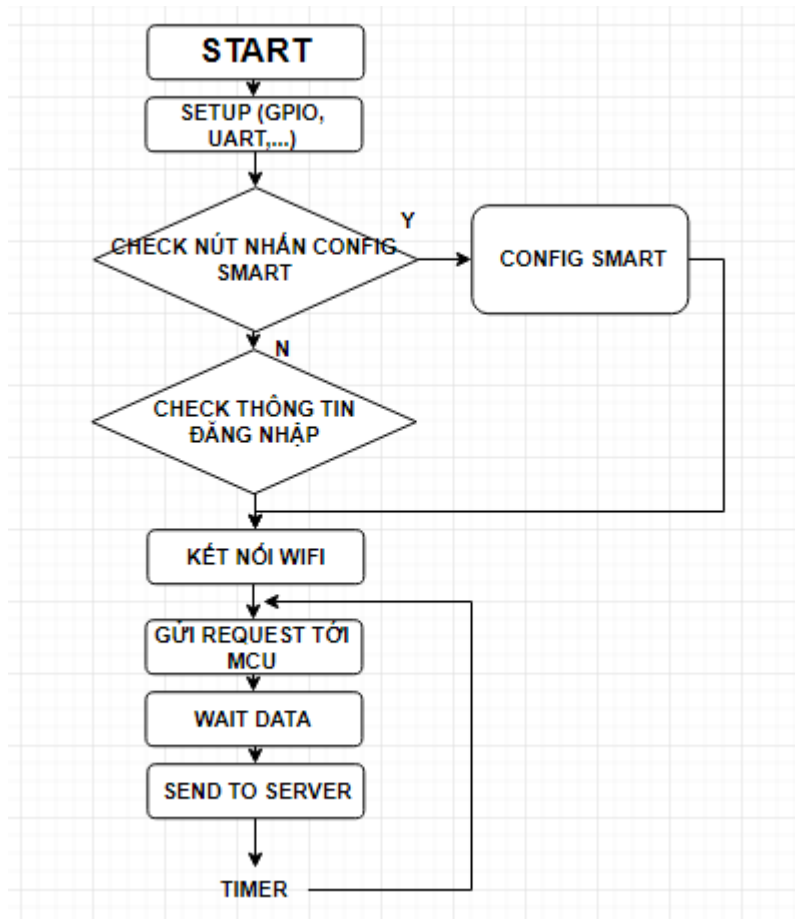
$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

$$U_{rms} = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt}$$

$$P_d = \frac{1}{T} \int_0^T u(t) i(t) dt$$

Sau khi lấy được mảng giá trị tức thời của dòng, áp trong 1 chu kì, ta sử dụng công thức trên để tính toán giá trị RMS

## 4.2 Giải thuật ESP8266



Hình 4.4 Giải thuật chương trình ESP8266

Ban đầu ESP sẽ setup các chức năng cần thiết, như gpio, uart. Sau đó, esp sẽ check nút nhấn xem người dùng có muốn config lại thông tin đăng nhập không, nếu có thì nó sẽ vào chương trình config smart, tại đây, nó sẽ nhận thông tin wifi, password từ app android của người dùng thông qua giao thức UDP, nếu không nhấn nút thì nó sẽ bắt đầu check thông tin đăng nhập, sau khi check thông tin đăng nhập xong, esp sẽ kết nối wifi, gửi request đọc data tới MCU, chờ MCU trả lời, rồi gửi nó cho server.

## 4.3 Xây dựng Webserver và Website

Để có một Website thì ta cần phải có

- Host
- Các file chứa mã nguồn (HTML, CSS, PHP ..)

- Cơ sở dữ liệu - Database

Khi truy cập tới tên miền, Host sẽ chạy các file chứa mã nguồn tương ứng, các file này truy xuất dữ liệu từ cơ sở dữ liệu để tương tác với người sử dụng.

