

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỀU KHIỂN
CÁC THIẾT BỊ ĐIỆN TRONG NHÀ
SỬ DỤNG CÔNG NGHỆ LORA.**

**GVHD: TS. NGUYỄN THANH NGHĨA
SVTH : LÊ ANH QUANG
TRẦN HỮU NGHĨA**



Tp. Hồ Chí Minh, tháng 6/2023

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



ĐỒ ÁN TỐT NGHIỆP

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỀU
KHIỂN CÁC THIẾT BỊ ĐIỆN TRONG NHÀ SỬ
DỤNG CÔNG NGHỆ LORA.**

SVTH: LÊ ANH QUANG
MSSV: 19161154
SVTH: TRẦN HỮU NGHĨA
MSSV: 19161141
Khoá: 2019
Ngành: CNKT ĐIỆN TỬ-VIỄN THÔNG
GVHD: TS NGUYỄN THANH NGHĨA

Tp. Hồ Chí Minh, tháng 06 năm 2023

Tp. Hồ Chí Minh, ngày 22 tháng 6 năm 2023

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên 1: Lê Anh Quang **MSSV:** 19161154

Họ và tên sinh viên 2: Trần Hữu Nghĩa **MSSV:** 19161141

Ngành: Công nghệ kỹ thuật điện tử - Viễn thông

Lớp: 19161CLĐT2B

Giảng viên hướng dẫn: TS. Nguyễn Thanh Nghĩa

Ngày nhận đề tài: 16/02/2023

Ngày nộp đề tài: 22/06/2023

1. Tên đề tài: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỀU KHIỂN CÁC THIẾT BỊ ĐIỆN TRONG NHÀ SỬ DỤNG CÔNG NGHỆ LORA.

2. Các số liệu, tài liệu ban đầu:

- Các loại vi điều khiển: Arduino Mega 2560, Arduino Uno, Raspberry Pi 3 model B, ModuleWifi ESP8266.
- Các loại module: Lora Ra-02 SX1278, Camera USB.
- Màn hình hiển thị: LCD 16x02.
- Cảm biến: Cảm biến đo nhiệt độ - độ ẩm DHT11, Cảm biến khí gas MQ-2.
- Điều khiển thiết bị: Relay 4 kênh và 8 kênh.
- Nguồn: Nguồn tổ ong, module hạ áp DC – DC.

3. Nội dung thực hiện đề tài:

- Tìm hiểu và tham khảo các tài liệu, giáo trình, nghiên cứu các chủ đề, các nội dung liên quan đến đề tài.
- Tìm hiểu về công nghệ Lora.
- Viết chương trình giao tiếp giữa các module để điều khiển các thiết bị điện, hệ thống cửa tự động.
- Viết chương trình nhận diện.
- Thiết kế ứng dụng điều khiển trên hệ điều hành Android.
- Thi công mô hình.
- Chạy thử nghiệm hệ thống.
- Chỉnh sửa lỗi xuất hiện.
- Đánh giá kết quả thực hiện.
- Viết báo cáo

TRƯỞNG NGÀNH

GIẢNG VIÊN HƯỚNG DẪN

LỜI CAM ĐOAN

Đề tài này là do nhóm em thực hiện dưới sự hướng dẫn của TS. Nguyễn Thanh Nghĩa dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài
Lê Anh Quang – Trần Hữu Nghĩa

LỜI CẢM ƠN

Để hoàn thành đề tài nghiên cứu này, lời đầu tiên cho phép chúng em được gửi lời cảm ơn chân thành đến toàn thể quý thầy cô Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM nói chung và các thầy cô trong Khoa Điện – Điện Tử nói riêng, những người đã tận tình dạy dỗ, trang bị cho chúng em những kiến thức nền tảng và kiến thức chuyên ngành quan trọng, giúp nhóm chúng em có được cơ sở lý thuyết vững vàng và đã luôn tạo điều kiện giúp đỡ tốt nhất cho chúng em trong quá trình học tập và nghiên cứu.

Đặc biệt, chúng em xin chân thành cảm ơn thầy **Nguyễn Thanh Nghĩa** đã tận tình giúp đỡ, đưa ra những định hướng nghiên cứu cũng như hướng giải quyết một số vấn đề để chúng em có thể thực hiện tốt đề tài. Trong thời gian làm việc với thầy, chúng em đã không ngừng tiếp thu thêm nhiều kiến thức được chỉ dạy từ thầy, luôn thể hiện một thái độ nghiên cứu nghiêm túc, hiệu quả và đây cũng là điều rất cần thiết trong quá trình học tập và làm việc sau này đối với chúng em.

Đồ án tốt nghiệp này không chỉ là thành tựu của mỗi cá nhân chúng em mà còn là kết quả của sự ủng hộ, động viên và tình yêu thương vô điều kiện mà gia đình đã dành cho chúng em suốt những năm qua. Cảm ơn gia đình vì luôn là nguồn động lực to lớn để chúng em tiếp tục vươn lên và hoàn thành đồ án tốt nghiệp của mình. Bằng sự quan tâm chân thành, sự hiểu biết và sự hỗ trợ của gia đình và bạn bè, chúng em đã có thể vượt qua những khó khăn, vực dậy trong quá trình thực hiện đồ án tốt nghiệp này. Xin gửi tới gia đình và bạn bè lời cảm ơn chân thành nhất.

Người thực hiện đề tài

Lê Anh Quang – Trần Hữu Nghĩa

TÓM TẮT

Trong thời đại công nghệ phát triển như hiện nay, việc ứng dụng các hệ thống điều khiển từ xa và kết nối không dây đang trở thành xu hướng quan trọng và tiên tiến. Hệ thống Lora (Long Range) đã trở thành một công nghệ giao tiếp không dây phổ biến và hiệu quả, cho phép truyền dữ liệu trên khoảng cách xa mà không cần sử dụng quá nhiều năng lượng. Điều này đã mở ra những cơ hội rộng lớn trong việc ứng dụng Lora vào các lĩnh vực như quản lý năng lượng, môi trường, nông nghiệp thông minh, nhà thông minh và nhiều ứng dụng khác. Qua quá trình nghiên cứu, nhóm chúng em đã tìm hiểu về cơ bản và nguyên lý hoạt động của công nghệ Lora, cùng với việc áp dụng nó vào việc điều khiển các thiết bị từ xa. Đồng thời, nhóm chúng em đã tiến hành xây dựng một ứng dụng thực tế sử dụng Lora để điều khiển các thiết bị điện, nhằm minh chứng tính khả thi và hiệu quả của hệ thống này trong thực tế. Đồ án tốt nghiệp này nhằm mục đích nghiên cứu và phát triển một hệ thống điều khiển một hệ thống điều khiển thiết bị sử dụng công nghệ Lora.

Bằng những ý trên, nhóm chúng em quyết định chọn đề tài: “Thiết kế và thi công hệ thống điều khiển các thiết bị điện trong nhà sử dụng công nghệ Lora”. Ở ĐATN này nhóm em sẽ thiết kế, thi công mô hình nhà 2 tầng với 1 khối điều khiển hệ thống tầng trệt và 1 khối điều khiển hệ thống tầng 2. Hệ thống tầng trệt sẽ xử lý các tín hiệu, truyền nhận dữ liệu sang hệ thống tầng 2 để điều khiển thiết bị, xử lý các thông số môi trường, các thông số này sẽ được giám sát tại App Android. Hệ thống tầng trệt có sử dụng hệ thống nhận diện gương mặt để điều khiển cửa ra vào. Người sử dụng có thể điều khiển thiết bị điện ở khoảng cách xa, ở bất cứ nơi nào có Internet, Wifi, 3G, 4G trên App Android. Các giá trị nhiệt độ, độ ẩm, khí gas và hình ảnh camera của mô hình cũng được cập nhật lên giao diện này. Hệ thống tích hợp hệ thống chuông báo động trong các sự cố như: có khí gas rò rỉ.

Mục lục

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	i
LỜI CAM ĐOAN	iii
LỜI CẢM ƠN	iv
TÓM TẮT	v
Danh mục các từ viết tắt	viii
Danh mục các bảng biểu	ix
Danh mục các hình ảnh	x
Chương 1. GIỚI THIỆU ĐỀ TÀI	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu	2
1.3 Nội dung nghiên cứu	2
1.4 Giới hạn	3
1.5 Bố cục	3
Chương 2. CƠ SỞ LÝ THUYẾT	5
2.1 Công nghệ IoT và hệ thống nhà thông minh	5
2.2 Bảo mật bằng phương pháp nhận diện gương mặt	6
2.3 Giới thiệu phần cứng	8
2.3.1 Arduino Mega 2560	8
2.3.2 Andruino Uno	10
2.3.3 Raspberry Pi 3 Model B	11
2.3.4 Module Wifi ESP8266	12
2.3.5 Module thu phát Lora RA-02 SX1278	13
2.3.6 Module Relay 4 kênh và 8 kênh	14
2.3.7 Module LCD 16x2 và I2C	15
2.3.8 Camera USB 2.0	16
2.3.9 Cảm biến khí gas MQ2	17
2.3.10 Cảm biến nhiệt độ, độ ẩm DHT11	18
2.3.12 Đèn và quạt	19
2.3.13 Bàn phím ma trận 3x4	19
2.3.14 Khối nguồn	20
2.4 Các chuẩn truyền dữ liệu	23
2.4.1 Chuẩn truyền thông UART	23
2.4.2 Chuẩn truyền thông SPI	25

2.4.3 Chuẩn giao tiếp I2C.....	26
Chương 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....	29
3.1 Giới thiệu	29
3.2 Thiết kế hệ thống	29
3.2.1 Thiết kế sơ đồ khối hệ thống	29
3.2.2 Thiết kế mạch – Tầng trệt.....	31
3.2.3 Thiết kế mạch – Tầng 2.....	36
3.3 Sơ đồ nguyên lý toàn bộ hệ thống	38
Chương 4. THI CÔNG HỆ THỐNG.....	41
4.1 Giới thiệu	41
4.2 Thi công hệ thống	41
4.2.1 Mô hình hệ thống tầng 1:	41
4.2.2 Mô hình hệ thống tầng 2.....	43
4.2.3 Khối nguồn	44
4.3 Đóng gói và thi công mô hình	44
4.4 Lập trình hệ thống	45
4.4.1 Lưu đồ giải thuật	45
4.4.2 Thiết kế giao diện trên App Android	51
4.5 Hướng dẫn sử dụng	63
Chương 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ	64
5.1 Giới thiệu	64
5.2 Kết quả đạt được.....	64
5.3 Nhận xét – Đánh giá	65
Chương 6. KẾT LUẬN - HƯỚNG PHÁT TRIỂN	70
6.1 Kết luận:	70
6.2 Hướng phát triển:.....	70
TÀI LIỆU THAM KHẢO	71
PHỤ LỤC	72

Danh mục các từ viết tắt

Tên viết tắt	Tên đầy đủ
ĐATN	Đồ án tốt nghiệp.
GPIO	General-purpose input/output.
ADC	Analog-to-Digital Converter.
DAC	Digital-to-Analog Converter.
UART	Universal Asynchronous Reveiver and Transmitter.
SPI	Serial Peripheral Interface Bus.
I2C	Inter-Intergrated Circuit.
USB	Universal Serial Bus.
IP	Internet Protcol.
IoT	Internet of Things.

Danh mục các bảng biểu

Bảng 4.1: Số liệu dòng và áp của các thiết bị tại tầng 1	42
Bảng 4.2: Số liệu dòng và áp của các thiết bị tại tầng 2	44
Bảng 5.1: Bảng đánh giá các thiết bị điện của hệ thống	68
Bảng 5.2: Bảng đánh giá số liệu thực hiện của hệ thống cửa.....	68

Danh mục các hình ảnh

Hình 2.1: Hệ thống nhà thông minh	5
Hình 2.2: Hệ thống nhận diện khuôn mặt.....	7
Hình 2.3: Arduino Mega 2560.....	9
Hình 2.4: Arduino Uno R3	10
Hình 2.5: Raspberry Pi 3 Model B	11
Hình 2.6: Module ESP8266 NodeMCU	12
Hình 2.7: Lora RA-02 SX1278	14
Hình 2.8: Module relay 4 kênh và 8 kênh	14
Hình 2.9: LCD 16x2	15
Hình 2.10: Module chuyển đổi I2C cho LCD 16x2	16
Hình 2.11: Webcam USB 2.0	16
Hình 2.12: Các chân của cảm biến MQ-2	17
Hình 2.13: Các chân của DHT11	18
Hình 2.14: Module buzzer	18
Hình 2.15: Led thanh 12V	19
Hình 2.16: Quạt tản nhiệt 12V	19
Hình 2.17: Bàn phím ma trận 3x4	19
Hình 2.18: Nguồn tổ ong và mạch hạ áp	20
Hình 2.19: Sơ đồ khói Nguồn tổ ong (nguồn xung)	21
Hình 2.20: Cấu tạo cơ bản một bộ nguồn tổ ong.....	21
Hình 2.21: Gói dữ liệu truyền của UART	23
Hình 2.22: Sóng truyền UART	24
Hình 2.23: Quá trình truyền UART	24
Hình 2.24: Quá trình nhận UART	24
Hình 2.25: Truyền dữ liệu SPI.....	26
Hình 2.26: Bus I2C và các thiết bị ngoại vi	26
Hình 2.27: Trình tự truyền bit trên đường truyền.....	27
Hình 3.1: Sơ đồ khói của hệ thống	29
Hình 3.2: Kết nối chân Arduino Mega 2560	31
Hình 3.3: Kết nối chân module Lora Ra-02	32
Hình 3.4: Kết nối chân ESP8266.....	32
Hình 3.5: Kết nối chân Raspberry Pi.....	33
Hình 3.6: Cảm biến nhiệt độ-độ ẩm DHT11 và cảm biến khí gas MQ2	33
Hình 3.7: Còi báo buzzer và nút nhấn reset.....	34
Hình 3.8: Màn hình hiển thị LCD	34
Hình 3.9: Sơ đồ nối dây các thiết bị ở tầng 1	35
Hình 3.10: Kết nối chân bàn phím ma trận 3x4	36
Hình 3.11: Kết nối chân Arduino Uno	36
Hình 3.12: Kết nối chân module Lora Ra-02	37
Hình 3.13: Sơ đồ nối dây các thiết bị ở tầng 2	37
Hình 3.14: Sơ đồ nguyên lý tầng 1	38
Hình 3.15: Sơ đồ nguyên lý tầng 2	39

Hình 4.1: Mặt trên của mô hình hệ thống tầng 1	41
Hình 4.2: Mặt dưới của mô hình hệ thống tầng 1	42
Hình 4.3: Mặt trên của mô hình hệ thống tầng 2	43
Hình 4.4: Mặt dưới của mô hình hệ thống tầng 2	43
Hình 4.5: Kiến trúc mô hình	44
Hình 4.6: Lưu đồ điều khiển chính tầng 1	45
Hình 4.11: Lưu đồ điều khiển ESP8266	47
Hình 4.13: Lưu đồ điều khiển hệ thống tầng 2	48
Hình 4.14: Lưu đồ xây dựng mô hình nhận diện khuôn mặt	49
Hình 4.15: Lưu đồ nhận diện khuôn mặt	50
Hình 4.16: Giao diện chính của phần mềm App Inventor	52
Hình 4.17: Khu vực Viewer	53
Hình 4.18: Khu vực Components	54
Hình 4.19: Khu vực Properties	55
Hình 4.20: Phần Block của khu vực lập trình	56
Hình 4.21: Phần Viewer của khu vực lập trình	56
Hình 4.22: Lựa chọn Android App (.apk) trong phần Build	57
Hình 4.23: Chọn cách tải về theo 2 cách	57
Hình 4.24: Giao diện home (Screen1)	58
Hình 4.25: Code của giao diện home	59
Hình 4.26: Giao diện tầng 1	60
Hình 4.27: Code giao diện tầng 1	60
Hình 4.28: Giao diện tầng 2	61
Hình 4.29: Code giao diện tầng 2	61
Hình 4.30: Lưu đồ giải thuật App Android	62
Hình 4.31: Mã QR để tải app	63
 Hình 5.1: Điều khiển đèn quạt tại tầng trệt	65
Hình 5.2: Điều khiển đèn quạt tại tầng 2	66
Hình 5.3: Mở cửa bằng cách nhập mật khẩu	66
Hình 5.4: Mở cửa bằng nhận diện khuôn mặt	67
Hình 5.5: Giá trị nhiệt độ - độ ẩm được hiển thị lên điện thoại	68

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Ngày nay với sự phát triển không ngừng của khoa học và công nghệ với những ứng dụng của khoa học kĩ thuật tiên tiến, thế giới chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Sự phát triển của kĩ thuật điện tử đã tạo ra hàng loạt những thiết bị với đặc điểm nổi bật như sự chính xác cao, tốc độ nhanh, gọn nhẹ là những yếu tố rất cần thiết cho hoạt động của con người đạt hiệu quả cao. Một trong những ứng dụng quan trọng trong công nghệ điện tử là kỹ thuật điều khiển từ xa. Nó đã góp phần rất lớn trong việc điều khiển các thiết bị từ xa nhằm đáp ứng nhu cầu của con người, giúp tiết kiệm được thời gian và quản lý dễ dàng.

Bắt nguồn từ những nhu cầu cần thiết đó và lấy cảm hứng từ các dự án như: “Thiết kế và thi công hệ thống giám sát và điều khiển thiết bị trong nhà kết hợp Wifi và Lora”[1], “Thi công hệ thống điều khiển giám sát thiết bị trong nhà sử dụng công nghệ Lora”[2], các đề tài này đã điều khiển các thiết bị trong nhà thông qua công nghệ Lora hay Wifi nhưng cũng chỉ dừng lại ở một vài thiết bị điện. Vì thế nhóm em đã quyết định chọn đề tài **“Thiết kế và thi công hệ thống điều khiển các thiết bị điện trong nhà sử dụng công nghệ Lora”** cải thiện thêm chức năng điều khiển được nhiều thiết bị thông qua chuẩn truyền không dây Lora, giám sát các cảm biến và camera nhận diện gương mặt để mở khóa cửa thêm hệ thống nhập mật khẩu để mở cửa.

Nội dung chính của đề tài:

- Sử dụng Arduino Mega 2560 làm board trung tâm xử lý và điều khiển các thiết bị tại tầng trệt.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

- Sử dụng Arduino Uno làm board trung tâm xử lý và điều khiển các thiết bị tại tầng 2.
- Sử dụng Raspberry Pi 3 model B xử lý, nhận diện khuôn mặt qua camera để đóng mở cửa tự động,
- Điều khiển các thiết bị bằng app Android kết hợp WiFi và sóng Lora.
- Giám sát trạng thái hoạt động, điều khiển các thiết bị, giá trị cảm biến qua app Android và LCD hiển thị tình trạng của cửa ra vào .
- Cảnh báo khi các giá trị cảm biến vượt ngưỡng cho phép.

1.2 Mục tiêu

Đề tài: “Thiết kế và thi công hệ thống điều khiển các thiết bị điện trong nhà sử dụng công nghệ Lora” bao gồm các vấn đề chính sau:

- Thiết kế và thi công mạch giám sát và điều khiển các thiết bị qua app Android ở 2 tầng.
- Thiết kế và thi công hệ thống cửa ra vào có bảo mật và nhận diện gương mặt để mở cửa tự động.
- Thi công mô hình nhà 2 tầng và chạy thử nghiệm.

1.3 Nội dung nghiên cứu

- Tìm hiểu và tham khảo các tài liệu, giáo trình, nghiên cứu các chủ đề, các nội dung liên quan đến đề tài.
- Thiết kế sơ đồ khối và sơ đồ nguyên lý cho hệ thống.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

- Thiết kế và thi công mạch xử lý điều khiển các thiết bị ở tầng trệt như: servo, cảm biến nhiệt độ - độ ẩm, cảm biến khí gas, đèn, quạt và module Lora.
- Thiết kế và thi công mạch xử lý giao tiếp giữa Raspberry Pi 3 model B, camera nhận diện gương mặt để đóng mở cửa tự động và LCD hiển thị tình trạng cửa đóng mở. Có bảo mật thêm bằng nhập mật khẩu.
- Thiết kế và thi công mạch xử lý điều khiển các thiết bị ở tầng 2 như: đèn, quạt và module Lora.
- Thiết kế app Android giám sát và điều khiển các thiết bị, hiển thị giá trị cảm biến.
- Viết các chương trình để giao tiếp giữa ESP8266, Raspberry Pi, Adruino Uno, Adruino Mega và Module Lora.

1.4 Giới hạn

- Mô hình hệ thống gồm 2 tầng nên số lượng node chỉ có 2
- Tầng 1 sử dụng 3 quạt, 4 đèn và cảm biến nhiệt độ- độ ẩm, khí gas
- Tầng 2 sử dụng 2 đèn và 2 quạt.
- Hệ thống mở cửa tự động gồm có camera, màn hình LCD, bàn phím và servo để đóng mở cửa.
- Khoảng cách tối đa giữa 2 tầng để hệ thống có thể hoạt động là 5 km đến 15 km nhưng trong đề tài đồ án này nhóm chúng em làm mô hình nhà 2 tầng nên khoảng cách ngắn.

1.5 Bô cục

- **Chương 1: Giới thiệu đề tài**

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Chương này trình bày đặt vấn đề dãnh nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bối cảnh đồ án.

- **Chương 2: Cơ sở lý thuyết**

Chương này trình bày về các lý thuyết có liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài.

- **Chương 3: Thiết Kế và Tính Toán**

Chương này giới thiệu tổng quan về các yêu cầu của đề tài mà mình thiết kế và các tính toán, thiết kế gồm những phần nào.

- **Chương 4: Thi công hệ thống**

Trình bày các mạch đã thiết kế và trình tự lắp ráp thi công mô hình, hướng dẫn cách cài đặt, sử dụng các phần mềm lập trình và trình bày các thao tác hướng dẫn sử dụng hệ thống.

- **Chương 5: Kết Luận, Nhận xét và Đánh giá**

Chương này đưa ra nhận xét và đánh giá sản phẩm mô hình đã hoàn thành.

- **Chương 6: Kết Luận và Hướng phát triển**

Chương này trình bày về những kết quả đã được mục tiêu đề ra sau quá trình nghiên cứu thi công. Từ những kết quả đạt được, những hạn chế, từ đó rút ra kết luận và hướng phát triển để giải quyết các vấn đề tồn đọng đề án hoàn thiện hơn.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1 Công nghệ IoT và hệ thống nhà thông minh

Internet of Things (IoT) đang ngày càng trở nên phổ biến và ảnh hưởng đến đời sống của con người một cách đáng kể. IoT cho phép các thiết bị được kết nối với nhau thông qua mạng internet, giúp cho việc quản lý và điều khiển các hoạt động trong nhà thông minh, văn phòng, công trình xây dựng, hay thậm chí cả các thành phố trở nên dễ dàng hơn. Bằng cách sử dụng các cảm biến, IoT cũng có thể giúp cho con người theo dõi sức khỏe của mình, phát hiện sớm các vấn đề về sức khỏe và tự động điều khiển các thiết bị trong nhà để tăng tính an toàn cho người dùng. Với sự tiến bộ của công nghệ, IoT đang có những ứng dụng tiềm năng trong nhiều lĩnh vực khác nhau, từ sản xuất, nông nghiệp đến giải trí và du lịch, và chắc chắn sẽ mang lại nhiều lợi ích cho đời sống của con người trong tương lai.

Những ứng dụng của IoT trong lĩnh vực nhà thông minh đã mang đến cho con người những tiện ích và trải nghiệm sống tiện lợi hơn. Nhờ vào những thiết bị kết nối mạng thông minh như đèn, máy lạnh, máy giặt, bếp điện và cả cửa ra vào, việc điều khiển và quản lý các thiết bị trong nhà trở nên dễ dàng hơn bao giờ hết.



Hình 2.1: Hệ thống nhà thông minh

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Với nhà thông minh, người dùng có thể điều khiển và giám sát các thiết bị trong nhà thông qua một ứng dụng trên điện thoại. Ví dụ, khi rời khỏi nhà, người dùng có thể tắt đèn, tắt máy lạnh hoặc kiểm tra lại xem cửa đã khóa chưa. Nếu có khách ghé thăm, người dùng có thể mở cửa từ xa cho họ vào mà không cần phải đến cửa.

Để sử dụng sản phẩm nhà thông minh một cách hiệu quả, người dùng cần có kiến thức và kỹ năng để cài đặt và sử dụng các thiết bị liên quan đến IoT. Các nhà sản xuất cần đưa ra các giải pháp tối ưu hóa để giúp người dùng dễ dàng tiếp cận và sử dụng sản phẩm nhà thông minh một cách tiện lợi và an toàn.

Việc phát triển sản phẩm nhà thông minh cũng đặt ra nhiều thách thức cho các nhà sản xuất, như bảo mật thông tin và đảm bảo tính ổn định của hệ thống. Do đó, các nhà sản xuất cần phải đầu tư vào nghiên cứu và phát triển công nghệ, cũng như kiểm tra và thử nghiệm sản phẩm trước khi đưa ra thị trường.

Trong tương lai, sản phẩm nhà thông minh sẽ ngày càng phổ biến và trở thành một phần không thể thiếu trong cuộc sống của con người. Đó sẽ là sự kết hợp hoàn hảo giữa công nghệ và cuộc sống, giúp con người tiết kiệm thời gian, công sức và tạo ra một môi trường sống tiện nghi và an toàn hơn.

2.2 Bảo mật bằng phương pháp nhận diện gương mặt

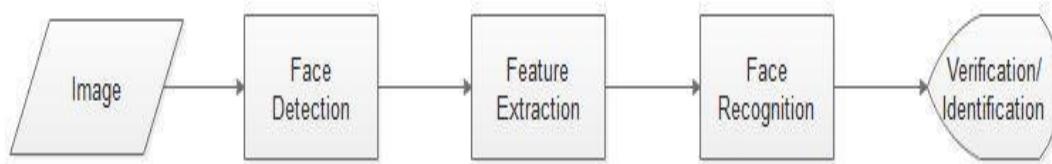
Nhận dạng khuôn mặt (Face Recognition)[3] là một phương pháp sinh trắc để xác định hoặc xác minh một cá nhân nào đó bằng cách so sánh dữ liệu hình ảnh chụp trực tiếp hoặc hình ảnh kỹ thuật số với bản ghi được lưu trữ cho người đó. Nó được xem là một lĩnh vực nghiên cứu của ngành Biometrics (tương tự như nhận dạng vân tay – Fingerprint Recognition, hay nhận dạng mống mắt – Iris Recognition).

So với nhận dạng vân tay và mống mắt, nhận dạng khuôn mặt có nguồn dữ liệu phong phú hơn (chúng ta có thể nhìn thấy mặt người ở bất cứ tấm ảnh, video clip nào liên quan tới con người trên mạng) và ít đòi hỏi sự tương tác có kiểm soát hơn (để thực hiện nhận dạng vân tay hay mống mắt, dữ liệu input lấy từ con người đòi hỏi có sự hợp tác trong môi trường có kiểm soát) [3].

