

Tp. HCM, ngày 20 tháng 03 năm 2018

NHIỆM VỤ ĐO ÁN TỐT NGHIỆP

Họ tên sinh viên:	Huỳnh Xuân Dũng	MSSV:	14141046
	Trần Nhật Minh	MSSV:	14141197
Chuyên ngành:	Kỹ thuật Điện tử - Truyền thông	Mã ngành:	141
Hệ đào tạo:	Đại học chính quy	Mã hệ:	1
Khóa:	2014	Lớp:	14141DT

I. TÊN ĐỀ TÀI: **HỆ THỐNG IoT ĐIỀU KHIỂN VÀ GIÁM SÁT NGÔI NHÀ**

II. NHIỆM VỤ

1. Các số liệu ban đầu:

- Trần Thu Hà – Trương Thị Bích Ngà – Nguyễn Thị Lưỡng – Bùi Thị Tuyết Đan – Phù Thị Ngọc Hiếu – Dương Thị Cẩm Tú, Giáo trình *Điện tử cơ bản*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh.
- Nguyễn Đình Phú, Giáo trình *Vi điều khiển*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.
- Nguyễn Đình Phú, Giáo trình *Vi xử lý nâng cao*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.
- Nguyễn Đình Phú – Nguyễn Trường Duy, Giáo trình *Kỹ thuật số*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh.
- Nguyễn Văn Hiệp – Đinh Quang Hiệp, Giáo trình *Lập trình Android cơ bản*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh.
- Nguyễn Văn Hiệp, Giáo trình *Lập trình Android trong ứng dụng điều khiển*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh.
- Nguyễn Văn Hiệp, Giáo trình *Công nghệ nhận dạng bằng sóng vô tuyến*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh.

2. Nội dung thực hiện:

- Tìm hiểu các hoạt động của các mô hình nhà IoT.

- Thu thập dữ liệu quy trình thiết kế một ngôi nhà IoT.
- Các giải pháp thiết kế hệ thống, mô hình nhà IoT.
- Lựa chọn các thiết bị trong việc thiết kế mô hình nhà IoT (vi điều khiển STM32F407 VGT6, Arduino Mega, Module wifi esp8266, Module Sim, Module thời gian thực, RFID, relay đóng ngắt, bơm nước, màn hình hiển thị, các cảm biến như cảm biến nhiệt độ, độ ẩm, khí gas, PIR ...)
- Tìm hiểu các chuẩn truyền thông UART, I2C, SPI.
- Thiết kế giao diện điều khiển và giám sát: Web server, App android, WPF (Windows Presentation Foundation).
- Thiết kế, thi công mạch nguồn.
- Thiết kế, thi công hệ thống điều khiển.
- Thiết kế, thi công mô hình ngôi nhà.
- Viết chương trình cho STM32F407, Arduino Mega và Esp8266.
- Lắp ráp hệ thống điều khiển vào mô hình và chạy thử nghiệm.
- Chỉnh sửa các lỗi xuất hiện.
- Đánh giá kết quả thực hiện.
- Viết báo cáo luận văn.
- Báo cáo đề tài tốt nghiệp.

III. NGÀY GIAO NHIỆM VỤ: 19/03/2018

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 06/07/2018

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. Nguyễn Đình Phú

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

Tp. HCM, ngày 20 tháng 03 năm 2018

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Huỳnh Xuân Dũng

Lớp: 14141DT1C

MSSV: 14141046

Họ tên sinh viên 2: Trần Nhật Minh

Lớp: 14141DT1C

MSSV: 14141197

Tên đề tài:

HỆ THỐNG IoT ĐIỀU KHIỂN VÀ GIÁM SÁT NGÔI NHÀ

Tuần/ngày	Nội dung	Xác nhận GVHD
Tuần 1 19/03-25/03	- Gặp giảng viên hướng dẫn và trao đổi về đề tài đồ án tốt nghiệp.	
Tuần 2 26/03-01/04	- Viết đề cương chi tiết. - Tìm hiểu các đề tài đã nghiên cứu có liên quan.	
Tuần 3 02/04-08/04	- Gặp và báo cáo với GVHD về hướng thực hiện đề tài. - Tìm hiểu các linh kiện sử dụng trong mạch.	
Tuần 4 09/04-15/04	- Tìm hiểu về giao tiếp giữa các cảm biến, module và các thiết bị với STM32F407 và Arduino. - Tìm hiểu về module wifi Esp8266.	
Tuần 5 16/04-22/04	- Lập trình STM32F407 và Arduino đọc cảm biến, điều khiển các LED đơn và kiểm tra việc thu nhận tín hiệu từ cảm biến.	
Tuần 6 23/04-29/04	- Báo cáo tiến độ cho GVHD. - Tìm hiểu lập trình Web Server, viết App Android và WPF, phương thức gửi dữ liệu thu thập từ bộ điều khiển lên web.	
Tuần 7 30/04-06/05	- Lập trình truyền nhận dữ liệu giữa STM32F407 với Arduino, giữa STM32F407 với Esp8266.	

Tuần 8, 9 07/05-20/05	<ul style="list-style-type: none"> - Báo cáo tiến độ cho GVHD. - Hoàn thành giao diện Web, App Android, WPF. 	
Tuần 10, 11 21/05-03/06	<ul style="list-style-type: none"> - Tổng hợp chương trình đọc tất cả cảm biến, giao tiếp module, truyền nhận dữ liệu và gửi dữ liệu qua internet. - Viết báo cáo. 	
Tuần 12 04/06-10/06	<ul style="list-style-type: none"> - Thiết kế, hoàn thành mô hình và tiến hành đính dây vào mô hình. - Kiểm tra hoạt động của hệ thống. - Viết báo cáo. 	
Tuần 13, 14, 15 11/06-01/07	<ul style="list-style-type: none"> - Chạy thử hệ thống, kiểm tra lại và sửa lỗi. - Viết và hoàn thiện báo cáo. 	

GV HƯỚNG DẪN

(Ký và ghi rõ họ và tên)

LỜI CAM ĐOAN

Đề tài này là do chúng tôi tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài

Huỳnh Xuân Dũng

Trần Nhật Minh

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn sâu sắc đến ThS. Nguyễn Đình Phú đã trực tiếp hướng dẫn, góp ý, chia sẻ nhiều kinh nghiệm quý báu, tận tình giúp đỡ và tạo điều kiện để chúng em hoàn thành tốt đê tài.

Chúng em xin gửi lời chân thành cảm ơn các thầy cô trong Khoa Điện - Điện Tử đã tạo những điều kiện tốt nhất cho em hoàn thành đê tài.

Chúng em cũng gửi lời đồng cảm ơn đến các bạn lớp 14141DT đã chia sẻ trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian thực hiện đê tài.

Cảm ơn đến cha mẹ đã tạo điều kiện tốt nhất về kinh tế và tinh thần để con hoàn thành tốt đê tài này.

Xin chân thành cảm ơn!

Người thực hiện đê tài

Huỳnh Xuân Dũng

Trần Nhật Minh

MỤC LỤC

<i>Nội dung</i>	<i>Trang</i>
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	i
LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP	iii
LỜI CAM ĐOAN	v
LỜI CẢM ƠN	vi
MỤC LỤC	vii
LIỆT KÊ HÌNH	x
LIỆT KÊ BẢNG	xiii
TÓM TẮT	xiv
Chương 1: TỔNG QUAN.....	1
1.1 ĐẶT VẤN ĐỀ	1
1.2 MỤC TIÊU	1
1.3 MỤC TIÊU CỦA NHÓM	2
1.4 NỘI DUNG NGHIÊN CỨU	2
1.5 GIỚI HẠN	3
1.6 BÓ CỤC	3
Chương 2: CƠ SỞ LÝ THUYẾT	5
2.1 GIỚI THIỆU CÁC THIẾT BỊ VÀO RA SỬ DỤNG TRONG ĐỀ TÀI	5
2.2 GIỚI THIỆU PHẦN CÙNG	5
2.2.1 Cảm biến	5
2.2.2 Module thời gian thực DS1307	15
2.2.3 Màn hình Oled	20
2.2.4 Vi Điều Khiển	21
2.2.5 Module SIM900A	29
2.2.6 Dòng chip Wi-Fi ESP8266	36
2.2.7 Công nghệ RFID	42
2.2.8 Bàn phím ma trận	47
2.2.9 Relay tiếp điểm cơ khí	48
2.2.10 Module hạ áp LM2596	49
2.3 CÁC CHUẨN TRUYỀN DỮ LIỆU	50
2.3.1 Chuẩn truyền thông UART	50
2.3.2 Chuẩn giao tiếp I2C	52
2.3.3 Chuẩn truyền thông SPI	55
Chương 3: TÍNH TOÁN VÀ THIẾT KẾ	56

3.1	GIỚI THIỆU	56
3.2	TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	56
3.2.1	Thiết kế sơ đồ khái hệ thống	56
3.2.2	Tính toán và thiết kế.....	58
3.2.3	Sơ đồ nguyên lý toàn mạch	73
Chương 4: THI CÔNG HỆ THỐNG	75	
4.1	GIỚI THIỆU	75
4.2	THI CÔNG HỆ THỐNG.....	75
4.2.1	Thi công board mạch.....	75
4.2.2	Lắp ráp và kiểm tra	82
4.3	ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH.....	85
4.3.1	Đóng gói bộ điều khiển	85
4.3.2	Thi công mô hình.....	86
4.4	LẬP TRÌNH HỆ THỐNG	89
4.4.1	Lưu đồ giải thuật.....	89
4.4.2	Phần mềm lập trình cho vi điều khiển	107
4.4.3	Phần mềm lập trình giao diện điều khiển.....	109
4.5	VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC	112
4.5.1	Viết tài liệu hướng dẫn sử dụng.....	112
4.5.2	Quy trình thao tác.....	113
Chương 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ	121	
5.1	GIỚI THIỆU	121
5.2	KẾT QUẢ ĐẠT ĐƯỢC.....	121
5.2.1	Biết cách sử dụng các cảm biến.....	121
5.2.2	Biết cách lập trình STM32F407	121
5.2.3	Biết cách lập trình Arduino Mega	121
5.2.4	Biết cách sử dụng module wifi ESP8266.....	122
5.2.5	Biết cách viết app android, web và WPF	122
5.2.6	Biết cách lập trình đóng mở cửa bằng RFID.....	122
5.2.7	Biết cách truyền nhận dữ liệu giữa các vi điều khiển.....	122
5.2.8	Biết cách truyền nhận dữ liệu qua wifi	122
5.3	KẾT QUẢ THỰC NGHIỆM	123
5.3.1	Quét thẻ RFID hoặc nhập mật khẩu đóng mở cửa	123
5.3.2	Điều khiển và giám sát thiết bị, cảm biến thông qua internet.....	126
5.3.3	Điều khiển thiết bị và giám sát an ninh qua Sim	136
5.3.4	Hiển thị màn hình oled.....	141

5.4 NHẬN XÉT – ĐÁNH GIÁ	142
5.4.1 Nhận xét	142
5.4.2 Đánh giá	142
Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	144
6.1 KẾT LUẬN	144
6.2 HƯỚNG PHÁT TRIỂN	144
TÀI LIỆU THAM KHẢO	145