Quy trình thực hiện phân lập trình web:

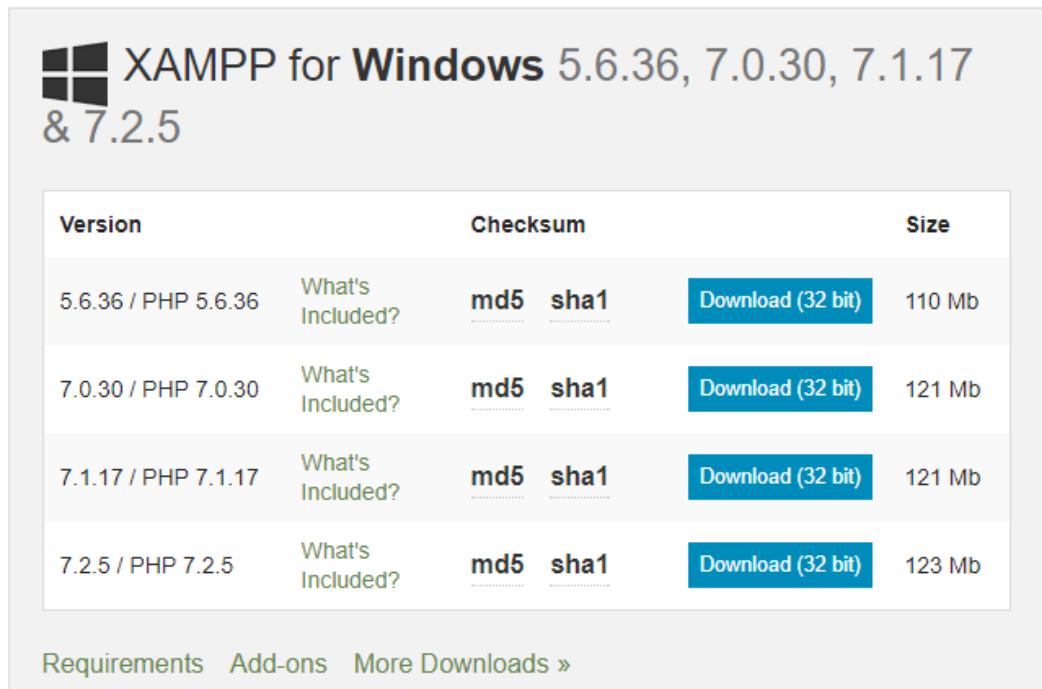
- Tạo máy chủ để chạy các file chứa mã nguồn
- Lập trình, test các file chứa mã nguồn
- Đăng kí tên miền, để trỏ tên miền về ip của máy chủ web

#### 4.3.1 Xamp trên laptop/pc

Tài nguyên và sức mạnh của máy tính nhúng Raspberry còn hạn chế, nên ta không thể cài đặt các Editor để thuận tiện cho việc build và develop code server. Vì vậy để thuận tiện, ta sẽ build và phát triển code trên Laptop/PC trước, sau khi hoàn thiện thì mới chuyển qua máy tính nhúng. Ta sẽ sử dụng XAMPP để tạo máy chủ web cho Laptop/PC

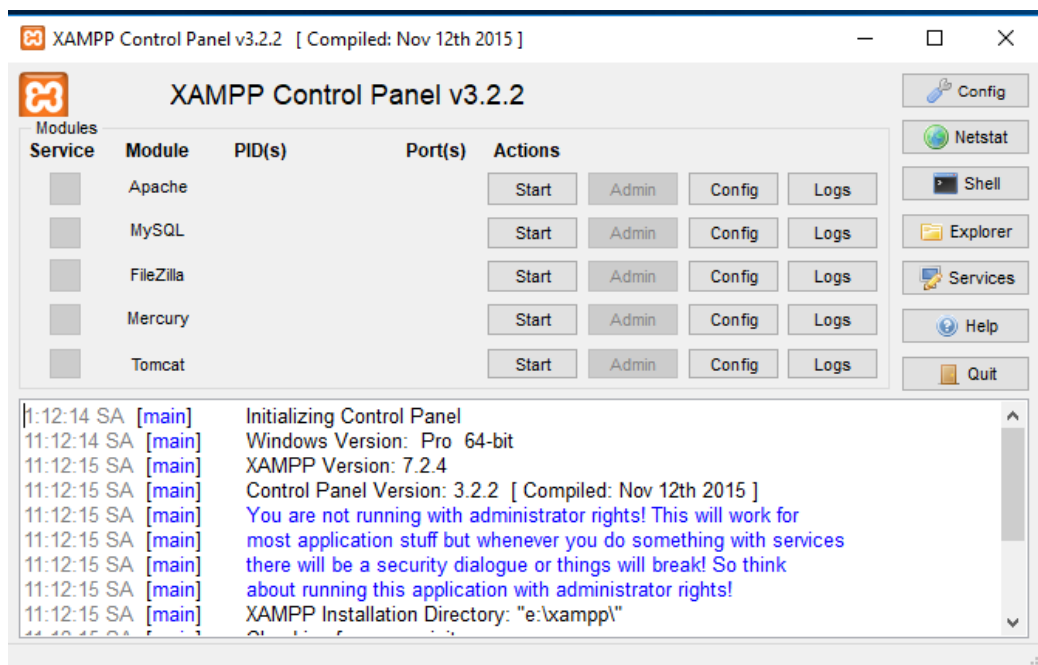
Xampp (X + Apache + Mysql + PHP + Perl) là chương trình tạo máy chủ Web (Web Server) cho được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