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Các hệ thống nhận dạng khuôn mặt thường được sử dụng cho các mục đích an ninh như kiểm soát an ninh tại tòa nhà, sân bay, máy ATM, tra cứu thông tin của tội phạm, phát hiện tội phạm ở nơi công cộng, ...và ngày càng được ứng dụng rộng rãi trong cuộc sống [3].

Để xây dựng một hệ thống nhận dạng khuôn mặt có đầu vào của hệ thống là một hình ảnh kỹ thuật số hay một khung hình video từ một nguồn video. Đầu ra là xác định hoặc xác minh người ở trong bức hình hoặc trong video đó là ai. Hướng tới mục tiêu này chúng ta thường chia thủ tục nhận dạng khuôn mặt gồm ba bước: Phát hiện khuôn mặt, trích rút đặc trưng và nhận dạng khuôn mặt [3].



Hình 2.2: Hệ thống nhận diện khuôn mặt

Phát hiện khuôn mặt (Face Detection): Chức năng chính của bước này là phát hiện ra khuôn mặt xem nó có xuất hiện ở trong một bức hình hay một đoạn video hay không? Tỉ lệ phát hiện ra khuôn mặt phụ thuộc nhiều vào điều kiện về độ sáng, hướng khuôn mặt, biểu hiện cảm xúc trên khuôn mặt hay các yếu tố môi trường khác. Để hệ thống nhận dạng hoạt động đạt hiệu quả cao thì hình ảnh khuôn mặt sau khi được phát hiện cần chuẩn hóa về kích thước, ánh sáng [3].

Trích rút đặc trưng (Feature Extraction): Sau khi phát hiện ra khuôn mặt trong bức ảnh, chúng ta tiến hành trích rút những đặc trưng của khuôn mặt. Bước này trích xuất ra một vector đặc trưng đại diện cho một khuôn mặt. Nó phải đảm bảo được tính duy nhất của một khuôn mặt [3].

Nhận dạng khuôn mặt (Face Recognition): Với hình ảnh đầu vào sau khi phát hiện ra khuôn mặt, trích rút các đặc trưng của khuôn mặt và đem so sánh các đặc trưng này với cơ sở dữ liệu khuôn mặt [3].

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Nhận dạng khuôn mặt được ứng dụng nhiều trong các lĩnh vực đòi hỏi đặc biệt ở những lĩnh vực công nghệ cao, yêu cầu về an ninh, bảo mật. Do đó để hệ thống nhận dạng khuôn mặt hoạt động mạnh mẽ với tốc độ và độ tin cậy thì có rất nhiều các phương pháp về nhận dạng khuôn mặt được đưa ra. Các phương pháp có thể được phân loại theo các tiêu chí khác nhau như nhận dạng với dữ liệu ảnh đầu vào là ảnh tĩnh 2D (Elastic Bunch Graph, Active Appearance Model). Phương pháp này là phổ biến nhất và tương lai sẽ là 3D (3D Morphable Model) [3]. Tuy nhiên trên thực tế người ta hay chia phương pháp nhận dạng khuôn mặt ra thành 2 loại:

- Nhận dạng dựa trên các đặc trưng của các phần tử trên khuôn mặt (Feature Base Face Recognition).
- Nhận dạng dựa trên xét tổng thể toàn khuôn mặt (Appearance Based Face Recognition).

2.3 Giới thiệu phần cứng

2.3.1 Arduino Mega 2560

Arduino Mega 2560 là sản phẩm tiêu biểu cho dòng mạch Mega, là dòng bo mạch có nhiều cải tiến so với Arduino Uno (54 chân digital IO và 16 chân analog IO). Đặc biệt bộ nhớ flash của MEGA được tăng lên một cách đáng kể, gấp 4 lần so với những phiên bản cũ của UNO R3. Arduino Mega 2560 có khả năng điều khiển một số lượng lớn các thiết bị ngoại vi và các cảm biến khác nhau. Nó cũng có các tính năng như cổng USB, các chân giao tiếp UART, I2C, SPI.

Arduino Mega 2560 được sử dụng rộng rãi trong các ứng dụng nhúng và điện tử, bao gồm các hệ thống kiểm soát và tự động hóa, hệ thống đo lường, các dự án robot và các ứng dụng IoT. Với khả năng mở rộng thông qua các mô-đun ngoại vi, các mạch phụ trợ và các shield Arduino, Arduino Mega 2560 cho phép người dùng tùy chỉnh và mở rộng các ứng dụng của mình một cách dễ dàng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.3: Arduino Mega 2560

Thông số kỹ thuật của Arduino Mega 2560:

- Mạch Arduino Mega 2560 có 70 chân Digital: Chân số: Từ 0-53 (digital) và 0-15 (analog) có thể được sử dụng làm đầu vào hoặc đầu ra cho thiết bị được thiết lập bằng các hàm Mode (), digitalWrite (), digitalRead (). Giá trị điện áp trên mỗi chân là 5V, dòng trên mỗi chân là 20mA và bên trong có điện trở kéo lên là 20-50 ohm. Dòng tối đa trên mỗi chân I/O không vượt quá 40mA để tránh trường hợp gây hỏng board mạch.
- Arduino mega2560 có 16 chân analog (A0 → A15) cung cấp độ phân giải tín hiệu 10bit ($0 \rightarrow 2^{10}-1$) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10 bit.
- Có 4 cặp chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này.
- Có 15 Chân PWM (~): 3, 5, 6, 9, 10,....: cho phép bạn xuất ra xung PWM với độ phân giải 8 bit bằng hàm analogWrite() và có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Chân giao tiếp SPI: 22 (SS), 24 (MOSI), 25 (MISO), 23 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
- Chân giao tiếp I2C: Chân 20 cho SDA và 21 cho SCK (Tốc độ 400khz) để cho phép liên lạc hai dây với các thiết bị khác. Hàm được sử dụng là wire.begin () để bắt đầu chuyển đổi I2C, với wire.Read () để đọc dữ liệu I2C và wire.Write () để ghi dữ liệu I2C.
- LED 13: trên Arduino mega2560 có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.
- Có 6 chân ngắt ngoài: từ 0 đến 5 tương ứng với các chân (2, 3, 21, 20, 19, 18) được ký hiệu là INT.

2.3.2 Andruino Uno

Arduino Uno R3 là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328 được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân đầu vào/ đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau. Mạch Arduino Uno thích hợp cho những bạn mới tiếp cận và đam mê về điện tử, lập trình... Dựa trên nền tảng mở do Arduino.cc cung cấp các bạn dễ dàng xây dựng cho mình một dự án nhanh nhất (lập trình Robot, xe tự hành, điều khiển bật tắt led...).



Hình 2.4: Arduino Uno R3

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật của Arduino Uno:

- Mạch Arduino UNO R3 với thiết kế tiêu chuẩn sử dụng vi điều khiển Atmega328.
- Mạch Arduino UNO R3 có 14 chân Digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).
- Arduino UNO Broad có 6 chân Analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 1023) để đọc giá trị điện áp trong khoảng 0V → 5V. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân Analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10 bit. Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

2.3.3 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz – nhanh nhất từ trước đến nay! Phiên bản mới còn hỗ trợ WiFi Dual-band 2.4GHz và 5GHz, Bluetooth 4.2/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.



Hình 2.5: Raspberry Pi 3 Model B

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Raspberry Pi 3 Model B được sử dụng rộng rãi trong các ứng dụng như máy tính để bàn nhỏ gọn, thiết bị định tuyến, máy tính chủ, máy chủ đa phương tiện, trò chơi điện tử và hệ thống giám sát, là một công cụ hữu ích để giúp tạo ra các dự án IoT (Internet of Things) và các dự án khác trong lĩnh vực công nghệ. Vì vậy trong đề tài nhóm chúng em sẽ sử dụng Raspberry Pi 3 để xử lý nhận dạng khuôn mặt.

2.3.4 Module Wifi ESP8266

ESP8266 NodeMCU là module được sử dụng phổ biến rất nhiều trong lĩnh vực IOT, phát triển dựa trên nền chip Wifi SoC ESP8266 với thiết kế dễ dàng sửa dụng vì tích hợp sẵn mạch nạp sử dụng chip CP2102 trên board. Bên trong ESP8266 có sẵn một lõi vi xử lý vì thế bạn có thể trực tiếp lập trình cho ESP8266 mà không cần thêm bất kì con vi xử lý nào nữa. Hiện tại có hai ngôn ngữ có thể lập trình cho ESP8266, sử dụng trực tiếp phần mềm IDE của Arduino để lập trình với bộ thư viện riêng hoặc sử dụng phần mềm node MCU.

Node MCU ESP8266 Lua CP2102 sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window và Linux, đây là phiên bản nâng cấp từ các phiên bản sử dụng IC nạp giá rẻ CH340.



Hình 2.6: Module ESP8266 NodeMCU

NodeMCU có tổng cộng 13 chân GPIO tuy nhiên một số chân được dùng cho những mục đích quan trọng khác vì vậy chúng ta phải lưu ý khi sử dụng như sau:

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Tất cả các GPIO đều có trở kéo lên nguồn bên trong (ngoại trừ GPIO16 có trở kéo xuống GND). Người dùng có thể cấu hình kích hoạt hoặc không kích hoạt trở kéo này.
- GPIO1 và GPIO3: hai GPIO này được nối với TX và RX của bộ UART0, NodeMCU nạp code thông qua bộ UART này nên tránh sử dụng 2 chân GPIO này.
- GPIO0, GPIO2, GPIO15: đây là các chân có nhiệm vụ cấu hình mode cho ESP8266 điều khiển quá trình nạp code nên bên trong NodeMCU (có tên gọi là strapping pins) có các trở kéo để định sẵn mức logic cho chúng như sau: GPIO0: HIGH, GPIO2: HIGH, GPIO15: LOW. Vì vậy khi muốn sử dụng các chân này ở vai trò GPIO cần phải thiết kế một nguyên lý riêng để tránh xung đột đến quá trình nạp code. Các bạn có thể tham khảo nguyên lý thiết kế mạch
- GPIO9, GPIO10: hai chân này được dùng để giao tiếp với External Flash của ESP8266 vì vậy cũng không thể dùng được (đã test thực nghiệm).

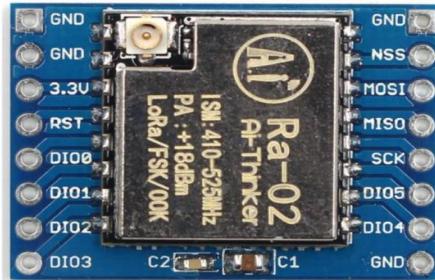
2.3.5 Module thu phát Lora RA-02 SX1278

Mạch thu phát RF SPI Lora SX1278 433Mhz Ra-02 Ai-Thinker được sản xuất bởi Ai-Thinker sử dụng chip SX1278 của nhà sản xuất SEMTECH chuẩn giao tiếp LORA (Long Range), chuẩn LORA mang đến hai yếu tố quan trọng là tiết kiệm năng lượng và khoảng cách phát siêu xa (Ultimate long range wireless solution), ngoài ra nó còn có khả năng cấu hình để tạo thành mạng truyền nhận nên hiện tại được phát triển và sử dụng rất nhiều trong các nghiên cứu về IoT.

Mạch thu phát RF SPI Lora SX1278 433Mhz Ra-02 Ai-Thinker có thiết kế nhỏ gọn dạng module giúp dễ dàng tích hợp trong các thiết kế mạch, mạch được thiết kế và đo đạc chuẩn để có thể đạt công suất và khoảng cách truyền

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

xa nhất, ngoài ra mạch còn có chất lượng linh kiện và gia công tốt cho nên có độ bền cao và khả năng hoạt động ổn định.



Hình 2.7: Lora RA-02 SX1278

2.3.6 Module Relay 4 kênh và 8 kênh

Relay 4 Kênh/ 8 Kênh gồm 4 relay/ 8 relay hoạt động tại điện áp 5VDC, chịu được hiệu điện thế lên đến 250VAC 10A. Relay 4 kênh/ 8 kênh được thiết kế chắc chắn, khả năng cách điện tốt. Trên module đã có sẵn mạch kích relay sử dụng transistor và IC cách ly quang giúp cách ly hoàn toàn mạch điều khiển (vi điều khiển) với relay bảo đảm vi điều khiển hoạt động ổn định. Có sẵn header rất tiện dụng khi kết nối với vi điều khiển.

Mạch relay 4 kênh/ 8 kênh sử dụng chân kích mức Thấp (0V), khi có tín hiệu 0V vào chân IN thì relay sẽ nhảy qua thường Hở của Relay. Ứng dụng với relay module khá nhiều bao gồm cả điện DC hay AC. Và trong đề tài này nhóm chúng em sử dụng module relay để bật tắt đèn quạt.

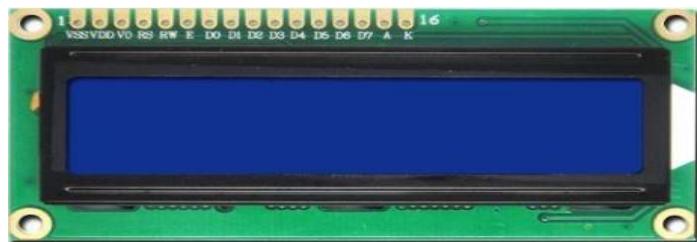


Hình 2.8: Module relay 4 kênh và 8 kênh

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.3.7 Module LCD 16x2 và I2C

Ngày nay, LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của vi điều khiển. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), 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 rẻ.



Hình 2.9: LCD 16x2

Trong 16 chân của LCD được chia ra làm 3 dạng tín hiệu như sau:

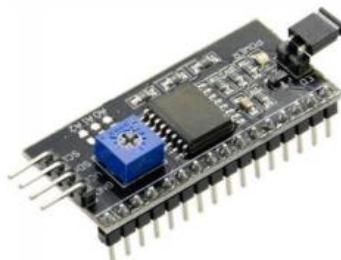
- Các chân cấp nguồn: Chân số 1 là chân nối mass (0V), chân thứ 2 là VDD nối với nguồn +5V. Chân thứ 3 dùng để chỉnh độ tương phản thường nối với biến trở.
- Các chân điều khiển: Chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
- Các chân dữ liệu D7-D0: Chân số 7 đến chân số 14 là 8 chân dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.

Để giao tiếp LCD thì vi điều khiển cần tối thiểu 6 chân là RS, EN, D7 – D4. Điều này sẽ gây khó khăn cho ứng dụng vì nó tốn chân của vi điều khiển. Trong khi đó, nếu sử dụng module chuyển đổi I2C thì sẽ tiết kiệm được rất nhiều chân cho vi điều khiển. Module cần 2 chân cấp nguồn và 2 chân giao tiếp truyền nhận dữ liệu

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

SCD và SDA. Ngoài ra có thể điều chỉnh được độ tương phản bởi biến trở gắn trên module thay vì gắn thêm biến trở ngoài nếu điều khiển trực tiếp.

Từ những phân tích trên, việc sử dụng module I2C cho LCD là điều cần thiết vì nó tiết kiệm chân điều khiển trên vi điều khiển, mở rộng khả năng điều khiển hoặc giao tiếp với các thiết bị khác.



Hình 2.10: Module chuyển đổi I2C cho LCD 16x2

Thông số kỹ thuật:

- Điện áp hoạt động: 2.5 – 6 VDC
- Hỗ trợ màn hình: LCD1602, 1604, 2004 (Driver HD44780)
- Giao tiếp: I2C
- Địa chỉ mặc định: 0X27 (có thể thay đổi địa chỉ bằng ba chân A0/A1/A2)
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

2.3.8 Camera USB 2.0

Webcam USB 2.0 là một thiết bị ngoại vi sử dụng giao tiếp USB 2.0 để kết nối với máy tính hoặc laptop. Nó được sử dụng để chụp ảnh và quay video, cung cấp hình ảnh trực tiếp hoặc truyền trực tuyến qua Internet.



Hình 2.11: Webcam USB 2.0

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

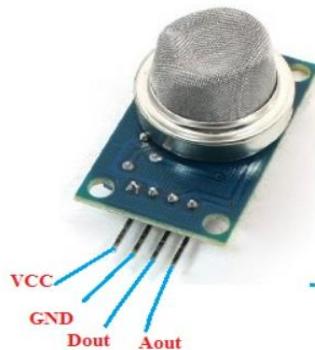
Webcam USB 2.0 thường có độ phân giải khá, với khả năng chụp ảnh và quay video ở độ nét đủ để sử dụng trong các cuộc gọi video, hội thảo trực tuyến, ghi lại video YouTube, livestream và các hoạt động khác. Nó thường được sử dụng cho mục đích cá nhân hoặc công việc từ xa.

Một số tính năng phổ biến của webcam USB 2.0 bao gồm khả năng xoay, nghiêng, thu phóng và tự động lấy nét để điều chỉnh góc nhìn. Ngoài ra, nó có thể đi kèm với microphone tích hợp để ghi âm âm thanh trong quá trình ghi video hoặc thực hiện cuộc gọi.

Tuy nhiên, webcam USB 2.0 có tốc độ truyền dữ liệu hạn chế so với các phiên bản giao tiếp USB mới hơn như USB 3.0 hoặc USB 3.1. Điều này có thể ảnh hưởng đến chất lượng hình ảnh và khả năng truyền trực tuyến của nó.

2.3.9 Cảm biến khí gas MQ2

Cảm biến MQ-2 có thể phát hiện khí Gas tập trung tại những nơi khác nhau từ 10-1000ppm. Cảm biến này với độ nhạy cao thời gian đáp ứng nhanh, tín hiệu ngõ ra dạng Analog và Digital. Cảm biến có thể hoạt động từ khoảng -20°C đến 55°C và tiêu thụ dòng khoảng 180mA tại 5V.



Hình 2.12: Các chân của cảm biến MQ-2

MQ2 là cảm biến khí, dùng để phát hiện các khí có thể gây cháy. Nó được cấu tạo từ chất bán dẫn SnO₂. Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất gây cháy, độ dẫn điện của nó thay đổi. Khi môi trường

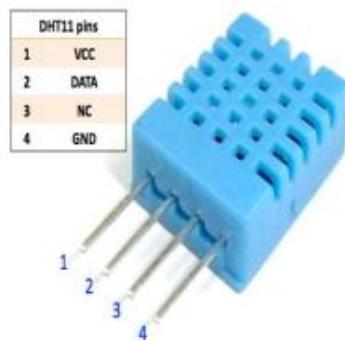
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quanh MQ-2 càng cao.

Các loại khí mà MQ-2 có thể phát hiện được: LPG (Khí hóa lỏng), i-butane, propane, methane, Alcohol, khí gas, khói.

2.3.10 Cảm biến nhiệt độ, độ ẩm DHT11

DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.



Hình 2.13: Các chân của DHT11

2.3.11 Module buzzer

Mạch còi buzzer được sử dụng để phát ra âm thanh khi kích tín hiệu, ứng dụng trong các hệ thống báo hiệu, báo trộm,...



Hình 2.14: Module buzzer

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.3.12 Đèn và quạt

Ở đề tài này nhóm chúng em sử dụng led thanh 12V và quạt 12V cho hệ thống vì cần đèn sáng và quạt quay mạnh.



Hình 2.15: Led thanh 12V



Hình 2.16: Quạt tản nhiệt 12V

2.3.13 Bàn phím ma trận 3x4

Nhóm sẽ sử dụng bàn phím ma trận 3x4 dùng để nhập mật khẩu mở cửa.



Hình 2.17: Bàn phím ma trận 3x4

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.3.14 Khối nguồn

Do các linh kiện đều sử dụng nguồn 12V và 5V nên nhóm em đã quyết định sử dụng bộ nguồn tổ ong chuyển 220VAC từ nguồn điện dân dụng sang 12VDC. Sau đó kết nối qua mạch nguồn hạ áp LM2596 để hạ xuống còn 5VDC cho một số linh kiện sử dụng.

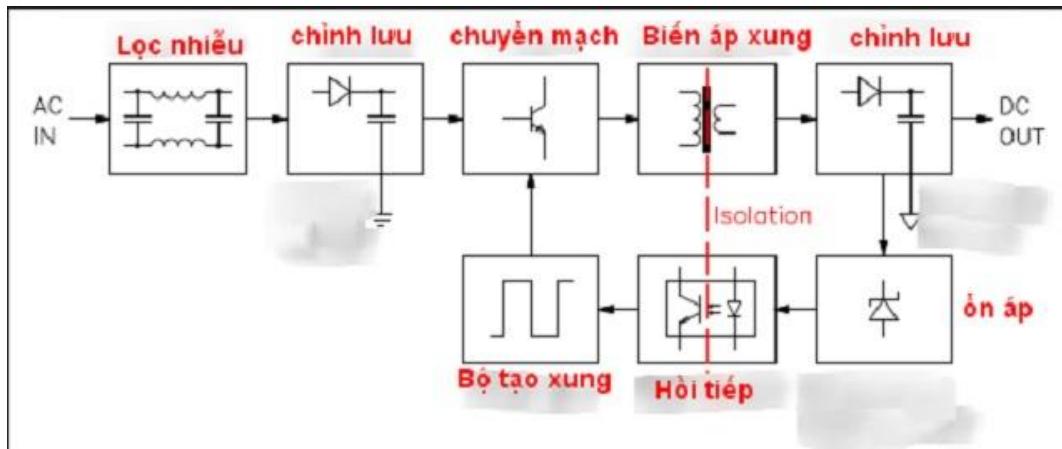


Hình 2.18: Nguồn tổ ong và mạch hạ áp

Nguồn tổ ong là cách gọi khác của nguồn xung. Nguồn xung là bộ nguồn có tác dụng biến đổi từ nguồn điện xoay chiều sang nguồn điện một chiều bằng chế độ dao động xung tạo bằng mạch điện tử kết hợp với một biến áp xung. Nguồn tuyển tính có điện sử dụng biến áp sắt từ để làm nhiệm vụ hạ áp rồi sau đó dùng chỉnh lưu kết hợp với ic nguồn tuyển tính tạo ra các cấp điện áp một chiều theo yêu cầu như: 3.3V, 5V, 6V...

Nguồn tuyển tích thường rất nặng, cồng kềnh và tốn vật liệu nên không còn được sử dụng nhiều trong một vài trường hợp (vì nguồn tuyển tính có một vài ưu điểm tốt mà nguồn xung không bao giờ đạt được, nguồn xung luôn tạo ra sóng hài làm ảnh hưởng xấu tới chất lượng lưới điện)

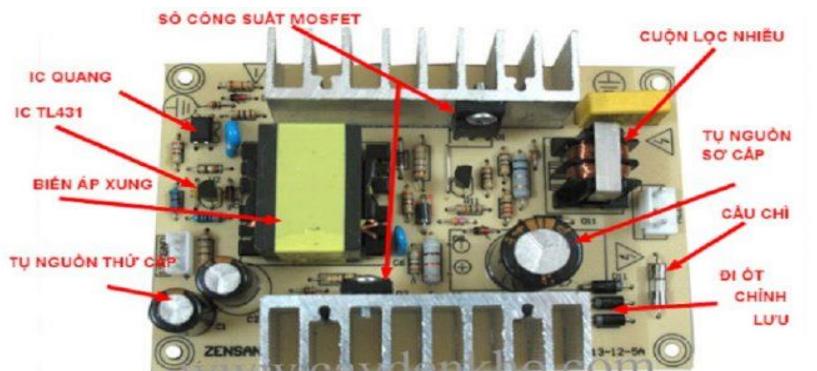
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.19: Sơ đồ khái Nguồn tổ ong (nguồn xung)

Thông số kỹ thuật của nguồn tổ ong 12V 5A:

- Điện áp ngõ vào: 110/220 VAC
- Điện áp ngõ ra: 12 VDC
- Sai số điện áp đầu ra: 1-3 %
- Công suất thực tế :88 %
- Nhiệt độ làm việc :0-70 độ C



Hình 2.20: Cấu tạo cơ bản một bộ nguồn tổ ong

Một bo nguồn xung sẽ bao gồm những linh kiện cơ bản sau:

- Biến áp xung: Cũng cấu tạo gồm các cuộn dây quấn trên một lõi từ giống như biến áp thông thường chỉ có điều biến áp này sử dụng lõi ferit còn

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

biến áp thường sử dụng lõi thép kỹ thuật điện. Với cùng một kích thước thì biến áp xung cho công suất lớn hơn biến áp thường rất nhiều lần. Ngoài ra biến áp xung hoạt động tốt ở dải tần cao còn biến áp thường chỉ hoạt động ở dải tần thấp.

- Cầu chì: Bảo vệ mạch nguồn bị ngắn mạch
- Cuộn chống nhiễu, tụ lọc sơ cấp, diode chỉnh lưu: Có nhiệm vụ biến đổi điện áp xoay chiều 220V thành điện áp một chiều tích trữ trên tụ lọc sơ cấp để cung cấp năng lượng cho cuộn sơ cấp của máy biến áp xung
- Sò công suất: Đây là một linh kiện bán dẫn dùng như một công tắc chuyển mạch, đó có thể là transistor, mosfet, IC tích hợp, IGBT có nhiệm vụ đóng cắt điện từ chân (+) của tụ lọc sơ cấp vào cuộn dây sơ cấp của biến áp xung rồi cho xuống mass.
- Tụ lọc nguồn thứ cấp: Dùng để tích trữ năng lượng điện từ cuộn thứ cấp của biến áp xung để cấp cho tải tiêu thụ. Chúng ta biết rằng khi cuộn sơ cấp của biến áp được đóng cắt điện liên tục bằng sò công suất thì xuất hiện từ trường biến thiên dẫn đến cuộn thứ cấp của biến áp cũng xuất hiện một điện áp ra. Điện áp này được chỉnh lưu qua một vài diode rồi đưa ra tụ lọc (tụ điện) thứ cấp để san phẳng điện áp.
- IC quang và IC TL431: Có nhiệm vụ tạo ra một điện áp cố định để không chế điện áp ra bên thứ cấp ổn định theo mong muốn. Chúng sẽ làm nhiệm vụ không chế dao động đóng cắt điện vào cuộn sơ cấp của biến áp xung sao cho điện áp ra bên thứ cấp đạt yêu cầu.

Mạch hạ áp LM2596 có thông số kỹ thuật như sau:

- Điện áp đầu vào: Từ 3V đến 30V.
- Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
- Dòng đáp ứng tối đa là 3A.
- Hiệu suất: 92%
- Công suất: 15W

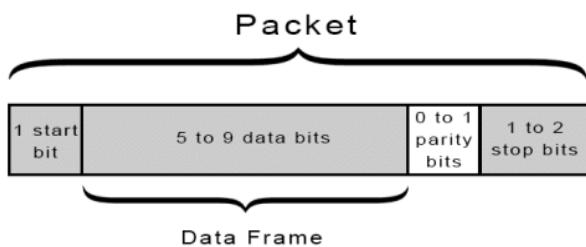
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.4 Các chuẩn truyền dữ liệu

2.4.1 Chuẩn truyền thông UART

UART [6] là viết tắt của Universal Asynchronous Receiver – Transmitter là kiểu truyền thông tin nối tiếp không đồng bộ thường là một mạch tích hợp. Mục đích của UART là để truyền tín hiệu qua lại lẫn nhau (ví dụ truyền tín hiệu từ Laptop vào Modem hay ngược lại) hay truyền từ vi điều khiển tới vi điều khiển, từ laptop tới vi điều khiển.