LIỆT KÊ HÌNH

Hình 2.1: Cảm biến LM35 và sơ đồ nối chân	5
Hình 2.2: Cảm biến DS18B20 và sơ đồ nối chân.....	6
Hình 2.3: Bộ nhớ ROM 64 bit	7
Hình 2.4: Tổ chức bộ nhớ ROM của cảm biến DS18B20.....	8
Hình 2.5: Byte thanh ghi điều khiển của cảm biến DS18B20.....	8
Hình 2.6: Cảm biến DHT21	11
Hình 2.7: Cảm biến DHT11 và sơ đồ nối chân.....	12
Hình 2.8: Cảm biến khí gas MQ2 và sơ đồ nối chân	13
Hình 2.9: Cảm biến MQ5	13
Hình 2.10: Cảm biến PIR.....	14
Hình 2.11: Cảm biến độ ẩm đất	15
Hình 2.12: Sơ đồ chân DS1307	16
Hình 2.13: Thanh ghi DS1307.....	17
Hình 2.14: Tổ chức thanh ghi DS1307	18
Hình 2.15: Module DS1307.....	20
Hình 2.16: Màn hình Oled	21
Hình 2.17: Board STM32F407 VGT6.....	22
Hình 2.18: Arduino Mega2560.....	27
Hình 2.19: Cấu trúc của mạng GSM	30
Hình 2.20: Module Sim900A	32
Hình 2.21: Sơ đồ nguyên lý ESP8266.....	38
Hình 2.22: Hình ảnh ESP-01.....	39
Hình 2.23: Hình ảnh ESP-07.....	40
Hình 2.24: Hình ảnh ESP-12.....	40
Hình 2.25: ESP8266 NodeMCU.....	42
Hình 2.26: Thiết bị IFF và thiết bị RFID hiện đại ngày nay	43
Hình 2.27: Sơ đồ khối của một hệ thống RFID.....	44
Hình 2.28: Hoạt động giữa tag và reader RFID.....	46
Hình 2.29: Bàn phím ma trận 4x4	47
Hình 2.30: Relay 5V.....	49
Hình 2.31: Module LM2596	49
Hình 2.32: Sơ đồ nguyên lý module LM2596.....	50
Hình 2.33: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC	51
Hình 2.34: Bus I2C và các thiết bị ngoại vi.....	52
Hình 2.35: Trình tự truyền bit trên đường truyền	53
Hình 2.36: Điều kiện start stop	54
Hình 2.37: Truyền dữ liệu I2C	54
Hình 2.38: Giao diện kết nối 4 dây chuẩn SPI.....	55
Hình 3.1: Sơ đồ khối của hệ thống	56
Hình 3.2: Thiết kế mô hình căn hộ	58
Hình 3.3: Sơ đồ nguyên lý kết nối cảm biến với STM32F407	59
Hình 3.4: Sơ đồ kết nối cảm biến với Arduino mega.....	59
Hình 3.5: Sơ đồ nguyên lý kết nối Oled với STM32F407	60
Hình 3.6: Sơ đồ nguyên lý kết nối DS1307 với STM32F407	60

Hình 3.7: Sơ đồ nguyên lý kết nối công suất với STM32F407.....	62
Hình 3.8: Sơ đồ nguyên lý kết nối module SIM với Arduino	63
Hình 3.9: Sơ đồ nguyên lý kết nối LCD với Arduino.....	64
Hình 3.10: Sơ đồ nguyên lý kết nối bàn phím 4x4 với Arduino	64
Hình 3.11: Sơ đồ nguyên lý kết nối đầu đọc RC522 với Arduino	65
Hình 3.12: Sơ đồ nguyên lý kết nối còi báo với Arduino.....	65
Hình 3.13: Sơ đồ kết nối khói động cơ với Arduino	66
Hình 3.14: Sơ đồ nguyên lý kết nối esp8266 với STM32F407.....	67
Hình 3.15: Khối điều khiển và hiển thị qua internet	68
Hình 3.16: Khối xử lý và điều khiển (chính)	68
Hình 3.17: Khối xử lý và điều khiển (phụ).....	69
Hình 3.18: Sơ đồ nguyên lý khói nguồn dự phòng	70
Hình 3.19: Sơ đồ nguyên lý toàn mạch (chính).....	73
Hình 3.20: Sơ đồ nguyên lý toàn mạch (phụ).....	74
Hình 4.1: Sơ đồ bố trí linh kiện mạch điều khiển trung tâm (chính)	76
Hình 4.2: Sơ đồ đi dây mạch điều khiển trung tâm (chính).....	77
Hình 4.3: Sơ đồ bố trí linh kiện mạch nút nhấn điều khiển thiết bị	77
Hình 4.4: Sơ đồ đi dây mạch điều khiển thiết bị	78
Hình 4.5: Sơ đồ bố trí linh kiện mạch màn hình hiển thị oled.....	78
Hình 4.6: Sơ đồ đi dây mạch màn hình hiển thị oled	79
Hình 4.7: Sơ đồ bố trí linh kiện mạch điều khiển (phụ)	79
Hình 4.8: Sơ đồ đi dây của mạch điều khiển (phụ)	80
Hình 4.9: Hình ảnh thực tế mạch điều khiển trung tâm (chính)	83
Hình 4.10: Hình ảnh thực tế mạch nút nhấn điều khiển thiết bị.....	83
Hình 4.11: Hình ảnh thực tế mạch màn hình hiển thị oled.....	84
Hình 4.12: Hình ảnh thực tế mạch điều khiển trung tâm (phụ)	85
Hình 4.13: Hộp bảo vệ bộ điều khiển của hệ thống	86
Hình 4.14: Hình thành ý tưởng thiết kế mô hình.....	87
Hình 4.15: Hoàn chỉnh bộ khung và tiến hành đi dây.....	88
Hình 4.16: Trang trí và lắp ráp hoàn thiện mô hình.....	88
Hình 4.17: Lưu đồ chương trình STM32F407	90
Hình 4.18: Lưu đồ chương trình con điều khiển thiết bị	92
Hình 4.19: Lưu đồ chương trình chính Arduino	94
Hình 4.20: Lưu đồ chương trình con đọc cảm biến	96
Hình 4.21: Lưu đồ chương trình con đăng nhập bằng phím	98
Hình 4.22: Lưu đồ chương trình con kiểm tra nút nhấn cửa	99
Hình 4.23: Lưu đồ chương trình con đổi mật khẩu	100
Hình 4.24: Lưu đồ chương trình con truyền nhận SIM	102
Hình 4.25: Lưu đồ chương trình con đọc thẻ RFID	104
Hình 4.26: Lưu đồ chương trình esp8266.....	106
Hình 4.27: Giao diện phần mềm KeilC	108
Hình 4.28: Logo phần mềm Arduino IDE	108
Hình 4.29: Giao diện lập trình của phần mềm Arduino IDE	109
Hình 4.30: Logo phần mềm Visual Studio	110
Hình 4.31: Giao diện lập trình web của phần mềm Visual Studio.....	111
Hình 4.32: Giao diện phần mềm Android studio	112
Hình 4.33: Lưu đồ quy trình thao tác	113

Hình 4.34: Giao diện app android	114
Hình 4.35: Các giao diện app android sau khi đăng nhập.....	115
Hình 4.36: Giao diện WPF.....	116
Hình 4.37: Giao diện lần đầu đăng nhập web	117
Hình 4.38: Giao diện web sau khi đăng nhập thành công.....	118
Hình 4.39: Giao diện trang điều khiển	119
Hình 4.40: Giao diện trang giám sát.....	120
Hình 5.1: Quét thẻ đúng và cửa mở ra, chuông kêu.....	123
Hình 5.2: Quét thẻ đúng và cửa đóng lại.....	124
Hình 5.3: Quét thẻ sai, cửa không mở, chuông kêu	124
Hình 5.4: Nhập đúng mật khẩu và cửa mở ra.....	125
Hình 5.5: Nhập sai mật khẩu, cửa không mở, chuông kêu	126
Hình 5.6: Giao diện trang giám sát.....	126
Hình 5.7: Tất cả thiết bị đều tắt khi chưa điều khiển	127
Hình 5.8: Tất cả các thiết bị đều tắt	127
Hình 5.9: Nhấn nút Light1, Fan1 để bật đèn và quạt phòng 1	128
Hình 5.10: Đèn và quạt phòng 1 được bật lên	128
Hình 5.11: Trạng thái thiết bị được cập nhật lên web	129
Hình 5.12: Bật tất cả đèn và quạt	129
Hình 5.13: Tất bơm nước	130
Hình 5.14: Giao diện app khi đăng nhập.....	130
Hình 5.15: Tất cả các thiết bị đều tắt	131
Hình 5.16: Nhấn nút Light3, Fan3 để bật đèn và quạt phòng 3	131
Hình 5.17: Đèn và quạt phòng 3 được bật lên	132
Hình 5.18: Trạng thái thiết bị được cập nhật lên app android.....	132
Hình 5.19: Giao diện WPF.....	133
Hình 5.20: Tất cả các thiết bị đều tắt	134
Hình 5.21: Nhấn tất cả các nút để bắt hé tết thiết bị	134
Hình 5.22: Bật hết tất cả các thiết bị	135
Hình 5.23: Trạng thái thiết bị được cập nhật lên WPF.....	135
Hình 5.24: Gửi tin nhắn Bat1 cho sim để bật Light1	136
Hình 5.25: Bật đèn phòng 1	137
Hình 5.26: Gửi tin nhắn "Trangt" để kiểm tra trạng thái thiết bị.....	138
Hình 5.27: Trạng thái thiết bị được gửi lại qua tin nhắn	138
Hình 5.28: Gửi tin nhắn "Batan" để bật an ninh	139
Hình 5.29: Trạng thái an ninh được gửi lại qua tin nhắn	139
Hình 5.30: Hệ thống an ninh gửi tin nhắn khi có trộm vào nhà	140
Hình 5.31: Hệ thống an ninh gọi điện khi có trộm vào nhà	140
Hình 5.32: Hiển thị thời gian thực trên oled.....	141
Hình 5.33: Hiện thị nhiệt độ phòng trên oled	141
Hình 5.34: Hiện thị các thông số cảm biến trên oled	141
Hình 5.35: Hiện thị trạng thái thiết bị trên oled	142

LIỆT KÊ BẢNG

<i>Bảng 2.2: Độ phân giải và thời gian chuyển đổi.....</i>	8
<i>Bảng 2.3: So sánh giữa các loại bộ nhớ</i>	29
<i>Bảng 2.4: Các chế độ lệnh AT</i>	33
<i>Bảng 3.1: Dòng điện của các linh kiện sử dụng trong mạch (STM32F407)</i>	71
<i>Bảng 3.2: Dòng các linh kiện sử dụng trong mạch (Arduino)</i>	72
<i>Bảng 3.3: Dòng điện của các linh kiện sử dụng trong mạch 12V.....</i>	72
<i>Bảng 4.1: Bảng thống kê linh kiện.....</i>	81
<i>Bảng 5.1: Số liệu thực nghiệm.....</i>	143

TÓM TẮT

Ngày nay với sự phát triển không ngừng của khoa học và công nghệ với những ứng dụng của khoa học kỹ thuật tiên tiến, thế giới chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Sự phát triển của kỹ 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 hay những thiết bị mà con người không thể trực tiếp chạm vào để vận hành điều khiển.

Nhìn chung các năm trước, đề tài nhà IoT được các sinh viên thực hiện xoay quanh các nội dung như: Điều khiển bằng tần số vô tuyến, bằng Bluetooth nhưng các phương pháp này phụ thuộc khoảng cách, chỉ có tác dụng trong một phạm vi hẹp, dễ bị nhiễu trong khi sử dụng. Do đó, việc thiết kế và thi công “Hệ thống IoT điều khiển và giám sát ngôi nhà” là một nhu cầu hết sức cần thiết và đây chính là lý do mà nhóm quyết định chọn đề tài này. Đề tài này không những là một thực tại khách quan mà nó còn đóng vai trò đặc biệt quan trọng thực sự hiện tại cũng như trong tương lai sau này.

Nội dung chính của đề tài

- Sử dụng Board STM32F407VGT6 để làm khói điều khiển trung tâm (chính).
- Sử dụng Board Arduino Mega để làm khói điều khiển trung tâm (phụ).
- Điều khiển và giám sát thiết bị qua mạng Internet (web server, app android và WPF).
- Điều khiển thiết bị qua Sim.
- Hệ thống tưới cây theo lịch trình.
- Giám sát hệ thống cảm biến (nhiệt độ, độ ẩm, độ ẩm đất, gas)
- Cảnh báo khi có người lạ vào nhà và gửi tin nhắn khi bị rò rỉ khí gas.
- Ứng dụng công nghệ RFID vào việc đóng mở cửa và ngoài ra có thẻ đóng mở cửa bằng cách nhập mật khẩu.

Chương 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Ngày nay, công nghệ kết nối đầu tiên cần nhắc đến hiển nhiên là Wifi – công nghệ kết nối không dây phổ biến nhất hiện nay. Cũng vì tính phổ biến của dạng kết nối này mà cái tên Wifi thường bị lạm dụng để chỉ kết nối không dây nói chung.

Lí do mà kết nối Wifi được ưa chuộng như vậy đơn giản là vì khả năng hoạt động hiệu quả trong phạm vi vài chục mét đến vài trăm mét của các mạng WLAN.

Và trong thời đại công nghiệp hóa hiện đại hóa hiện nay, việc phát minh và chế tạo ra các thiết bị thông minh có khả năng điều khiển từ xa đang và sẽ rất được quan tâm và rất hữu ích cho cuộc sống hàng ngày.

