

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH



# ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

## ỨNG DỤNG CÔNG NGHỆ IOT GIÁM SÁT MỨC TIÊU THỤ ĐIỆN – NƯỚC

GVHD: Ths. Nguyễn Thanh Nghĩa

SVTH: Nguyễn Thanh Ti – 11141210

Phạm Quốc Hưng - 12141105

Tp. Hồ Chí Minh - 07/2018

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

ỨNG DỤNG CÔNG NGHỆ IOT GIÁM  
SÁT MỨC TIÊU THỤ ĐIỆN – NƯỚC

GVHD: Ths. Nguyễn Thanh Nghĩa

SVTH: Nguyễn Thanh Ti – 11141210

Phạm Quốc Hưng - 12141105

Tp. Hồ Chí Minh - 07/2018

TRƯỜNG ĐH. SƯ PHẠM KỸ THUẬT  
TP. HỒ CHÍ MINH  
**KHOA ĐIỆN-ĐIỆN TỬ**  
**BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP**  
– Y SINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

ĐỘC LẬP - TỰ DO - HẠNH PHÚC

Tp. HCM, ngày 16 tháng 07 năm 2018

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Phạm Quốc Hưng MSSV: 12141105

Nguyễn Thanh Ti MSSV: 11141210

Chuyên ngành: Kỹ thuật Điện - Điện tử Mã ngành: 510302

Hệ đào tạo: Đại học chính quy Mã hệ: D

### I. TÊN ĐỀ TÀI: **ỨNG DỤNG CÔNG NGHỆ IOT GIÁM SÁT MỨC TIÊU THU ĐIỆN – NUỚC**

#### II. NHIỆM VỤ

##### 1. Các số liệu ban đầu:

- Kit Arduino Mega, NodeMCU và ngôn ngữ lập trình
- Tài liệu nghiên cứu Arduino Mega, NodeMCU, Firebase
- Tài liệu nghiên cứu cảm biến dòng ACS712 và lưu lượng S201

##### 2. Nội dung thực hiện:

- Kết nối các cảm biến, nodemcu, mạch đo áp vào mạch Arduino.
- Lập trình cho kit Arduino và nodemcu.
- Thiết kế mô hình hộp chứa mạch điều khiển.
- Xây dựng giao diện và lập trình trang web giám sát từ xa.
- Chạy thử nghiệm.
- Cân chỉnh hệ thống.

III. NGÀY GIAO NHIỆM VỤ: 21/03/2018

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 16/07/2018

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: Ths. Nguyễn Thanh Nghĩa

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

KHOA ĐIỆN-ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

Tp. HCM, ngày tháng năm 2018

## LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Phạm Quốc Hưng .....

Lớp: 12141DT1A..... MSSV: 12141105 .....

Họ tên sinh viên 2: Nguyễn Thanh Ti .....

Lớp: 1114DT1D..... MSSV: 11141210 .....

Tên đề tài: Ứng dụng công nghệ IoT giám sát mức tiêu thụ điện - nước.....

<i>Tuần/ngày</i>	<i>Nội dung</i>	<i>Xác nhận GVHD</i>
<b>Tuần 1</b> (26/03-01/04)	<ul style="list-style-type: none"><li>_ Báo cáo GVHD</li><li>_ Tìm đề tài</li></ul>	
<b>Tuần 2</b> (02/04-08/04)	<ul style="list-style-type: none"><li>_ Báo cáo GVHD</li><li>_ Tìm hiểu về đề tài, các công thức tính toán, các phương pháp thực hiện</li></ul>	
<b>Tuần 3</b> (09/04-15/04)	<ul style="list-style-type: none"><li>_ Báo cáo GVHD</li><li>_ Tổng hợp các linh kiện cần dùng cho đề tài</li></ul>	
<b>Tuần 4</b> (16/04-22/04)	<ul style="list-style-type: none"><li>_ Báo cáo GVHD</li><li>_ Tìm hiểu về hoạt động của Arduino, Nodecmu.</li></ul>	
<b>Tuần 5</b> (23/04-29/04)	<ul style="list-style-type: none"><li>_ Báo cáo GVHD</li><li>_ Tìm hiểu về thiết kế giao diện giám sát qua web server.</li></ul>	

<b>Tuần 6</b> (30/04-06/05)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Tìm hiểu về Firebase, và thiết kế giao diện web</li> </ul>	
<b>Tuần 7</b> (07/05-13/05)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Tìm hiểu về nguyên lý hoạt động của các cảm biến</li> </ul>	
<b>Tuần 8</b> (14/05-20/05)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Tìm hiểu kết nối và test hoạt động của các module, cảm biến với Arduino, Nodemcu</li> </ul>	
<b>Tuần 9</b> (21/05-27/05)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Viết chương trình cho toàn hệ thống</li> </ul>	
<b>Tuần 10</b> (28/05-03/06)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Thi công mô hình thiết kế vỏ hộp</li> </ul>	
<b>Tuần 11</b> (04/06-10/06)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Thi công mạch điều khiển và chỉnh sửa</li> </ul>	
<b>Tuần 12</b> (11/06-17/06)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Đóng gói mạch điều khiển, chạy thử nghiệm</li> </ul>	
<b>Tuần 13</b> (18/06-24/06)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Chỉnh sửa, điều chỉnh lại mạch</li> </ul>	
<b>Tuần 14</b> (25/06-01/07)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Chỉnh sửa luận văn</li> </ul>	
<b>Tuần 15</b> (02/07-08/07)	<ul style="list-style-type: none"> <li>_ Báo cáo GVHD</li> <li>_ Chỉnh sửa và in luận văn</li> </ul>	

GV HƯỚNG DẪN

(Ký và ghi rõ họ và tên)

## LỜI CAM ĐOAN

Đề tài này là chúng tôi tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó. Nếu có bất kỳ sự gian lận nào chúng tôi xin chịu trách nhiệm về nội dung đồ án của mình.

Người thực hiện đề tài

Phạm Quốc Hưng – 12141105

Nguyễn Thanh Ti - 11141210

## LỜI CẢM ƠN

Nhóm em chân thành cảm ơn đến các thầy, cô Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM đã tận tình chỉ dạy, giúp đỡ nhóm trong suốt quá trình học tập tích lũy kiến thức ở trường, đặc biệt là các thầy, cô của khoa Điện-Điện Tử.

Đặc biệt, chúng em muôn cảm ơn Thầy Nguyễn Thanh Nghĩa đã tận tình giúp đỡ cũng như hỗ trợ trong suốt quá trình thực hiện đề tài. Trong suốt quá trình thực hiện, thầy đã tận tình góp ý, chỉ dẫn và đôn đốc sinh viên để hoàn thành đề tài hoàn chỉnh và đúng hạn. Một lần nữa em xin cảm ơn thầy.

Cuối cùng, chúng con xin chân thành cảm ơn sự động viên và hỗ trợ của gia đình và cha mẹ trong suốt quá trình học tập. Chúng con xin gửi cảm ơn trân trọng đến các bậc sinh thành đã nuôι dưỡng, hỗ trợ chúng con từ kinh phí cũng như tinh thần giúp chúng con hoàn thành tốt đề tài.

Người thực hiện đề tài

Phạm Quốc Hưng – 12141105

Nguyễn Thanh Ti - 11141210

# MỤC LỤC

Trang bìa.....	i
Nhiệm vụ đồ án .....	ii
Lịch trình .....	iii
Cam đoan .....	v
Lời cảm ơn.....	vi
Mục lục .....	vii
Liệt kê hình vẽ .....	x
Liệt kê bảng .....	xiii
Tóm tắt .....	xiv
<b>CHƯƠNG 1. TỔNG QUAN.....</b>	<b>1</b>
1.1. Đặt vấn đề .....	1
1.2. Mục tiêu .....	1
1.3. Nội dung nghiên cứu .....	1
1.4. Giới hạn .....	2
1.5. Bố cục .....	2
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>3</b>
2.1. Giới thiệu phần cứng.....	3
2.1.1. Thiết bị đầu vào .....	3
2.1.1.1. Module cảm biến dòng điện ACS712 - 20A.....	3
2.1.1.2. Cảm biến lưu lượng S201.....	5
2.1.2. Thiết bị đầu ra – Màn hình LCD 16x2 và mạch LCD I2C .....	7
2.1.2.1. LCD 16x2 .....	7
2.1.2.2. Module giao tiếp LCD I2C.....	8

2.1.3. Arduino Mega 2560 .....	10
2.1.3.1. Giới thiệu.....	10
2.1.3.2. Thông số kỹ thuật.....	10
2.1.4. NodeMCU 1.0.....	11
2.1.4.1. Giới thiệu.....	11
2.1.4.2. Thông số kỹ thuật.....	12
2.2. Chuẩn truyền dữ liệu.....	14
2.2.1 Giao tiếp UART .....	14
2.2.1.1. Giới thiệu.....	14
2.2.1.2. Các thông số trong truyền nhận UART .....	15
2.2.2 Chuẩn giao tiếp I2C .....	15
2.2.2.1. Giới thiệu.....	15
2.2.2.2. Đặc điểm giao tiếp I2C.....	16
2.2.2.3 Trình tự truyền bit trên đường truyền.....	17
2.2.2.4 Điều kiện START và STOP.....	18
2.2.3 Chuẩn giao tiếp Wifi .....	19
2.2.3.1 Giới thiệu.....	19
2.2.3.2 Nguyên tắc hoạt động.....	19
2.2.3.3 Một số chuẩn kết nối Wifi.....	20
2.3. Firebase Hosting .....	22
2.3.1 Giới thiệu .....	22
2.3.2 Ưu điểm của Firebase .....	23
2.4. Firebase Realtime Database .....	24
2.4.1 Giới thiệu .....	24
2.4.2 Những đặc điểm nổi bật.....	24

2.4.2.1	Cách dữ liệu được lưu trữ.....	24
2.4.2.2	Dữ liệu offline .....	25
2.4.2.3	Cập nhật dữ liệu thời gian thực .....	25
2.4.2.4	Tính bảo mật và quy định.....	25
<b>CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG.....</b>		<b>27</b>
3.1.	Giới thiệu .....	27
3.2.	Tính toán và thiết kế hệ thống .....	27
3.2.1.	Thiết kế sơ đồ khái niệm hệ thống.....	27
3.2.2.	Tính toán và thiết kế mạch.....	28
3.2.2.1.	Thiết kế khái niệm hiển thị .....	28
3.2.2.2.	Thiết kế khái niệm xử lý .....	28
3.2.2.3	Thiết kế khái niệm thiết bị đầu vào .....	29
3.2.2.4	Thiết kế khái niệm nguồn.....	30
3.2.3.	Sơ đồ nguyên lý của toàn mạch .....	31
<b>CHƯƠNG 4. THI CÔNG HỆ THỐNG .....</b>		<b>32</b>
4.1.	Giới thiệu .....	32
4.2	Thi công hệ thống .....	32
4.3	Thi công và đóng gói mô hình.....	35
4.3.1	Đóng gói bộ điều khiển.....	35
4.3.2	Thi công mô hình .....	36
4.4	Lập trình hệ thống .....	37
4.4.1	Lưu đồ giải thuật.....	37
4.4.2	Phần mềm lập trình .....	40
4.4.3	Hướng dẫn xây dựng Firebase project.....	48
4.5	Hướng dẫn sử dụng .....	53

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ .....54**

5.1. Kết quả đạt được.....	54
5.1.1 Kết quả lý thuyết.....	54
5.1.2. Kết quả chạy hệ thống .....	55
5.2. Đánh giá, nhận xét.....	59

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....61**

6.1. Kết luận. ....	61
6.2. Hướng phát triển .....	61

## **TÀI LIỆU THAM KHẢO**

## **PHỤ LỤC**

# LIỆT KÊ HÌNH VẼ

Hình	Trang
Hình 2.1 Hình ảnh của Module và IC cảm biến dòng ACS712 .....	3
Hình 2.2 Hình ảnh minh họa kết nối với module ACS712.....	4
Hình 2.3 Hình ảnh của cảm biến lưu lượng S201 và kết nối .....	5
Hình 2.4 Sơ đồ chân của LCD 16x2 .....	7
Hình 2.5 Module giao tiếp LCD I2C.....	9
Hình 2.6 Thành phần Arduino Mega 2560 .....	11
Hình 2.7 NodeMCU 1.0 (ESP-12E Module) .....	12
Hình 2.8 Sơ đồ chân NodeMCU 1.0 (ESP-12E Module) .....	13
Hình 2.9 Truyền dữ liệu bằng chuẩn UART .....	14
Hình 2.10 Thiết bị ngoại vi giao tiếp bus I2C .....	15
Hình 2.11 Thiết bị kết nối vào I2C ở chế độ chuẩn và chế độ nhanh.....	16
Hình 2.12 Quá trình giao tiếp giữa thiết bị chủ - tớ.....	17
Hình 2.13 Trình tự truyền dữ liệu .....	17
Hình 2.14 Giản đồ thời gian điều kiện START và STOP .....	18
Hình 2.15 Mô hình hoạt động của mạng Wifi.....	19
Hình 2.16 Bảng so sánh thông số các chuẩn wifi .....	22
Hình 2.17 Các dịch vụ hỗ trợ của Firebase.....	23
Hình 2.18 Realtime database của Firebase .....	24
Hình 2.19 Quy tắc đóng, cần xác thực để đọc ghi dữ liệu .....	25
Hình 2.20 Quy tắc mở, cho phép mọi người đọc ghi dữ liệu.....	26
Hình 3.1 Sơ đồ khái niệm .....	28
Hình 3.2 Mạch đo điện áp .....	29
Hình 3.3 Mạch nguồn cấp cho Arduino và NodeMCU.....	30
Hình 3.4 Sơ đồ nguyên lý toàn mạch .....	31

Hình 4.1 Sơ đồ mạch in .....	33
Hình 4.2 Mặt ngoài của mô hình .....	35
Hình 4.3 Mặt trong của mô hình và kết nối .....	36
Hình 4.4 Lưu đồ giải thuật của NodeMcu .....	38
Hình 4.5 Lưu đồ giải thuật của Arduino .....	39
Hình 4.6 Giao diện tải Arduino IDE .....	40
Hình 4.7 Ủng hộ nhà phát triển Arduino IDE .....	41
Hình 4.8 Giao diện chính của Arduino IDE.....	41
Hình 4.9 Cài đặt Driver cho Arduino 1 .....	42
Hình 4.10 Cài đặt Driver cho Arduino 2 .....	43
Hình 4.11 Cài đặt Driver cho Arduino 3 .....	43
Hình 4.12 Cài đặt Driver cho NodeMCU 1 .....	44
Hình 4.13 Cài đặt Driver cho NodeMCU 2 .....	44
Hình 4.14 Cài đặt Driver cho NodeMCU 3 .....	45
Hình 4.15 Cài đặt Driver cho NodeMCU 4 .....	45
Hình 4.16 Chọn phần cứng để lập trình.....	46
Hình 4.17 Chọn Port kết nối.....	47
Hình 4.18 Cài đặt thư viện cho Arduino IDE .....	47
Hình 4.19 Giao diện Sublime Text .....	48
Hình 4.20 Giao diện Firebase console.....	49
Hình 4.21 Tạo một project mới .....	49
Hình 4.22: Các bước tạo một project.....	50
Hình 4.23 Chọn nền tảng xây dựng project .....	50
Hình 4.24 Giao diện download node.js .....	51
Hình 4.25 Giao diện làm việc của node.js .....	51
Hình 4.26 Khởi tạo firebase từ cửa sổ cmd .....	52
Hình 5.1 Màn hình LCD hiển thị giá trị điện năng.....	55

Hình 5.2 Màn hình LCD hiển thị giá trị nước sinh hoạt .....	55
Hình 5.3 Giao diện đăng nhập của người dùng .....	56
Hình 5.4 Giao diện quản lý của admin - 1 .....	56
Hình 5.5 Giao diện quản lý của admin – 2 .....	57
Hình 5.6 Giao diện quản lý giám sát của user - 1 .....	57
Hình 5.7: Giao diện quản lý giám sát của user – 2 .....	58
Hình 5.8: Quản lý tài khoản người dùng trong database .....	58

## **LIỆT KÊ BẢNG**

<b>Bảng</b>	<b>Trang</b>
Bảng 2.1 Kết nối LCD với Nodemcu .....	9
Bảng 4.1 Danh sách các linh kiện điện tử .....	34

## TÓM TẮT

Trong những năm gần đây, dường như thuật ngữ IoT (hay Internet of Things) hay “Vạn vật kết nối internet” đã không còn trở nên quá xa lạ, ta có thể đôi lần bắt gặp cụm từ này ở bất kỳ đâu, từ những bản tin thời sự - công nghệ trên tivi, trên các trang mạng điện tử, hoặc cụ thể là những ứng dụng thiết thực trong đời sống. Đúng như tên gọi, đây là một hệ thống các thiết bị công nghệ có liên quan đến nhau, mọi vật được kết nối với nhau dựa trên giao thức chung, đó là mạng truyền thông – hay Internet. Chỉ cần một thiết bị có kết nối mạng, là bạn có thể hoàn toàn kiểm tra, điều khiển các thiết bị trong nhà, bất kể bạn đang ở đâu. Công nghệ IoT đã và đang phát triển trong rất nhiều lĩnh vực.

Với những lợi ích trông thấy, bạn cũng muốn sở hữu một ứng dụng IoT cho căn nhà của bạn phải không nào? Vậy ứng dụng vào đâu bây giờ, ngoài việc chỉ điều khiển các thiết bị điện từ xa? Vậy có bao giờ bạn phải đau đầu tự hỏi tháng này hóa đơn tiền điện nước lại tăng lên trong khi bạn nghĩ là đã sử dụng chúng một cách hợp lý và tiết kiệm chưa? Chẳng lẽ đồng hồ lại báo số sai?, cũng có thể. Như vậy, bạn cần phải có một ứng dụng để có thể giám sát thông số điện-nước mà gia đình bạn sử dụng hàng ngày; đến cuối tháng, bạn tổng kết lại, đối chiếu với hóa đơn điện-nước trong tháng này, chứ không còn phụ thuộc vào hóa đơn của công ty điện nước như trước kia nữa. Thực ra, trên thị trường đã có những thiết bị như thế này rồi, với độ chính xác cao, nhưng giá thành lại rất mắc, và lại không thể giám sát được từ xa.

Nắm bắt được điều này, vận dụng kiến thức đã học, nhóm em đã tiến hành thực hiện đề tài với tên “**ỨNG DỤNG CÔNG NGHỆ IOT GIÁM SÁT MỨC TIÊU THỤ ĐIỆN - NƯỚC**”, thực hiện công việc đo và giám sát, hiển thị và cập nhật lên màn hình thiết bị và trên web, giúp cho người sử dụng có thể dễ dàng quan sát cũng như thống kê được lượng điện - nước mà họ đã và đang sử dụng. Với đề tài này, nhóm hy vọng sẽ làm cơ sở nghiên cứu để các nhóm sau có thể phát triển và cải tiến thêm nữa.

# Chương 1. TỔNG QUAN

## 1.1. ĐẶT VẤN ĐỀ

Hàng tháng, chúng ta phải luôn trả các hóa đơn điện – nước mà hầu như số tiền phải đóng lại ngày một tăng cao. Lý do ở đây là ta không thể hoàn toàn kiểm soát được mức điện – nước đã được sử dụng, bởi hầu như ta không có bất kỳ con số thống kê cụ thể nào cả, ngoài việc tự ước lượng. Hiện nay, trên thị trường cũng có bán các thiết bị để giám sát điện năng tiêu thụ với độ chính xác khá cao, nhưng giá thành thì lại không hề rẻ, cũng như hạn chế về mặt giám sát từ xa.

Nhận thấy được điều này, nhóm chúng em muốn tạo ra một ứng dụng giúp cho các hộ gia đình có thể dễ dàng thống kê - giám sát được lượng điện - nước mà họ sử dụng hàng ngày; để từ đó họ có thể kiểm soát và đề ra phương án sử dụng một cách hiệu quả và tiết kiệm hơn. Đó là lý do nhóm em quyết định lựa chọn và thực hiện đề tài “**ỨNG DỤNG CÔNG NGHỆ IOT GIÁM SÁT MỨC TIÊU THỤ ĐIỆN - NƯỚC**”.

## 1.2. MỤC TIÊU

Thiết kế một hệ thống tiến hành đo lượng điện – nước tiêu thụ, và đều đặn cập nhật các thông số đó lên một trang web-host để thuận tiện cho công việc giám sát. Hệ thống ứng dụng công nghệ IoT, giúp cho người dùng ở bất kỳ đâu cũng có thể dễ dàng truy cập được. Đồng thời, ứng dụng cũng xây dựng một hệ thống các user, giúp cho quản trị viên dễ dàng hơn trong việc kiểm soát thông tin người dùng.

## 1.3. NỘI DUNG NGHIÊN CỨU

- **NỘI DUNG 1:** Tìm hiểu và lựa chọn các giải pháp thiết kế.
- **NỘI DUNG 2:** Thu thập tài liệu về các cảm biến, module wifi, bộ vi xử lý, cũng như tìm kiếm một web host khả dụng.
- **NỘI DUNG 3:** Thiết kế, lập trình cho hệ thống điều khiển, chạy thử nghiệm.
- **NỘI DUNG 4:** Thiết kế mô hình, chỉnh sửa và cải tiến từ những phương án đã chọn
- **NỘI DUNG 5:** Đánh giá kết quả thực hiện

### 1.4. GIỚI HẠN

- Hệ thống chỉ dùng lại ở công việc giám sát các thông số
- Mô hình không quá to để có thể dễ dàng sử dụng ở nhà
- Sử dụng nguồn điện lấy trực tiếp từ lưới điện gia đình.
- Sai số hệ thống chấp nhận được

### 1.5. BỐ CỤC

- **Chương 1:** Tổng Quan

Đặt vấn đề, dẫn nhập đề tài: nêu mục tiêu và nội dung nghiên cứu, những hạn chế và bối cảnh của đề tài.

- **Chương 2:** Cơ Sở Lý Thuyết

Giới thiệu phần cứng: các thiết bị đầu vào, đầu ra. Các chuẩn truyền dữ liệu. Hướng dẫn cách lưu trữ nội dung trên firebase hosting .

- **Chương 3:** Tính toán và thiết kế

Tính toán và thiết kế hệ thống: nêu sơ đồ nguyên lý toàn mạch, và thiết kế mô hình.

- **Chương 4:** Thi công hệ thống.

Giới thiệu phần thi công mạch, đóng gói bộ điều khiển, các bước thi công mô hình hoàn chỉnh. Viết hướng dẫn sử dụng.

- **Chương 5:** Kết Quả, Nhận Xét và Đánh Giá

Nêu lên kết quả đã hoàn thành được, hình ảnh hoạt động của mạch, nhận xét và đánh giá.

- **Chương 6:** Kết Luận và Hướng Phát Triển

Nêu ra những gì đã thực hiện được trong đề tài và hướng phát triển của đề tài.

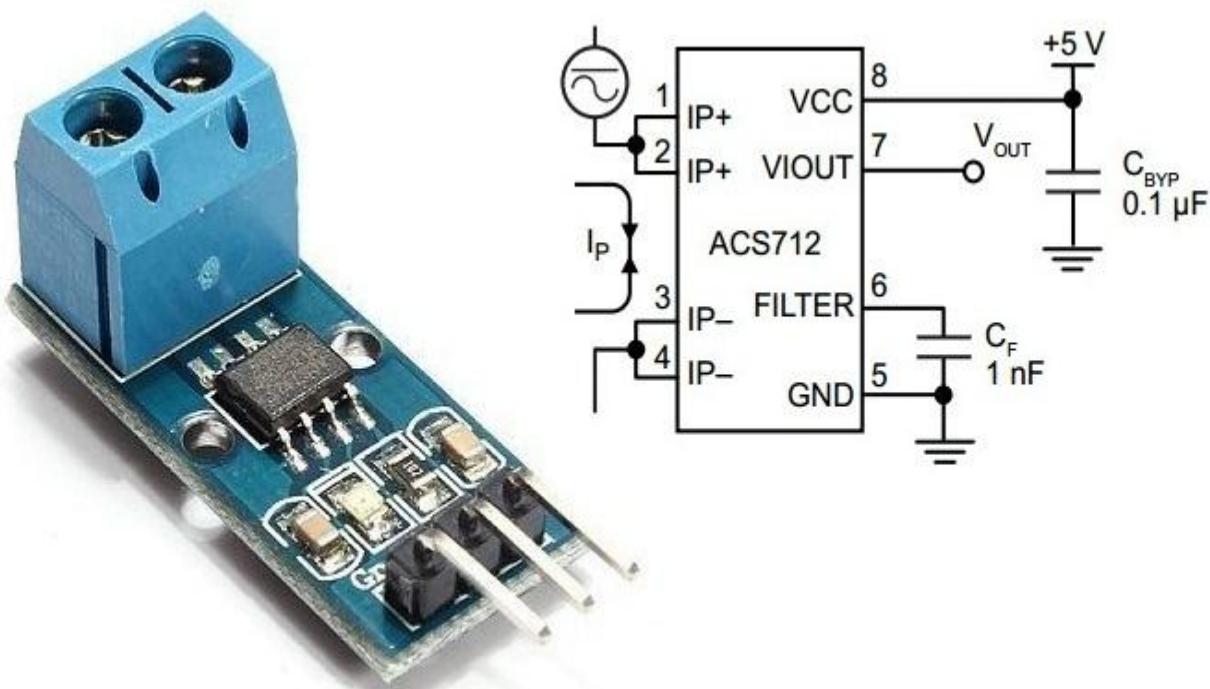
## Chương 2. CƠ SỞ LÝ THUYẾT

### 2.1 GIỚI THIỆU PHẦN CỨNG

#### 2.1.1 Thiết bị đầu vào

##### 2.1.1.1 Module cảm biến dòng điện ACS712 - 20A

Để đo dòng điện AC, ta có thể dùng máy biến dòng CT, đây là một loại “công cụ đo lường dòng điện” được thiết kế nhằm tạo ra một dòng điện xoay chiều có cường độ tỷ lệ với cường độ dòng điện ban đầu. Tuy nhiên, giá thành của thiết bị này khá là mắc. Ta có thể dùng cảm biến ACS712 được tích hợp sẵn vào module để thực hiện đo dòng điện với độ chính xác khá cao, kết nối đơn giản – thuận tiện và giá thành phải chăng.



**Hình 2.1** Hình ảnh của Module và IC cảm biến dòng ACS712

Module ở hình trên sử dụng cảm biến dòng điện ACS712 – đây là một cảm biến dòng tuyến tính dựa trên hiệu ứng Hall, giúp chuyển đổi dòng điện cần đo thành giá trị điện thế. Chân ACS712 sẽ xuất ra một tín hiệu analog ở chân Vout biến đổi tuyến tính theo  $I_p$  (dòng điện cần đo) được lấy mẫu thử cấp DC (hoặc AC) trong phạm vi cho phép. Tụ  $C_F$  dùng cho mục đích chống nhiễu.