Các thông số trong chuẩn truyền UART:



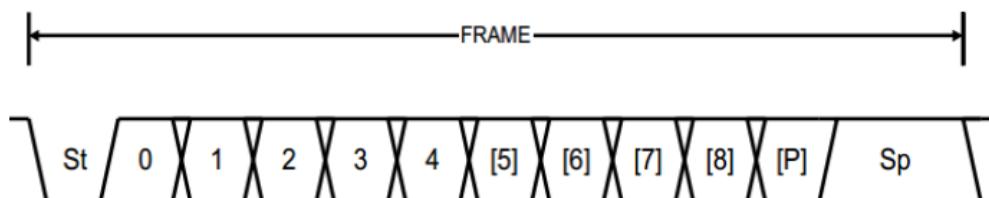
Hình 2.21: Gói dữ liệu truyền của UART

- **Packet:** Một gói dữ liệu được truyền đi, bao gồm bit Start, khung truyền dữ liệu, bit parity, bit Stop
- **Bit Start:** thường được giữ ở mức điện áp cao khi nó không truyền dữ liệu. Để bắt đầu truyền dữ liệu, UART truyền sẽ kéo bit này từ cao xuống thấp trong một chu kỳ xung nhịp. Khi UART nhận phát hiện sự chuyển đổi điện áp cao sang thấp, nó bắt đầu đọc các bit trong khung dữ liệu ở tần số của tốc độ truyền.
- **Khung dữ liệu:** chứa dữ liệu được truyền, nó có thể dài từ 5 đến 8 bit nếu sử dụng một bit parity. Nếu không có bit parity nào được sử dụng, khung dữ liệu có thể dài 9 bit. Bit parity có tác dụng kiểm tra xem dữ liệu có bị thay đổi trong quá trình truyền không bằng cách là kiểm tra tổng số bit 1 là chẵn hay lẻ rồi so sánh với dữ liệu. Nếu tổng số bit 1 là chẵn mà bit parity bằng 0 thì quá trình truyền không có lỗi và nếu bằng 1 thì đường truyền bị lỗi khiến dữ liệu bị thay đổi.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- **Bit Stop:** để báo hiệu sự kết thúc của gói dữ liệu, UART gửi sẽ điều khiển đường truyền dữ liệu từ điện áp thấp đến điện áp cao trong ít nhất hai bit.

Nguyên lý hoạt động: UART truyền sẽ nhận dữ liệu từ một bus dữ liệu. Dữ liệu được truyền từ bus dữ liệu sang UART truyền ở dạng song song rồi thêm bit start, bit parity và bit stop để tạo gói dữ liệu. Sau đó, gói dữ liệu được xuất ra dạng nối tiếp tuần tự ở chân Tx rồi truyền qua chân Rx của UART nhận. UART nhận đọc các gói dữ liệu nhận được rồi loại bỏ bit start, bit parity và bit stop. Sau đó chuyển đổi dữ liệu trở lại dạng song song. Cuối cùng, UART nhận chuyển gói dữ liệu song song qua bus dữ liệu [6].



Hình 2.22: Sóng truyền UART



Hình 2.23: Quá trình truyền UART



Hình 2.24: Quá trình nhận UART

2.4.2 Chuẩn truyền thông SPI

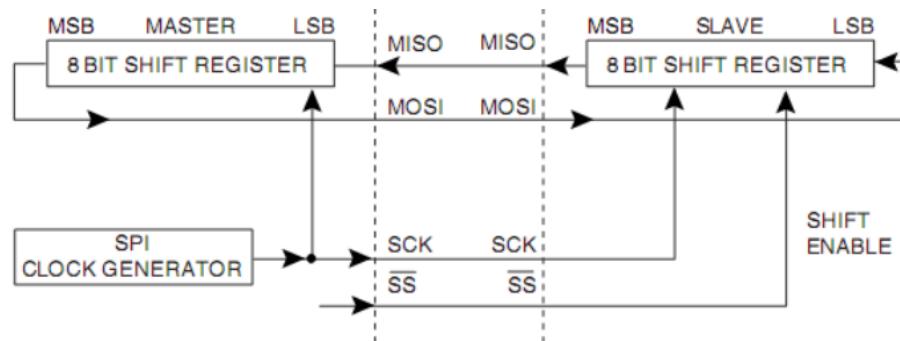
SPI (Serial Peripheral Bus) [6]: là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất. Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip Master điều phối quá trình truyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave. SPI là một cách truyền song công (full duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn này đó là SC (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Output Slave Input) và SS (Slave Select).

- **SCK**: Xung giữ nhịp cho giao tiếp SPI, vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SC báo 1bit dữ liệu đến hoặc đi. Sự tồn tại của chân SC giúp quá trình truyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.
- **MISO** (Master Input /Slave Output): Nếu là chip Master thì đây là đường Input còn nếu là chip Slave thì MISO lại là Output. MISO của Master và các Slaves được nối trực tiếp với nhau.
- **MOSI** (Master Output / Slave Input): Nếu là chip Master thì đây là đường Output còn nếu là chip Slave thì MOSI là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.
- **SS** (Slave Select): SS là đường chọn Slave cần giao tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc. Nếu chip Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó. Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master, tùy thuộc vào thiết kế của người dùng.

Nguyên lý hoạt động: Mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bit. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, 1bit trong thanh ghi của Master được truyền qua Slave trên đường MOSI, đồng thời 1bit trong thanh

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

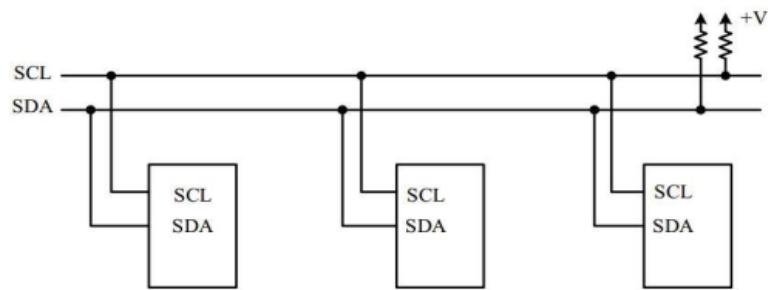
ghi của chip Slave cũng được truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip được gởi qua lại đồng thời nên quá trình truyền dữ liệu này được gọi là “song công”. Hình dưới đây mô tả quá trình truyền 1 gói dữ liệu thực hiện bởi module SPI trong AVR, bên trái là chip Master và bên phải là Slave [6].



Hình 2.25: Truyền dữ liệu SPI

2.4.3 Chuẩn giao tiếp I2C

I2C [6] là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. 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, ARM.



Hình 2.26: Bus I2C và các thiết bị ngoại vi

Đặc điểm giao tiếp I2C:

- Một giao tiếp I2C gồm có 2 dây: Serial Data (SDA) và Serial Clock (SCL). SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung đồng hồ để đồng bộ và chỉ theo một hướng. Khi một thiết bị ngoại

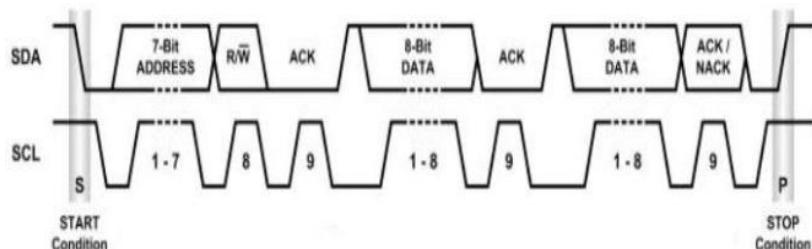
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

viết nối vào đường bus I2C thì chân SDA của nó sẽ nối với dây SDA của bus, chân SCL sẽ nối với dây SCL.

- Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên (pullup resistor). Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực mảng hở (opendrain hay opencollector). Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng $1\text{ K}\Omega$ đến $4.7\text{ K}\Omega$.

Chế độ hoạt động (tốc độ truyền): Các bus I2C có thể hoạt động ở ba chế độ, hay nói cách khác các dữ liệu trên bus I2C có thể được truyền trong ba chế độ khác nhau [6]

- Chế độ tiêu chuẩn (Standard mode)
- Chế độ nhanh (Fast mode)
- Chế độ cao tốc High-Speed (Hs) mode



Hình 2.27: Trình tự truyền bit trên đường truyền

Trình tự truyền bit trên đường truyền:

- Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tò lăng nghe dữ liệu trên đường truyền.
- Thiết bị chủ gửi địa chỉ của thiết bị tò mà thiết bị chủ muốn giao tiếp và cờ đọc/ghi dữ liệu (nếu cờ thiết lập lên 1 thì byte tiếp theo được truyền từ thiết bị tò đến thiết bị chủ, nếu cờ thiết lập xuống 0 thì byte tiếp theo truyền từ thiết bị chủ đến thiết bị tò).

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Khi thiết bị tớ trên bus I2C có địa chỉ đúng với địa chỉ mà thiết bị chủ gửi sẽ phản hồi lại bằng một xung ACK.
- Giao tiếp giữa thiết bị chủ và tớ trên bus dữ liệu bắt đầu. Cả chủ và tớ đều có thể nhận hoặc truyền dữ liệu tùy thuộc vào việc truyền thông là đọc hay viết. Bộ truyền gửi 8bit dữ liệu tới bộ nhận, bộ nhận trả lời với một bit ACK.
- Để kết thúc quá trình giao tiếp, thiết bị chủ tạo ra một điều kiện stop.

Chương 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

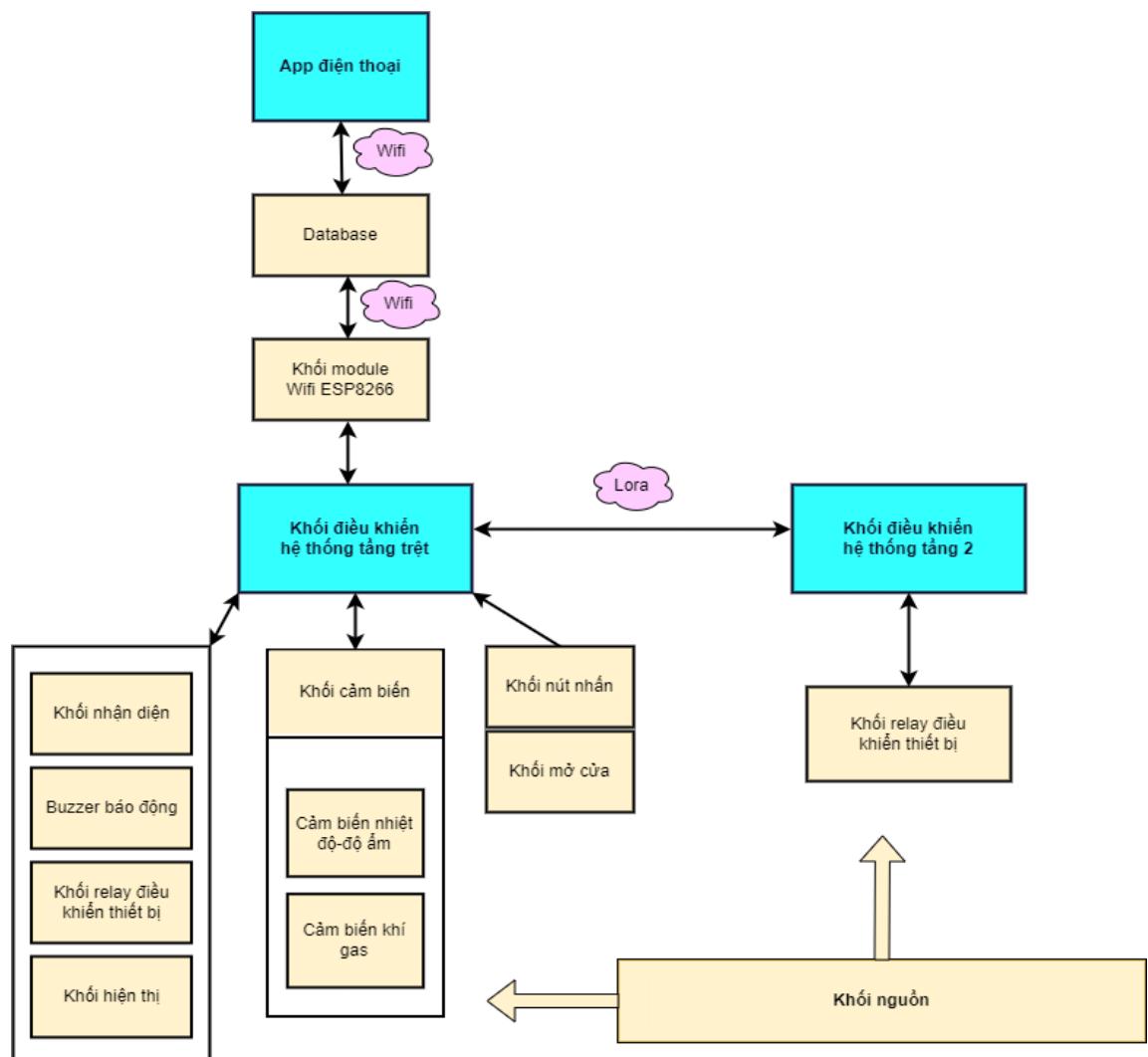
3.1 Giới thiệu

Trong chương này, trình bày về cách tính toán, sơ đồ khói, sơ đồ nguyên lý của các board mạch của hệ thống.

3.2 Thiết kế hệ thống

3.2.1 Thiết kế sơ đồ khói hệ thống

Với các yêu cầu đã đưa ra nhóm em đã hình thành sơ đồ khói cho hệ thống như sau:



Hình 3.1: Sơ đồ khói của hệ thống

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Chức năng từng khối:

Tại tầng 1:

- Khối điều khiển hệ thống tầng trệt: Sử dụng Arduino Mega 2560 để làm vi xử lý trung tâm kết nối với Raspberry Pi 3 mode B để điều khiển camera, kết nối với bàn phím ma trận, kết nối với các cảm biến để đưa dữ liệu nhiệt độ - độ ẩm lên database (firebase của Google) và lấy dữ liệu từ firebase xuống điều khiển các thiết bị điện tầng trệt và tầng 2.
- App điện thoại: Nhận và cập nhật giá trị nhiệt độ - độ ẩm, đồng thời truyền dữ liệu xuống để điều khiển các thiết bị.
- Lora: Truyền dữ liệu giữa khói điều khiển hệ thống tầng trệt và khói điều khiển hệ thống tầng 2.
- Khối hiển thị: Hiển thị trạng thái của cửa lên LCD.
- Khối cảm biến: Khối này gồm các cảm biến khí gas MQ2, cảm biến nhiệt độ - độ ẩm DHT11.
- Khối relay điều khiển thiết bị : Khối này bao gồm 1 module Relay 8 kênh kết nối với 4 bóng đèn và 3 quạt nhằm mục đích để điều khiển bật, tắt thông qua Android.
- Khối nút nhấn: Khối này gồm nút nhấn dùng để reset chuông buzzer và bàn phím ma trận 3x4 để nhập mật khẩu mở cửa.
- Khối buzzer báo động: Khối này gồm chuông buzzer. Nhận tín hiệu từ khi khói điều khiển hệ thống tầng trệt khi có khí Gas vượt hoặc có khói.
- Khối mở cửa: Sử dụng servo để mở cửa
- Khối nguồn: Cung cấp nguồn cho các khối hoạt động .

Tại tầng 2:

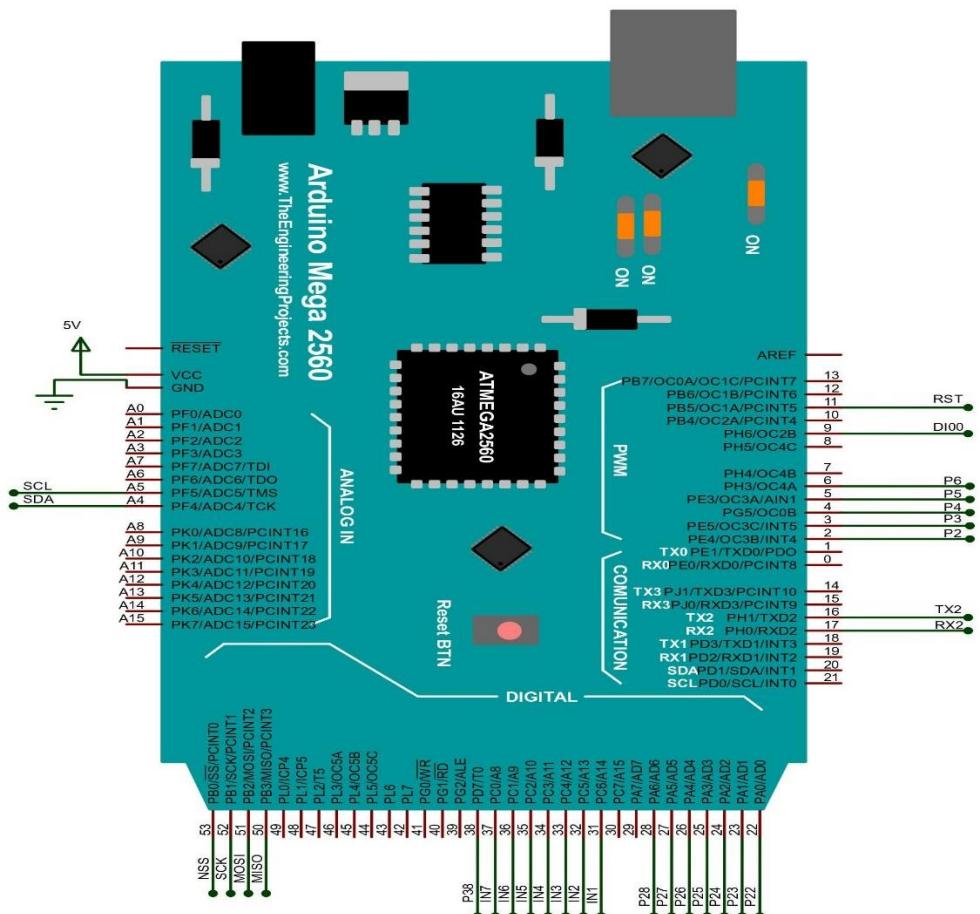
- Khối điều khiển hệ thống tầng 2: Dùng Arduino Uno để nhận dữ liệu từ khói điều khiển trung tâm thông qua Lora để điều khiển các thiết bị điện.

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

- Lora: Nhận dữ liệu giữa khói điều khiển hệ thống tầng trệt và khói điều khiển hệ thống tầng 2.
- Khối relay điều khiển thiết bị: Khối này bao gồm 2 module Relay 4 kênh kết nối với 2 bóng đèn và 2 quạt nhằm mục đích để điều khiển bật, tắt thông qua App điện thoại.
- Khối nguồn: Cung cấp nguồn cho các khói hoạt động

3.2.2 Thiết kế mạch – Tầng trệt

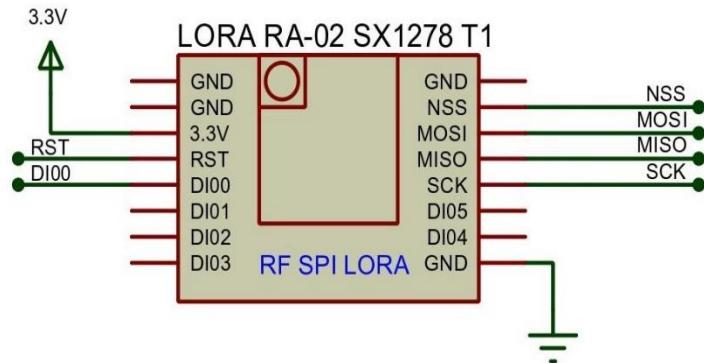
Ở đề tài này nhóm sẽ điều khiển nhiều đèn, quạt, một vài module khác như ESP8266, Lora, cảm biến nhiệt độ độ ẩm DHT11, cảm biến khí gas MQ-02, Raspberry Pi 3, servo, LCD 16x2 nên cần số lượng chân GPIO để kết nối với các thiết bị đó.



Hình 3.2: Kết nối chân Arduino Mega 2560

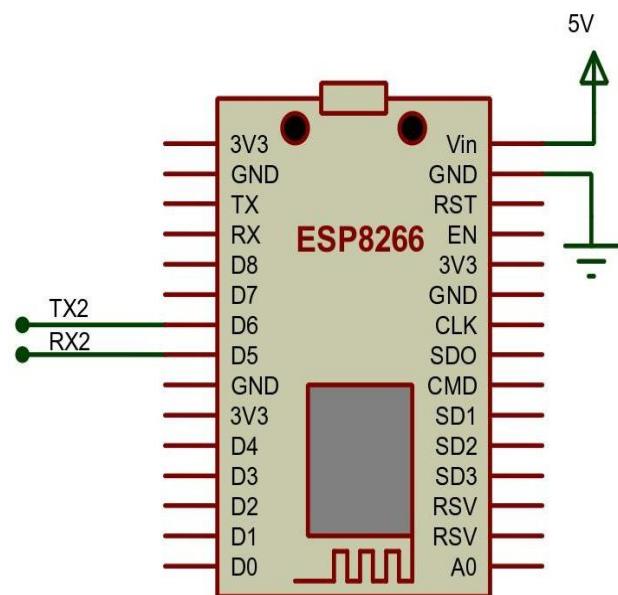
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Module Lora Ra-02 được kết nối với Arduino Mega 2560 qua bộ SPI với 4 chân SCK, MISO, MOSI, NSS của module Lora Ra-02 với lần lượt các chân 52 (SCK), 50 (MISO), 51 (MOSI), 53 (NSS) của Arduino Mega 2560.



Hình 3.3: Kết nối chân module Lora Ra-02

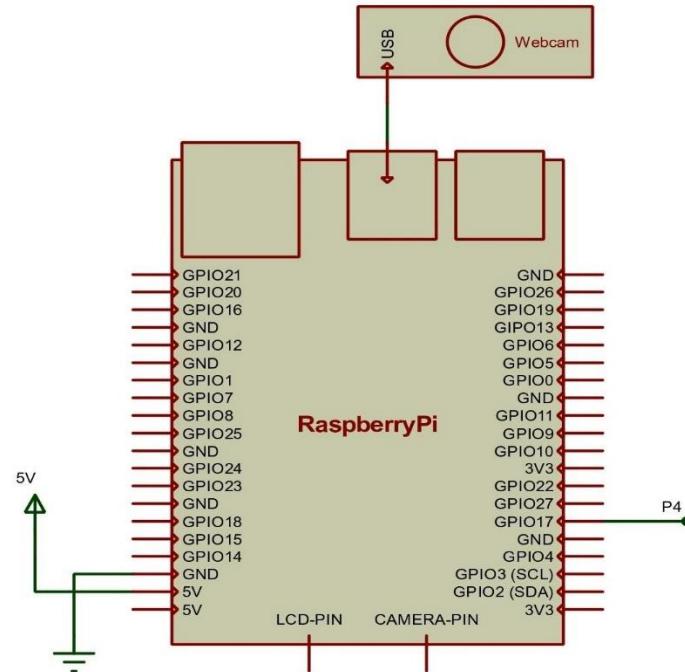
ESP8266 giao tiếp với Arduino Mega 2560 qua bộ UART 2 (RX-D5, TX-D6).



Hình 3.4: Kết nối chân ESP8266

Raspberry Pi 3 kết nối chân GPIO17 với chân 4 của Arduino Mega 2560 để cấp nguồn cho Raspberry hoạt động

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

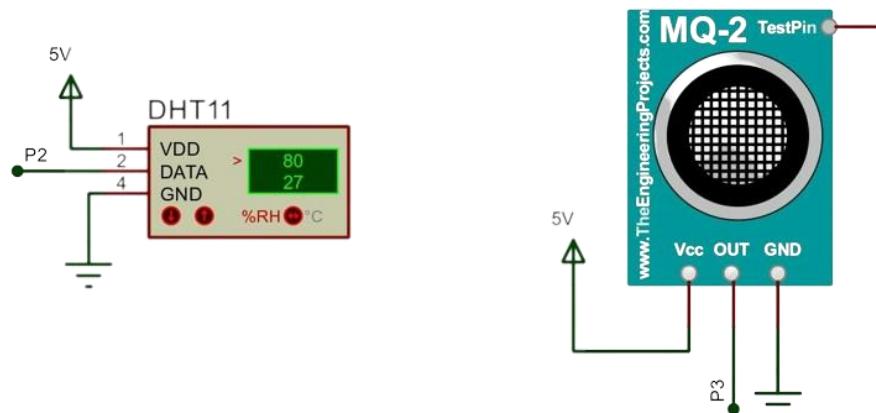


Hình 3.5: Kết nối chân Raspberry Pi

Khối cảm biến bao gồm module cảm biến khí gas MQ-2 và cảm biến đo nhiệt độ - độ ẩm DHT11.

Cảm biến đo nhiệt độ - độ ẩm DHT11 giao tiếp với vi điều khiển qua chuẩn giao tiếp One-wire, gửi và nhận dữ liệu trên 1 dây.

Cảm biến khí gas MQ-2 giao tiếp với vi điều khiển bằng chân Out nối với chân 3 của Arduino Mega 2560. Sơ đồ nối dây chi tiết của khối cảm biến này được trình bày như hình 3.6

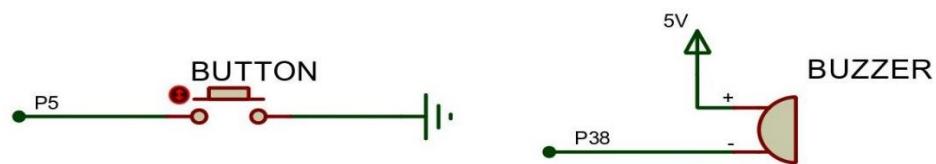


Hình 3.6: Cảm biến nhiệt độ-độ ẩm DHT11 và cảm biến khí gas MQ2

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

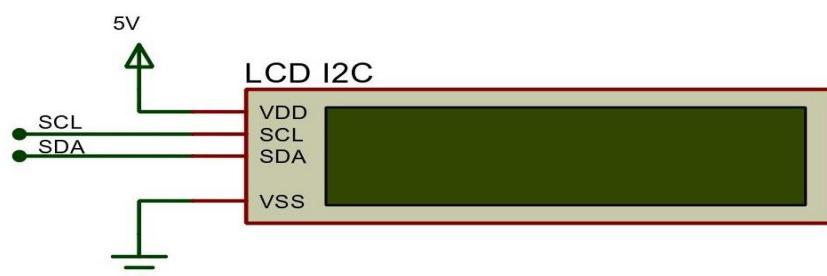
Ở đề tài này nhóm dùng buzzer loại 2 dây để báo động khi có cháy hoặc khí gas rò rỉ. Chân âm của buzzer được nối với chân 38 và chân dương của buzzer được nối 5V của Arduino Mega 2560.