Vì mục tiêu công nghệ hiện đại hóa ngày càng phát triển, chúng tôi đã quyết định làm một đồ án “**Hệ thống IoT điều khiển và giám sát ngôi nhà**”. Đề tài này đã được thực hiện khá nhiều nhưng họ chỉ dừng lại ở việc đo đặc, điều khiển các thiết bị trong nhà qua internet. Đề tài của chúng tôi ngoài việc điều khiển thiết bị độc lập thì còn giám sát các cảm biến đặt trong ngôi nhà và cảnh báo khi gặp sự cố. Khi dự án hoàn thành chúng ta có thể cảnh báo trộm, báo khí gas rò rỉ, tự động tưới cây theo lịch trình và điều khiển các thiết bị điện trong nhà... bằng cách tương tác qua các nút nhấn để hiển thị trạng thái hoạt động trên web, giao diện Android và WPF trên máy tính. Như vậy, dù chúng ta ở bất cứ nơi nào có internet đều có thể giám sát và điều khiển được các thiết bị đã kết nối với module điều khiển.

Khi dự án thành công và được áp dụng rộng rãi thì sẽ rất tiện lợi cho cuộc sống thường ngày, giúp cho đất nước ngày càng phát triển.

1.2 MỤC TIÊU

Thiết kế hệ thống giám sát và điều khiển thiết bị trong nhà, điều khiển các thiết bị thông qua App Android, Web, WPF, SIM và phím nhấn cứng. Hệ thống có thể gửi tin nhắn báo trộm, báo khí gas rò rỉ và tưới cây theo lịch trình, đóng mở cửa bằng RFID và bàn phím ma trận, kiểm tra trạng thái cửa. Ngoài ra, có thêm PIN dự phòng để đóng mở cửa khi có sự cố mất điện. Các thông số hiển thị như trạng thái cửa, trạng thái thiết bị, nhiệt độ, độ ẩm, độ ẩm đất, khí gas được hiển thị trên màn hình OLED và trên các giao diện App Android, WPF, Web một cách trực quan.

CHƯƠNG 1: TỔNG QUAN

Tìm hiểu giao thức truyền nhận dữ liệu giữa vi điều khiển ARM và Arduino, giữa ARM và ESP8266.

1.3 MỤC TIÊU CỦA NHÓM

Thiết kế hệ thống giám sát và điều khiển thiết bị trong nhà, điều khiển các thiết bị thông App Android, WPF và phím nhấn cứng. Hệ thống tưới cây theo lịch trình. Các thông số hiển thị như trạng thái cửa, trạng thái thiết bị, nhiệt độ, độ ẩm, độ ẩm đất, khí gas được hiển thị trên màn hình OLED và trên các giao diện App Android, WPF một cách trực quan.

Tìm hiểu giao thức truyền nhận dữ liệu của vi điều khiển ARM, ESP8266.

1.4 NỘI DUNG NGHIÊN CỨU

Đề tài “hệ thống IoT điều khiển và giám sát ngôi nhà” có các nội dung chính như sau:

- Tìm hiểu các hoạt động của các mô hình nhà IoT.
- Thu thập dữ liệu quy trình thiết kế một ngôi nhà IoT.
- Các giải pháp thiết kế hệ thống, mô hình nhà IoT.
- Lựa chọn các thiết bị trong việc thiết kế mô hình nhà IoT (vi điều khiển STM32F407 VGT6, Arduino Mega, Module wifi esp8266, Module Sim, Module thời gian thực, RFID, relay đóng ngắt, bơm nước, màn hình hiển thị, các cảm biến như cảm biến nhiệt độ, độ ẩm, khí gas, PIR ...)
- Tìm hiểu các chuẩn truyền thông UART, I2C, SPI.
- Thiết kế giao diện điều khiển và giám sát: Web server, App android, WPF (Windows Presentation Foundation).
- Thiết kế, thi công mạch nguồn.
- Thiết kế, thi công hệ thống điều khiển.
- Thiết kế, thi công mô hình ngôi nhà.
- Viết chương trình cho STM32F407, Arduino Mega và Esp8266.
- Lắp ráp hệ thống điều khiển vào mô hình và chạy thử nghiệm.
- Chỉnh sửa các lỗi xuất hiện.
- Đánh giá kết quả thực hiện.
- Viết báo cáo luận văn.
- Báo cáo đề tài tốt nghiệp.

1.5 GIỚI HẠN

Đề tài “hệ thống IoT điều khiển và giám sát ngôi nhà” có những giới hạn sau:

- Thiết kế mô hình nhà.
- Điều khiển tắt mở các thiết bị gia dụng có công suất thấp như: bóng đèn, quạt...
- Nhận biết trạng thái tắt mở thông qua LED đơn (LED sáng trạng thái mở, LED tắt trạng thái tắt).
- Hiển thị nhiệt độ, độ ẩm, khí gas, đồng hồ thời gian thực và trạng thái thiết bị trên màn hình LCD.
- Hệ thống an ninh với khóa số điện tử có ma trận phím, RFID, cảm biến gas báo khí gas rò rỉ qua còi, cảm biến PIR phát hiện chuyển động.
- Dùng module SIM900A để gửi tin nhắn thông báo khi có người nhập mật khẩu khóa số vượt quá số lần quy định, khi cảm biến khí gas cảnh báo, điều khiển bật tắt thiết bị bằng tin nhắn.
- Sử dụng STM32F407VGT6 và ARDUINO trong việc lập trình điều khiển.
- Viết chương trình kết hợp sử dụng ESP8226 trong việc gửi và nhận dữ liệu.
- Đo nhiệt độ sử dụng cảm biến DS18B20.
- Đo khí gas sử dụng cảm biến MQ2.
- Gửi dữ liệu (trạng thái thiết bị, nhiệt độ, độ ẩm, khí gas) lên Database.
- Sử dụng bơm điều khiển việc tưới cây theo lịch trình.
- Đo nhiệt độ, độ ẩm khu vườn bằng cảm biến DHT11.
- Chỉ điều khiển trên điện thoại và máy tính chạy hệ điều hành window.

1.6 BỐ CỤC

Nội dung đề tài gồm các phần sau:

Chương 1: Tổng quan

- Đặt vấn đề
- Mục tiêu của đề tài
- Nhiệm vụ của đề tài
- Giới hạn

Chương 2: Cơ sở lý thuyết

- Giới thiệu chuẩn truyền thông UART, I2C, SPI
- Giới thiệu Board STM32F407VGT6 và Arduino và các linh kiện khác

CHƯƠNG 1: TỔNG QUAN

- Giao tiếp Wifi
- Giao tiếp SIM900A
- Công nghệ RFID

Chương 3: Tính toán và thiết kế

- Thiết kế sơ đồ khói
- Thiết kế cho từng khói
- Sơ đồ nguyên lý toàn mạch

Chương 4: Thi công hệ thống

- Thi công các board mạch
- Thi công mô hình
- Lưu đồ, giới thiệu phần mềm lập trình và viết chương trình

Chương 5: Kết quả_nhận xét đánh giá

- Kết quả đạt được
- Kết quả thực nghiệm
- Nhận xét đánh giá

Chương 6: Kết luận_hướng phát triển

- Kết luận
- Hướng phát triển

Chương 2: CƠ SỞ LÝ THUYẾT

2.1 GIỚI THIỆU CÁC THIẾT BỊ VÀO RA SỬ DỤNG TRONG ĐỀ TÀI

- Thiết bị đầu vào: Cảm biến nhiệt độ - độ ẩm DHT11, cảm biến nhiệt độ DS18B20, cảm biến khí gas, cảm biến chuyển động PIR, cảm biến độ ẩm đất, nút nhấn, RFID, Module thời gian thực DS1307.
- Thiết bị đầu ra: thiết bị công suất: động cơ servo, bơm nước, động cơ DC, đèn 220V; thiết bị giao tiếp công suất: transistor, opto, relay; thiết bị hiển thị: LCD 16x2, Oled.
- Thiết bị điều khiển trung tâm: STM32F407VGT6, Arduino Mega.
- Module wifi: Esp8266 NodeMCU.
- Các chuẩn truyền dữ liệu: UART, I2C, SPI.
- Thiết bị giao diện điều khiển: điện thoại Android, Laptop.
- Web server, app android, WPF, Module Sim900A.

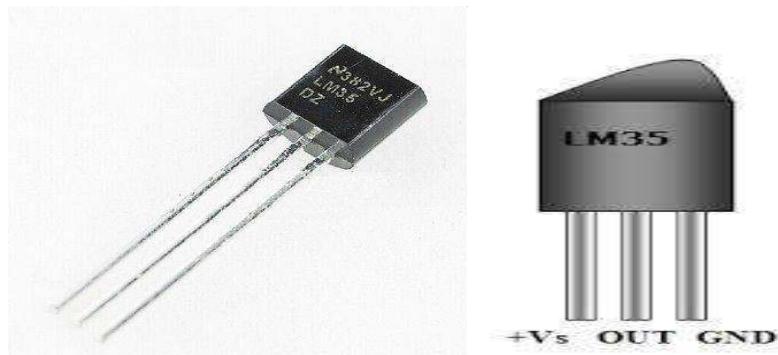
2.2 GIỚI THIỆU PHẦN CỨNG

2.2.1 Cảm biến

a. Cảm biến nhiệt độ - độ ẩm

Cảm biến có chức năng đo nhiệt độ, độ ẩm, khí gas. Hiện nay có rất nhiều loại cảm biến có thể làm được chức năng này như LM35, DS18B20, DHT11, DHT21...

❖ **Cảm biến LM35:** Là bộ cảm biến nhiệt mạch tích hợp chính xác cao mà điện áp đầu ra của nó tỷ lệ tuyến tính với nhiệt độ theo thang độ C. Chúng cũng không yêu cầu cân chỉnh ngoài vì vốn chúng đã được cân chỉnh. Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.



Hình 2.1: Cảm biến LM35 và sơ đồ nối chân

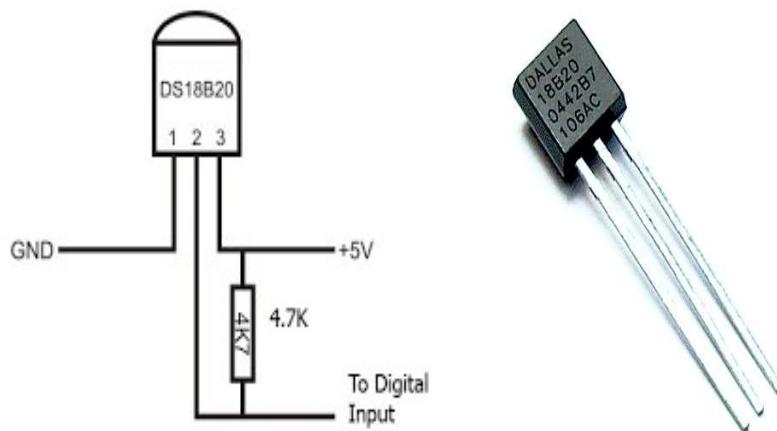
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật:

- Điện áp đầu vào từ 4V đến 30V
- Điện áp ra: -1V đến 6V
- Công suất tiêu thụ là 60uA
- Độ phân giải điện áp đầu ra là 10mV/°C
- Độ chính xác cao ở 25°C là 0.5°C
- Trở kháng đầu ra thấp 0.1 cho 1mA tải
- Độ chính xác thực tế: 1/4°C ở nhiệt độ phòng và 3/4°C ngoài khoảng -55°C tới 150°C

❖ **Cảm biến DS18B20:** Là cảm biến đã chuyển đổi ADC sẵn và lưu trữ giá trị trong Eeprom của chính nó, có thể chọn đọc nhiều độ phân giải. Sử dụng giao tiếp 1 dây rất gọn gàng, dễ lập trình và giao tiếp nhiều DS18B20 trên cùng 1 dây.

Để đo được nhiệt độ ta cần thêm 1 điện trở 4.7k Ohm nối từ chân DQ lên VCC.



