

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC BÀ RỊA-VŨNG TÀU



**BARIA VUNGTAU**  
UNIVERSITY  
CAP SAINT JACQUES

## **ĐỒ ÁN TỐT NGHIỆP**

### **THIẾT KẾ, THI CÔNG MÔ HÌNH NHÀ THÔNG MINH**

<b>Trình độ đào tạo</b>	<b>: Đại học chính quy</b>
<b>Ngành</b>	<b>: Công nghệ kỹ thuật điện – Điện tử</b>
<b>Chuyên ngành</b>	<b>: Điều khiển và tự động hóa</b>
<b>Giảng viên hướng dẫn</b>	<b>: ThS. Lưu Hoàng</b>
<b>Sinh viên thực hiện</b>	<b>: Nguyễn Nhật Tiến</b>
<b>MSSV: 19033788</b>	<b>Lớp: DH19TD</b>

**TRƯỜNG ĐẠI HỌC BÀ RỊA-VŨNG TÀU**  
**KHOA KỸ THUẬT – CÔNG NGHỆ**

---

**PHIẾU GIAO ĐỀ TÀI**  
**ĐỒ ÁN/ KHOÁ LUẬN TỐT NGHIỆP**

*(Đính kèm Quy định về việc tổ chức, quản lý các hình thức tốt nghiệp ĐH, CĐ ban hành kèm theo Quyết định số 585/QĐ-ĐHBRVT ngày 16/7/2013 của Hiệu trưởng Trường Đại học BR-VT)*

---

**Họ và tên sinh viên:** Nguyễn Nhật Tiến      Ngày sinh: 14/11/2001

MSSV: 19034176      Lớp: DH19TD

E-mail: nguyennhattien12304@gmail.com

Hệ đào tạo: Đại học chính quy

Chuyên ngành: Điều khiển và Tự động hóa

**1. Tên đề tài:** THIẾT KẾ, THI CÔNG MÔ HÌNH NHÀ THÔNG MINH.....

**2. Giảng viên hướng dẫn:** ThS. Lưu Hoàng

**3. Ngày giao đề tài:** 23/12/2022

**4. Ngày hoàn thành đồ án/ khoá luận tốt nghiệp:** 30/4/2023

*Bà Rịa-Vũng Tàu, ngày.... tháng.... năm ..*

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**SINH VIÊN THỰC HIỆN**

*(Ký và ghi rõ họ tên)*

ThS. Lưu Hoàng

Nguyễn Nhật Tiến

**TRƯỞNG BỘ MÔN**

*(Ký và ghi rõ họ tên)*

**TRƯỞNG KHOA**

*(Ký và ghi rõ họ tên)*

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

- Thái độ, tác phong và nhận thức trong quá trình thực hiện:

.....

...

.....

....

.....

....

- Kiến thức chuyên môn:

.....

..

.....

..

.....

..

- Hình thức, bố cục trình bày:

.....

..

.....

..

.....

..

- Nội dung, kết quả:

.....

..

.....

..

.....

..

• Nhận xét khác:

.....

..

.....

..

.....

..

Bà Rịa – Vũng Tàu, ngày ... tháng ... năm 2023

**Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

## NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

- Thái độ, tác phong và nhận thức trong quá trình thực hiện:

.....

...

.....

....

.....

....

- Kiến thức chuyên môn:

.....

..

.....

..

.....

..

- Hình thức, bố cục trình bày:

.....

..

.....

..

.....

..

- Nội dung, kết quả:

.....

..

.....

.....

...

• Nhận xét khác:

.....

..

.....

..

.....

..

Bà Rịa – Vũng Tàu, ngày ... tháng ... năm 2023

**Giảng viên phản biện**  
(Ký và ghi rõ họ tên)

## LỜI CẢM ƠN

Thực tế không có sự thành công nào mà không có sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của mọi người xung quanh. Trong quá trình làm đồ án tốt nghiệp em nhận được sự giúp đỡ nhiệt tình của giảng viên hướng dẫn. Em xin cảm ơn thầy Lưu Hoàng – người đã tận tình hướng dẫn, giúp đỡ cho em trong khi lựa chọn đề tài cũng như trong quá trình thực hiện đề tài này. Khi thực hiện đồ án cũng đã xảy ra nhiều khó khăn, thiếu sót nhưng được sự hỗ trợ và góp ý của Thầy nên bản thân em đã hoàn thành được đồ án.

Trong suốt thời gian được theo học tại trường Đại Học Bà Rịa Vũng Tàu, em đã nhận được nhiều sự quan tâm và giúp đỡ từ thầy cô và bạn bè. Cảm ơn tới hiệu trưởng cùng các quý thầy cô trường Đại Học Bà Rịa – Vũng Tàu đã hỗ trợ tận tình về trang thiết bị, phần mềm, cơ sở vật chất tạo điều kiện hoàn thành đồ án. Với lòng biết ơn sâu sắc, em xin gửi lời cảm ơn tới quý thầy cô, những người đã truyền lại cho em rất nhiều kinh nghiệm và kiến thức quý báu, những sự giúp đỡ ấy đã tiếp thêm động lực cho em vững bước trên con đường mình đã chọn. Và đặc biệt là thầy, cô khoa Điện - Điện tử đã truyền đạt kiến thức, kinh nghiệm cũng như tạo những điều kiện tốt nhất để nhóm em hoàn thành đề tài.

Xin cảm ơn các bạn cùng khóa, cùng khoa đã đồng viên, khích lệ, ủng hộ về nhiều mặt góp phần làm nên sự hoàn thiện của đồ án này.

Cảm ơn trường Đại Học Bà Rịa Vũng Tàu! Em xin chân thành cảm ơn!

*Người thực hiện đề tài*

*Nguyễn Nhật Tiến*

## MỤC LỤC

LỜI CẢM ƠN.....	7
CHƯƠNG I: TỔNG QUAN .....	13
CHƯƠNG II: CƠ SỞ LÝ THUYẾT .....	19
2.1    GIỚI THIỆU VỀ PHẦN CỨNG.....	19
2.1.1    Arduino Uno R3.....	19
2.1.2    Màn hình LCD 20x4 .....	22
2.1.3    Mạch giao tiếp LCD I2C.....	23
2.1.4    Cảm biến mưa .....	24
2.1.5    Động cơ servo SG90 .....	26
2.1.6    Cảm biến khí gas MQ2 .....	27
2.1.7    Kit ESP8266.....	29
2.1.8    Cảm biến DHT11 .....	30
CHƯƠNG III: NỘI DUNG ĐỀ TÀI .....	32
3.1    GIỚI THIỆU .....	32
3.2    TÍNH TOÁN VÀ THIẾT KẾ.....	33
3.2.1    Sơ đồ khối.....	33
3.2.2    Sơ đồ kết cấu.....	34
3.2.3    Giao tiếp ESP8266 chuẩn UART .....	36
3.2.4    Các đặc tính kỹ thuật của chuẩn RS-232 .....	36
3.2.5    Các mức điện áp của đường truyền: .....	37
3.2.6    Sơ đồ chân công kết nối.....	38
3.2.7    Truyền dữ liệu.....	39
3.2.8    Giao tiếp chuẩn I2C .....	41
3.2.9    Gửi dữ liệu đến thiết bị Slave .....	42
3.2.10    Các chế độ hoạt động của I2C .....	47
3.3    CHƯƠNG TRÌNH ĐIỀU KHIỂN .....	48
3.3.1    Hệ thống cửa và đèn .....	48
3.3.2    Hệ thống quạt, đèn: .....	51
3.3.3    Hệ thống cảm biến và LCD : .....	56
3.3.4    Mô hình hoàn thiện .....	64
3.3.5    Điều khiển các thiết bị trong nhà qua giao diện web.....	71
3.3.6    Truyền dữ liệu từ các thiết bị lên giao diện web.....	72
3.3.7    Điều khiển các thiết bị trong nhà và truyền dữ liệu từ các thiết bị lên giao diện app điện thoại.....	73
CHƯƠNG IV: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN .....	74



4.1	KẾT QUẢ ĐẠT ĐƯỢC:.....	74
4.2	ĐÁNH GIÁ KẾT QUẢ:.....	74
4.3	KẾT LUẬN.....	74
4.4	HƯỚNG PHÁT TRIỂN.....	75
	TÀI LIỆU THAM KHẢO.....	77

## MỤC LỤC HÌNH ẢNH

<i>Hình 1.1</i>	<i>Mô hình tổng quát nhà thông minh.....</i>	<i>14</i>
<i>Hình 2.1</i>	<i>Sơ đồ Arduino Uno R3.....</i>	<i>20</i>
<i>Hình 2.2</i>	<i>Màn hình LCD.....</i>	<i>22</i>
<i>Hình 2.3</i>	<i>Mạch I2C.....</i>	<i>24</i>
<i>Hình 2.4</i>	<i>Cảm biến mưa.....</i>	<i>25</i>
<i>Hình 2.5</i>	<i>Động cơ servo.....</i>	<i>26</i>
<i>Hình 2.6</i>	<i>Cảm biến gas MQ2.....</i>	<i>28</i>
<i>Hình 2.7</i>	<i>Sơ đồ chân của ESP8266.....</i>	<i>29</i>
<i>Hình 2.8</i>	<i>Sơ đồ chân của DHT11.....</i>	<i>30</i>
<i>Hình 3.1</i>	<i>Các chức năng sử dụng.....</i>	<i>32</i>
<i>Hình 3.2</i>	<i>Sơ đồ khối.....</i>	<i>33</i>
<i>Hình 3.3</i>	<i>Sơ đồ kết cấu ngôi nhà thông minh.....</i>	<i>34</i>
<i>Hình 3.4</i>	<i>Mô hình hoàn thiện.....</i>	<i>35</i>
<i>Hình 3.5</i>	<i>Mô hình hoàn thiện.....</i>	<i>35</i>
<i>Hình 3.6</i>	<i>Màn hình LCD phía trong và ngoài hệ thống.....</i>	<i>64</i>
<i>Hình 3.7</i>	<i>Đèn quạt phía trong hệ thống.....</i>	<i>65</i>

---

<i>Hình 3.8</i>	<i>Trạng thái sevor khi hệ thống nhận đúng mật khẩu .....</i>	<i>66</i>
<i>Hình 3.9</i>	<i>Trạng thái độ ẩm nhiệt độ.....</i>	<i>68</i>
<i>Hình 3.10</i>	<i>LCD báo hệ thống không phát hiện mưa.....</i>	<i>69</i>
<i>Hình 3.11</i>	<i>LCD báo hệ thống không rò rỉ khí gas .....</i>	<i>69</i>
<i>Hình 3.12</i>	<i>LCD thông báo có mưa.....</i>	<i>69</i>
<i>Hình 3.13</i>	<i>LCD hiển thị hệ thống rò rỉ khí gas .....</i>	<i>70</i>
<i>Hình 3.14</i>	<i>Cảnh báo mưa và khí ga trên app điện thoại .....</i>	<i>70</i>
<i>Hình 3.15</i>	<i>Cảnh báo mưa và khí ga trên web .....</i>	<i>71</i>
<i>Hình 3.16</i>	<i>Giao diện web điều khiển các thiết bị.....</i>	<i>71</i>
<i>Hình 3.17</i>	<i>Giao diện web giám sát các cảm biến .....</i>	<i>72</i>
<i>Hình 3.18</i>	<i>Giao diện giám sát và điều khiển các thiết bị qua app.....</i>	<i>73</i>

## LỜI MỞ ĐẦU

Thế kỷ 21 là thời đại của công nghệ, khi những thiết bị thông minh xuất hiện và lan tỏa khắp nơi. Smart phone, Smart Tivi không còn là xa lạ với cuộc sống hiện đại của con người. Không chỉ thỏa mãn những nhu cầu cơ bản, những thiết bị smart còn mang đến những tiện ích vượt trội, khi chúng có thể tự động hoặc từ xa điều khiển các chức năng khác một cách chính xác, thuận tiện và hiệu quả.

Với sự tiên bộ của công nghệ thông tin và viễn thông, việc sử dụng nhà thông minh trở thành một phong cách sống không thể thiếu trong thời đại hiện đại. Nhà thông minh là một hệ thống liên kết nhiều thiết bị điện tử thông qua mạng internet hoặc di động, cho phép người dùng có thể tự động hoặc từ xa điều khiển các thiết bị trong nhà theo ý muốn của mình. Thông qua việc này, người dùng có thể tận dụng nhiều lợi ích, chẳng hạn như tiết kiệm năng lượng, tăng cường an ninh, cải thiện chất lượng cuộc sống.

Tuy nhiên, để thực hiện ý tưởng nhà thông minh, có rất nhiều phương pháp khác nhau, phụ thuộc vào nền tảng công nghệ mà người dùng sử dụng. Trong đồ án này tôi sẽ giới thiệu một giải pháp điều khiển và giám sát ngôi nhà thông minh qua internet và Smartphone. Giải pháp này có tính khả thi cao với cơ chế hoạt động chắc chắn và ổn định để tạo nền tảng cho một hệ thống thông minh áp dụng vào thực tế. Dữ liệu và các thông số thiết bị của nhà thông minh được gửi lên website và phần mềm giao diện để bảo đảm cơ chế an ninh, an toàn và tự động. Đồ án của tôi có đề tài là "THIẾT KẾ, THI CÔNG MÔ HÌNH NHÀ THÔNG MINH ". Mục tiêu của đồ án là thiết kế và chế tạo một mô hình nhà thông minh với các chức năng cơ bản như bật tắt đèn, quạt, cửa sổ, còi báo động; điều khiển từ xa thông qua ứng dụng trên điện thoại hoặc máy tính; theo dõi các thông số môi trường như nhiệt độ, độ ẩm,... Đồ án của tôi là một giải pháp đơn giản và hiệu quả cho việc

xây dựng nhà thông minh với chi phí thấp và khả năng linh hoạt cao.

Em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong khoa đã dạy dỗ và hướng dẫn em để em có thể thực hiện được đề tài nghiên cứu này. Đặc biệt em xin gửi lời cảm ơn chân thành đến giáo viên hướng dẫn ThS. Lưu Hoàng đã tận tâm hỗ trợ, đóng góp ý kiến cho em. Do thời gian và kiến thức có giới hạn, đề tài của em chắc chắn sẽ còn nhiều thiếu sót, em rất mong nhận được sự góp ý, chỉnh sửa của thầy cô để có thể hoàn thiện hơn.

## CHƯƠNG I: TỔNG QUAN

### 1.1 Đặt vấn đề

Trong những năm gần đây cùng với sự phát triển của nền kinh tế, thu nhập đời sống của người dân được nâng lên cùng với đó là sự gia tăng chóng mặt của số lượng nhà cửa, chung cư, hay vô số các kiểu căn hộ. Trong đó vượt trội lên là các kiểu hệ thống nhà thông minh có tích hợp nhiều chức năng khác.

Trong khi đó, so sánh với một ngôi nhà thông minh, một ngôi nhà không tích hợp hệ thống thông minh có thể có những khuyết điểm. Đầu tiên, việc kiểm soát các thiết bị điện trong nhà cần phải được thực hiện thủ công bằng tay hoặc phải di chuyển đến từng thiết bị để điều khiển, làm tốn thời gian và công sức. Bên cạnh đó, các thiết bị điện trong nhà không được kiểm soát theo cách thông minh có thể tiêu thụ nhiều năng lượng hơn so với khi được lập trình để hoạt động theo các thời gian khác nhau trong ngày.

Ngoài ra, việc không có hệ thống giám sát và báo động khi xảy ra sự cố như cháy nổ, đột nhập hoặc khói độc có thể gây nguy hiểm cho người sử dụng. Hơn nữa, việc kiểm soát các thiết bị giải trí như âm thanh, ánh sáng và video không được tiện lợi và dễ dàng như trong một ngôi nhà thông minh. Cuối cùng, việc không có hệ thống thiết kế nhà thông minh có thể làm cho không gian sống trở nên khó nhìn và thiếu thẩm mỹ.



**Hình 1.1 Mô hình tổng quát nhà thông minh.**

Tóm lại, một ngôi nhà không tích hợp hệ thống thông minh có thể không tiện lợi, tiêu thụ năng lượng cao, không an toàn và không có tính giải trí và thẩm mỹ, trong khi một ngôi nhà thông minh lại cung cấp nhiều tiện ích và tiết kiệm<sup>2</sup>

## 1.2 Mục tiêu đề tài

- Nghiên cứu kết nối thu thập thông tin dữ liệu mưa, ga, nhiệt độ,... từ các cảm biến.
- Nghiên cứu màn hình led LCD và phương thức hiển thị thông tin.

- Nghiên cứu dựa trên kiến thức đã học như: Kỹ thuật số, kỹ thuật cảm biến, linh kiện điện tử....
- Tự học thêm các kiến thức về lập trình Arduino Uno R3, ESP32, ESP8266...
- Xây dựng mô hình hoàn chỉnh để hiển thị các thông tin từ cảm biến, điều khiển các thiết bị trong hệ thống

### **1.3 Nội dung nghiên cứu**

- Tìm hiểu công dụng của từng thiết bị điện, điện tử cụ thể là: Màn hình hiển thị LCD, động cơ servo, cảm biến, bàn phím 4x3, dây nối và Arduino Uno R3, ESP32, ESP8266...
- Tìm hiểu phần mềm để lập trình cho Arduino Uno R3, ESP32, ESP8266...: Ngôn ngữ Arduino, Arduino IDE
- Đưa ra các phương án nghiên cứu.
- Kiểm tra, đánh giá tính ứng dụng của đề tài.

