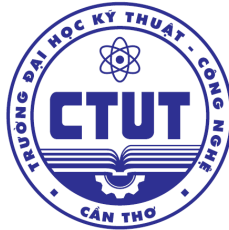


**TRƯỜNG ĐẠI HỌC
KỸ THUẬT – CÔNG NGHỆ CẦN THƠ
KHOA ĐIỆN – ĐIỆN TỬ – VIỄN THÔNG**



ĐỒ ÁN HỆ THỐNG ĐIỆN

**ĐỀ TÀI
THEO DÕI ĐIỆN NĂNG TIÊU THỤ CỦA PHỤ TẢI
MỘT PHA VỚI CẢM BIẾN DÒNG ACS712**

Giảng viên hướng dẫn: Huỳnh Nguyễn Xuân Cần

Lớp: Công nghệ kỹ thuật điện, điện tử 2014-2

Nhóm sinh viên thực hiện:

1. Huỳnh Như (MSSV: 1400012)
2. Nguyễn Thị Anh Thư (MSSV: 1400450)

Cần Thơ, Năm 2017

Lời cảm ơn

Trong thời gian làm đồ án, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô và bạn bè.

Chúng em chân thành cảm ơn thầy đã tận tình hướng dẫn nhóm chúng em trong quá trình hoàn thành đồ án môn học. Cảm ơn thầy đã tạo điều kiện cho chúng em được thực hiện đồ án trên các board mạch thực tế để ứng dụng các kiến thức đã học vào thực hành.

Nhóm sinh viên thực hiện,

Huỳnh Như

Nguyễn Thị Anh Thư

Mục lục

CHƯƠNG 1	CƠ SỞ LÝ THUYẾT	1
1.1	Module ESP8266 Wemos D1	1
1.1.1	Giới thiệu	1
1.1.2	Lập trình với module ESP8266 Wemos D1	1
1.2	Công cụ và ngôn ngữ lập trình cho Module ESP8266	3
1.2.1	Cài đặt môi trường lập trình Arduino IDE	3
1.2.2	Cài đặt môi trường lập trình cho ESP8266 trên Arduino IDE	3
1.3	Giao thức SmartConfig kết nối WiFi cho Module ESP8266	4
1.3.1	Giới thiệu giao thức SmartConfig	4
1.3.2	Một số ứng dụng trên điện thoại để thực hiện SmartConfig	5
1.3.3	Sử dụng App ESP Touch trên iOS thực hiện SmartConfig cho ESP8266	5
1.4	Ứng dụng Blynk trên điện thoại	6
1.4.1	Giới thiệu	6
1.4.2	Cài đặt App Blynk và thư viện Bkynk cho lập trình	6
1.4.3	Tạo project trong App Blynk	6
1.4.4	Kết nối giữa các phần tử trên App Blynk và vi điều khiển	7
1.5	Ứng dụng Web Server Thingspeak	8
1.5.1	Giới thiệu	8
1.5.2	Cài đặt thư viện Thingspeak cho lập trình	8
1.5.3	Tạo tài khoản sử dụng và tạo project trên Thingspeak	8
1.5.4	Cách giao tiếp vi điều khiển và Web Server Thingspeak	9
1.6	Cảm biến dòng điện ACS712	9
1.6.1	Giới thiệu	9
1.6.2	Đặc điểm của cảm biến	9

1.6.3	Cách sử dụng module cảm biến dòng ACS712	10
1.7	Xác định công suất, điện năng tiêu thụ của phụ tải xoay chiều một pha dựa vào dòng điện qua tải và thời gian hoạt động của tải	11
CHƯƠNG 2	THIẾT KẾ MẠCH VÀ GIAO DIỆN ĐIỀU KHIỂN	12
2.1	Sơ đồ khối	12
2.2	Sơ đồ mạch nguyên lý	13
2.3	Mô tả các khối chức năng trên sơ đồ mạch nguyên lý	13
2.4	Giao diện điều khiển trên Blynk	14
2.5	Giao diện thu thập dữ liệu trên Thingspeak	15
CHƯƠNG 3	LƯU ĐỒ GIẢI THUẬT VÀ CHƯƠNG TRÌNH	16
3.1	Lưu đồ giải thuật	16
3.2	Chương trình điều khiển	16
3.3	Kết quả thu được	18
CHƯƠNG 4	KẾT LUẬN	20
4.1	Kết quả	20
4.2	Hạn chế	20
TÀI LIỆU THAM KHẢO		21

CHƯƠNG 1

CƠ SỞ LÝ THUYẾT

1.1 Module ESP8266 Wemos D1

1.1.1 Giới thiệu

Module ESP8266 Wemos D1 là phiên bản được thiết kế với hình dạng gần giống board Arduino Uno với trung tâm là module Wifi SoC ESP8266 được xây dựng lại firmware để có thể chạy được với chương trình Arduino.

ESP8266 Wemos D1 dễ dàng sử dụng và thích hợp với các ứng dụng thu thập dữ liệu và điều khiển qua Wifi.



Hình 1.1: Module ESP8266 Wemos D1

Sử dụng cổng Micro USB kết nối module với máy tính để nạp chương trình cho module. Nguồn cấp cho module ESP8266 Wemos D1 hoạt động là 5VDC.

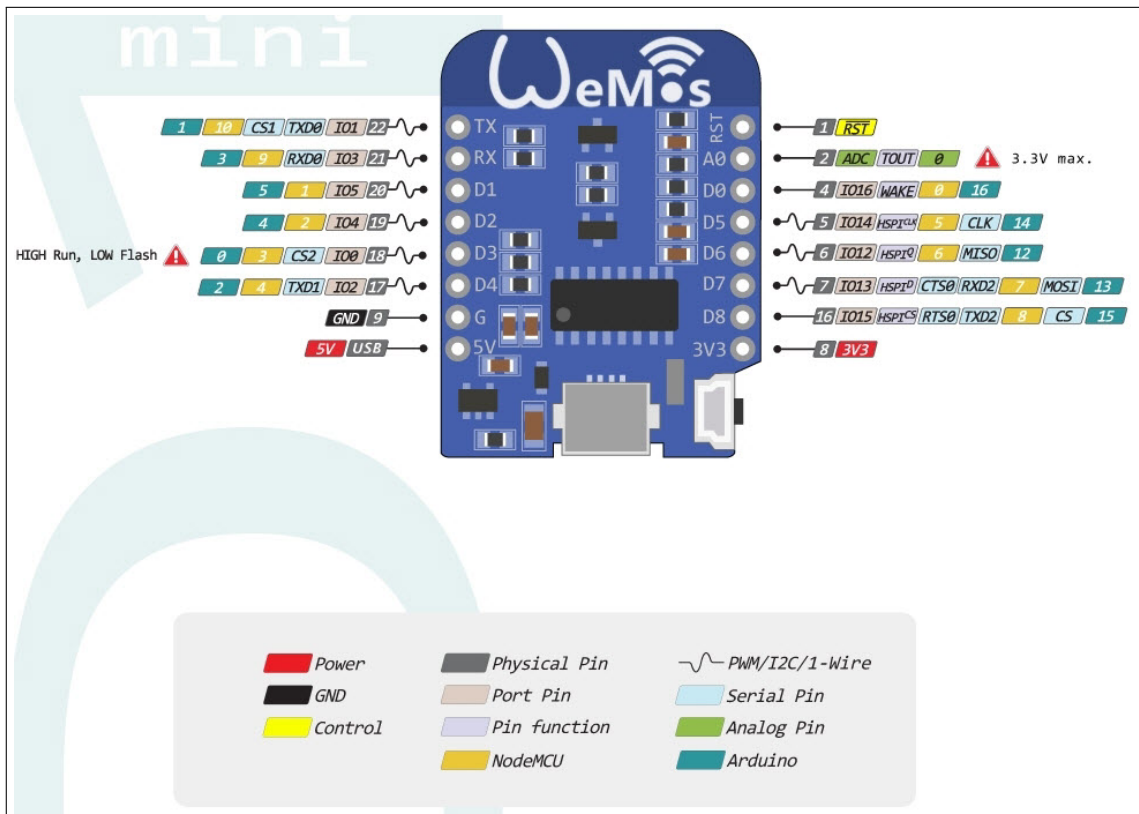
Quan sát bên ngoài phần cứng, module ESP8266 Wemos D1 có 11 chân dùng để xử lý tín hiệu digital và 1 chân dùng để xử lý tín hiệu analog.