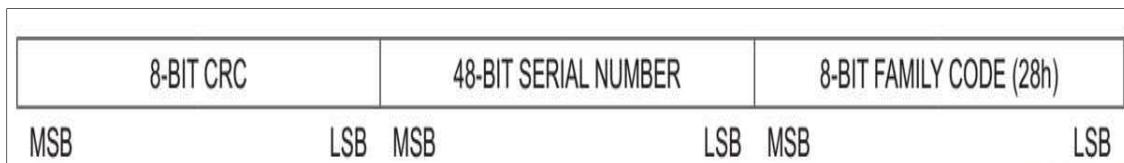
Hình 2.2: Cảm biến DS18B20 và sơ đồ nối chân

Thông số kỹ thuật:

- Nguồn: 3 → 5.5V
- Dải đo nhiệt độ: -55 → 125°C (-67 → 257°F)
- Sai số: $\pm 0.5^{\circ}\text{C}$ khi đo ở dải -10 → 85°C
- Độ phân giải: người dùng có thể chọn từ 9 → 12 bits
- Chuẩn giao tiếp: 1-Wire (1 dây).
- Có cảnh báo nhiệt khi vượt ngưỡng cho phép và cấp nguồn từ chân data.
- Thời gian chuyển đổi nhiệt độ tối đa: 750ms (khi chọn độ phân giải 12bit).

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Mỗi IC có một mã riêng (lưu trên Eeprom của IC) nên có thể giao tiếp nhiều DS18B20 trên cùng 1 dây.
- Mỗi DS18B20 đều có mã định danh với độ lớn 64 bit duy nhất được lưu trong ROM.
- 8 bit thấp nhất của ROM chứa đựng mã quy ước của họ dòng đo nhiệt độ 1 dây DS18B20 với mã là: 28h.
- 48 bit tiếp theo là số serial duy nhất của thiết bị.
- 8 bit cuối cùng mà mã kiểm tra CRC tính toán từ 56 bit trước.



Hình 2.3: Bộ nhớ ROM 64 bit

Bộ nhớ

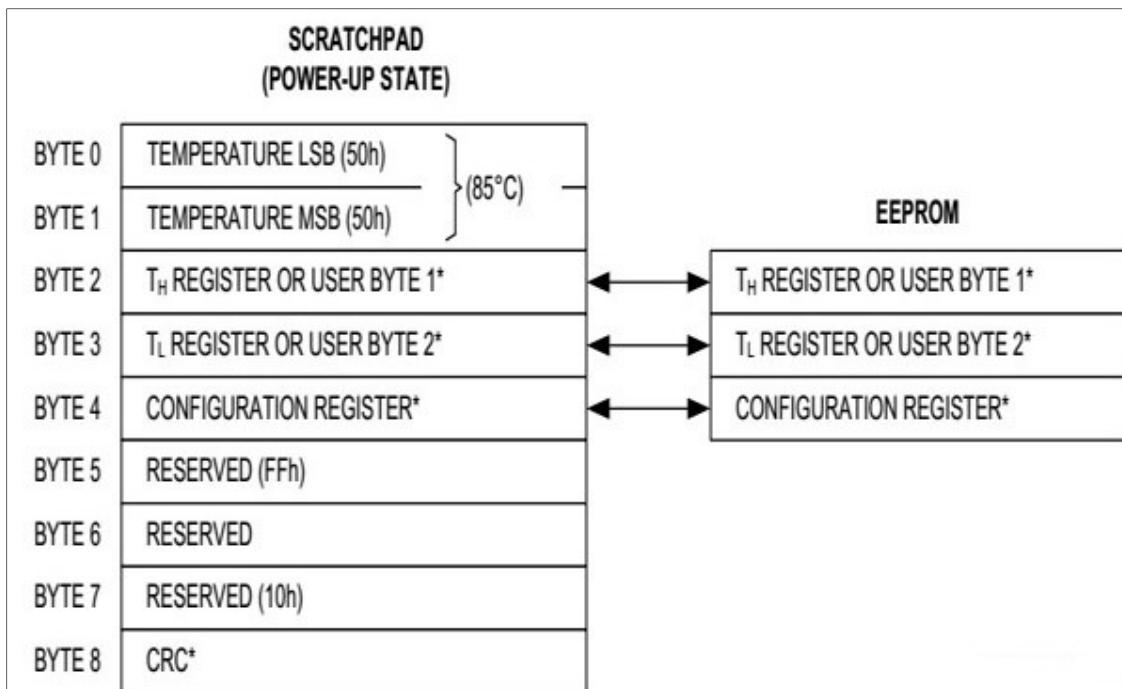
Bộ nhớ của DS18B20 bao gồm bộ nhớ nháp (SRAM scratchpad), thanh ghi lưu trữ kích hoạt cảnh báo cao và thấp (T_H và T_L) và thanh ghi cấu hình, cả hai thanh ghi này đều trang bị bộ nhớ Eeprom. Nếu nếu chức năng cảnh báo không được sử dụng thì thanh ghi T_H và T_L có thể được sử dụng như bộ nhớ đệm mục đích.

Byte 0 và Byte 1 của bộ nhớ nháp chứa đựng LSB và MSB của thanh ghi nhiệt độ. Những Byte này chỉ có thể đọc. Byte 2 và 3 truy cập thanh ghi T_H và T_L . Byte 4 chứa đựng dữ liệu của thanh ghi cấu hình. Byte 5, 6 và 7 để dành riêng cho sử dụng bởi thiết bị, ta không thể ghi đến các byte này.

Byte 8 chứa đựng mã CRC của byte 0 đến byte 7 của bộ nhớ nháp.

Dữ liệu trong các thanh ghi EEPROM không mất đi khi ngắt nguồn cấp, khi có nguồn cấp lại dữ liệu trong các thanh ghi này sẽ được nạp vào bộ nhớ nháp theo vị trí byte tương ứng. Dữ liệu này có thể nạp lại bằng lệnh từ EEPROM bằng lệnh E2 [B8h]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.4: Tổ chức bộ nhớ ROM của cảm biến DS18B20

Bộ nhớ SRAM scratchpad

Thanh ghi cấu hình:

Byte thứ 4 của bộ nhớ nháp chứa đựng thanh ghi cấu hình. Người dùng có thể thiết lập độ phân giải của DS18B20 sử dụng bit R0 và R1. Khi cấp nguồn mặc định R0=1, R1=1 (độ phân giải 12 bit). Bit thứ 7 và bit 0 đến bit 4 trong thanh ghi cấu hình được để dành riêng cho thiết bị và không thể ghi đè.



Hình 2.5: Byte thanh ghi điều khiển của cảm biến DS18B20

Bảng 2.1: Độ phân giải và thời gian chuyển đổi

R1	R0	Độ phân giải (Bit)	Thời gian chuyển đổi tối đa	
0	0	9	93.75ms	(tCONV/8)
0	1	10	187.5ms	(tCONV/4)
1	0	11	375ms	(tCONV/2)
1	1	12	750ms	(tCONV)

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Các lệnh cơ bản:

SEARCH ROM [F0h]

Khi hệ thống được cấp nguồn, thiết bị đóng vai trò master phải xác định mã ROM của tất cả các thiết bị slave được đấu trên cùng bus, việc làm này cho phép thiết bị master xác định số lượng thiết bị slave và kiểu thiết bị.

READ ROM (33h)

Lệnh này chỉ dùng khi trên bus có 1 cảm biến DS1820, nếu không sẽ xảy ra xung đột trên bus do tất cả các thiết bị tó cùng đáp ứng. Nó cho phép đọc 64 bit mã ROM (8 bit mã định tên linh kiện (10h), 48 bit số xuất xưởng, 8 bit kiểm tra CRC) và không sử dụng quy trình Search Rom.

MATCH ROM (55h)

Lệnh này được gửi đi cùng với 64 bit ROM tiếp theo, cho phép bộ điều khiển bus chọn ra chỉ một cảm biến DS1820 cụ thể khi trên bus có nhiều cảm biến DS1820 cùng nối vào. Chỉ có DS1820 nào có 64 bit trên ROM trùng khớp với chuỗi 64 bit vừa được gửi tới mới đáp ứng lại các lệnh về bộ nhớ tiếp theo. Còn các cảm biến DS1820 có 64 bit ROM không trùng khớp sẽ tiếp tục chờ một xung reset. Lệnh này được sử dụng cả trong trường hợp có một cảm biến một dây, cả trong trường hợp có nhiều cảm biến một dây.

SKIP ROM (CCh)

Thiết bị master có thể sử dụng lệnh này để gửi đến tất cả các thiết bị slave trên bus một cách đồng thời mà không cần gửi mã ROM định danh của thiết bị. Ví dụ như thiết bị master ra lệnh cho tất cả các DS18B20 trên Bus chuyển đổi nhiệt độ một cách đồng thời bởi gửi lệnh **SKIP ROM** và lệnh **CONVERT T (44h)**

Lệnh **READ SCRATCHPAD** có thể theo sau lệnh **SKIP ROM** chỉ khi trên bus chỉ có một thiết bị slave. Trong trường hợp này, ta tiết kiệm được thời gian bởi nó cho phép đọc từ thiết bị slave mà không cần gửi mã ROM của thiết bị. Trong trường hợp trên bus có nhiều thiết bị tó nếu ta sử dụng 2 lệnh này sẽ xảy ra xung đột dữ liệu.

ALARM SEARCH (ECh)

Tiến trình của lệnh này giống hệt như lệnh Search ROM, nhưng cảm biến DS1820 chỉ đáp ứng lệnh này khi xuất hiện điều kiện cảnh báo trong phép đo nhiệt độ cuối cùng. Điều kiện cảnh báo ở đây được định nghĩa là giá trị nhiệt độ đo được lớn hơn giá trị T_H

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

và nhỏ hơn giá trị T_L là hai giá trị nhiệt độ cao nhất và nhiệt độ thấp nhất đã được đặt trên thanh ghi trong bộ nhớ của cảm biến.

Sau khi thiết bị chủ (thường là một vi điều khiển) sử dụng các lệnh ROM để định địa chỉ cho các cảm biến một dây đang được đấu vào bus, thiết bị chủ sẽ đưa ra các lệnh chức năng DS1820. Bằng các lệnh chức năng thiết bị chủ có thể đọc ra và ghi vào bộ nhớ nháp (scratchpath) của cảm biến DS1820. Khởi tạo quá trình chuyển đổi giá trị nhiệt độ đo được và xác định chế độ cung cấp điện áp nguồn. Các lệnh chức năng có thể được mô tả ngắn gọn như sau:

WRITE SCRATCHPAD (4Eh)

Lệnh này cho phép ghi 2 byte dữ liệu vào bộ nhớ nháp của DS1820. Byte đầu tiên được ghi vào thanh ghi TH (byte 2 của bộ nhớ nháp) còn byte thứ hai được ghi vào thanh ghi TL (byte 3 của bộ nhớ nháp). Dữ liệu truyền theo trình tự đầu tiên là bit có ý nghĩa nhất và kế tiếp là những bit có ý nghĩa giảm dần. Cả hai byte này phải được ghi trước khi thiết bị chủ xuất ra một xung reset hoặc khi có dữ liệu khác xuất hiện.

READ SCRATCHPAD (BEh)

Lệnh này cho phép thiết bị chủ đọc nội dung bộ nhớ nháp. Quá trình đọc bắt đầu từ bit có ý nghĩa nhất của byte 0 và tiếp tục cho đến byte thứ 9 (byte 8 – CRC). Thiết bị chủ có thể xuất ra một xung reset để làm dừng quá trình đọc bất kỳ lúc nào nếu chỉ có một phần của dữ liệu trên bộ nhớ cần được đọc.

COPYSCRATCHPAD (48h)

Lệnh này copy nội dung của hai thanh ghi T_H và T_L (byte 2 và byte 3), và thanh ghi cấu hình từ bộ nhớ nháp đến bộ nhớ EEPROM. Nếu cảm biến được sử dụng trong chế độ sử dụng nguồn ký sinh, trong 10us (tối đa) sau khi truyền lệnh này, thiết bị master phải cho phép một “strong pull-up” lên bus.

CONVERT T (44h)

Lệnh này khởi động một quá trình đo và chuyển đổi giá trị nhiệt độ thành số (nhị phân). Sau khi chuyển đổi giá trị kết quả đo nhiệt độ được lưu trữ trên thanh ghi nhiệt độ 2 byte trong bộ nhớ nháp. Trong thời gian đang chuyển đổi nếu thực hiện lệnh đọc thì các giá trị đọc ra đều bằng 0.