### **1.4 Đối tượng, phạm vi nghiên cứu và phương pháp nghiên cứu**

- Đối tượng nghiên cứu:
  - + Về phần cứng: Cảm biến ( mưa, gas, nhiệt độ, độ ẩm,...), màn hình LCD 20x4, mạch giao tiếp LCD2004 sang I2C, động cơ servo SG90, led, quạt, relay, bàn phím 4\*3, dây nối và Arduino Uno R3, ESP32, ESP8266....
  - + Về phần mềm: Arduino IDE
- Phạm vi nghiên cứu: Thiết kế mạch liên kết các cảm biến thu thập dữ liệu đưa lên Arduino, ESP32, ESP8266..., từ đó sẽ điều khiển các thiết bị và hiển thị dữ liệu lên LCD cũng như hệ thống web, app.
- Cách tiếp cận và phương pháp nghiên cứu: Trong quá trình thực hiện, em đã nghiên cứu, tìm hiểu qua sách, báo, các tài liệu trên internet, nghiên cứu cơ sở lý thuyết. Ứng dụng những kiến thức đã học ở trường cộng thêm

học ở mạng như: tập lập trình Arduino Uno R3, liên kết input/output.... Từ cơ sở đó có thể xây dựng được mô hình này qua 3 bước chính. Thiết kế mạch điều khiển, tiến hành lắp ráp mạch và hoàn thiện mô hình.

## **1.5 Ý nghĩa khoa học và thực tiễn của đề tài**

### **❖ Ý nghĩa khoa học:**

Trước hết, đây là một đồ án liên quan đến lĩnh vực Công nghệ thông tin - Điện tử, vì vậy nó có thể áp dụng nhiều công nghệ tiên tiến và thông minh, cũng như các kỹ thuật thiết kế, lập trình, điện tử, điều khiển và các hệ thống nhúng.

Thứ hai, đồ án này mang tính ứng dụng cao, nó hướng tới việc tạo ra một mô hình nhà thông minh thực tế, cho phép các thiết bị điện tử trong nhà được kiểm soát và hoạt động thông minh, đồng thời giảm thiểu chi phí sử dụng năng lượng, tăng cường tính an toàn và thuận tiện cho người sử dụng.

Thứ ba, đồ án này có thể cung cấp cho các nhà thiết kế và các nhà sản xuất các thiết bị điện tử thông minh những gợi ý và kinh nghiệm để phát triển và cải tiến sản phẩm của mình, từ đó cải thiện chất lượng cuộc sống của con người.

Cuối cùng, đồ án này có thể đóng góp vào việc phát triển xã hội thông minh và bền vững, đặc biệt là trong bối cảnh môi trường hiện nay đang phải đối mặt với những thách thức về tài nguyên và môi trường, đồng thời hướng tới mục tiêu giảm thiểu tác động của con người lên môi trường tự nhiên.

### **❖ Thực tiễn của đề tài:**



Trong thời đại hiện nay, công nghệ thông minh đang ngày càng được ứng dụng rộng rãi trong đời sống, và nhà thông minh là một trong những ứng dụng tiêu biểu của công nghệ này. Việc xây dựng một mô hình nhà thông minh có thể giúp cho cuộc sống của con người trở nên tiện lợi và an toàn hơn.

Thực tế, hiện nay trên thế giới đã có rất nhiều hãng sản xuất thiết bị nhà thông minh, và số lượng người sử dụng cũng ngày càng tăng lên. Việc xây dựng một mô hình nhà thông minh sẽ giúp người sử dụng hiểu rõ hơn về cách thức hoạt động và cách điều khiển các thiết bị điện tử trong nhà thông minh, đồng thời giúp họ tiết kiệm được chi phí sử dụng năng lượng.

Đồng thời, việc xây dựng mô hình nhà thông minh cũng giúp người ta hiểu rõ hơn về các công nghệ mới, đặc biệt là công nghệ điện tử và công nghệ thông tin, từ đó giúp họ có thể áp dụng và phát triển các sản phẩm công nghệ thông minh khác.

Vì vậy, đề tài “Thiết kế, Thi công mô hình nhà thông minh” là một đề tài rất cần thiết và có tính thực tiễn cao, nó đóng góp vào việc nghiên cứu và ứng dụng công nghệ thông minh, từ đó giúp cho cuộc sống của con người trở nên tiện lợi hơn và bền vững hơn.

## **1.6      Bố cục**

### **- Chương I: Tổng quan**

Trình bày, đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đề án.

### **- Chương II: Cơ sở lý thuyết**

Trình bày các lý thuyết liên quan đến vấn đề mà đề tài sẽ dùng để thực hiện thiết kế cho đề tài.

- Chương III: Nội dung đề tài

Trình bày tổng quan các yêu cầu của đề tài về thiết kế. Trình bày kết quả thi công phần cứng và kết quả hình ảnh trên màn hình hay mô phỏng tín hiệu, kết quả thống kê.

- Chương IV: Kết luận và phương hướng phát triển

Trình bày kết quả đạt được so với mục tiêu đề ra ban đầu, nhận xét và đánh giá kết quả đạt được của đề tài nghiên cứu. Hướng phát triển của đề tài sau này trong quá trình nghiên cứu.

- Tài liệu tham khảo

- Phụ lục

## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### 2.1 GIỚI THIỆU VỀ PHẦN CỨNG

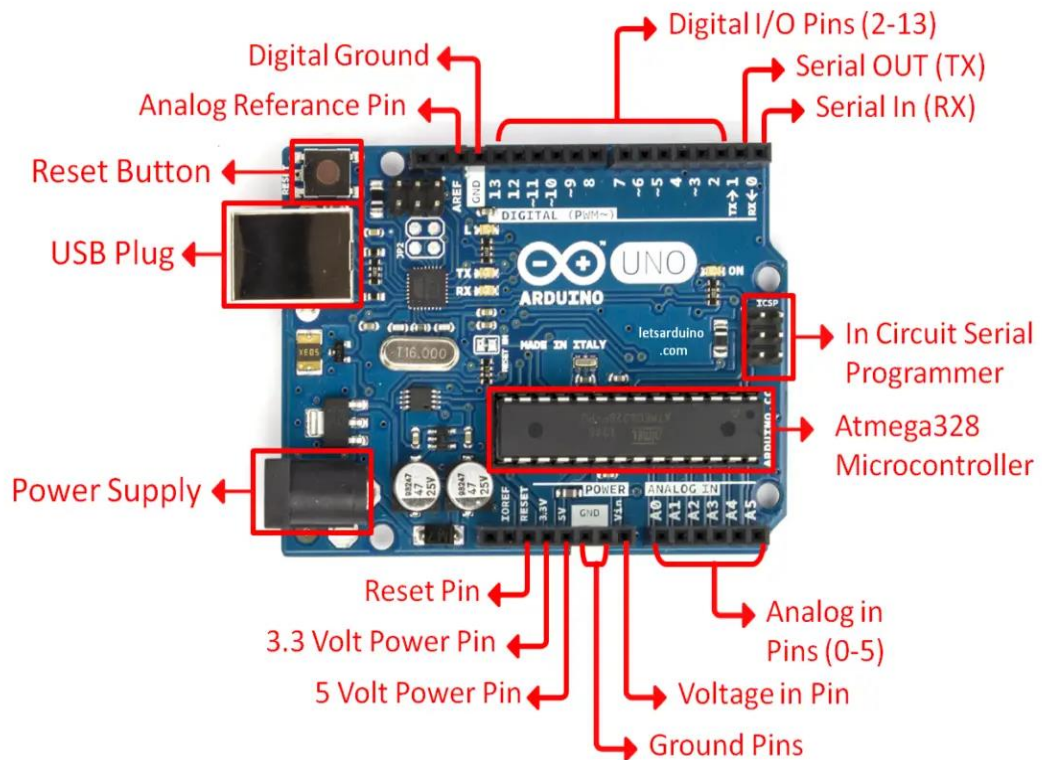
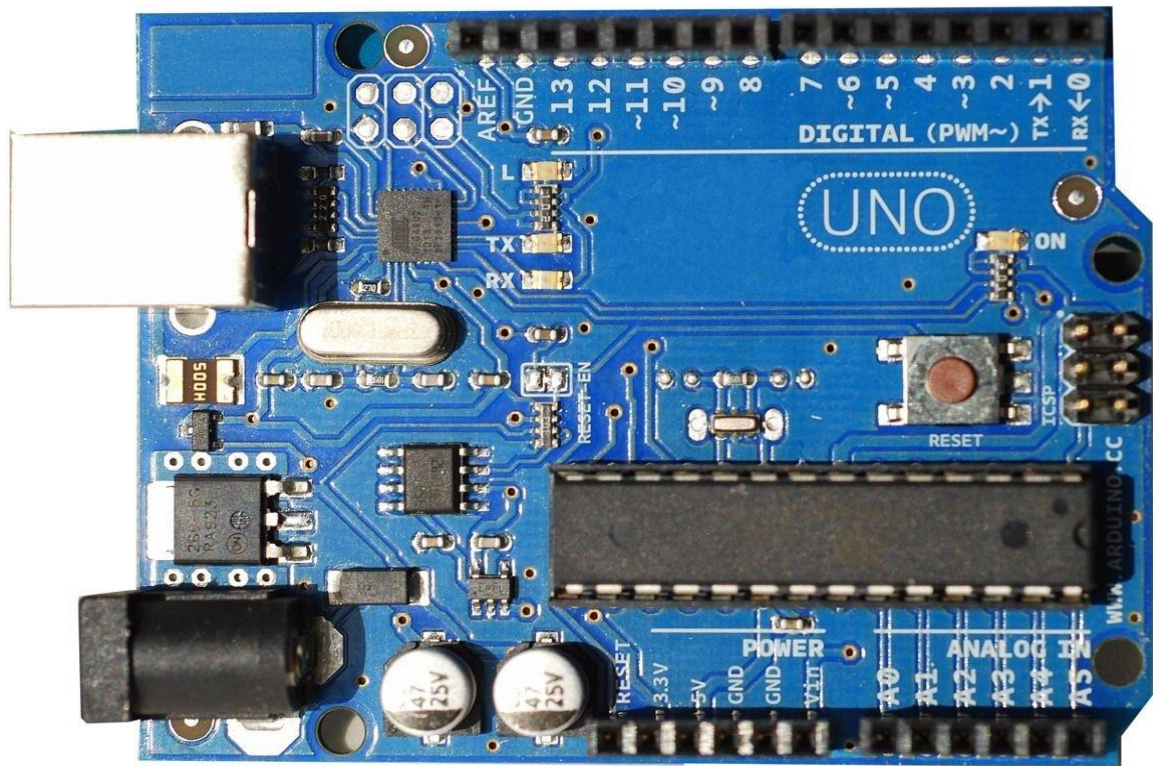
#### 2.1.1 Arduino Uno R3

Arduino Uno R3 là một board phát triển nhỏ gọn được sử dụng phổ biến trong lĩnh vực điện tử và lập trình. Nó được phát triển bởi hãng Arduino LLC và sử dụng chip ATmega328P.

Arduino Uno R3 có các tính năng như: 14 chân input/output kỹ thuật số, trong đó 6 chân có thể được sử dụng như chân PWM; 6 chân input analog; cổng kết nối USB cho phép lập trình và nạp chương trình từ máy tính; đèn LED báo tín hiệu; nút nhấn để reset board.

Arduino Uno R3 được sử dụng rộng rãi trong các dự án điện tử và lập trình, từ các dự án đơn giản như tạo đèn LED nhấp nháy đến các dự án phức tạp hơn như tạo robot hoặc các hệ thống tự động hóa. Arduino Uno R3 cũng được sử dụng trong giáo dục và đào tạo để giúp sinh viên và học sinh hiểu rõ hơn về lĩnh vực điện tử và lập trình.

Với việc sử dụng một môi trường lập trình dễ sử dụng và miễn phí như Arduino IDE, việc lập trình cho Arduino Uno R3 trở nên đơn giản hơn bao giờ hết. Do đó, Arduino Uno R3 là một board phát triển được yêu thích và sử dụng rộng rãi trong cộng đồng lập trình viên và những người đam mê điện tử.



Hình 2.1 Sơ đồ Arduino Uno R3

❖ **Thông số kỹ thuật:**

Vi điều khiển	Atmega328P
Điện áp hoạt động	5V
Điện áp cấp (hoạt động tốt)	7 – 12 V
Điện áp cấp (giới hạn)	6 – 12 V
Chân I/O digital	14 ( có 6 chân xuất xung PWM)
Chân Input analog	6 (A0 – A5)
Dòng điện mỗi chân I/O	20 mA
Dòng điện chân 3.3V	50 mA
Bộ nhớ Flash	32 kB (Atmega328P) – trong đó 0.5 kB dùng cho bootloader.
SRAM	2 kB (Atmega328P)
EEPROM	1 kB (Atmega328P)
Tốc độ xung nhịp	16 MHz
Kích thước	68.6 x 53.4 mm
Trọng lượng	25 g

**Chức năng của các chân trên board Arduino Uno R3:**

Chân 0 (RX): Nhận dữ liệu từ kết nối ngoài vào UART

Chân 1 (TX): Gửi dữ liệu ra ngoài UART

Chân 2-13: Điều khiển các thiết bị như LED, còi, motor, cảm biến, ...

Chân 3, 5, 6, 9, 10, 11: Có thể sử dụng chế độ PWM để tạo tín hiệu xung với tần số và mức độ độ rộng khác nhau

Chân A0-A5: Chân vào analog để đọc giá trị tín hiệu analog từ các sensor hoặc thiết bị đo

Chân Vin: Điện áp đầu vào từ nguồn ngoài

Chân 5V: Điện áp ra 5V

Chân 3.3V: Điện áp ra 3.3V

Chân GND: Đất (0V) cho mạch điện tử.

### 2.1.2 Màn hình LCD 20x4

Màn hình LCD (Liquid Crystal Display) là một loại màn hình hiển thị được sử dụng phổ biến trong các thiết bị điện tử, như máy tính, điện thoại, máy ảnh, máy chơi game và các hệ thống nhúng.

Màn hình LCD được tạo ra bằng cách sử dụng các lớp vật liệu dẫn điện và dẫn ánh sáng, gọi là lớp tinh thể lỏng, được chứa trong hai lớp kính. Khi một tín hiệu điện được áp dụng vào một lớp tinh thể lỏng, nó thay đổi hướng của ánh sáng khi đi qua lớp đó, tạo ra hình ảnh trên màn hình.

Màn hình LCD thường được sử dụng để hiển thị các thông tin văn bản và hình ảnh đơn giản. Một số màn hình LCD cũng được thiết kế để hiển thị đồ họa và video.



**Hình 2.2 Màn hình LCD**

Trong các dự án điện tử và lập trình, màn hình LCD thường được sử dụng để hiển thị các thông tin về trạng thái của hệ thống, các giá trị đo lường từ các cảm biến, và các thông báo và lệnh điều khiển từ người sử dụng. Màn hình LCD có nhiều loại và kích thước khác nhau để phù hợp với các nhu cầu của các dự án khác nhau.

Tuy nhiên, điểm yếu của màn hình LCD là khả năng hiển thị màu sắc và độ phân giải thấp hơn so với các loại màn hình khác như OLED hay LED, và màn hình LCD không thể được sử dụng trong các ứng dụng yêu cầu tốc độ và độ phân giải cao.

Thông số kỹ thuật của LCD 20x4:

- Điện áp: 5V
- Ngõ giao tiếp: 16 chân
- Màu sắc: xanh lá hoặc xanh dương
- Module hỗ trợ giao tiếp với vi điều khiển: LCD I2C

### **2.1.3 Mạch giao tiếp LCD I2C**

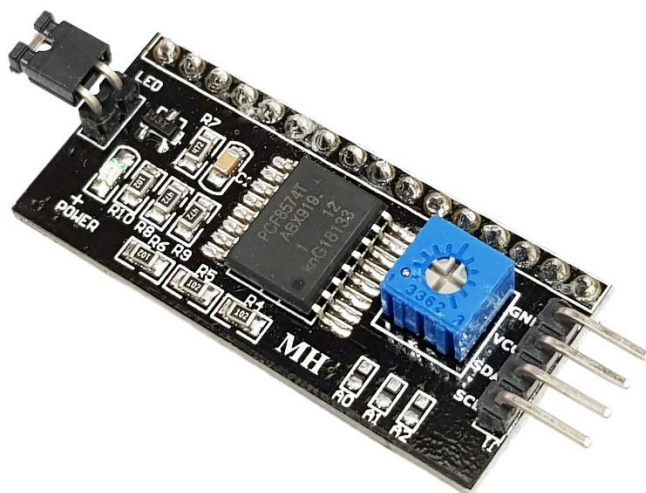
Mạch giao tiếp LCD I2C (Inter-Integrated Circuit) là một mạch mở rộng được sử dụng để kết nối màn hình LCD với vi điều khiển Arduino thông qua giao tiếp I2C. Giao tiếp I2C là một giao thức truyền thông hai chiều dựa trên đường dẫn hai dây (SCL và SDA) để truyền tải dữ liệu giữa các thiết bị điện tử.

Mạch giao tiếp LCD I2C giúp giảm số lượng chân kết nối giữa màn hình LCD và vi điều khiển Arduino, giúp cho việc kết nối trở nên đơn giản hơn và giảm tối đa các lỗi kết nối. Mạch giao tiếp LCD I2C cũng cung cấp các điều khiển đơn giản để hiển thị dữ liệu lên màn hình LCD, bao gồm điều khiển hiển thị, điều



khiển con trỏ, điều khiển cài đặt và cấu hình các tham số khác như độ sáng, độ tương phản và độ phân giải.