Module ESP8266 Wemos D1 cho phép sử dụng trực tiếp phần mềm lập trình Arduino IDE để viết chương trình và nạp chương trình cho nó với một số cài đặt cần thiết trong Arduino IDE.

1.1.2 Lập trình với module ESP8266 Wemos D1

- Phần cứng:
 - Module ESP8266 Wemos D1.
 - Cáp Micro USB dùng để nạp chương trình.

- Nguồn 5VDC cấp cho module ESP8266 Wemos D1 hoạt động.
- Các phần cứng khác tùy vào ứng dụng cụ thể.
- Phần mềm:
 - Phần mềm lập trình Arduino IDE (arduino.cc/en/Main/Software).
 - Driver cài đặt cho cáp Micro USB.
- Gọi tên các chân GPIO khi lập trình:
 - Gọi theo các tên ký hiệu trên module, ví dụ: D0, D1, ... A0 (trên Hình 1.1).
 - Gọi tên chân theo tên GPIO: được ghi trong Hình 1.2.
 - Một số chân thực hiện được đa chức năng như Interrupt/PWM/I2C/1-Wire, ... được mô tả trên Hình 1.2. Với chân analog A0 tín hiệu đưa vào tối đa là 3.3V, module ESP8266 Wemos D1 chỉ hỗ trợ một chân xử lý tín hiệu analog.



Hình 1.2: Chức năng của các chân trên module ESP8266 Wemos D1 (Mini)

1.2 Công cụ và ngôn ngữ lập trình cho Module ESP8266

- Công cụ lập trình: sử dụng Arduino IDE với cấu hình cần thiết.
- Ngôn ngữ lập trình: sử dụng ngôn ngữ lập trình Arduino.

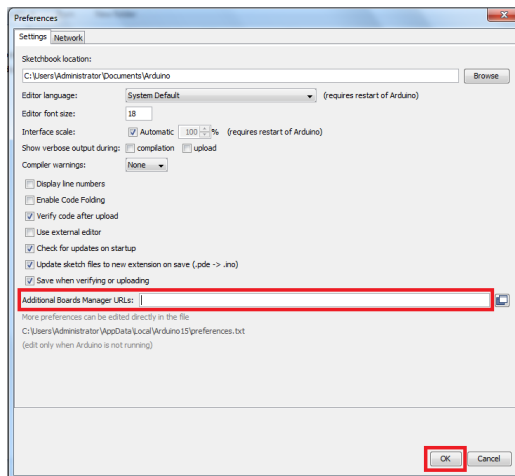
1.2.1 Cài đặt môi trường lập trình Arduino IDE

Truy cập vào địa chỉ <https://www.arduino.cc/en/Main/Software> để tải phần mềm về sử dụng trên máy tính. Sau khi lựa chọn phiên bản cài đặt phù hợp, chọn JUST DOWNLOAD để tiến hành tải về máy.

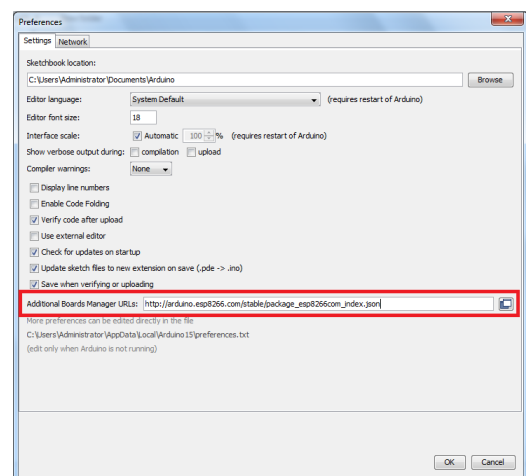
1.2.2 Cài đặt môi trường lập trình cho ESP8266 trên Arduino IDE

Để sử dụng môi trường lập trình Arduino IDE lập trình cho module ESP8266, cần thiết lập theo các bước sau:

- Mở Arduino IDE, chọn File/Preferences được giao diện như Hình 1.3a.



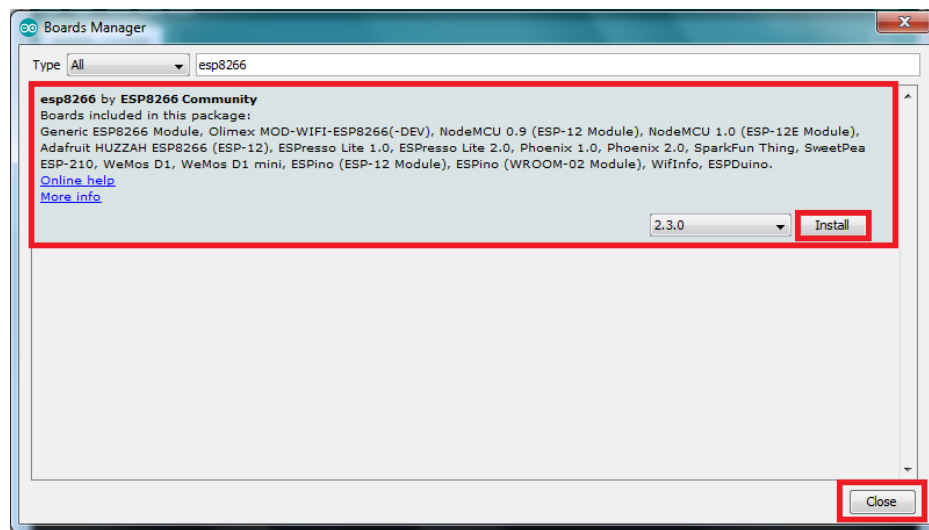
(a) Tab Preferences



(b) Điền URLs vào Additional Boards Manager URLs

Hình 1.3: Arduino IDE Preferences

- Điền địa chỉ sau vào khung Additional Boards Manager URLs (Hình 1.3b) và chọn OK: http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Chọn Tool/Board/Boards Manager... Nhập từ khóa esp8266 vào khung tìm kiếm, chọn ESP8266 by ESP8266 Community và nhấn Install để cài đặt. Cài đặt hoàn thành chọn Close (Hình 1.4).



Hình 1.4: Arduino IDE Boards Manager – ESP8266

- Chọn Board để nạp chương trình cho ESP8266. Vào Tools, thiết lập các tùy chọn như sau:
 - Board: "WeMos D1 (Retried)".
 - CPU Frequency: "80 MHz".
 - Flash Size: "4M (3M SPIFFS)".
 - Upload Speed: "115200".
 - Port: "COM5" (thay đổi tên cổng COM cho phù hợp với từng máy tính).
 - Programmer AVRISP mkII.