READ POWER SUPPLY (B4h)

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Một lệnh đọc tiếp sau lệnh này sẽ cho biết DS1820 đang sử dụng chế độ cấp nguồn như thế nào, giá trị đọc được bằng 0 nếu cấp nguồn bằng chính đường dẫn dữ liệu và bằng 1 nếu cấp nguồn qua một đường dẫn riêng.

❖ **Cảm biến DHT21:** Cảm biến độ ẩm, nhiệt độ DHT21 AM2301 tích hợp cảm biến độ ẩm điện dung và cảm biến nhiệt độ có độ chính xác cao, khả năng chống nhiễu mạnh, giao tiếp duy nhất 1 dây. Kích thước nhỏ, tiêu thụ điện năng thấp, khoảng cách truyền dẫn tín hiệu lên đến 20m, ổn định lâu dài, thiết bị đo nhiệt độ chính xác cao.

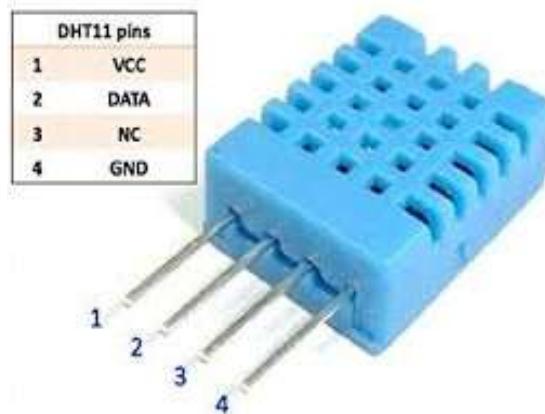


Hình 2.6: Cảm biến DHT21

Thông số kỹ thuật:

- Áp nguồn: 3.3 → 5V
- Dòng tiêu thụ: 300 uA
- Kích thước: 58.8 x 26.7 x 13.8 (mm)
- Model: AM2301
- Độ phân giải chính xác: 0.1
- Khoảng đo: 0100% RH
- Khoảng đo nhiệt độ: -40 °C → 80 °C
- Đo lường chính xác độ ẩm: ± 3% RH
- Đo lường chính xác nhiệt độ: ± 0.5 °C

❖ **Cảm biến DHT11:** Cảm biến độ ẩm và nhiệt độ 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. Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.



Hình 2.7: Cảm biến DHT11 và sơ đồ nối chân

Thông tin kỹ thuật:

- Nguồn: 3 → 5 VDC.
- Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
- Đo tốt ở độ ẩm 2080%RH với sai số 5%.
- Đo tốt ở nhiệt độ 0 → 50°C sai số ±2°C.
- Tần số lấy mẫu tối đa 1Hz (1 giây 1 lần)
- Kích thước 15mm x 12mm x 5.5mm.
- 4 chân, khoảng cách chân 0.1".

❖ **Lựa chọn cảm biến:** Trong quá trình học cùng với thực tập, nhóm đã được biết cách sử dụng cảm biến nhiệt độ DS18B20 do vậy nên nhóm chọn cảm biến DS18B20 để thực hiện việc đọc nhiệt độ phòng. Để đọc nhiệt độ, độ ẩm cho vườn cây nhóm sử dụng cảm biến DHT11 do nó có chi phí rẻ và có thể đọc được nhiệt độ lẫn độ ẩm.

b. Cảm biến khí Gas

❖ **Cảm biến khí Gas (LPG/CO/CH4) MQ-2:** MQ-2 sử dụng phần tử SnO₂ có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO₂ sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện. MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH₄) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau. Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở. Nguồn sử dụng 5VDC.



Hình 2.8: Cảm biến khí gas MQ2 và sơ đồ nối chân

❖ **Cảm biến khí gas MQ5:** Sử dụng để đo khí gas trong môi trường. Cảm biến có độ nhạy cao, khả năng phản hồi nhanh, độ nhạy có thể điều chỉnh được bằng biến trở. Nguồn hoạt động 5V.



Hình 2.9: Cảm biến MQ5

❖ **Lựa chọn cảm biến:** Giữa hai loại cảm biến MQ2 và MQ5 thì MQ2 phổ biến, có giá thành rẻ hơn nên nhóm sử dụng cảm biến MQ2 để đọc khí gas.

c. Cảm biến thân nhiệt chuyển động

Cảm biến thân nhiệt chuyển động PIR HC-SR501 được sử dụng để phát hiện chuyển động của các vật thể phát ra bức xạ hồng ngoại (con người, con vật, các vật phát nhiệt...), cảm biến có thể chỉnh được độ nhạy để giới hạn khoảng cách bắt xa gần cũng như cường độ bức xạ của vật thể mong muốn, ngoài ra cảm biến còn có thể điều chỉnh thời gian kích trễ (giữ tín hiệu bao lâu sau khi kích hoạt) qua biến trờ tích hợp sẵn.



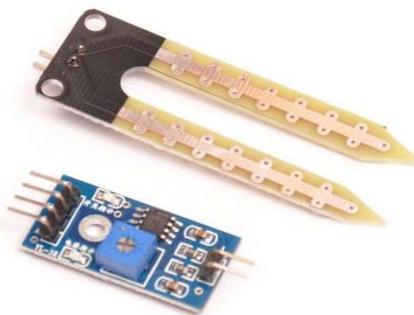
Hình 2.10: Cảm biến PIR

Thông số kỹ thuật:

- Phạm vi phát hiện: góc 360° hình nón, độ xa tối đa 6m.
- Nhiệt độ hoạt động: $32 \rightarrow 122^\circ\text{F}$ ($0 \rightarrow 50^\circ\text{C}$)
- Điện áp hoạt động: DC $3.8\text{V} \rightarrow 5\text{V}$.
- Mức tiêu thụ dòng: $\leq 50\text{ uA}$.
- Thời gian báo: 30 giây có thể tùy chỉnh bằng biến trờ.
- Độ nhạy có thể điều chỉnh bằng biến trờ.
- Kích thước: $1.27 \times 0.96 \times 1.0$ ($32.2 \times 24.3 \times 25.4$ mm).

d. Cảm biến độ ẩm đất

Cảm biến độ ẩm đất, trạng thái đầu ra mức thấp (0V), khi đất thiếu nước đầu ra sẽ là mức cao (5V), độ nhạy cao chúng ta có thể điều chỉnh được bằng biến trờ. Cảm biến độ ẩm đất có thể sử dụng tưới hoa tự động khi không có người quản lý khu vườn hoặc dùng trong những ứng dụng tương tự như trồng cây. Độ nhạy của cảm biến phát hiện độ ẩm đất có thể tùy chỉnh được. Phần đầu đo được cắm vào đất để phát hiện độ ẩm của đất, khi độ ẩm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao.



Hình 2.11: Cảm biến độ ẩm đất

Thông số kỹ thuật

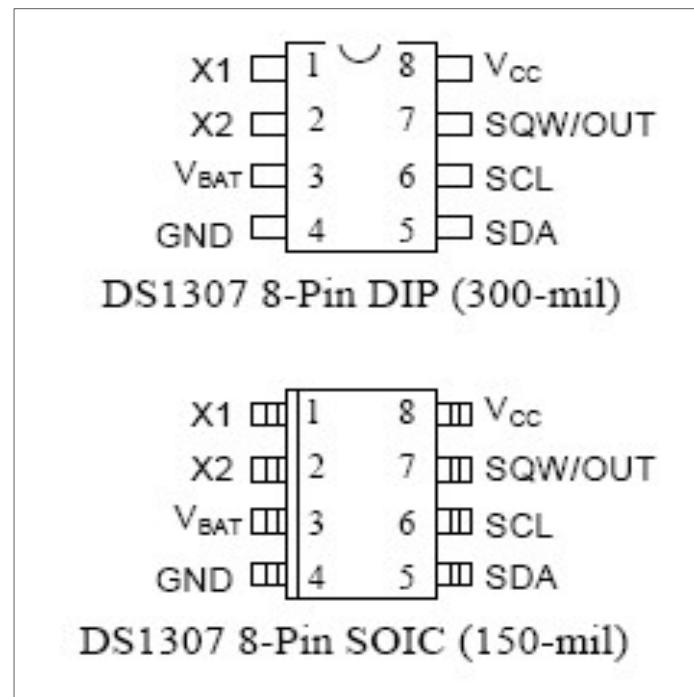
- Điện áp hoạt động: 3.3V → 5V.
- Kích thước PCB: 3 x 1.6 cm.
- Led đỏ báo nguồn vào, led xanh báo độ ẩm.
- IC so sánh: LM393.
- VCC: 3.3V → 5V.
- GND: 0V.
- DO: Đầu ra tín hiệu số (0 và 1).
- AO: Đầu ra Analog (Tín hiệu tương tự).

2.2.2 Module thời gian thực DS1307

a. Giới thiệu DS1307

DS1307 là chip thời gian thực hay RTC (Read time clock). Đây là một IC tích hợp cho thời gian bởi vì tính chính xác về thời gian tuyệt đối cho thời gian: Thứ, ngày, tháng, năm, giờ, phút, giây. DS1307 là chế tạo bởi Dallas. Chip này có 7 thanh ghi 8 bit mỗi thanh ghi này chứa: thứ, ngày, tháng, năm, giờ, phút, giây. Ngoài ra DS1307 còn chứa 1 thanh ghi điều khiển ngõ ra phụ và 56 thanh ghi trống các thanh ghi này có thể dùng như là RAM. DS1307 được đọc thông qua chuẩn truyền thông I2C nên do đó để đọc được và ghi từ DS1307 thông qua chuẩn truyền thông này. Do nó được giao tiếp chuẩn I2C nên cấu tạo bên ngoài nó rất đơn giản.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.12: Sơ đồ chân DS1307

Trên là hai dạng cấu tạo của DS1307. Chip này có 8 chân và chúng ta hay dùng là dạng Dip và các chân nó được mô tả như sau:

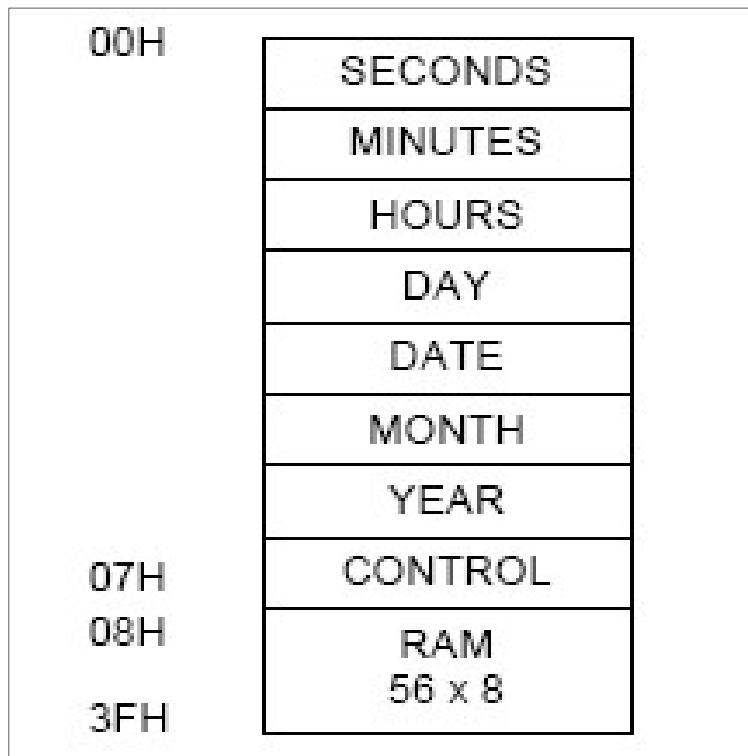
- **X1** và **X2** là đầu vào dao động cho DS1307. Cần giao động thạch anh 32.768 Khz.
- **Vbat** là nguồn nuôi cho chip. Nguồn này từ (2V- 3.5V) ta lấy pin có nguồn 3V. Đây là nguồn cho chip hoạt động liên tục khi không có nguồn Vcc mà DS1307 vẫn hoạt động theo thời gian.
- **Vcc** là nguồn cho giao tiếp I₂C. Điện áp cung cấp là 5V chuẩn và được dùng chung với vi xử lý. Nếu Vcc không có mà Vbat có thì DS1307 vẫn hoạt động bình thường nhưng mà không ghi và đọc được dữ liệu.
- **GND** là nguồn mass chung cho cả Vcc và Vbat.
- **SQW/OUT** là một ngõ ra phụ tạo xung dao động (xung vuông). Chân này không ảnh hưởng đến thời gian thực nên chúng ta không sử dụng chân này trong thời gian thực và bỏ trống chân này.
- **SCL** và **SDA** là hai bus dữ liệu của DS1307. Thông tin truyền và ghi đều được truyền qua 2 đường truyền này theo chuẩn I₂C.

b. Tổ chức thanh ghi trong DS1307

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Cấu tạo bên trong của DS1307 bao gồm mạch nguồn, dao động, logic và con trỏ, thanh ghi thực hiện việc ghi đọc. Do trong các bài toán chúng ta thường sử dụng DS1307 cho đồng hồ thời gian thực nên do đó chúng ta chỉ quan tâm đến việc ghi đọc các thanh ghi cần thiết (sec, min, hour...) thông qua chuẩn truyền thông I2C.