Sử dụng mạch giao tiếp LCD I2C cũng giúp tối ưu hóa tài nguyên của vi điều khiển Arduino, giảm thiểu việc sử dụng nhiều chân kết nối và giảm độ trễ trong quá trình truyền tải dữ liệu. Do đó, mạch giao tiếp LCD I2C là một lựa chọn phổ biến cho các dự án điện tử và lập trình sử dụng màn hình LCD.



**Hình 2.3 Mạch I2C**

#### **2.1.4 Cảm biến mưa**

Cảm biến mưa là một loại cảm biến dùng để phát hiện và đo lượng mưa rơi xuống. Cảm biến mưa thường sử dụng công nghệ dựa trên nguyên lý đo điện trở hoặc đo tần số.

Cảm biến mưa bao gồm một bộ chuyển đổi tín hiệu để chuyển đổi tín hiệu từ cảm biến thành tín hiệu điện thế hoặc tần số, từ đó đo được lượng mưa rơi xuống. Các loại cảm biến mưa thông thường được thiết kế với đầu dò cảm ứng mưa, bao gồm các mạch điện tử và cảm biến lực.



Cảm biến mưa được sử dụng rộng rãi trong các ứng dụng liên quan đến môi trường, đặc biệt là trong các hệ thống quan trắc thời tiết, thu thập dữ liệu và dự báo thời tiết. Cảm biến mưa cũng được sử dụng trong các hệ thống tưới cây tự động hoặc hệ thống kiểm soát và phát triển nông nghiệp hiện đại, giúp cho việc quản lý và sử dụng tài nguyên nước trở nên hiệu quả hơn.

Trong các dự án điện tử và lập trình, cảm biến mưa được sử dụng để đo lượng mưa và điều khiển các thiết bị khác như đèn chiếu sáng, bơm tưới cây, và các thiết bị khác phụ thuộc vào thời tiết.



**Hình 2.4 Cảm biến mưa**

**Thông số kỹ thuật:**

- Điện áp sử dụng: 5V
- Có đèn báo hiệu nguồn và đầu ra: Led báo nguồn ( Màu xanh), Led cảnh báo mưa ( Màu đỏ). LED sáng lên khi không có mưa đầu ra cao, có mưa, đầu ra thấp LED tắt.
- Hoạt động dựa trên nguyên lý: Nước rơi vào board sẽ tạo ra môi trường dẫn điện.

- Tín hiệu đầu ra: Digital TTL (0VDC / 5VDC) và đầu ra Analog A0 trả giá trị điện áp tuyến tính theo lượng nước tiếp xúc với cảm biến. (Có thể sử dụng trực tiếp Relay, Còi công suất nhỏ...)
- Độ nhạy có thể được điều chỉnh thông qua chiết áp (biến trở)

### 2.1.5 Động cơ servo SG90

Động cơ servo SG90 là một loại động cơ servo mini, được sử dụng rộng rãi trong các ứng dụng điện tử và robot. Động cơ servo SG90 có khối lượng nhẹ, kích thước nhỏ gọn và chi phí thấp, là lựa chọn phổ biến cho các dự án điện tử và robot DIY.

Động cơ servo SG90 được điều khiển bằng tín hiệu điều khiển PWM (Pulse Width Modulation), tín hiệu này sẽ điều chỉnh động cơ servo SG90 quay về vị trí mong muốn. Điều này làm cho động cơ servo SG90 được sử dụng rộng rãi trong các ứng dụng cần kiểm soát vị trí và góc quay như robot, máy in 3D, máy bay điều khiển từ xa, các thiết bị tăng cường thực tế ảo...



**Hình 2.5 Động cơ servo**

Động cơ servo SG90 thường được sử dụng để điều khiển chuyển động của các bộ phận trong robot như động cơ, cánh tay, chân, khớp nối và các thiết bị khác. Nó cũng được sử dụng để điều khiển các thiết bị tại vị trí cụ thể hoặc góc quay cụ thể.

Động cơ servo SG90 có độ chính xác cao, tốc độ quay nhanh và tải trọng đáp ứng tốt. Nó có thể hoạt động ở điện áp thấp, từ 4,8V đến 6V, điều này khiến cho nó phù hợp với nhiều ứng dụng khác nhau và dễ dàng tích hợp với các loại vi điều khiển khác nhau.

**Thông số kỹ thuật:**

- Điện áp hoạt động: 4.8-5VDC
- Tốc độ: 0.12 sec/ 60 deg (4.8VDC)
- Lực kéo: 1.6 Kg.cm
- Kích thước: 21x12x22mm
- Trọng lượng: 9g.

**2.1.6 Cảm biến khí gas MQ2**

Cảm biến khí gas MQ-2 là một loại cảm biến khí được sử dụng để phát hiện sự hiện diện của các khí như khí CO, khí LPG, khí butan, khí propane, khí metan, khí hydro, và các khí khác.

Cảm biến MQ-2 hoạt động bằng cách sử dụng nguyên lý tương tác giữa các khí và các chất hoạt động trên bề mặt của các tín hiệu điện trở. Khi khí được phát hiện, nó sẽ tương tác với các chất hoạt động trên bề mặt của cảm biến và thay đổi trở kháng của nó. Sau đó, một tín hiệu điện sẽ được tạo ra và điều khiển bởi

một bộ vi xử lý để phân tích dữ liệu và cảnh báo người dùng về mức độ nguy hiểm của khí.

Cảm biến MQ-2 thường được sử dụng trong các hệ thống báo động khí, hệ thống giám sát an ninh, hệ thống quản lý chất lượng không khí trong nhà, hệ thống cảnh báo cháy, hệ thống điều khiển đèn và các thiết bị thông minh khác.



**Hình 2.6 Cảm biến gas MQ2**

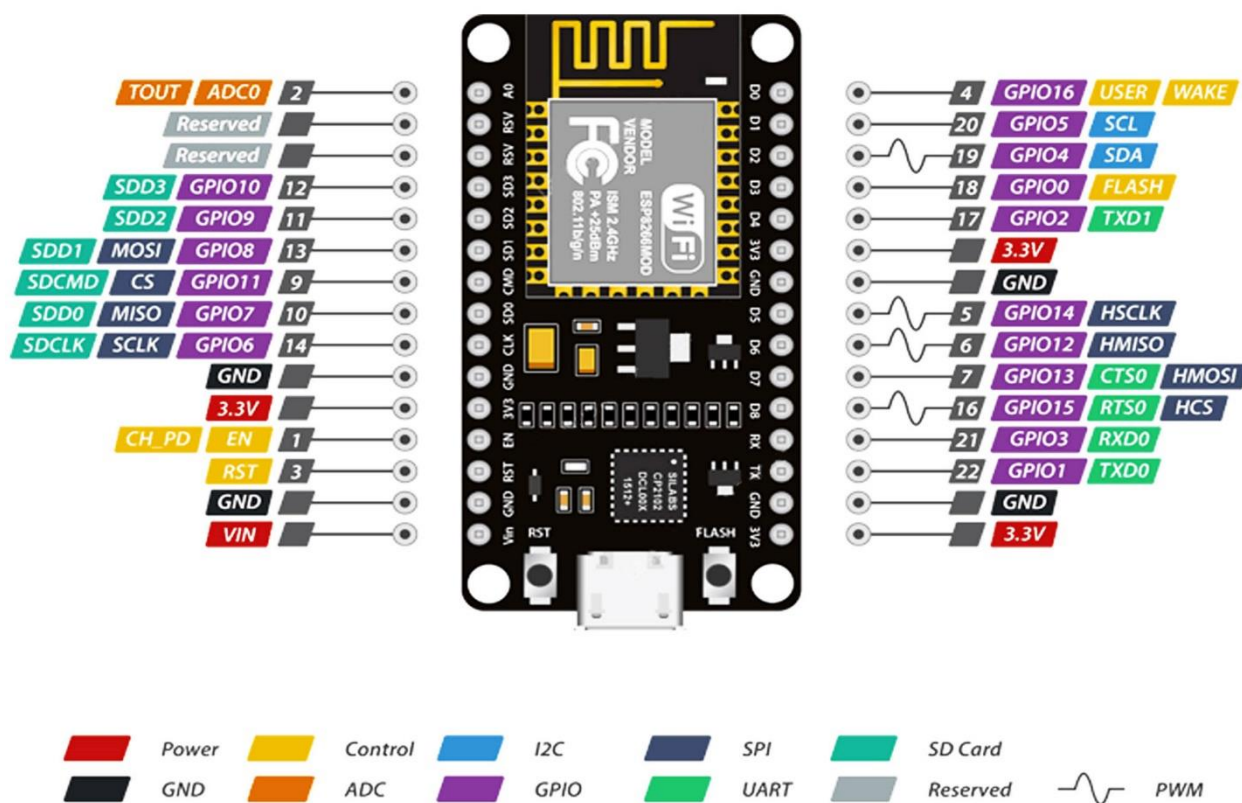
**Thông số kỹ thuật:**

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao
- Mạch đơn giản
- Ổn định khi sử dụng trong thời gian dài

### 2.1.7 Kit ESP8266

ESP8266 là một module Wi-Fi đa năng được phát triển bởi Espressif Systems. Nó cung cấp cho người dùng khả năng kết nối Internet thông qua một mạng Wi-Fi, cũng như cho phép lập trình thiết bị IoT (Internet of Things) và các ứng dụng nhà thông minh.

Module ESP8266 được tích hợp sẵn chip Wi-Fi, bộ xử lý, bộ nhớ và các thành phần khác trên một board nhỏ gọn. Nó có thể hoạt động dưới dạng một điểm truy cập Wi-Fi, một thiết bị kết nối mạng (station mode) hoặc một chế độ kết hợp của cả hai.



Hình 2.7 Sơ đồ chân của ESP8266

Với giá thành thấp và khả năng lập trình đa năng, module ESP8266 đã trở thành một lựa chọn phổ biến cho các ứng dụng IoT và nhà thông minh. Nó cũng có thể

được sử dụng để thực hiện các chức năng như kiểm soát thiết bị thông qua mạng Wi-Fi, thu thập dữ liệu từ các cảm biến, và truyền dữ liệu lên Internet.

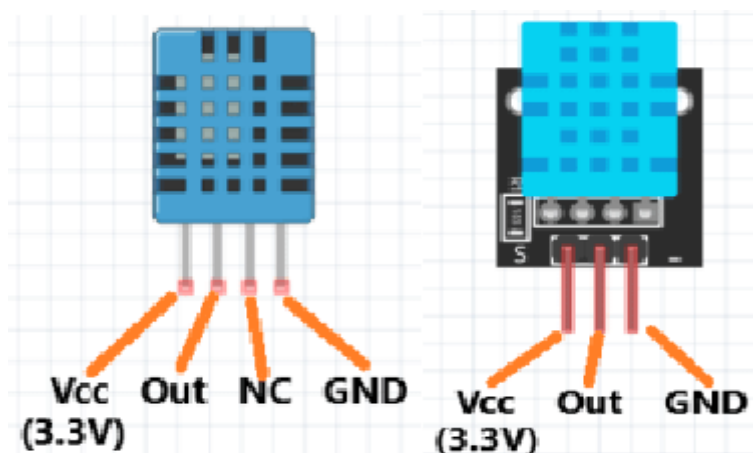
Module ESP8266 được hỗ trợ bởi một cộng đồng lớn của các nhà phát triển, đảm bảo rằng người dùng có thể tìm thấy các tài liệu hướng dẫn và các thư viện mã nguồn mở để lập trình và tùy chỉnh module của mình một cách dễ dàng.

### 2.1.8 Cảm biến DHT11

DHT11 là một cảm biến đo độ ẩm và nhiệt độ trong không khí. Nó là một cảm biến giá rẻ và đơn giản được sử dụng rộng rãi trong các ứng dụng nhà thông minh và IoT.

Cảm biến DHT11 hoạt động bằng cách đo sự thay đổi điện trở của chất lượng không khí. Điện trở của không khí thay đổi tùy thuộc vào độ ẩm và nhiệt độ, và DHT11 sử dụng sự thay đổi này để đo độ ẩm và nhiệt độ.

DHT11 có ba chân: VCC, GND và OUT. Chân VCC được kết nối với nguồn cấp 3-5V, chân GND được kết nối với đất, và chân OUT được kết nối với một chân điều khiển để đọc giá trị đo được. Dữ liệu được truyền qua chân OUT dưới dạng tín hiệu kỹ thuật số.



Hình 2.8 Sơ đồ chân của DHT11

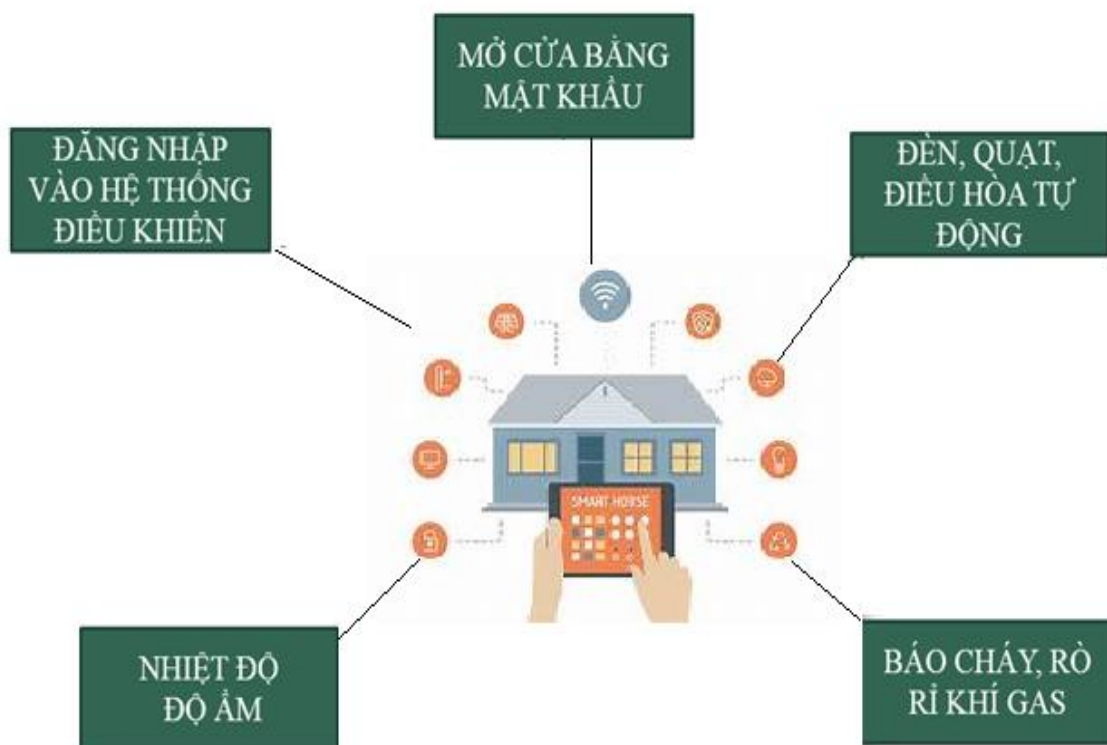
Cảm biến DHT11 có độ chính xác tương đối thấp so với các cảm biến đo độ ẩm và nhiệt độ khác. Tuy nhiên, với giá thành thấp và tính đơn giản, DHT11 vẫn được sử dụng rộng rãi trong các ứng dụng nhà thông minh và IoT, nhất là khi độ chính xác không phải là yếu tố quan trọng.