Nút nhấn có chức năng tắt buzzer reo, nên nhóm sử dụng loại nút nhấn 2 chân. Trong chương trình xuất nút nhấn ở dạng INPUT_PULLUP nên sẽ kết nối trực tiếp với Arduino Mega 2560 mà không cần thông qua điện trở. Sơ đồ nối dây chi tiết của khói nút nhấn này được trình bày như hình 3.7



Hình 3.7: Còi báo buzzer và nút nhấn reset

Nhằm tiết kiệm các chân giao tiếp của Arduino Mega 2560 chúng em giao tiếp LCD thông qua module I2C, chân VDD được nối với nguồn 5V, chân VSS được nối mass, chân SCL được nối với chân A5 của Arduino Mega 2560, chân SDA được nối với chân A4 của Arduino Mega 2560. Sơ đồ nối dây chi tiết được trình bày như hình 3.8:

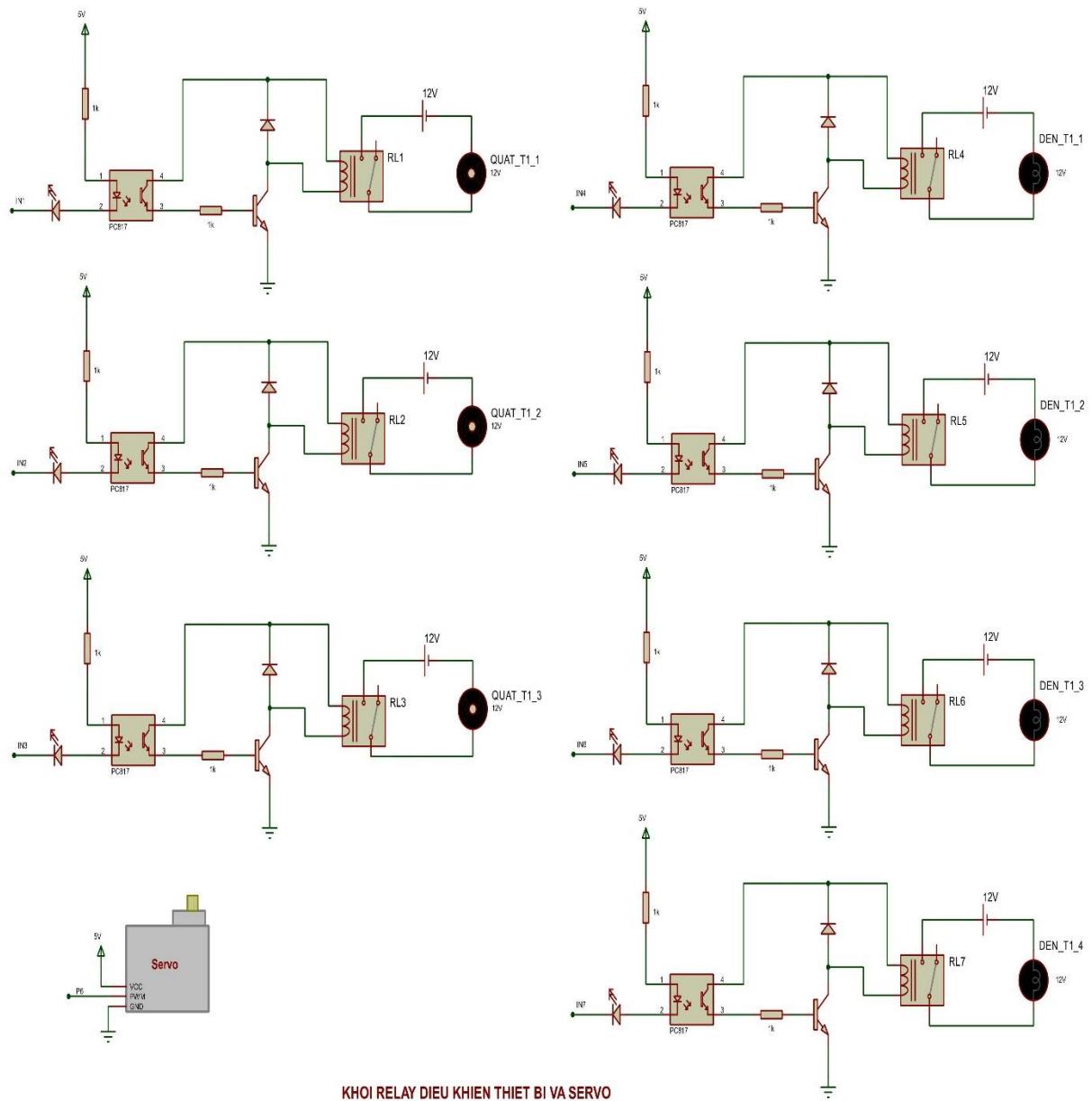


Hình 3.8: Màn hình hiển thị LCD

Để phù hợp cho hệ thống nhóm em chọn module relay 5V 8 kênh. Dòng tối đa mà relay này có thể chịu được tối đa trên tiếp điểm là 10A, nên đảm bảo dòng điện

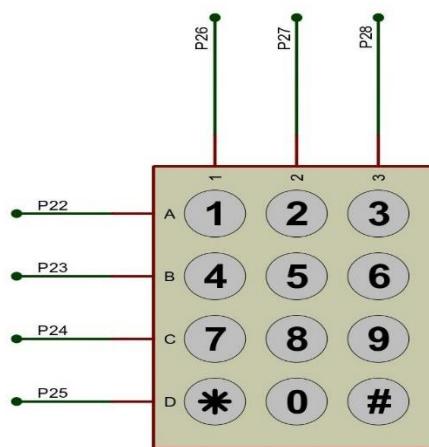
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

trong quạt, đèn khi chạy qua các tiếp điểm sẽ an toàn. Các chân 31, 32, 33, 34, 35, 36, 37 của Arduino Mega 2560 sẽ điều khiển kích hoạt các relay. Chân 6 của Arduino Mega 2560 sẽ điều khiển servo để mở cửa. Sơ đồ nối dây chi tiết được trình bày như hình 3.9



Hình 3.9: Sơ đồ nối dây các thiết bị ở tầng 1

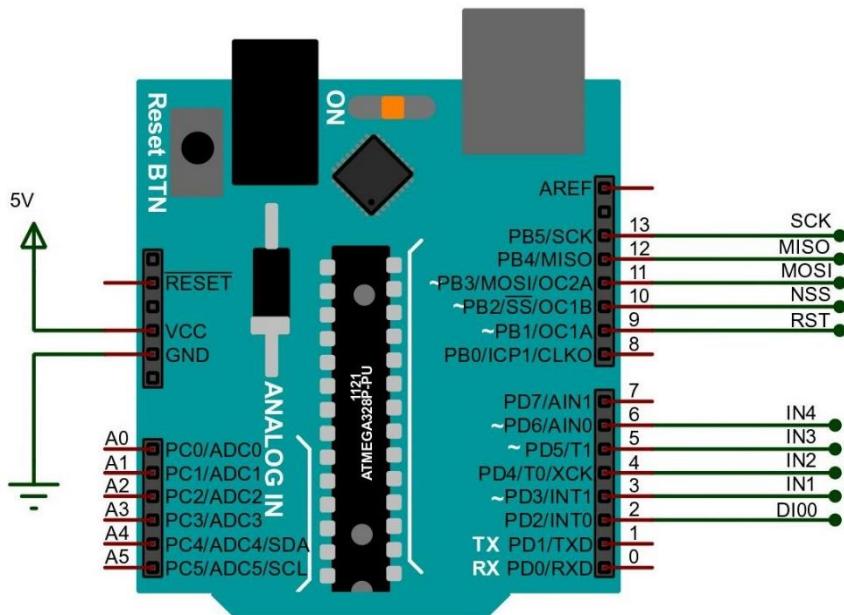
Trong đề tài này chúng em dùng bàn phím ma trận 3x4 để dùng nhập mật khẩu để mở cửa. Sơ đồ nối dây chi tiết được trình bày như hình 3.10



Hình 3.10: Kết nối chân bàn phím ma trận 3x4

3.2.3 Thiết kế mạch – Tầng 2

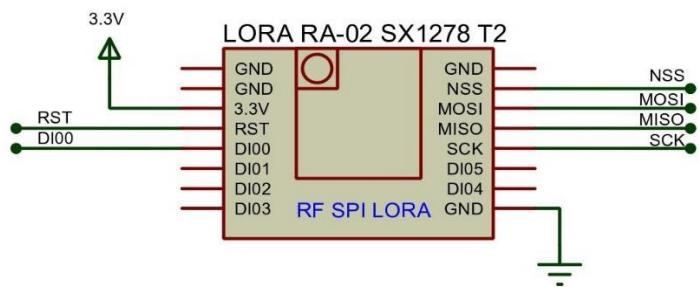
Ở tầng 2 này, chỉ điều khiển đèn và quạt, lora nên không cần số lượng chân nhiều như tầng 1 nên vì thế chúng em sử dụng Arduino Uno để điều khiển các thiết bị ở tầng này.



Hình 3.11: Kết nối chân Arduino Uno

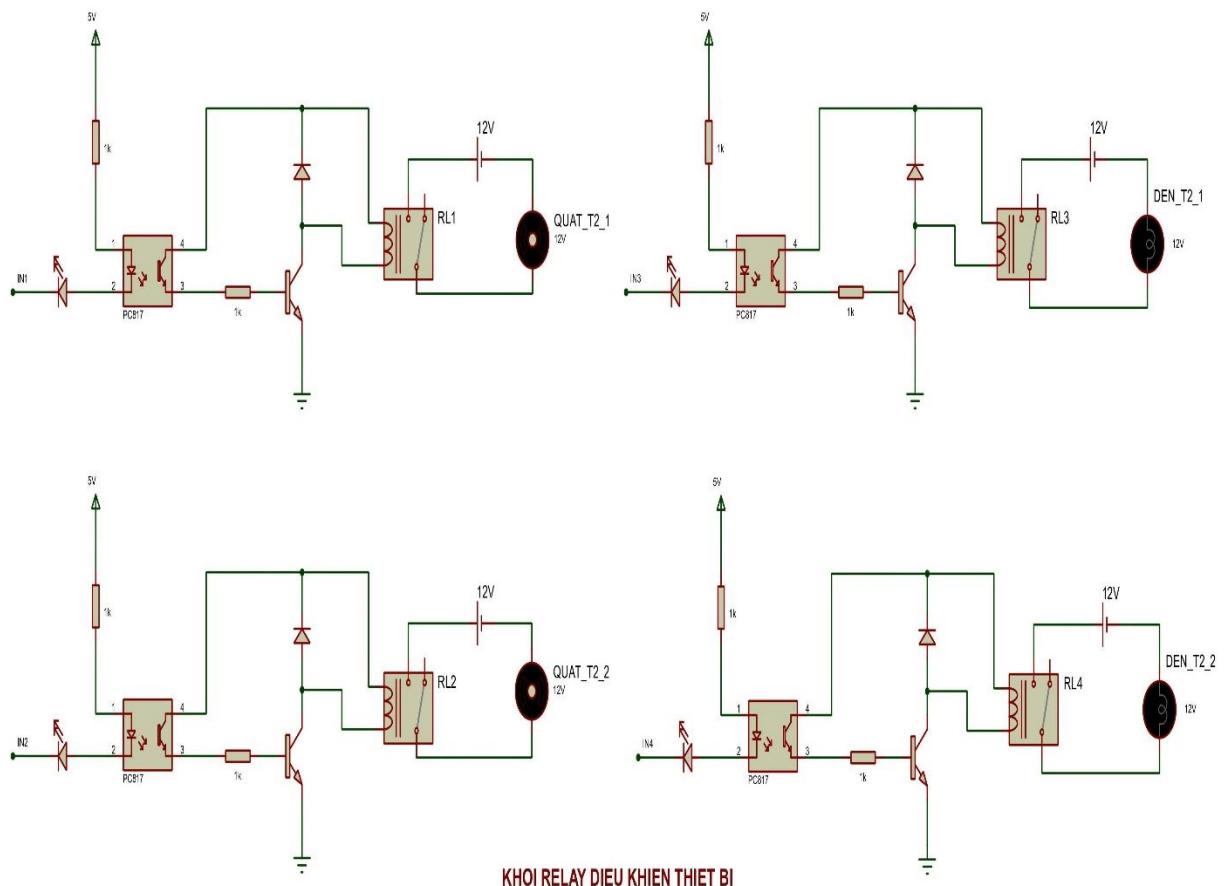
Module Lora Ra-02 ở tầng 2 được kết nối với Arduino Uno qua bộ SPI với 4 chân SCK, MISO, MOSI, NSS của module Lora Ra-02 với lần lượt các chân 13 (SCK), 12 (MISO), 11 (MOSI), 10 (NSS) của Arduino Uno.

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG



Hình 3.12: Kết nối chân module Lora Ra-02

Để phù hợp cho hệ thống ở tầng 2 (điều khiển 2 đèn và 2 quạt) nhóm em chọn module relay 5V 4 kênh . Dòng tối đa mà relay này có thể chịu được tối đa trên tiếp điểm là 10A, nên đảm bảo dòng điện trong quạt, đèn khi chạy qua các tiếp điểm sẽ an toàn. Các chân 3, 4, 5, 6, của Arduino Uno sẽ điều khiển kích hoạt các relay. Sơ đồ nối dây chi tiết được trình bày như hình 3.13

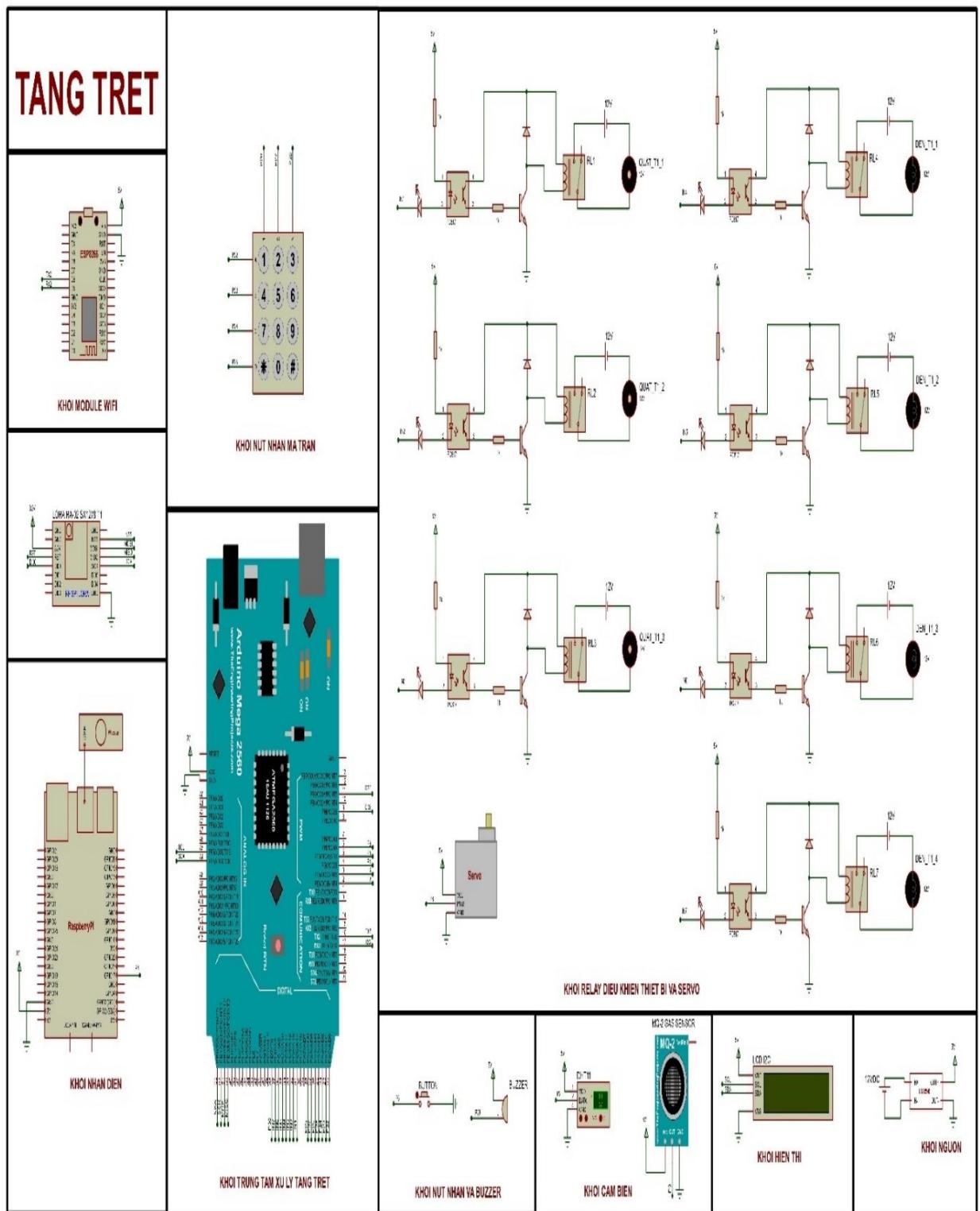


Hình 3.13: Sơ đồ nối dây các thiết bị ở tầng 2

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

3.3 Sơ đồ nguyên lý toàn bộ hệ thống

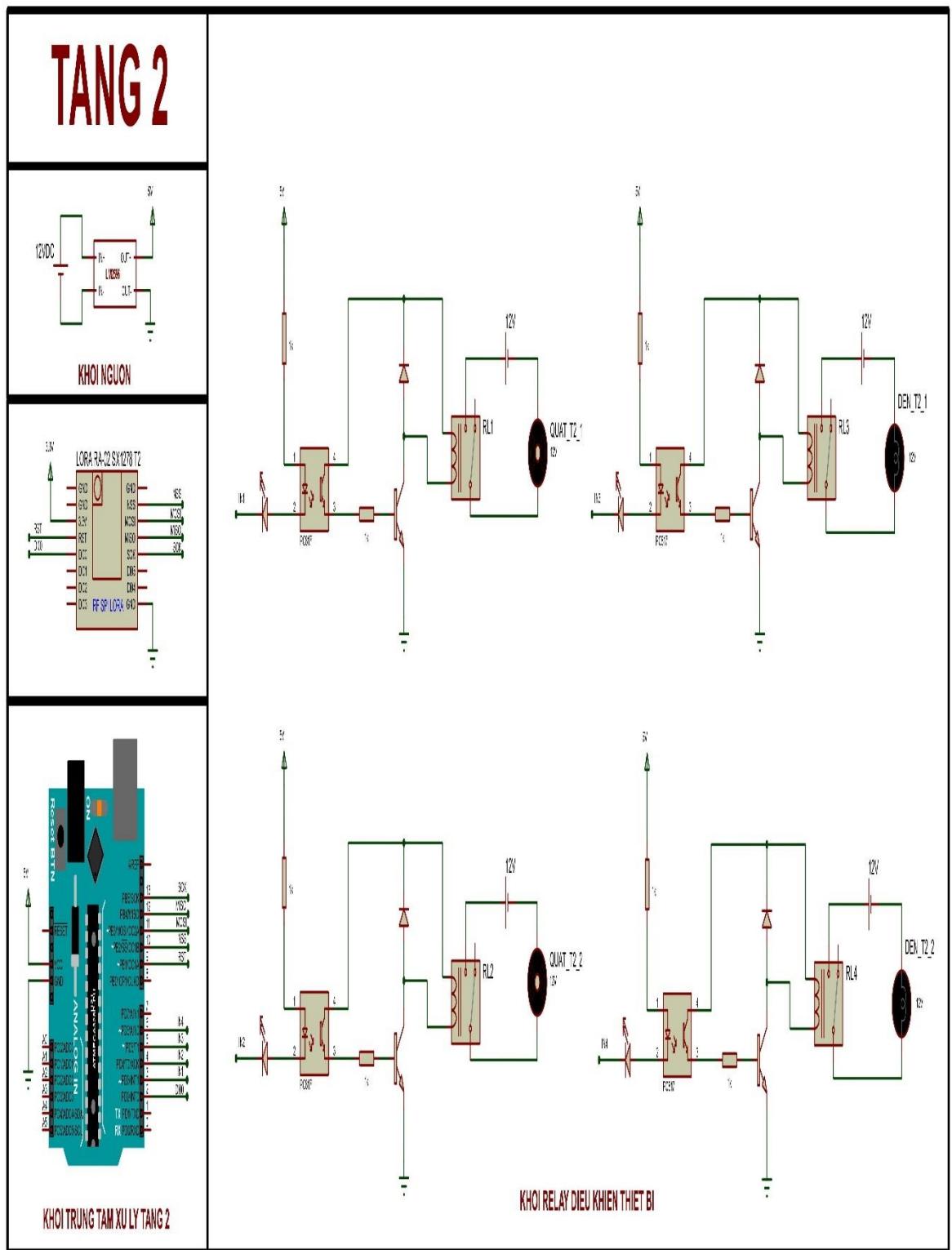
Sơ đồ nguyên lý tầng 1:



Hình 3.14: Sơ đồ nguyên lý tầng 1

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Sơ đồ nguyên lý tầng 2:



Hình 3.15: Sơ đồ nguyên lý tầng 2

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Giải thích nguyên lý:

Hệ thống tầng 1: Arduino Mega 2560 có chức năng là vi điều khiển chính, truyền và nhận dữ liệu với hệ thống ở tầng 2 thông qua module Lora Ra-02, điều khiển các thiết bị điện (đèn, quạt) bằng cách nhấn các biểu tượng nút nhấn trên app Android mit app inventor khi đó dữ liệu từ firebase sẽ được đưa xuống ESP8266 rồi chuyển sang cho Arduino Mega 2560 xử lý bật, tắt các thiết bị. Các cảm biến DHT11 và MQ-2 dùng nguồn 5V từ Arduino Mega 2560, cảm biến MQ-2 được Arduino Mega 2560 đọc và nếu có cháy hay khói thì còi sẽ kêu lên và để tắt thì nhấn nút nhấn (button), còi sẽ ngừng kêu. Cảm biến DHT11 được Arduino Mega 2560 đọc các giá trị sau đó chuyển qua ESP8266 rồi đưa các giá trị đọc được hiển thị lên Android. Ở đê tài này chúng em có 2 cách để mở cửa cho nhà: nhận diện khuôn mặt để mở cửa hoặc nhập mật khẩu từ bàn phím ma trận 3x4. Raspberry dùng nguồn 5V của Arduino Mega 2560 có chức năng xử lý nhận diện khuôn mặt để mở cửa, nếu đúng khuôn mặt thì servo sẽ kích để mở cửa nếu không đúng khuôn mặt thì cửa không mở. LCD sẽ hiển thị Open Door hoặc Close Door.

Hệ thống tầng 2: Arduino Uno có chức năng nhận dữ liệu yêu cầu điều khiển từ hệ thống tầng 1, lấy dữ liệu và xuất các giá trị tại chân 3, 4, 5, 6 để điều khiển module relay để điều khiển các thiết bị điện ở tầng 2.

Chương 4. THI CÔNG HỆ THỐNG

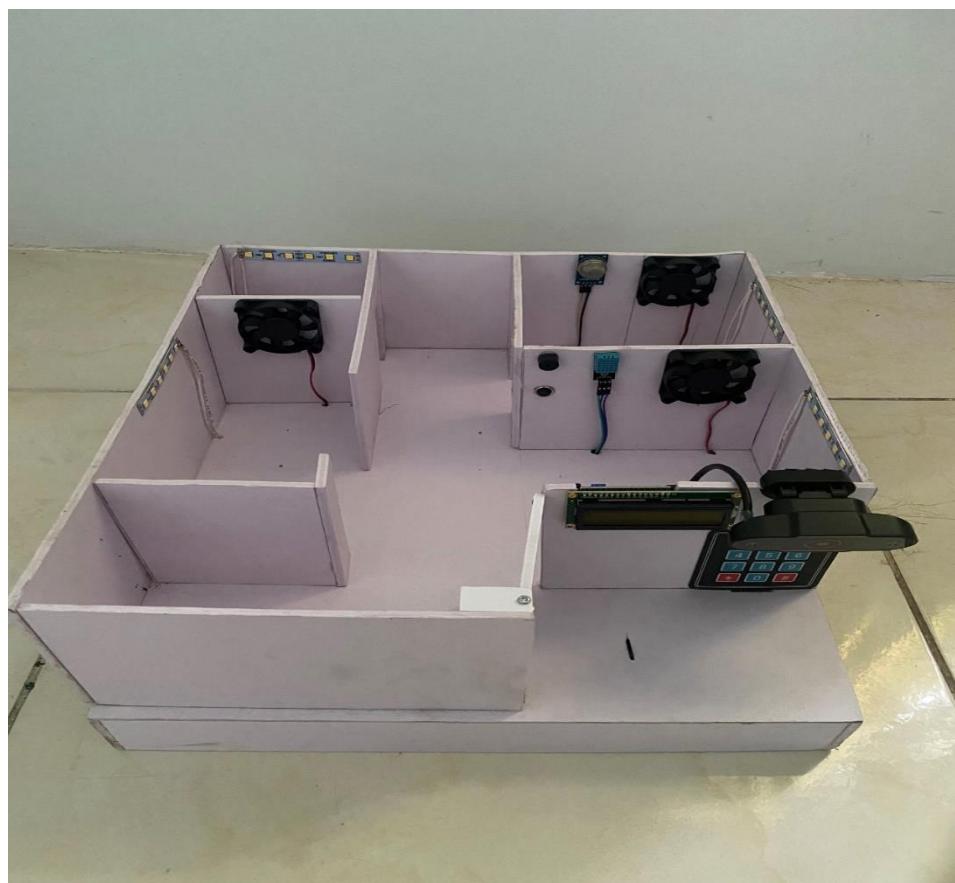
4.1 Giới thiệu

Sau khi tính toán và thiết nhóm sẽ thi công mô hình cho hệ thống. Hệ thống tầng 1 gồm các thiết bị như: Arduino Mega 2560, Raspberry Pi 3 Model B, ESP8266 Node MCU, Lora Ra-02, Module relay 5V 8 kênh, servo, cảm biến DHT11, cảm biến MQ-2, LCD 16x2, bàn phím ma trận 3x4, buzzer, nút nhấn reset, camera USB. Hệ thống tầng 2 gồm các thiết bị: Arduino Uno, Lora Ra-02, Module relay 5V 4 kênh.