1.3 Giao thức SmartConfig kết nối WiFi cho Module ESP8266

1.3.1 Giới thiệu giao thức SmartConfig

SmartConfig là một khái niệm được nhắc đến khi muốn cấu hình thông tin cho thiết bị WiFi kết nối nhanh chóng đến Internet nhất từ người dùng bằng chính thiết bị (điện thoại) của họ.

SmartConfig có thể hiểu là chúng ta gửi thông tin mạng WiFi (bao gồm tên WiFi và password WiFi) cho ESP thông qua smartphone thay cho cách thông thường là phải khai báo thông tin này trong chương trình và nạp firmware xuống.

SmartConfig có một số lợi ích sau:

- Dễ dàng cấu hình WiFi cho ESP8266 thông qua smartphone.

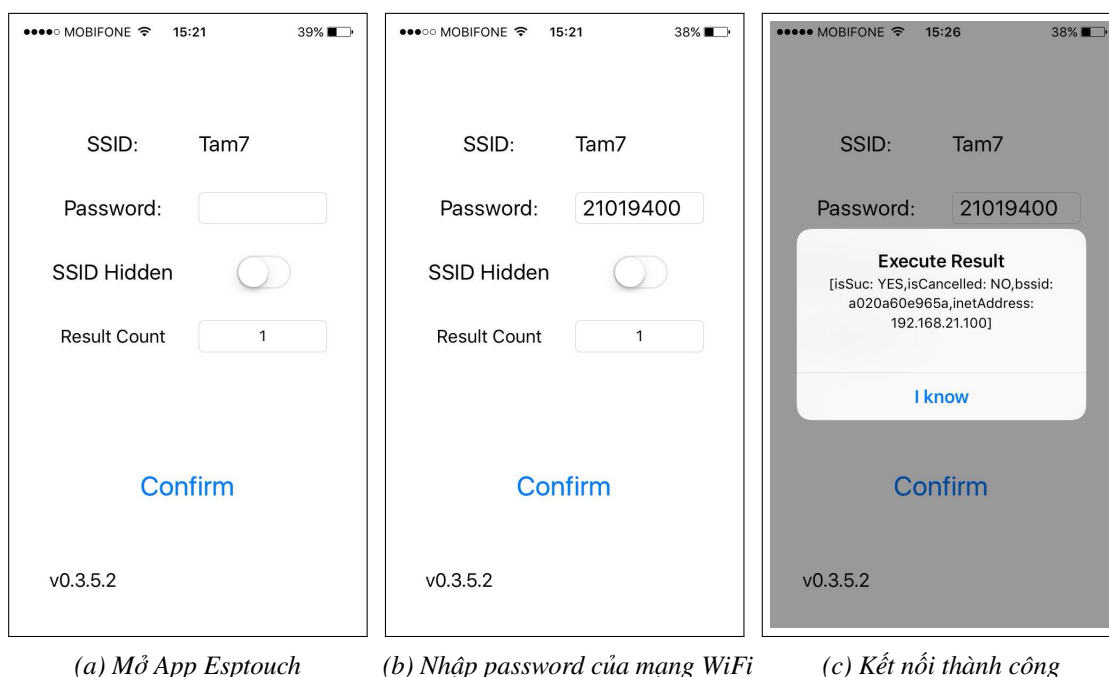
- Không cần phải nạp lại code để cấu hình.
- Có thể dùng SmartConfig để cấu hình nhiều thiết bị cùng một lúc.

1.3.2 Một số ứng dụng trên điện thoại để thực hiện SmartConfig

- Sử dụng App ESP Touch (trên iOS) hoặc App ESP SmartConfig (trên Android) để thực hiện kết nối WiFi qua giao thức SmartConfig cho ESP8266.
- Ví dụ về cách hoạt động của giao thức ESP Touch gửi thông tin của mạng WiFi cho ESP8266: ESP Touch là protocol được dùng trong SmartConfig để người dùng có thể kết nối tới các phiên bản module ESP8266 thông qua cấu hình đơn giản trên Smartphone. Ban đầu không thể kết nối với ESP8266, nhưng thông qua giao thức ESP TOUCH thì Smartphone sẽ gửi gói UDP tới Access Point (AP) ở đây là ESP8266, mã hóa SSID và mật khẩu thành trường Length trong gói UDP, để ESP8266 có thể hiểu và giải mã được thông tin.

1.3.3 Sử dụng App ESP Touch trên iOS thực hiện SmartConfig cho ESP8266

- Cài đặt App Esptouch cho điện thoại (trên hệ điều hành iOS).
- Cách thực hiện kết nối WiFi cho ESP8266 thông qua SmartConfig với App Esptouch:
 - Mở App Esptouch (Hình 1.5a), nhập mật khẩu WiFi mà điện thoại đang kết nối vào (Hình 1.5b).
 - Nhấn Confirm để gửi thông tin của mạng WiFi cho ESP8266 và đợi phản hồi về: nếu kết quả giống Hình 1.5c là ESP8266 kết nối WiFi thành công.



Hình 1.5: Thực hiện SmartConfig với App Esptouch trên iOS

1.4 Ứng dụng Blynk trên điện thoại

1.4.1 Giới thiệu

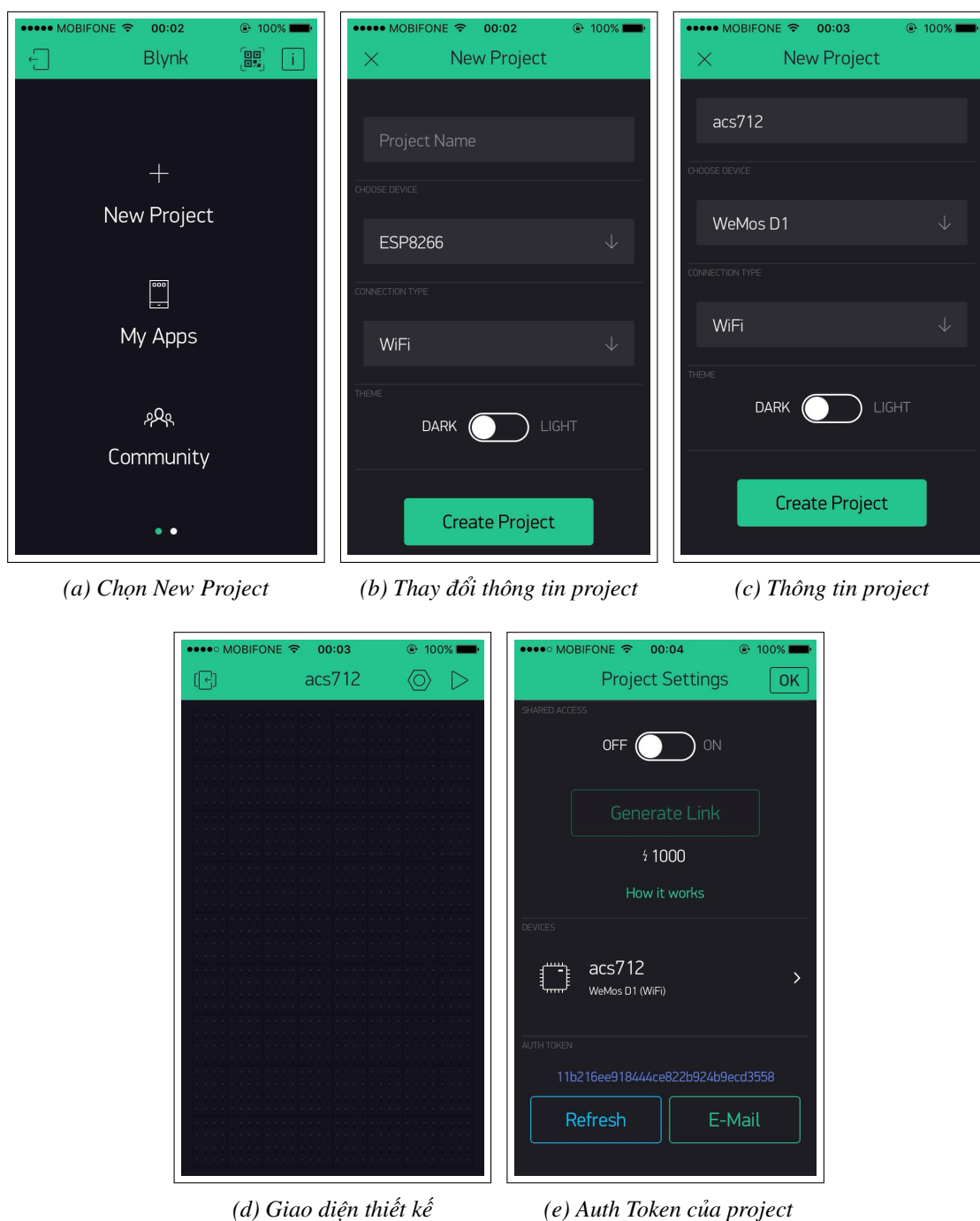
Blynk là ứng dụng điện thoại trên hệ điều hành iOS và Android hỗ trợ viết các ứng dụng di động cho các thiết bị thông minh. Ứng dụng cho phép chúng ta dễ dàng kết nối với các mạch tích hợp thông dụng như Arduino, Raspberry Pi, ESP8266, Particle (Photon/SparkCore) và điều khiển thông qua Internet.