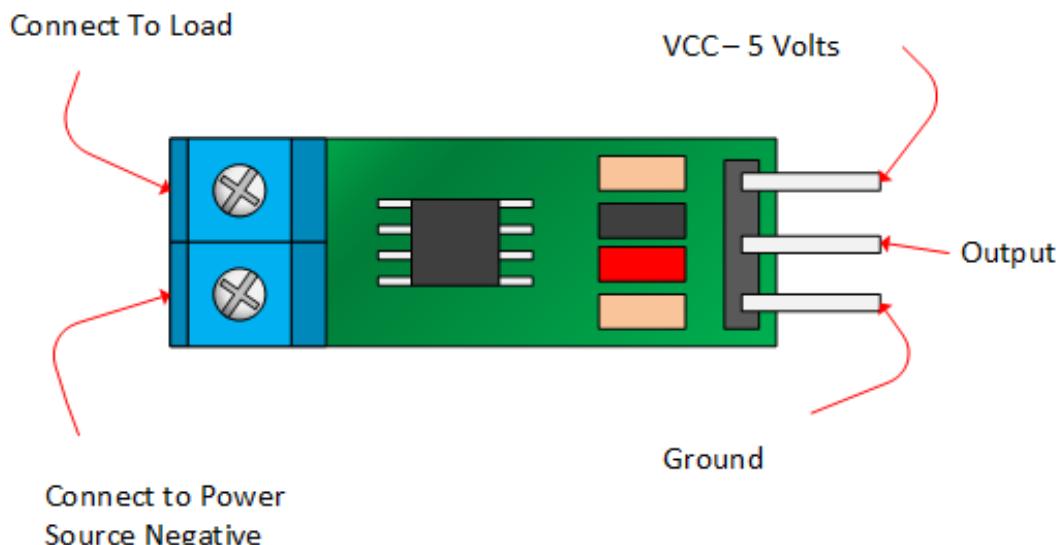
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Các thông số kỹ thuật của module ACS712 -20A:

- Đường tín hiệu analog có độ nhiễu thấp
- Thời gian tăng của đầu ra để đáp ứng với đầu vào là  $5\mu s$
- Băng thông 80 Khz
- Tổng lỗi ngõ ra tại  $T_a = 25^{\circ}C$  là 1,5%
- Điện trở dây dẫn trong là  $1,2m\Omega$
- Nguồn vận hành đơn : 5V
- Dòng tiêu thụ (max): 13mA
- Độ nhạy đầu ra từ 96 – 104mV/A
- Điện áp ngõ ra tương ứng với dòng AC hoặc DC
- Điện áp ngõ ra cực kỳ ổn định
- $I_p$ : từ -20A đến 20A
- Nhiệt độ hoạt động từ -40 đến  $85^{\circ}C$

Ngoài ra còn có các loại cảm biến dòng khác như :

- ACS712 – 5A: khoảng đo từ -5A đến 5A, độ nhạy điện áp 180 - 190mV/A
- ACS712 – 30A: khoảng đo từ -30A đến 30A, độ nhạy điện áp 64 - 68mV/A



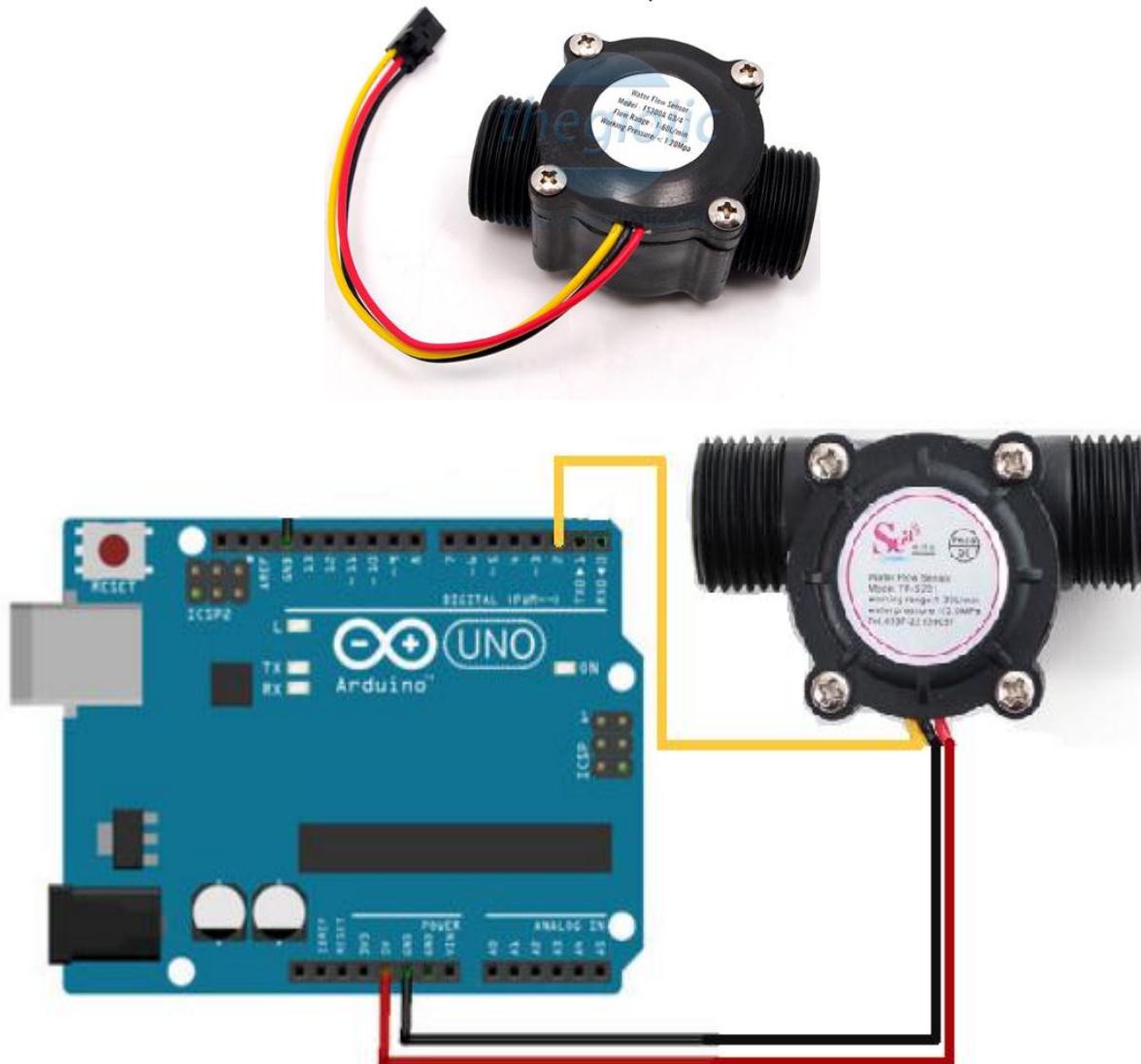
Hình 2.2 Hình ảnh minh họa kết nối với module ACS712

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1.1.2 Cảm biến lưu lượng S201

Ta có thể sử dụng đồng hồ đo nước thông dụng để biết được lượng nước sinh hoạt đã sử dụng. Tuy nhiên, để có thể giám sát ở bất cứ đâu thông qua internet ứng dụng công nghệ IoT, ta cần phải sử dụng một cảm biến chuyên dụng, để đọc và gửi dữ liệu từ cảm biến về vi bộ xử lý. Trong đề này sử dụng cảm biến lưu lượng S201 để đo.

Cảm biến S201 bên trong có chứa một cánh quạt để đếm lượng chất lỏng chảy qua nó và có một cảm biến từ Hall xuất ra các xung khi có sự thay đổi trạng thái đầu ra. Cảm biến Hall được hàn kín trong ống để được an toàn và khô ráo.



Hình 2.3 Hình ảnh của cảm biến lưu lượng S201 và kết nối

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Cảm biến lưu lượng có 3 dây:

- Dây đỏ: cấp nguồn 5V – 24VDC
- Dây đen: GND
- Dây vàng: ngõ ra của cảm biến Hall

Thông số kỹ thuật của cảm biến S201:

- Điện áp làm việc: 5V – 24VDC
- Loại ngõ ra: 5V TTL
- Dòng điện cao nhất: 15mA (5V)
- Mức độ dòng chảy: từ 1 đến 30 L/phút
- Vận hành ở nhiệt độ: -25°C – 80°C
- Nhiệt độ dòng chảy: <120°C
- Áp lực nước tối đa: 2Mpa
- Vận hành ở độ ẩm: 35% - 80% RH
- Sai số: 10%
- Số xung trên lít: 450
- Thời gian xung ngõ ra ở mức cao: 0,04μs
- Thời gian xung ngõ ra ở mức thấp: 0,18 μs

Với các tín hiệu xung ra là một dải xung vuông đơn giản, ta có thể dễ dàng đọc và tính được lưu lượng nước bằng việc đếm xung từ ngõ ra của cảm biến theo công thức sau:

$$\text{Tần số xung (Hz) / 7,5} = \text{tốc độ dòng chảy (L / phút)}$$

Một số lưu ý khi sử dụng:

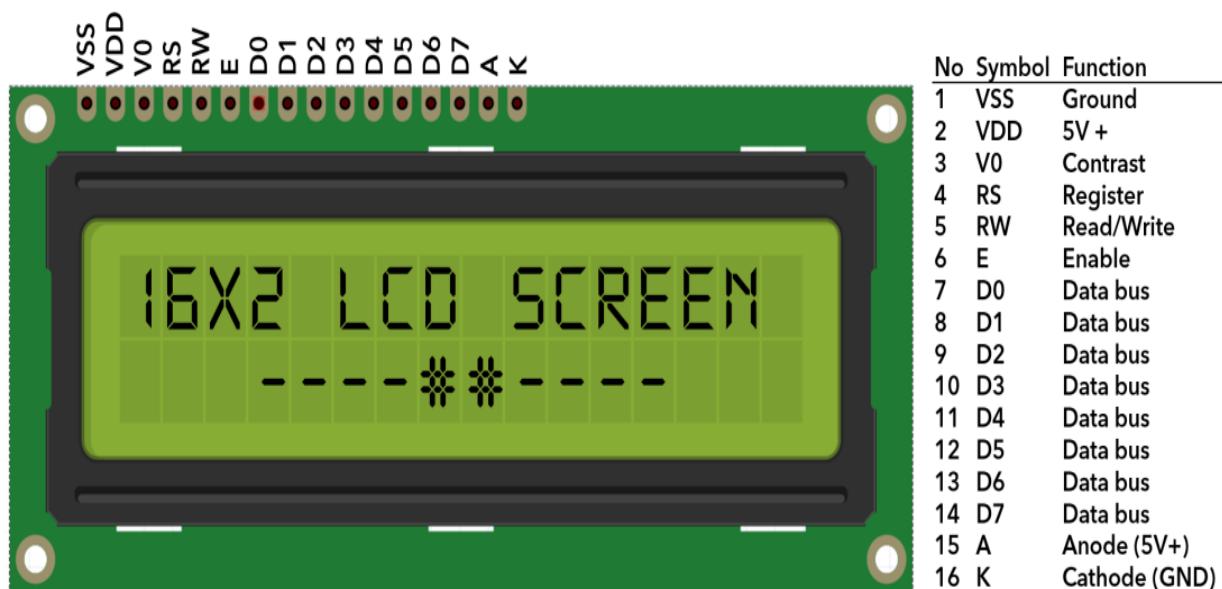
- Nên đặt cảm biến ở trên cùng dòng chảy
- Không cho dòng chảy có chất hóa học, ăn mòn
- Không chịu va đập khi sử dụng
- Đặt cảm biến thẳng đứng không lệch quá 5 độ
- Nhiệt độ nước chảy qua dưới 120 độ C

### 2.1.2 Thiết bị đầu ra – Màn hình LCD 16x2 và mạch LCD I2C

#### 2.1.2.1 LCD 16x2

LCD (Liquid Crystal Display) được sử dụng rất nhiều trong các ứng dụng vi điều khiển, nó có nhiều ưu điểm so với các dạng hiển thị khác như: khả năng hiển thị ký tự đa dạng trực quang, dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành lại rẻ.

Trên thị trường có rất nhiều loại LCD đa dạng về kích cỡ và hình dáng, như hình 2.4 bên dưới, đây là một loại LCD rất thông dụng. Bên trong lớp vỏ của LCD được tích hợp chip điều khiển HD44780 và chỉ đưa các chân giao tiếp cần thiết ra bên ngoài, và được ghi chú hướng dẫn cụ thể.



Hình 2.4 Sơ đồ chân của LCD 16x2

#### Thông số kỹ thuật của LCD 16x2:

- Điện áp lớn nhất: 7V
- Điện áp nhỏ nhất: - 0,3V
- Điện áp hoạt động: 2,7V - 5,5V
- Dòng điện cấp nguồn: 350 $\mu$ A - 600 $\mu$ A
- Nhiệt độ hoạt động: - 30 - 75°C

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

### Chức năng các chân LCD 16x2:

1. VSS: Chân nối đất của LCD, trong thiết kế ta nối chân này chung với GND của mạch điều khiển
2. VDD: Chân cấp nguồn của LCD, ta nối chung chân này với VCC (5V) của mạch điều khiển
3. Vo: Có chức năng điều chỉnh độ tương phản cho LCD
4. RS: Chân chọn thanh ghi. Nối chân RS với mức logic 0 hoặc 1 để chọn thanh ghi:
  - Logic 0: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD ở chế độ ghi – write, hoặc nối với bộ đếm địa chỉ của LCD ở chế độ đọc – read
  - Logic 1: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
5. R/W: Chân chọn chế độ đọc/ghi (R/W). Ở mức logic 0, LCD hoạt động ở chế độ ghi; và ở mức logic 1, LCD ở chế độ đọc
6. E: Đây là chân cho phép của LCD. Khi tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép ở chân E
  - Ở chế độ đọc: LCD xuất ra dữ liệu DB0-DB7 khi chân E xuất hiện cạnh lên (low-to-high) và LCD sẽ giữ ở bus cho đến khi nào chân E quay về lại mức thấp
  - Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào thanh ghi bên trong khi chân E xuất hiện xung cạnh xuống (high-to-low)
- 7-14. DB0-DB7: 8 đường bus này dùng để trao đổi dữ liệu với MCU theo 2 chế độ:
  - Chế độ 8 bit : Dữ liệu truyền trên cả 8 đường, với bit MSB là bit DB7
  - Chế độ 4 bit : Dữ liệu truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7
15. A: Nguồn dương cho đèn nền
16. K: Nguồn âm cho đèn nền

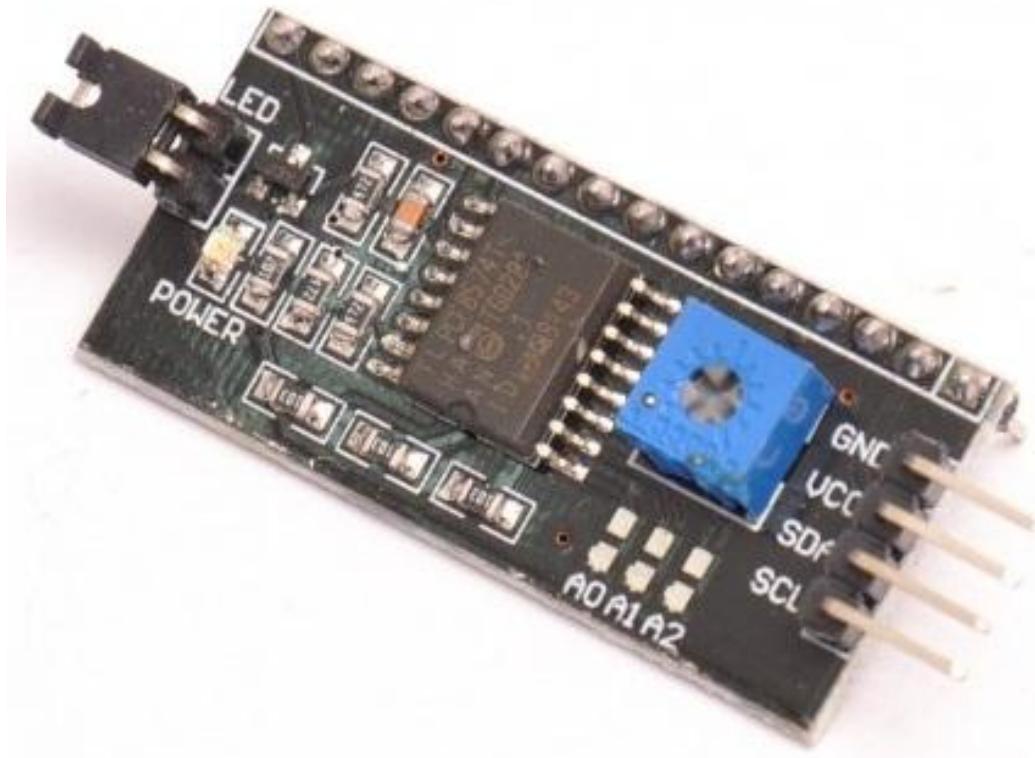
#### 2.1.2.2 Module giao tiếp LCD I2C

Thông thường, các vi điều khiển sẽ cần ít nhất 6 chân để thực hiện việc điều khiển hiển thị LCD 16x2, đây có thể là một sự lãng phí; để tiết kiệm số chân, ta dùng một loại mạch điều khiển LCD giao tiếp theo chuẩn I2C, và ta chỉ cần sử dụng 2 chân của vi điều

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

khiến để điều khiển màn hình. Ta cần tải thư viện LCD I2C về, thêm vào Arduino IDE để sử dụng.



**Hình 2.5** Module giao tiếp LCD I2C

### Kết nối LCD với Nodemcu

**Bảng 2.1** Kết nối LCD với NodeMCU

Module màn hình LCD (16x2)	NodeMCU
GND	GND
Vcc	(Nguồn riêng 5V)
SDA	D1
SCL	D0

### 2.1.3 Arduino Mega 2560

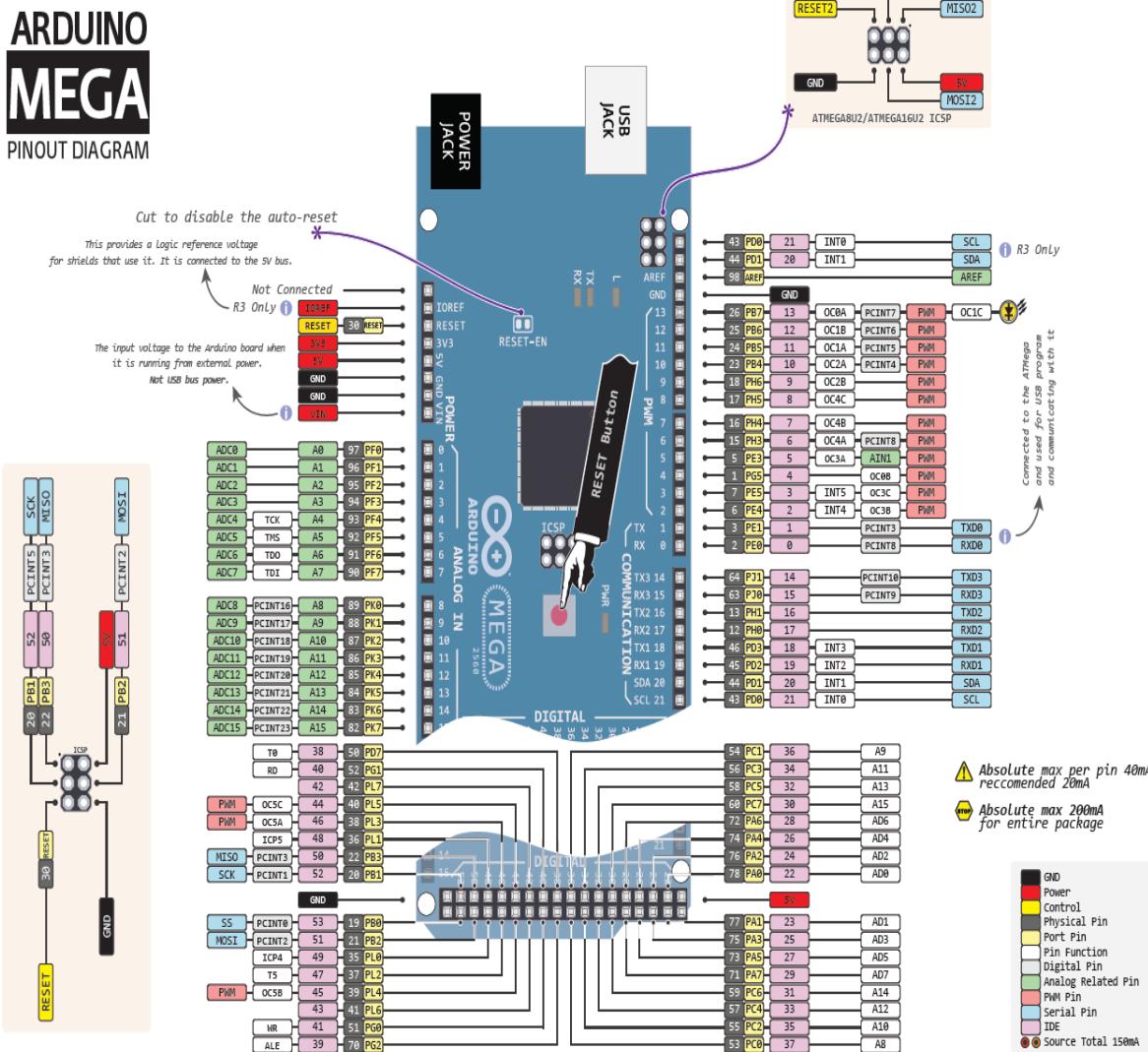
#### 2.1.3.1 Giới thiệu

Nếu cần tìm kiếm một bộ vi xử lý được hỗ trợ mạnh mẽ với mã nguồn mở, có một cộng đồng sử dụng rộng lớn, ngôn ngữ lập trình dễ tiếp cận, và giá thành hợp lý, thì Arduino chính là thứ mà bạn đang tìm kiếm. Arduino trải qua rất nhiều phiên bản cải tiến, mở rộng và một trong những phiên bản được sử dụng rộng rãi là Arduino Mega.

Arduino Mega 2560 là một phiên bản nâng cấp của Arduino Uno R3 với số chân giao tiếp, ngoại vi và bộ nhớ nhiều hơn; phù hợp cho các ứng dụng cần nhiều bộ nhớ hoặc nhiều chân, cổng giao tiếp hơn so với Arduino Uno. Arduino Mega 2560 cung cấp mọi thứ cần thiết để hỗ trợ vi điều khiển; chỉ cần cáp USB kết nối đến máy vi tính hoặc cáp điện bằng bộ điều hợp AC-DC hoặc dùng pin để cấp nguồn là ta đã có thể sử dụng được. Arduino Mega 2560 tương thích với hầu hết các “shield”.

#### 2.1.3.2 Thông số kỹ thuật

- Vi điều khiển chính: Atmega2560
- IC nạp và giao tiếp UART: Atmega16U2
- Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn bên ngoài
- Số chân Digital: 54 (15 chân PWM)
- Số chân Analog: 16
- Giao tiếp UART: 4 bộ UART
- Giao tiếp SPI: 1 bộ (chân 50 – 53), dùng với thư viện SPI của Arduino
- Giao tiếp I2C: 1 bộ
- Ngắt ngoài: 6 chân
- Bộ nhớ Flash: 256kb; 8kb sử dụng cho Bootloader
- SRAM: 8kb
- EEPROM: 4kb
- Xung clock: 16Mhz
- 1 nút nhấn reset board



Hình 2.6 Thành phần Arduino Mega 2560

### 2.1.4 NodeMCU 1.0

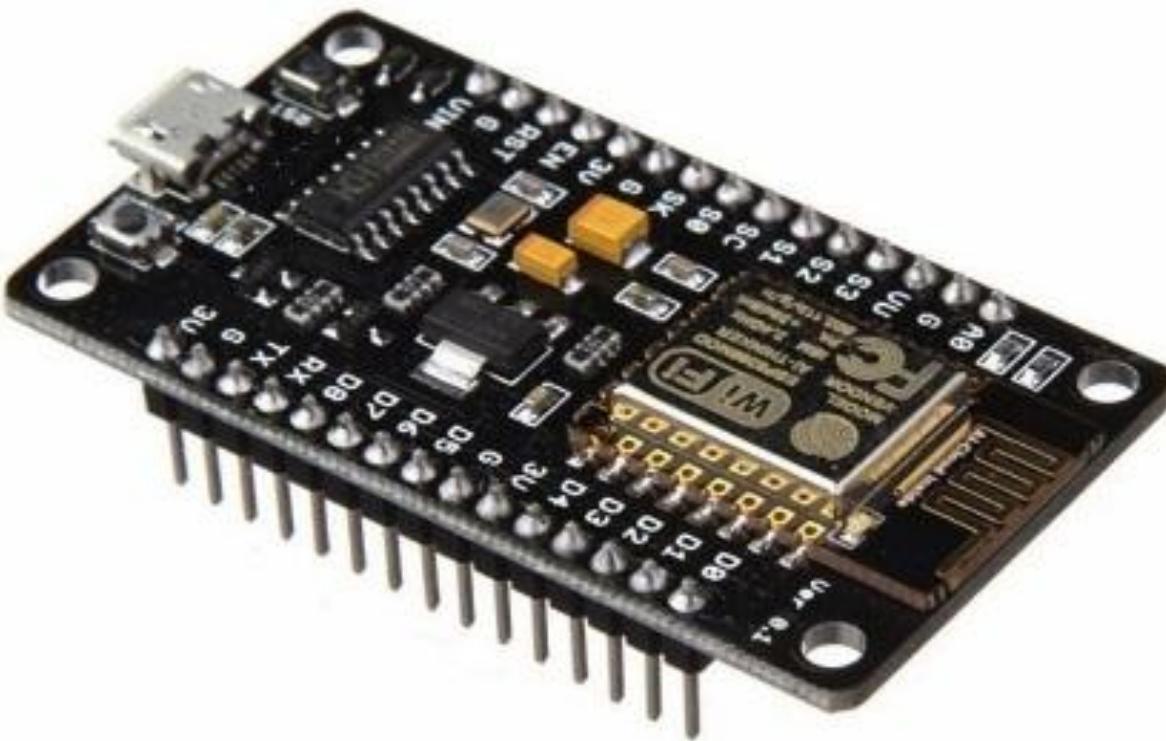
#### 2.1.4.1 Giới thiệu:

NodeMCU V1.0 là một dạng vi điều khiển có tích hợp Wifi, được phát triển dựa trên chip Wifi ESP8266EX bên trong module ESP – 12E, cho phép dễ dàng kết nối wifi với một vài thao tác. Board còn tích hợp IC CP2102, giúp dễ dàng giao tiếp máy tính thông qua Micro USB để thao tác với board. NodeMCU giúp lập trình viên thực hiện các tác vụ TCP/IP đơn giản để xây dựng các ứng dụng khác nhau, đặc biệt là ứng dụng IoT.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Với việc sử dụng và kết nối dễ dàng, có thể lập trình và nạp chương trình trực tiếp trên phần mềm Arduino IDE; Có rất nhiều thư viện và tài liệu hỗ trợ người dùng, NodeMCU cùng với Arduino chính là công cụ thúc đẩy cho các ứng dụng về lĩnh vực IoT phát triển mạnh mẽ như ngày hôm nay.