## CHƯƠNG III: NỘI DUNG ĐỀ TÀI

### 3.1 GIỚI THIỆU

Dựa trên tài liệu tham khảo và giáo trình, giảng viên hướng dẫn đã tạo cơ hội cho bản thân em nghiên cứu về đề tài. Sau một khoảng thời gian tìm hiểu, mặc dù kiến thức của bản thân còn nhiều hạn chế, em đã cố gắng tìm hiểu nhiều nhất trong khả năng của mình, cùng với đó em đã ứng dụng các kiến thức để làm ra mô hình thực tế để giúp mọi người thấy được thành quả của em. Mô hình thiết kế phải đáp ứng được các yêu cầu:

- Mở cửa bằng mật khẩu
- Hệ thống thông báo nhiệt độ, độ ẩm.
- Hệ thống báo mưa và rò rỉ khí Gas tự động cảnh báo về điện thoại.
- Hệ thống điều hòa, đèn, quạt tự động.
- Hệ thống điều khiển từ xa qua giao diện Web và app điện thoại

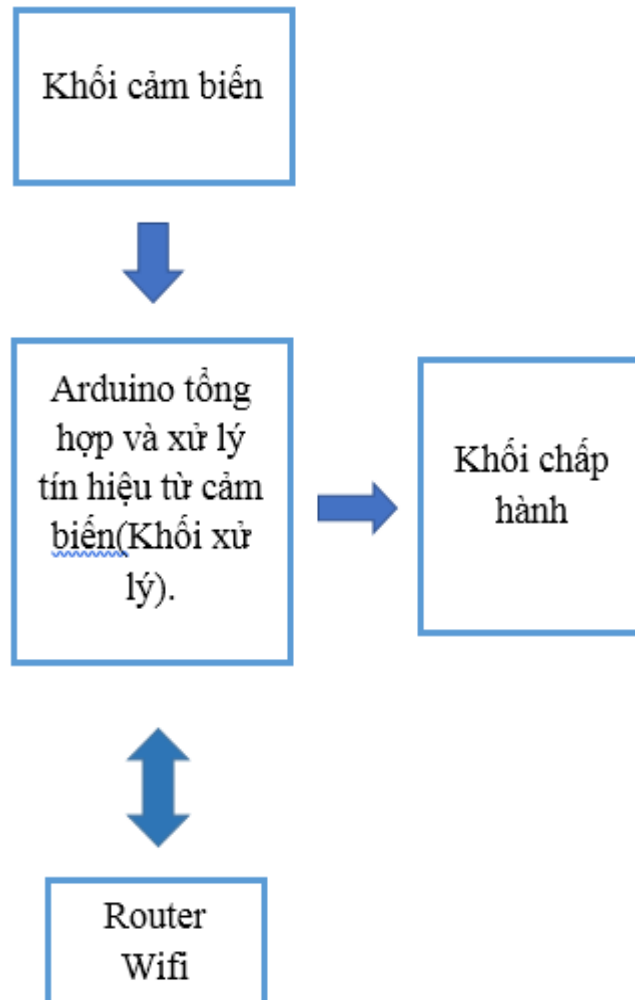


Hình 3.1 Các chức năng sử dụng



## 3.2 TÍNH TOÁN VÀ THIẾT KẾ

### 3.2.1 Sơ đồ khối



**Hình 3.2 Sơ đồ khối**

*a) Khối cảm biến:*

- Cảm biến nhiệt độ: tín hiệu đầu ra là tín hiệu tương tự.
- Cảm biến độ ẩm: tín hiệu đầu ra là tín hiệu số.
- Cảm biến mưa: tín hiệu đầu ra là tín hiệu số.
- Cảm biến khí gas: tín hiệu đầu ra là tín hiệu số.
- Cảm biến hồng ngoại: tín hiệu đầu ra là tín hiệu số

b) *bKhối xử lý:*

- Arduino Uno R3.
- ESP8266
- ESP8266Mod

c) *Router wifi*

d) *Khối chấp hành:*

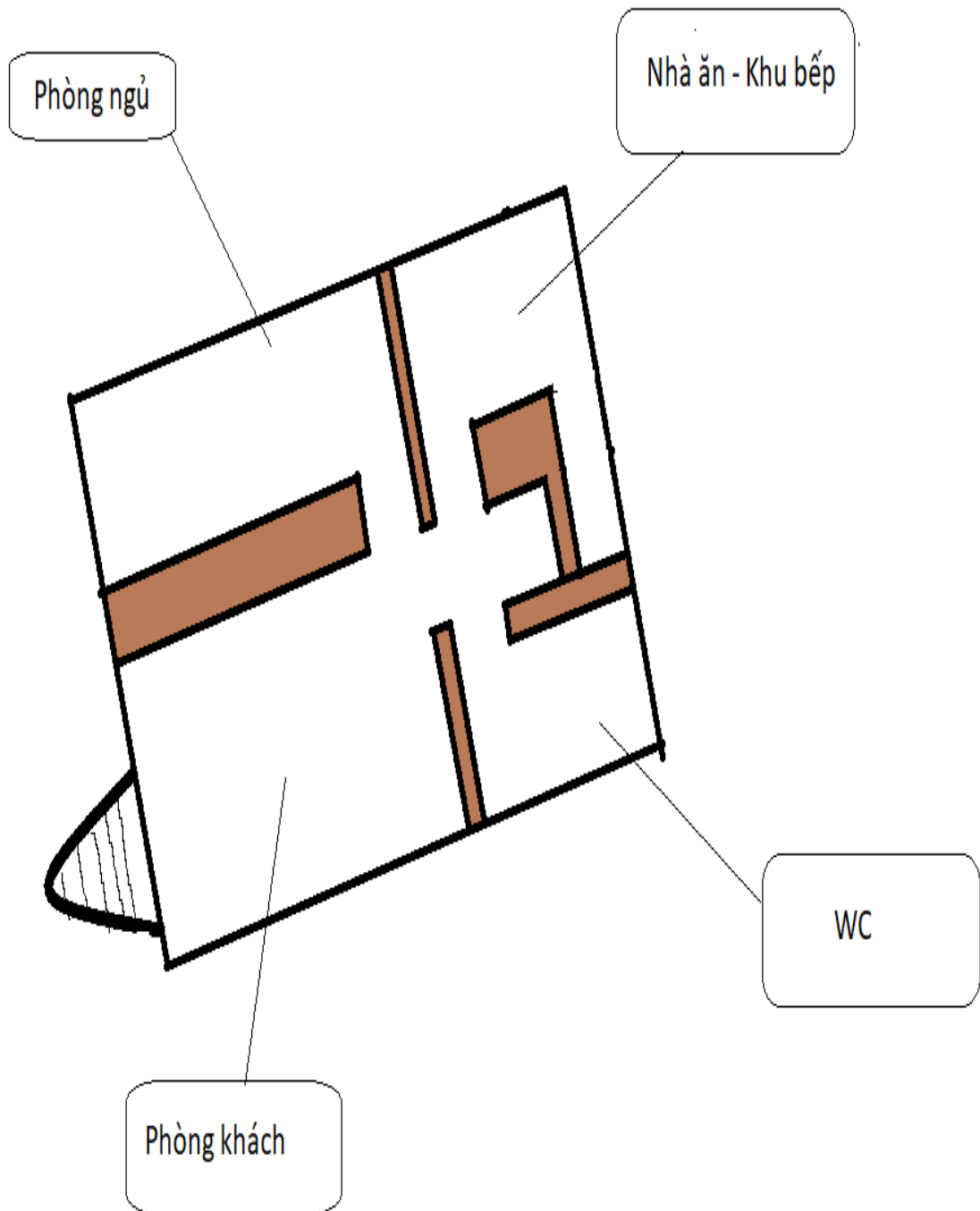
- Cửa ra vào.
- Đèn, quạt.
- Còi báo động.
- LCD.

### **3.2.2 Sơ đồ kết cấu**

Từ một ngôi nhà thông thường, chúng em lựa chọn thiết kế ra một mô hình cơ bản dành cho 1 gia đình với 2 người ở, với thiết kế gồm:

- 1 phòng khách
- 1 phòng ngủ
- 1 khu bếp - nhà ăn

- 1 nhà vệ sinh



**Hình 3.3 Sơ đồ kết cấu ngôi nhà thông minh**

### 3.2.3 Giao tiếp ESP8266 chuẩn UART

Chuẩn giao tiếp UART là viết tắt của Universal Asynchronous Receiver/Transmitter. Đây không phải là một giao thức truyền thông như SPI và I2C, mà là một mạch vật lý trong vi điều khiển hoặc một IC độc lập. Mục đích chính của UART là truyền và nhận dữ liệu nối tiếp. Một trong những điều tốt nhất về UART là nó chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị.

Trong giao tiếp UART, hai UART truyền thông trực tiếp với nhau. UART truyền tải chuyển đổi dữ liệu song song từ một thiết bị điều khiển như CPU thành dạng nối tiếp, truyền tải nó theo dạng nối tiếp đến UART nhận, sau đó chuyển đổi dữ liệu nối tiếp trở lại thành dạng song song cho thiết bị nhận. Chỉ cần hai dây để truyền dữ liệu giữa hai UART. Dữ liệu chảy từ chân Tx của UART truyền tải đến chân Rx của UART nhận.

UARTs truyền tải dữ liệu không đồng bộ, có nghĩa là không có tín hiệu đồng hồ để đồng bộ hóa đầu ra của các bit từ UART truyền tải đến việc lấy mẫu các bit bởi UART nhận. Thay vì tín hiệu đồng hồ, UART truyền tải thêm các bit bắt đầu và kết thúc vào gói dữ liệu được chuyển. Các bit này xác định phần đầu và cuối của gói dữ liệu để UART nhận biết khi nào bắt đầu đọc các bit.

#### 3.2.4 Các đặc tính kỹ thuật của chuẩn RS-232 theo tiêu chuẩn TIA/EIA-232-F như sau:

Số lượng dây sử dụng	2
Tốc độ truyền	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000
Phương pháp truyền	Bất đồng bộ

Truyền nối tiếp hay song song?	Nối tiếp
Số lượng thiết bị chủ tối đa	1
Số lượng thiết bị tớ tối đa	1

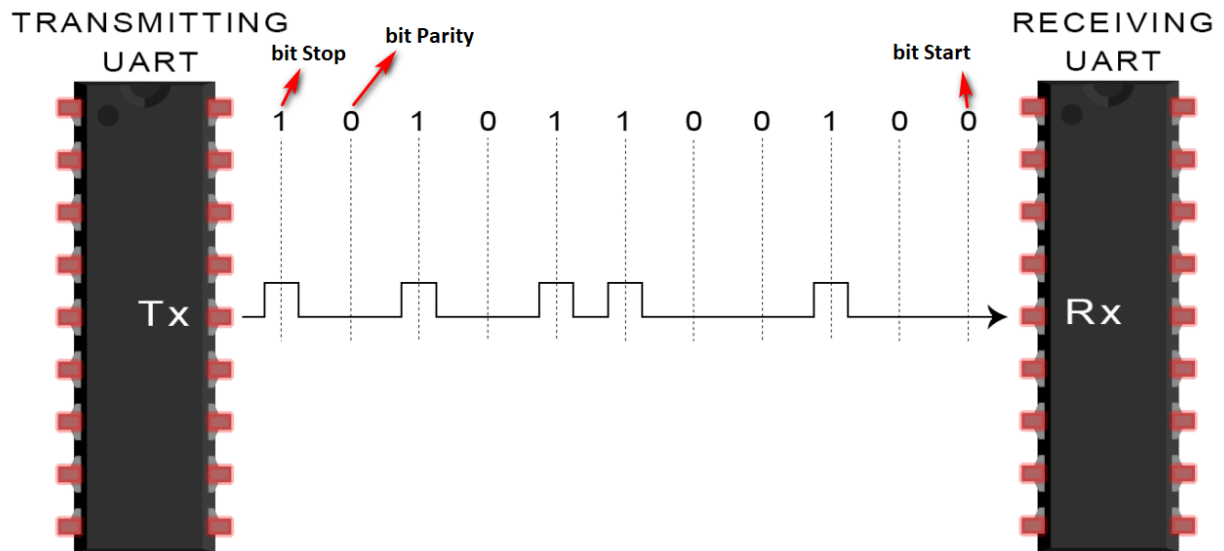
Các tốc độ truyền dữ liệu thông dụng trong cổng nối tiếp là: 9600, 19200, 38400, 7600, 115200, 230400, 460800, 921600, 1000000, 1500000

### 3.2.5 Các mức điện áp của đường truyền:

Mức điện áp của tiêu chuẩn UART có thể khác nhau tùy thuộc vào thiết bị và các yêu cầu của ứng dụng cụ thể. Tuy nhiên, trong các ứng dụng nhúng và vi xử lý, UART thường được sử dụng với mức điện áp logic 0V và 3.3V hoặc 5V.

Trong một số thiết bị, đặc biệt là các thiết bị di động và các ứng dụng tiết kiệm năng lượng, mức điện áp logic 1.8V hoặc 2.5V có thể được sử dụng. Tuy nhiên, các mức điện áp này không phổ biến như mức điện áp 3.3V và 5V.

Ngoài ra, các tiêu chuẩn UART có thể định nghĩa các yêu cầu khác về mức điện áp, chẳng hạn như độ chính xác mức điện áp, điện áp tối thiểu và tối đa, hoặc độ chênh lệch tín hiệu. Do đó, khi sử dụng UART trong ứng dụng cụ thể, cần xem xét kỹ các đặc tính kỹ thuật của tiêu chuẩn UART và các yêu cầu của thiết bị để đảm bảo độ tin cậy và đồng bộ trong truyền dữ liệu.



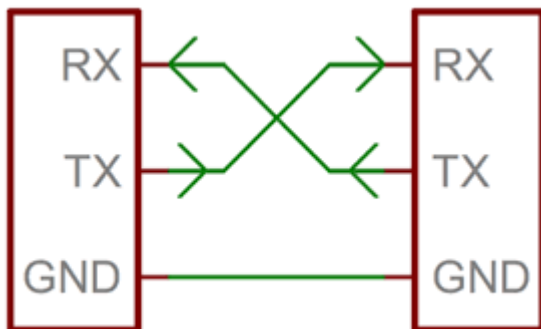
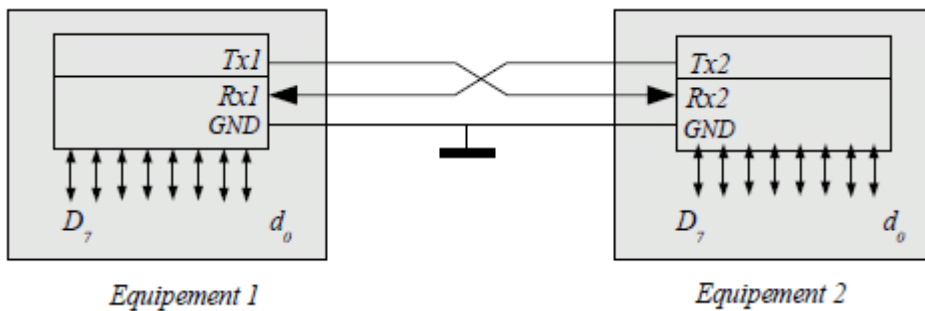
### 3.2.6 Sơ đồ chân cổng kết nối

Sơ đồ chân cổng kết nối UART thường khác nhau tùy thuộc vào loại vi xử lý hoặc thiết bị điện tử cụ thể. Tuy nhiên, các chân cơ bản trong giao diện UART bao gồm:

1. TX (Transmit): Chân truyền dữ liệu từ thiết bị gửi (VD: vi xử lý) tới thiết bị nhận (VD: module giao tiếp wifi).
2. RX (Receive): Chân nhận dữ liệu từ thiết bị nhận (VD: module giao tiếp wifi) tới thiết bị gửi (VD: vi xử lý).
3. GND (Ground): Chân đất, được sử dụng để tạo địa chỉ chung cho các tín hiệu truyền.

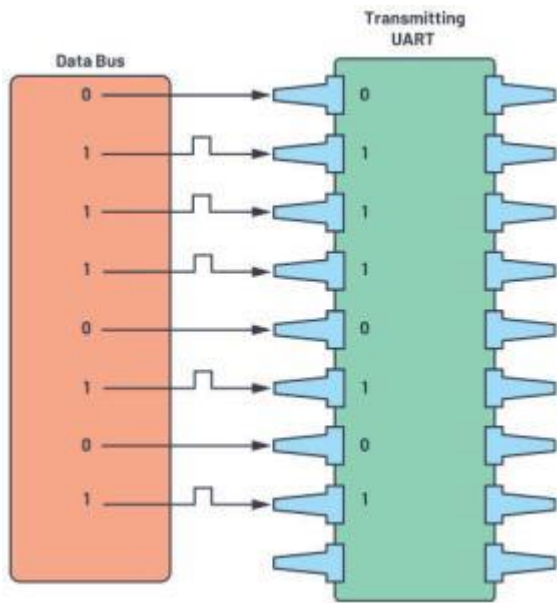
Ngoài các chân cơ bản này, một số cổng UART có thể cung cấp thêm các chức năng như:

4. RTS (Ready to Send): Chân điều khiển truyền tín hiệu từ thiết bị gửi tới thiết bị nhận.
5. CTS (Clear to Send): Chân điều khiển truyền tín hiệu từ thiết bị nhận tới thiết bị gửi.
6. DTR (Data Terminal Ready): Chân điều khiển sẽ bật khi thiết bị gửi đã sẵn sàng để truyền dữ liệu.
7. DSR (Data Set Ready): Chân điều khiển sẽ bật khi thiết bị nhận đã sẵn sàng để nhận dữ liệu.

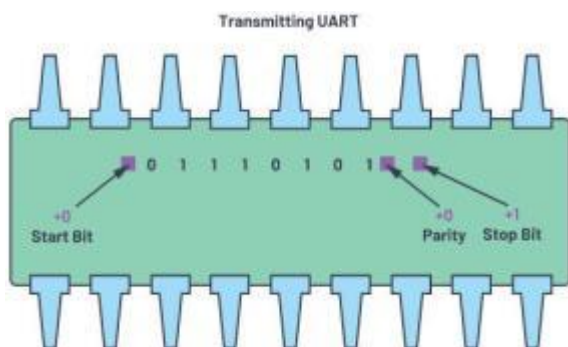


### 3.2.7 Truyền dữ liệu:

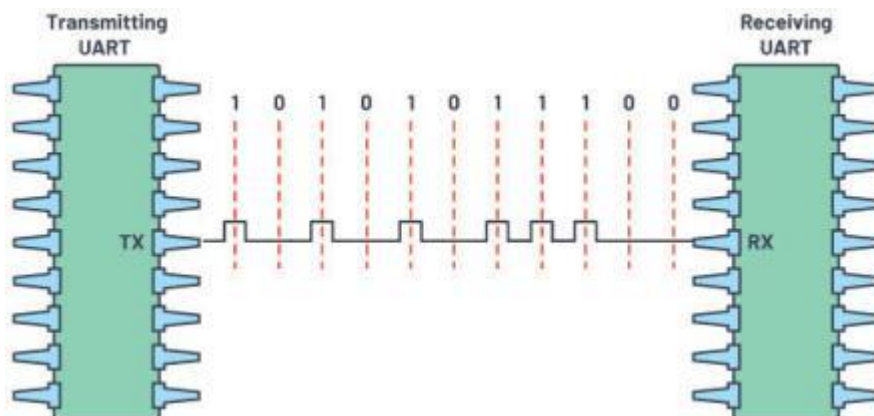
1. UART truyền nhận dữ liệu song song từ bus dữ liệu.