1.4.2 Cài đặt App Blynk và thư viện Blynk cho lập trình

- Tải App Blynk về cài đặt trên điện thoại.
- Mở App Blynk và tạo tài khoản sử dụng App.
- Tải thư viện Blynk hỗ trợ viết chương trình: github.com/blynkkk/blynk-library

1.4.3 Tạo project trong App Blynk

Các thuộc tính cần lưu ý: Choose Device (vi điều khiển đang sử dụng), Connection Type (hình thức kết nối để điều khiển, ví dụ là WiFi) và Auth Token (mã xác thực kết nối giữa vi điều khiển và App Blynk).



Hình 1.6: Tạo project trong App Blynk

1.4.4 Kết nối giữa các phần tử trên App Blynk và vi điều khiển

- Để kết nối giữa App Blynk và vi điều khiển: cần chọn đúng Device, Connection Type và lấy Auth Token (được gửi qua Email đăng ký tài khoản).
- Để kết nối các phần tử điều khiển trên App Blynk với vi điều khiển: cần biết

chính xác PIN (Analog, Digital, Virtual) của các phần tử trên App Blynk để viết chương trình điều khiển tương ứng.

- Các PIN Analog và Digital thường dùng để điều khiển trực tiếp các phần cứng kết nối với vi điều khiển.
- Các PIN Virtual để tạo phần cứng ảo hỗ trợ cho vi điều khiển: điều khiển, hiển thị, truyền nhận dữ liệu giữa App và vi điều khiển.

1.5 Ứng dụng Web Server Thingspeak

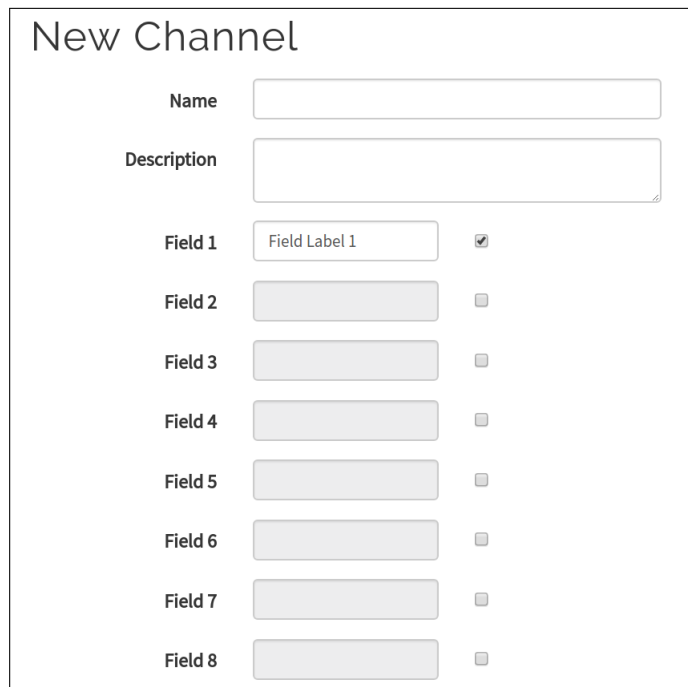
1.5.1 Giới thiệu

ThingSpeak là mã nguồn mở cho các ứng dụng Internet of Things – IoT, hỗ trợ các API lưu trữ, lấy dữ liệu từ thiết bị sử dụng HTTP thông qua kết nối Internet.

1.5.2 Cài đặt thư viện Thingspeak cho lập trình

Tải thư viện Thingspeak hỗ trợ viết chương trình: github.com/mathworks/thingspeak-arduino

1.5.3 Tạo tài khoản sử dụng và tạo project trên Thingspeak



The image shows the 'New Channel' form in the Thingspeak web interface. It contains the following elements:

- Name:** A text input field.
- Description:** A larger text area for a description.
- Field 1:** A text input field containing 'Field Label 1' and a checked checkbox.
- Field 2:** An empty text input field and an unchecked checkbox.
- Field 3:** An empty text input field and an unchecked checkbox.
- Field 4:** An empty text input field and an unchecked checkbox.
- Field 5:** An empty text input field and an unchecked checkbox.
- Field 6:** An empty text input field and an unchecked checkbox.
- Field 7:** An empty text input field and an unchecked checkbox.
- Field 8:** An empty text input field and an unchecked checkbox.

Hình 1.7: Các Field dùng lưu trữ dữ liệu trên Thingspeak

- Truy cập vào địa chỉ thingspeak.com, tạo tài khoản sử dụng Thingspeak.

- Chọn New Channel để tạo project mới: đặt tên cho project, thông tin mô tả cho project, tên cho các Field và chọn Save Channel. Các Field 1 đến Field 8 là các field dùng để lưu trữ dữ liệu từ vi điều khiển gửi lên Thingspeak (Hình 1.7).
- Để kết nối giữa Web Server Thingspeak và vi điều khiển cần có Channel ID và Write API Key (vào tab API Keys để lấy Write API Key).