**Hình 2.7 NodeMCU 1.0 (ESP-12E Module)**

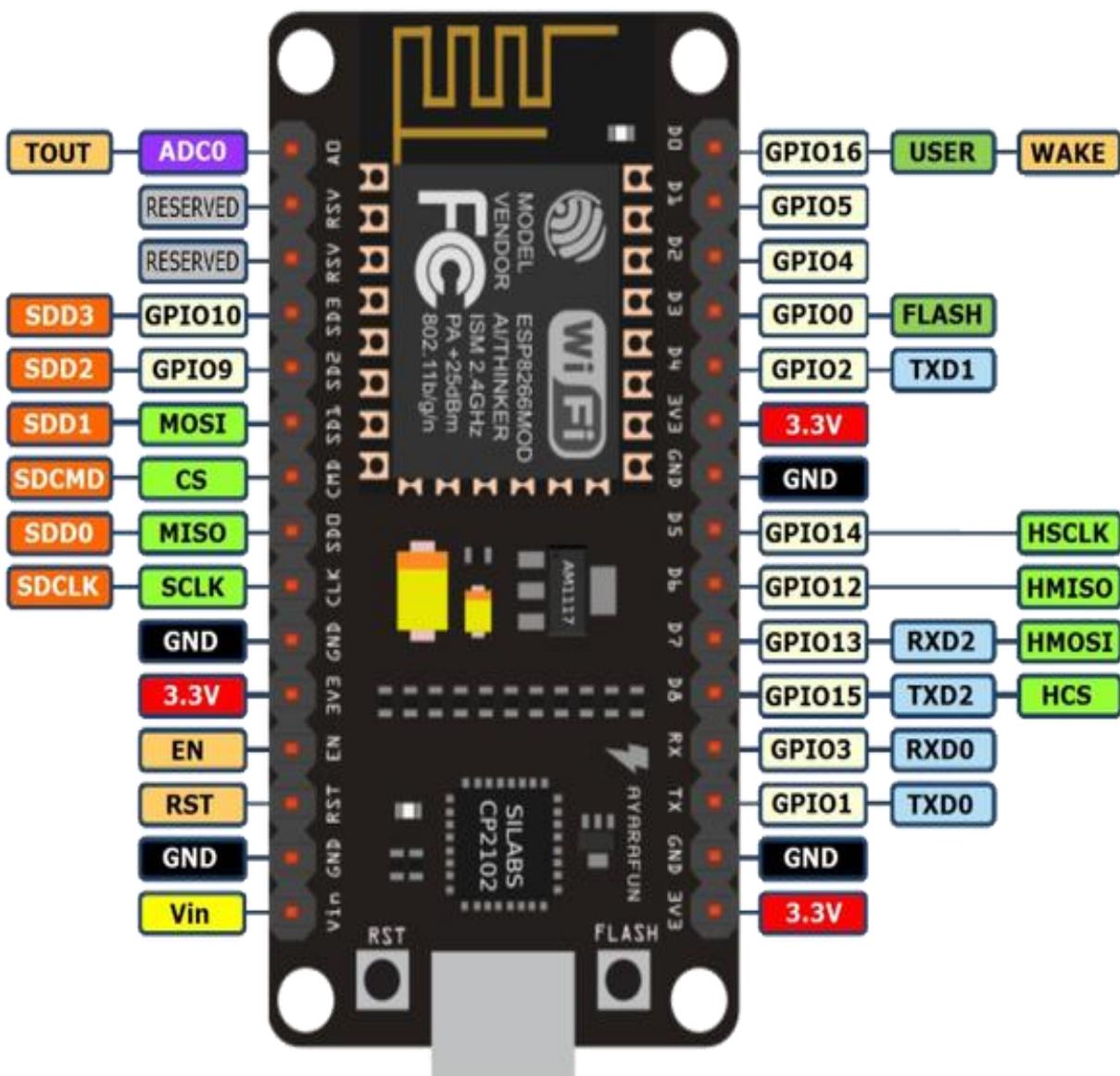
Lưu ý là nếu gặp lỗi không giao tiếp được với NodeMCU, ta nên kiểm tra lại cable (vì có một số loại cable không truyền dữ liệu được), và cài đặt driver xem đã đúng chưa. Khi test mạch, ta cần để ý đến nguồn, tránh việc sơ ý bị chòng nguồn gây cháy module.

### 2.1.4.2 Thông số kỹ thuật

- Chip: ESP8266EX
- Wifi: 2,4Ghz hỗ trợ chuẩn 802,11b/g/n
- Điện áp hoạt động: 3.3V
- Điện áp vào: 5V (qua cáp USB) hoặc nguồn bên ngoài
- Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Số chân analog input: 1 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4Mb
- Giao tiếp: Cable Micro USB
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp giao thức: TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, MicroPython, NodeMCU – LUA



Hình 2.8 Sơ đồ chân NodeMCU 1.0 (ESP-12E Module)

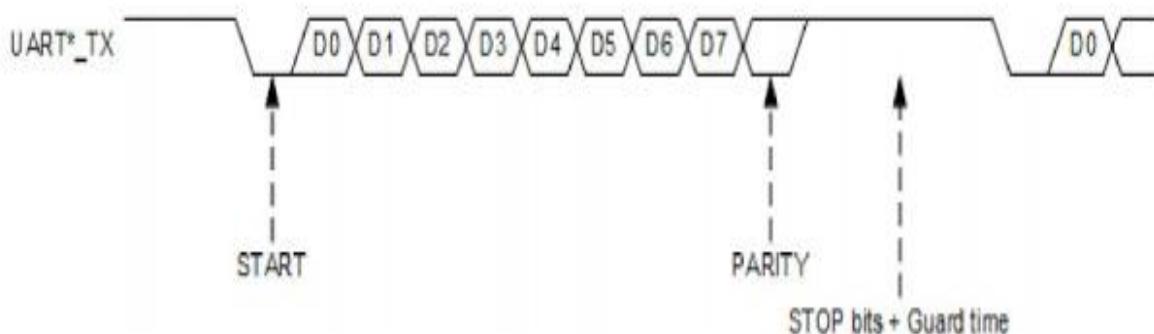
### 2.2 CHUẨN TRUYỀN DỮ LIỆU

#### 2.2.1 Giao tiếp UART

##### 2.2.1.1 Giới thiệu

Hiện nay, chuẩn UART (Universal Asynchronous Receiver – Transmitter) được sử dụng rất nhiều trong các board mạch điều khiển để truyền nhận dữ liệu giữa các thiết bị với nhau. Rất nhiều vi điều khiển hiện nay được tích hợp UART. Khác với giao tiếp SPI và I2C cần phải có 1 dây truyền dữ liệu và 1 dây dùng để truyền xung clock (SCL) cho đồng bộ; đối với chuẩn UART thì không sử dụng dây SCL, công việc truyền UART được thực hiện giữa 2 bộ vi xử lý, và mỗi vi xử lý đều tự tạo ra xung clock.

Để thực hiện công việc truyền dữ liệu, bắt đầu bằng việc gửi đi một bit START, tiếp theo là các bit dữ liệu và cuối cùng là bit stop để kết thúc.



**Hình 2.9** Truyền dữ liệu bằng chuẩn UART

Khi chưa truyền dữ liệu thì ban đầu điện thế ở mức logic 1 (mức cao). Khi bắt đầu truyền dữ liệu, bit START chuyển từ mức logic 1 về logic 0, báo cho bộ nhận là việc truyền dữ liệu bắt đầu được thực hiện. Tiếp theo là truyền đi các bit dữ liệu D0-D7 (có thể là logic 1 hoặc 0). Bộ nhận sẽ kiểm tra tính đúng đắn của dữ liệu truyền đi dựa theo bit PARITY (kiểm tra chẵn/lẻ). Cuối cùng bit STOP sẽ báo cho thiết bị rằng dữ liệu đã được gửi đi hoàn tất.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Tất cả các board Arduino đều có ít nhất 1 cổng UART hoặc USART. Cổng giao tiếp UART trên chân TX/RX sử dụng mức logic TTL (5V hoặc 3,3V) để giao tiếp với máy tính hay các thiết bị khác. Nếu đã sử dụng 2 chân TX/RX này thì không thể dùng với mục đích input/output của Arduino nữa.

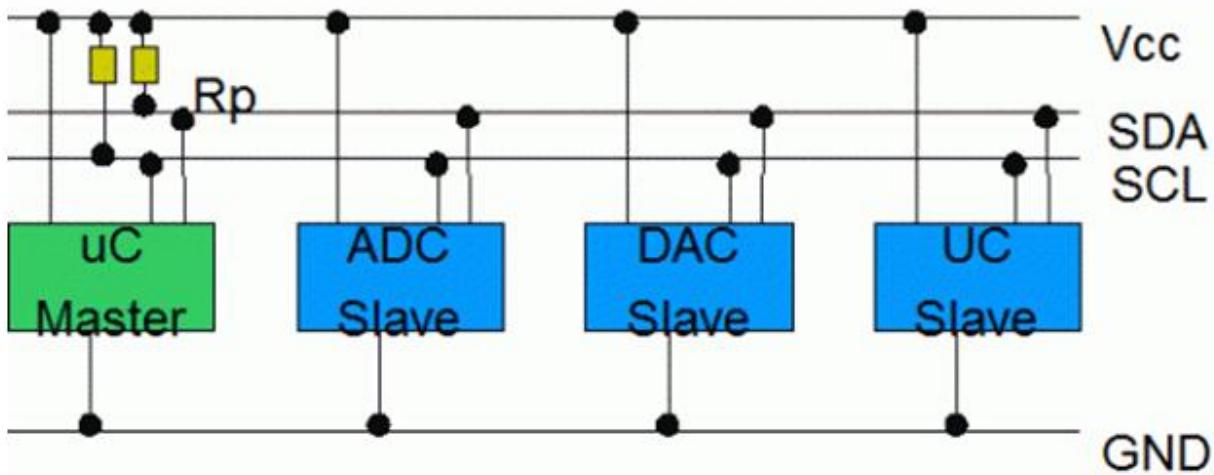
### 2.2.1.2 Các thông số trong truyền nhận UART

- **Baud rate:** Hay còn gọi là tốc độ Baud, đây là khoảng thời gian của 1 bit được truyền đi. Lưu ý là phải được cài đặt giống nhau ở thiết bị gửi và nhận.
- **Frame:** Khung truyền quy định về số bit trong mỗi lần truyền.
- **Bit start:** đây là bit đầu tiên truyền đi trong một Frame để báo hiệu cho thiết bị nhận sẽ có dữ liệu sắp được truyền đến.
- **Data:** đây là dữ liệu cần gửi; bit trọng số nhỏ nhất (LSB) được truyền đi trước, và cuối cùng là bit MSB.
- **Parity bit:** kiểm tra tính chẵn/lẻ của dữ liệu được truyền đi.
- **Stop bit:** đây là bit báo cho thiết bị nhận rằng việc gửi dữ liệu đi đã hoàn tất. Thiết bị nhận sẽ kiểm tra khung truyền nhằm đảm bảo tính đúng đắn của dữ liệu.

### 2.2.2 Chuẩn giao tiếp I2C

#### 2.2.2.1 Giới thiệu

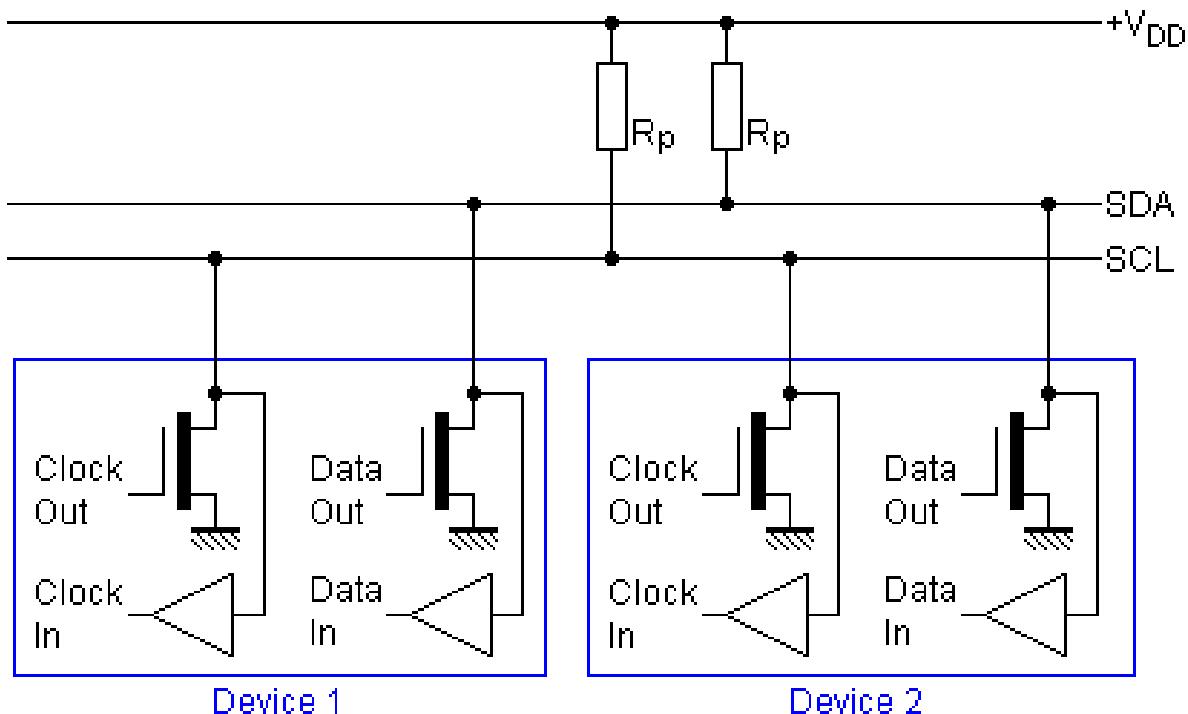
Đầu năm 1980 Phillips đã phát triển một chuẩn giao tiếp 2 dây được gọi là I2C. I2C là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. I2C mặc dù được phát triển bởi Philips, nhưng nó đã được rất nhiều nhà sản xuất IC trên thế giới sử dụng. I2C trở thành một chuẩn công nghiệp cho các giao tiếp điều khiển, có thể kể ra đây một vài tên tuổi ngoài Philips như: Texas Instrument(TI), MaximDallas, analog Device, National Semiconductor ... Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM... chip nhớ như: RAM tĩnh (Static Ram), EEPROM, bộ chuyển đổi tương tự số (ADC), số tương tự(DAC), IC điều khiển LCD, LED...



**Hình 2.10** Thiết bị ngoại vi giao tiếp bus I2C

### 2.2.2.2 Đặc điểm giao tiếp I2C

Một giao tiếp I2C gồm có 2 dây là Serial Data (SDA) và Serial Clock (SCL). SDA truyền dữ liệu theo 2 hướng, còn SCL là đường truyền một hướng để truyền xung clock đồng bộ.



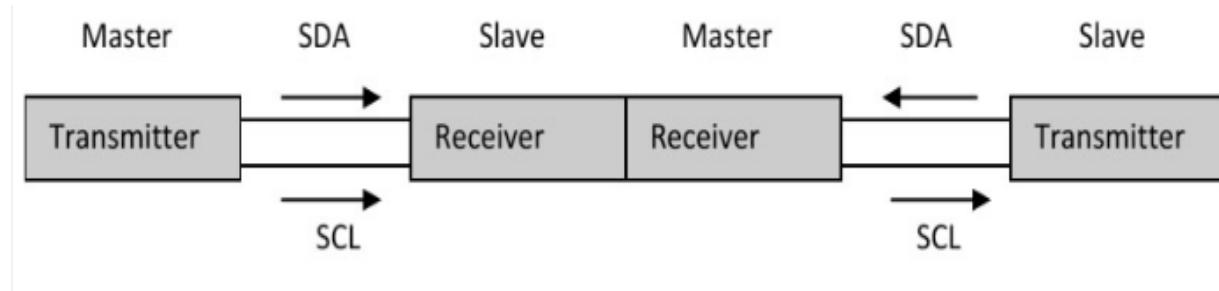
**Hình 2.11** Thiết bị kết nối vào I2C ở chế độ chuẩn và chế độ nhanh

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Mỗi dây SDA và SCL đều nối với điện áp dương của nguồn thông qua một điện trở kéo lên. Giá trị của các điện trở này khác nhau tùy thuộc vào từng thiết bị và chuẩn giao tiếp.

Ở hình 2.10 bên trên, có rất nhiều thiết bị kết cùng kết nối vào một bus, tuy nhiên sẽ không xảy ra trường hợp nhầm lẫn giữa các thiết bị, vì mỗi thiết bị sẽ được nhận ra bởi một địa chỉ duy nhất, với một quan hệ chủ/tớ tồn tại trong suốt quá trình kết nối. Một thiết bị có thể hoạt động như một thiết bị truyền hay nhận dữ liệu hoặc vừa truyền vừa nhận.

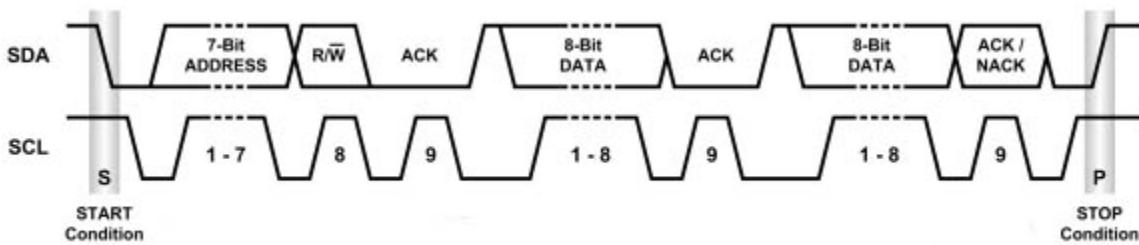
Một thiết bị hay IC khi kết nối với I2C, ngoài địa chỉ duy nhất để phân biệt, nó còn được cấu hình là một thiết bị chủ hay tớ - với quyền điều khiển thuộc về thiết bị chủ. Khi giữa 2 thiết bị chủ - tớ giao tiếp, thì thiết bị chủ có vai trò tạo xung clock và quản lý địa chỉ của thiết bị tớ trong suốt quá trình giao tiếp.



Hình 2.12 Quá trình giao tiếp giữa thiết bị chủ - tớ

Như hình trên, xung đồng hồ chỉ đi một hướng từ chủ đến tớ, còn luồng dữ liệu có thể đi theo 2 hướng.

### 2.2.2.3 Trình tự truyền bit trên đường truyền



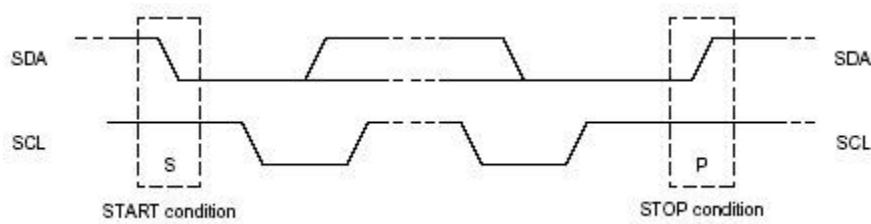
Hình 2.13 Trình tự truyền dữ liệu

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tớ lắng nghe dữ liệu trên đường truyền. Sau đó, thiết bị chủ sẽ gửi đi một địa chỉ của thiết bị tớ mà thiết bị chủ muốn giao tiếp và cờ đọc/ghi dữ liệu. Thiết bị tớ mang địa chỉ đó trên bus I2C sẽ phản hồi lại bằng một xung ACK. Khi đó, việc giao tiếp giữa thiết bị chủ - tớ bắt đầu. Bộ truyền gửi 8 bit dữ liệu đến bộ nhận, bộ nhận trả lời với 1 bit ACK. Để kết thúc, thiết bị chủ tạo ra một điều kiện STOP.

### 2.2.2.4 Điều kiện START và STOP

START là điều kiện khởi đầu, báo hiệu bắt đầu của giao tiếp, và ngược lại, STOP báo hiệu kết thúc.



**Hình 2.14** Giản đồ thời gian điều kiện START và STOP

Khi chưa thực hiện giao tiếp, cả SDA và SCL đều ở mức cao.

- Điều kiện START: sự chuyển đổi từ mức logic cao xuống thấp trên đường SDA trong khi SCL vẫn đang ở mức cao báo hiệu một điều kiện START.
- Điều kiện STOP: sự chuyển đổi trạng thái từ mức logic thấp lên cao trên đường SDA trong khi đường SCL đang ở mức cao. Cả hai điều kiện START, STOP đều được tạo ra bởi thiết bị chủ. Sau tín hiệu START, bus I2C xem như đang ở trong trạng thái làm việc. Bus I2C sẽ sẵn sàng cho một giao tiếp mới sau tín hiệu STOP từ phía máy chủ.

Sau khi có một điều kiện START, trong quá trình giao tiếp, khi có một tín hiệu START được lặp lại thay vì một tín hiệu STOP thì bus I2C vẫn tiếp tục trong trạng thái bận. Tín hiệu START và lặp lại START (Repeated START) đều có chức năng giống nhau là khởi tạo một giao tiếp.

### 2.2.3 Chuẩn giao tiếp Wifi

#### 2.2.3.1 Giới thiệu

Wifi là viết tắt của Wireless Fidelity, được gọi chung là mạng không dây sử dụng sóng vô tuyến, loại sóng vô tuyến này tương tự như sóng truyền hình, điện thoại và radio. Wifi phát sóng trong phạm vi nhất định, các thiết bị điện tử tiêu dùng ngày nay như laptop, smartphone hoặc máy tính bảng có thể kết nối và truy cập internet trong tầm phủ sóng.

#### 2.2.3.2 Nguyên tắc hoạt động

Để tạo được kết nối Wifi nhất thiết phải có Router (bộ thu phát), Router này lấy thông tin từ mạng Internet qua kết nối hữu tuyến rồi chuyển nó sang tín hiệu vô tuyến và gửi đi, bộ chuyển tín hiệu không dây (adapter) trên các thiết bị di động thu nhận tín hiệu này rồi giải mã nó sang những dữ liệu cần thiết. Quá trình này có thể thực hiện ngược lại, Router nhận tín hiệu vô tuyến từ Adapter và giải mã chúng rồi gởi qua Internet.



Hình 2.15 Mô hình hoạt động của mạng Wifi

### 2.2.3.3 Một số chuẩn kết nối Wifi

Tuy nói wifi tương tự như sóng vô tuyến truyền hình, radio hay điện thoại nhưng nó vẫn khác các loại sóng kia ở mức độ tần số hoạt động.

Sóng wifi truyền nhận dữ liệu ở tần số 2,5Ghz đến 5Ghz. Tần số cao này cho phép nó mang nhiều dữ liệu hơn nhưng phạm vi truyền của nó bị giới hạn; còn các loại sóng khác, tuy tần số thấp nhưng có thể truyền đi được rất xa.

Kết nối 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 a/b/g/n:

- Chuẩn wifi đầu tiên 802.11: năm 1997, IEEE đã giới thiệu chuẩn đầu tiên này cho WLAN. Tuy nhiên, 802.11 chỉ hỗ trợ cho băng tần mạng cực đại lên đến 2Mbps – quá chậm đối với hầu hết mọi ứng dụng. Và với lý do đó, các sản phẩm không dây thiết kế theo chuẩn 802.11 ban đầu không được sản xuất nữa.
- Chuẩn wifi 802.11b: IEEE đã mở rộng trên chuẩn gốc 802.11 để tạo ra chuẩn 802.11b vào tháng 7/1999. Chuẩn này hỗ trợ băng thông lên đến 11Mbps, tương ứng với Ethernet truyền thông.
  - Chuẩn này sử dụng tần số tín hiệu vô tuyến không được kiểm soát 2.4Ghz, các nhà sản xuất rất thích sử dụng tần số này để giảm chi phí sản xuất. Tuy nhiên, các thiết bị 802.11b có thể bị xuyên nhiễu từ các thiết bị điện thoại không dây, lò vi sóng hoặc các thiết bị khác sử dụng cùng dải tần số. Tuy nhiên, ta có thể giảm được hiện tượng xuyên nhiễu này bằng cách lắp thiết bị 802.11b cách xa các thiết bị như vậy
  - Ưu điểm của 802.11b: giá thành thấp nhất, phạm vi tín hiệu tốt và không dễ bị cản trở.
  - Nhược điểm của 802.11b: tốc độ tối đa thấp nhất, các thiết bị gia dụng có thể gây cản trở.
- Chuẩn wifi 802.11a: trong khi 802.11b vẫn đang được phát triển, IEEE đã tạo ra một mở rộng thứ 2 có tên gọi là 802.11a. Do giá thành cao hơn nên 802.11a thường

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

được sử dụng cho các mạng doanh nghiệp, còn 802.11b thích hợp hơn cho các hộ gia đình.

- 802.11a hỗ trợ băng thông lên đến 54Mbps và sử dụng tần số 5Ghz. Do 802.11a và 802.11b sử dụng 2 tần số khác nhau, nên 2 công nghệ này không thể tương thích với nhau. Do đó, một số hãng đã cung cấp các thiết bị mạng lai cho 802.11a/b, nhưng sản phẩm này chỉ đơn thuần là thực hiện 2 chuẩn song song.
- Ưu điểm: tốc độ cực nhanh, tần số được kiểm soát nên tránh được sự xuyên nhiễu từ các thiết bị khác.
- Nhược điểm: giá thành đắt, phạm vi hẹp và dễ bị cản trở.
- Chuẩn wifi 802.11g: vào năm 2002 và 2003, các sản phẩm WLAN hỗ trợ một chuẩn mới hơn đó là 802.11g, rất được đánh giá cao trên thị trường. Đây là một nỗ lực kết hợp ưu điểm của cả 802.11a và 802.11b, hỗ trợ băng thông lên đến 54Mbps và sử dụng tần số 2.4Ghz để có phạm vi rộng.
  - Ưu điểm: tốc độ cực nhanh, phạm vi tín hiệu tốt và ít bị cản trở.
  - Nhược điểm: giá thành đắt hơn 802.11b, các thiết bị có thể dễ bị xuyên nhiễu từ những đồ gia dụng sử dụng cùng tần số tín hiệu vô tuyến không được kiểm soát.
- Chuẩn wifi 802.11n: 802.11n - đôi khi được gọi tắt là wireless, được thiết kế để cải thiện cho 802.11g trong tổng số băng thông được hỗ trợ bằng cách tận dụng nhiều tín hiệu không dây và anten. Được phê chuẩn vào năm 2009, với băng thông tối đa lên đến 600Mbps, 802.11n cũng cung cấp phạm vi tốt hơn những chuẩn wifi trước đó, do cường độ tín hiệu của nó đã tăng lên.
  - Ưu điểm: tốc độ tối đa nhanh nhất, phạm vi tín hiệu tốt nhất và khả năng chống nhiễu tốt hơn từ các nguồn bên ngoài.
  - Nhược điểm: giá thành đắt hơn 802.11g, việc sử dụng nhiều tín hiệu có thể gây nhiễu với các mạng dựa trên chuẩn 802.11b và 802.11g ở gần.
- Chuẩn wifi 802.11ac: đây là chuẩn wifi lớn nhất, được sử dụng phổ biến nhất hiện nay. 802.11ac sử dụng công nghệ không dây băng tần kép, hỗ trợ các kết nối đồng

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

thời trên cả băng tần 2.4Ghz và 5Ghz. 802.11ac có băng thông đạt tới 1.300Mbps trên băng tần 5Ghz và 450Mbps trên 2.4Ghz.

CÁC CHUẨN WIFI 802.11					
Chuẩn IEEE	802.11a	802.11b	802.11g	802.11n	802.11ac
Năm phát hành	1999	1999	2003	2009	2013
Tần số	5 GHz	2.4 GHz	2.4 GHz	2.4/5 GHz	5 GHz
Tốc độ tối đa	54 Mbps	11 Mbps	54 Mbps	600 Mbps	1 Gbps
Phạm vi trong nhà	100 ft.	100 ft.	125 ft.	225 ft.	90 ft.
Phạm vi ngoài trời	400 ft.	450 ft.	450 ft.	825 ft.	1,000 ft.