2. UART truyền thêm bit start, bit chẵn lẻ và bit dừng vào khung dữ liệu.

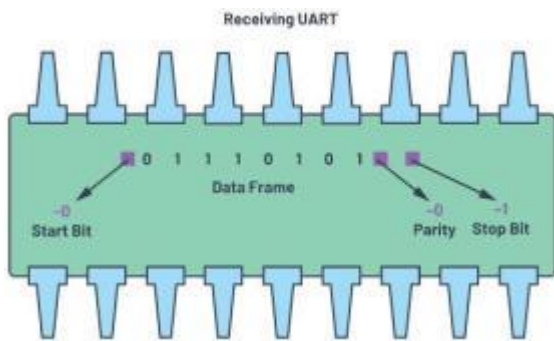


3. Toàn bộ gói được gửi nối tiếp từ UART truyền đến UART nhận. UART nhận lấy mẫu đường dữ liệu ở tốc độ truyền được định cấu hình trước.

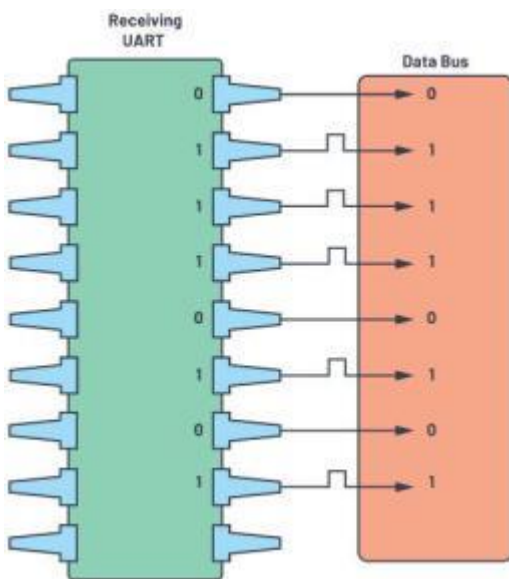




4. UART nhận loại bỏ bit start, bit chẵn lẻ và bit stop khỏi khung dữ liệu.



5. UART nhận chuyển đổi dữ liệu nối tiếp trở lại thành song song và chuyển nó đến bus dữ liệu ở đầu nhận.



### 3.2.8 Giao tiếp chuẩn I2C

I2C (Inter-Integrated Circuit) là một giao thức truyền thông nối tiếp đồng bộ, đa chủ/đa tớ (controller/target), chuyên mạch gói, đơn cuối được phát minh vào năm 1982 bởi Philips Semiconductors. Nó được sử dụng rộng rãi để kết nối các IC ngoại vi tốc độ thấp với bộ xử lý và vi điều khiển trong giao tiếp ngắn khoảng cách, trong bo mạch. Nhiều đối thủ cạnh tranh, như Siemens, NEC, Texas Instruments, STMicroelectronics, Motorola, Nordic Semiconductor và Intersil

đã giới thiệu các sản phẩm I2C tương thích với thị trường kể từ giữa những năm 1990.

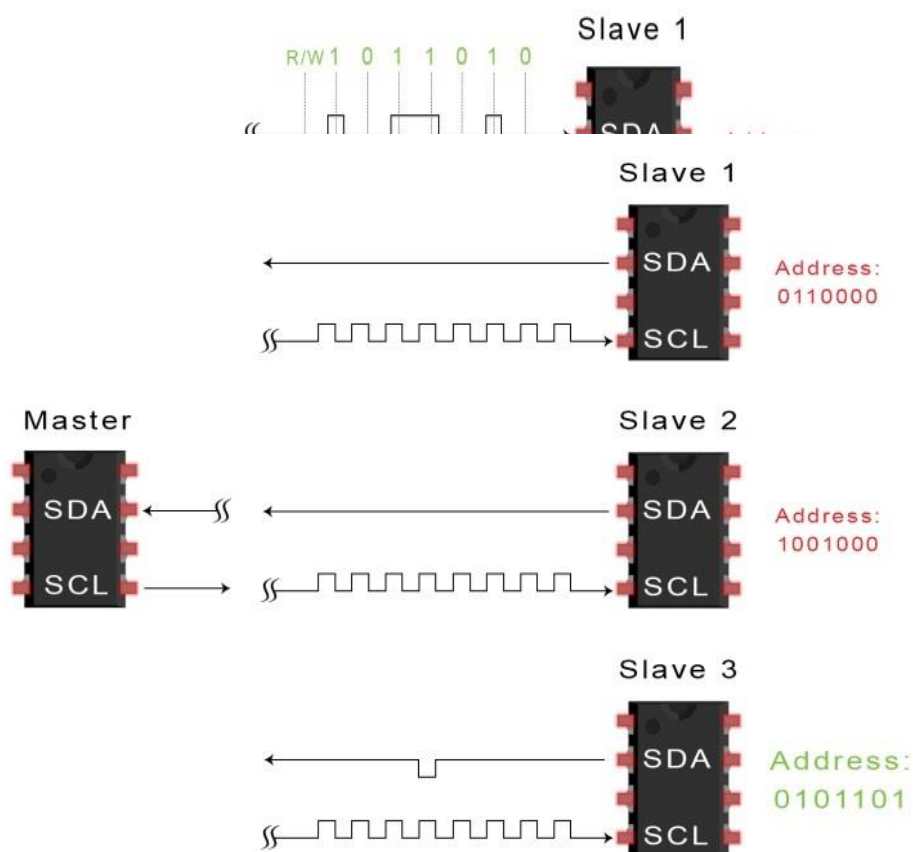
I2C kết hợp các tính năng tốt nhất của SPI và UART. Với I2C, bạn có thể kết nối nhiều thiết bị tớ với một thiết bị chủ (giống như SPI) và bạn có thể có nhiều thiết bị chủ điều khiển một hoặc nhiều thiết bị tớ. Điều này rất hữu ích khi bạn muốn có nhiều hơn một vi điều khiển ghi dữ liệu vào một thẻ nhớ duy nhất hoặc hiển thị văn bản lên một LCD duy nhất. Giống như giao tiếp UART, I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị: SDA (Serial Data) - Dây cho thiết bị chủ và thiết bị tớ để gửi và nhận dữ liệu. SCL (Serial Clock) - Dây mang tín hiệu đồng hồ.

Với I2C, dữ liệu được truyền trong các tin nhắn. Các tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của thiết bị tớ và một hoặc nhiều khung dữ liệu chứa dữ liệu được truyền. Tin nhắn cũng bao gồm các điều kiện bắt đầu và kết thúc, bit đọc/ghi và bit ACK/NACK giữa mỗi khung dữ liệu.

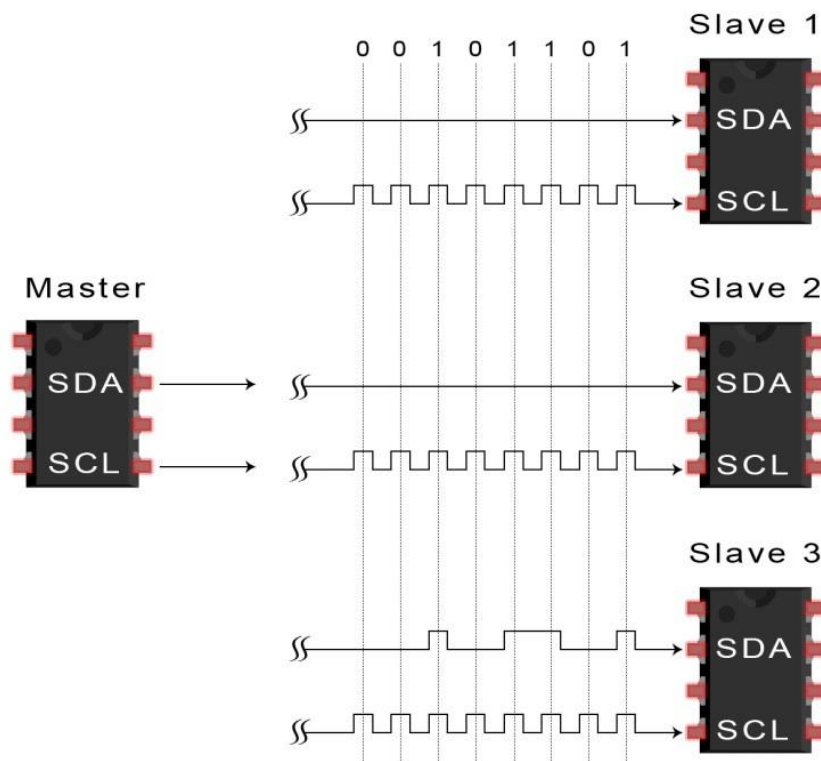
### **3.2.9 Gửi dữ liệu đến thiết bị Slave**

Trình tự hoạt động sau đây diễn ra khi một thiết bị Master gửi dữ liệu đến một thiết bị Slave cụ thể thông qua bus I2C:

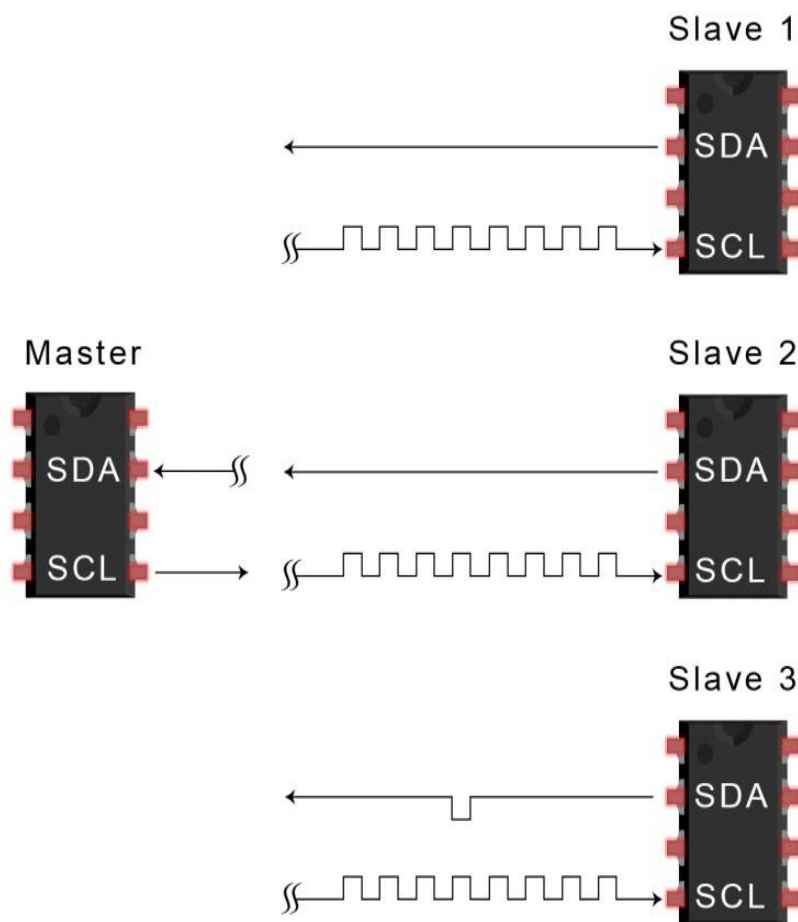
- Thiết bị Master gửi điều kiện bắt đầu đến tất cả các thiết bị Slave
- Thiết bị Master gửi 7 bit địa chỉ của thiết bị Slave mà thiết bị Master muốn giao tiếp cùng với bit Read/Write



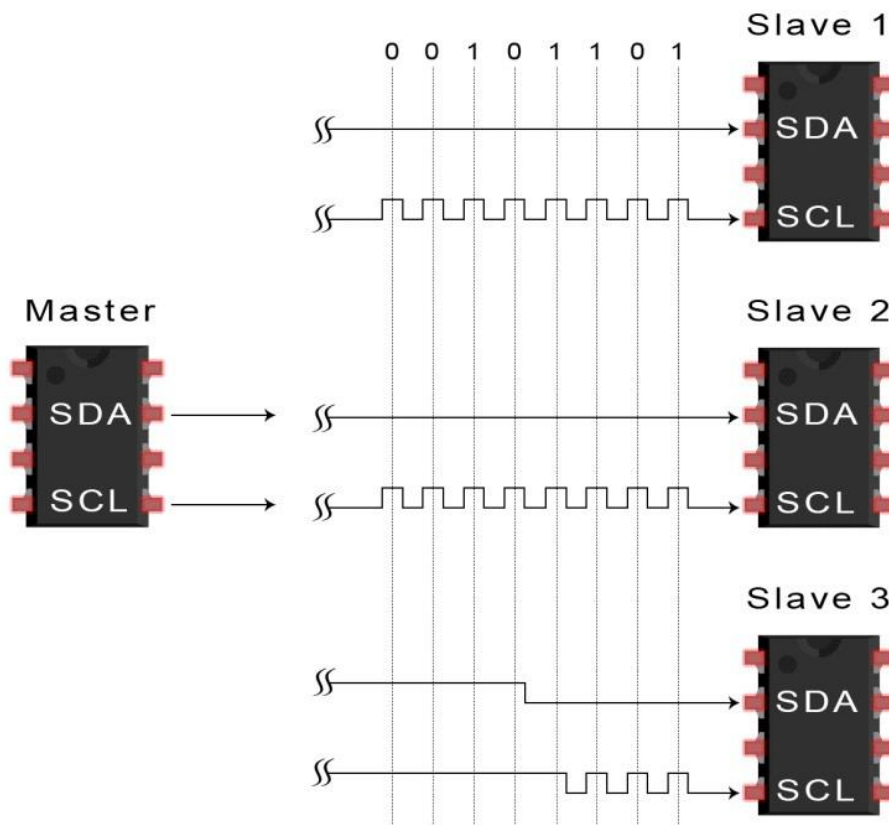
- Mỗi thiết bị Slave so sánh địa chỉ được gửi từ thiết bị Master đến địa chỉ riêng của nó. Nếu địa chỉ trùng khớp, thiết bị Slave gửi về một bit ACK bằng cách kéo đường SDA xuống thấp và bit ACK / NACK được thiết lập là '0'. Nếu địa chỉ từ thiết bị Master không khớp với địa chỉ riêng của thiết bị Slave thì đường SDA ở mức cao và bit ACK / NACK sẽ ở mức '1' (mặc định).
- Thiết bị Master gửi hoặc nhận khung dữ liệu. Nếu thiết bị Master muốn gửi dữ liệu đến thiết bị Slave, bit Read / Write là mức điện áp thấp. Nếu thiết bị Master đang nhận dữ liệu từ thiết bị Slave, bit này là mức điện áp cao.



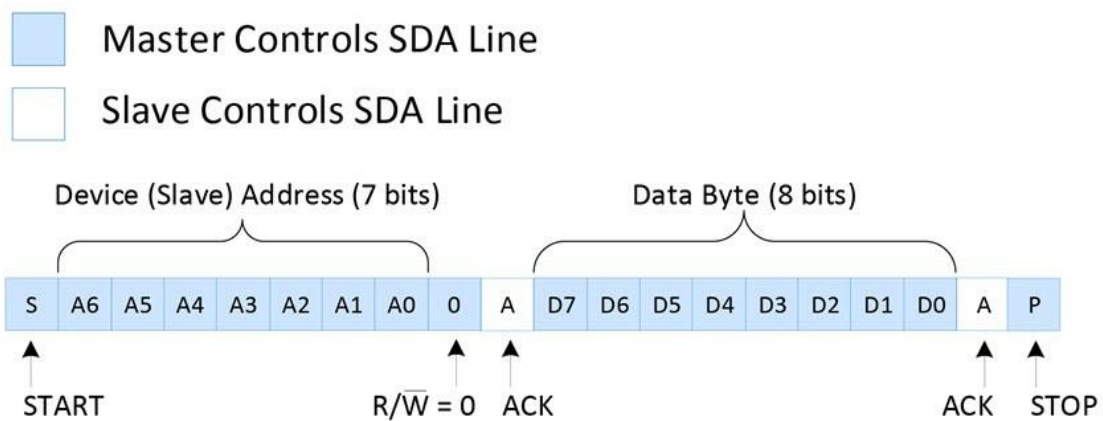
- Nếu khung dữ liệu được thiết bị Slave nhận được thành công, nó sẽ thiết lập bit ACK / NACK thành '0', báo hiệu cho thiết bị Master tiếp tục



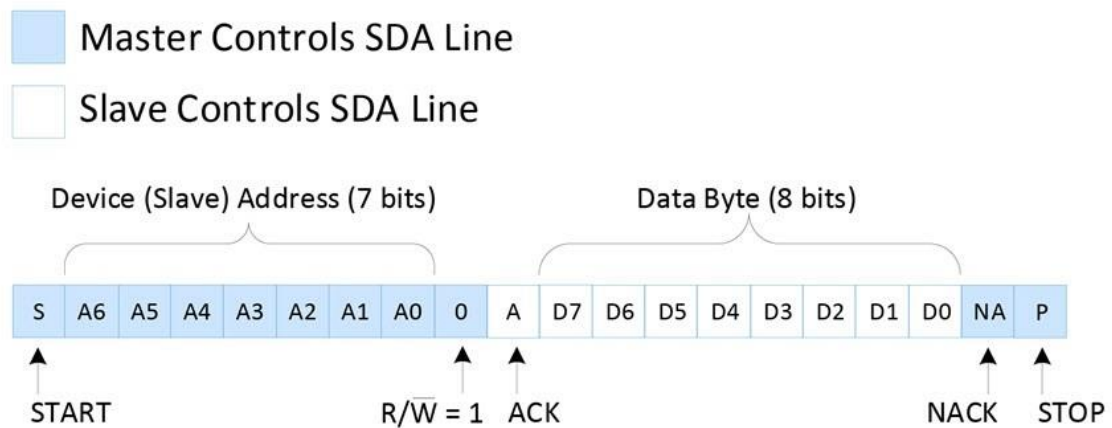
- Sau khi tất cả dữ liệu được gửi đến thiết bị Slave, thiết bị Master gửi điều kiện dừng để báo hiệu cho tất cả các thiết bị Slave biết rằng việc truyền dữ liệu đã kết thúc.