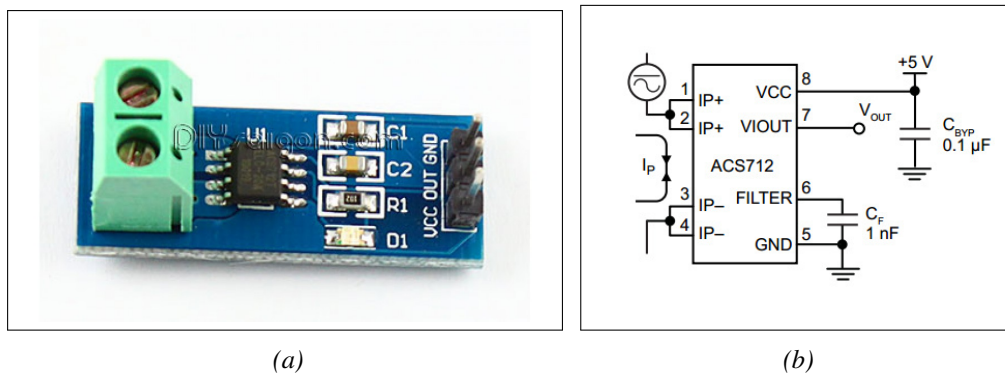
1.5.4 Cách giao tiếp vi điều khiển và Web Server Thingspeak

Sử dụng phương thức POST/GET/... trong giao thức HTTP để gửi dữ liệu lên Web Server Thingspeak.

1.6 Cảm biến dòng điện ACS712

1.6.1 Giới thiệu

Cảm biến dòng điện ACS712 là IC cảm biến dòng tuyến tính dựa trên hiệu ứng Hall. Cảm biến xuất ra tín hiệu analog V_{out} biến đổi tuyến tính theo sự thay đổi của dòng điện được lấy mẫu thứ cấp DC (hoặc AC), trong phạm vi đã cho. Tụ (C_f theo sơ đồ) được dùng với mục đích chống nhiễu và có giá trị tùy thuộc vào từng mục đích sử dụng.



Hình 1.8: Cảm biến dòng ACS712

1.6.2 Đặc điểm của cảm biến

- Thời gian tăng của đầu ra để đáp ứng với đầu vào là $5\mu s$.
- Điện trở dây dẫn trong là $1.2m\Omega$.
- Điện áp hoạt động 5V.

- Độ nhạy đầu ra từ 63 – 190mV/A ứng với từng loại cảm biến: 64 – 68 mV/A (loại 30A); 96 – 104mV/A (loại 20A) và 180 – 190mV/A (loại 5A).
- Điện áp ra ổn định.

1.6.3 Cách sử dụng module cảm biến dòng ACS712

- Đo dòng điện DC:
 - Khi đo DC phải mắc tải nối tiếp I_{p+} và I_{p-} đúng chiều, dòng điện đi từ I_{p+} đến I_{p-} để Vout ra mức điện thế 2.5V đến 5V tương ứng dòng 0 đến I_{dmcb} (dòng điện định mức mà cảm biến đo được), nếu mắc ngược Vout sẽ ra điện thế 2.5V đến 0V tương ứng với 0A đến $-I_{dmcb}$.
 - Cấp nguồn 5V cho module khi chưa có dòng I_p (chưa có tải mắc nối tiếp với domino), thì Vout là 2.5V. Khi dòng I_p (dòng của tải) bằng I_{dmcb} thì Vout là 5V, Vout sẽ tuyến tính với dòng I_p , trong khoảng 2.5V đến 5V tương ứng với dòng 0 đến I_{dmcb} .
 - Công thức tính dòng điện DC (với V_m là độ nhạy ứng với từng loại cảm biến dòng ACS712 được trình bày ở trên):

$$V_{(mV)} = \frac{adc \times 5000.0}{1024.0}; \quad I_{(mA)} = \frac{V - 2500}{V_m} \quad (\text{với } V_m \text{ là độ nhạy})$$

- Đo dòng điện AC:
 - Khi đo dòng điện AC, do dòng điện AC không có chiều nên không cần quan tâm chiều.
 - Cấp nguồn 5V cho module khi chưa có dòng I_p (chưa có tải mắc nối tiếp với domino), thì Vout là 2.5V. khi có dòng xoay chiều I_p (dòng AC) do dòng xoay chiều độ lớn thay đổi liên tục theo hàm sin, nên điện thế Vout sẽ là điện thế xoay chiều hình sin có độ lớn tuyến tính với dòng điện AC, 0 đến 5V (thế xoay chiều) tương ứng với $-I_{dmcb}$ đến $+I_{dmcb}$ (dòng xoay chiều).
 - Công thức tính dòng điện AC (với V_m là độ nhạy ứng với từng loại cảm biến dòng ACS712 được trình bày ở trên):

$$V_{(mV)} = \frac{adc_{MaxPoint} \times 5000.0}{1024.0}; \quad I_{(mA)} = \frac{V - 2500}{\sqrt{2}V_m} \quad (\text{với } V_m \text{ là độ nhạy})$$

1.7 Xác định công suất, điện năng tiêu thụ của phụ tải xoay chiều một pha dựa vào dòng điện qua tải và thời gian hoạt động của tải

- Công suất tiêu thụ của phụ tải xoay chiều một pha: $P = UI \cos \varphi$.
- Giá trị dòng điện đo được là giá trị tức thời: I_i .
- Công suất tức thời: $P = U_i I_i \cos \varphi_i$
- Công suất trung bình có thể xác định gần đúng bằng công thức sau:

$$P_{tb} = \frac{1}{n} \sum_{i=1}^n P_i$$

- Do cảm biến ACS712 chỉ có thể được dòng điện đi qua phụ tải, không đo được điện áp cấp cho phụ tải. Để đơn giản chọn điện áp U cấp cho phụ tải một pha và hệ số công suất $\cos \varphi$ là không đổi, khi đó ta có:

$$P_{tb} = \frac{1}{n} \sum_{i=1}^n P_i = \frac{U \cos \varphi}{n} \sum_{i=1}^n I_i$$