4.2 Thi công hệ thống

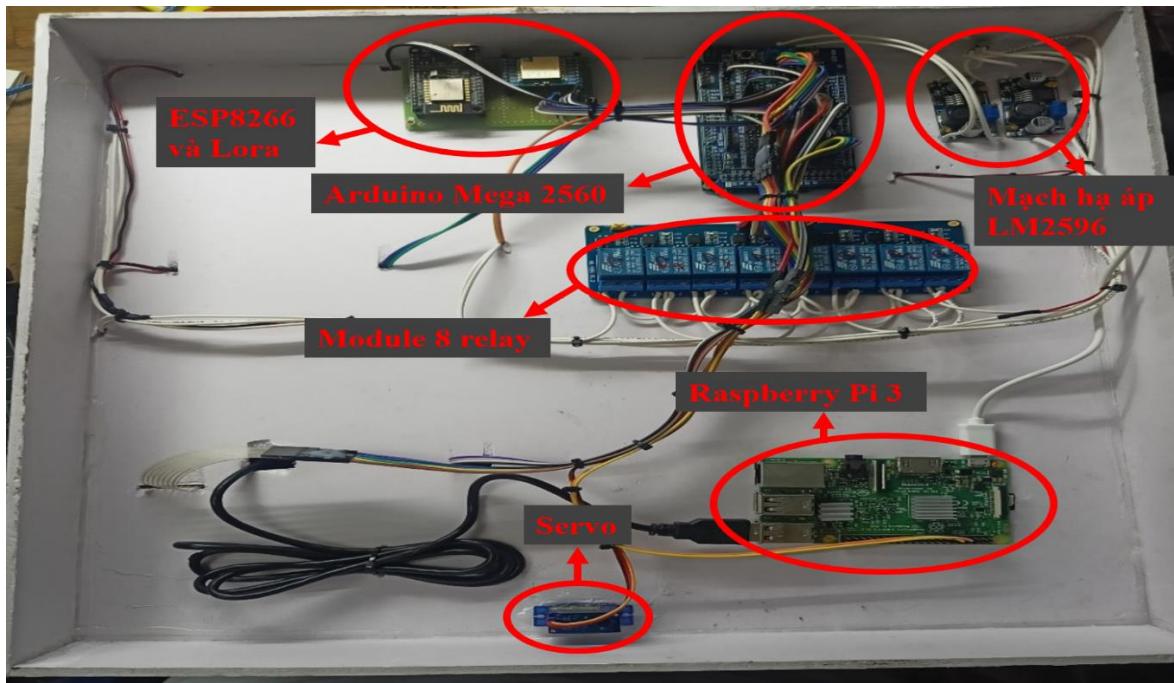
4.2.1 Mô hình hệ thống tầng 1:

Dưới đây là bố trí linh kiện của hệ thống tầng 1 với mặt trên và mặt dưới, tương ứng với hình 4.1 và 4.2



Hình 4.1: Mặt trên của mô hình hệ thống tầng 1

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 4.2: Mặt dưới của mô hình hệ thống tầng 1

Bảng 4.1: Số liệu dòng và áp của các thiết bị tại tầng 1

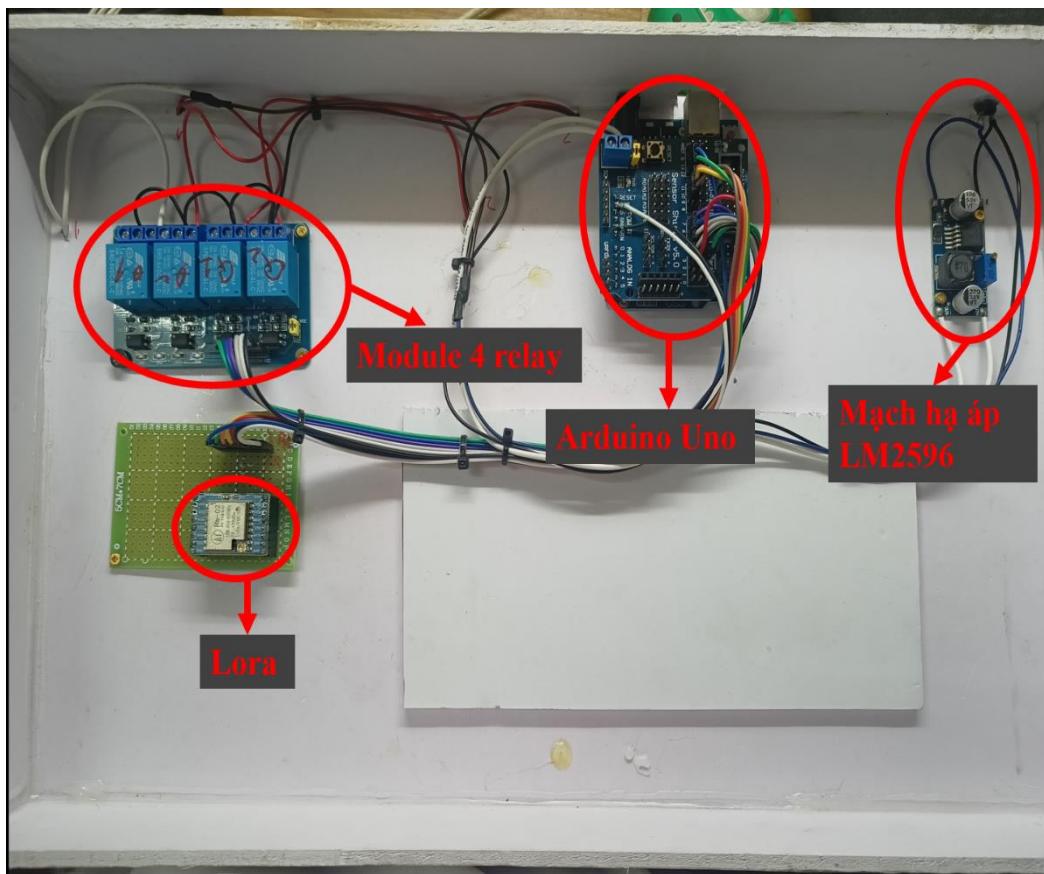
Tên linh kiện	Số lượng	Giá trị dòng (A)	Giá trị áp(V)
Module Lora	1	170m	3.3
Module Adruino Mega	1	1.5	5
Module Relay 8 kênh	1	10	5
RaspberryPi 3B	1	2.5	5
Camera	1		5
Bàn phím ma trận 4x4	1	20m	5
Cảm biến DHT11	1	5m	5
Cảm biến khí gas MQ2	1	120m	5
Đèn led	4	36m	12
Quạt	3	400m	12
Chuông báo	1	20m	5
Nút nhấn	1	3	5
LCD 16x2 và module I2C	1	50m	5
ESP8266	1	120m	5

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

4.2.2 Mô hình hệ thống tầng 2



Hình 4.3: Mặt trên của mô hình hệ thống tầng 2



Hình 4.4: Mặt dưới của mô hình hệ thống tầng 2

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Bảng 4.2: Số liệu dòng và áp của các thiết bị tại tầng 2

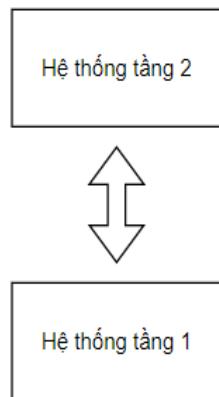
Tên linh kiện	Số lượng	Giá trị dòng (A)	Giá trị áp(V)
Module Lora	1	170m	3,3
Module Adruino Uno	1	680m	5
Module Relay 4 kênh	1	10	5
Đèn led	2	36m	12
Quạt	2	400m	12

4.2.3 Khối nguồn

Dựa vào các thông số tại bảng 4.1 và 4.2 ở trên do các linh kiện đều sử dụng nguồn 12V và 5V nên nhóm em đã quyết định sử dụng bộ nguồn tổ ong chuyền 220 VAC từ nguồn điện dân dụng sang 12 VDC, để cấp nguồn cho linh kiện đó hoạt động. Còn các linh kiện cần nguồn 5 VDC để hoạt động thì sử dụng mạch nguồn hạ áp LM2596 để hạ từ 12 VDC xuống 5VDC để cấp cho linh kiện đó hoạt động.

4.3 Đóng gói và thi công mô hình

Sau khi kiểm tra mạch hoạt động tốt ta tiến hành đóng gói thành mô hình hoàn chỉnh. Hệ thống được xây dựng gồm một hệ thống tầng 1 và hệ thống tầng 2 có kiến trúc như hình 4.4 bên dưới.



Hình 4.5: Kiến trúc mô hình

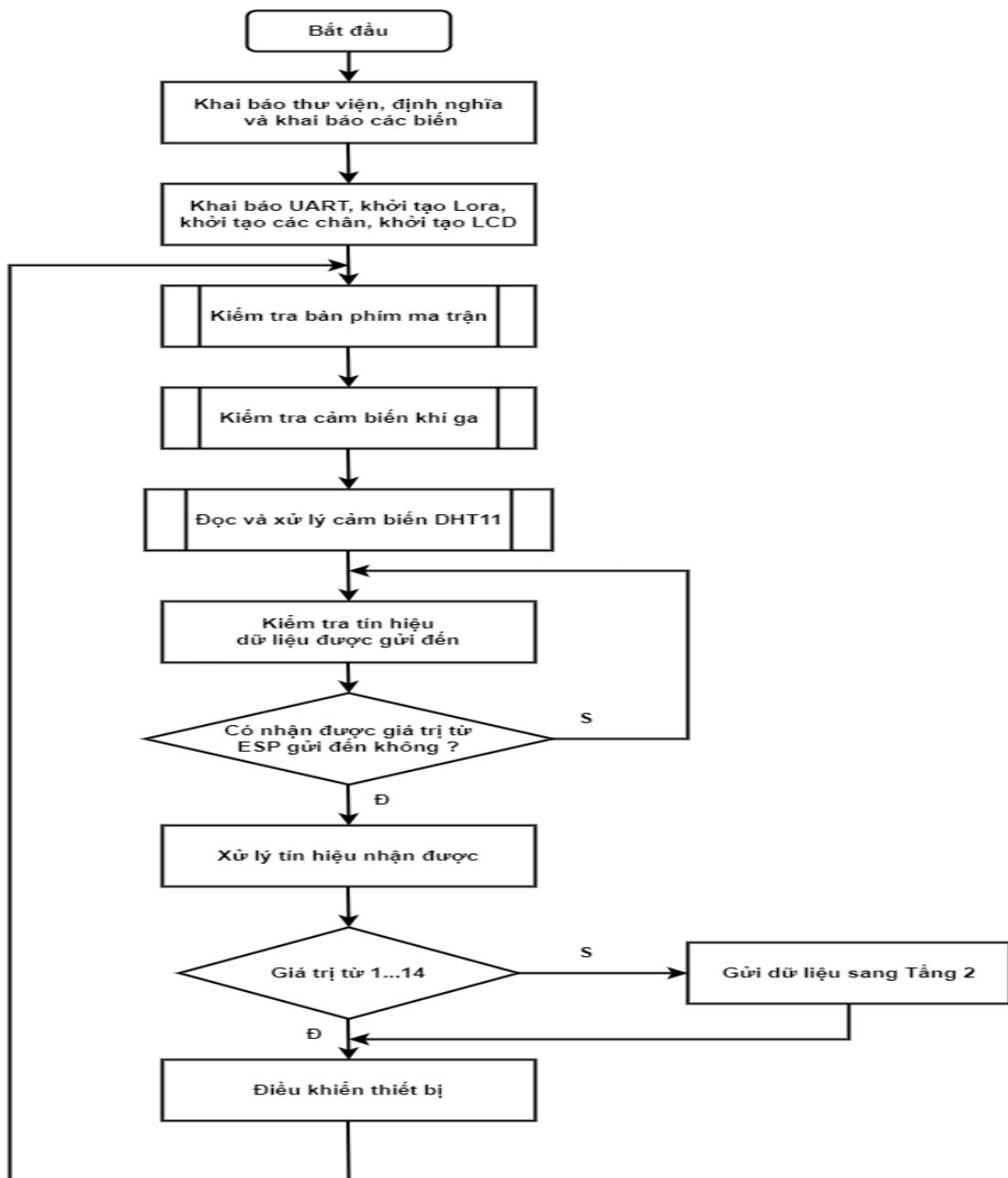
CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Ý tưởng của nhóm ở đây sẽ thi công hệ thống thành ngôi nhà 2 tầng. Mô hình được làm bằng bìa PVC – Foam.

4.4 Lập trình hệ thống

4.4.1 Lưu đồ giải thuật

Lưu đồ điều khiển chính hệ thống tầng 1



Hình 4.6: Lưu đồ điều khiển chính tầng 1

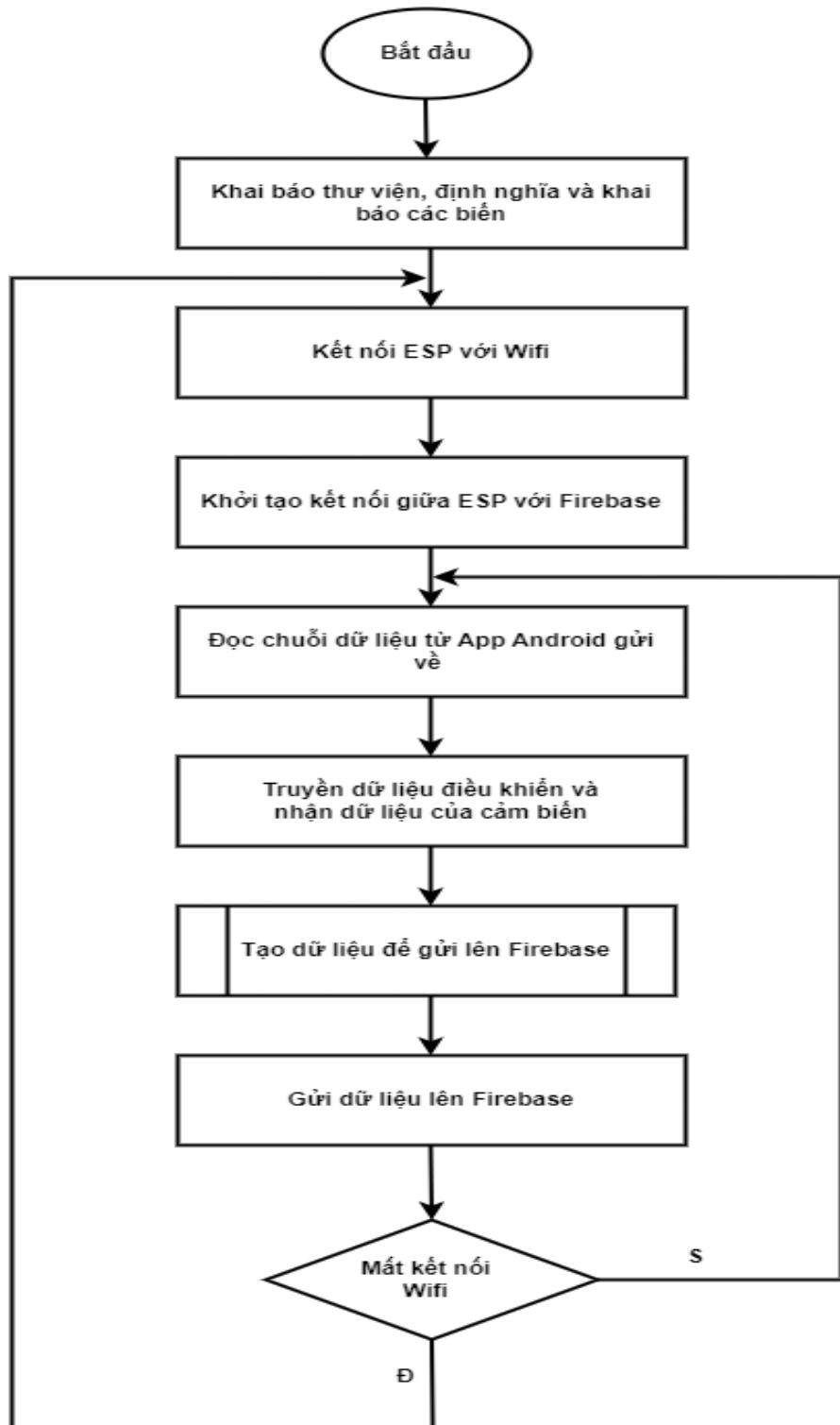
CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Giải thích lưu đồ

Đầu tiên khai báo các thư viện sử dụng, định nghĩa và khai báo các biến, khai báo tốc độ baud của UART, khởi tạo Lora và cài đặt cấu hình tần số cho Lora, khởi tạo các chân kết nối. Tiếp đến thực hiện các chương trình có trong vòng lặp:

- Kiểm tra bàn phím ma trận.
- Kiểm tra xem có khí gas rò rỉ hay không ?
- Đọc các giá trị cảm biến nhiệt độ - độ ẩm.
- Kiểm tra xem có nhận được chuỗi giá trị từ ESP gửi đến hay không, nếu có thì sẽ tiến hành xử lý tín hiệu đó (sau khi xử lý chuỗi dữ liệu đó thì nó sẽ có các giá trị từ 1 đến 22).
- Nếu có giá trị từ 1 đến 14 thì nó sẽ giữ lại để cho Arduino Mega điều khiển các thiết bị đèn quạt ở tầng 1 còn các giá trị từ 15 đến 22 thì sẽ được gửi qua tầng 2 thông qua module Lora để điều khiển các thiết bị ở tầng 2.

Lưu đồ điều khiển ESP8266



Hình 4.7: Lưu đồ điều khiển ESP8266

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

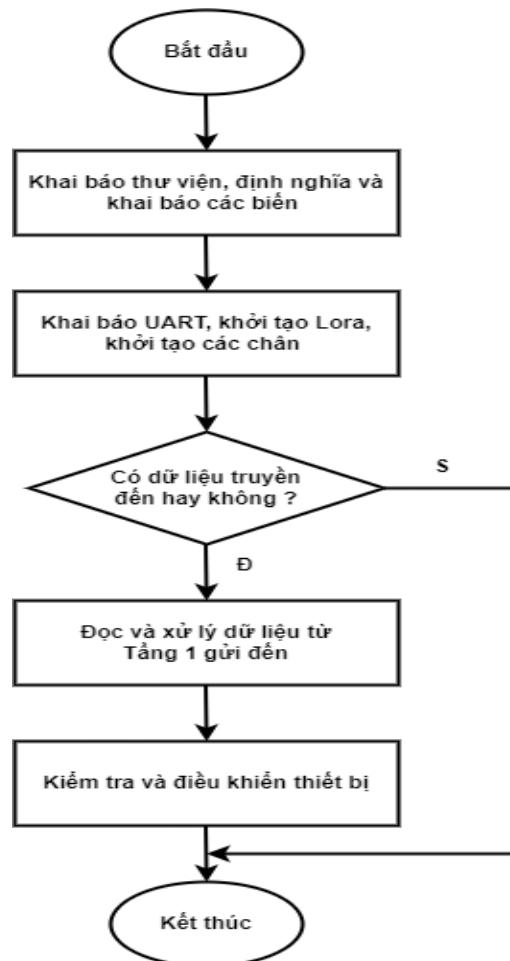
Giải thích lưu đồ

Đầu tiên khai báo các thư viện được sử dụng, định nghĩa và khai báo các biến. Sau đó thực hiện kết nối vào mạng Wifi. Khởi tạo kết nối giữa ESP8266 với Firebase.

Khi có kết nối Wifi và nhận được hành động từ bên App Android thì ESP8266 sẽ nhận các chuỗi dữ liệu để điều khiển các thiết bị và nhận các dữ liệu của cảm biến DHT11 gửi đến.

Các giá trị của cảm biến sẽ được cập nhật lên Firebase, và chuỗi dữ liệu để điều khiển thiết bị sẽ được chuyển đến Arduino Mega để xử lý và điều khiển các thiết bị tại đó.

Lưu đồ điều khiển hệ thống tầng 2

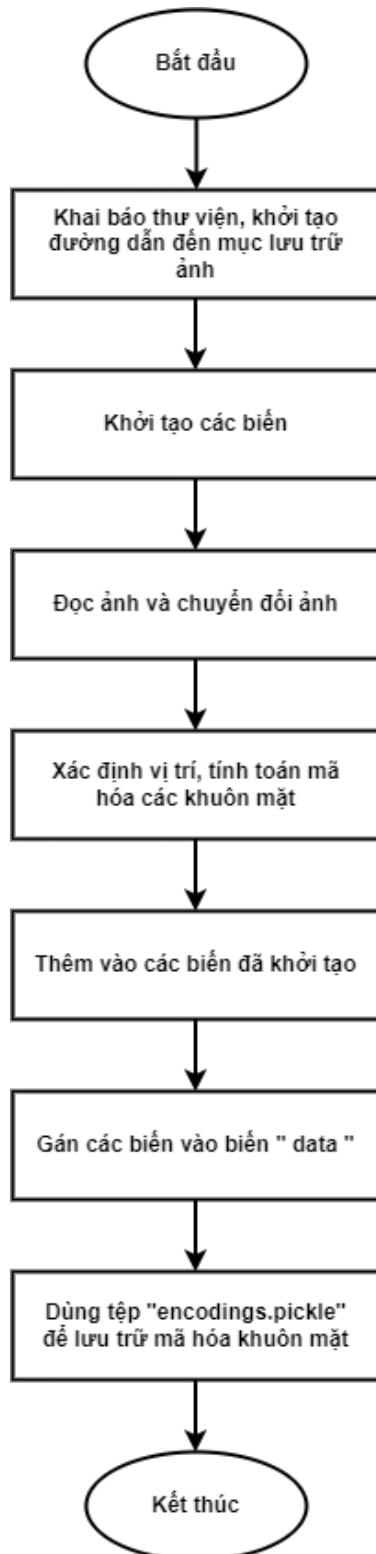


Giải thích lưu đồ:

- Khai báo các thư viện sử dụng, định nghĩa và khai báo các biến, khai báo tốc độ baud của UART, khởi tạo Lora và cài đặt cấu hình tần số cho Lora, khởi tạo các chân kết nối.
- Kiểm tra xem có dữ liệu truyền đến không. Nếu có sẽ xử lý dữ liệu đó để điều khiển các thiết bị đèn quạt tại tầng 2.

Hình 4.8: Lưu đồ điều khiển hệ thống tầng 2

Lưu đồ xây dựng mô hình nhận diện khuôn mặt



Giải thích lưu đồ:

Đầu tiên khai báo các thư viện sử dụng, tạo một danh sách các đường dẫn đến ảnh trong thư mục "dataset" (chứa các ảnh mẫu của khuôn mặt). Sau đó khởi tạo danh sách rỗng để lưu trữ các mã hóa khuôn mặt và tên của các khuôn mặt đã biết (.knownEncodings và knownNames.).

Tiếp đến dùng vòng lặp for để duyệt các từng danh sách trong đường dẫn áy:

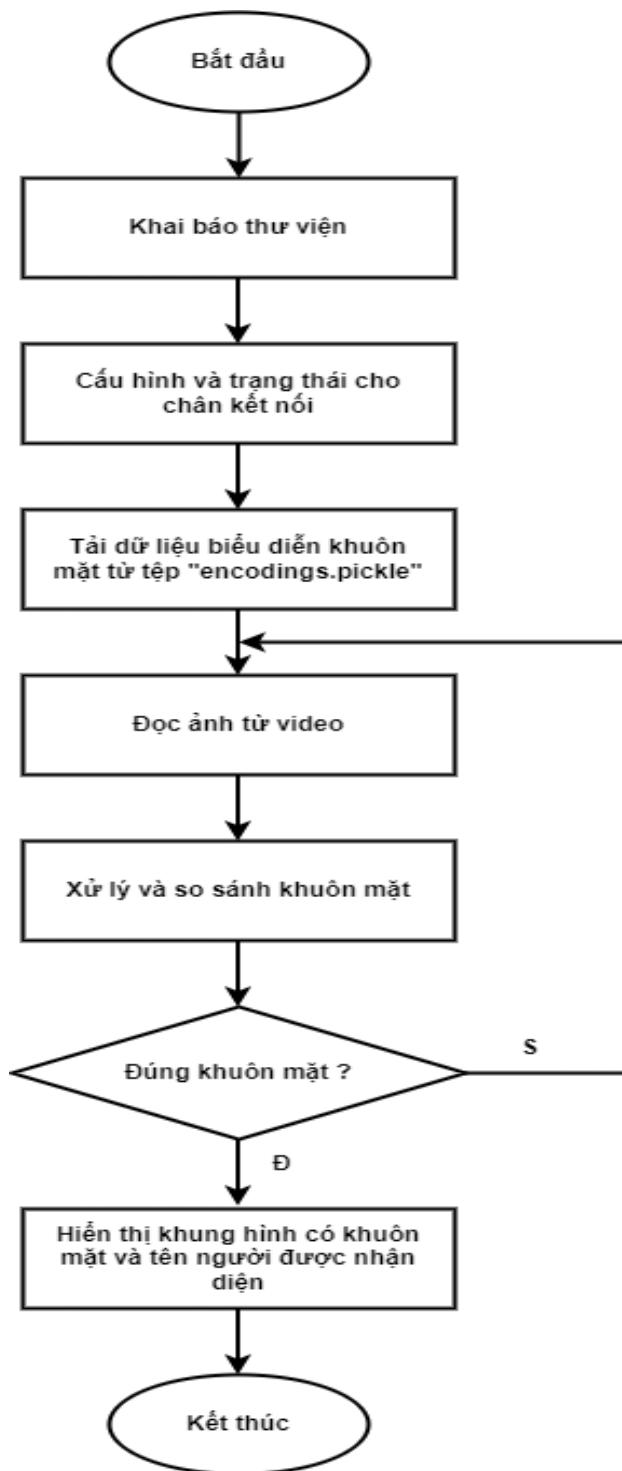
- Trích xuất từng tên người từ đường dẫn.
- Đọc ảnh.
- Chuyển đổi ảnh từ BGR sang RGB.
- Xác định vị trí khuôn mặt trong ảnh sử dụng phương pháp HOG.
- Tính toán các biểu diễn khuôn mặt cho từng khuôn mặt được phát hiện (mã hóa khuôn mặt).

Sau đó thêm biểu diễn khuôn mặt và tên từng người tương ứng vào danh sách các biểu diễn đã biết. Sau khi thêm vào các danh sách áy chúng ta tạo thêm biến "data" lưu trữ danh sách các biểu diễn khuôn mặt và danh sách các tên. Mở một tệp có tên "encodings.pickle" để lưu trữ dữ liệu và sử dụng dòng lệnh `f.write(pickle.dumps(data))` để chuyển đổi dữ liệu thành chuỗi bytes và ghi vào

Hình 4.9: Lưu đồ xây dựng mô hình nhận diện khuôn mặt

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Lưu đồ nhận diện khuôn mặt



Hình 4.10: Lưu đồ nhận diện khuôn mặt

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Giải thích lưu ý:

Đầu tiên khai báo các thư viện sử dụng, cấu hình cho chân GPIO 17 của Raspberry là đầu ra và trạng thái LOW. Sau đó chúng ta sử dụng dòng lệnh “ data = pickle.loads(open("encodings.pickle", "rb").read()) ” để tải dữ liệu biểu diễn khuôn mặt đã được lưu trữ từ tệp "encodings.pickle" và chuyển đổi chuỗi bytes thành dữ liệu Python.