Hình 2.16 Bảng so sánh thông số các chuẩn wifi

### 2.3 FIREBASE HOSTING

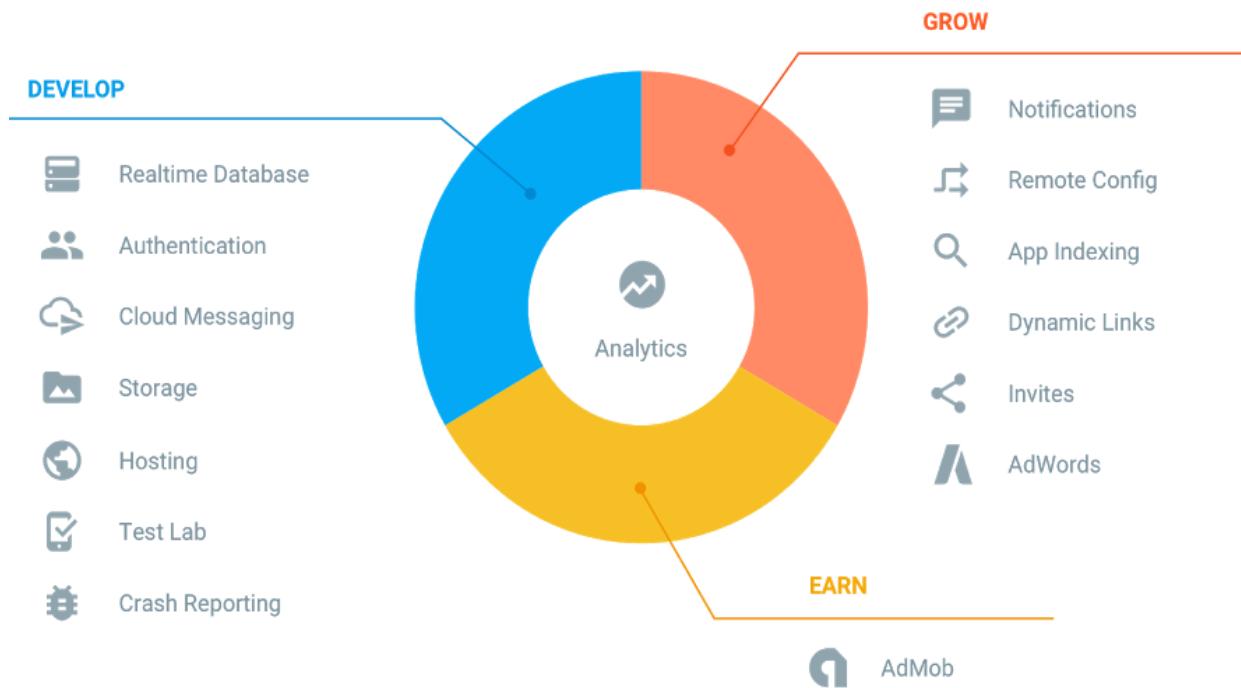
#### 2.3.1 Giới thiệu

Firebase là một nền tảng ứng dụng di động và web, với các “công cụ và hạ tầng” được thiết kế giúp cho các lập trình viên xây dựng các ứng dụng chất lượng cao. Nói cách khác, thay vì trực tiếp cung cấp các ứng dụng, Firebase cung cấp các dịch vụ nền tảng, giúp cho bạn xây dựng các ứng dụng cũng như hỗ trợ để tối ưu hóa, tối đa hóa các ứng dụng của mình.

Firebase (tiền thân là Evolve) trước đây là một start up được thành lập năm 2011 bởi Andrew Lee và James Tamplin, đầu tiên chỉ cung cấp cơ sở dữ liệu cho các lập trình viên xây dựng các ứng dụng chat. Sau đó, họ nhìn thấy được tiềm năng sản phẩm của mình, họ đã tách mảng realtime ra để thành lập một công ty độc lập, là Firebase vào tháng 4/2012. Vào tháng 10/2014, Firebase gia nhập “kẻ khổng lồ” Google.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Từ đó, Firebase có điều kiện phát triển thần tốc, mở rộng số lượng các dịch vụ con, còn Google có được một đội ngũ nhân viên chất lượng cao, cũng như cơ sở hạ tầng và sự hiệu quả từ nền tảng mà Firebase mang lại. Firebase, theo hướng đi của Google, chính thức hỗ trợ cho cả 3 nền tảng Android, IOS và Web.



**Hình 2.17** Các dịch vụ hỗ trợ của Firebase

### 2.3.2 Ưu điểm của Firebase

- **Phục vụ kết nối an toàn:** tất cả các project mà bạn xây dựng sẽ được bảo vệ. Nội dung được truyền tải một cách an toàn với việc tích hợp cấu hình Zero SSL.
- **Truyền dữ liệu đi một cách nhanh chóng:** dữ liệu của bạn khi được tải lên sẽ được lưu trữ ở ổ SSD, và dữ liệu sẽ được truyền một cách nhanh chóng bất kể bạn đang có mặt ở đâu.
- **Triển khai dự án nhanh chóng:** có thể upload file lên project và chạy ứng dụng ngay lập tức với việc sử dụng Firebase CLI.
- **Hỗ trợ khôi phục dữ liệu:** có thể khôi phục lại các file dữ liệu đã tải lên trước đó.

### 2.4 FIREBASE REALTIME DATABASE

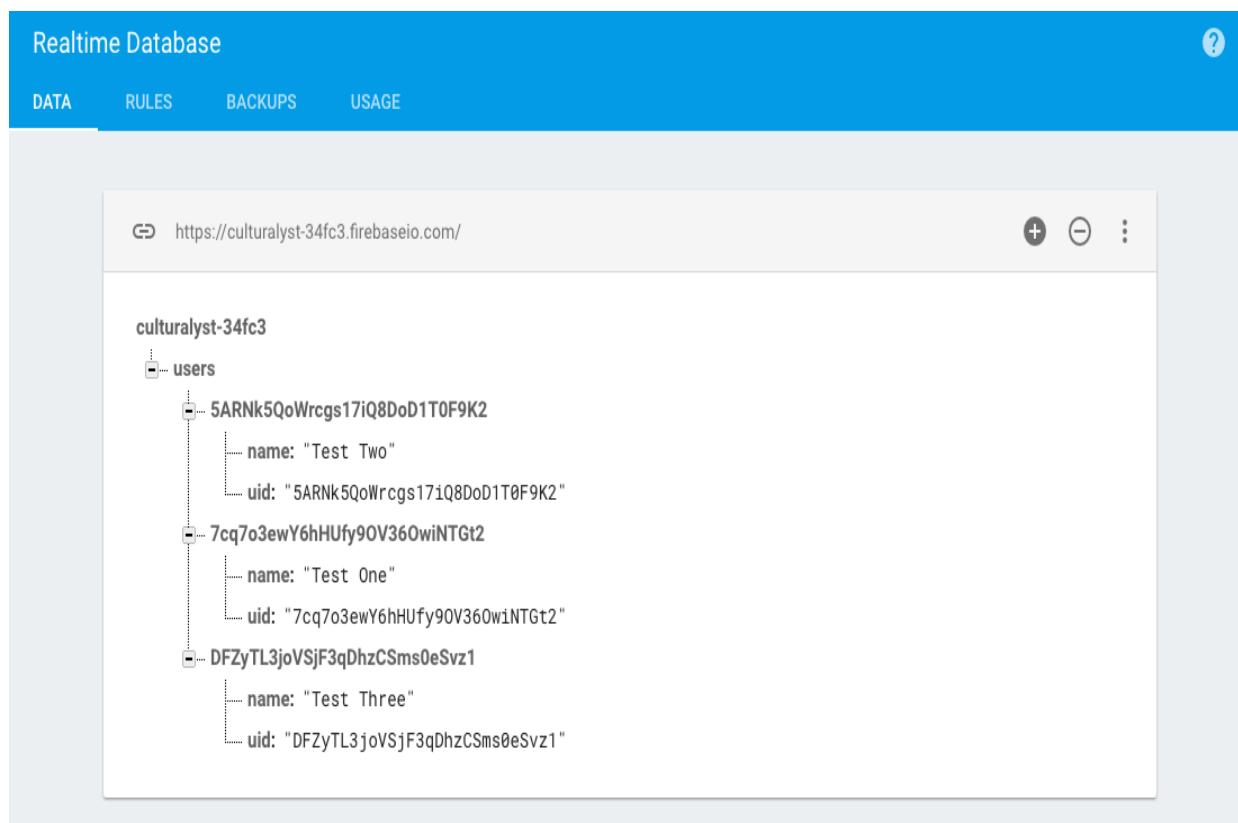
#### 2.4.1 Giới thiệu

Firebase Realtime Database là một cloud hosted database hỗ trợ đa nền tảng: Android, IOS và Web. Tất cả dữ liệu được lưu trữ ở định dạng JSON, và với bất kỳ sự thay đổi dữ liệu nào thì sẽ có sự phản hồi ngay lập tức, hiển thị đồng bộ trên các nền tảng và các thiết bị.

#### 2.4.2 Những đặc điểm nổi bật

##### 2.4.2.1 Cách dữ liệu được lưu trữ

Dữ liệu trong cơ sở dữ liệu Firebase sẽ được lưu trữ theo dạng JSON và đồng bộ realtime đến mọi kết nối client. Về cơ bản thì toàn bộ dữ liệu trong database là một JSON tree lớn cùng với nhiều điểm node. Do đó, khi xây dựng dữ liệu, cần xây dựng một cấu trúc JSON để dễ dàng cho việc truy cập, tránh việc các node con bị trùng nhau .



Hình 2.18 Realtime database của Firebase

### 2.4.2.2 Dữ liệu offline

Firebase tự động lưu trữ offline khi không có kết nối internet. Tuy nhiên, nó sẽ cho phép lưu trữ vào ổ đĩa persistence khi data offline thậm chí khi ứng dụng restart. Khi kết nối được thiết lập lại, thiết bị khách sẽ nhận được bất kỳ thay đổi nào mà nó bị mất, đồng bộ hóa nó với trạng thái máy chủ hiện tại.

### 2.4.2.3 Cập nhật dữ liệu thời gian thực

Cơ sở dữ liệu Firebase Realtime sử dụng đồng bộ hóa dữ liệu, khi dữ liệu có sự thay đổi, tất cả các thiết bị kết nối đến đều nhận được cập nhật đó trong thời gian ngắn.

### 2.4.2.4 Tính bảo mật và quy định

Firebase cung cấp một cách để xác định vai trò của người dùng khi thực hiện đọc và ghi. Những quy định này sẽ đóng vai trò một lớp bảo mật trên máy chủ trước khi thực hiện bất kỳ các hoạt động truy xuất nào. Mặc định các quy định cho phép người dùng cho phép thực hiện các hoạt động đọc ghi chỉ sau khi đã xác thực.

Quy tắc dưới đây cho phép người sử dụng chứng thực chỉ có thẻ ghi hoặc đọc dữ liệu:

```
{  
  "rules": {  
    ".read": "auth != null",  
    ".write": "auth != null"  
  }  
}
```

Hình 2.19 Quy tắc đóng, cần xác thực để đọc ghi dữ liệu

Quy tắc dưới đây cho phép tất cả mọi người có thẻ đọc và ghi dữ liệu mà không cần xác thực:

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

**Hình 2.20** Quy tắc mở, cho phép mọi người đọc ghi dữ liệu

Ngoài ra, cũng có thể thiết lập các quy tắc để phân quyền hoặc giới hạn quyền truy cập đọc/ghi dữ liệu cho từng người dùng hay đối tượng cụ thể.

# Chương 3. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

## 3.1 GIỚI THIỆU

Hệ thống này sẽ thực hiện công việc giám sát thông số điện năng tiêu thụ và lượng nước sinh hoạt hàng ngày, sau đó đều đặn gửi những thông tin này ra màn hình hiển thị trên thiết bị hay cập nhật lên giao diện web để người dùng tiện theo dõi, quan sát.

Giả dụ, đối với những ai muốn sử dụng hệ thống này cho hộ gia đình của họ, điều đầu tiên là họ cần mua bộ thiết bị giám sát điện – nước, sau đó phải cung cấp cho một tài khoản gmail đáng tin cậy (đây xem như là một tài khoản mà họ đăng ký) để có thể đăng nhập vào hệ thống và giám sát các thông số điện - nước của chính họ trực tiếp qua internet.

Admin dựa vào database của Firebase, sẽ hoàn toàn kiểm soát được thông tin của từng người dùng, đồng thời cũng có thể thực hiện khóa hoặc cho phép tài khoản gmail đó có tiếp tục được sử dụng dịch vụ giám sát qua internet nữa hay không.

## 3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

### 3.2.1 Thiết kế sơ đồ khái niệm hệ thống

Với những yêu cầu kể trên, hệ thống sẽ bao gồm các khối sau: khối nguồn, khối xử lý trung tâm, khối hiển thị, khối thiết bị đầu vào và WebServer.

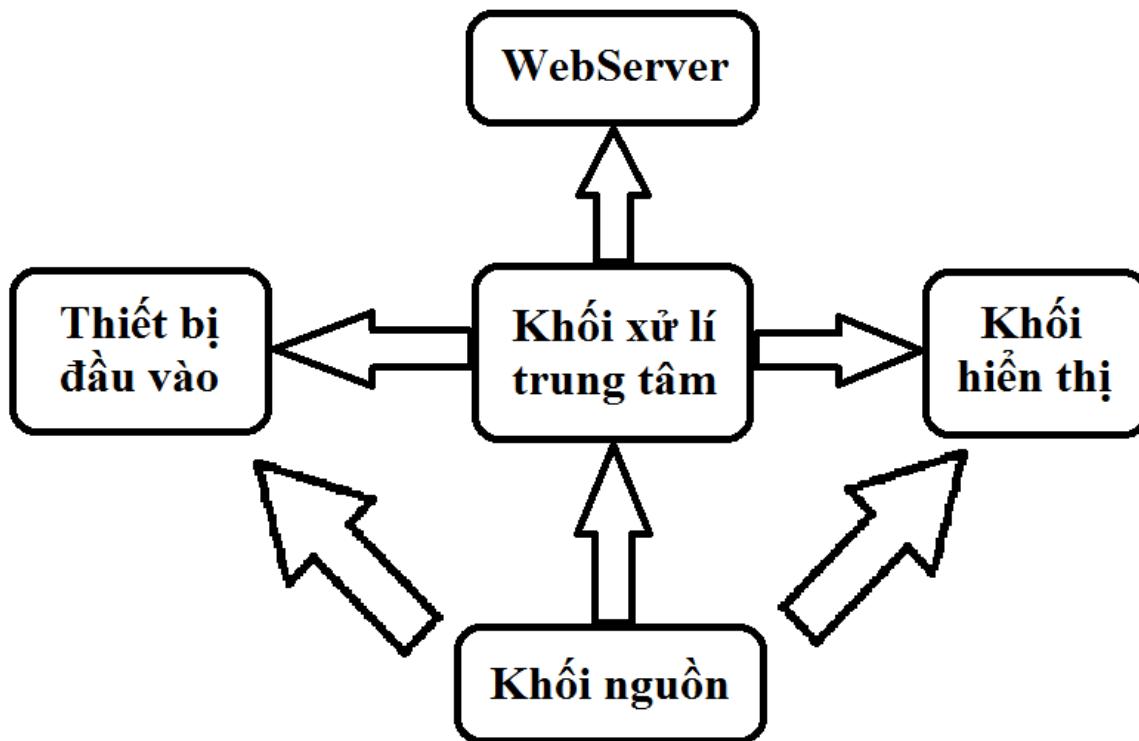
**Khối nguồn:** cấp nguồn cho hệ thống hoạt động.

**Khối xử lý trung tâm:** nhận dữ liệu từ khái thiết bị đầu vào, tính toán các giá trị điện - nước sau đó xuất tín hiệu ra khái hiển thị để hiển thị các giá trị điện - nước đã sử dụng. Kết nối mạng không dây để đọc thời gian thực từ internet về so sánh để đều đặn lấy dữ liệu từ database về xử lý và cập nhật dữ liệu mới lên database.

**Khối hiển thị:** hiển thị thông tin về lượng điện - nước tiêu thụ.

**Khối thiết bị đầu vào:** bao gồm cảm biến dòng, mạch đo áp và cảm biến lưu lượng.

**WebServer:** bao gồm một database để lưu trữ dữ liệu và một web host để xây dựng trang web giám sát điện - nước.



Hình 3.1: Sơ đồ khối hệ thống

#### 3.2.2 Tính toán và thiết kế mạch

##### 3.2.2.1 Thiết kế khói hiển thị

Trên mô hình sẽ có một màn hình LCD 16x2 hiển thị thông tin về mức điện năng đã sử dụng hoặc lượng nước đã sử dụng. Sử dụng một nút nhấn để chuyển đổi hiển thị qua lại giữa hai thông tin điện – nước. LCD 16x2 sẽ sử dụng nguồn 5V từ Arduino để hoạt động và tiêu tốn dòng tối đa là 120mA.

##### 3.2.2.2 Thiết kế khói xử lý

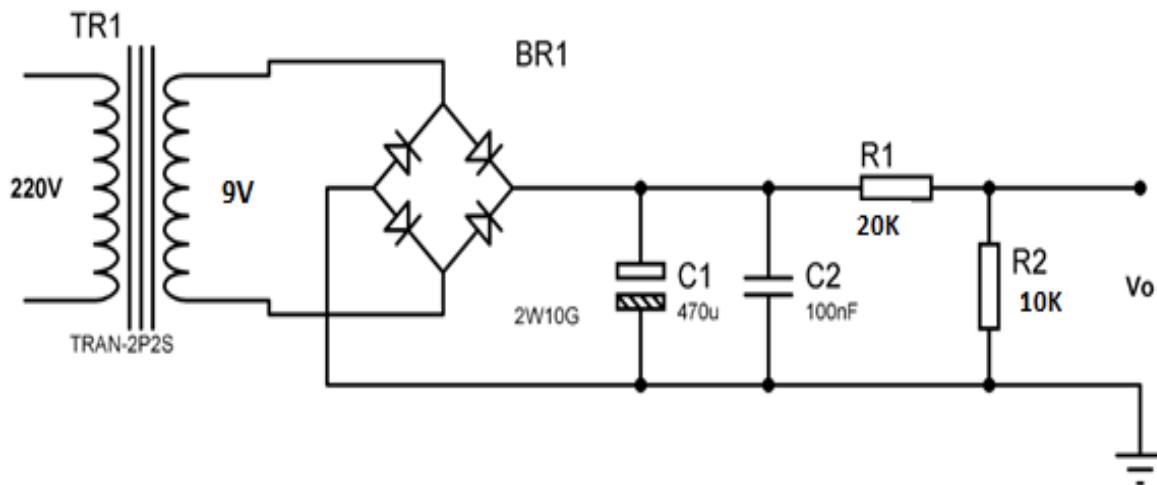
Sử dụng Arduino Mega để tiếp nhận và xử lý dữ liệu từ khói thiết bị đầu vào. Arduino Mega có dòng tiêu thụ là 200mA. Module giao tiếp wifi NodeMcu sẽ tiếp nhận dữ liệu để xử lý, so sánh và cập nhật thông tin lên web server; khi hoạt động ở mức tối đa thì tiêu thụ dòng 170mA.

Về **WebServer** nhóm sử dụng database Firebase để lưu trữ dữ liệu và xây dựng trang web với Firebase hosting.

### 3.2.2.3 Thiết kế khối thiết bị đầu vào

#### Thiết kế mạch đo điện áp:

Có nhiều phương pháp để đo điện áp của nguồn điện. Nếu không xét về kích thước hời công kèm và giá thành của sản phẩm, thì việc sử dụng biến áp để giảm áp là một trong những cách đơn giản và an toàn.



**Hình 3.2:** Mạch đo điện áp

Như sơ đồ nguyên lý bên trên, nguồn 220Vac khi qua biến áp sẽ được giảm xuống còn 9Vac hiệu dụng (điện áp đỉnh là 12.7V). Sử dụng cầu diode để chỉnh lưu thành điện áp một chiều 11.3V đỉnh (sụt áp trên diode là 1.4V). Sau đó sử dụng 2 tụ điện để lọc nhiễu và làm phẳng điện áp.

Nhưng để bộ ADC của Arduino đo được điện áp thì điện áp ngõ ra Vo phải nhỏ hơn 5V, và để tránh trường hợp áp tăng cao đột ngột gây hại cho Arduino thì ta xử lí cho điện áp nhỏ hơn 4V. Ta sử dụng cầu phân áp để xử lý vấn đề này, theo sơ đồ hình 3.2 điện áp ngõ ra Vo được chia xuống còn 1/3 tức sẽ có giá trị khoảng 3.78V.

Độ phân giải của ADC step-size được tính theo công thức sau:

$$SS = \frac{\text{điện áp tham chiếu}}{\text{giá trị max của ADC}}$$

Với giá trị max của ADC là 1023 (ứng với 10bit) và điện áp tham chiếu là 3.88V.

### Cảm biến đầu vào:

Sử dụng một module đo dòng điện ACS712 có dòng tiêu thụ là 13 mA và một cảm biến lưu lượng S201 với mức dòng tiêu tốn cao nhất 15 mA. Cả 2 cảm biến này sử dụng nguồn 5V lấy từ arduino để hoạt động.

#### 3.2.2.4 Thiết kế khối nguồn

Với thông số mức dòng điện tiêu thụ đã kê ở trên, ta tính được dòng điện mà hệ thống cần phải cung cấp:

Dòng tiêu thụ khói hiển thị: 120mA

Dòng tiêu thụ của khói xử lý:

$$200\text{mA} (\text{Arduino Mega}) + 170\text{mA} (\text{NodeMCU}) = 370\text{mA}$$

Dòng tiêu thụ khói thiết bị đầu vào:

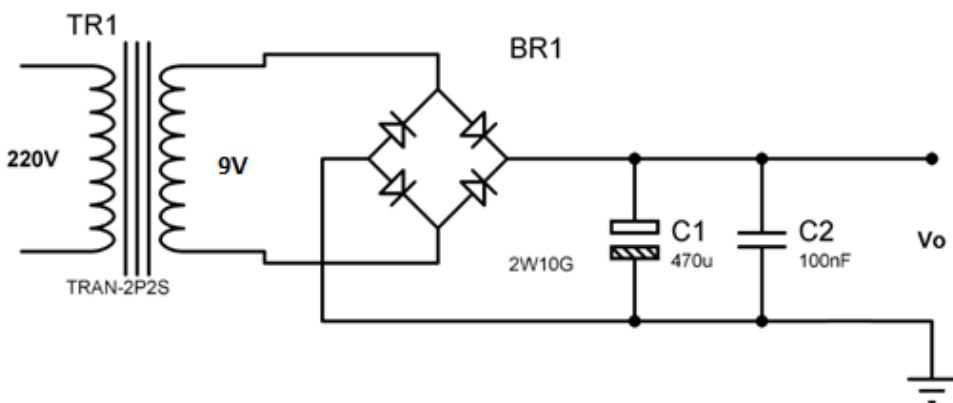
$$13\text{mA} (\text{ACS712}) + 15\text{mA} (\text{S201}) = 28\text{mA}$$

Như vậy dòng tổng cần cung cấp cho cả hệ thống là:

$$I_{tổng} = \mathbf{120mA + 370mA + 28mA = 518mA}$$

Cả arduino và NodeMCU đều sử dụng IC AMS1117 để ổn định điện áp cho mạch hoạt động nên nguồn cấp cho board có thể dao động từ 7V – 12V (giới hạn từ 6V – 15V).

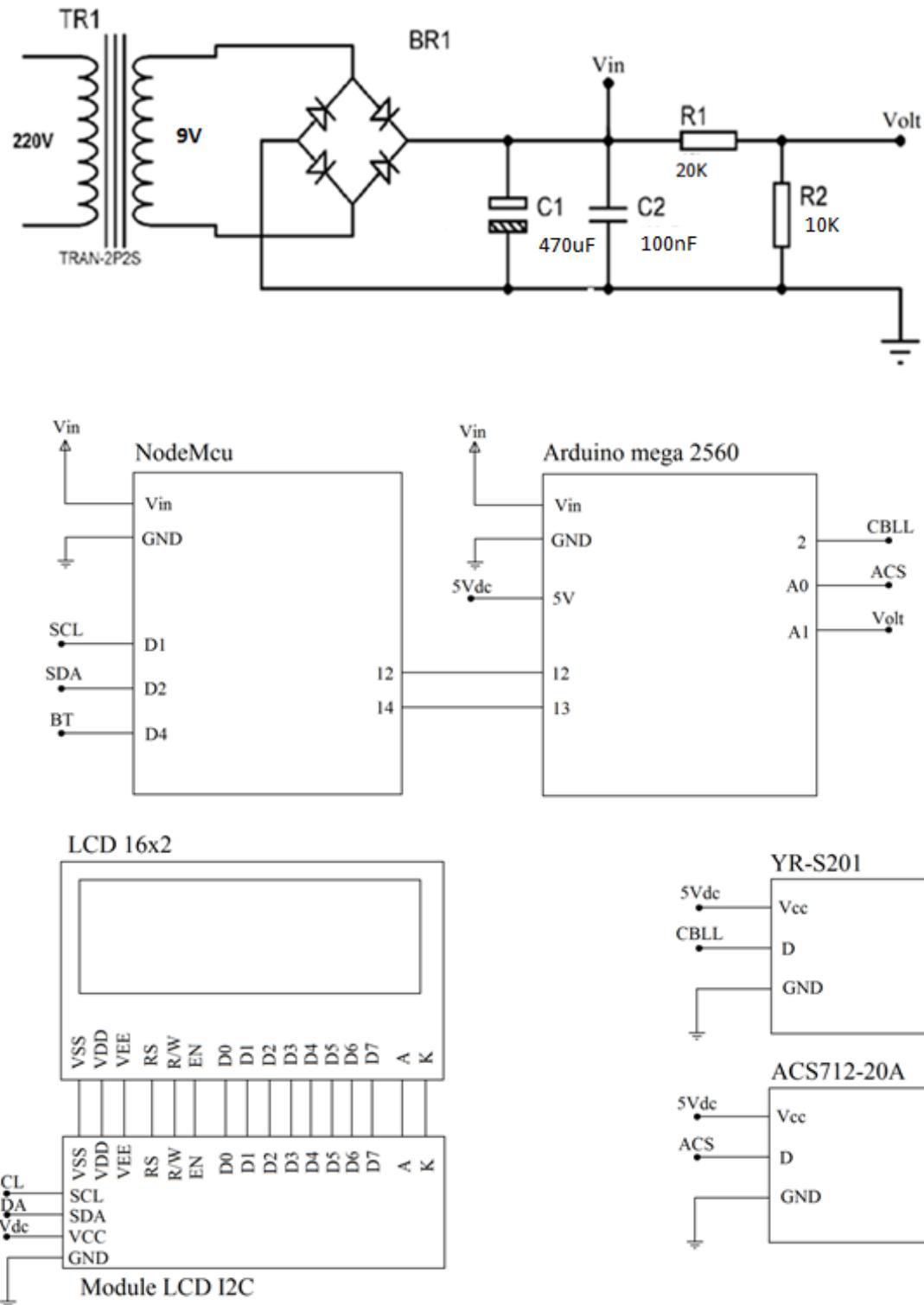
Ta thiết kế mạch nguồn như sau:



Hình 3.3: Mạch nguồn cấp cho Arduino và NodeMCU

Điện áp đầu ra của mạch nguồn được tính toán giống như mạch đo áp với  $Vo$  khoảng 11.3V.

### 3.2.3 Sơ đồ nguyên lý của toàn mạch



**Hình 3.4:** Sơ đồ nguyên lý toàn mạch

# Chương 4. THI CÔNG HỆ THỐNG

## 4.1 GIỚI THIỆU

Sau khi đã tính toán và lựa chọn thiết bị cụ thể, ta sẽ bước sang giai đoạn sau cùng là thi công hệ thống. Về phần cứng, phần lớn các thiết bị được sử dụng trong đề tài là các module và cảm biến đều có sẵn trên thị trường: Arduino Mega, NodeMCU, ACS712, S201... Ngoài ra sẽ thiết kế mạch để đo giá trị điện áp AC 220V.

Đối với phần mềm, chúng ta sẽ tiến hành lập trình điều khiển cho Arduino Mega và NodeMCU; đồng thời sẽ thiết kế giao diện web server và xây dựng cơ sở dữ liệu Realtime Database Firebase.