Khi tính toán, có thể chọn gần đúng $U = 220V$ và $\cos \varphi = 0.86$

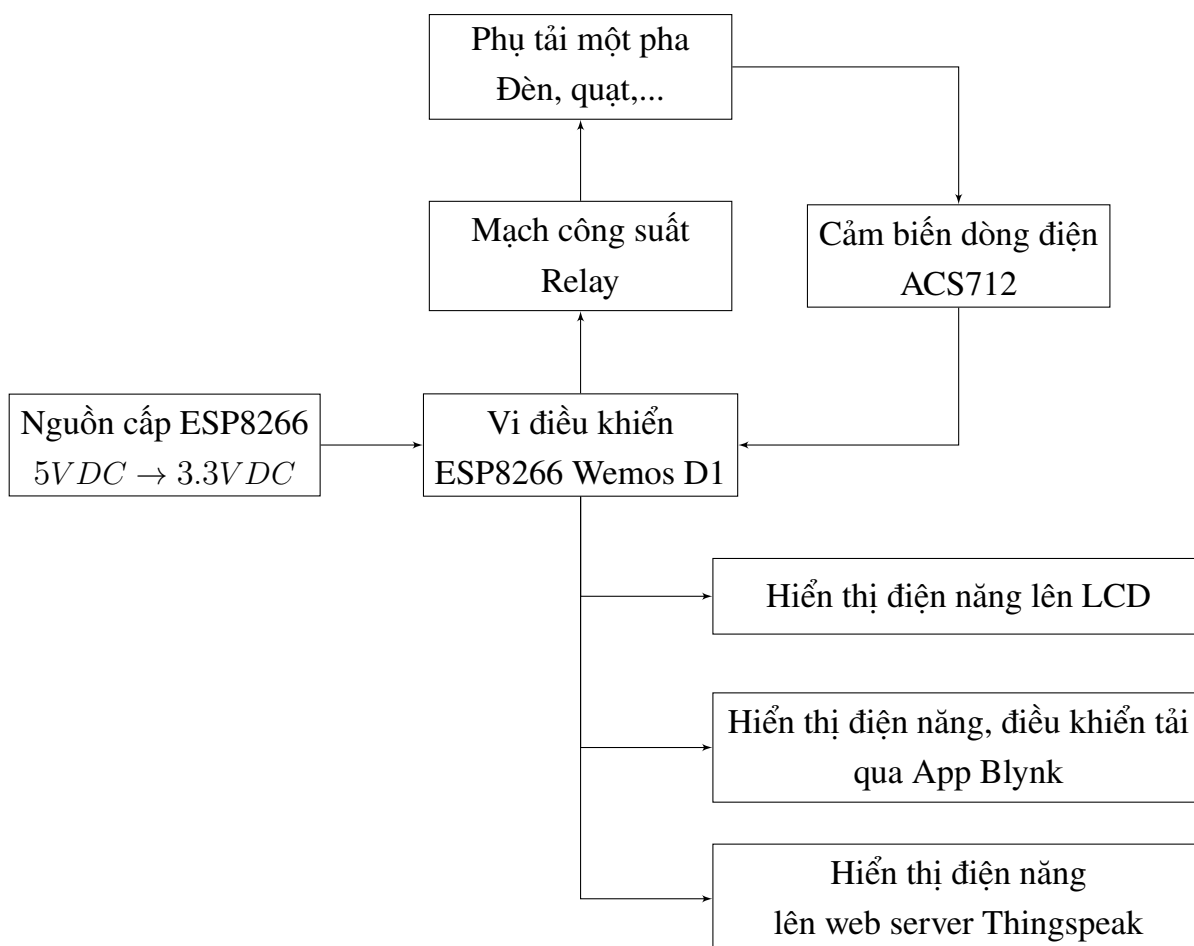
- Điện năng tiêu thụ của phụ tải: $A = \sum_{j=1}^m P_{tb_j} t_j$ với P_{tb_j} là công suất trung bình trong thời gian phụ tải hoạt động t_j .

CHƯƠNG 2

THIẾT KẾ MẠCH VÀ GIAO DIỆN ĐIỀU KHIỂN

2.1 Sơ đồ khối

Sơ đồ khối thiết kế bộ điều khiển được mô tả trên Hình 2.1:



Hình 2.1: Sơ đồ khối thiết kế bộ điều khiển

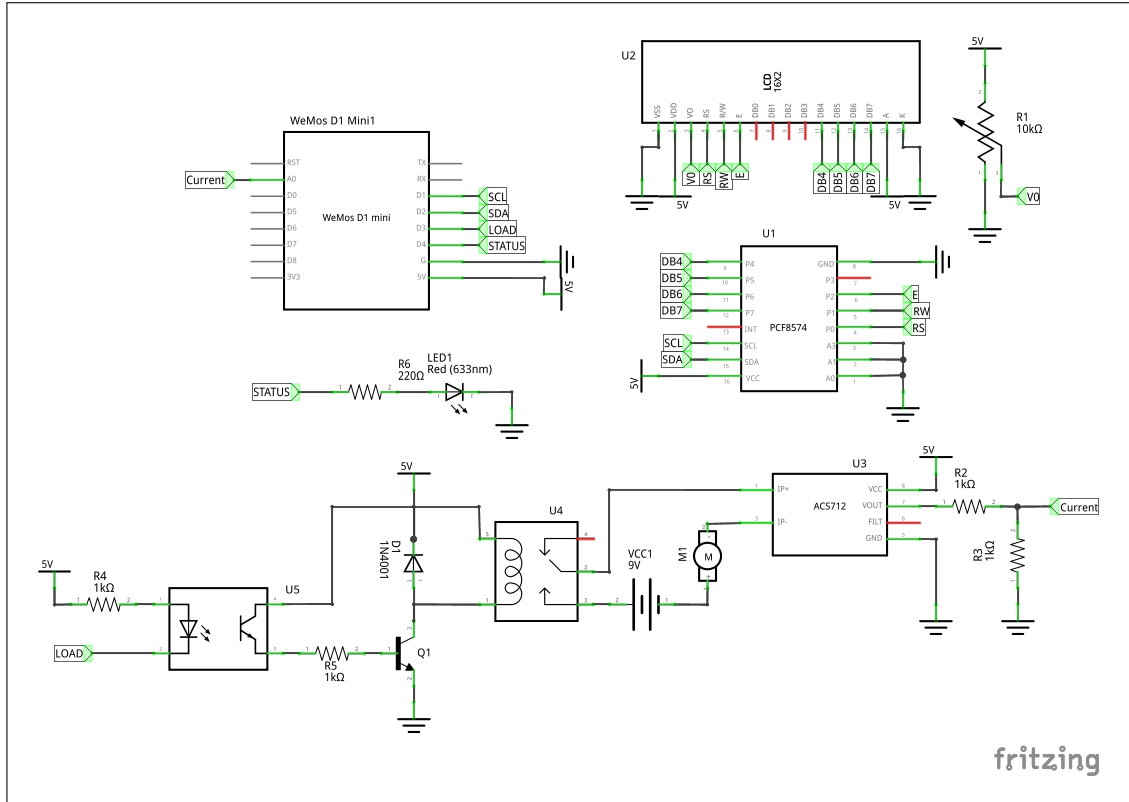
Mô tả các khối chức năng trên Hình 2.1:

- Sử dụng vi điều khiển ESP8266 Wemos D1 với khả năng giao tiếp WiFi (với nguồn hoạt động 5VDC → 3.3VDC).
- Điều khiển tải thông qua mạch công suất với Relay (sử dụng nguồn kích 5VDC).
- Sử dụng cảm biến dòng điện ACS712 để đo dòng điện tiêu thụ của phụ tải xoay chiều một pha, từ đó tính công suất, điện năng tiêu thụ của phụ tải.

- Hiển thị, gửi dữ liệu thu thập được (dòng điện, công suất) lên LCD (quan sát trực tiếp), App Blynk (quan sát trên thiết bị di động) và Server Thingspeak (quan sát trên Web).

2.2 Sơ đồ mạch nguyên lý

Sơ đồ mạch nguyên lý được mô tả trên Hình 2.2:



Hình 2.2: Sơ đồ mạch nguyên lý

2.3 Mô tả các khối chức năng trên sơ đồ mạch nguyên lý

- Hiển thị dòng điện và công suất với LCD 16x02 (sử dụng giao tiếp I2C).
- Đo dòng điện tiêu thụ của phụ tải với cảm biến dòng ACS712.
 - Do tín hiệu điện áp đưa vào chân analog của ESP8266 Wemos D1 tối đa là 3.3V, trong khi đó tín hiệu trả về của cảm biến có thể lên đến 5.0V.
 - Sử dụng cầu chia điện áp tại ngõ ra của cảm biến dòng để điện áp đưa vào chân analog của ESP8266 nhỏ hơn 3.3V. Do $R_2 = R_3 = R = 1k\Omega$, nên:

$$V_o = \frac{R}{R + R} V_i = \frac{V_i}{2} \implies V_i = 2.0V_o$$

– Khi đó các công thức tính dòng điện được xác định lại như sau:

* Với dòng DC:

$$V_{(mV)} = \frac{2.0 \times adc \times 5000.0}{1024.0} = \frac{adc \times 5000.0}{521.0};$$
$$I_{(mA)} = \frac{V - 2500}{V_m} \quad (\text{với } V_m \text{ là độ nhạy})$$

* Với dòng AC:

$$V_{(mV)} = \frac{2.0 \times adc_{MaxPoint} \times 5000.0}{1024.0} = \frac{adc_{MaxPoint} \times 5000.0}{512.0};$$
$$I_{(mA)} = \frac{V - 2500}{\sqrt{2}V_m} \quad (\text{với } V_m \text{ là độ nhạy})$$

- Điều khiển tải thông qua relay.
- LED báo trạng thái quá tải.