Trong bộ nhớ của DS1307 có tất cả 64 thanh ghi địa chỉ từ 0 đến 63 và được bắt đầu từ 0x00 đến 0x3F nhưng trong đó chỉ có 8 thanh ghi đầu là thanh ghi thời gian thực.



Hình 2.13: Thanh ghi DS1307

Các thanh ghi thời gian thực nó được sắp xếp theo thứ tự: giây, phút, giờ, thứ, ngày, tháng, năm và bắt đầu từ thanh ghi giây (0x00) và kết thúc bằng thanh ghi năm (0x06). Riêng thanh ghi Control dùng để điều khiển ngõ ra của chân SQW/OUT nên trong thực tế không mấy ai sử dụng thanh ghi này trong thời gian thực nên chúng ta bỏ qua thanh ghi này.

Do 7 thanh ghi đầu tiên là khá quan trọng cho thời gian thực và là thanh ghi quan trọng nhất trong con DS1307 nên chúng ta phải hiểu được cách tổ chức thanh ghi này trong DS1307.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

	BIT7							BIT0
00H	CH	10 SECONDS			SECONDS			00-59
	0	10 MINUTES			MINUTES			00-59
	0	12 24	10 HR A/P	10 HR	HOURS			01-12 00-23
	0	0	0	0	0	DAY		1-7
	0	0	10 DATE		DATE			01-28/29 01-30 01-31
	0	0	0	10 MONTH	MONTH			01-12
	10 YEAR				YEAR			00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0

Hình 2.14: Tổ chức thanh ghi DS1307

Nhìn bảng trên chúng ta thấy các thanh ghi được mã hóa theo bit. Mỗi bit trong thanh ghi đều có chức năng riêng.

Thanh ghi giây (0x00): Đây là thanh ghi giây của DS1307. Nhìn hình trên ta thấy được từ bit 0 đến bit 3 là dùng để mã hóa số BCD hàng đơn vị của giây. Tiếp theo, từ bit 4 đến bit 6 dùng để mã hóa BCD hàng chục của giây. Tại sao nó chỉ sử dụng có 3 bit này là do giây của chúng ta lớn nhất chỉ đến 59 nên hàng chục lớn nhất là 5 nên chỉ cần 3 thanh ghi này là đủ mã hóa. Còn bit thứ 7 có tên là “CH” là “Clock Halt – Treo đồng hồ”, do đó nếu bit 7 này được đưa lên 1 tức là khóa đồng hồ nên do đó nó vô hiệu hóa chip và chip không hoạt động. Nên do vậy lúc nào cũng phải cho bit 7 này luôn xuống 0 từ lúc đầu (cái này sử dụng lệnh and với 0x7F).

Thanh ghi phút (0x01): Đây là thanh ghi phút của DS1307. Cũng nhìn trên hình thanh ghi này được tổ chức như thanh ghi giây. Cũng là 3 bit thấp dùng để mã hóa BCD chữ số hàng đơn vị và số hàng trực chỉ lớn nhất là 5 nên do đó chỉ cần dùng từ bit 4 đến bit 6 để mã hóa BCD tiếp chữ số hàng chục. Nhưng thanh ghi này có sự khác biệt với

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

thanh ghi giờ là bit 7 nó đã mặc định bằng 0 rồi nên do đó chúng ta không phải làm gì với bit 7.

Thanh ghi giờ (0x02): Đây là thanh ghi giờ của DS1307, thanh ghi này được coi là phức tạp nhất nhưng mà nhìn bảng thì thấy các tổ chức của nó cũng hợp lý. Trước tiên chúng ta thấy được rằng từ bit 0 đến bit 3 nó dùng để mã hóa BCD của chữ số hàng đơn vị của giờ. Giờ còn có chế độ 24h và 12h nên do đó nó phức tạp ở các bit cao (bit 4 đến bit 7) và sự chọn chế độ 12h và 24h nó lại nằm ở bit 6. Nếu bit 6 = 0 thì ở chế độ 24h thì do chữ số hàng chục lớn nhất là 2 nên do đó nó chỉ dùng 2 bit (bit 4 và bit 5) để mã hóa BCD chữ số hàng chục của giờ. Nếu bit 6 = 1 thì chế độ 12h được chọn nhưng do chữ số của hàng chục của giờ trong chế độ này chỉ lớn nhất là 1 nên do đó bit thứ 4 là đủ để mã hóa BCD chữ số hàng chục của giờ rồi nhưng mà bit thứ 5 nó lại dùng để chỉ buổi sáng hay chiều, nếu mà bit 5 = 0 là AM và bit 5 = 1 là PM. Trong cả 2 chế độ 12h và 24h thì bit 7 = 0 nên ta ko cần chú ý đến thanh ghi này.

Thanh ghi thứ (0x03): Đây là thanh ghi thứ trong tuần của DS1307 và thanh ghi này khá là đơn giản trong DS1307. Nó dùng số để chỉ thứ trong tuần nên do đó nó chỉ lấy từ 1 đến 7 tương đương từ thứ hai đến chủ nhật. Nên do đó nó dùng 3 bit thấp (bit 0 đến bit 2) để mã hóa BCD ra thứ trong ngày. Còn các bit từ 3 đến 7 thì nó mặc định bằng 0 và ta không làm gì với các bit này.

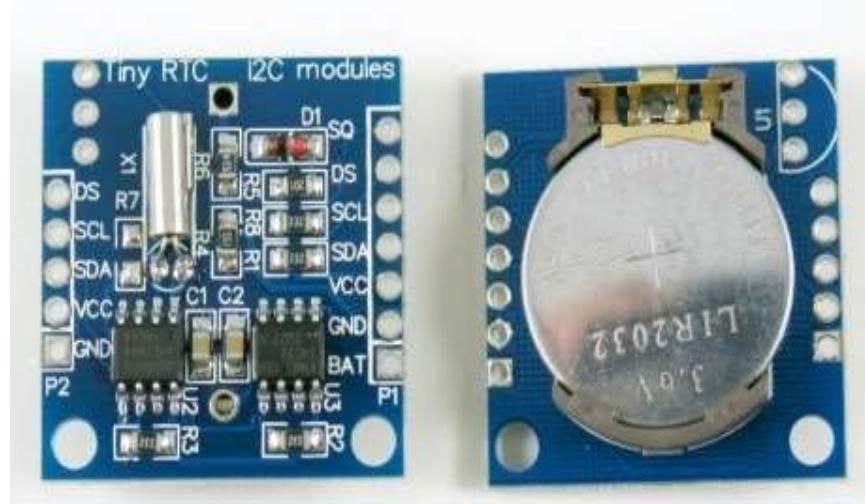
Thanh ghi ngày (0x04): Đây là thanh ghi ngày trong tháng của DS1307. Do trong các tháng có số ngày khác nhau nhưng mà nằm trong khoảng từ 1 đến 31 ngày. Do đó thanh ghi này các bit được tổ chức khá là đơn giản. Nó dùng 4 bit thấp (bit 0 đến bit 3) dùng để mã hóa BCD ra chữ số hàng đơn vị của ngày trong tháng. Nhưng do chữ số hàng chục của ngày trong tháng chỉ lớn nhất là 3 nên chỉ dùng bit 4 và bit 5 là đủ mã hóa BCD rồi. Còn bit 6 và bit 7 chúng ta không làm gì và nó mặc định bằng 0.

Thanh ghi tháng (0x05): Đây là thanh ghi tháng trong năm của DS1307. Tháng trong năm chỉ có từ 1 đến 12 tháng nên việc tổ chức trong bit cũng tương tự như ngày trong tháng nên do cũng 4 bit thấp (từ bit 0 đến bit 3) mã hóa BCD hàng đơn vị của tháng. Nhưng do hàng chục chỉ lớn nhất là 1 nên chỉ dùng 1 bit thứ 4 để mã hóa BCD ra chữ số hàng trực và các bit còn lại từ bit 5 đến bit 7 thì bỏ trống và nó mặc định cho xuống mức 0.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thanh ghi năm (0x06): Đây là thanh ghi năm trong DS1307. DS1307 chỉ có 100 năm tương đương với 00 đến 99 nên nó dùng tất cả các bit thấp và bit cao để mã hóa BCD ra năm.

Thanh ghi điều khiển (0x07): Đây là thanh ghi điều khiển quá trình ghi của DS1307 và Quá trình ghi phải được kết thúc bằng địa chỉ 0x93.



Hình 2.15: Module DS1307

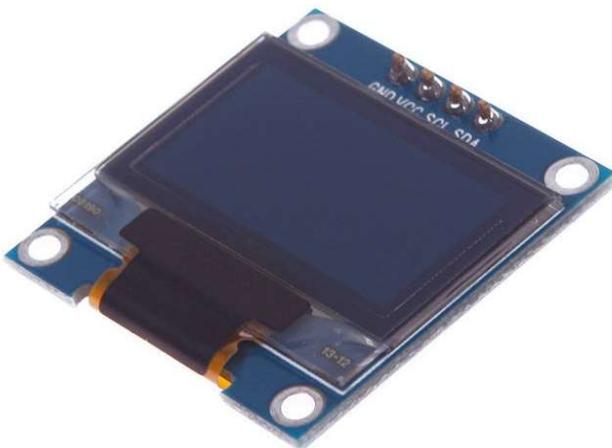
Thông tin kỹ thuật:

- Nguồn cung cấp: 5VDC.
- Khả năng lưu trữ 32K bit với EEPROM AT24C32.
- Sử dụng giao thức 2 dây I2C.
- Lưu trữ thông tin giờ phút giây AM/PM.
- Lịch lưu trữ chính xác lên đến năm 2100.
- Có pin đồng hồ lưu trữ thông tin.
- Có ngõ ra tần số 1Hz.
- Kích thước: 16 x 22 x 23mm.

2.2.3 Màn hình Oled

Màn hình Oled 0.96 inch giao tiếp I2C cho khả năng hiển thị đẹp, sang trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO.

Màn hình dùng để hiển thị các thông số như ngày tháng năm, giờ phút giây, nhiệt độ các phòng, thông số khí gas, nhiệt độ, độ ẩm vườn cây hay trạng thái thiết bị...



Hình 2.16: Màn hình Oled

Thông tin kỹ thuật:

- Điện áp sử dụng: 2.2 → 5.5VDC.
- Công suất tiêu thụ: 0.04W.
- Góc hiển thị: lớn hơn 160°.
- Số điểm hiển thị: 128x64 điểm.
- Độ rộng màn hình: 0.96 inch.
- Màu hiển thị: Trắng / Xanh Dương.
- Giao tiếp: I2C.
- Driver: SSD1306.

2.2.4 Vi Điều Khiển

Xử lí và điều khiển các tín hiệu trong đề tài này có thể đáp ứng bởi một trong nhiều loại VDK khác nhau như: PIC, AVR, ARM... Hoặc các mạch vi điều khiển khác như Arduino, Pi... Mỗi loại có ưu và nhược điểm riêng tùy vào nhu cầu sử dụng của người dùng và chi phí thực hiện của mỗi người mà lựa chọn thích hợp.

PIC: VDK đã được học ở môn vi xử lý với đầy đủ các tính năng như UART, SPI, I2C, USB, ADC. Tuy vậy PIC thì không được sử dụng rộng rãi ở ngoài thực tế.

ARM: VDK hiện đang phổ biến nhiều hơn cả bởi vì có đủ các tính năng kể trên có nhiều chấn hơn VDK PIC nhưng giá thành lại tốt hơn.