Camera sẽ nhận diện khuôn mặt và so sánh xem khuôn mặt đó có giống với khuôn mặt được mã hóa trong tệp "encodings.pickle" hay không. Nếu giống thì sẽ hiển thị khung hình có khuôn mặt và tên của người được nhận diện.

4.4.2 Thiết kế giao diện trên App Android

Ở đây tài liệu này nhóm sử dụng phần mềm lập trình App Android miễn phí, dễ sử dụng đó là: App Inventor.

4.4.2.1 Tổng quan về app inventor

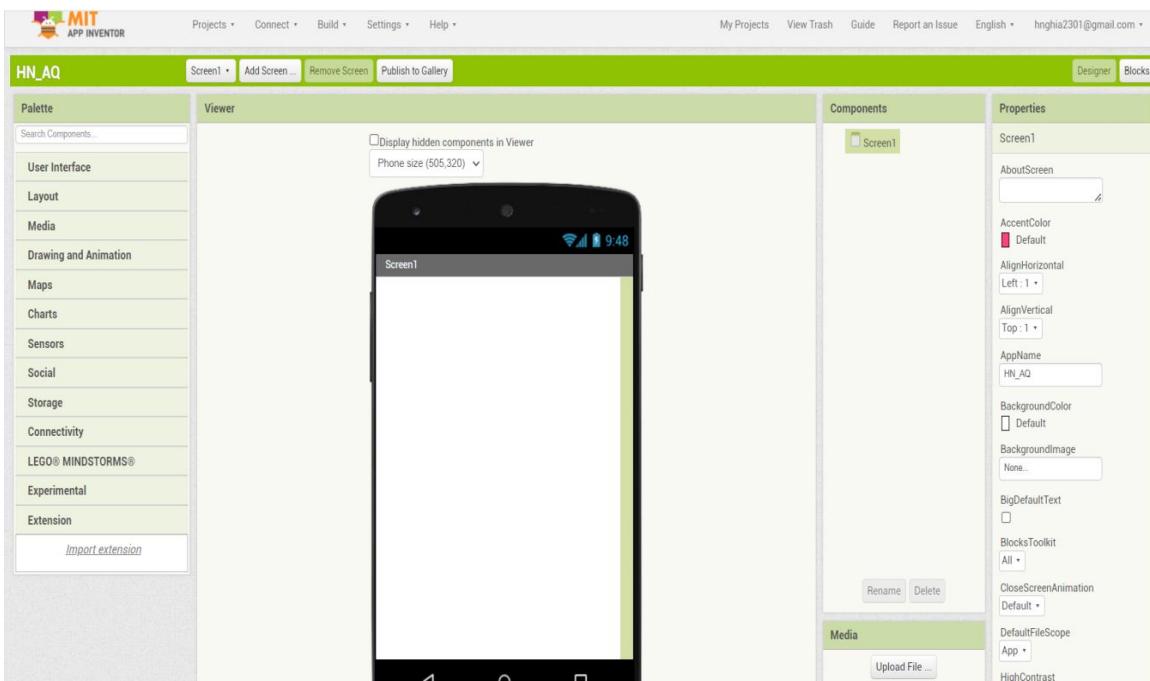
App Inventor là một nền tảng web mã nguồn mở được cung cấp ban đầu bởi Google, năm 2012 Google chuyển giao Trung tâm nghiên cứu Di động của Viện công nghệ Massachusetts (MIT) quản lý và phát triển

Môi trường lập trình App Inventor rất dễ dàng cho người mới, câu lệnh đều là những khôi lệnh trực quan được kéo thả để tạo ra các ứng dụng từ đơn giản tới phức tạp. Các ứng dụng có thể chạy trực tiếp trên các thiết bị di động như điện thoại và máy tính bảng.

4.4.2.2 Giới thiệu tổng quan về giao diện của phần mềm App Inventor 2

Giao diện làm việc của app inventor được chia làm 2 khu chính: Khu thiết kế (Designer) và khu lập trình (Blocks) [5]. Khi chọn một dự án từ danh sách từ danh sách dự án của bạn, giao diện mặc định được hiển thị là Khu thiết kế như hình dưới

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



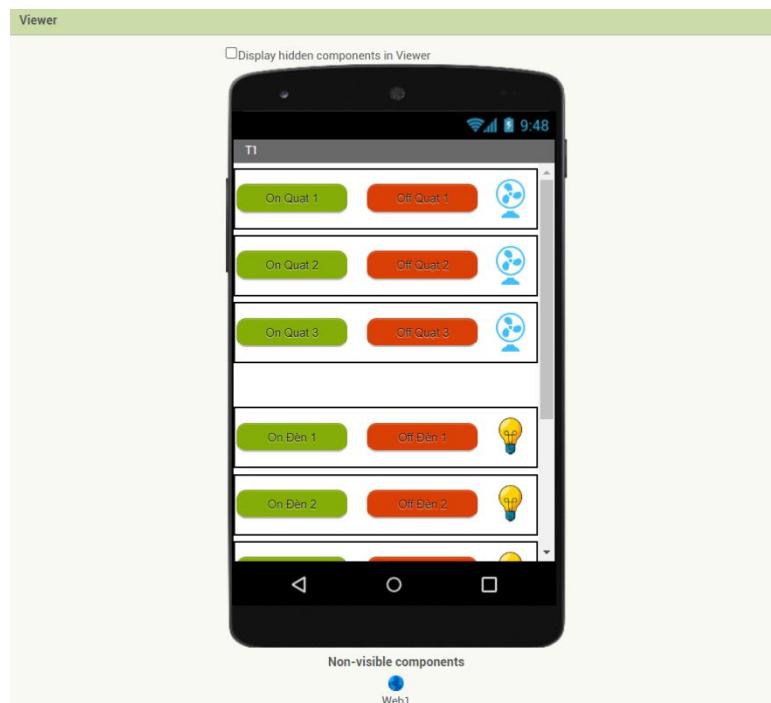
Hình 4.11: Giao diện chính của phần mềm App Inventor

Khu thiết kế có chức năng như thêm các đối tượng vào chương trình, xóa đối tượng, sắp xếp vị trí các đối tượng,... và được chia thành 4 phần: Palette, Viewer, Components và Properties [5].

- Palette: Khu vực này chứa tất cả các loại đối tượng có thể thêm vào chương trình, các đối tượng được chia thành các nhóm dựa theo chức năng:
 - Nhóm User Interface: Gồm các đối tượng giao diện như nút, ảnh, hộp thoại thông báo, nhãn,...
 - Nhóm Layout: Gồm các đối tượng hỗ trợ việc sắp xếp bố cục, vị trí các đối tượng khác.
 - Nhóm Media: Gồm các đối tượng đa phương tiện, dùng để quay phim, chụp ảnh, ghi âm, chơi nhạc,...
 - Nhóm Drawing and Animation: Gồm các đối tượng dùng để vẽ và mô phỏng chuyển động.
 - Nhóm Sensor: Gồm các đối tượng cảm biến như đồng hồ, cảm biến gia tốc, cảm biến vị trí, cảm biến tiệm cận,...

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

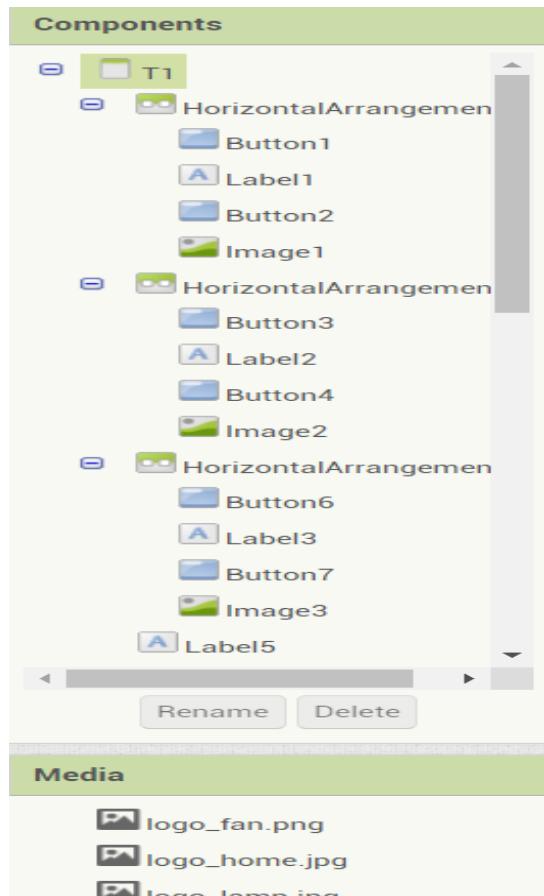
- Nhóm Social: Gồm các đối tượng hỗ trợ việc liên lạc và chia sẻ như gọi điện, nhắn tin, Twitter,...
 - Nhóm Storage: Gồm các đối tượng việc lưu trữ dữ liệu như tệp tin, cơ sở dữ liệu,...
 - Nhóm Connectivity: Gồm các đối tượng hỗ trợ việc kết nối như web, Bluetooth,...
 - Nhóm LEGO MINDSTORM: Gồm các đối tượng hỗ trợ điều khiển robot LEGO MINDSTORMS NXT.
 - Nhóm Experimental: Cung cấp đối tượng giao tiếp với nền tảng Firebase của Google.
 - Nhóm Extension: Hỗ trợ thêm các đối tượng mở rộng
- Viewer: Khu vực này hiển thị các đối tượng trên màn hình điện thoại mô phỏng. Để thêm một đối tượng vào chương trình, nhấn giữ loại đối tượng muốn thêm bên khu Palette và kéo sang khu Viewer. Chúng ta có thể dùng chuột kéo thả để sắp xếp vị trí các đối tượng [5].



Hình 4.12: Khu vực Viewer

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

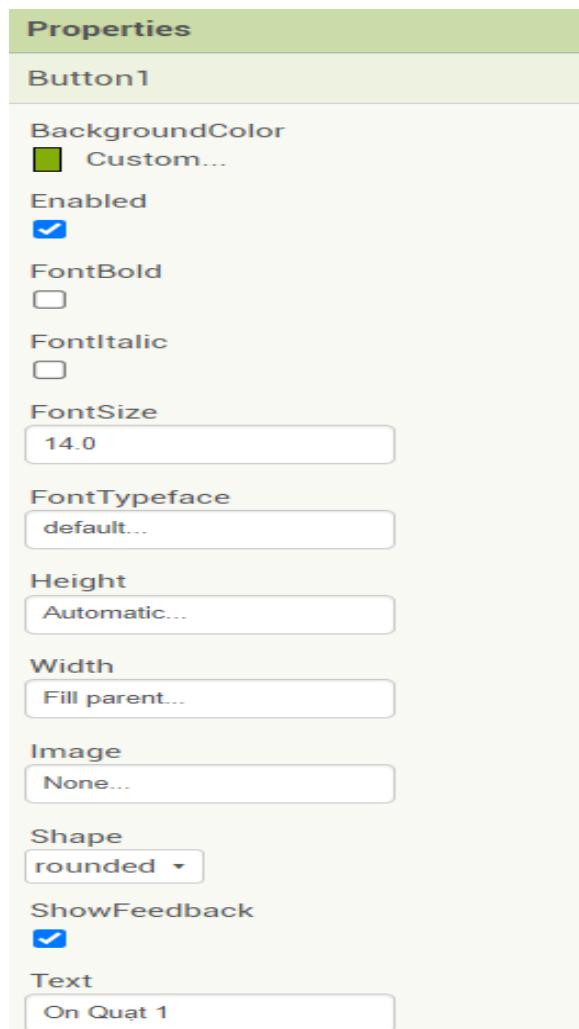
- Components: Khu vực này chứa danh sách các đối tượng đã được thêm vào chương trình. Chúng ta có thể đổi tên và xóa các đối tượng.



Hình 4.13: Khu vực Components

- Properties: Khu vực này chứa các thuộc tính (đặc tính, thông tin, thông số) của đối tượng được chọn ở khu Components. Mỗi loại đối tượng khác nhau có thể có các thuộc tính giống nhau hoặc khác nhau. Ví dụ đối tượng Button (Button1) và Image (Image1) như hình dưới đều có thuộc tính Width (chiều rộng) và Height (chiều cao), nhưng đối tượng Button có thêm thuộc tính BackgroundColor (màu nền) trong khi đối tượng Image không có thuộc tính này [5].

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 4.14: Khu vực Properties

Khu lập trình cung cấp các khối lệnh để lập trình các đối tượng có trong chương trình. Khu vực được chia thành 2 phần chính: Blocks và Viewer

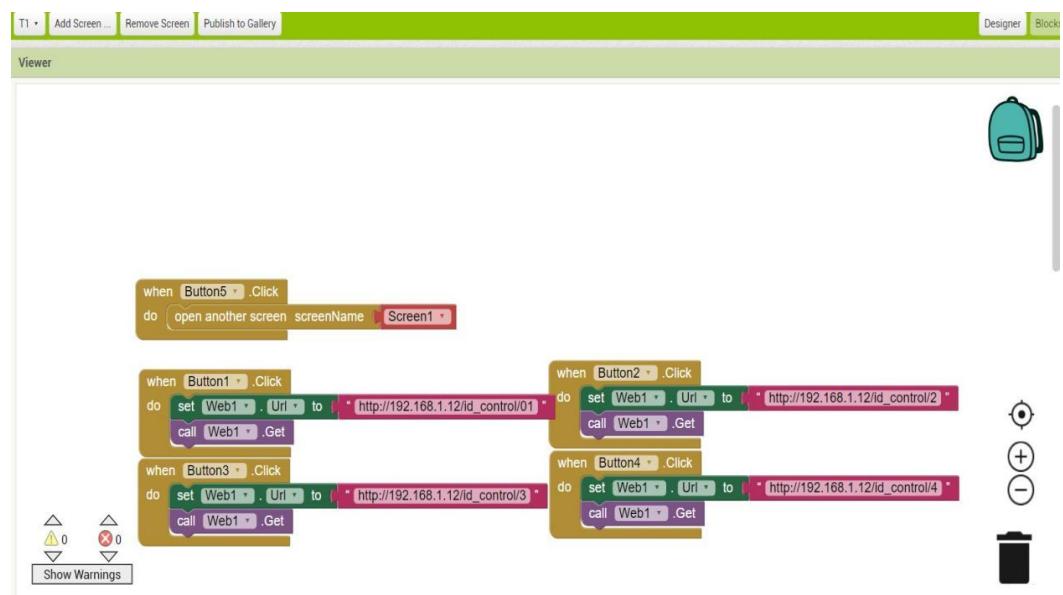
- Blocks: khu vực này chứa danh sách các khối lệnh, được phân thành 2 nhóm chính: nhóm khối lệnh có sẵn Built-in và nhóm khối lệnh bổ sung. Built-in là nhóm chứa các khối lệnh cơ bản. Nhóm khối lệnh bổ sung chứa các khối lệnh của các đối tượng trong dự án. Nhấn vào một đối tượng bất kỳ, danh sách các khối lệnh của đối tượng đó xuất hiện [5].

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 4.15: Phần Block của khu vực lập trình

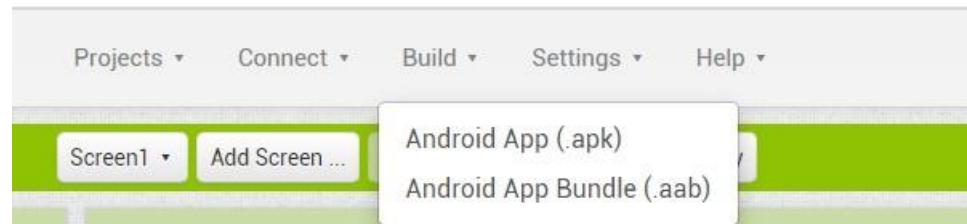
- Viewer: Đây là nơi lập trình viên kéo các khối lệnh từ danh sách các khối lệnh và ghép chúng lại với nhau. Các khối lệnh sẽ được thực hiện theo thứ tự mà lập trình viên đã ghép [5].



Hình 4.16: Phần Viewer của khu vực lập trình

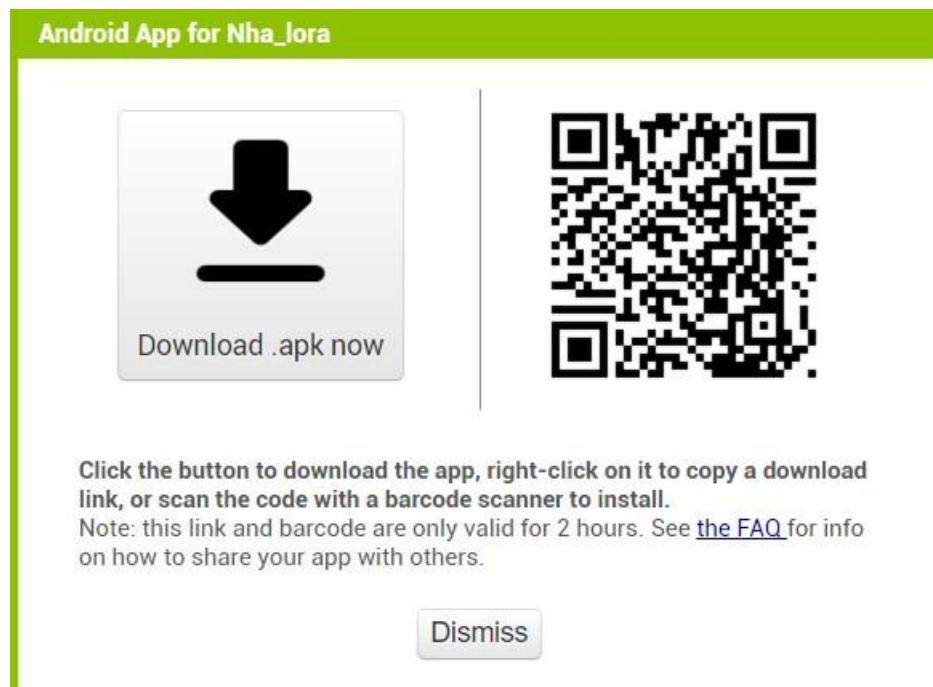
CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Sau khi thiết kế giao diện và lập trình xong thì xuất thành bộ cài đặt trên thiết bị di động. Để có thể tải app đã lập trình về thì các bạn lựa chọn Android App (.apk) trong phần Build trên thanh mục như hình bên dưới.



Hình 4.17: Lựa chọn Android App (.apk) trong phần Build

Sau đó màn hình sẽ hiện 2 kiểu tải về như hình dưới, tải trực tiếp hoặc tải bằng mã QR.

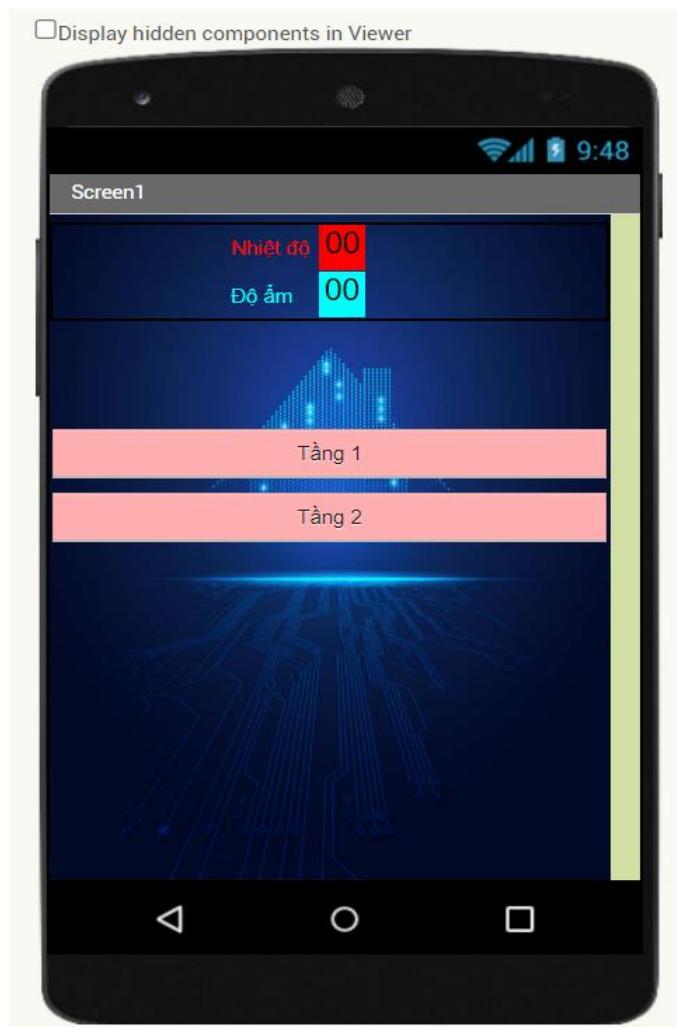


Hình 4.18: Chọn cách tải về theo 2 cách

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

4.4.2.3 Thiết kế và lập trình app cho đồ án

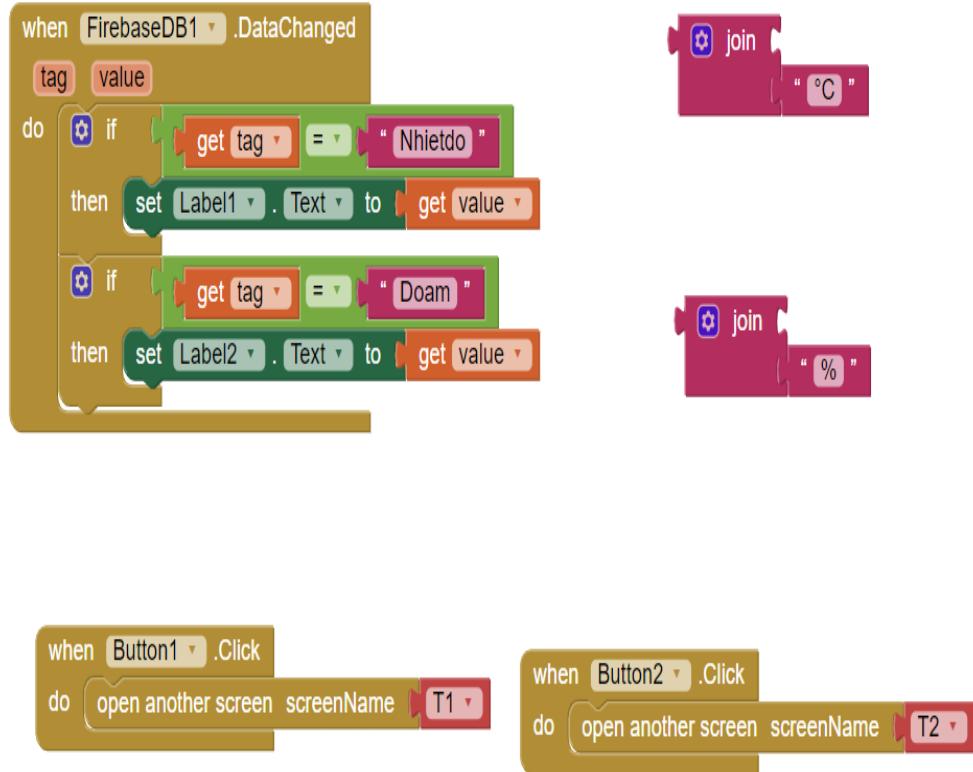
Bằng cách kéo thả các nhãn ở khu vực Palette ta sắp xếp giao diện theo chủ ý của mình và kéo thả các lệnh tương ứng của khu vực lập trình để khởi tạo. Hình dưới đây là giao diện Screen1 và code cho màn hình Screen1.



Hình 4.19: Giao diện home (Screen1)

Theo yêu cầu đặt ra, App Android sẽ hiển thị giá trị của cảm biến nhiệt độ - độ ẩm, lên giao diện Home, và các nút nhấn T1 và T2 để chuyển sang giao diện của tầng 1 và tầng 2. Đoạn code hình 4.25 là đoạn code cho giao diện Home.

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



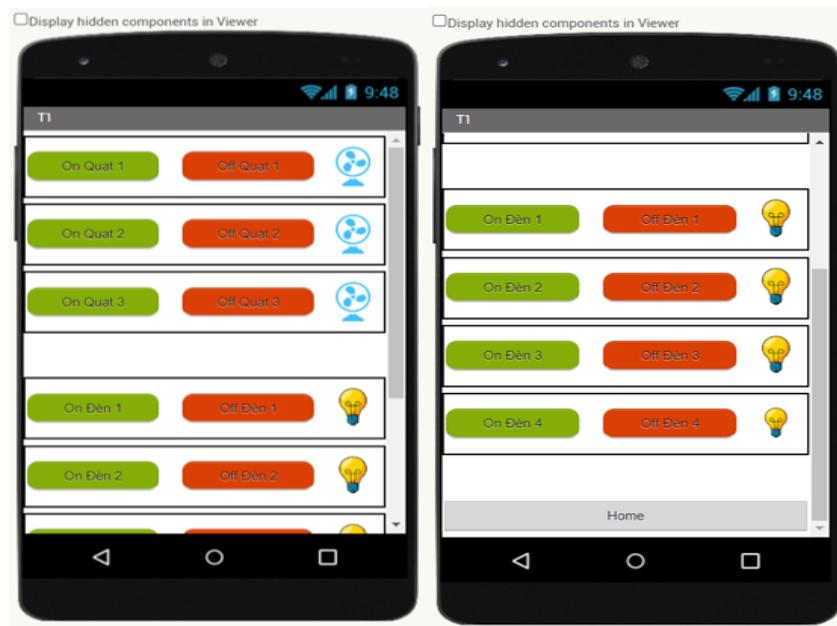
Hình 4.20: Code của giao diện home

Bằng cách kéo thả các nhãn ở khu vực Palette ta sắp xếp giao diện theo chủ ý của mình và kéo thả các lệnh tương ứng của khu vực lập trình để khởi tạo. Hình 4.26 và 4.28 là giao diện T1 với T2

Theo yêu cầu đặt ra ban đầu, app điều khiển sẽ điều khiển các thiết bị đèn quạt tại tầng 1 và tầng 2, hình 4.27 và hình 4.29 là code cho giao diện T1 với giao diện T2.

Lưu ý: Địa chỉ IP sẽ thay đổi khi sử dụng mạng internet khác để điều khiển thiết bị.