Hình dưới đây thể hiện toàn bộ các bit dữ liệu được gửi trên đường SDA và thiết bị điều khiển chúng khi thiết bị Master gửi dữ liệu đến thiết bị Slave.



Hình dưới đây thể hiện toàn bộ các bit dữ liệu được gửi trên đường SDA và thiết bị điều khiển chúng khi thiết bị Master nhận dữ liệu đến thiết bị Slave.



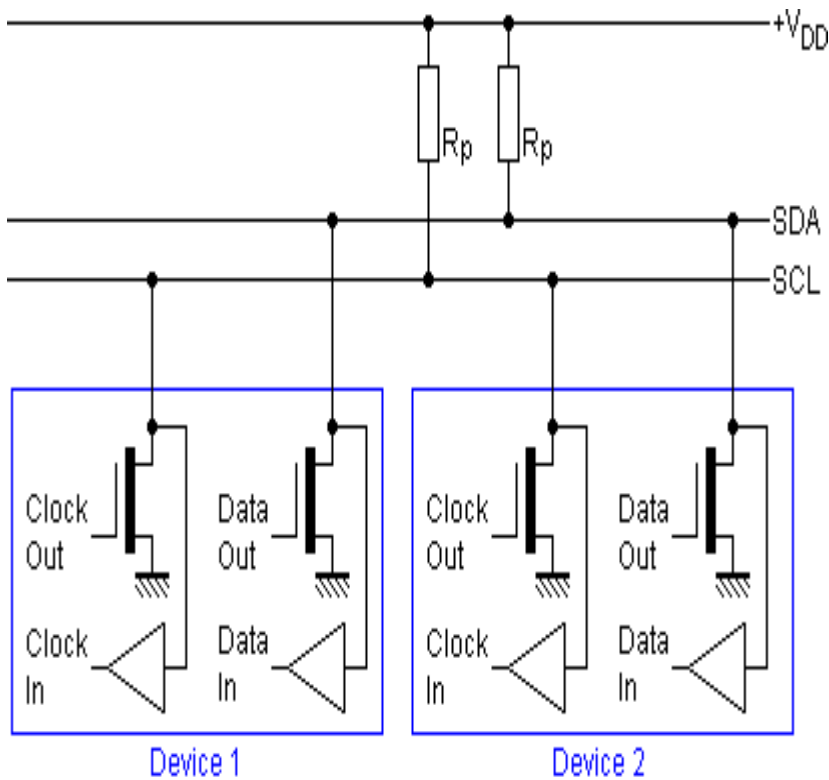
### 3.2.10 Các chế độ hoạt động của I2C

Dựa vào tốc độ ta chia làm 2 loại

- Chế độ chuẩn (standard mode) hoạt động ở tốc độ 100 Kbit/s.
- Chế độ tốc độ thấp (low-speed mode) hoạt động ở tốc độ 10 Kbit/s.

Nếu chia theo quan hệ chủ tớ:

- Một Master một Slave.
- Một Master nhiều Slave.
- Nhiều Master nhiều Slave.



### 3.3 CHƯƠNG TRÌNH ĐIỀU KHIỂN

#### 3.3.1 Hệ thống cửa và đèn

```
#include <Servo.h> // Khai báo thư viện sử dụng cho động cơ
#include <LiquidCrystal_I2C.h> // Khai báo thư viện LCD sử dụng I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // 0x27 địa chỉ LCD, 16 cột và 2 hàng
#include <Keypad.h> // Khai báo thư viện Keypad
```

```
const byte ROWS = 4; // Bốn hàng
```

```
const byte COLS = 3; // Ba cột
```

```
char keys[ROWS][COLS] = {
```

```
    {'1', '2', '3'}, {'4', '5', '6'}, {'7', '8', '9'}, {'*', '0', '#'};}
```

```
byte rowPins[ROWS] = {5, 4, 3, 2};
```

```
byte colPins[COLS] = {8, 7, 6};
```

```
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
COLS);
```



```
Servo myServo;
```

```
char STR[4] = {'2', '0', '2', '3'}; // Cài đặt mật khẩu tùy ý
```

```
char str[4] = {' ', ' ', ' ', ' '};
```

```
int i, j, count = 0;
```

```
const int ledPin = 10; // chân kết nối với LED
```

```
const int pirPin = 11; // chân kết nối với cảm biến hồng ngoại
```

```
void setup() {
```

```
    pinMode(ledPin, OUTPUT);
```

```
    myServo.attach(9); // Khai báo chân điều khiển động cơ
```

```
    lcd.init(); // Khai báo sử dụng LCD
```

```
    lcd.begin(16, 2);
```

```
    lcd.backlight();
```

```
    lcd.clear();
```

```
    lcd.print(" Welcome Home");
```

```
    delay(5000);
```

```
    lcd.begin(16, 2);
```

```
    lcd.backlight();
```

```
    lcd.clear();
```

```
    lcd.print(" Enter Password");
```

```
}
```

```
void loop() {
```

```
    char key = keypad.getKey(); // Ký tự nhập vào sẽ gán cho biến Key
```

```
    if (key) {
```

```
        if (i < 4) {
```

```
    str[i] = key;
    lcd.setCursor(i + 6, 1);
    lcd.print(str[i]);
    delay(1000);
    lcd.setCursor(i + 6, 1);
    lcd.print("*");
    i++;
}
if (i == 4) {
    count = 1;
}
}

if (count == 1) {
    if (str[0] == STR[0] && str[1] == STR[1] && str[2] == STR[2] &&
        str[3] == STR[3]) {
        lcd.clear();
        lcd.print("  Correct!");
        delay(3000);
        digitalWrite(ledPin, HIGH);
        myServo.write(180); // Mở cửa
        lcd.clear();
        lcd.print("  Opened!");
        i = 0;
        count = 0;
        delay(8000); // Đợi trong 8 giây
        digitalWrite(ledPin, LOW);
        myServo.write(90); // Đóng cửa
```

```
    lcd.clear();
    lcd.print(" Enter Password");
} else {
    lcd.clear();
    lcd.print(" Incorrect!");
    delay(3000);
    lcd.clear();
    lcd.print(" Try Again!");
    delay(3000);
    lcd.clear();
    lcd.print(" Enter Password");
    i = 0;
    count = 0;
}
}

switch (key) {
case '#':
    lcd.clear();
    myServo.write(90); //Đóng cửa
    lcd.print(" Closed!");
    delay(10000);
    lcd.clear();
    lcd.print(" Enter Password");
    i = 0;
    break;
}
```

### **3.3.2 Hệ thống quạt, đèn:**

```
#define BLYNK_TEMPLATE_ID "TMPL6Bmyd_fJt"
#define BLYNK_TEMPLATE_NAME "Hệ Thống Đền Quạt"
#define BLYNK_AUTH_TOKEN
"gHFi3W7XPSggdnaRz50913Td0zJKZZsq"

// Khai báo các thư viện cần thiết
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h> // Include the WiFiManager library
#include <ESP8266WiFi.h> // Include the Wi-Fi library

// Khai báo các chân kết nối với Led
#define LED1 D1
#define LED2 D2
#define LED3 D3
#define LED4 D4
#define Fan1 D5
#define Fan2 D6

// Khai báo mã token của Blynk
char auth[] = BLYNK_AUTH_TOKEN;

// Khai báo thông tin wifi
char ssid[] = "";
char pass[] = "";
```

```
// Hàm thiết lập
void setup()
{
  // Khởi tạo serial monitor với tốc độ 9600 baud
  Serial.begin(115200);

  WiFiManager wifiManager; // Create an instance of the WiFiManager class

  wifiManager.autoConnect("AutoConnectAP"); // Automatically connect to a
  previously saved network
  // If there is no previous network, it will create an access point with the name
  "AutoConnectAP"
  // You can use any Wi-Fi device to connect to that access point and enter a new
  SSID and password

  Serial.println("Connected to WiFi"); // Print a message when connected

  // Khởi tạo các chân kết nối với Led là output
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(Fan1, OUTPUT);
  pinMode(Fan2, OUTPUT);
  // Kết nối wifi và Blynk
  Blynk.begin(auth, ssid, pass);
}
```

```
// Hàm vòng lặp chính
void loop()
{
  // Cập nhật trạng thái của Blynk
  Blynk.run();
}

// Hàm điều khiển Led 1 theo nút bấm trên app Blynk
BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // Lấy giá trị từ nút bấm

  if (pinValue == 1) {
    digitalWrite(LED1, HIGH); // Bật Led 1
  } else {
    digitalWrite(LED1, LOW); // Tắt Led 1
  }
}

// Hàm điều khiển Led 2 theo nút bấm trên app Blynk
BLYNK_WRITE(V2)
{
  int pinValue = param.asInt(); // Lấy giá trị từ nút bấm

  if (pinValue == 1) {
    digitalWrite(LED2, HIGH); // Bật Led 2
  } else {
    digitalWrite(LED2, LOW); // Tắt Led 2
  }
}
```

```
}  
}  
  
// Hàm điều khiển Led 3 theo nút bấm trên app Blynk  
BLYNK_WRITE(V3)  
{  
  int pinValue = param.asInt(); // Lấy giá trị từ nút bấm  
  
  if (pinValue == 1) {  
    digitalWrite(LED3, HIGH); // Bật Led 3  
  } else {  
    digitalWrite(LED3, LOW); // Tắt Led 3  
  }  
}  
  
// Hàm điều khiển Led 4 theo nút bấm trên app Blynk  
BLYNK_WRITE(V4)  
{  
  int pinValue = param.asInt(); // Lấy giá trị từ nút bấm  
  
  if (pinValue == 1) {  
    digitalWrite(LED4, HIGH); // Bật Led 4  
  } else {  
    digitalWrite(LED4, LOW); // Tắt Led 4  
  }  
}  
  
// Hàm điều khiển Quạt 1 theo nút bấm trên app Blynk  
BLYNK_WRITE(V5)  
{
```

```
int pinValue = param.asInt(); // Lấy giá trị từ nút bấm

if (pinValue == 1) {
    digitalWrite(Fan1, HIGH); // Bật Quạt 1
} else {
    digitalWrite(Fan1, LOW); // Tắt Quạt 1
}
}

// Hàm điều khiển Quạt 2 theo nút bấm trên app Blynk
BLYNK_WRITE(V6)
{
    int pinValue = param.asInt(); // Lấy giá trị từ nút bấm

    if (pinValue == 1) {
        digitalWrite(Fan2, HIGH); // Bật Quạt 1
    } else {
        digitalWrite(Fan2, LOW); // Tắt Quạt 2
    }
}
```

### 3.3.3 Hệ thống cảm biến và LCD :

```
#define BLYNK_TEMPLATE_ID "TMPLH7PMUMgg"
#define BLYNK_TEMPLATE_NAME "ESP8266"
#define BLYNK_AUTH_TOKEN "LgZdNCTITd6gLDn0juGptZraemHv_TII"

// Your WiFi Credentials.
// Set password to "" for open networks.
char ssid[] = "Sinh Vien"; // type your wifi name
char pass[] = "66778899"; // type your wifi password
```



```
// define the GPIO connected with Sensors & LEDs
#define MQ2_SENSOR  A0 //A0
#define RAIN_SENSOR  D5 //D1
#define DHTPIN      D3 //DHT11 Pin
#define BUZZERPIN   D6

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h> // Include the WiFiManager library#include
<ESP8266WiFi.h> // Include the Wi-Fi library

#include <WiFiManager.h> // Include the WiFiManager library
#define DHTTYPE DHT11 // DHT 11

int MQ2_SENSOR_Value = 0;
int RAIN_SENSOR_Value = 0;
float humi = 0;
float temp = 0;
bool isconnected = false;
char auth[] = BLYNK_AUTH_TOKEN;
```

```
#define VPIN_BUTTON_1  V2
```

```
#define VPIN_BUTTON_2  V8
```

```
BlynkTimer timer;
```

```
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void checkBlynkStatus() { // called every 2 seconds by SimpleTimer
```

```
  getSensorData();
```

```
  isConnected = Blynk.connected();
```

```
  if (isconnected == true) {
```

```
    Serial.println("Blynk Connected");
```

```
    sendSensorData();
```

```
    //Serial.println("Blynk Connected");
```

```
  }
```

```
  else{
```

```
    Serial.println("Blynk Not Connected");
```

```
  }
```

```
}
```

```
void getSensorData()
```

```
{
```

```
  MQ2_SENSOR_Value = map(analogRead(MQ2_SENSOR), 0, 1024, 0, 100);
```

```
  RAIN_SENSOR_Value = digitalRead(RAIN_SENSOR);
```

```
  humi = dht.readHumidity(); // Read humidity value from DHT sensor
```

```
  temp = dht.readTemperature(); // Read temperature value from DHT sensor
```

```
if (MQ2_SENSOR_Value > 50 ){
    tone(BUZZERPIN, 1500, 500);
}
else if (RAIN_SENSOR_Value == 0 ){
    tone(BUZZERPIN, 1500, 500);
}
else{
    Serial.println("Bình thường");
}

}

void sendSensorData()
{
    humi = dht.readHumidity(); // Read humidity value from DHT sensor
    temp = dht.readTemperature(); // Read temperature value from DHT sensor
    Blynk.virtualWrite(VPIN_BUTTON_1, MQ2_SENSOR_Value);
    Blynk.virtualWrite(VPIN_BUTTON_2, "No Water Detected.");

    if (MQ2_SENSOR_Value > 50 )
    {
        Blynk.logEvent("gas", "Gas Detected!");
    }
    else if (RAIN_SENSOR_Value == 0 )
    {
        Blynk.logEvent("rain", "Water Detected!");
        Blynk.virtualWrite(VPIN_BUTTON_2, "Water Detected!");
    }
}
```

```
}
```

```
Blynk.virtualWrite(V0, temp); // Send temperature value to Blynk app
```

```
Blynk.virtualWrite(V1, humi); // Send humidity value to Blynk app
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Temp: ");
```

```
lcd.print(temp);
```

```
lcd.print(" C");
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Humi: ");
```

```
lcd.print(humi);
```

```
lcd.print(" %");
```

```
delay(4000);
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("MQ2: ");
```

```
lcd.print(MQ2_SENSOR_Value);
```

```
lcd.setCursor(0, 1);
```

```
if (MQ2_SENSOR_Value > 50) {
```

```
    lcd.print("Gas Detected!");
```

```
} else {
```

```
    lcd.print("No Gas Detected");
```

```
}
```

```
delay(3000);
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
lcd.print("Rain: ");
if (RAIN_SENSOR_Value == LOW) {
  lcd.print("Detected");
} else {
  lcd.print("No Rain");
}
delay(3000);
lcd.clear();
}
void setup()
{
  Serial.begin(9600);
  delay(10);
  // Initialize DHT11 sensor
  dht.begin();
  pinMode(MQ2_SENSOR, INPUT);
  pinMode(RAIN_SENSOR, INPUT);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("  WELCOME");
  lcd.setCursor(0, 1);
  lcd.print("  SMART HOME");
  delay(3000);
  lcd.clear();
  Serial.println("\n");
```

```
WiFiManager wifiManager; // Create an instance of the WiFiManager class

wifiManager.autoConnect("AutoConnectAP"); // Automatically connect to a
previously saved network
// If there is no previous network, it will create an access point with the name
"AutoConnectAP"
// You can use any Wi-Fi device to connect to that access point and enter a new
SSID and password
Serial.println("Connected to WiFi"); // Print a message when connected
pinMode(MQ2_SENSOR, INPUT);
pinMode(RAIN_SENSOR, INPUT);
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("  WELCOME");
lcd.setCursor(0, 1);
lcd.print("  SMART HOME");
delay(3000);
lcd.clear();
WiFi.begin(ssid, pass);
timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is
connected every 2 seconds
Blynk.config(auth);
delay(1000);
}
void loop()
```

```
{  
  Blynk.run();  
  timer.run();  
}
```