2.4 Giao diện điều khiển trên Blynk



Hình 2.3: Giao diện điều khiển trên Blynk

Phần tử	Pin	Giải thích
LCD	V0	Hiển thị dữ liệu
Button	D5	Điều khiển tải
LED	V1	Báo trạng thái quá tải
History Graph	V3/V4	Dữ liệu dòng điện DC/AC

Bảng 2.1: Các phần tử trên App Blynk

2.5 Giao diện thu thập dữ liệu trên Thingspeak

Tên Field	Giải thích
Field 3	Dòng DC
Field 4	Trạng thái quá tải
Field 5	Công suất tiêu thụ

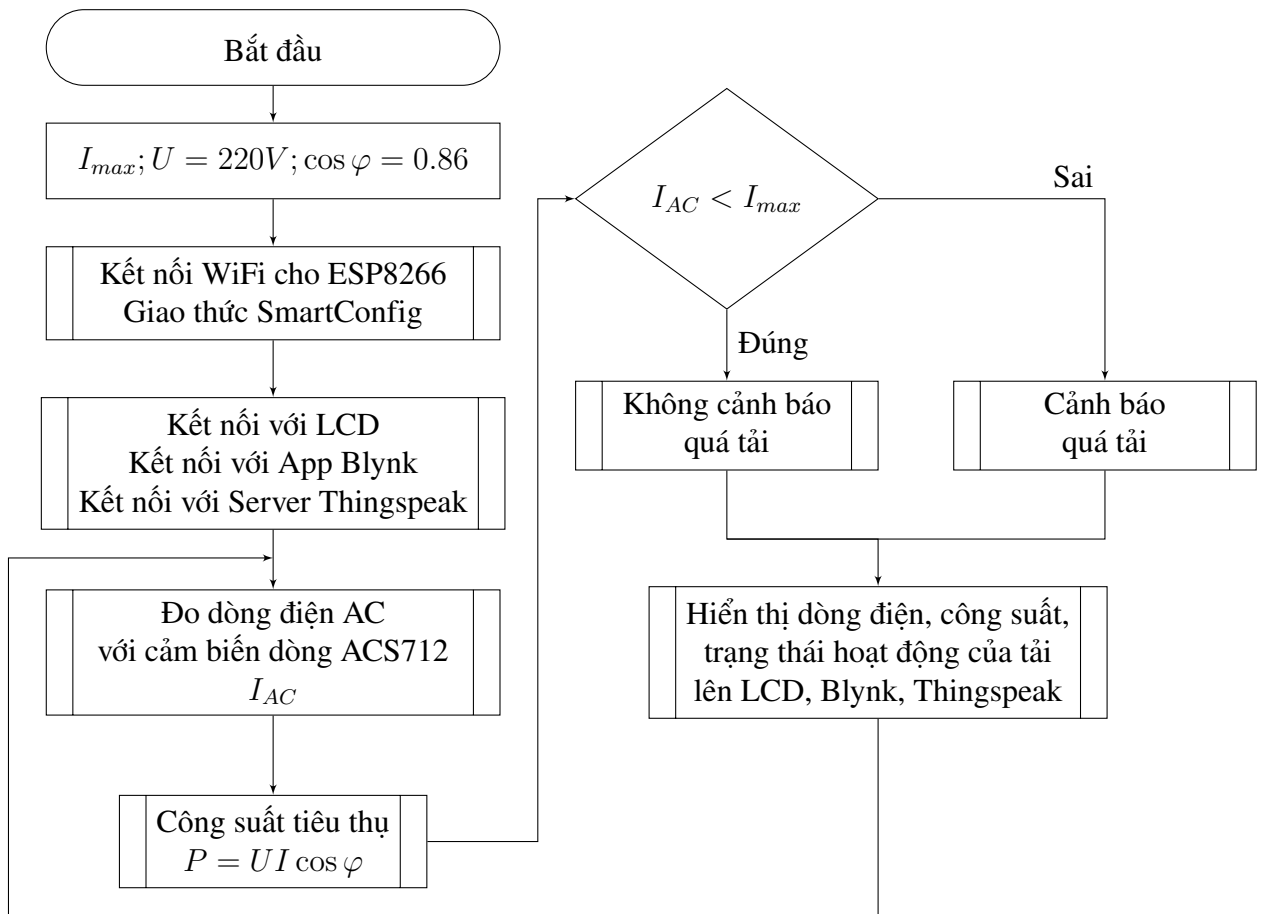
Bảng 2.2: Các Field trên web Thingspeak

CHƯƠNG 3

LƯU ĐỒ GIẢI THUẬT VÀ CHƯƠNG TRÌNH

3.1 Lưu đồ giải thuật

Lưu đồ giải thuật dùng cho chương trình được mô tả trên Hình 3.1:



Hình 3.1: Sơ đồ khối thiết kế bộ điều khiển

3.2 Chương trình điều khiển

```

#include <WiFiSupport.h>
#include <DisplayLCD.h>
#include <PowerAC.h>
#include <ControlLoad.h>
#include <BlynkControl.h>
#include <ThingspeakServer.h>
    
```

CHƯƠNG 3. LƯU ĐỒ GIẢI THUẬT VÀ CHƯƠNG TRÌNH

```
#define PIN_CONTROL_LOAD 2 // D4
#define PIN_CURRENT_AC A0

#define BLYNK_HISTORY_GRAPH_DC_CURRENT_PIN 3 // V3
#define BLYNK_HISTORY_GRAPH_AC_CURRENT_PIN 4 // V4

#define THINGSPEAK_AC_FIELD_CURRENT 3
#define THINGSPEAK_AC_FIELD_WORKING 4
#define THINGSPEAK_AC_FIELD_POWER 5

#define MAX_CURRENT_AMPERE 15

unsigned long channelNumberThingspeak = 382411;
const char* writeAPIKeyThingspeak = "MNM4PVW6OK72MSFR";
char authBlynk[] = "8125642efd0a4fdc98bbeaaa760405b0";

WiFiSupport WiFiConnect;
PowerAC AC(PIN_CURRENT_AC, 20, 3.3);
DisplayLCD LCD;
ControlLoad Load(PIN_CONTROL_LOAD);
BlynkControl BlynkApp;
ThingspeakServer Server(channelNumberThingspeak,
    writeAPIKeyThingspeak);

void setup() {
    Serial.begin(115200);
    WiFiConnect.smartConfig(LED_BUILTIN, false);
    BlynkApp.init(authBlynk);
    Server.init();
    LCD.init();
}

void loop() {
    float ampereCurrent = AC.getCurrent();
    float waltPower = AC.getPower(ampereCurrent);
    bool statusLoad = Load.isStatusControl(ampereCurrent,
        MAX_CURRENT_AMPERE);

    Load.control(statusLoad);
```