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 4.21: Giao diện tầng 1

```

when Button5.Click
do open another screen screenName Screen1

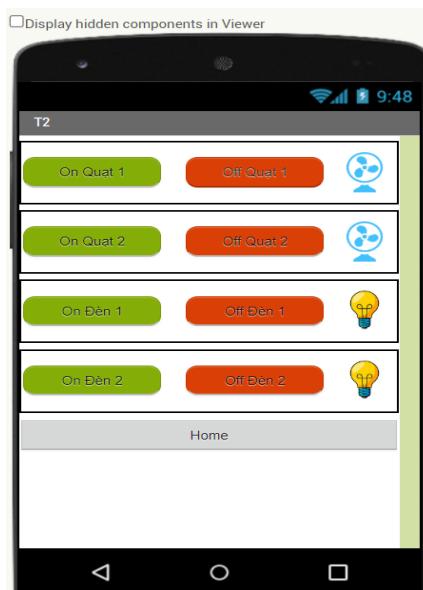
when Button1.Click
do set Web1.Url to "http://192.168.1.4/id_control/01"
call Web1.Get
when Button3.Click
do set Web1.Url to "http://192.168.1.4/id_control/3"
call Web1.Get
when Button2.Click
do set Web1.Url to "http://192.168.1.4/id_control/2"
call Web1.Get
when Button4.Click
do set Web1.Url to "http://192.168.1.4/id_control/4"
call Web1.Get

when Button6.Click
do set Web1.Url to "http://192.168.1.4/id_control/5"
call Web1.Get
when Button7.Click
do set Web1.Url to "http://192.168.1.4/id_control/6"
call Web1.Get
when Button10.Click
do set Web1.Url to "http://192.168.1.4/id_control/7"
call Web1.Get
when Button11.Click
do set Web1.Url to "http://192.168.1.4/id_control/8"
call Web1.Get
when Button12.Click
do set Web1.Url to "http://192.168.1.4/id_control/9"
call Web1.Get
when Button13.Click
do set Web1.Url to "http://192.168.1.4/id_control/10"
call Web1.Get
when Button14.Click
do set Web1.Url to "http://192.168.1.4/id_control/11"
call Web1.Get
when Button15.Click
do set Web1.Url to "http://192.168.1.4/id_control/12"
call Web1.Get
when Button8.Click
do set Web1.Url to "http://192.168.1.4/id_control/13"
call Web1.Get
when Button9.Click
do set Web1.Url to "http://192.168.1.4/id_control/14"
call Web1.Get

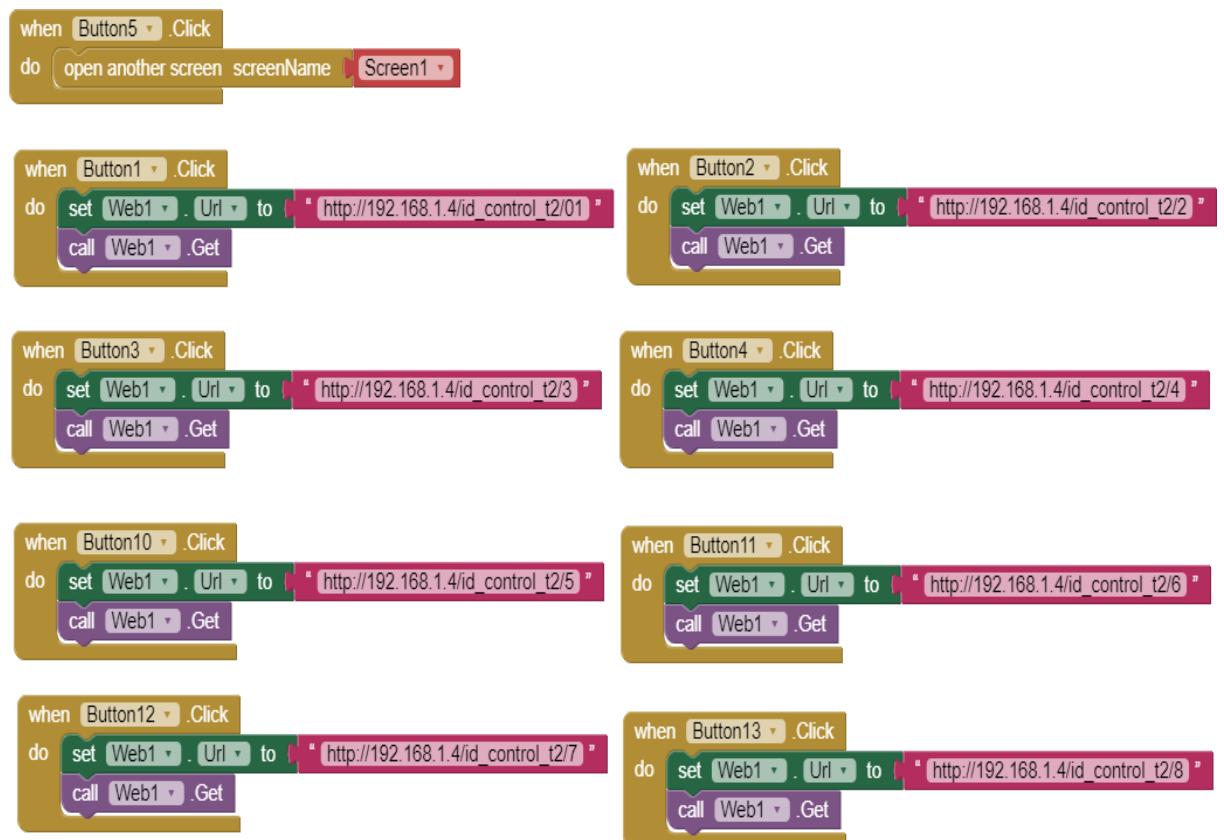
```

Hình 4.22: Code giao diện tầng 1

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



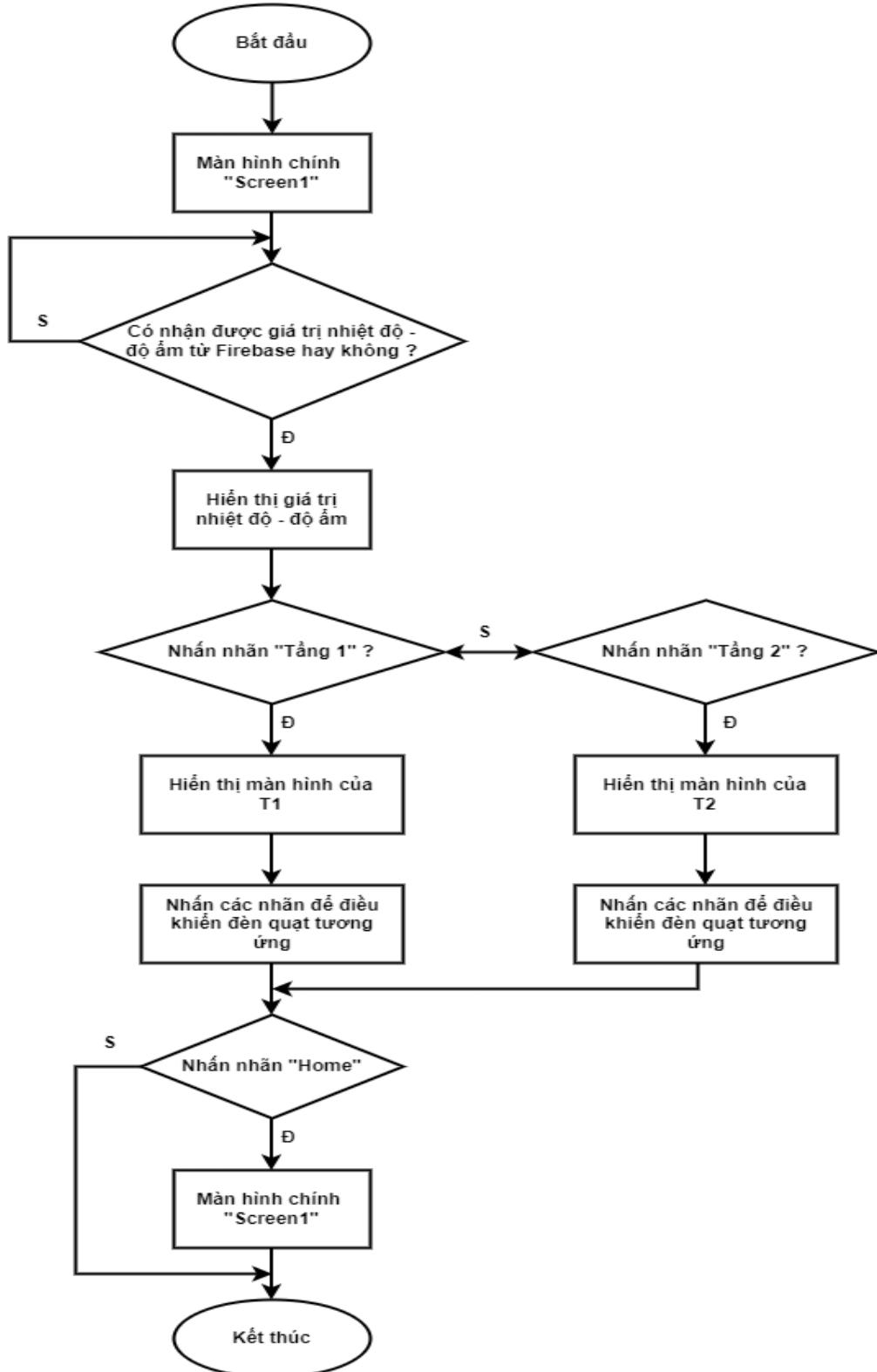
Hình 4.23: Giao diện tầng 2



Hình 4.24: Code giao diện tầng 2

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

4.4.2.3 Lưu đồ giải thuật App Android



Hình 4.25: Lưu đồ giải thuật App Android

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

Đầu tiên hiển thị màn hình chính “Screen1”, giá trị nhiệt độ độ ẩm sẽ hiển thị và cập nhật tại màn hình chính

Nếu nhấn nút tầng 1 thì màn hình sẽ chuyển qua giao diện T1, ở đó ta sẽ nhấn vào các biểu tượng nhãn tương ứng để điều khiển quạt đèn của tầng 1.

Nếu nhấn nút tầng 2 thì màn hình sẽ chuyển qua giao diện T2, ở đó ta sẽ nhấn vào các biểu tượng nhãn tương ứng để điều khiển quạt đèn của tầng 2.

Nhấn home để trở về màn hình chính “Screen1”.

4.5 Hướng dẫn sử dụng

Đầu tiên ta tải phần mềm điều khiển thiết bị thông qua mã QR sau:



Hình 4.26: Mã QR để tải app

Tiến hành cài đặt app lên điện thoại giao diện giống với mục trên. Cắm nguồn cho mô hình để khởi động hệ thống. Sau đó nhấn các biểu tượng trên app Android để điều khiển đèn và quạt của từng tầng. Các giá trị của cảm biến nhiệt độ độ ẩm sẽ hiện thị tại màn hình chính “Screen1”. Đối với cảm biến khí gas, nếu có khí thì còi sẽ báo, nhấn button để tắt còi báo.

Mở cửa bằng 2 chế độ:

- Chế độ nhập mật khẩu: Nhập “123” nhấn nút # để mở khóa, để xóa 1 số thì nhấn nút * (nút * còn có chức năng để đóng cửa).
- Chế độ nhận diện bằng camera thì cắm dây cáp vào raspberry tìm ip tĩnh xong chạy code face_door (dưới phụ lục) để mở camera. Sau đó sẽ nhận diện khuôn mặt nếu đúng thì mở cửa.

Chương 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

5.1 Giới thiệu

Chương này trình bày kết quả của cả quá trình nghiên cứu làm đề tài trong 15 tuần, đồng thời là các nhận xét, đánh giá về đề tài.

5.2 Kết quả đạt được

Sau khi quá trình thực hiện đề tài, nhóm chúng em đã hoàn thành các mục tiêu đề tài, có thêm nhiều kiến thức và thực hiện được các tính năng sau:

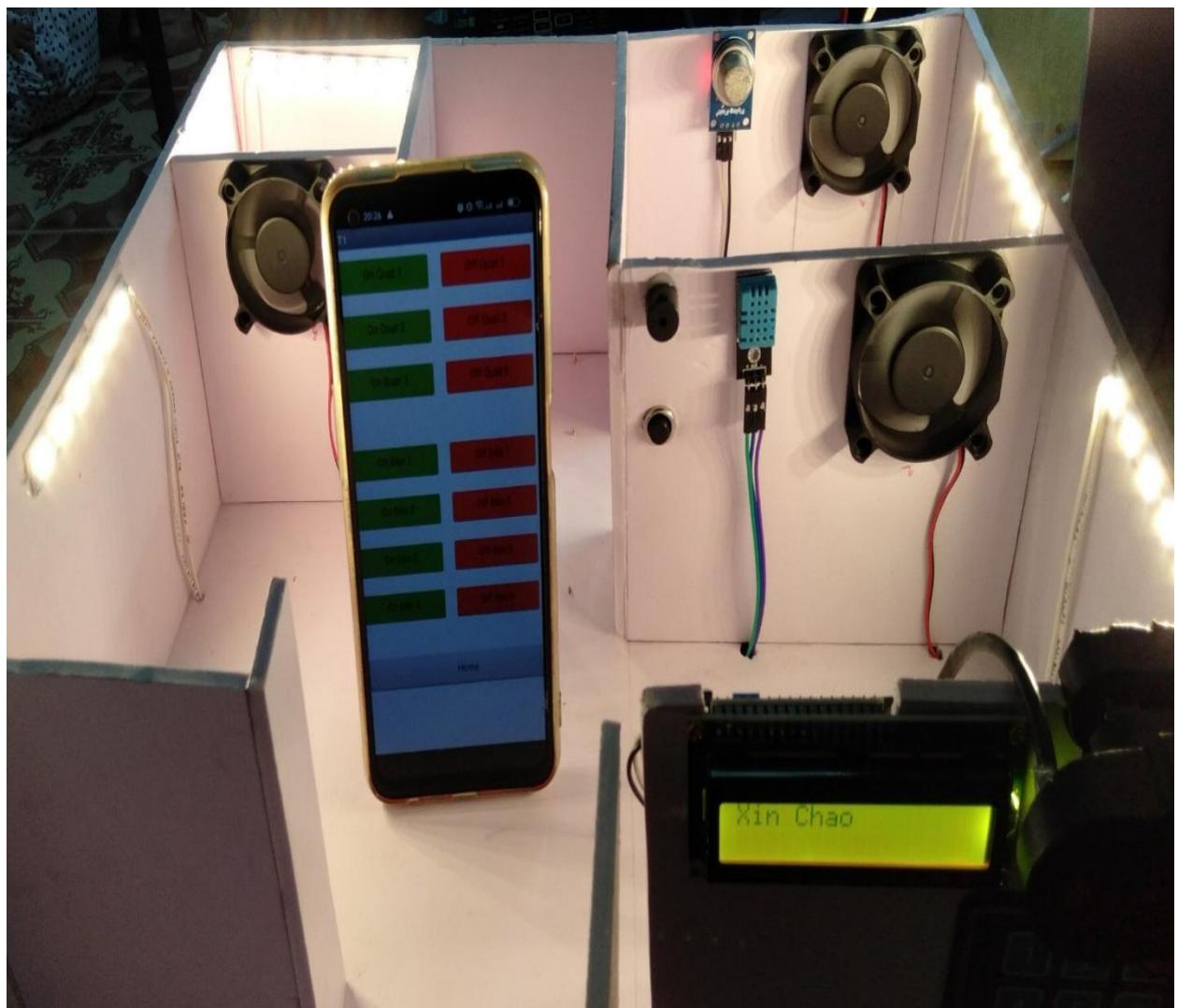
- Hiểu biết sâu hơn về sử dụng Board Arduino, ứng dụng hiệu quả board mạch này trong ĐATN.
- Tìm hiểu và ứng dụng được các mô đun Wifi ESP8266, Lora SX1278.
- Biết cách truyền - nhận dữ liệu giữa nhiều Lora (cụ thể là 2 trong ĐATN).
- Tìm hiểu và ứng dụng được xử lý ảnh để nhận diện khuôn mặt.
- Tìm hiểu và ứng dụng được Raspberry Pi.
- Thiết kế được giao diện Android để điều khiển, giám sát thiết bị.
- Điều khiển bật, tắt các thiết bị điện qua mạng Wifi và Lora.
- Tính toán, thiết kế, chọn linh kiện để hoàn thiện được board mạch chính đủ dòng áp chạy đáp ứng được yêu cầu đề tài.
- Tiếp cận sử dụng được nhiều phần mềm mới: Spyder, App Inventor 2.
- Giám sát được các giá trị nhiệt độ, độ ẩm thông qua App Android.
- Có còi buzzer báo động kêu to, rõ khi khí gas, khói.
- Hệ thống mở cửa bằng 2 cách: nhận diện khuôn mặt và nhập mật khẩu.
- Thi công và đóng gói thành công mô hình.
- Lập trình cho hệ thống hoạt động ổn định qua nhiều lần kiểm tra, xử lý triệt để các tình huống mạch hoạt động sai so với yêu cầu. Cảm biến hoạt động chính xác, đáp ứng trong các trường hợp sự cố nhanh, hiệu quả.
- Giao diện điều khiển trên App Android dễ sử dụng.

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

5.3 Nhận xét – Đánh giá

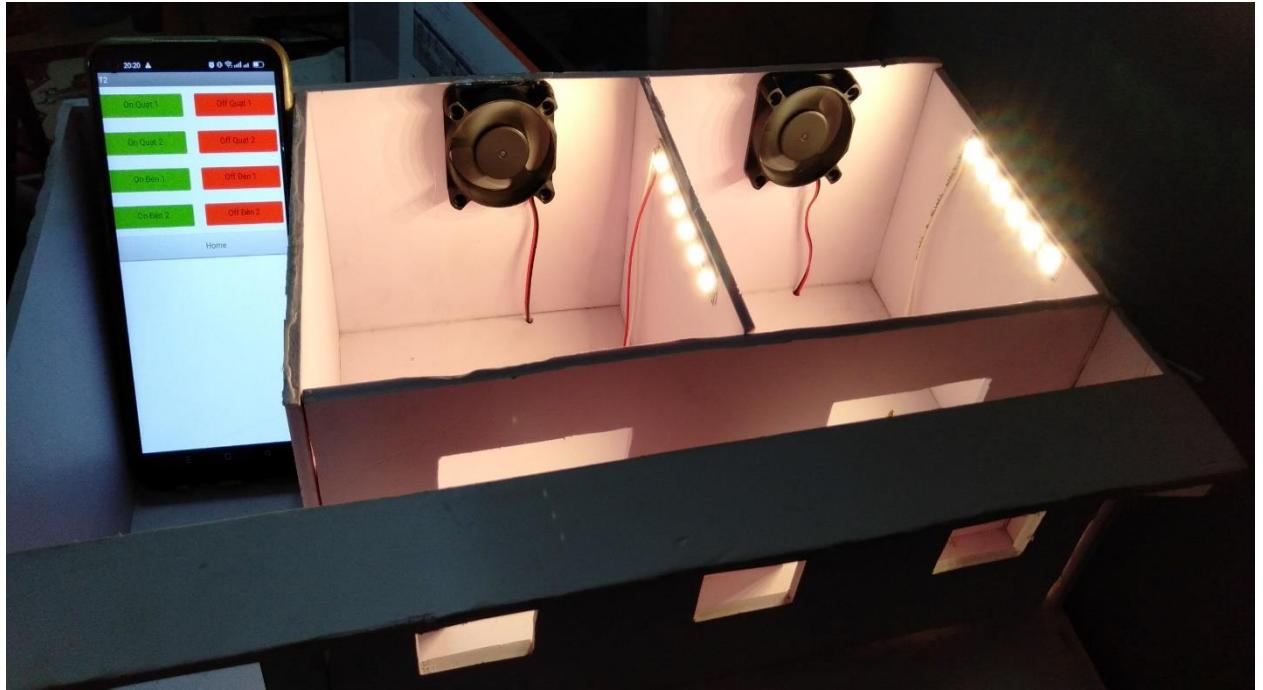
Mô hình hệ thống sau khi hoàn thành đạt đầy đủ chức năng cơ bản, hoạt động khá ổn định. Bên dưới là một số kết quả chạy thử cho toàn bộ hệ thống.

Hình 5.1 và hình 5.2 là kết quả điều khiển bật đèn và quạt của 2 tầng thông qua App điều khiển.



Hình 5.1: Điều khiển đèn quạt tại tầng trệt

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



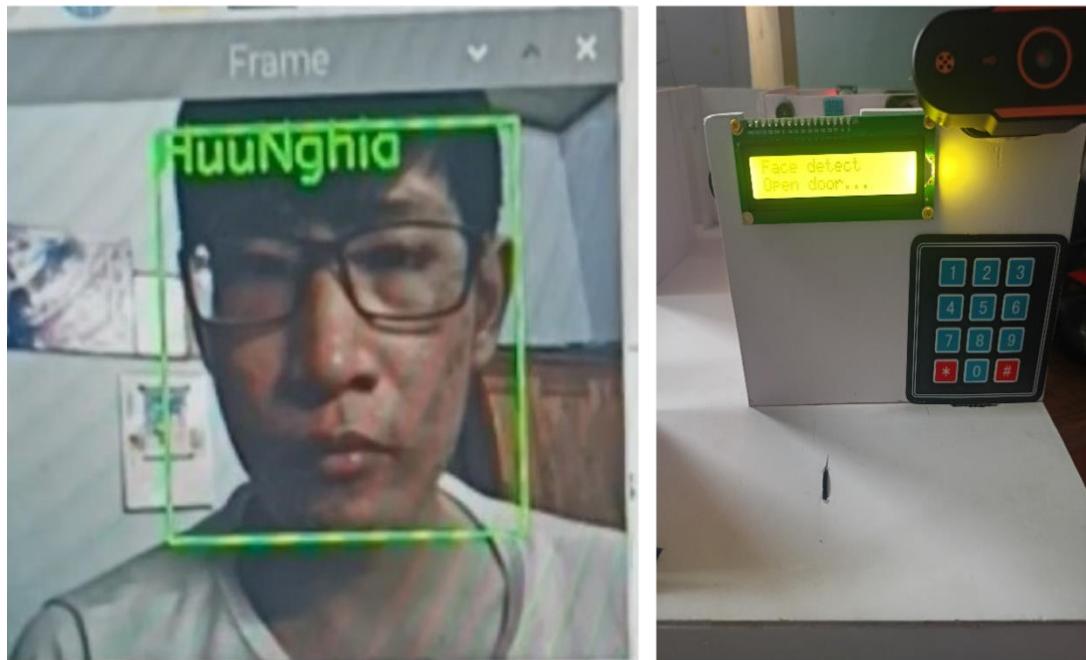
Hình 5.2: Điều khiển đèn quạt tại tầng 2

Và hình 5.3 và 5.4 là kết quả mở cửa bằng 2 cách: nhập mật khẩu và nhận diện khuôn mặt



Hình 5.3: Mở cửa bằng cách nhập mật khẩu

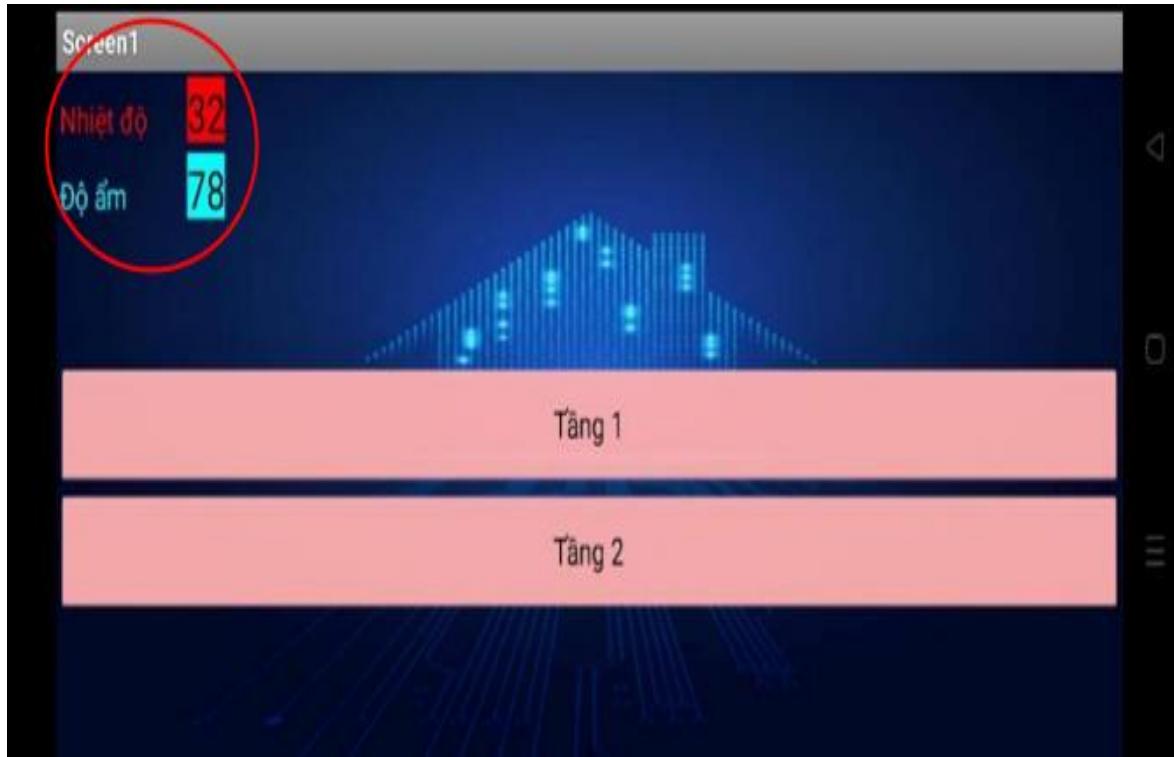
CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 5.4: Mở cửa bằng nhận diện khuôn mặt

Hình 5.5 bên dưới thể hiện giá trị nhiệt độ độ ẩm được cập nhật và hiện thị lên giao diện chính của App điều khiển.

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ



Hình 5.5: Giá trị nhiệt độ - độ ẩm được hiển thị lên điện thoại
Dưới đây là bảng thống kê chạy thử nghiệm hệ thống sau 30 lần

Bảng 5.1: Bảng đánh giá các thiết bị điện của hệ thống

	Thiết bị	Số lần thực hiện thành công	Đánh giá
Tầng 1	Đèn	30 lần	Đạt
	Quạt	30 lần	Đạt
	DHT11	30 lần	Đạt
	MQ-2	30 lần	Đạt
Tầng 2	Đèn	27 lần	Thời gian phản hồi 3 - 4 giây
	Quạt	27 lần	Thời gian phản hồi 3 - 4 giây

Bảng 5.2: Bảng đánh giá số liệu thực hiện của hệ thống cửa

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT – ĐÁNH GIÁ

	Số lần thực hiện thành công	Đánh giá
Nhập mật khẩu	30 lần	Đạt
Nhận diện khuôn mặt	30 lần	Đạt

Sau quá trình vận hành thử hệ thống, nhóm có những đánh giá sau đây:

Hệ thống hoạt động khá ổn định, đạt được những mục tiêu đề ra ban đầu. Mô hình có tính thẩm mỹ, giao diện app điều khiển thiết bị dễ sử dụng. Tuy nhiên thời gian đáp ứng điều khiển thiết bị tại tầng 2 vẫn chưa nhanh lăm (thời gian từ lúc nhấn nút điều khiển đến lúc thiết bị được bật trung bình từ 3 đến 4 giây) do truyền nhận dữ liệu từ tầng trệt sang.