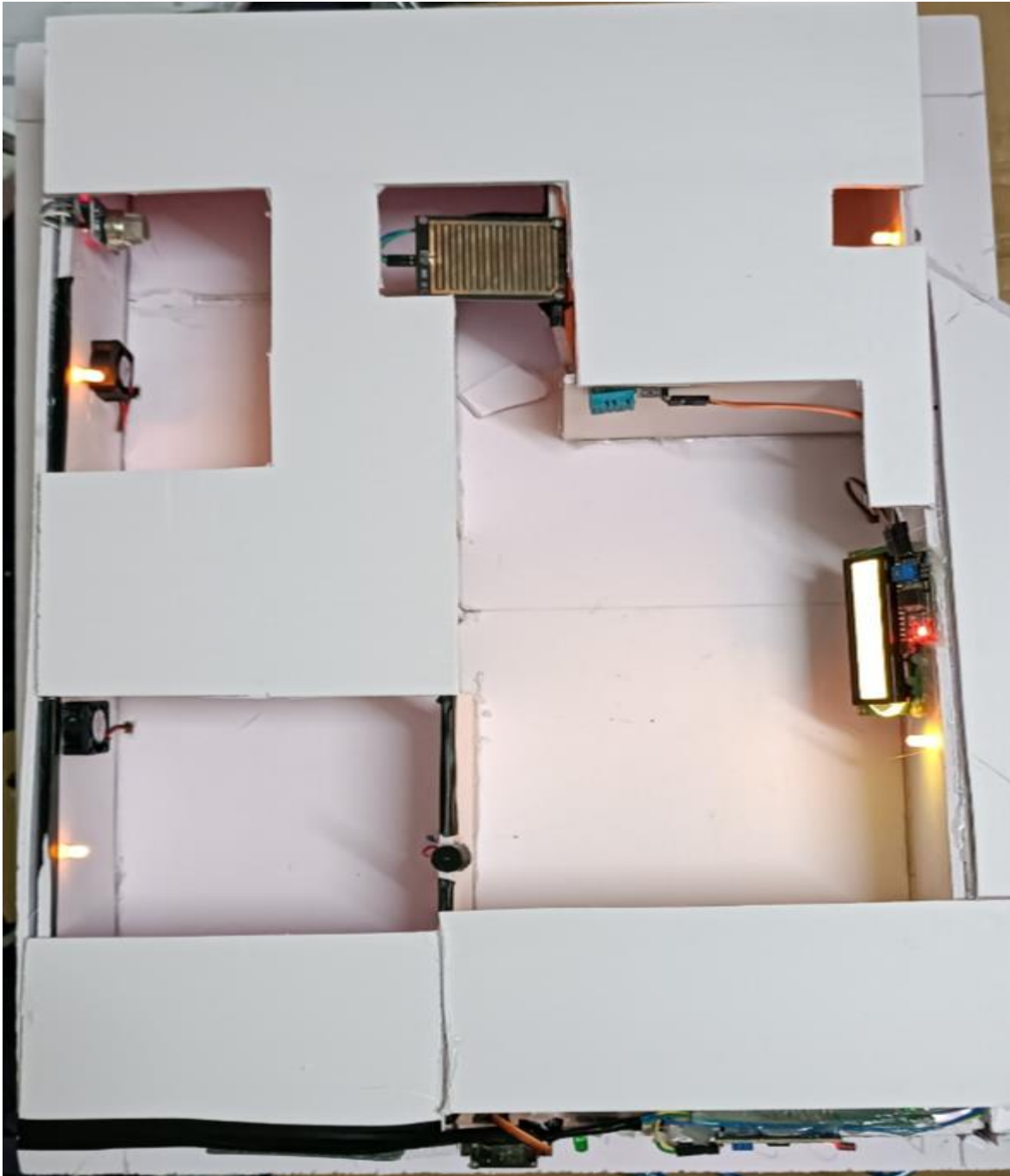
```
}
```

### 3.3.4 Mô hình hoàn thiện



**Hình 3.4 Mô hình hoàn thiện**

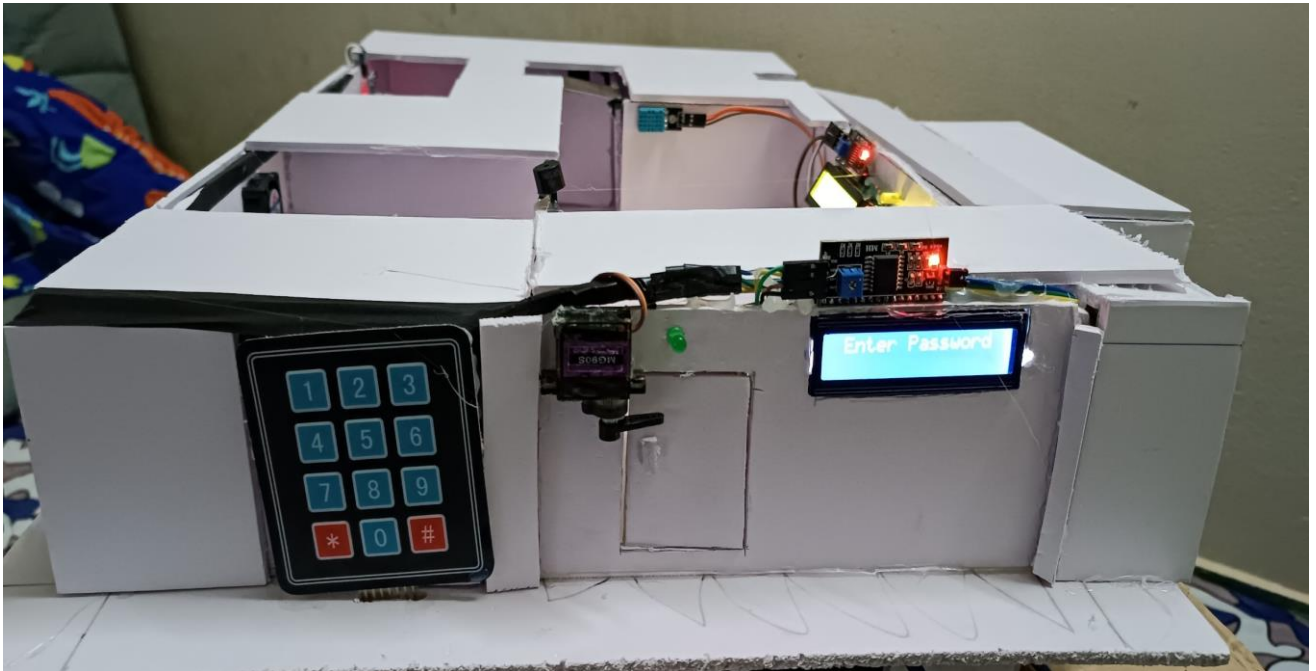




**Hình 3.5 Mô hình hoàn thiện**

## Thực nghiệm

Khi khởi động hệ thống, màn hình lcd hiển thị:

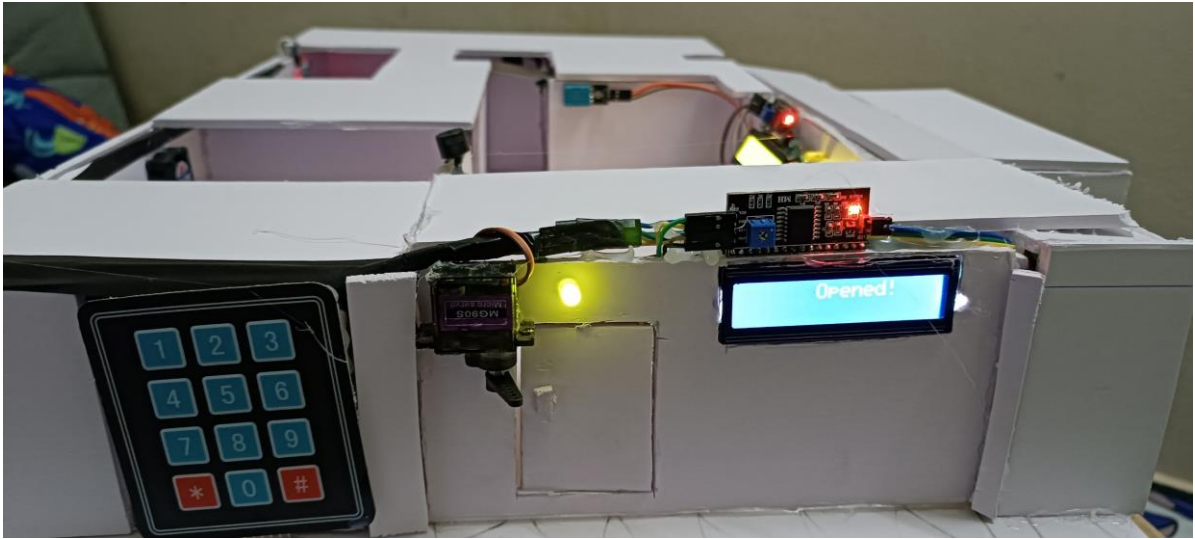


**Hình 3.6** Màn hình LCD phía trong và ngoài hệ thống



**Hình 3.7 Đèn quạt phía trong hệ thống**

Khi người dùng nhập đúng mật khẩu được cài đặt trong hệ thống thì cửa mở.



**Hình 3.8** Trạng thái server khi hệ thống nhận đúng mật khẩu

Tiếp tục ở phía trong, hệ thống cảm biến sẽ hoạt động và liên tục hiển thị ở LCD phía trong nhà.



**Hình 3.9** Trạng thái độ ẩm nhiệt độ



**Hình 3.10 LCD báo hệ thống không phát hiện mưa**



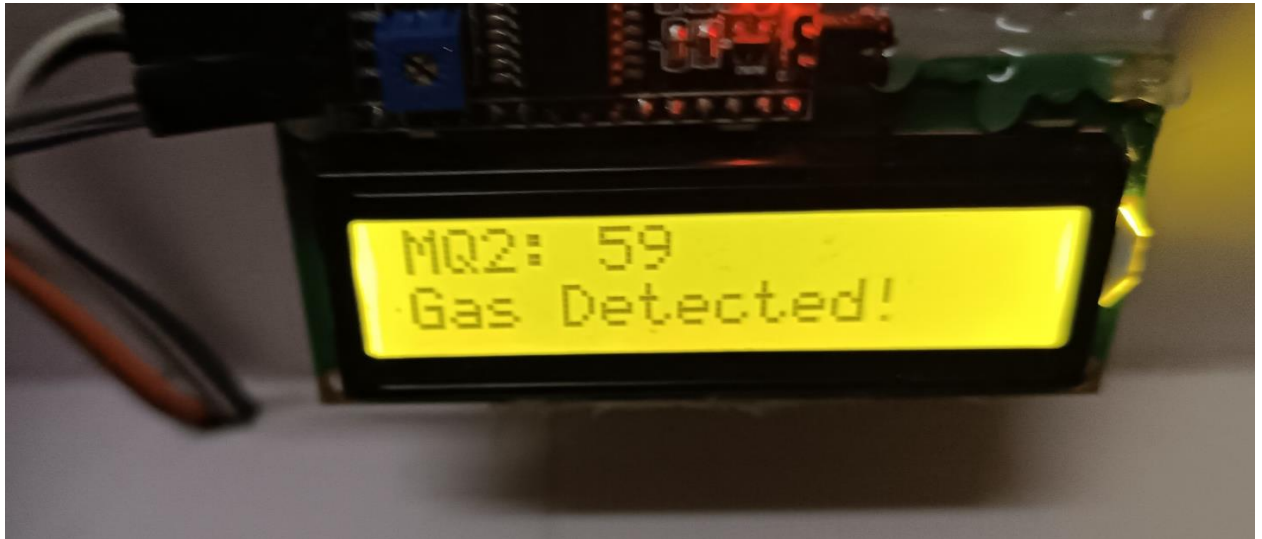
**Hình 3.11 LCD báo hệ thống không rò rỉ khí gas**

Ngược lại, nếu hệ thống cảm biến phát hiện có mưa hệ thống sẽ đồng thời phát ra tín hiệu phía còi hú, và thông báo lên LCD, hệ thống báo động trên điện thoại hoặc web.