Arduino: VDK có độ phổ biến cao và có cộng đồng hỗ trợ đông đảo. Arduino đã hộ trợ rất nhiều trong quá trình lập trình và thiết kế. Phần cứng của thiết bị tích hợp nhiều chức năng cơ bản và là mã nguồn mở.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

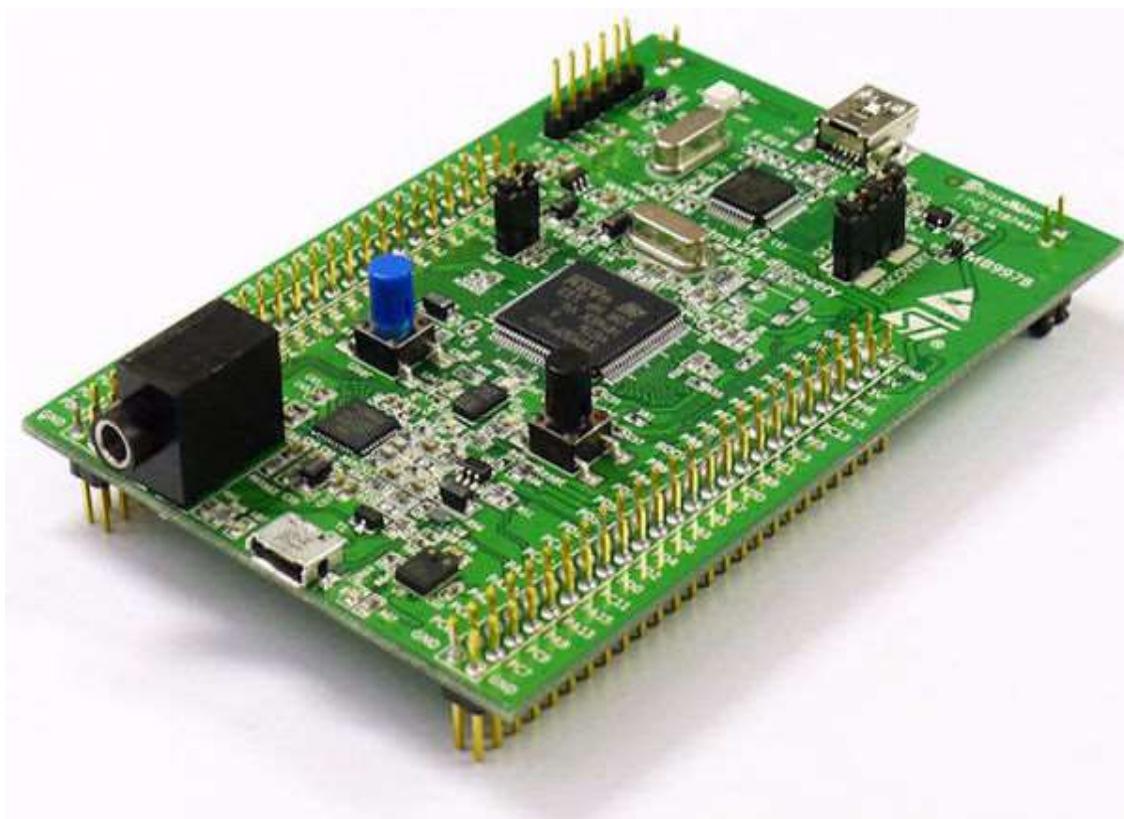
Sau cùng, nhóm đã chọn VĐK ARM STM32F07 VGT6 và Arduino Mega phục vụ cho việc nghiên cứu và thực hiện đề tài này.

a. Board STM32F407 VGT6

Dòng STM32F407xx được dựa trên lõi RISC 32-bit ARM®Cortex®-M4 hiệu suất cao hoạt động ở tần số lên đến 168 MHz. Core Cortex-M4 có độ chính xác đơn điểm nổi (FPU) hỗ trợ tất cả các hướng dẫn xử lý dữ liệu và các kiểu dữ liệu có độ chính xác đơn ARM. Nó cũng thực hiện một bộ đầy đủ các hướng dẫn DSP và một đơn vị bảo vệ bộ nhớ (MPU) để tăng cường bảo mật ứng dụng.

Hệ STM32F407xx kết hợp các bộ nhớ nhúng tốc độ cao (bộ nhớ Flash lên tới 1 Mbyte, lên tới 192 Kbytes SRAM), lên tới 4 Kbytes SRAM sao lưu, và một loạt các thiết bị ngoại vi và thiết bị ngoại vi được kết nối với hai bus APB, ba bus AHB và một trận bus đa AHB 32 bit.

Tất cả các thiết bị đều cung cấp ba bộ ADC 12 bit, hai DAC, một RTC công suất thấp, 12 bộ định thời 16 bit chung bao gồm hai bộ đếm thời gian PWM để điều khiển động cơ, hai bộ định thời 32-bit có mục đích chung. Một trình tạo số ngẫu nhiên thực (RNG). Họ cũng có giao diện truyền thông tiêu chuẩn và tiên tiến.



Hình 2.17: Board STM32F407 VGT6

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

❖ Đặc tính quan trọng:

- Core: CPU Cortex® -M4 32-bit của ARM® với FPU, Bộ tăng tốc thời gian thực thích ứng (ART Accelerator™) cho phép thực thi trạng thái chờ 0 từ bộ nhớ Flash, tần số lên tới 168 MHz, bộ bảo vệ bộ nhớ, 210 DMIPS / 1.25 DMIPS / MHz (Dhrystone 2.1) và hướng dẫn DSP
- Bộ nhớ Flash lên tới 1 Mbyte
- Đến 192 + 4 Kbytes SRAM bao gồm RAM dữ liệu 64-Kbyte CCM (bộ nhớ ghép đôi lõi)
- Bộ điều khiển bộ nhớ tĩnh linh hoạt hỗ trợ các bộ nhớ Compact Flash, SRAM, PSRAM, NOR và NAND
- Giao diện LCD song song, chế độ 8080/6800
- Đồng hồ, reset và quản lý nguồn cung cấp
- 1,8 V đến 3,6 V cung cấp ứng dụng và I/O
- Dao động tinh thể 4 đến 26 MHz
- Bộ vi xử lý RC nội bộ 16 MHz
- Bộ dao động 32 kHz cho RTC với hiệu chuẩn
- Dao động nội 32 kHz RC
- Chế độ Sleep, Stop và Standby
- Cung cấp VBAT cho RTC, đăng ký dự phòng 20×32 bit + SRAM sao lưu 4 KB tùy chọn
- Bộ chuyển đổi A/D 3×12 bit, 2,4 MSPS: tối đa 24 kênh và 7,2 MSPS ở chế độ xen kẽ ba
- Bộ chuyển đổi D/A 2×12 bit
- DMA đa mục đích: Bộ điều khiển DMA 16 dòng với FIFO và hỗ trợ burst
- Tối đa 17 bộ hẹn giờ: lên đến 12 giờ 16 bit và hai bộ định thời 32 bit lên tới 168 MHz, mỗi bộ đếm lên tới 4 đầu vào bộ mã hóa IC / OC / PWM hoặc bộ đếm xung và đầu vào tần số (tăng dần)
- Chế độ kiểm tra sửa lỗi
- Giao diện debug dây nối tiếp (SWD) & JTAG
- 140 cổng I/O với khả năng ngắt
- 136 I/O với tần số lên tới 84 MHz
- 138 I/O chịu được mức điện áp 5V

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- 15 giao diện truyền thông
- 3 × Giao diện I2C (SMBus / PMBus)
- Tối đa 4 USART / 2 UART (10.5 Mbit / s, giao diện ISO 7816, LIN, IrDA, điều khiển modem)
- 3 SPI (42 Mbits / s), 2 với I2S full-duplex muxed để đạt được độ chính xác lớp âm thanh thông qua âm thanh nội bộ PLL hoặc đồng hồ bên ngoài
- 2 × CAN giao diện (2.0B hoạt động)
- Giao diện SDIO
- Kết nối nâng cao
- Bộ điều khiển thiết bị / máy chủ / bộ điều khiển OTG tốc độ đầy đủ USB 2.0 với PHY trên chip
- Bộ điều khiển thiết bị tốc độ cao / tốc độ cao / bộ điều khiển lưu trữ / tốc độ USB 2.0 với DMA chuyên dụng, PHY tốc độ cao trên toàn bộ chip và ULPI
- 10/100 Ethernet MAC với DMA chuyên dụng: hỗ trợ phần cứng IEEE 1588v2, MII / RMII
- Giao diện camera song song 8-14 bit lên tới 54 Mbytes / s
- Trình tạo số ngẫu nhiên đúng
- Đơn vị tính CRC
- ID duy nhất 96 bit
- RTC: độ chính xác phụ, lịch phần cứng

❖ Điện áp hoạt động và dòng tiêu thụ:

- Nguồn cung cấp: 1.7 → 3.6 V.
- Dòng tiêu thụ: tối đa 160 mA.
- Dải nhiệt độ hoạt động: -40 → 125 °C.

b. Board Arduino Mega

Tổng quan về Arduino:

Arduino là một nền tảng phần cứng mã nguồn mở được sử dụng cho các dự án xây dựng các ứng dụng mạch điện tử. Arduino ra đời tại thị trấn Ivrea thuộc nước Ý và được đặt theo tên một vị vua vào thế kỷ thứ 9 là King Arduino. Arduino chính thức được đưa ra giới thiệu vào năm 2005 như là một công cụ khiêm tốn dành cho các sinh viên của giáo sư Massimo Banzi, là một trong những người phát triển Arduino, tại trường Interaction Design Institute Ivrea (IDI), Italia.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Cấu tạo của Arduino là một mạch vi điều khiển, sử dụng bộ xử lý Atmel 8 bit hoặc Atmel 32 bit. Nó được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác bằng ngôn ngữ C, C++. Đặc điểm nổi bật của Arduino là tính chất nguồn mở từ phần cứng tới phần mềm, môi trường phát triển ứng dụng cực kỳ dễ sử dụng, ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Ngoài ra, Arduino còn được ưa chuộng là do mức giá rất thấp phù hợp với sinh viên.

Tính tới thời điểm này thì Arduino đã cho ra mắt rất nhiều loại board mạch, mỗi loại đều có các thông số khác nhau và phục vụ cho các ứng dụng khác nhau như: Arduino UNO dành cho người mới bắt đầu, Arduino Mega 2560 dành cho các ứng dụng phức tạp, Arduino Lilypad dành cho các ứng dụng gắn lên quần áo hoặc đeo... Trong đề tài này, chúng ta chỉ tập trung tìm hiểu sâu về board mạch Arduino Mega2560.

Giới thiệu board mạch Arduino Mega2560:

Arduino Mega 2560 là board mạch vi điều khiển sử dụng chip xử lý Atmega2560. Nó có ngoại hình khá lớn so với các board mạch Arduino khác do sử dụng chip xử lý lớn, có nhiều chân I/O, chân Analog.

❖ Các thông số kỹ thuật của bo mạch Arduino Mega 2560:

Thuộc tính	Thông số
Chip xử lý	Atmega2560
Điện áp hoạt động	5V
Điện áp vào (khuyến nghị)	7-12V
Điện áp vào (giới hạn)	6-20V
Tổng số chân I/O	54 (có 15 chân có thể phát xung PWM)
Tổng số chân analog	16
Dòng DC trên mỗi chân I/O	20 mA
Dòng DC trên chân 3.3V	50 mA
Dung lượng bộ nhớ Flash	256 KB trong đó 8KB dùng cho bootloader
Dung lượng bộ nhớ SRAM	8 KB

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Dung lượng bộ nhớ EEPROM	4 KB
Tốc độ thạch anh	16 MHz
Chân LED được tích hợp sẵn	13
Độ dài	101.52 mm
Độ rộng	53.3 mm
Cân nặng	37 g

❖ Nguồn cấp cho Arduino:

Arduino Mega 2560 có thể được cấp nguồn bằng cổng USB hoặc bằng nguồn ngoài và việc chọn nguồn cấp được diễn ra hoàn toàn tự động. Tức là ta có thể cấp cả 2 nguồn vào cùng lúc, nếu nguồn ngoài không có hoặc quá bé thì Arduino sẽ lấy nguồn từ cổng USB và ngược lại.