Truy cập vào link <https://www.apachefriends.org/download.html> để download phần mềm Xampp



Hình 4.5 Trang chủ của web tải phần mềm localhost-Xampp

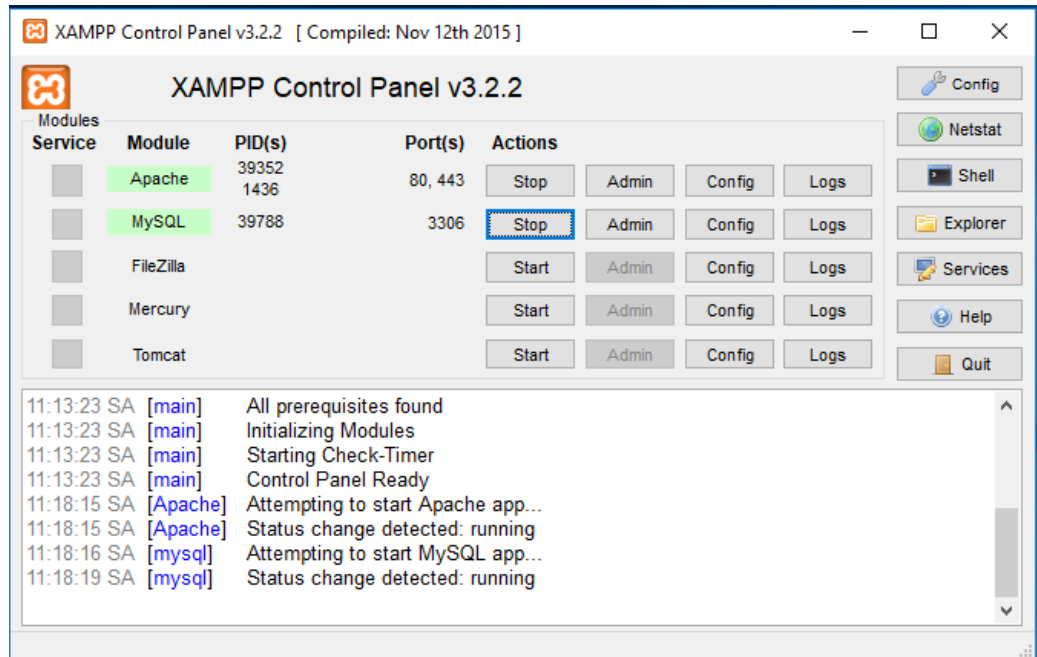
Sau khi cài đặt xong phần mềm, khi khởi động bảng điều khiển của phần mềm, ta sẽ thấy giao diện