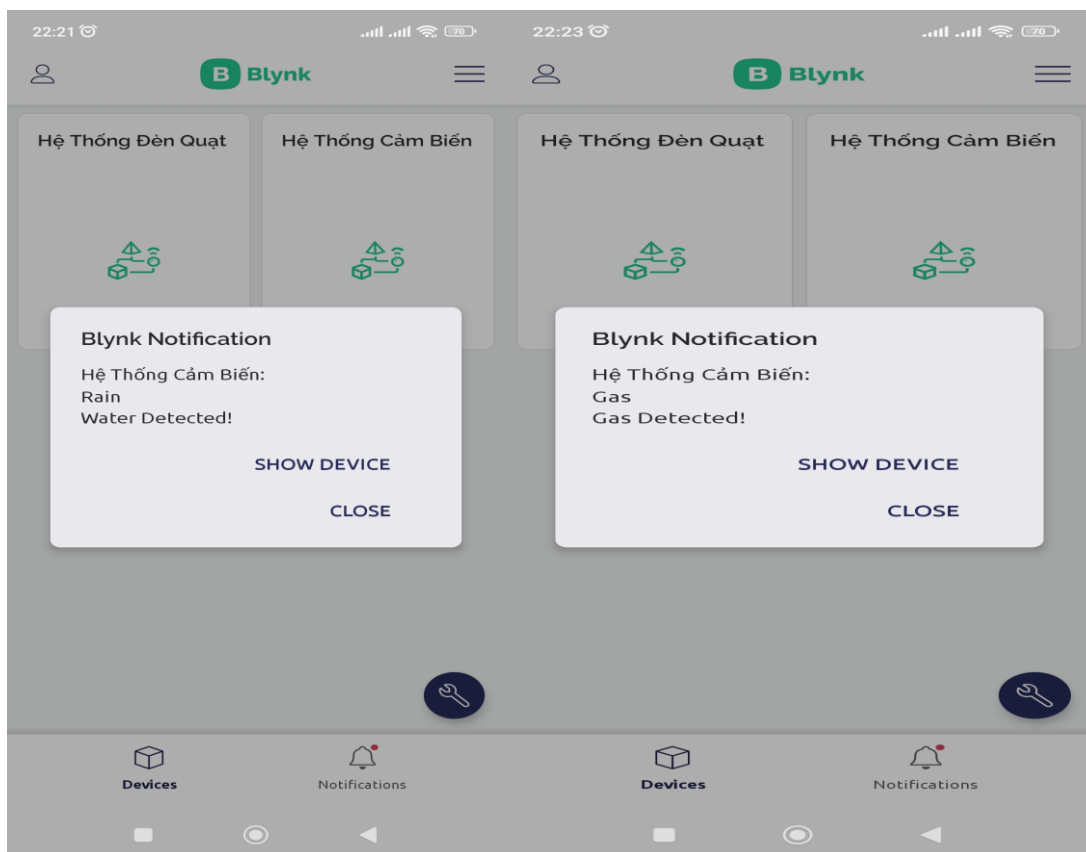


**Hình 3.12 LCD thông báo có mưa**

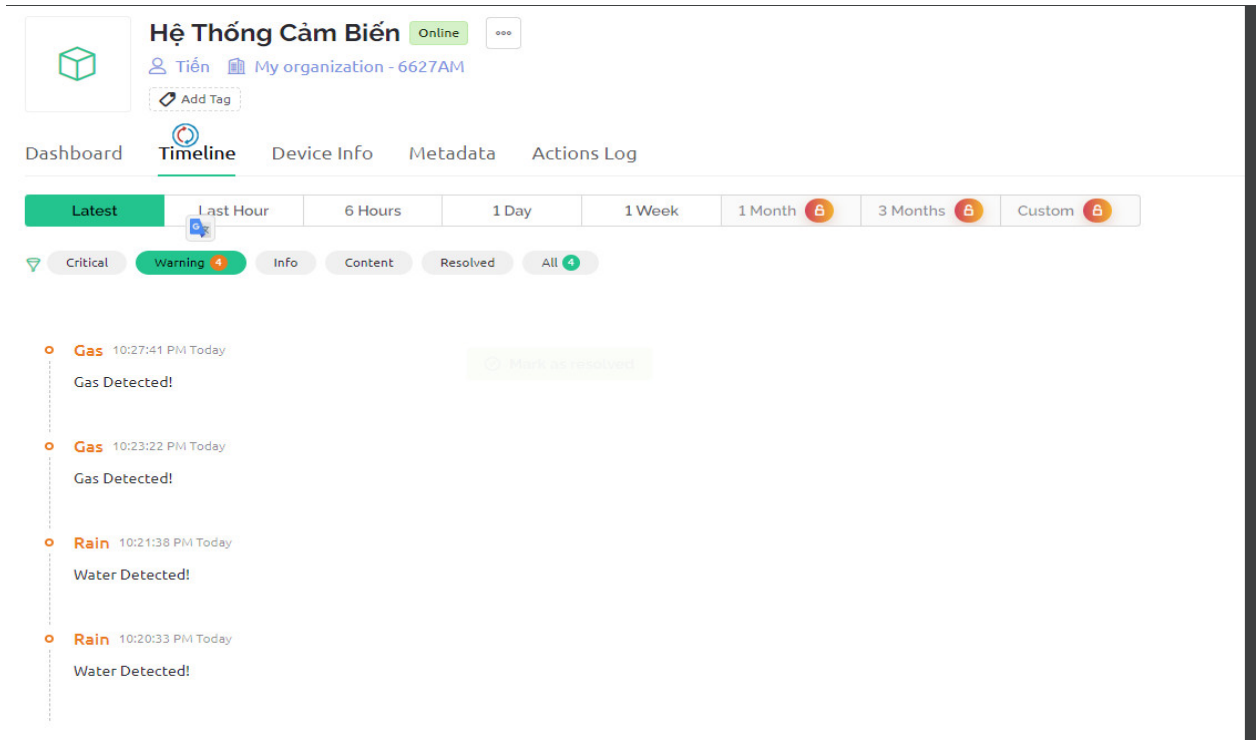




**Hình 3.13 LCD hiển thị hệ thống rò rỉ khí gas**

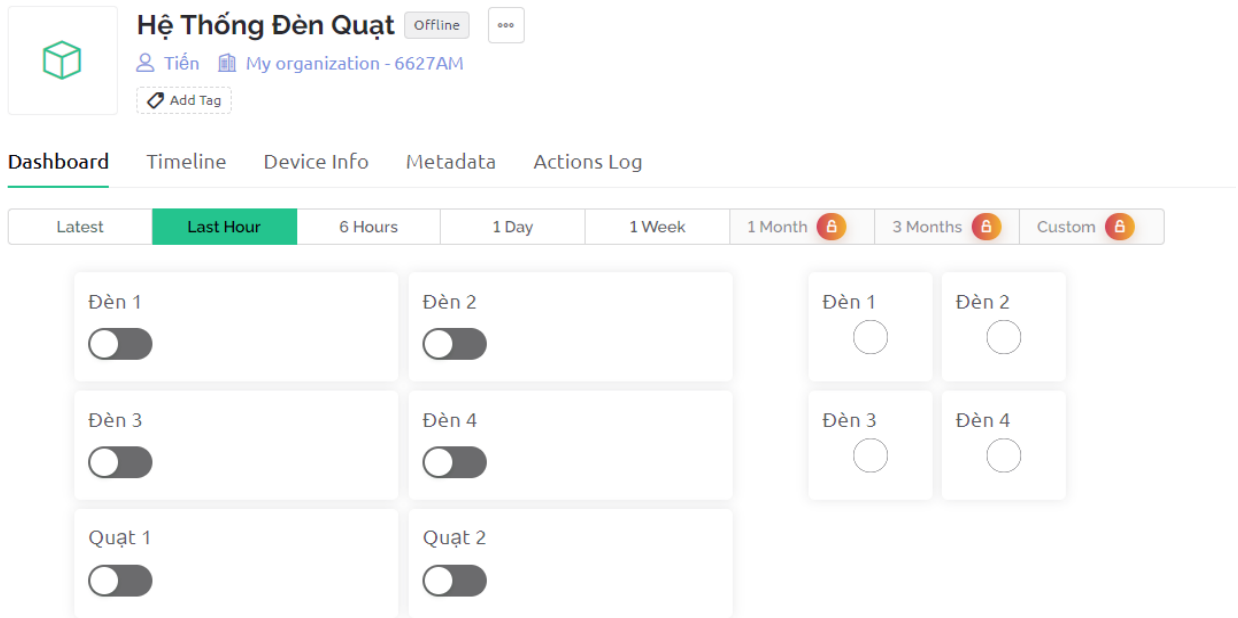


**Hình 3.14 Cảnh báo mưa và khí ga trên app điện thoại**



**Hình 3.15 Cảnh báo mưa và khí ga trên web**

### 3.3.5 Điều khiển các thiết bị trong nhà qua giao diện web

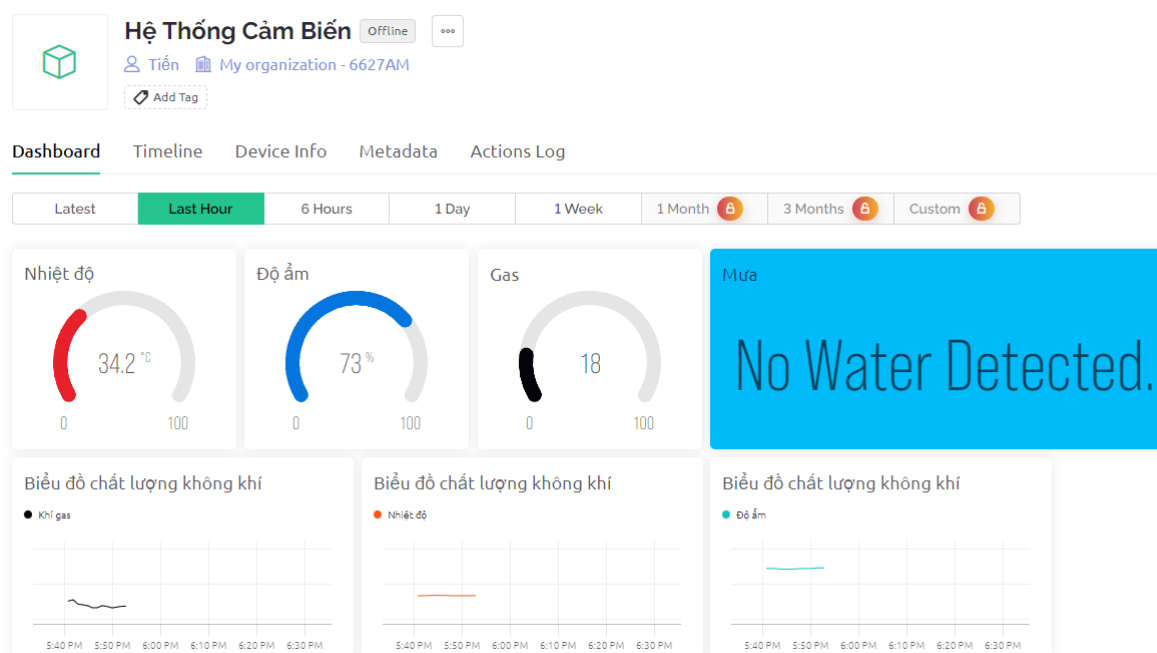


**Hình 3.16 Giao diện web điều khiển các thiết bị**

Khi nhấn nút điều khiển trên web để bật tắt các thiết bị hay chế độ, mỗi nút điều khiển đã được đặt trước 1 kí tự tương ứng, kí tự này cơ chế chủ động-

truy vấn (push-pull). Theo cơ chế này, các dữ liệu được truyền từ thiết bị điều khiển đến Blynk Server sẽ được đẩy (push) liên tục theo một chu kỳ thời gian nhất định. Các dữ liệu này sẽ được lưu trữ tạm thời trong Blynk Server để có thể được truy cập bởi các thiết bị khác thông qua giao thức Internet hoặc Bluetooth., tại đây, kí tự này được liên tục so sánh với các lệnh trong bộ điều khiển, từ đó kí tự tương ứng với thiết bị nào được truyền về bộ điều khiển quản lý thiết bị đó ở các phòng, từ đó sẽ điều khiển trạng thái của thiết bị.

### 3.3.6 Truyền dữ liệu từ các thiết bị lên giao diện web



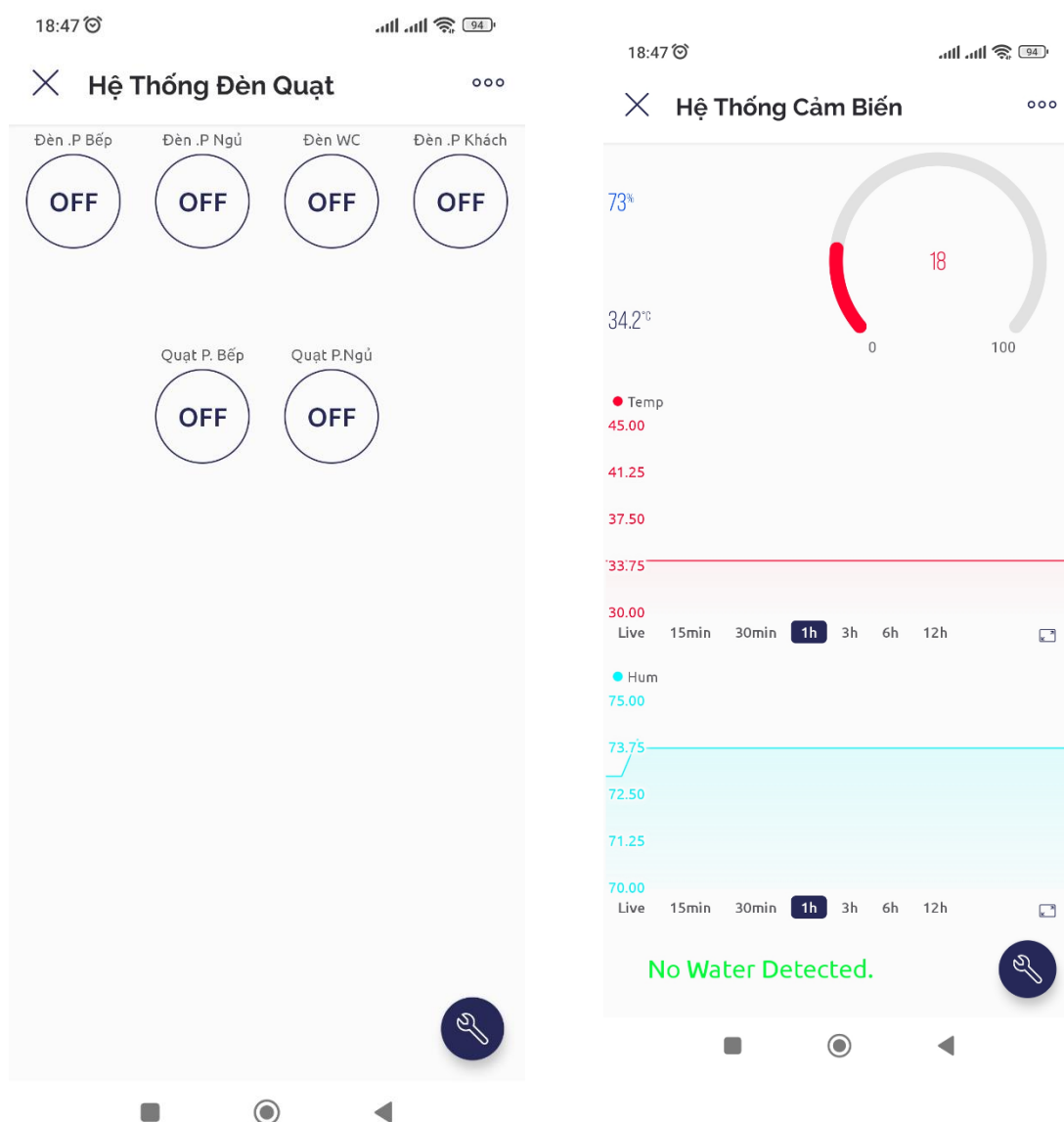
**Hình 3.17** Giao diện web giám sát các cảm biến

Quá trình truyền dữ liệu trạng thái của các thiết bị lên web ngược lại với quá trình nhận dữ liệu ở trên: Khi ta sử dụng các nút vật lý để bật tắt các thiết bị trong nhà, 1 kí tự được ấn định tương ứng cho mỗi thiết bị cũng sẽ được bộ điều khiển quản lý thiết bị đó truyền tới bo điều khiển trung tâm, tại đây, dữ liệu này được truyền tới server và qua giao thức web sẽ cập nhật trạng thái các thiết bị lại trên giao diện web điều khiển.



### 3.3.7 Điều khiển các thiết bị trong nhà và truyền dữ liệu từ các thiết bị lên giao diện app điện thoại

Tương tự như ở web, cơ chế hoạt động của hệ thống vẫn không thay đổi nhiều, hệ thống vẫn được thực hiện thông qua cơ chế chủ động-truy vấn (push-pull) và sử dụng một giao thức truyền thông đặc biệt để đảm bảo tính ổn định và đáng tin cậy của dữ liệu. Các thiết bị được kết nối với Blynk Server cũng có thể lắng nghe và gửi yêu cầu để truy cập dữ liệu từ các thiết bị khác trên mạng.



**Hình 3.18** Giao diện giám sát và điều khiển các thiết bị qua app

## CHƯƠNG IV: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

### 4.1 KẾT QUẢ ĐẠT ĐƯỢC:

- Hệ thống điều khiển qua giao diện web và truyền nhận trạng thái thiết bị đã đi vào hoạt động với các chức năng:
  - + Bật, tắt các thiết bị như: đèn, điều hòa, cửa ra vào mở bằng mặt khẩu
  - + Cảnh báo mưa, cảnh báo cháy, rò rỉ khí gas

### 4.2 ĐÁNH GIÁ KẾT QUẢ:

- Ưu điểm:
  - + Hệ thống đáp ứng được các chức năng yêu cầu
  - + Giao diện web và app đơn giản, dễ dàng thao tác
  - + Thuận tiện hơn cho người sử dụng quản lý các thiết bị trong nhà khi ra ngoài
- Nhược điểm:
  - + Đôi khi đáp ứng chưa nhạy, có độ trễ
  - + Xử lý nhiễu chưa ổn định

### 4.3 KẾT LUẬN

Sau khi nghiên cứu và hoàn thành đề tài, em nhận thấy mô hình đã hiệu quả được 80%. Trong thời gian nghiên cứu và thực hiện đề tài, bản thân em đã học hỏi và tìm hiểu thêm được nhiều kiến thức mới cũng như củng cố lại kiến thức đã học giúp hoàn thành đề tài này.

Vì đây là đề tài hướng đến việc giúp cho hoạt động của một hệ thống của nhà và phải hoạt động một cách tự hành nên phải chú trọng độ ổn định và chính xác dẫn đến nhiều khó khăn trong quá trình lập trình. Đồng thời nhờ sự hướng dẫn của giảng viên hướng dẫn và các tài liệu tham khảo thì em đã giải quyết được tương đối yêu cầu của đề tài.

Trong quá trình thực hiện đề tài về phần cứng, em đã tìm hiểu được chức năng của các chân IO của board Arduino, Esp8266, kết hợp với các cảm biến

để đọc giá trị...Việc kết hợp các module này lại với nhau để làm việc ổn định mất nhiều thời gian nhưng cũng học hỏi được rất nhiều trong quá trình làm.

Tuy rằng sản phẩm đã được hoàn thành nhưng đối với bản thân em vẫn nhận thấy sản phẩm còn nhiều thiếu sót, cần được chỉnh sửa và cải tiến hơn

#### **4.4 HƯỚNG PHÁT TRIỂN**

Hệ thống điều khiển và giám sát qua mạng internet đang ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như an ninh, quốc phòng, chăn nuôi, y tế, gia đình, trồng trọt,... Điều này cho thấy sự phát triển mạnh mẽ của công nghệ thông tin và truyền thông. Việc sử dụng kiến thức học được để tiếp cận với công nghệ và xu hướng của thế giới là rất cần thiết để phát triển hệ thống điều khiển và giám sát qua mạng internet.

Để phát triển hệ thống này, có thể đề xuất một số hướng như sau:

Lắp đặt hệ thống camera giám sát: Hệ thống camera giám sát là một trong những thành phần quan trọng của hệ thống điều khiển và giám sát qua mạng internet. Với việc lắp đặt hệ thống camera giám sát, người dùng có thể quản lý, giám sát và điều khiển các thiết bị từ xa thông qua mạng internet.

Ứng dụng hệ thống giám sát và điều khiển trong y tế, giáo dục: Hệ thống điều khiển và giám sát qua mạng internet cũng có thể được áp dụng trong lĩnh vực y tế và giáo dục. Ví dụ, hệ thống này có thể được sử dụng để giám sát sức khỏe của bệnh nhân từ xa, hoặc để giáo dục từ xa.

Xây dựng phần mềm quản lý trên các thiết bị di động: Phần mềm quản lý trên các thiết bị di động là một trong những cách tiện lợi để quản lý và giám sát hệ thống từ xa. Phần mềm này có thể được thiết kế để tự động thông báo cho người dùng về các điều kiện bất lợi mà không cần phải truy cập vào trang web. Ngoài ra, phần mềm này cũng có thể cung cấp các tính năng quản lý thông minh như thông báo lỗi, tùy chỉnh thông báo, và hỗ trợ tương tác từ xa.

Qua đề tài này, ta có thể nhận thấy rằng hệ thống điều khiển và giám sát qua mạng internet đang ngày càng trở nên quan trọng trong cuộc sống hiện đại. Vì vậy, việc nghiên cứu và phát triển hệ thống này là rất cần thiết để đáp ứng.

## TÀI LIỆU THAM KHẢO

- [1] CẢM BIẾN ĐO LƯỜNG VÀ ĐIỀU KHIỂN – THS. NGUYỄN TẤN PHƯỚC
- [2] LÝ THUYẾT ĐIỀU KHIỂN TỰ ĐỘNG – NGUYỄN THỊ PHƯƠNG HÀ, HUỖNH THÁI HOÀNG
- [3] TÀI LIỆU MÔN KỸ THUẬT SỐ – THS. LƯU HOÀNG
- [4] ĐO LƯỜNG ĐIỀU KHIỂN BẰNG MÁY TÍNH – THS. NGUYỄN LƯƠNG THANH TÙNG
- [5] SMART HOME AUTOMATION WITH LINUX AND RASPBERRY PI - STEVEN GOODWIN
- [6] BUILDING SMART HOMES WITH RASPBERRY PI ZERO - MARCO SCHWARTZ
- [7] HOME AUTOMATION WITH ARDUINO: AUTOMATE YOUR HOME USING OPEN-SOURCE HARDWARE - MARCO SCHWARTZ VÀ OGÉ NWOKOLO
- [8] [HTTPS://ARDUINO.VN/BAI-VIET/68-CAI-DAT-DRIVER-VA-ARDUINO-IDE](https://arduino.vn/bai-viet/68-cai-dat-driver-va-arduino-ide)
- [9] [HTTPS://WWW.INSTRUCTABLES.COM/ESP8266-BLYNK-SMART-HOME/](https://www.instructables.com/ESP8266-BLYNK-SMART-HOME/)
- [10] [HTTPS://WWW.INSTRUCTABLES.COM/ARDUINO-SMART-HOME/](https://www.instructables.com/ARDUINO-SMART-HOME/)
- [11] [HTTPS://WWW.IOTFORALL.COM/ULTIMATE-GUIDE-TO-BUILDING-A-SMART-HOME](https://www.iotforall.com/ultimate-guide-to-building-a-smart-home)