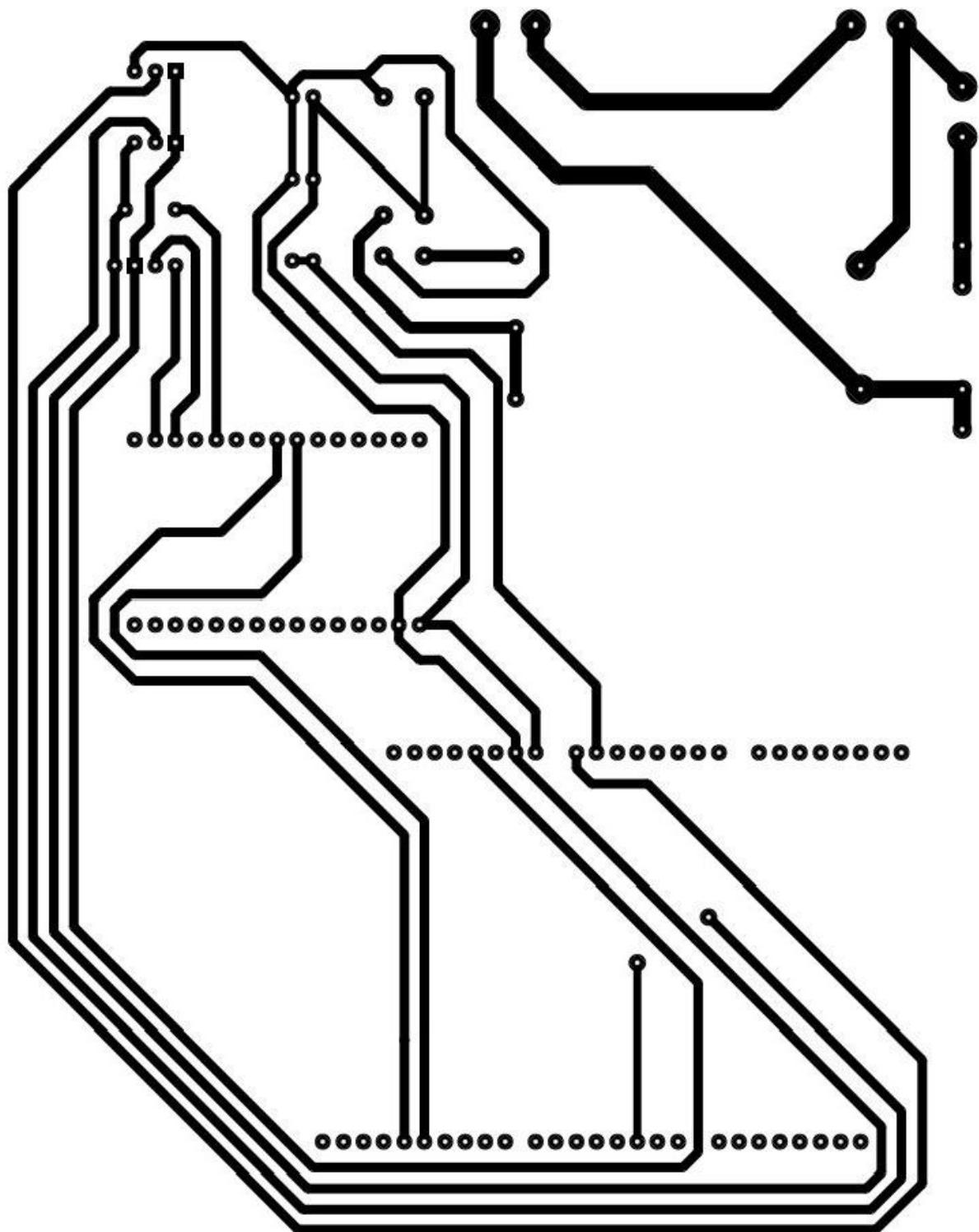
## 4.2 THI CÔNG HỆ THỐNG

Hệ thống này thực tế có thể thiết kế để tối đa 250 tài khoản người dùng đăng ký và sử dụng (số lượng tối đa mà Firebase cho phép trong một project), tuy nhiên, trong đề tài này chỉ sử dụng 4 tài khoản gmail để đăng ký và sử dụng với vai trò là user, và một tài khoản được sử dụng với vai trò là admin. Các user chỉ có thể xem thông tin giám sát của chính họ và nhận những thông báo từ hệ thống, còn admin thì có thể giám sát được tất cả thông số điện – nước của người dùng thông qua giao diện web.

Từ đó, nhóm thi công 2 bộ mô hình giống nhau để giám sát điện – nước sinh hoạt, tượng trưng cho 2 hộ gia đình trên thực tế. Hai mô hình này sẽ thực hiện công việc đo đạc các thông số và tính toán các giá trị về lượng điện - nước đã tiêu thụ rồi gửi dữ liệu lên database Firebase để hiển thị ra trang web. Các tài khoản mail đăng ký còn lại, cũng sẽ đóng vai trò là các hộ gia đình, nhưng chỉ hiển thị các thông số mang tính chất minh họa tượng trưng.

Vì 2 bộ mô hình được thiết kế giống nhau nên nhóm sẽ trình bày quá trình thi công của một bộ mô hình giám sát điện - nước sinh hoạt hằng ngày.

Từ sơ đồ nguyên lý của hệ thống đã được tính toán và thiết kế ở Chương 3 nhóm đã thiết kế được mạch điện của hệ thống theo sơ đồ mạch in cùng với danh sách các linh kiện được sử dụng như sau:



Hình 4.1: Sơ đồ mạch in

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

---

Bảng 4.1. Danh sách các linh kiện điện tử

STT	Tên linh kiện	Số lượng	Giá trị
1	Arduino mega	1	
2	Nodemcu	1	
3	ACS712	1	
4	S201	1	
5	Biến áp	1	
6	Diode cầu	1	
7	Tụ hóa	1	470uF
8	Tụ gốm	1	100nF
9	Điện trở	1 1	10K 20K
10	Cầu chì	1	2A
11	Nút nhấn	1	
12	LCD 16x2	1	
13	Module LCD I2C	1	

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

---

### 4.3 THI CÔNG VÀ ĐÓNG GÓI MÔ HÌNH

#### 4.3.1 Đóng gói bộ điều khiển

Dùng mica trong có kích thước 20cm x 17cm x 9cm để thiết kế vỏ hộp.



**Hình 4.2:** Mặt ngoài của mô hình

### 4.3.2 Thi công mô hình

Sử dụng một bảng mạch để gắn các thiết bị và linh kiện và hàng rào kết nối lên trên. Dùng ốc vít để gắn chặt bảng mạch vào mặt dưới của hộp. Lớp vỏ hộp bên ngoài dùng mica trong, có thể tìm thấy ở các tiệm điện – nước.

Để cắt mica, ta phải dùng dao chuyên dụng cắt mica. Đây là công việc đòi hỏi sự chính xác và tính thẩm mỹ bởi mica rất khó cắt. Dung keo silicon để dán các miếng mica lại thành hình hộp. Mặt nắp trên sử dụng bản lề nhằm mục đích đóng mở, có khoét lỗ để gắn cố định màn hình hiển thị LCD và nút bấm trên bề mặt. Đi dây kết nối các bộ điều khiển, cảm biến, LCD, biến áp và ống cắm.



**Hình 4.3:** Mặt trong của mô hình và kết nối

### 4.4 LẬP TRÌNH HỆ THỐNG

#### 4.4.1 Lưu đồ giải thuật

Khi cấp nguồn, màn hình LCD sẽ hiển thị 3 thông số là điện năng tiêu thụ, dòng điện và điện áp hiện tại lấy từ mạch đo áp và cảm biến dòng ACS712. Khi nhấn nút chuyển chế độ, LCD sẽ hiển thị mức nước tiêu thụ lấy từ cảm biến lưu lượng S201, nhấn lần nữa để lại hiển thị điện năng.

Arduino sẽ đều đặn gửi dữ liệu qua module wifi Nodemcu. Nodemcu tiếp nhận dữ liệu và cập nhật hiển thị LCD và database của Firebase. Khi xảy ra trường hợp mất điện, thì Nodemcu chỉ việc lấy dữ liệu từ database Firebase và cộng với mức tiêu thụ mà arduino gửi qua để tiếp tục hiển thị khi có điện lại.

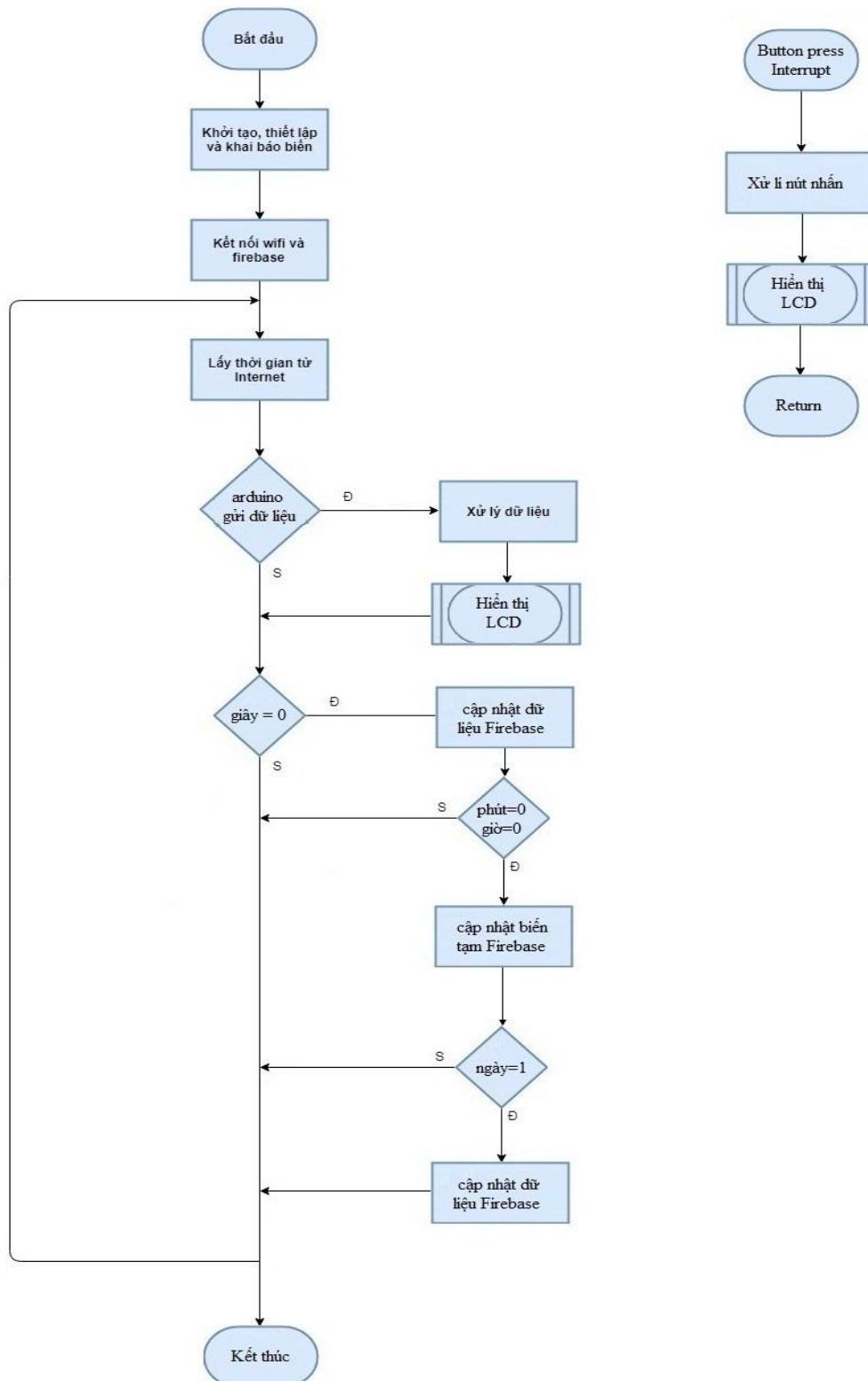
Đối với Arduino, ta sẽ tiến hành khởi tạo thiết lập kết nối, khai báo biến và bật các chức năng cần thiết. Cả hai chương trình đo điện và nước tiêu thụ được thực hiện song song. Sử dụng hàm millis() được hỗ trợ sẵn từ Arduino IDE để canh thời gian, tiến hành lấy mẫu dòng điện, điện áp trong 1 giây; sau đó sẽ thực hiện công việc tính toán. Bên cạnh đó, Arduino sẽ thực hiện ngắn để chạy chương trình đếm số lượng xung và dừng ngắn để tính toán lượng nước tiêu thụ trong 1 giây (cũng sử dụng hàm millis()).

Sau khi đã tính toán xong thông số điện – nước, biến đếm sẽ tăng lên 1; khi biến đếm bằng 10 thì sẽ gửi dữ liệu sang Nodemcu (tức mỗi 10 giây sẽ gửi dữ liệu sang Nodemcu một lần).

Đối với Nodemcu, khi khởi động sẽ thực hiện khởi tạo kết nối và bật các chức năng cần thiết. Sau đó sẽ thực hiện kết nối wifi và Firebase hosting dựa trên các khai báo. Nodemcu sẽ kiểm tra và nhận dữ liệu từ arduino, khi arduino gửi dữ liệu sang thì Nodemcu sẽ nhận, xử lý để hiển thị lên lcd và gửi lên Firebase. NodeMcu lấy thời gian thực trực tiếp từ Internet, thực hiện so sánh và cập nhật dữ liệu lên database Firebase sau mỗi 120s. Đầu tháng sẽ reset các dữ liệu tổng về 0, bên cạnh đó sẽ lưu giá trị tổng điện – nước của tháng này để thống kê mức độ sử dụng điện - nước trong 3 tháng gần nhất. Nếu nhấn nút hiển thị LCD, chương trình sẽ thực hiện ngắn để xử lý trong thời gian rất ngắn, sau đó tiếp tục hoạt động.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

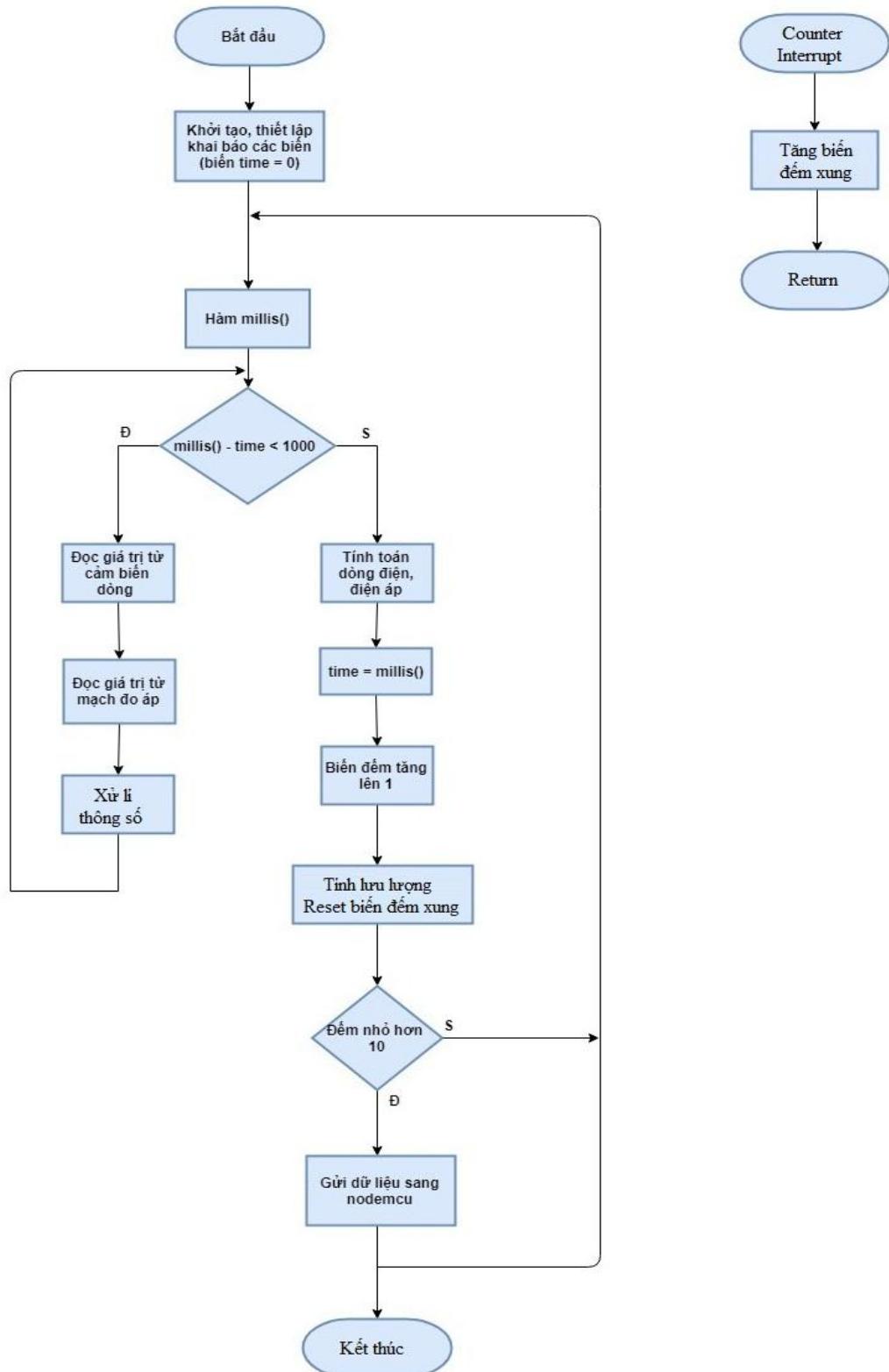
### Lưu đồ NodeMCU



Hình 4.4: Lưu đồ giải thuật của NodeMcu

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### Lưu đồ Arduino Mega 2560



Hình 4.5: Lưu đồ giải thuật của Arduino

### 4.4.2 Phần mềm lập trình

Để tài sử dụng phần mềm ArduinoIDE (Arduino Intergrated Development Environment) để lập trình cho Arduino Mega 2560 và cả NodeMCU; Ngôn ngữ được sử dụng ở ArduinoIDE là C/C++. Tất cả đều là mã nguồn mở, được đóng góp và hỗ trợ rất nhiều từ cộng đồng, rất thích hợp cho những ai mới bắt đầu tìm hiểu hoặc không chuyên để dễ dàng tiếp cận, nắm bắt và triển khai nhanh chóng. ArduinoIDE hoạt động được trên cả 3 nền tảng: Windows, MAC OS và Linux.

Bên dưới là hướng dẫn về cách cài đặt, cách tạo project, viết code và biên dịch chương trình trên hệ điều hành Windows. **Cài đặt:**

#### Cài đặt Java Runtime Environment (JRE)

Vì Arduino IDE được viết trên Java, nên ta cần phải cài đặt JRE trước; nếu không, Arduino IDE sẽ không hoạt động được. JRE có 2 bản phổ biến nhất hiện nay dành cho Windows 32 bit (x86) và Windows 64 bit (x64).

Link download: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

#### Cài đặt Arduino IDE

Link download: <https://www.arduino.cc/en/Main/Software/>

**Bước 1:** Ta click vào đường dẫn ở trên, chọn “Windows ZIP file for non admin install” như hình sau:



**Hình 4.6:** Giao diện tải Arduino IDE

Tiếp tục bấm vào “**JUST DOWNLOAD**” để tải phần mềm Arduino IDE, bạn cũng có thể đóng góp ở ngay phím bên cạnh.

### Contribute to the Arduino Software

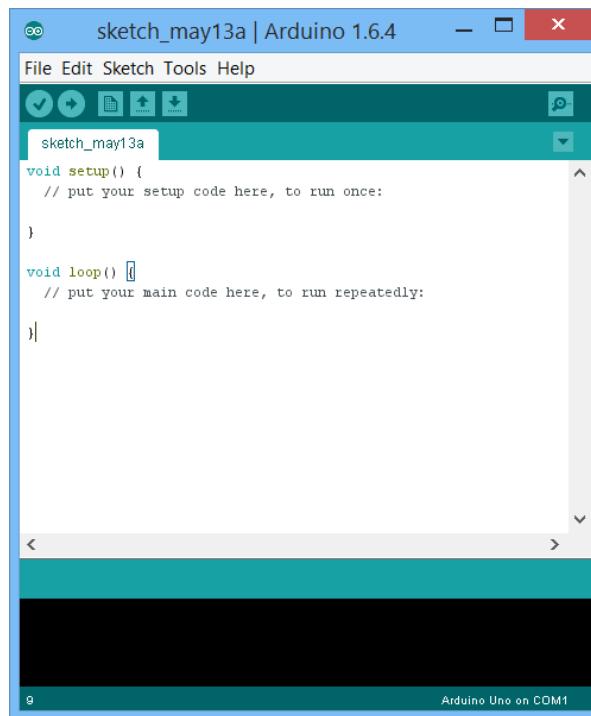
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



Hình 4.7: Ủng hộ nhà phát triển Arduino IDE

**Bước 2:**Khi đã tải xong, giải nén file vừa tải. Sau đó copy thư mục đó đến nơi lưu trữ mong muốn.

**Bước 3:** Chạy file **arduino.exe** để khởi động Arduino IDE



Hình 4.8: Giao diện chính của Arduino IDE

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Sau khi đã cài đặt xong, mở phần mềm lên ta sẽ thấy một giao diện rất dễ nhìn và thân thiện hơn so với đa phần các phần mềm lập trình khác. Arduino IDE làm việc với dưới dạng bản soạn thảo gọi là Sketch, ta sẽ soạn các lệnh lập trình (code) vào Sketch rồi sử dụng thao tác biên tập và upload chương trình đó xuống board Arduino đã cắm vào máy. Khi tiến hành nạp code thì ta cần phải chắc chắn rằng phần mềm đã nhận được tín hiệu của board Arduino (Arduino COM port detect); bản Sketch đang soạn nạp đúng với board Arduino tương ứng (khi cần soạn hai Sketch giao tiếp giữa hai board Arduino và cắm vào cùng máy tính thì vấn đề như vậy sẽ bắt đầu phát sinh). Khi cắm board Arduino vào máy tính cổng COM sẽ được nhận và ta vào phần Tools -> Port để chọn cổng COM kết nối Arduino IDE với board. Sau khi máy đã nhận cổng COM thì ta cần điều chỉnh phần mềm lập trình Arduino xác nhận đúng loại board đang muốn nạp.

### Cài đặt driver cho Arduino Mega

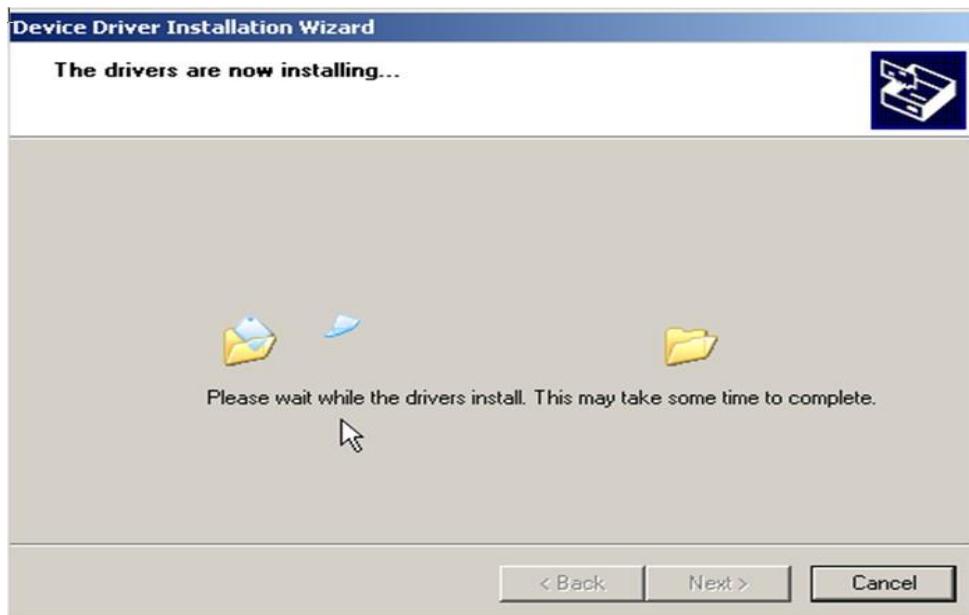
Để arduino giao tiếp được với máy tính, ta cần phải cài đặt driver. Trong thư mục đã giải nén, tìm thư mục drivers. Ta chọn cài đặt **dpinst-amd64.exe** đối với Windows 64bit và **dpinst-x86.exe** ứng với Windows 32bit. Một cửa sổ hiện ra như hình bên dưới, ta chọn “**Next**”:



Hình 4.9: Cài đặt Driver cho Arduino 1

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Tiếp tục chọn “Install”, đợi một khoảng thời gian ngắn để tiến hành cài đặt driver:



**Hình 4.10:** Cài đặt Driver cho Arduino 2

Nhấn “Finish” để kết thúc cài đặt:

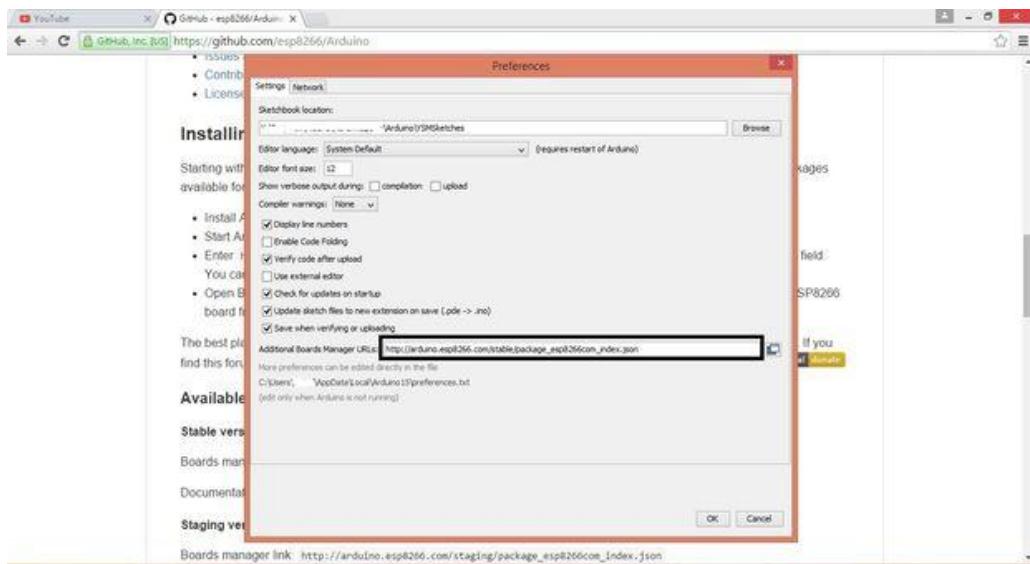


**Hình 4.11:** Cài đặt Driver cho Arduino 3

### Cài đặt Driver cho NodeMCU

Link-download: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

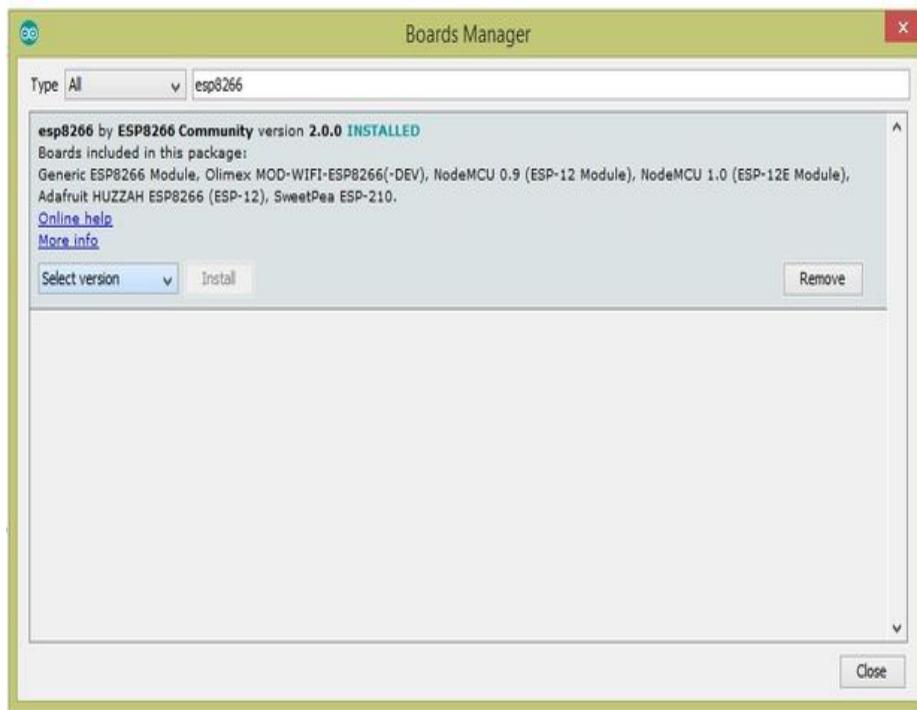


Hình 4.12: Cài đặt Driver cho NodeMCU 1

**Bước 1:** Trong màn hình chính của Arduino, ta chọn File -> Preference, copy đường link trên vào mục Additional Boards Manager URLs; click chọn OK và khởi động lại Arduino IDE.