Chương 6. KẾT LUẬN - HƯỚNG PHÁT TRIỂN

6.1 Kết luận:

Sau khoảng 15 tuần nghiên cứu và tìm hiểu, nhóm đã hoàn thành đồ án và thi công mô hình theo yêu cầu đã đặt ra ban đầu. Trong quá trình thực hiện, nhóm đã thu được những kết quả nhất định.

- Hệ thống hoạt động ổn định
- Mô hình có tính thẩm mỹ, gọn nhẹ.
- Mô hình hoạt động đạt yêu cầu và hoàn thành đúng thời gian quy định.
- Giao diện app điều khiển thiết bị trực quan dễ sử dụng.

6.2 Hướng phát triển:

Một số hướng phát triển để hoàn thành đề tài:

- Thêm mạch công suất lớn để có thể điều khiển nhiều tải công suất lớn.
- Thêm pin phòng trường hợp mất điện.
- Tăng thêm số lượng tầng bằng cách tăng thêm số lượng node.

TÀI LIỆU THAM KHẢO

Sách tham khảo

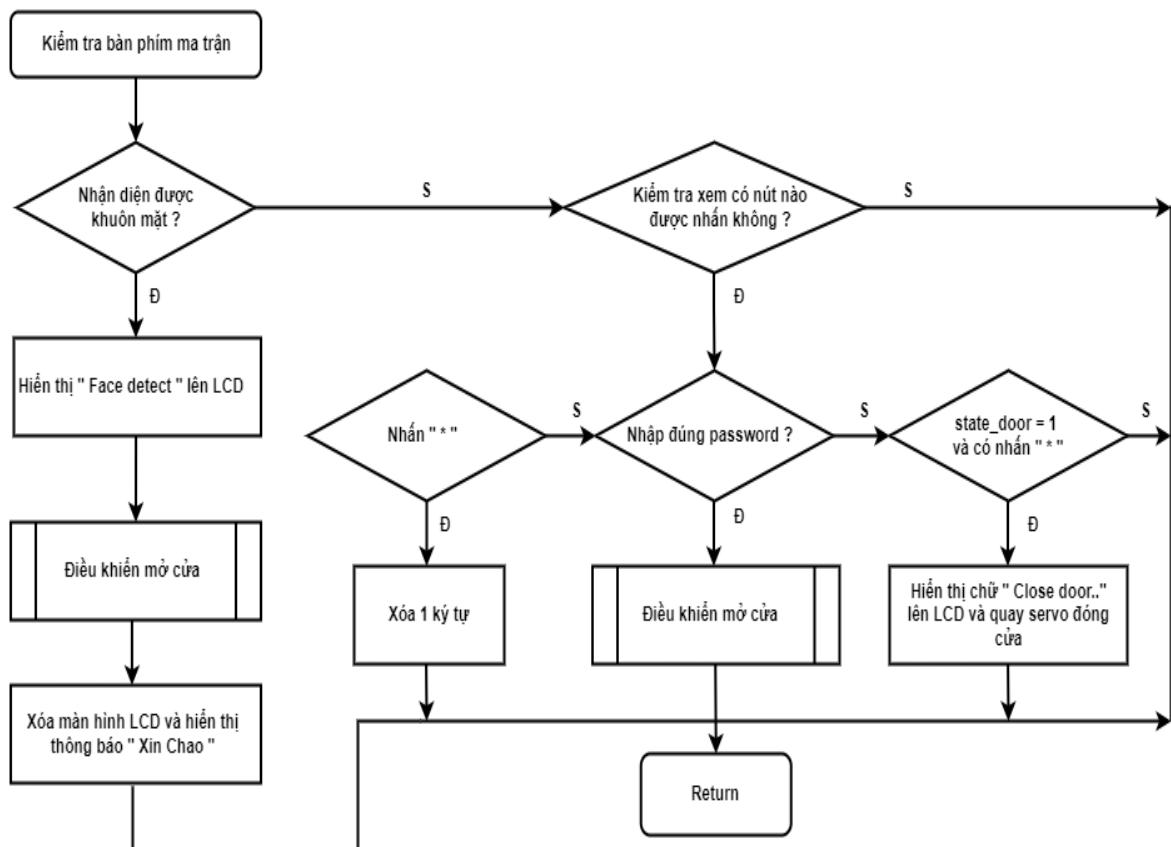
- [1] Nguyễn Đức Dũy, “Đồ án – Thiết kế và thi công hệ thống giám sát và điều khiển thiết bị trong nhà kết hợp Wifi và Lora”, Trường Đại Học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh, 2018.
- [2] Lê Đại Minh, Trương Đức Hoàng Long, “Đồ án – Thiết kế và thi công hệ thống điều khiển, giám sát thiết bị trong nhà sử dụng công nghệ Lora”, Trường Đại Học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh, 2018.
- [3] Mai Xuân Phú, “Luận văn – Nghiên cứu kỹ thuật nhận dạng khuôn mặt người. Xây dựng hệ thống camera chống trộm”, Đại học Đà Nẵng, 2017.
- [4] Trần Thu Hà, “Điện tử cơ bản”, NXB ĐH Quốc Gia Tp.HCM, 2013.
- [5] Nguyễn Hữu Hưng, “Lập trình ứng dụng di động với App Inventor”, NXB ĐH Quốc Gia Hà Nội, 2020.
- [6] Nguyễn Việt Hùng, Nguyễn Ngô Lâm, Nguyễn Văn Phúc, Đặng Phuoc Hải Trang, “Giáo trình Truyền Sô Liệu”, NXB Đại Học Quốc Gia Tp.HCM, 2013.

Website tham khảo

- [1] <https://hoangantech.github.io>
- [2] <https://github.com/>
- [3] <https://iotcircuithub.com/lora-esp8266-arduino-iot-project/>
- [4] <http://www.appinventor.org/>
- [5] <http://www.appinventor.org/book2>
- [6] <https://circuitdigest.com/microcontroller-projects/arduino-lora-sx1278-interfacing-tutorial>
- [7] <https://huongdan.cytrontech.vn/ket-noi-vnc-truc-tiep-tu-laptop-pc-voi-raspberry-pi-voi-day-cap-mang.html>
- [8] huynhnhattung.com
- [9] [GitHub - Dedepya/Face-Recognition-Using-SVM: Code for a face recognition engine based on OpenCV to detect faces via a live webcam feed](https://github.com/Dedepya/Face-Recognition-Using-SVM)

PHỤ LỤC

Lưu đồ chương trình con kiểm tra bàn phím ma trận



Hình 1: Lưu đồ kiểm tra bàn phím ma trận

Giải thích lưu đồ:

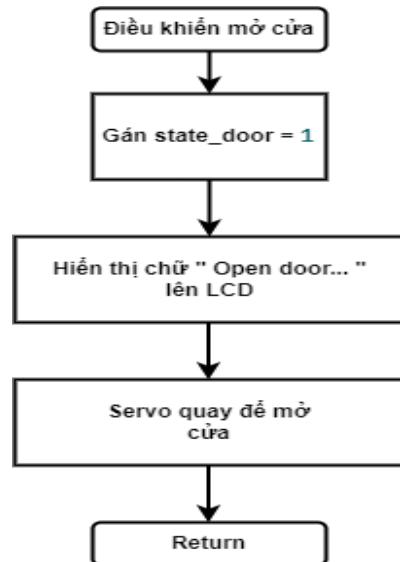
Đầu tiên kiểm tra xem Raspberry có đang hoạt động không ? (nếu nhận diện được khuôn mặt), nếu có thì sẽ hiển thị lên LCD dòng chữ “Face detect” và cửa sẽ mở sau đó sẽ xóa màn hình LCD và hiển thị thông báo “Xin Chao”.

Trong trường hợp không nhận diện được khuôn mặt thì ta có thể mở cửa bằng cách nhập đúng password được lập trình (ở đây password là “123”). Chúng ta kiểm tra xem có nút nào được nhấn hay không ?

- Nếu như chúng ta nhập đúng password thì cửa sẽ mở.
- Trong trường hợp chúng ta nhập sai thì nhấn “*” để xóa 1 ký tự.
- Nếu trong trường hợp mà cửa đang mở và ta nhấn nút thì cửa sẽ đóng lại.

PHỤ LỤC

Lưu đồ chương trình con điều khiển mở cửa

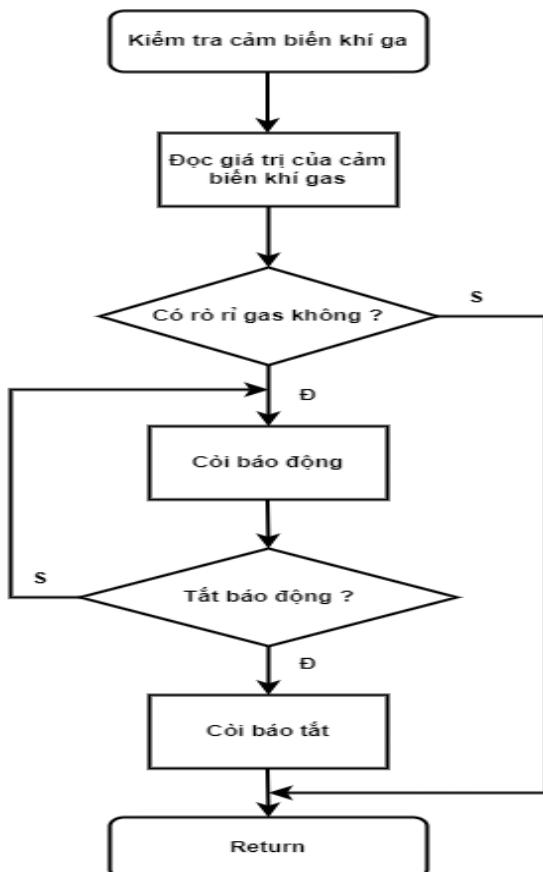


Giải thích lưu đồ:

Giá trị state_door = 1 có nghĩa là trạng thái mở cửa, LCD sẽ hiện dòng chữ “Open door...” và servo sẽ quay để mở cửa.

Hình 2: Lưu đồ chương trình con điều khiển mở cửa

Lưu đồ chương trình con kiểm tra cảm biến khí gas



Giải thích lưu đồ:

- Nếu như có khí gas rò rỉ thì còi sẽ báo động
- Nếu chúng ta nhấn nút button thì còi báo sẽ ngừng.

Hình 3: Lưu đồ kiểm tra cảm biến khí gas

PHỤ LỤC

Lưu đồ chương trình con đọc và xử lý cảm biến DHT11



Giải thích lưu đồ:

Đọc giá trị nhiệt độ độ ẩm và lưu các giá trị đó vào biến t và h tương ứng với nhiệt độ và độ ẩm. Các biến đó được truyền đến ESP dưới dạng chuỗi.

Hình 4: Lưu đồ chương trình con đọc và xử lý cảm biến DHT11

Lưu đồ chương trình con tạo dữ liệu để gửi lên Firebase



Giải thích lưu đồ:

Các dữ liệu của cảm biến được gửi đến ESP8266 dưới dạng chuỗi. Sau đó chuyển các chuỗi dữ liệu ấy thành số nguyên, rồi cập nhật các giá trị ấy lên Firebase.

Hình 5: Lưu đồ chương trình con tạo dữ liệu để gửi lên Firebase

PHỤ LỤC

Chương trình kết nối ESP8266 với App Android

```
#define FIREBASE_HOST "https://lora-home-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "TQnMogeNSOHBQ4n5CDstCNfQZCAS3ElqsyEW3juy"

FirebaseData firebaseData;
WiFiServer server(80);
SoftwareSerial EspSerial(12, 14); // RX, TX

const char* ssid = "Nain";
const char* password = "nghiahuutran23012001";

int val, nhietdo, doam;
String req;
String data2mega = "";

void setup() {
    Serial.begin(9600); //Default Baud Rate for NodeMCU
    EspSerial.begin(9600);

    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi connected");
    server.begin();
    Serial.println("Server started");
    Serial.println(WiFi.localIP());
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {

    WiFiClient client = server.available();
    delay(50);

    NHAN_DATA();

    if (client.available()) {
        req = client.readStringUntil('\r');
        client.flush();
        if (req.indexOf("/id_control/01") != -1)
```

```
    val = 1;
    else if (req.indexOf("/id_control/2") != -1)
        val = 2;
    else if (req.indexOf("/id_control/3") != -1)
        val = 3;
    else if (req.indexOf("/id_control/4") != -1)
        val = 4;
    else if (req.indexOf("/id_control/5") != -1)
        val = 5;
    else if (req.indexOf("/id_control/6") != -1)
        val = 6;
    else if (req.indexOf("/id_control/7") != -1)
        val = 7;
    else if (req.indexOf("/id_control/8") != -1)
        val = 8;
    else if (req.indexOf("/id_control/9") != -1)
        val = 9;
    else if (req.indexOf("/id_control/10") != -1)
        val = 10;
    else if (req.indexOf("/id_control/11") != -1)
        val = 11;
    else if (req.indexOf("/id_control/12") != -1)
        val = 12;
    else if (req.indexOf("/id_control/13") != -1)
        val = 13;
    else if (req.indexOf("/id_control/14") != -1)
        val = 14;
    else if (req.indexOf("/id_control_t2/01") != -1)
        val = 15;
    else if (req.indexOf("/id_control_t2/2") != -1)
        val = 16;
    else if (req.indexOf("/id_control_t2/3") != -1)
        val = 17;
    else if (req.indexOf("/id_control_t2/4") != -1)
        val = 18;
    else if (req.indexOf("/id_control_t2/5") != -1)
        val = 19;
    else if (req.indexOf("/id_control_t2/6") != -1)
        val = 20;
    else if (req.indexOf("/id_control_t2/7") != -1)
        val = 21;
    else if (req.indexOf("/id_control_t2/8") != -1)
        val = 22;
    else {
        Serial.println("invalid request");
        client.stop();
```

```
        return;
    }
    client.flush();
    GUI_DATA();
}

void GUI_DATA() {
    data2mega = "";
    data2mega = String(val);

    Serial.println(data2mega);
    EspSerial.println(data2mega);
}

void NHAN_DATA() {
    String inString = "";      // string to hold input
    String v1, v2, symbol;
    int lengthString, k[1];
    int j = 0;
    if (EspSerial.available() > 0)
    {
        inString = EspSerial.readStringUntil('\n');
        lengthString = inString.length();
        for (int i = 0; i < inString.length(); i++)
        {
            symbol = inString.charAt(i);
            if (symbol.equals(" ") == true)
            {
                k[j] = i;
                j++;
            }
        }
        v1 = inString.substring(0, k[0]);
        v2 = inString.substring(k[0], lengthString);
        nhietdo = v2.toInt();
        doam = v1.toInt();
        Serial.print(doam);
        Serial.print("    ");
        Serial.println(nhietdo);
        Firebase.set(firebaseData, "Doam", doam);
        Firebase.set(firebaseData, "Nhietdo", nhietdo);
    }
}
```

PHỤ LỤC

Chương trình điều khiển chính tầng 1

```
// khai bao chan cam bien
#define PIN_DHT11 2
#define PIN_MQ2 3
#define PIN_BUTTON 5
#define PIN_RASP 4
// khai bao chan quat
#define PIN_QUAT1 31
#define PIN_QUAT2 32
#define PIN_QUAT3 33
// khai bao chan den
#define PIN_DEN1 34
#define PIN_DEN2 35
#define PIN_DEN3 36
#define PIN_DEN4 37
// khai bao chan buzzer
#define PIN_BUZZER 38
// khai bao chan servo
#define PIN_SERVO 6

//Khai bao thu vien
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <SPI.h>
#include <LoRa.h>
#include <Servo.h>
#include "Keypad.h";

// khai bao chan lora
#define rst 11
#define dio0 9
#define ss 53

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(PIN_DHT11, 11);
Servo myservo;
//Khai bao bien
int st_quat1 = 0, st_quat2 = 0, st_quat3 = 0, st_den1 = 0, st_den2 = 0,
st_den3 = 0, st_den4 = 0;
int t, h, idesp, state_door = 0;
unsigned long time;
String datalora, dataesp;
String esp2mega = "", keyid = "", passw = "123";
```

PHỤ LỤC

```
const byte Rows = 4; //number of rows on the keypad i.e. 4
const byte Cols = 3; //number of columns on the keypad i,e, 3

//we will definne the key map as on the key pad:
char keymap[Rows][Cols] =
{
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};

byte rPins[Rows] = {25, 24, 23, 22}; //Rows 0 to 3
byte cPins[Cols] = {28, 27, 26}; //Columns 0 to 2
Keypad kpd = Keypad(makeKeymap(keymap), rPins, cPins, Rows, Cols);

void setup() {
    //khai bao ket noi uart
    Serial.begin(9600); //
    Serial2.begin(9600); // ket noi esp8266
    //khai bao input
    dht.begin();
    pinMode(PIN_BUTTON, INPUT_PULLUP);
    pinMode(PIN_MQ2, INPUT);
    pinMode(PIN_RASP, INPUT);

    //khai bao output
    pinMode(PIN_QUAT1, OUTPUT);
    pinMode(PIN_QUAT2, OUTPUT);
    pinMode(PIN_QUAT3, OUTPUT);
    pinMode(PIN_DEN1, OUTPUT);
    pinMode(PIN_DEN2, OUTPUT);
    pinMode(PIN_DEN3, OUTPUT);
    pinMode(PIN_DEN4, OUTPUT);
    pinMode(PIN_BUZZER, OUTPUT);
    digitalWrite(PIN_QUAT1, LOW);
    digitalWrite(PIN_QUAT2, HIGH);
    digitalWrite(PIN_QUAT3, HIGH);
    digitalWrite(PIN_DEN1, HIGH);
    digitalWrite(PIN_DEN2, HIGH);
    digitalWrite(PIN_DEN3, HIGH);
    digitalWrite(PIN_DEN4, HIGH);
    digitalWrite(PIN_BUZZER, HIGH);

    //khai bao servo
```

PHỤ LỤC

```
myservo.attach(PIN_SERVO);
myservo.write(80);
//khai bao lcd i2c
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Xin Chao");

// cài đặt tần số LoRa 433 MHz
LoRa.setPins(ss, rst, dio0);
if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
}

LoRa.setSpreadingFactor(10);
LoRa.setSignalBandwidth(12345);
LoRa.crc();

time = millis();
}

void loop() {
    KEY_PAD();
    CB_GAZ();
    if ( (unsigned long) (millis() - time) > 5000) {
        DOC_CB();
        SEND_ESP();
        time = millis(); // cập nhật lại biến time
    }

    if (Serial2.available() > 0) {
        esp2mega = "";
        esp2mega = Serial2.readStringUntil('\n'); // val 1-22

        idesp = esp2mega.toInt();
        Serial.println(esp2mega);
        if (idesp < 15)
            DK_RELAY_T1();
        else
            SEND_LORA();
    }
}

void CB_GAZ() {
```

```
if (digitalRead(PIN_MQ2) == 0) {
    Serial.println("roga");
    digitalWrite(PIN_BUZZER, LOW);
}
if (digitalRead(PIN_BUTTON) == 0)
    digitalWrite(PIN_BUZZER, HIGH);
}
void KEY_PAD() {
    if (digitalRead(PIN_RASP) == 1) {
        lcd.setCursor(0, 0);
        lcd.print("Face detect");
        OPEN_DOOR();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Xin Chao");
    }
    char keypressed = kpd.getKey();
    if (keypressed != NO_KEY)
    {
        DOC_CB();
        if (keypressed == '*') {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Xin Chao");
            keyid = keyid.substring(0, keyid.length() - 1);
        }
        else if (keypressed == '#') {
            if (keyid == passw) {
                OPEN_DOOR();
                lcd.clear();
                keyid = "";
                lcd.setCursor(0, 0);
                lcd.print("Xin Chao");
            }
        } else {
            keyid = keyid + String(keypressed);
        }
        lcd.setCursor(0, 1);
        lcd.print(keyid);

        if (keypressed == '*' && state_door == 1) {
            state_door = 0;
            lcd.setCursor(0, 1);
            lcd.print("Close door...");
            for (int pos = 160; pos >= 80; pos -= 1) {
                myservo.write(pos);
            }
        }
    }
}
```

PHỤ LỤC

```
        delay(12);
    }
}
}

void OPEN_DOOR() {
    state_door = 1;
    lcd.setCursor(0, 1);
    lcd.print("Open door... ");
    for (int pos = 80; pos <= 160; pos += 1) {
        myservo.write(pos);
        delay(12);
    }
}

void SEND_LORA() {
    datalora = idesp;
    LoRa.beginPacket();
    LoRa.print(datalora);
    LoRa.endPacket();
}

void SEND_ESP() {
    dataesp = String(h) + " " + String(t);
    Serial2.println(dataesp);
}

void DOC_CB() {
    h = dht.readHumidity(); // do am
    t = dht.readTemperature(); // nhiet do
    Serial.print(h);
    Serial.print("      ");
    Serial.println(t);
}

void DK_RELAY_T1() {
    if (idesp == 1)
        digitalWrite(PIN_QUAT1, LOW);
    if (idesp == 2)
        digitalWrite(PIN_QUAT1, HIGH);

    if (idesp == 3)
        digitalWrite(PIN_QUAT2, LOW);
    if (idesp == 4)
        digitalWrite(PIN_QUAT2, HIGH);
```

PHỤ LỤC

```
if (idesp == 5)
    digitalWrite(PIN_QUAT3, LOW);
if (idesp == 6)
    digitalWrite(PIN_QUAT3, HIGH);

if (idesp == 7)
    digitalWrite(PIN_DEN1, LOW);
if (idesp == 8)
    digitalWrite(PIN_DEN1, HIGH);
if (idesp == 9)
    digitalWrite(PIN_DEN2, LOW);
if (idesp == 10)
    digitalWrite(PIN_DEN2, HIGH);
if (idesp == 11)
    digitalWrite(PIN_DEN3, LOW );
if (idesp == 12)
    digitalWrite(PIN_DEN3, HIGH );
if (idesp == 13)
    digitalWrite(PIN_DEN4, LOW );
if (idesp == 14)
    digitalWrite(PIN_DEN4, HIGH );
}
```

Chương trình điều khiển tầng 2

```
#include <LoRa.h>
#include <SPI.h>

// khai bao chan quat
#define PIN_QUAT1 3
#define PIN_QUAT2 4

// khai bao chan den
#define PIN_DEN1 5
#define PIN_DEN2 6

String mega2uno = "";
int idesp;
void setup() {
    //khai bao ket noi uart
    Serial.begin(9600);
    while (!Serial);
    Serial.println("LoRa Receiver");
    // cài đặt tần số LoRa 433 MHz
```

```
if (!LoRa.begin(433E6)) {
    Serial.println("Lora Error");
    while (1);
}
LoRa.setSpreadingFactor(10);
LoRa.setSignalBandwidth(12345);
LoRa.crc();
//khai bao output
pinMode(PIN_QUAT1, OUTPUT);
pinMode(PIN_QUAT2, OUTPUT);
pinMode(PIN_DEN1, OUTPUT);
pinMode(PIN_DEN2, OUTPUT);

digitalWrite(PIN_QUAT1, LOW);
digitalWrite(PIN_QUAT2, HIGH);
digitalWrite(PIN_DEN1, HIGH);
digitalWrite(PIN_DEN2, HIGH);
}

void loop() {
    READ_LORA();
    DK_RELAY_T2();
}

void READ_LORA() {
    if (LoRa.parsePacket()) { // try to parse packet
        // read packet
        mega2uno = "";
        if (LoRa.available()) {
            mega2uno = LoRa.readString();
            idesp = mega2uno.toInt();
            Serial.println(mega2uno);
        }
    }
}

void DK_RELAY_T2() {
    if (idesp == 15)
        digitalWrite(PIN_QUAT1, LOW);
    if (idesp == 16)
        digitalWrite(PIN_QUAT1, HIGH);
    if (idesp == 17)
        digitalWrite(PIN_QUAT2, LOW);
    if (idesp == 18)
        digitalWrite(PIN_QUAT2, HIGH);
    if (idesp == 19)
```

```
    digitalWrite(PIN_DEN1, LOW);
    if (idesp == 20)
        digitalWrite(PIN_DEN1, HIGH);
    if (idesp == 21)
        digitalWrite(PIN_DEN2, LOW);
    if (idesp == 22)
        digitalWrite(PIN_DEN2, HIGH);
}
```

Chương trình xây dựng mô hình nhận diện khuôn mặt

```
import pickle
import dlib
import imutils
from imutils import face_utils
from imutils import paths
import cv2
import face_recognition
import os
print("[INFO] quantifying faces...")
imagePaths = list(paths.list_images("dataset"))

# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []

# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    print("[INFO] processing image {}/{}".format(i + 1, len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]

    # load the input image and convert it from RGB (OpenCV ordering)
    # to dlib ordering (RGB)
    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # detect the (x, y)-coordinates of the bounding boxes
    # corresponding to each face in the input image
    boxes = face_recognition.face_locations(rgb,
        model="hog")

    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)

    # loop over the encodings
    for encoding in encodings:
        # add each encoding + name to our set of known names and
        # encodings
        knownEncodings.append(encoding)
        knownNames.append(name)

# dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
```

```
f.write(pickle.dumps(data))
f.close()
```

Chương trình nhận diện khuôn mặt

```
import pickle
from imutils import face_utils
import imutils
import time
import dlib
import cv2
from imutils.video import VideoStream
import face_recognition
import RPi.GPIO as GPIO

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.output(17,GPIO.LOW)
data = pickle.loads(open("encodings.pickle", "rb").read())

print("[INFO] loading HOG + Linear SVM face detector...")
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
vs = VideoStream(src=0).start()
while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (7, 7), 0)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)
    boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

    # compute the facial embeddings for each face bounding box
    encodings = face_recognition.face_encodings(rgb, boxes)
    names = []
    for encoding in encodings:
        matches = face_recognition.compare_faces(data["encodings"],
encoding, 0.4)
        name = "Unknown"
        if True in matches:
            GPIO.output(17,GPIO.HIGH)
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            counts = {}
            for i in matchedIdxs:
                name = data["names"][i]
                counts[name] = counts.get(name, 0) + 1
            name = max(counts, key=counts.get)
            names.append(name)

        for (top, right, bottom, left), name in zip(boxes, names):
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
            y = top - 15 if top - 15 > 15 else top + 15
            cv2.putText(frame, name, (left, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 255, 0), 2)
            GPIO.output(17,GPIO.LOW)
```

PHỤ LỤC

```
cv2.imshow('Frame', frame)
key = cv2.waitKey(1) & 0xFF
# if the `q` key was pressed, break from the loop
if key == ord("q"):
    break

# show the output image
cv2.destroyAllWindows()
vs.stop()
GPIO.cleanup()
```