CHƯƠNG 3. LƯU ĐỒ GIẢI THUẬT VÀ CHƯƠNG TRÌNH

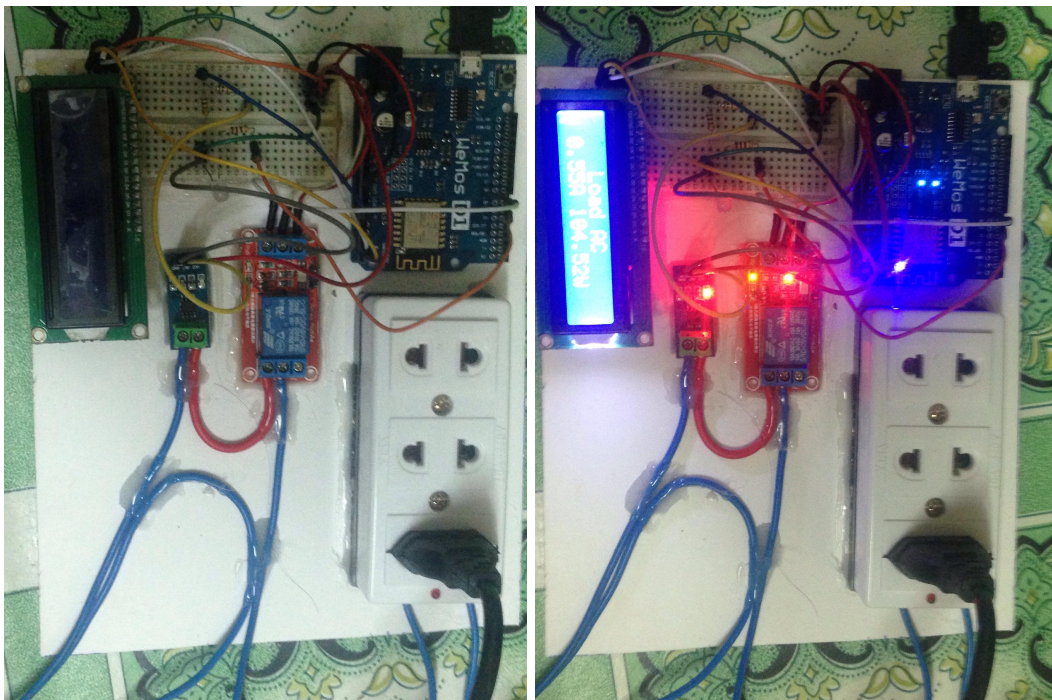
```
String textCurrent = String(ampereCurrent) + "A";
String textPower = String(waltPower) + "W";
LCD.text("Load AC", textCurrent + " " + textPower);

BlynkApp.textLCD("Load AC", textCurrent + " " + textPower);
BlynkApp.sendStatus(statusLoad);
BlynkApp.send(BLYNK_HISTORY_GRAPH_AC_CURRENT_PIN,
    ampereCurrent);

Server.set(THINGSPEAK_AC_FIELD_CURRENT, ampereCurrent);
Server.set(THINGSPEAK_AC_FIELD_WORKING, statusLoad);
Server.set(THINGSPEAK_AC_FIELD_POWER, waltPower);
Server.send();

BlynkApp.delay(15000);
BlynkApp.run();
}
```

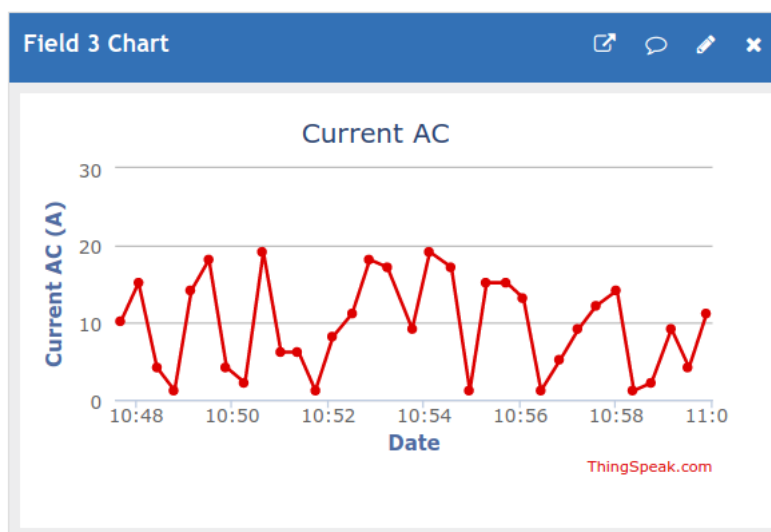
3.3 Kết quả thu được



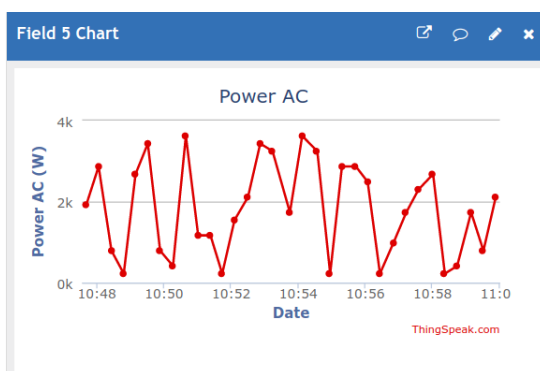
Hình 3.2: Kết quả thu được về phần cứng



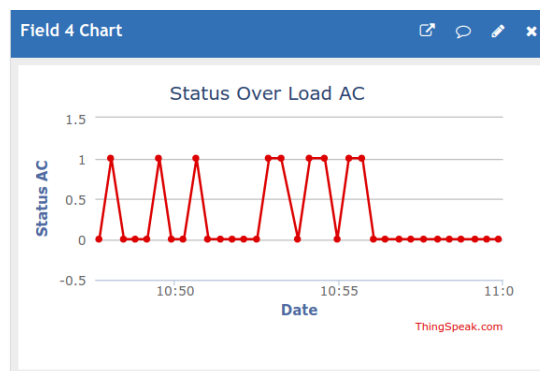
(a) Kết quả trên Blynk



(b) Kết quả dòng điện trên Thingspeak



(c) Kết quả công suất trên Thingspeak



(d) Kết quả trạng thái quá tải trên Thingspeak

Hình 3.3: Kết quả thu được quan sát trên ứng dụng

CHƯƠNG 4

KẾT LUẬN

4.1 Kết quả

Qua thời gian thực hiện đồ án, nhóm chúng em đạt được một số kết quả sau:

- Biết cách sử dụng module ESP8266 vào một số ứng dụng có kết nối WiFi.
- Điều khiển thiết bị từ xa và thu thập dữ liệu thông qua mạng WiFi với ứng dụng Blynk Web Server Thingspeak.
- Xây dựng ứng dụng theo dõi điện năng của phụ tải một pha với điều khiển và giám sát dữ liệu qua mạng Internet.

4.2 Hạn chế

Nhóm chúng em đã cố gắng tìm hiểu nhưng vẫn còn một số hạn chế sau:

- Chưa sử dụng nhiều tính năng của module ESP8266, App Blynk và Web Server Thingspeak.
- Trong ứng dụng theo dõi điện năng: chỉ đo một thông số là dòng điện tiêu thụ của phụ tải, cần kết hợp đo nhiều thông số hơn: điện áp, hệ số công suất, tần số dòng điện để có thể theo dõi chất lượng điện năng tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] Internet Of Things (IoT) cho người mới bắt đầu, IoT Maker Việt Nam.
- [2] AC current measurement using acs712 hall effect current sensor and Arduino, Microcontrollers Lab.
- [3] DC current measurement using acs712 hall effect current sensor and Arduino, Microcontrollers Lab.
- [4] ThingSpeak, Blynk Examples.