**Bước 2:** Cài đặt Firmware ESP8266 cho Arduino IDE.

Vào Tools -> Boards Manager -> tìm “esp8266” -> Install -> Khởi động lại IDE.



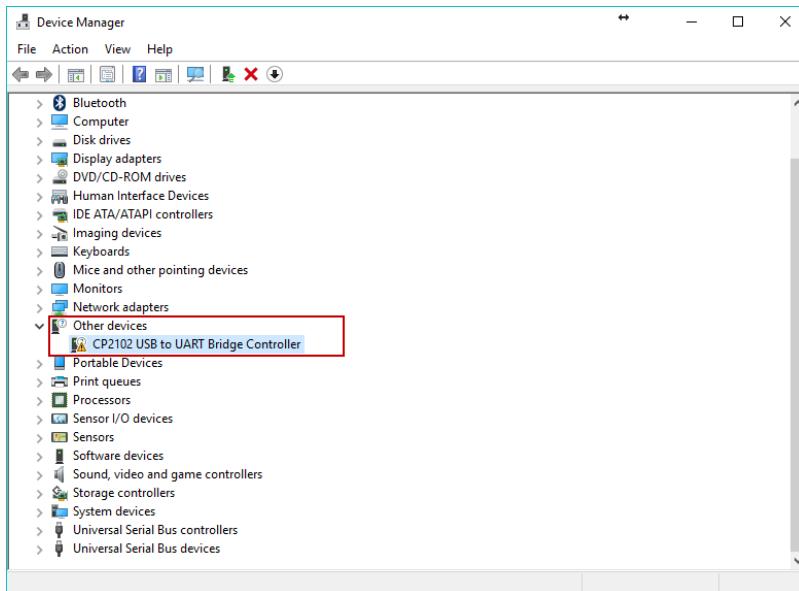
Hình 4.13: Cài đặt Driver cho NodeMCU 2

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

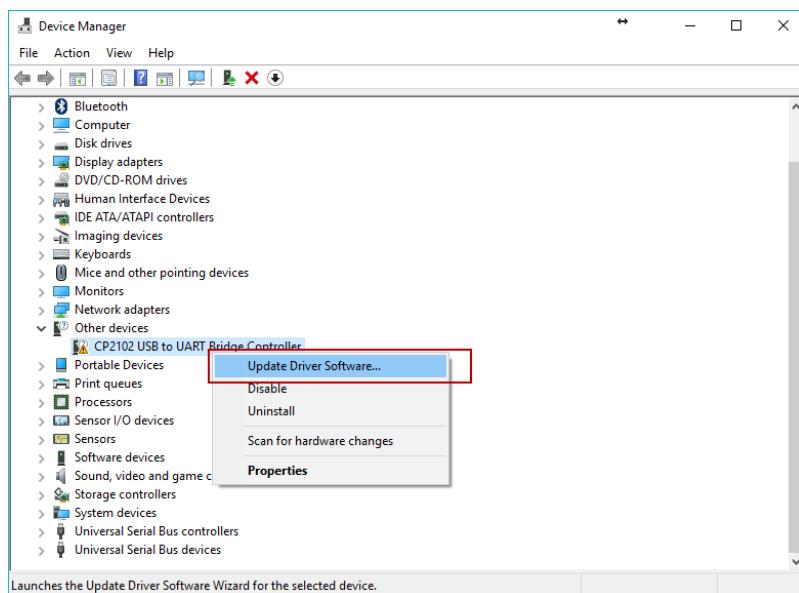
Vào Device Manager để kiểm tra xem driver đã nhận được hay chưa. Nếu trong trường hợp xuất hiện dấu chấm than như hình 4.12 bên dưới thì tức là máy tính đang sử dụng chưa có driver (đây là một trong những lỗi rất thường gặp khi máy tính không giao tiếp được với module Wifi). Ta cần tải driver bằng link bên dưới rồi giải nén. Tiếp theo click phải chuột để chọn Update Driver Software

Link Driver:

[https://www.pololu.com/file/download/pololu-cp2102-windows-121204.zip?file\\_id=0J14](https://www.pololu.com/file/download/pololu-cp2102-windows-121204.zip?file_id=0J14)



Hình 4.14: Cài đặt Driver cho NodeMCU 3



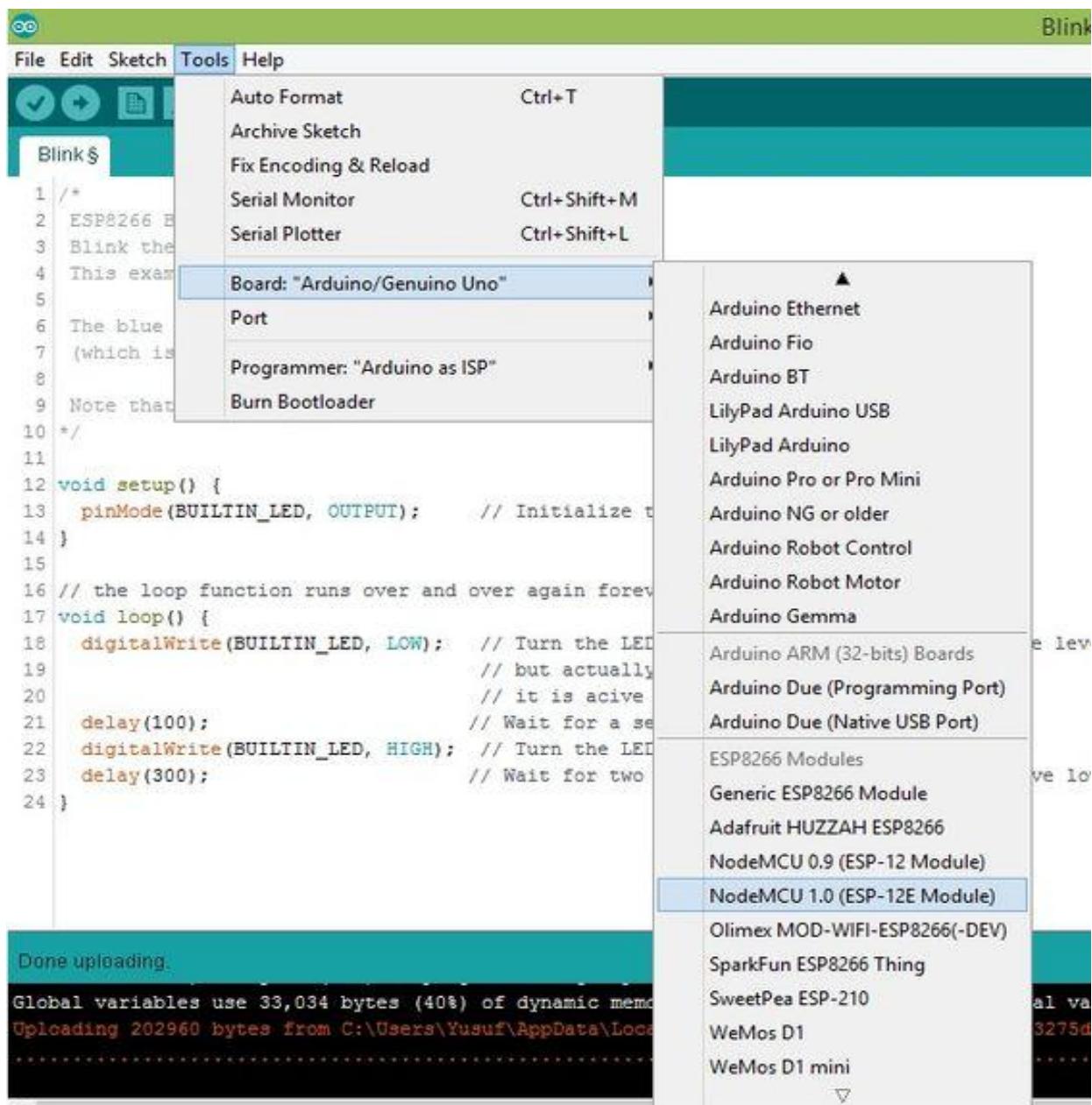
Hình 4.15: Cài đặt Driver cho NodeMCU 4

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Tiếp tục chọn “Browse my computer for driver software” và trả về thư mục có chứa driver để cài đặt ở trên. Vào Device Manager một lần nữa để kiểm tra xem driver đã cài đặt được chưa.

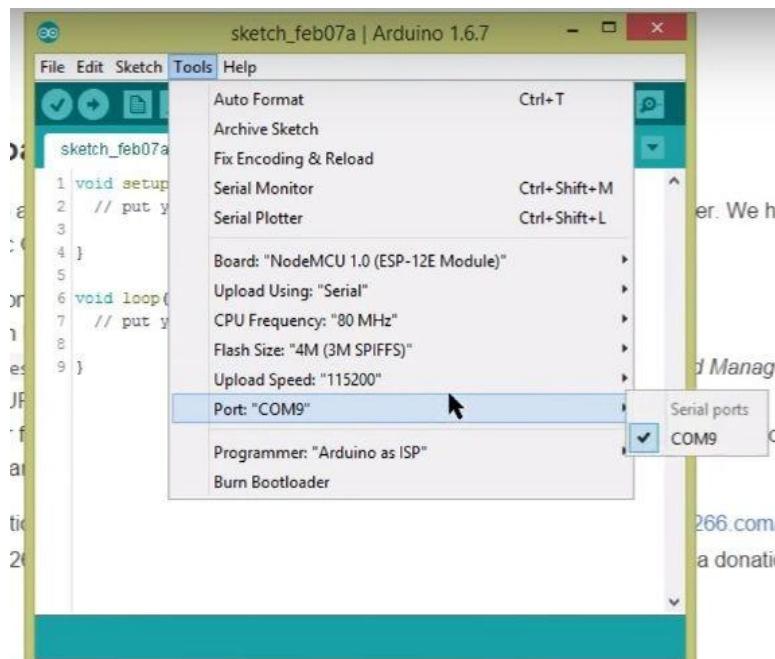
### Chọn phần cứng để lập trình

Vào Tools -> Board -> chọn loại board cần lập trình và chọn Port mà board đang kết nối vào máy tính.



Hình 4.16: Chọn phần cứng để lập trình

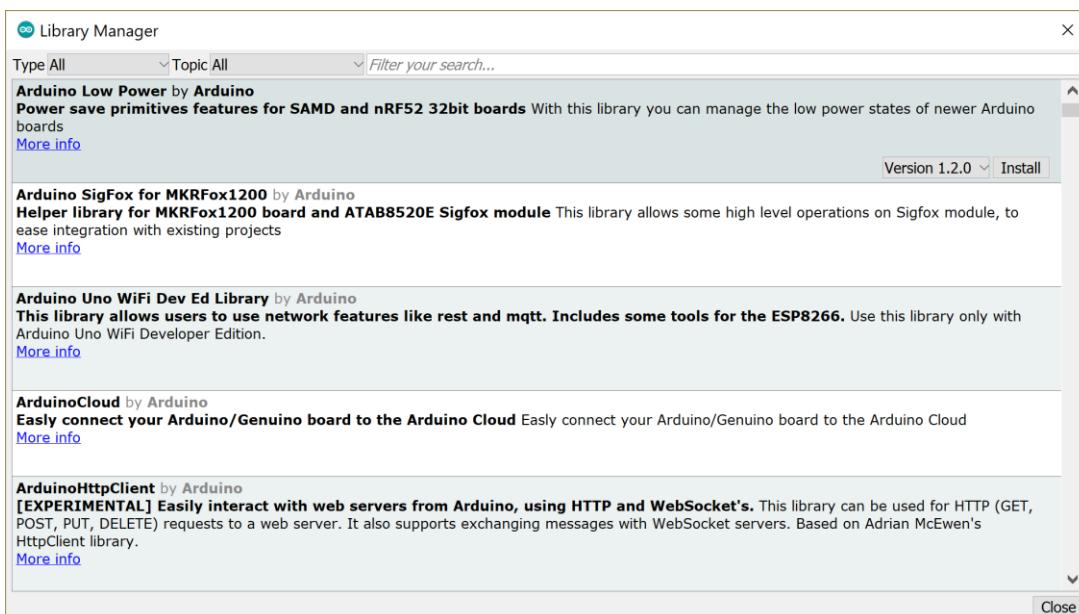
## CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.17: Chọn Port kết nối

### Cài đặt thư viện bổ sung cho Arduino IDE

Để giúp cho việc lập trình dễ dàng hơn, ta cài đặt bộ thư viện bổ sung bằng việc chọn Sketch -> Include Library -> Add .ZIP Library -> trỏ đến thư mục bổ sung định dạng .zip hoặc ta có thể cài đặt gián tiếp bằng cách vào Manage Libraries, tìm tên thư viện cần tải, bấm Install để cài đặt.



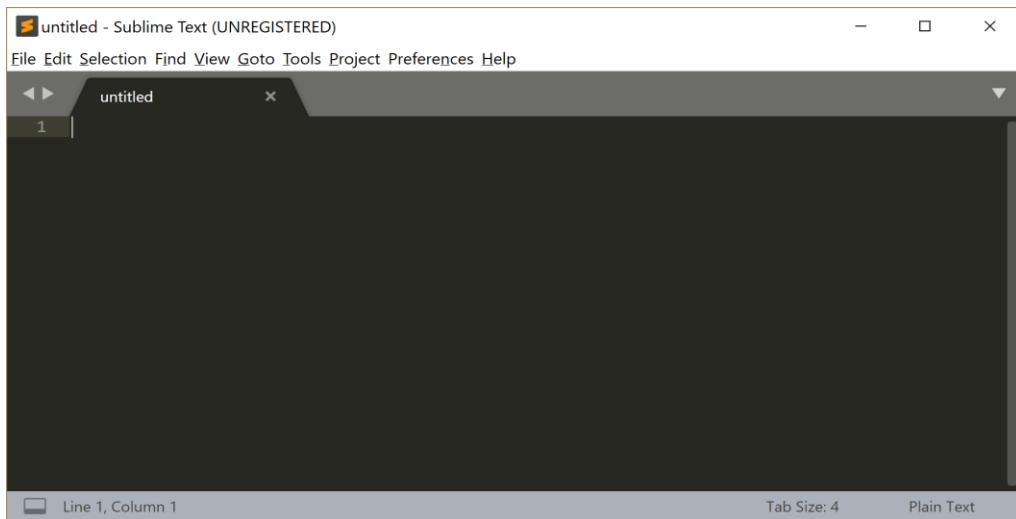
Hình 4.18: Cài đặt thư viện cho Arduino IDE

### Phần mềm lập trình web

Có rất nhiều phần mềm soạn thảo phổ biến để lập trình HTML, CSS như Sublime Text, Notepad++, TextMate Eclipse...Ở đây, tôi sử dụng phần mềm Sublime Text để lập trình web bởi tính giao diện trực quan và dễ sử dụng. Bạn có thể cài đặt Sublime Text theo đường link <http://www.sublimetext.com/>

#### Sử dụng Sublime Text

Trong phần mềm này, ta chỉ cần quan tâm đến ba lệnh cơ bản đó là File -> New để tạo trang mới, File -> Save để lưu lại file đang soạn thảo và File -> Open để mở file sẵn có. Ta cần phải đặt tên theo định dạng cho file HTML là <file\_name>.html, tên cho file JavaScript là <file\_name>.js và <file\_name>.css cho file CSS. Một lưu ý là sau khi đã soạn thảo và muốn upload các file lên một web host, cần phải gom chung các file kể trên vào cùng một thư mục. Đôi khi, file HTML cần phải đặt tên mặc định là index.html.



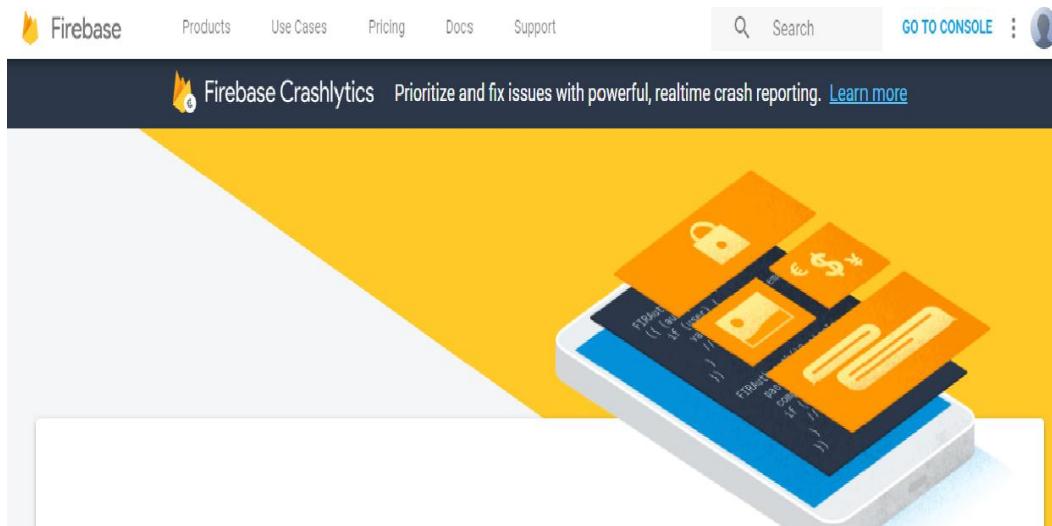
Hình 4.19: Giao diện Sublime Text

#### 4.4.3 Hướng dẫn xây dựng Firebase project

Để bắt tay vào việc xây dựng một project, trái với những hosting khác có giới hạn về mặt thời gian(khoảng 30 ngày sử dụng miễn phí), bạn chỉ cần duy nhất một tài khoản gmail để sở hữu một firebase hosting với nhiều tính năng được hỗ trợ mạnh mẽ của google. Bạn có thể xây dựng tối đa 10 project miễn phí và điều đặc biệt hơn là hoàn toàn không giới hạn về mặt thời gian.

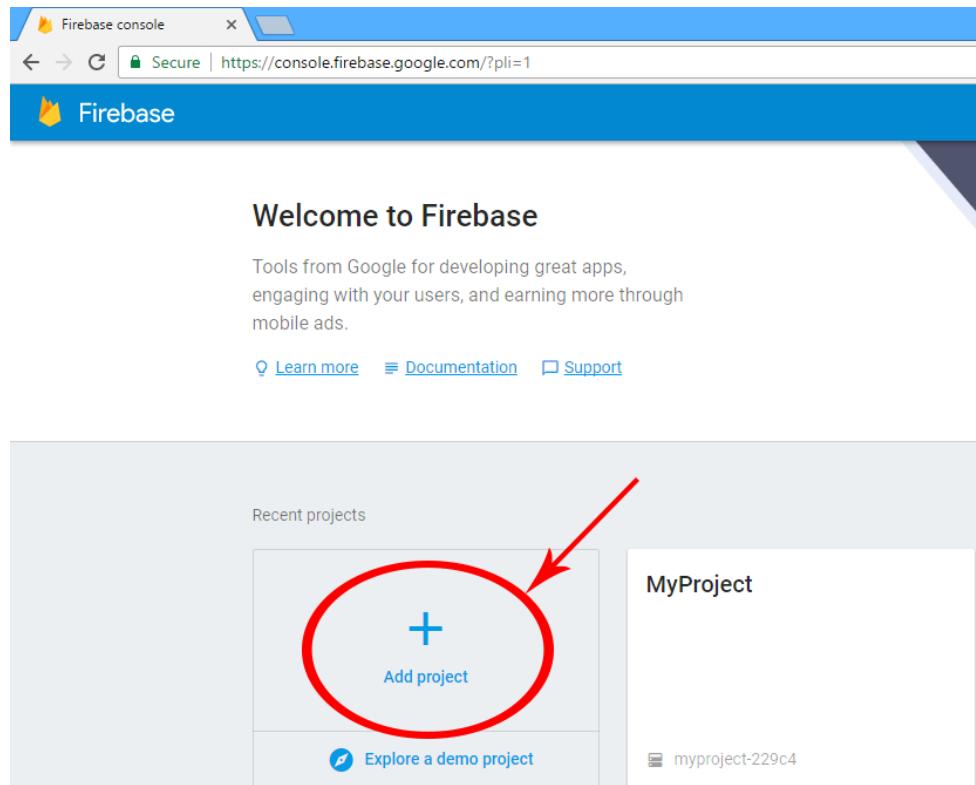
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Ta cần phải tạo sẵn một project trên firebase để có thể upload project từ máy tính lên. Truy cập: <https://firebase.google.com/> để đi tới website sau:



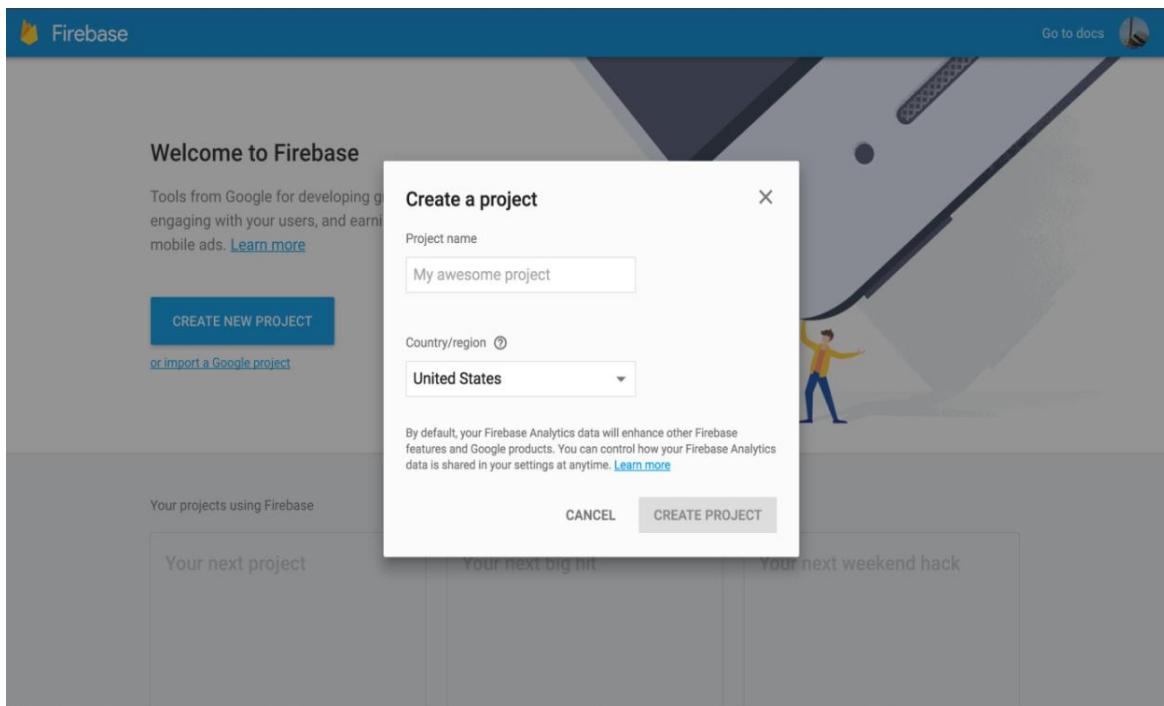
Hình 4.20: Giao diện Firebase console

Click chọn “GO TO CONSOLE” để đi đến bước tạo project. Chọn “Add project” và đặt tên cho project và chọn quốc gia:



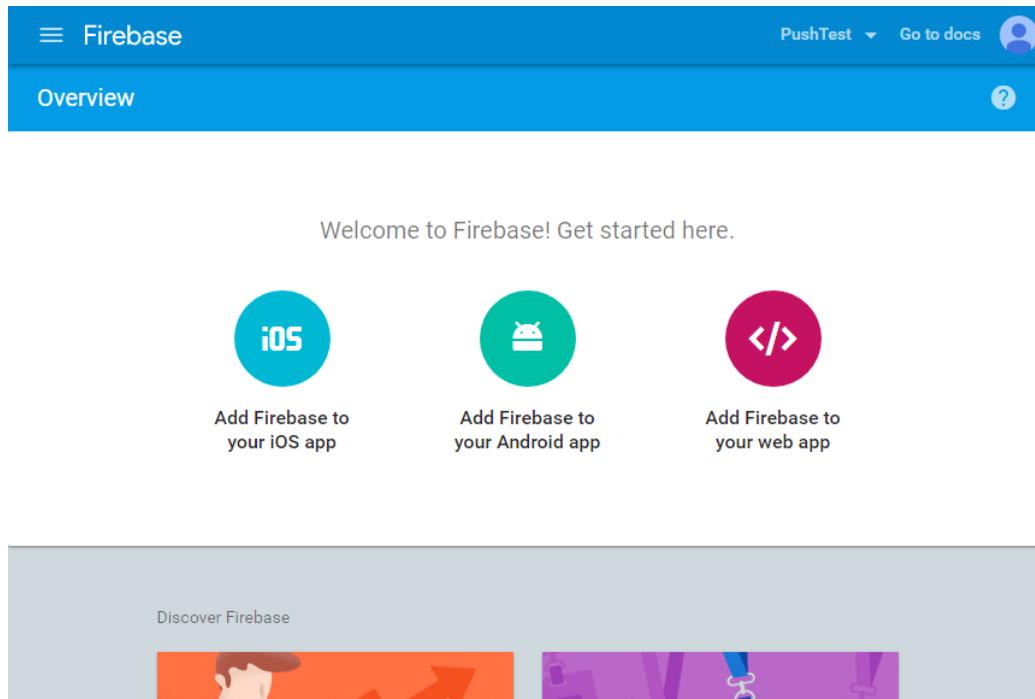
Hình 4.21: Tạo một project mới

## CHƯƠNG 4. THI CÔNG HỆ THỐNG



**Hình 4.22:** Các bước tạo một project

Cuối cùng, chọn 1 trong 3 nền tảng mà project của bạn muốn hướng tới:



**Hình 4.23:** Chọn nền tảng xây dựng project

Như vậy là kết thúc bước tạo một project. Tiếp theo đó, bạn cần phải tải và cài đặt phần mềm tên Node.js về máy tính, theo đường link sau <https://nodejs.org/en/download>

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Một cửa sổ web hiện ra với những phiên bản khác nhau, ta đơn giản click chọn vào tab “**Windows Installer**” ứng với hệ điều hành window để tải tự động phiên bản phù hợp nhất cho máy tính của bạn.

### Downloads

Latest LTS Version: 8.11.2 (includes npm 5.6.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.



Hình 4.24: Giao diện download node.js

Sau khi cài đặt Node.js, mỗi khi khởi động phần mềm này, một cửa sổ màu đen tương tự “**command prompt**” sẽ hiện ra, đây sẽ là nơi làm việc – tương tác với firebase hosting.



Hình 4.25: Giao diện làm việc của node.js

Tiếp theo, tiến hành cài đặt Firebase CLI, giúp cung cấp đa dạng các công cụ cho việc quản lý, giám sát cũng như triển khai một project lên Firebase hosting.