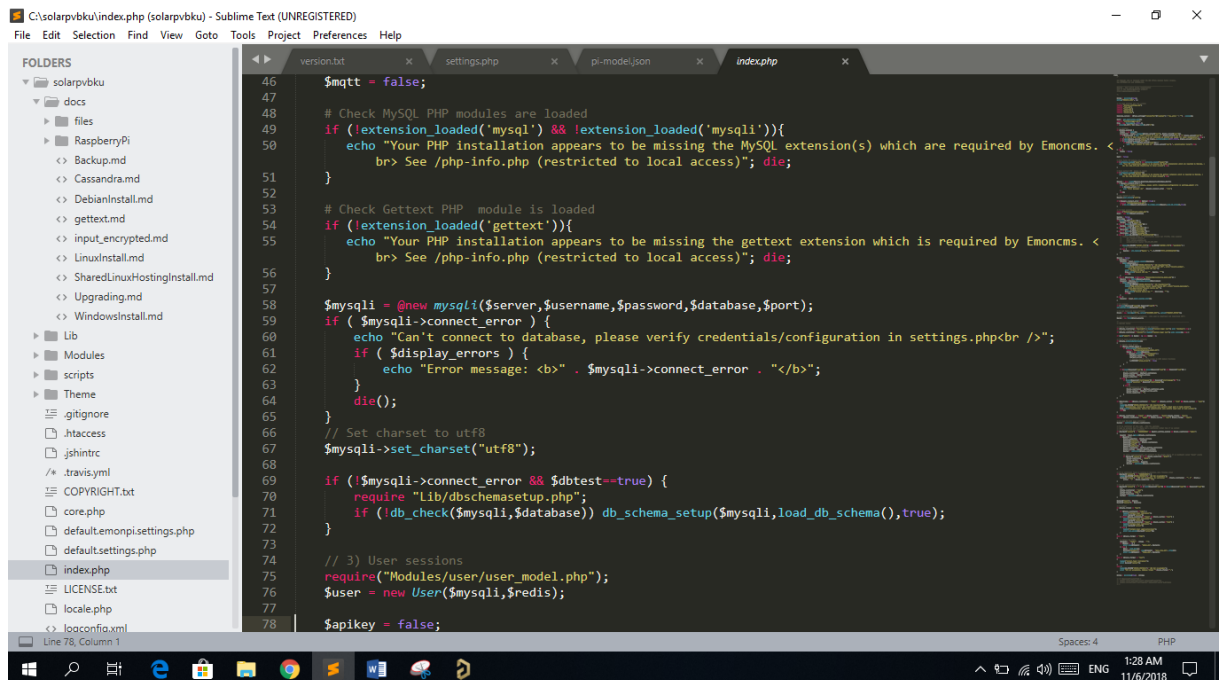
Hình 4.6 Bảng điều khiển khi khởi động Xampp

Để chạy được Localhost thì ta phải khởi động hai ứng dụng Webserver Apache, và MySQL Server bằng cách nhấn nút Start tương ứng. Nếu cả hai ứng dụng chuyển

sang màu xanh nghĩa là đã khởi động hoàn tất.



Hình 4.7 Giao diện Xampp khi khởi động hoàn tất các ứng dụng Apache và MySQL  
Để lập trình server, ta sử dụng Editor sublime Text, dưới đây giao diện của nó:

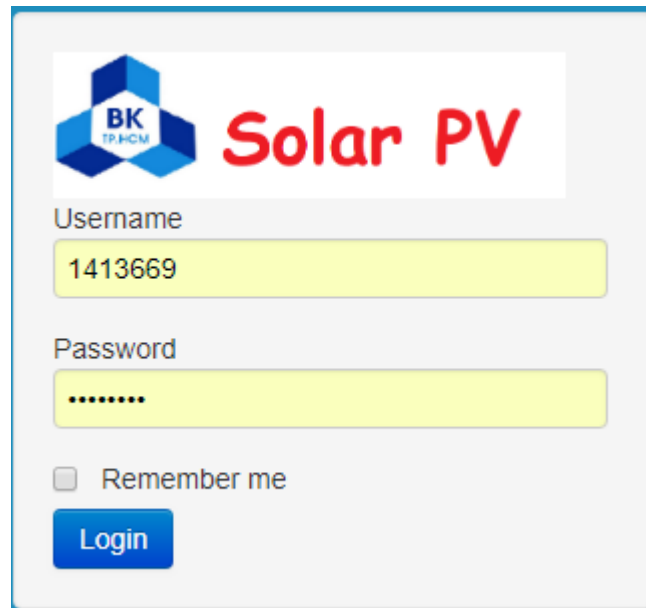


Hình 4.8 Giao diện của Editor Sublime Text

#### 4.3.2 Xây dựng web

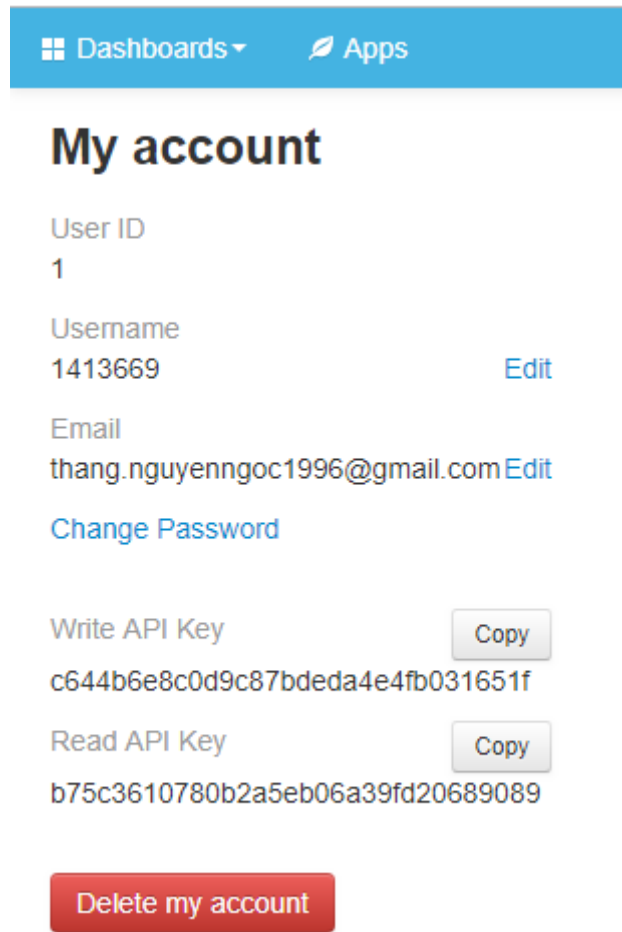
Để có thể phát triển tốt dự án một cách nhanh chóng, ta sẽ phát triển từ các open source. Thuật ngữ Open source (mã nguồn mở) đề cập đến một thứ gì đó mà mọi người có thể thay đổi và chia sẻ bởi vì thiết kế của nó mang tính chất cộng đồng. Luận văn cũng được phát triển từ mã nguồn của <https://github.com/emoncms>

##### 4.3.2.1 User



Hình 4.9 Giao diện trang đăng nhập

Để đảm bảo bảo mật thông tin, ta cần tạo tài khoản, ứng với mỗi user, server sẽ cung cấp 1 key để có thể get dữ liệu từ hệ thống đo đạc đến server.



Hình 4.10 Thông tin Account của 1 user

#### 4.3.2.2 Get dữ liệu

Server sẽ nhận dữ liệu thông qua giao thức HTTP GET,

Đối với dữ liệu vào số lượng nhỏ, cấu trúc của URL như sau:

[http://homeofme.tk/solarpvbku/input/post?node=hethongdodac2&fulljson={\"power1\":100,\"power2\":200,\"power3\":300}&apikey=c644b6e8c0d9c87bdeda4e4fb031651f](http://homeofme.tk/solarpvbku/input/post?node=hethongdodac2&fulljson={\)