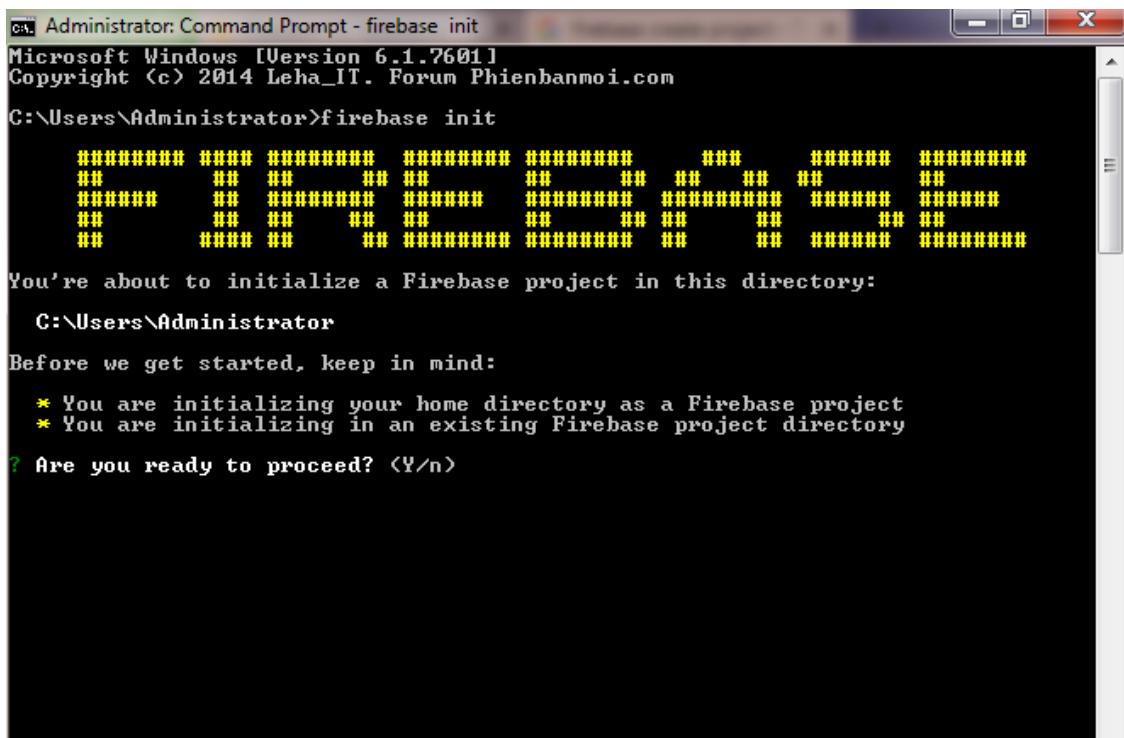
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Để cài đặt Firebase CLI, ngay tại cửa sổ bên trên, ta nhập lệnh command sau:

```
npm install -g firebase-tools
```

và đợi một khoảng thời gian để cài đặt xong.

Dùng lệnh command: **firebase login** để đăng nhập vào gmail mà bạn muốn sử dụng, lưu ý là máy tính của bạn phải đang đăng nhập vào tài khoản gmail đó. Tương tự dùng lệnh **firebase logout** để đăng xuất. Có một lưu ý là khi đăng xuất hoặc đăng nhập vào một tài khoản gmail khác, thì ta cần gõ lệnh **firebase logout** để đăng xuất khỏi tài khoản gmail đã soạn thảo trước đó, và gõ lệnh **firebase login** để truy cập đến tài khoản gmail muốn sử dụng. Dùng lệnh command **cd** để di chuyển đến thư mục chứa project của bạn, chẳng hạn tên thư mục của bạn là **firebase\_pro** trong ô đĩa d. Sau đó nhập tiếp lệnh **firebase init** trong cửa sổ để tiến hành khởi tạo.



**Hình 4.26:** Khởi tạo firebase từ cửa sổ cmd

Click chọn: Y -> Hosting -> chọn một project ở trên firebase -> đặt tên thư mục để chứa các file của project (chẳng hạn tên thư mục là public) -> Y.

Như vậy là xem như hoàn thành bước khởi tạo Firebase.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

---

Cuối cùng, di chuyển các file project của bạn vào thư mục d:/firebase\_pro/public và gõ lệnh command: **firebase deploy** để triển khai project lên firebase.

### 4.5 HƯỚNG DẪN SỬ DỤNG

**Bước 1:** Thiết bị sử dụng nguồn được cung cấp lấy trực tiếp từ nguồn điện nhà 220Vac. Khi cấp điện, màn hình LCD sẽ hiển thị thông số điện năng tiêu thụ (Wh). Dùng nút nhấn để chuyển sang hiển thị lượng nước tiêu thụ (Lít). Do đây chỉ là mô hình thực hiện đo lường các đối tượng tiêu thụ điện năng có công suất nhỏ - lượng nước dùng để đo có dung tích nhỏ, nên đơn vị đo chỉ là Wh và Lít.

**Bước 2:** Sử dụng một tài khoản gmail đang sử dụng đã được admin duyệt, và cho phép để truy cập vào web server của hệ thống mỗi khi muốn giám sát qua internet. Tại giao diện web, người dùng có thể quan sát được mức tiêu thụ điện – nước của họ theo từng ngày, và tiếp nhận những thông báo từ hệ thống gửi đến.

# **Chương 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

## **5.1 KẾT QUẢ ĐẠT ĐƯỢC**

### **5.1.1 Kết quả lý thuyết**

Đồ án này được thực hiện khoảng 3 tháng rưỡi, bắt đầu từ cuối tháng 3 đến ngày 16/7 (kết thúc). Trong khoảng thời gian này, nhóm đã tìm hiểu để thiết kế một hệ thống cả phần cứng lẫn phần mềm.

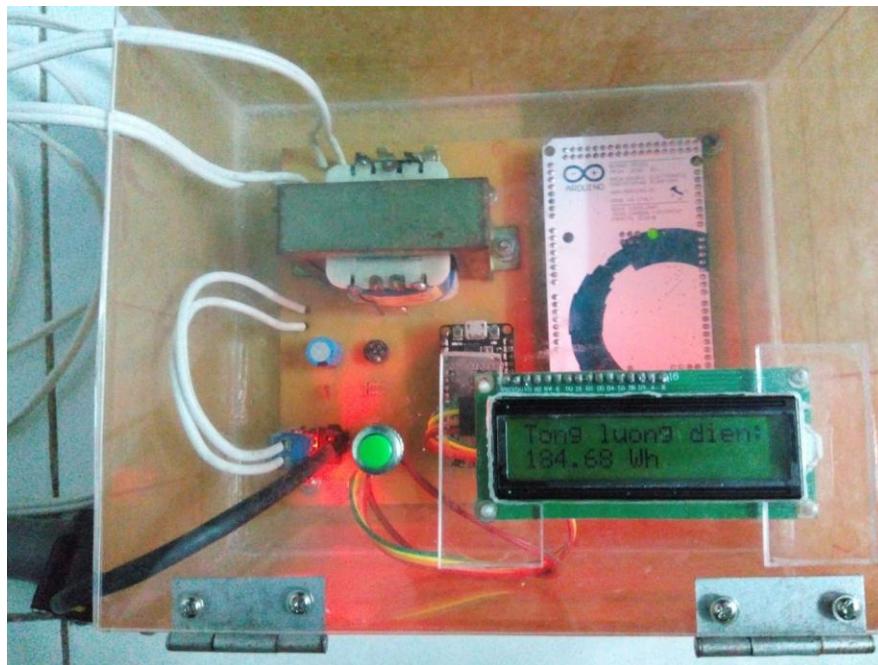
Ở phần cứng, nhóm tìm hiểu nguyên lý và cách thức hoạt động của các module cảm biến, giao tiếp với Module LCD I2C; biết cách thiết kế mạch đo điện áp thông dụng. Khảo sát các khối ngoại vi, thông số kỹ thuật; biết cách giao tiếp, sử dụng Arduino Mega và module giao tiếp wifi NodeMCU.

Về phần mềm, nhóm biết cách sử dụng phần mềm Arduino IDE để lập trình điều khiển trên kit Arduino và NodeMCU. Tìm hiểu về các bộ thư viện có sẵn, để từ đó lập trình điều khiển cho hệ thống. Hiểu được cách trao đổi truyền nhận dữ liệu giữa 2 vi điều khiển theo chuẩn UART, I2C. Biết cách lập trình một web server cơ bản sử dụng các ngôn ngữ JavaScript, HTML và CSS, đồng thời upload trang web đó lên một hosting cụ thể đó là Firebase. Hiểu được nguyên lý và cách sử dụng cơ sở dữ liệu của Realtime Database Firebase.

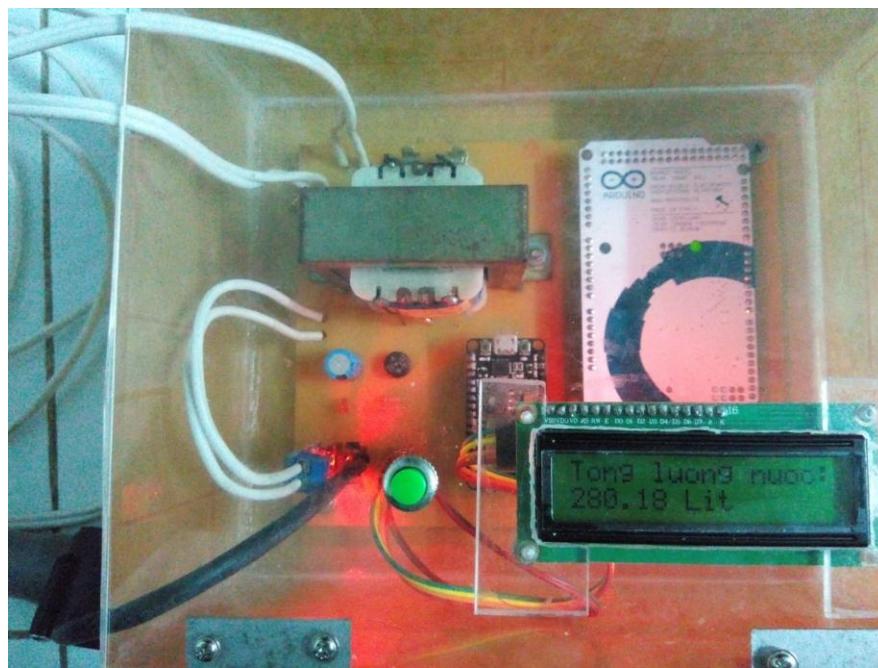
## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

---

### 5.1.2 Kết quả chạy hệ thống



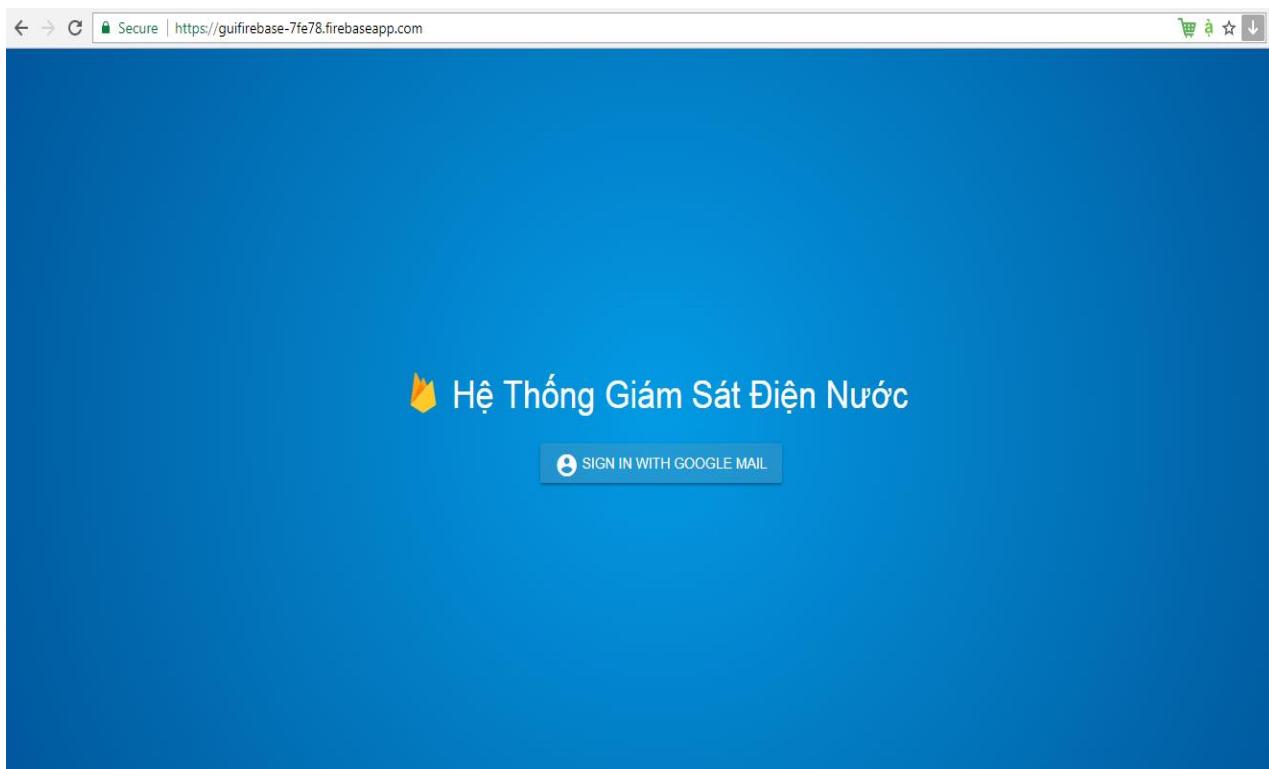
**Hình 5.1:** Màn hình LCD hiển thị giá trị điện năng



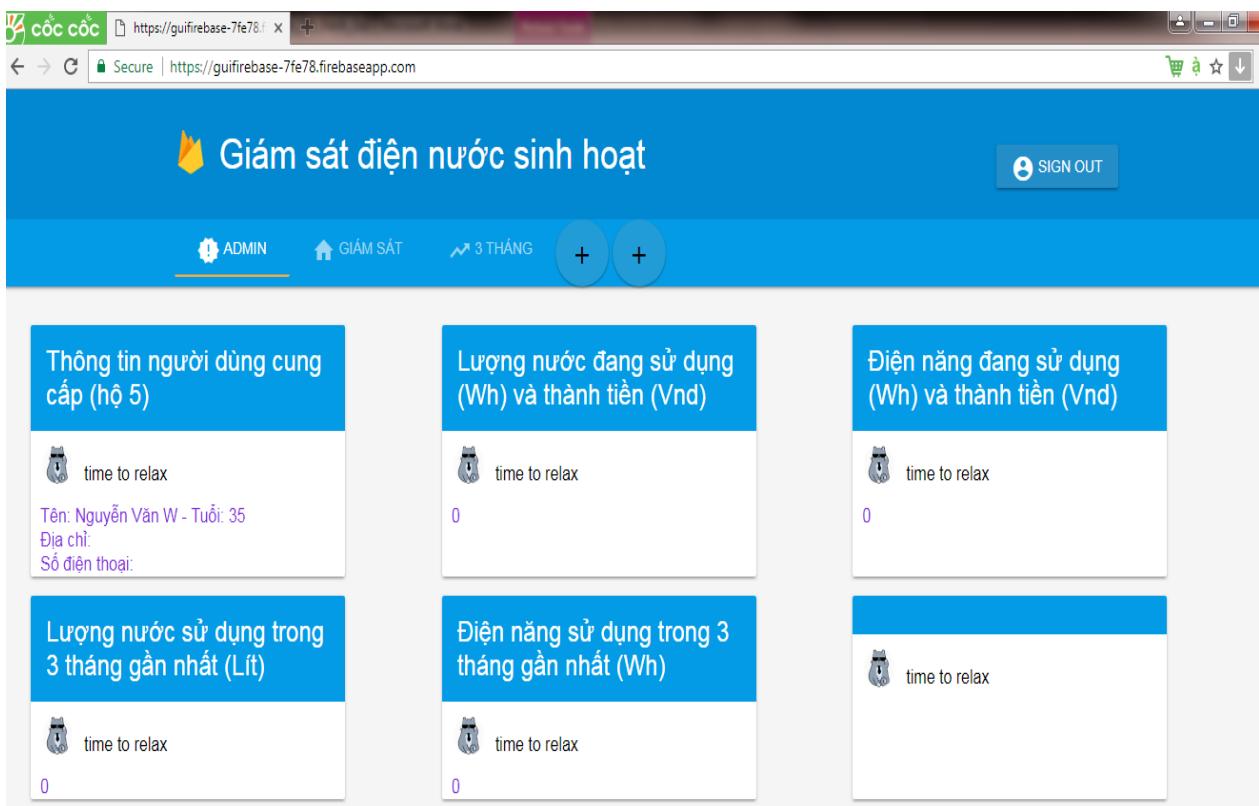
**Hình 5.2:** Màn hình LCD hiển thị giá trị nước sinh hoạt

Phía trên mặt mô hình, ta có một nút nhấn dùng để chuyển đổi hiển thị qua lại giữa 2 giá trị điện và nước.

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ



Hình 5.3: Giao diện đăng nhập của người dùng



Hình 5.4: Giao diện quản lý của admin - 1

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

Thông tin của người dùng (hộ 2)

iot doan

Tên: Nguyễn Văn X - Tuổi: 34  
Địa chỉ:  
Số điện thoại:

Lượng nước đang sử dụng (Wh) và thành tiền (Vnd)

iot doan

Nước: 28.21 Lít  
Thành tiền: 141050.00 vnđ

Điện năng đang sử dụng (Wh) và thành tiền (Vnd)

iot doan

Điện: 207.95 Wh  
Thành tiền: 415900.00 vnđ

Lượng nước sử dụng trong 3 tháng gần nhất (Lít)

iot doan

Tháng 5: 42.14 Lít  
Tháng 6: 46.24 Lít  
Tháng 7: 28.21 Lít

Điện năng sử dụng trong 3 tháng gần nhất (Wh)

iot doan

Tháng 5: 356.15 Wh  
Tháng 6: 324.81 Wh  
Tháng 7: 207.95 Wh

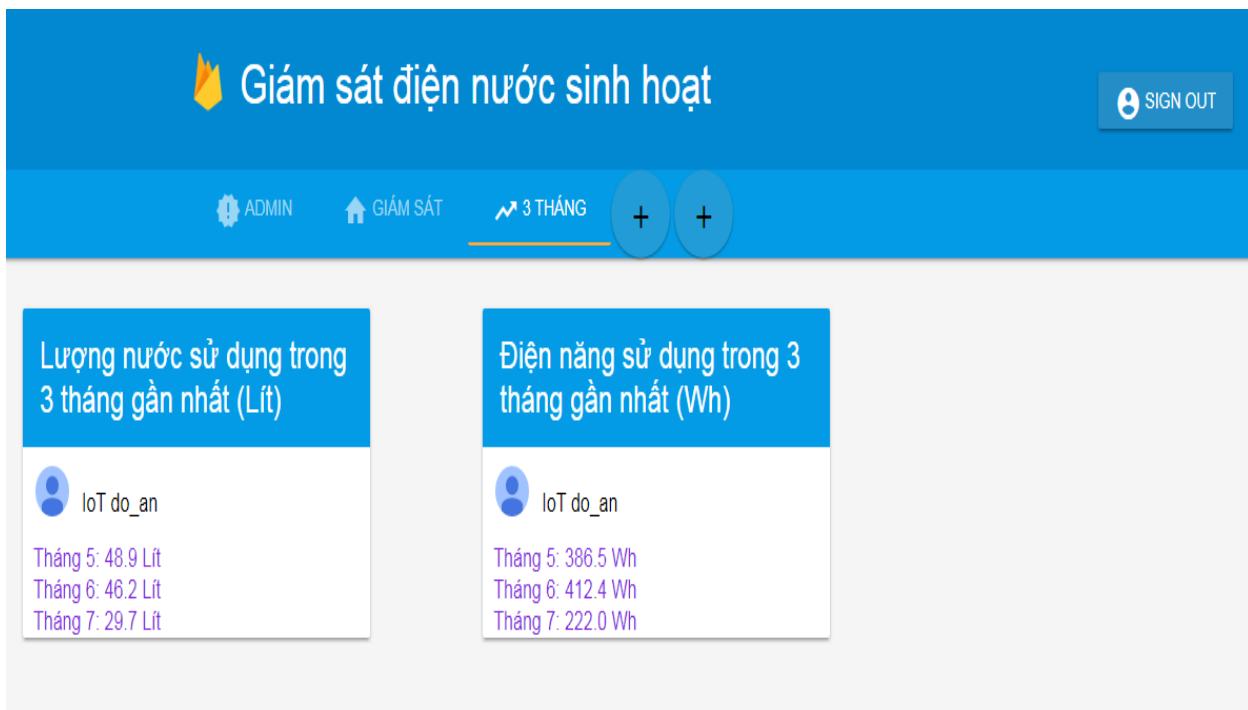
Hình 5.5: Giao diện quản lý của admin - 2

The screenshot shows a mobile application interface with a blue header bar. The title is "Giám sát điện nước sinh hoạt". On the right side of the header is a "SIGN OUT" button. Below the header, there are several cards displaying user information and usage statistics.

- Thông tin của người dùng (hộ 1)**  
IoT do\_an  
Tên: Nguyễn Văn F - Tuổi: 32  
Địa chỉ:  
Số điện thoại:
- Lượng nước đang sử dụng (Wh) và thành tiền (Vnd)**  
IoT do\_an  
Nước: 29.7 Lít  
Thành tiền: 148500.00 vnđ
- Điện năng đang sử dụng (Wh) và thành tiền (Vnd)**  
IoT do\_an  
Điện: 222.0 Wh  
Thành tiền: 444000.00 vnđ
- Lượng nước sử dụng theo ngày (Lít)**  
IoT do\_an  
10/7/2018: 5.2 Lít  
11/7/2018: 9.5 Lít  
12/7/2018: 3.7 Lít  
13/7/2018: 4.1 Lít  
14/7/2018: 4.8 Lít  
15/7/2018: 2.4 Lít
- Điện năng sử dụng theo ngày (Wh)**  
IoT do\_an  
10/7/2018: 42.3 Wh  
11/7/2018: 29.5 Wh  
12/7/2018: 39.7 Wh  
13/7/2018: 40.1 Wh  
14/7/2018: 48.0 Wh  
15/7/2018: 22.4 Wh

Hình 5.6: Giao diện quản lý giám sát của user - 1

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ



Hình 5.7: Giao diện quản lý giám sát của user - 2

Identifier	Providers	Created	Signed In	User UID
lifeofmine233@gmail.com	G	Jul 7, 2018	Jul 8, 2018	1KSF5zb0SRXRwZQYzzXM1NDRx...
itstimeforyou23@gmail.com	G	Jul 8, 2018	Jul 9, 2018	3ZqUNSl5a6MfgLXDTZKNGlyWEv2
doaniot123@gmail.com	G	Jun 16, 2018	Jul 11, 2018	DtpX2PGSHeUs0QXWkVCxaZM2N...
doaniot1234@gmail.com	G	Jun 16, 2018	Jul 11, 2018	OhPO9rPwfwmIsXgnUrTTDX5Jebb2
12141105@student.hcmute...	G	Jul 6, 2018	Jul 11, 2018	sySRJtfCMIQJB94aMqspdla9X4Y

Hình 5.8: Quản lý tài khoản người dùng trong database

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

Người dùng khi truy cập vào địa chỉ web thông qua một trình duyệt web như Chrome hoặc CocCoc, một giao diện đăng nhập sẽ hiện ra như hình 5.3, nhấn vào nút đăng nhập, một cửa sổ pop-up sẽ hiện ra và truy cập đến những tài khoản gmail có sẵn trên máy tính. Chọn một tài khoản mà bạn đã cung cấp cho admin của hệ thống, và admin đã duyệt để đăng nhập vào giao diện giám sát dành cho người dùng. Mỗi một tài khoản gmail của người dùng, khi lần đầu tiên đăng nhập vào hệ thống, các thông tin cơ bản như tên, địa chỉ mail, ảnh đại diện sẽ được lưu vào database của hệ thống, và truy xuất hiển thị trên mỗi form giám sát của họ. Mỗi người dùng có thẻ giám sát mức sử dụng điện – nước của họ theo ngày như hình 5.6 và họ cũng có thể so sánh trong ba tháng gần nhất như hình 5.7 (tính luôn tháng hiện tại). Ngoài ra, họ còn biết được số tiền phải trả được quy ra từ lượng điện – nước mà họ đang sử dụng hàng ngày. Đối với admin, họ có thể xem thông tin giám sát của tất cả người dùng thông qua tài khoản gmail được cho phép như hình 5.4, 5.5 và 5.6 hoặc có xem ở database của hệ thống.

Admin có thể dễ dàng quản lý được tất cả tài khoản của người dùng dựa trên tài khoản gmail được người dùng cung cấp như hình 5.8. Tại đây ta có thể thấy thời gian mà gmail đó đã đăng nhập, có thể thực hiện chặn/cho phép/ xóa tài khoản gmail đó; mỗi một gmail đều được gán một UID (user-id) riêng biệt ngay khi đăng nhập thành công vào hệ thống.

### **5.2 ĐÁNH GIÁ, NHẬN XÉT**

Mô hình thực hiện được công việc đo, và cập nhật dữ liệu lên web server theo thời gian, theo từng ngày nhưng vẫn còn sai số (chấp nhận được). Yêu cầu đặt ra là mô hình cần phải khắc phục được tình trạng nhiễu để hệ thống hoạt động chính xác hơn.

Web server hiển thị đầy đủ những thông tin giám sát cần thiết cho từng người dùng cụ thể, tuy nhiên còn hạn chế về phương diện đăng ký người dùng, hiện tại chỉ có thể đăng ký qua gmail; Giao diện hiển thị được những thông số giám sát điện – nước tiêu thụ cho từng người dùng theo từng ngày và theo tháng, và người quản lý có thể giám sát mức tiêu thụ điện – nước của tất cả người dùng giao diện web. Giao diện sử dụng các

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

form để hiển thị nội dung, giúp cho người dùng dễ dàng quan sát, và tạo cảm giác gần gũi do giống với giao diện của các smartphone hiện nay. Tuy nhiên, giao diện nhìn chung vẫn còn khá đơn giản, chỉ thực hiện những mục đích chính.

Đề tài chỉ sử dụng 2 mô hình tương ứng với 2 hộ gia đình sử dụng để gửi dữ liệu lên Firebase, việc gửi dữ liệu không gặp khó khăn gì; Tuy nhiên cũng phải cần xét trường hợp thực tế là có rất nhiều hộ gia đình cùng gửi dữ liệu lên Firebase cùng lúc, như vậy có xảy ra những sự cố trong quá trình truyền dữ liệu hay không? Đây là một câu hỏi.

Nhìn chung, thiết bị đo đơn giản, dễ sử dụng, màn hình LCD hiển thị thông tin chi tiết, rõ nét; vỏ hộp bên ngoài thiết bị chắc chắn, an toàn, chống thấm nước; linh kiện điện tử bên trong được bố trí gọn gàng, chắc chắn. Nếu tiếp tục phát triển cải tiến, sản phẩm có tính ứng dụng thực tế và có tiềm năng phát triển trên thị trường.

# **Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **6.1 KẾT LUẬN**

Sau hơn 3 tháng thực hiện đề tài với những nỗ lực và cố gắng của nhóm, cùng với sự hướng dẫn, góp ý tận tình của thầy Nguyễn Thanh Nghĩa, đề tài đã hoàn thành kịp tiến độ quy định theo yêu cầu đặt ra là giám sát mức tiêu thụ điện nước thông qua web server.

Về phần cứng, mô hình thực hiện đo mức tiêu thụ điện – nước sử dụng và cập nhật gửi lên trang web host. Mô hình được thiết kế với kích thước tương đối nhỏ gọn, bố trí bên trong gọn gàng, dễ quan sát. Tuy nhiên, hệ thống xuất hiện tình trạng nhiễu nên các giá trị đọc từ cảm biến chưa thật sự chính xác..

Về phần mềm, đã thiết kế được giao diện web để hiển thị các dữ liệu, cho phép các user đăng nhập vào hệ thống chỉ đơn giản bằng tài khoản gmail để giám sát mức tiêu thụ điện – nước của chính họ; đồng thời, admin được phép giám sát mức tiêu thụ của tất cả các user.