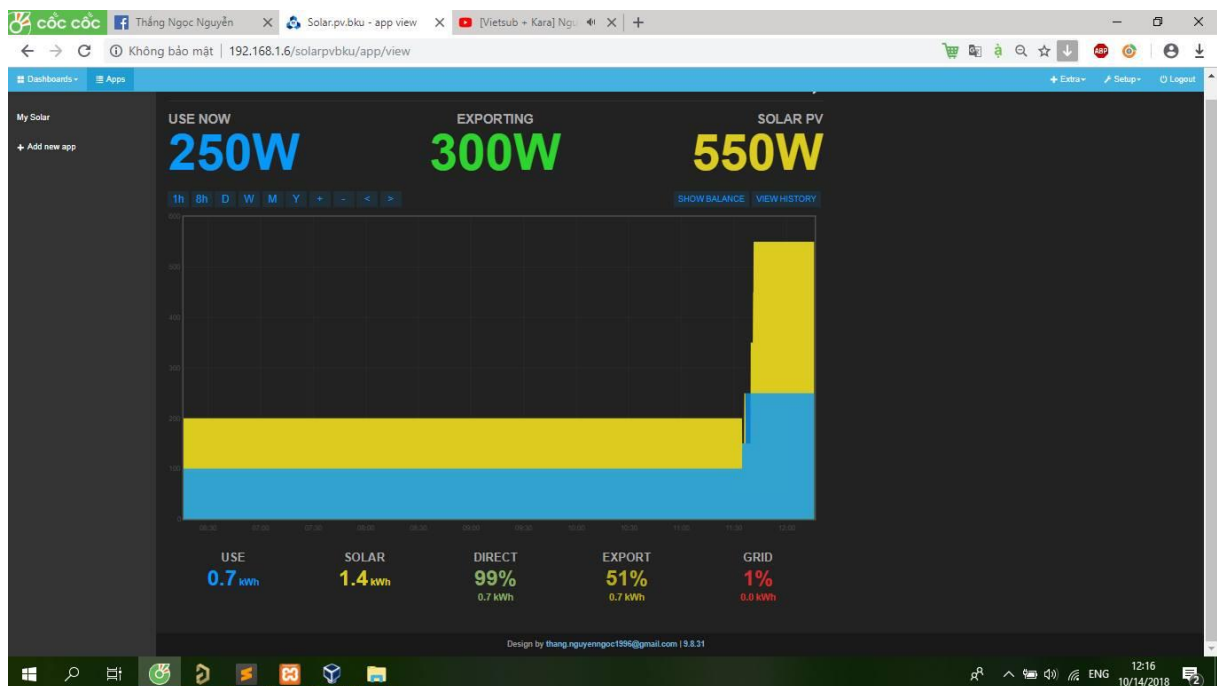
“apikey=c644b6e8c0d9c87bdeda4e4fb031651f” chính là key cho mỗi user

Đối với dữ liệu đầu vào số lượng lớn, cấu trúc URL như sau

[http://homeofme.tk/solarpvbku/input/bulk?data=\[\[0,16,1137\],\[2,17,1437,3164\],\[4,19,1412,3077\]\]&apikey=c644b6e8c0d9c87bdeda4e4fb031651f](http://homeofme.tk/solarpvbku/input/bulk?data=[[0,16,1137],[2,17,1437,3164],[4,19,1412,3077]]&apikey=c644b6e8c0d9c87bdeda4e4fb031651f)



### 4.3.2.3 Hiện thị Chart



Hình 4.11 Chart hiển thị thông số hệ thống

## 5. KẾT QUẢ THỰC HIỆN

## 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 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 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 dễ dàng giám sát.

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/app

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

- Giám sát một chiều từ thiết bị gửi về server
- Mới chỉ thiết kế mô phỏng với hệ thống lưới điện 1 pha, công suất tải tiêu thụ và công suất phát của pin mặt trời nhỏ.
- Mới chỉ thiết kế mô phỏng với 1 hệ thống, chưa kết hợp giám sát được với nhiều hệ thống.

### 6.2 Hướng phát triển

- Phát triển hoàn thiện sản phẩm là hệ thống theo dõi với lưới điện 3 pha, công suất lớn như ở xưởng, nhà máy
- Tích hợp khai thác thêm chức năng giám sát nhiều điểm. Ví dụ như ở xưởng có nhiều khu, ở mỗi khu ta lắp pin mặt trời, thì ta phải giám sát được ở mỗi khu đó.

## 7. TÀI LIỆU THAM KHẢO

- [1] <http://openenergymonitor.org>
- [2] <https://solarenergypanels.in/solar-power-plants>
- [3] Datasheet STM32F103x6, STM32F103x8, STM32F103xB, [www.st.com](http://www.st.com),  
<https://www.keil.com>
- [4] <https://arduino.esp8266.vn/>
- [5] <https://github.com/emoncms>
- [6] <https://www.raspberrypi.org>
- [7] <https://raspberrypi.vn>
- [8] [https://vietjack.com/http/http\\_methods.jsp](https://vietjack.com/http/http_methods.jsp)

## **8. PHỤ LỤC**

### **8.1 Chương trình ở vi điều khiển:**