## **6.2 HƯỚNG PHÁT TRIỂN**

Nhóm còn nhiều ý tưởng để hệ thống phát triển với nhiều chức năng hơn cụ thể:

- Có hệ thống cảnh báo nếu giá trị điện – nước đo được vượt quá ngưỡng cài đặt cho phép (dùng loa, đèn báo), hoặc gửi tin nhắn qua sim.
- Thay màn hình LCD 16x2 đơn điệu bằng màn hình cảm ứng có giao diện trực quan.
- Mở rộng hệ thống để có thể điều khiển các thiết bị từ xa như: công tắc, van điện tử...
- Mô hình cần được cải tiến để trở nên nhỏ gọn và bắt mắt hơn.
- Website hỗ trợ để người dùng đăng nhập không chỉ từ tài khoản gmail mà còn từ các mạng xã hội khác nhau như: Facebook, Twitter, Github... Thiết kế giao diện website bắt mắt hơn, có nhiều chức năng hơn.
- Phân quyền truy cập cho người dùng cụ thể và chi tiết hơn trong Firebase Authentication.

# TÀI LIỆU THAM KHẢO

### Sách tham khảo

- [1] Hoàng Ngọc Văn, “Điện Tử Công Suất”, Trường đại học SPKT Tp.HCM, 2007
- [2] Trần Thu Hà (chủ biên), “Điện Tử Cơ Bản”, NXB ĐH Quốc Gia Tp.HCM, 2013
- [3] Nguyễn Đình Phú – Trương Ngọc Anh, “Vì xử lý”, NXB ĐH Quốc Gia Tp.HCM, 2013
- [4] Google, “Firebase Documentaion”, <https://firebase.google.com/docs/web/setup>
- [5] Bùi Minh Phúc, “ESP8266 NodeMCU - Một lựa chọn khác ngoài Adruino”,  
<https://www.makerspace.vn/2017/07/26/esp8266-nodemcu-mot-lua-chon-khac-ngoai-adruino/>, 26/07/2017
- [6] Phạm Trần Đăng Khoa, “Tự học thiết kế Web tĩnh cơ bản: HTML, CSS & jQuery tại KhoaPham.Vn”,  
[https://www.youtube.com/watch?v=7x1PDHsQyGw&list=PLzrVYRai0riRDreiNl\\_QJ\\_Oc9BEq3GE7Yp](https://www.youtube.com/watch?v=7x1PDHsQyGw&list=PLzrVYRai0riRDreiNl_QJ_Oc9BEq3GE7Yp), 2015
- [7] Material design lite tutorial, 2018  
<https://www.tutorialspoint.com/materialdesignlite/index.htm>
- [8] Instructables – How to make anything, <https://www.instructables.com/>
- [9] Github – The world’s leading software development platform <https://github.com/>
- [10] Giao tiếp I2C (Phần 1) <http://www.ytuongnhanh.vn>
- [11] Các chuẩn mạng WIFI <https://tinhte.vn>

## **PHỤ LỤC**

Chương trình của Arduino:

```
#include <SoftwareSerial.h>
#include <SimpleKalmanFilter.h>

SimpleKalmanFilter bo_loc(1, 1, 0.2);
SoftwareSerial mySerial(12, 13); // RX, TX

const int S201 = 2;    // Chan noi voi cam bien luu luong
const int ACS = A0;    // Chan noi voi ACS
const int DienAp = A1; // Chan noi voi mach do ap

unsigned int sensorValue1 = 0, sensorValue2 = 0, dem = 0;
unsigned long tim = 0;
float Vo = 0, Ampere = 0, Volt = 0, Dien = 0, Vo_K = 0;
unsigned int max1 = 0, min1 = 1023, max2 = 0;
volatile byte pulseCount;
float flowRate;
unsigned int flowsec, totalMilliLitres = 0;

void setup()
{
    Serial.begin(115200);
```

## **Phụ lục**

---

```
mySerial.begin(115200);

pinMode(S201, INPUT);
digitalWrite(S201, HIGH);

pulseCount      = 0;
flowRate       = 0.0;
flowsec        = 0;
totalMilliLitres = 0;

attachInterrupt(0, pulseCounter, FALLING);

}

void loop()
{
    // Code do dong, ap va luong dien tieu thu
    while (millis() - tim < 250)
    {
        sensorValue1 = analogRead(ACS);
        sensorValue2 = analogRead(DienAp);
        if (max1 < sensorValue1)
            max1 = sensorValue1;
    }
}
```

## Phụ lục

---

```
if (min1 > sensorValue1)
    min1 = sensorValue1;
if (max2 < sensorValue2)
    max2 = sensorValue2;
}
dem++;
tim = millis();
Vo = (max1 - min1) / 1024.0 * 2.5 / 1.4142;
Vo_K = bo_loc.updateEstimate(Vo); // Lọc nhiễu với thuật toán Kalman
max1 = 0, min1 = 1023;
if (dem % 4 == 0)
{
    Ampere = Vo_K * 10.0;
    Volt = ((max2 / 102.4) + 1.4) / 1.4142 / 6.0 * 220.0 + 3;
    Dien = Dien + (Volt * Ampere * 0.9);
    max2 = 0;
    /////////////////////////////////
// Code do luong nuoc tieu thu
    ///////////////////////////////
detachInterrupt(0); // Dung ngat de tinh toan
flowRate = pulseCount / 7.5; // Toc do dong chay Lit/phut
flowsec = flowRate / 60.0 * 1000; // Luong nuoc trong 1s mL
totalMilliLitres += flowsec; // Tong luong nuoc
pulseCount = 0;
```

Phụ lục

```
Serial.println(totalMilliLitres);

attachInterrupt(0, pulseCounter, FALLING);

}

//////////



// Code gui du lieu sang nodemcu

//////////


if (dem >= 40)

{

    mySerial.write(int(Dien / 10) >> 8);

    mySerial.write(int(Dien / 10));

    mySerial.write((totalMilliLitres + 30000) >> 8);

    mySerial.write(totalMilliLitres + 30000);

    dem = 0;

    totalMilliLitres = 0;

    Dien = 0;

}

Serial.println(millis());

}

void pulseCounter()

{

    pulseCount++;

}
```

## Phụ lục

---

Chương trình của Nodemcu:

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NTPtimeESP.h>

#define FIREBASE_HOST "guifirebase-7fe78.firebaseio.com" // Link save data tren Firebase
#define FIREBASE_AUTH "AdU9m9jwZ3YkCQmQvRYSFyNtuj9dLwiKcVwdulwX" // Pass
dang nhap Firebase
#define WIFI_SSID "Thanh Ti" // Ten dang nhap wifi
#define WIFI_PASSWORD "Ting1235" // Pass dang nhap wifi

NTPtime NTPch("ch.pool.ntp.org"); // Server NTP
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dia chi va kich thuoc LCD
SoftwareSerial mySerial(14, 12); // RX, TX
const int btnPin = 2; // Chan ket noi nut nhan

strDateTime date;
unsigned long tg = 0;
unsigned int tam = 0;
float Dien = 0, Nuoc = 0;
byte ht = 0, i = 0;

///////////
// Chuong trinh ngat xu li nut nhan
///////////

void ButtonPress()
{

```

## Phụ lục

---

```
detachInterrupt(btnPin);
tg = millis();
while (millis() - tg < 65);
if (digitalRead(btnPin) == 0)
{
    if (ht == 0)
        ht = 1;
    else
        ht = 0;
    htLCD();
}
attachInterrupt(btnPin, ButtonPress, FALLING);
}

///////////////////////////////
// Chuong trinh con hien thi LCD
/////////////////////////////
void htLCD()
{
    if (ht == 0)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Tong luong dien:");
        lcd.setCursor(0, 1);
        lcd.print(String(Dien) + " Wh");
    }
    else if (ht == 1)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
```

## Phụ lục

---

```
lcd.print("Tong luong nuoc:");
lcd.setCursor(0, 1);
lcd.print(String(Nuoc) + " Lit");
}

}

void setup()
{
Serial.begin(115200);
mySerial.begin(115200);
lcd.begin();
lcd.backlight();
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("Connecting...");
// connect to wifi.
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting");
while (WiFi.status() != WL_CONNECTED)
{
Serial.print(".");
delay(500);
if (millis() > 30000)
{
WiFi.beginSmartConfig();
Serial.println(".");
Serial.print("beginSmartConfig");
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("Time out");
lcd.setCursor(0, 1);
lcd.print("beginSmartConfig");
```

## Phụ lục

---

```
while (1)
{
    Serial.print(".");
    delay(500);
    if (WiFi.smartConfigDone())
    {
        Serial.println("SmartConfig Success");
        delay(500);
        break;
    }
}
lcd.clear();
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());
lcd.setCursor(1, 0);
lcd.print("connected:");
lcd.setCursor(1, 1);
lcd.print(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
pinMode(btnPin, INPUT_PULLUP);
// Đọc dữ liệu từ Firebase để tiếp tục tính toán mức tiêu thụ
Dien = Firebase.getFloat("IoTdoan/dien/thang");
Nuoc = Firebase.getFloat("IoTdoan/nuoc/thang");

attachInterrupt(btnPin, ButtonPress, FALLING);
}

void loop()
```

## Phụ lục

---

```
{  
    dateTIme = NTPch.getNTPtime(7.0, 0); // Chọn múi giờ +7, tắt chế độ daylight  
    while (mySerial.available() > 0)  
    {  
        delay(5);  
        if (i == 0)  
        {  
            tam = mySerial.read();  
            i = 1;  
        }  
        else  
        {  
            tam = (tam << 8) + mySerial.read();  
            if (tam < 30000)  
                Dien = Dien + (tam / 360.0);  
            else  
                Nuoc = Nuoc + ((tam - 30000) / 1000.0);  
            htLCD();  
            tam = 0;  
            i = 0;  
        }  
    }  
    if (dateTIme.second == 0)  
    {  
        if (dateTIme.minute % 2 == 0)  
        {  
            ////////// Cập nhật biến  
            Firebase.setFloat("IoTdoan/dien/thang", Dien);  
            Firebase.setFloat("IoTdoan/nuoc/thang", Nuoc);  
  
            ////////// Send to User, Dien
    }
}
```

## Phụ lục

---

```
// Tổng
Firebase.setString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-LHMG-
rl4bGoEvtIMuj9/body", "Điện: " + String(Dien)
    + " Wh\nThành tiền: " + String(int(Dien) * Firebase.getFloat("giatien/tiendien"))
+ " vnd");
// Ngày
Firebase.setString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMF9biNkxP5oJerpDu/body",
    Firebase.getString("IoTdoan/dien/ngay") + "\n" + String(dateTime.day) + "/" +
String(dateTime.month)
    + "/" + String(dateTime.year) + ":" + String(Dien -
Firebase.getFloat("IoTdoan/dien/tong")) + " Wh");
// Tháng
if (dateTime.month == 1)
    Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMDwvufBpphcG9Q_ai/body",
        "Tháng 11: " + String(Firebase.getFloat("IoTdoan/dien/thang2"))
        + " Wh\nTháng 12: " + String(Firebase.getFloat("IoTdoan/dien/thang1"))
        + " Wh\nTháng 1: " + String(Dien) + " Wh");
else if (dateTime.month == 2)
    Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMDwvufBpphcG9Q_ai/body",
        "Tháng 12: " + String(Firebase.getFloat("IoTdoan/dien/thang2"))
        + " Wh\nTháng 1: " + String(Firebase.getFloat("IoTdoan/dien/thang1"))
        + " Wh\nTháng 2: " + String(Dien) + " Wh");
else
    Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMDwvufBpphcG9Q_ai/body",
        "Tháng " + String(dateTime.month - 2) + ":" +
String(Firebase.getFloat("IoTdoan/dien/thang2"))
```

## Phụ lục

---

```
+ " Wh\nTháng " + String(dateTime.month - 1) + ":" +
String(Firebase.getFloat("IoTdoan/dien/thang1"))
+ " Wh\nTháng " + String(dateTime.month) + ":" + String(Dien) + " Wh");

////////// Send to User, Nuoc
// Tông
Firebase.setString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMGSP_hlGUI_j2UFrJ/body", "Nước: " + String(Nuoc)
+ " Lít\nThành tiền: " + String(int(Nuoc) * Firebase.getFloat("giatien/tiennuoc"))
+ " vnd");
// Ngày
Firebase.setString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMFVzLLktqR8Oekw5W/body",
Firebase.getString("IoTdoan/nuoc/ngay") + "\n" + String(dateTime.day) + "/" +
String(dateTime.month)
+ "/" + String(dateTime.year) + ":" + String(Nuoc) -
Firebase.getFloat("IoTdoan/nuoc/tong")) + " Lít");
// Tháng
if (dateTime.month == 1)
Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMETi5wFvfSRACMlH0/body",
"Tháng 11: " + String(Firebase.getFloat("IoTdoan/nuoc/thang2"))
+ " Lít\nTháng 12: " + String(Firebase.getFloat("IoTdoan/nuoc/thang1"))
+ " Lít\nTháng 1: " + String(Nuoc) + " Lít");
else if (dateTime.month == 2)
Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMETi5wFvfSRACMlH0/body",
"Tháng 12: " + String(Firebase.getFloat("IoTdoan/nuoc/thang2"))
+ " Lít\nTháng 1: " + String(Firebase.getFloat("IoTdoan/nuoc/thang1"))
+ " Lít\nTháng 2: " + String(Nuoc) + " Lít");
else
```

## Phụ lục

---

```
Firebase.setString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-  
LHMETi5wFvfSRACMIH0/body",  
    "Tháng " + String(dateTime.month - 2) + ":" +  
    String(Firebase.getFloat("IoTdoan/nuoc/thang2"))  
    + " Lít\nTháng " + String(dateTime.month - 1) + ":" +  
    String(Firebase.getFloat("IoTdoan/nuoc/thang1"))  
    + " Lít\nTháng " + String(dateTime.month) + ":" + String(Nuoc) + " Lít");  
  
////////// Send to Admin, Dien  
// Tổng  
Firebase.setString("posts/-LHMG-rl4bGoEvtIMuj9/body",  
    Firebase.getString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-LHMG-  
rl4bGoEvtIMuj9/body"));  
// Tháng  
Firebase.setString("posts/-LHMDwvufBpphcG9Q_ai/body",  
    Firebase.getString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-  
LHMDwvufBpphcG9Q_ai/body"));  
  
////////// Send to Admin, Nuoc  
// Tổng  
Firebase.setString("posts/-LHMGSP_hlGUI_j2UFrJ/body",  
    Firebase.getString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-  
LHMGSP_hlGUI_j2UFrJ/body"));  
// Tháng  
Firebase.setString("posts/-LHMETi5wFvfSRACMIH0/body",  
    Firebase.getString("month/IXr3SMUIkveztZHAjgCuCtND13c2/-  
LHMETi5wFvfSRACMIH0/body"));  
if (dateTime.minute == 0)  
{  
    if (dateTime.hour == 0)  
    {
```

## Phụ lục

---

```
Firebase.setString("IoTdoan/dien/ngay",
    Firebase.getString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMF9biNkxP5oJerpDu/body")));
Firebase.setFloat("IoTdoan/dien/tong", Dien);

Firebase.setString("IoTdoan/nuoc/ngay",
    Firebase.getString("user-posts/IXr3SMUIkveztZHAjgCuCtND13c2/-
LHMFVzLLktqR8Oekw5W/body")));
Firebase.setFloat("IoTdoan/nuoc/tong", Nuoc);
if (dateTime.day == 1)
{
    // Reset data, Dien
    Firebase.setFloat("IoTdoan/dien/thang2", Firebase.getFloat("IoTdoan/dien/thang1"));
    Firebase.setFloat("IoTdoan/dien/thang1", Firebase.getFloat("IoTdoan/dien/thang"));
    Firebase.setFloat("IoTdoan/dien/thang", 0);
    Firebase.setFloat("IoTdoan/dien/tong", 0);

    // Reset data, Nuoc
    Firebase.setFloat("IoTdoan/nuoc/thang2", Firebase.getFloat("IoTdoan/nuoc/thang1"));
    Firebase.setFloat("IoTdoan/nuoc/thang1", Firebase.getFloat("IoTdoan/nuoc/thang"));
    Firebase.setFloat("IoTdoan/nuoc/thang", 0);
    Firebase.setFloat("IoTdoan/nuoc/tong", 0);

    Dien = 0;
    Nuoc = 0;
}
```

### Chương trình code web:

#### Index.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="description" content="Demonstrates the use of Google Cloud Database with a
Firebase DB">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Firebase IO</title>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
<link rel="stylesheet" href="https://code.getmdl.io/1.1.3/material.blue_grey-orange.min.css">
<script src="https://code.getmdl.io/1.1.3/material.min.js"></script>

<link rel="stylesheet" href="main.css">

</head>
<body>
<div class="demo-layout mdl-layout mdl-js-layout mdl-layout--fixed-header">
<section id="page-splash">
<h3 class="logo">Hệ Thông Giám Sát Điện Nước Sinh Hoạt</h3>
<div>
<button id="sign-in-button" class="mdl-button--raised mdl-button mdl-js-button mdl-js-
ripple-effect"><i class="material-icons">account_circle</i> Sign in with Google Mail</button>
</div>
</section>

<header class="header mdl-layout__header mdl-color-text--white mdl-color--light-blue-700">
```

## Phụ lục

---

```
<div class="mdl-layout__header-row titlebar">
    <h3 class="logo">Home water-electricity monitor</h3>
    <button id="sign-out-button" class="mdl-button--raised mdl-button mdl-js-button mdl-js-ripple-effect"><i class="material-icons">account_circle</i> Sign out</button>
</div>

<div class="tab mdl-layout__header-row mdl-color--light-blue-600">
    <div class="mdl-tab">
        <div id="menu-recent" class="mdl-layout__tab is-active mdl-button mdl-js-button mdl-js-ripple-effect">
            <i class="material-icons">new_releases</i> Admin
        </div>
        <div id="menu-my-posts" class="mdl-layout__tab mdl-button mdl-js-button mdl-js-ripple-effect">
            <i class="material-icons">home</i> My monitor
        </div>
        <div id="menu-my-top-posts" class="mdl-layout__tab mdl-button mdl-js-button mdl-js-ripple-effect">
            <i class="material-icons">trending_up</i> Notice
        </div>
        <button class="mdl-button mdl-js-button mdl-button--fab mdl-color--amber-400 mdl-shadow--4dp mdl-js-ripple-effect" id="add">
            <i class="material-icons">mode_edit</i>
        </button>
    </div>
</div>
</header>

<main class="mdl-layout__content mdl-color--grey-100">

    <section class="mdl-grid content" id="add-post" style="display:none">
```

## Phụ lục

---

```
<div class="mdl-cell mdl-cell--12-col mdl-grid">

    <div class="mdl-card mdl-shadow--2dp mdl-cell mdl-cell--12-col mdl-cell--8-col-tablet
        mdl-cell--6-col-desktop">
        <div class="mdl-card__title mdl-color--light-blue-600 mdl-color-text--white">
            <h2 class="mdl-card__title-text">New Post</h2>
        </div>
        <div class="mdl-card__supporting-text mdl-color-text--grey-600">
            <form id="message-form" action="#">
                <div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label">
                    <input class="mdl-textfield__input" type="text" id="new-post-title">
                    <label class="mdl-textfield__label" for="new-post-title">Post title...</label>
                </div>
                <div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label">
                    <textarea class="mdl-textfield__input" rows="3" id="new-post-
message"></textarea>
                    <label class="mdl-textfield__label" for="new-post-message">Post message...</label>
                </div>
                <button type="submit" class="mdl-button mdl-js-button mdl-button--raised mdl-js-
ripple-effect">
                    Add post
                </button>
            </form>
        </div>
    </div>
</div>
```

---

```
<section class="mdl-grid content" id="recent-posts-list" style="display:none">
    <div class="posts-container mdl-cell mdl-cell--12-col mdl-grid">
    </div>
```

## Phụ lục

---

```
</section>

<section class="mdl-grid content" id="user-posts-list" style="display:none">
  <div class="posts-container mdl-cell mdl-cell--12-col mdl-grid">
    </div>
</section>

<section class="mdl-grid content" id="top-user-posts-list" style="display:none">
  <div class="posts-container mdl-cell mdl-cell--12-col mdl-grid">
    </div>
</section>
</main>
</div>

<script src="/__/firebase/5.0.0/firebase-app.js"></script>
<script src="/__/firebase/5.0.0/firebase-auth.js"></script>
<script src="/__/firebase/5.0.0.firebaseio-database.js"></script>
<script src="/__/firebase/init.js"></script>

<script src="scripts/main.js"></script>
</body>
</html>
```

### Main.js

```
'use strict';

var messageForm = document.getElementById('message-form');
var messageInput = document.getElementById('new-post-message');
var titleInput = document.getElementById('new-post-title');
var signInButton = document.getElementById('sign-in-button');
var signOutButton = document.getElementById('sign-out-button');
```

## Phụ lục

---

```
var splashPage = document.getElementById('page-splash');
var addPost = document.getElementById('add-post');
var addButton = document.getElementById('add');
var recentPostsSection = document.getElementById('recent-posts-list');
var userPostsSection = document.getElementById('user-posts-list');
var topUserPostsSection = document.getElementById('top-user-posts-list');
var recentMenuButton = document.getElementById('menu-recent');
var myPostsMenuButton = document.getElementById('menu-my-posts');
var myTopPostsMenuButton = document.getElementById('menu-my-top-posts');
var listeningFirebaseRefs = [];

function writeNewPost(uid, username, picture, title, body) {
    var postData = {
        author: username,
        uid: uid,
        body: body,
        title: title,
        authorPic: picture
    };
    var newPostKey = firebase.database().ref().child('posts').push().key;
    var updates = { };
    updates['/posts/' + newPostKey] = postData;
    updates['/user-posts/' + uid + '/' + newPostKey] = postData;
    return firebase.database().ref().update(updates);
}

function createPostElement(postId, title, text, author, authorId, authorPic) {
    var uid = firebase.auth().currentUser.uid;
    var html =
        '<div class="post post-' + postId + ' mdl-cell mdl-cell--12-col ' +
        'mdl-cell--6-col-tablet mdl-cell--4-col-desktop mdl-grid mdl-grid--no-spacing">' +
```

## Phụ lục

---

```
'<div class="mdl-card mdl-shadow--2dp">' +
'<div class="mdl-card__title mdl-color--light-blue-600 mdl-color-text--white">' +
'<h4 class="mdl-card__title-text"></h4>' +
'</div>' +
'<div class="header">' +
'<div>' +
'<div class="avatar"></div>' +
'<div class="username mdl-color-text--black"></div>' +
'</div>' +
'</div>' +  
  
'<div class="text"></div>' +
'</div>' +
'</div>;  
  
var div = document.createElement('div');
div.innerHTML = html;
var postElement = div.firstChild;  
  
postElement.getElementsByClassName('text')[0].innerText = text;
postElement.getElementsByClassName('mdl-card__title-text')[0].innerText = title;
postElement.getElementsByClassName('username')[0].innerText = author || 'Anonymous';
postElement.getElementsByClassName('avatar')[0].style.backgroundImage = 'url("' +
(authorPic || './silhouette.jpg') + ""');  
  
return postElement;
}  
  
function startDatabaseQueries() {
  var myUserId = firebase.auth().currentUser.uid;
  var recentPostsRef = firebase.database().ref('posts').limitToLast(100);
```

## Phụ lục

---

```
var topUserPostsRef = firebase.database().ref('notice').limitToLast(100);

var userPostsRef = firebase.database().ref('user-posts/' + myUserId);

var fetchPosts = function(postsRef, sectionElement) {
  postsRef.on('child_added', function(data) {
    var author = data.val().author || 'Anonymous';
    var containerElement = sectionElement.getElementsByClassName('posts-container')[0];
    containerElement.insertBefore(
      createPostElement(data.key, data.val().title, data.val().body, author, data.val().uid,
data.val().authorPic),
      containerElement.firstChild);
  });
  postsRef.on('child_changed', function(data) {
    var containerElement = sectionElement.getElementsByClassName('posts-container')[0];
    var postElement = containerElement.getElementsByClassName('post-' + data.key)[0];
    postElement.getElementsByClassName('mdl-card__title-text')[0].innerText = data.val().title;
    postElement.getElementsByClassName('username')[0].innerText = data.val().author;
    postElement.getElementsByClassName('text')[0].innerText = data.val().body;
  });
  postsRef.on('child_removed', function(data) {
    var containerElement = sectionElement.getElementsByClassName('posts-container')[0];
    var post = containerElement.getElementsByClassName('post-' + data.key)[0];
    post.parentElement.removeChild(post);
  });
};

fetchPosts(topUserPostsRef, topUserPostsSection);
fetchPosts(recentPostsRef, recentPostsSection);
fetchPosts(userPostsRef, userPostsSection);
```

## Phụ lục

---

```
listeningFirebaseRefs.push(topUserPostsRef);
listeningFirebaseRefs.push(recentPostsRef);
listeningFirebaseRefs.push(userPostsRef);
}

function writeUserData(userId, name, email, imageUrl) {
  firebase.database().ref('users/' + userId).set({
    username: name,
    email: email,
    profile_picture : imageUrl
  });
}

function cleanupUi() {
  topUserPostsSection.getElementsByClassName('posts-container')[0].innerHTML = "";
  recentPostsSection.getElementsByClassName('posts-container')[0].innerHTML = "";
  userPostsSection.getElementsByClassName('posts-container')[0].innerHTML = "";

  listeningFirebaseRefs.forEach(function(ref) {
    ref.off();
  });
  listeningFirebaseRefs = [];
}

var currentUID;
var admin = "DtpX2PGSHeUs0QXWkVCxaZM2Njv2";
var n = 1;

function onAuthStateChanged(user) {
  if (user && currentUID === user.uid) {
```

## Phụ lục

---

```
return;
}

cleanupUi();
if (user) {
    currentUID = user.uid;
    if(currentUID === admin)
    {
        n=1;
    }
    else
    {
        n=0;
    }
    splashPage.style.display = 'none';
    writeUserData(user.uid, user.displayName, user.email, user.photoURL);
    startDatabaseQueries();
} else {
    currentUID = null;
    splashPage.style.display = "";
}
}

function newPostForCurrentUser(title, text) {
    var userId = firebase.auth().currentUser.uid;
    return firebase.database().ref('/users/' + userId).once('value').then(function(snapshot) {
        var username = (snapshot.val() && snapshot.val().username) || 'Anonymous';

        return writeNewPost(firebase.auth().currentUser.uid, username,
            firebase.auth().currentUser.photoURL,
            title, text);
    })
}
```

## Phụ lục

---

```
)};

}

function showSection(sectionElement, buttonElement) {
    recentPostsSection.style.display = 'none';
    userPostsSection.style.display = 'none';
    topUserPostsSection.style.display = 'none';

    addPost.style.display = 'none';
    myTopPostsMenuButton.classList.remove('is-active');

    recentMenuButton.classList.remove('is-active');
    myPostsMenuButton.classList.remove('is-active');

    if (sectionElement) {
        sectionElement.style.display = 'block';
    }
    if (buttonElement) {
        buttonElement.classList.add('is-active');
    }
}

window.addEventListener('load', function() {
    signInButton.addEventListener('click', function() {
        var provider = new firebase.auth.GoogleAuthProvider();
        firebase.auth().signInWithPopup(provider);
    });

    signOutButton.addEventListener('click', function() {
        firebase.auth().signOut();
    });
});
```

## Phụ lục

---

```
firebase.auth().onAuthStateChanged(onAuthStateChanged);
messageForm.onsubmit = function(e) {
    e.preventDefault();
    var text = messageInput.value;
    var title = titleInput.value;
    if (text && title) {
        newPostForCurrentUser(title, text).then(function() {
            myPostsMenuButton.click();
        });
        messageInput.value = "";
        titleInput.value = "";
    }
};

recentMenuButton.onclick = function() {
    if(n==1)
    {
        showSection(recentPostsSection, recentMenuButton);
    }
};

myPostsMenuButton.onclick = function() {
    showSection(userPostsSection, myPostsMenuButton);
};

myTopPostsMenuButton.onclick = function() {
    showSection(topUserPostsSection, myTopPostsMenuButton);
};

addButton.onclick = function() {
    showSection(addPost);
```

## **Phụ lục**

---

```
messageInput.value = "";
titleInput.value = "";
};

myPostsMenuButton.onclick;
}, false);
```