Nguồn ngoài có thể lấy từ adapter AC-DC thông qua jack cắm 3.5mm hoặc từ pin bằng cách nối cực dương của pin vào chân Vin và cực âm vào chân GND. Dù cho ta dùng nguồn nào thì điện áp cấp vào phải nằm trong ngưỡng 7-12 V theo khuyến cáo của nhà sản xuất. Nếu ta cấp nguồn dưới 7V vào thì chân 5V sẽ không cho ra đủ điện áp 5V, mạch sẽ thiếu ổn định. Còn nếu cấp nguồn lớn hơn 12V vào thì IC ổn áp có thể nóng lên, làm hỏng cả board mạch.

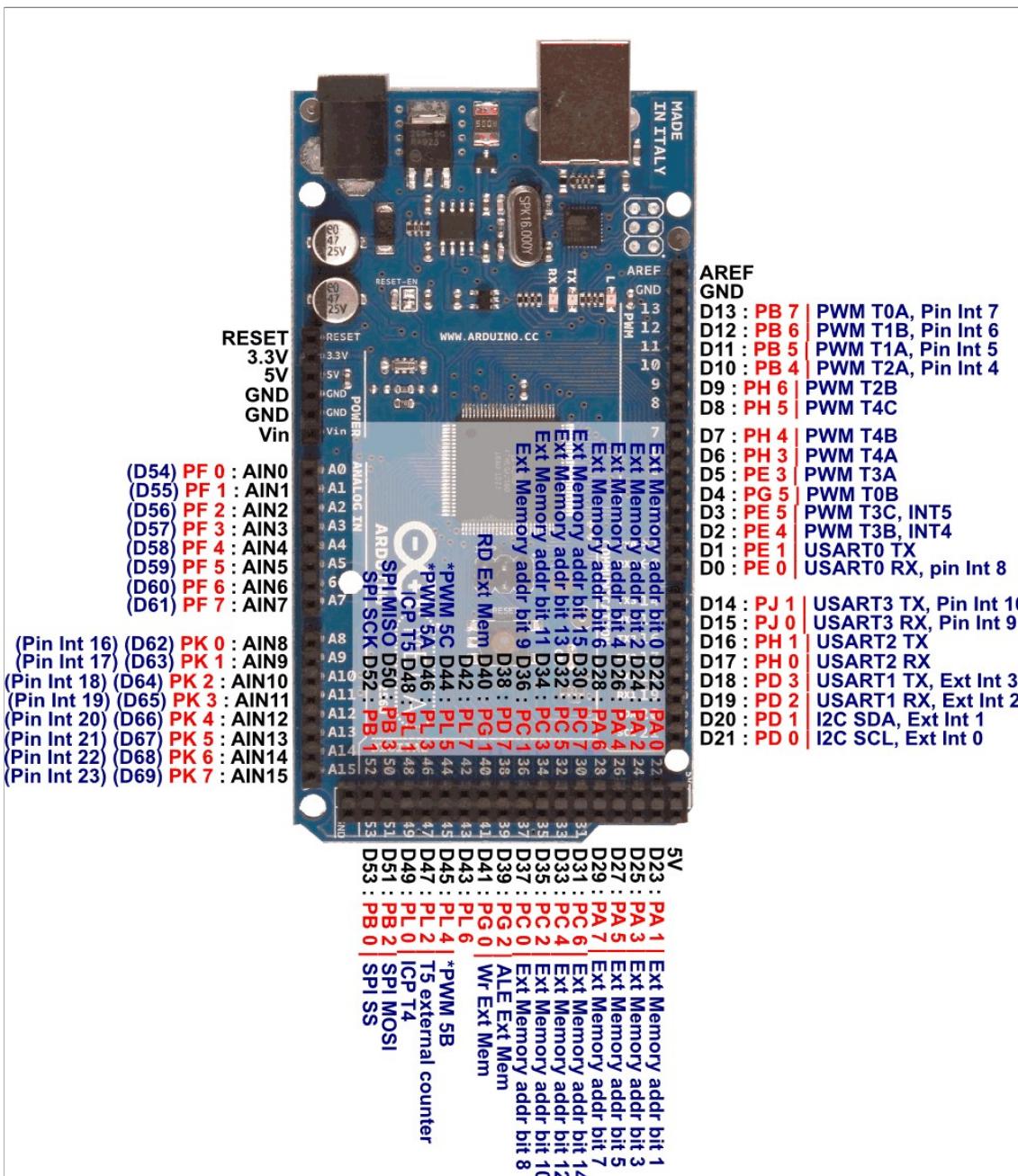
Chân cấp nguồn gồm những chân sau:

- GND (Ground): cực âm của nguồn điện cấp cho Arduino. Khi ta dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
- 5V: cấp điện áp 5V đầu ra, dòng tối đa cho phép ở chân này là 500mA.
- 3.3V: cấp điện áp 3.3V đầu ra, dòng tối đa cho phép ở chân này là 50mA.
- Vin (Voltage Input): để cấp nguồn ngoài cho Arduino, ta nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
- IOREF: điện áp hoạt động của vi điều khiển trên Arduino có thể được đo ở chân này. Mặc dù vậy ta không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn mà chỉ là tham chiếu điện áp hoạt động của vi xử lý.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- #### ❖ Các ngõ vào/ra (I/O pins) của Arduino Mega2560:

Arduino Mega 2560 có tổng cộng 54 chân digital, mỗi chân đều có thể là ngõ vào hoặc ngõ ra tùy theo ta lập trình. Chúng chỉ cho ra 2 mức điện áp là 0V hoặc 5V với dòng vào ra là 20mA ở điều kiện hoạt động được khuyến nghị theo nhà sản xuất, tối đa là 40mA. Nếu vượt quá ngưỡng 40mA này thì board mạch sẽ hư hỏng. Ngoài ra trên mỗi chân digital còn có một điện trở nội kéo lên với giá trị 20-50 k Ω , mặc định điện trở này sẽ không được kết nối với chân digital.



Hình 2.18: Arduino Mega2560

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

❖ Một số chân có chức năng đặc biệt:

- Serial: gồm 4 cổng serial là các chân 0 (RX) và 1 (TX); Serial 1: 19 (RX) và 18 (TX); Serial 2: 17 (RX) và 16 (TX); Serial 3: 15 (RX) và 14 (TX) dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial.
- Ngắt ngoài: gồm 6 chân ngắt ngoài là chân 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) và 21 (interrupt 2). Những chân này dùng để kích hoạt ngắt khi chân có mức điện áp thấp, cao, xung cạnh lên, xung cạnh xuống hoặc có sự thay đổi điện áp.
- PWM: gồm 15 chân là các chân 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 44, 45, 46 cho phép ta xuất ra xung PWM với độ phân giải 8bit (giá trị từ $0 \rightarrow 2^8 - 1$ tương ứng với các mức điện áp từ 0V đến 5V).
- SPI: gồm các chân 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). 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.
- LED: một Led tích hợp trên board mạch đã được kết nối sẵn vào chân 13. Khi điện áp trên chân này ở mức cao thì Led sẽ sáng và ngược lại Led sẽ tắt.
- TWI: gồm 2 chân 20 (SDA) và 21 (SCL) hỗ trợ giao tiếp TWI/I2C với các thiết bị khác.
- Reset: khi nhấn nút reset thì ta đã cấp mức điện áp thấp vào chân reset làm khởi động lại vi điều khiển.
- AREF: đây là chân mà ta đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu cấp điện áp 2.5V vào chân này thì ta có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải là 10 bit.

Arduino Mega 2560 còn có 16 chân ngõ vào analog cung cấp độ phân giải tín hiệu 10bit ($0 \rightarrow 2^{10} - 1$) để đọc giá trị điện áp trong khoảng 0V → 5V.

❖ Bộ nhớ:

Arduino Mega 2560 được trang bị chip Atmega2560 đã tích hợp sẵn 256 KB dung lượng bộ nhớ Flash, 8 KB bộ nhớ SRAM và 4KB bộ nhớ EEPROM. Trong 256 KB bộ nhớ Flash thì 8 KB dành cho bootloader, tức là ta chỉ có 248 KB để dành cho việc lưu chương trình. Bộ nhớ SRAM có đặc điểm là mất dữ liệu khi mất điện nên dùng để lưu các giá trị biến trong chương trình, còn bộ nhớ EEPROM thì không mất dữ liệu khi mất

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

điện nên là ta sẽ lưu các biến dữ liệu quan trọng vào bộ nhớ này để khi xảy ra sự cố về điện thì mạch vẫn chạy đúng.

Bảng 2.2: So sánh giữa các loại bộ nhớ

Loại	Mất dữ liệu khi mất điện	Xoá nhiều lần	Tốc độ	Giá thành
SRAM	Có	Không giới hạn	Nhanh	Đắt
EEPROM	Không	Giới hạn	Nhanh cho đọc, chậm cho xoá và ghi	Đắt
Flash	Không	Giới hạn	Nhanh cho đọc, chậm cho xoá và ghi	Vừa phải

2.2.5 Module SIM900A

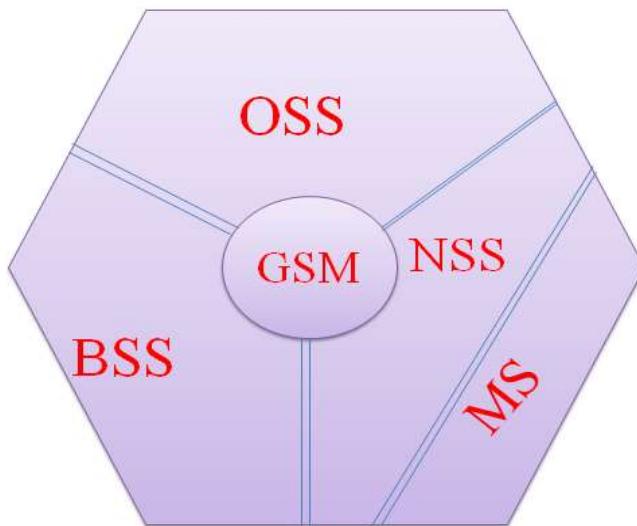
a. Hệ thống thông tin di động GSM

❖ Đặc điểm của công nghệ GMS

Công nghệ GSM có một số đặc điểm cơ bản như sau:

- Cho phép gửi và nhận những mẫu tin nhắn văn bản bằng kí tự dài đến 126 kí tự.
- Cho phép chuyển giao và nhận dữ liệu, FAX giữa các mạng GSM với tốc độ hiện hành lên đến 9600 bps.
- Tính phủ sóng cao: Công nghệ GSM không chỉ cho phép chuyển giao trong toàn mạng mà còn chuyển giao giữa các mạng GSM trên toàn cầu mà không có một sự thay đổi, điều chỉnh nào. Đây là một tính năng nổi bật nhất của công nghệ GSM (dịch vụ roaming).
- Sử dụng công nghệ phân chia theo thời gian TDM (Time division multiplexing) để chia ra 8 kênh full rate hay 16 kênh half rate.
- Công suất phát của máy điện thoại được giới hạn tối đa là 2 watts đối với băng tần GSM 850/900Mhz và tối đa là 1 watts đối với băng tần GSM 1800/1900Mhz.
- Mạng GSM sử dụng 2 kiểu mã hoá âm thanh để nén tín hiệu âm thanh 3,1khz đó là mã hoá 6 và 13kbps gọi là Full rate (13kbps) và half rate (6kbps).

❖ Cấu trúc của mạng GSM



Hình 2.19: Cấu trúc của mạng GSM

Hệ thống GSM được chia thành nhiều hệ thống con như sau:

- Phân hệ chuyển mạch NSS (Network Switching Subsystem).
- Phân hệ trạm gốc BSS (Base Station Subsystem).
- Phân hệ bảo dưỡng và khai thác OSS (Operation Subsystem).
- Trạm di động MS (Mobile Station).

b. Tổng quan về tin nhắn SMS

❖ Giới thiệu về tin nhắn SMS

SMS là từ viết tắt của Short Message Service. Đó là một công nghệ cho phép gửi và nhận các tin nhắn giữa các điện thoại với nhau. Dữ liệu có thể được lưu giữ bởi một tin nhắn SMS là rất giới hạn. Một tin nhắn SMS có thể chứa tối đa là 140 byte (1120 bit) dữ liệu. Vì vậy, một tin nhắn SMS chỉ có thể chứa:

- 160 ký tự nếu như mã hóa ký tự 7 bit được sử dụng.
- 70 ký tự nếu như mã hóa ký tự 16 bit Unicode UCS2 được sử dụng.
- Các tin nhắn SMS có thể được gửi và đọc tại bất kỳ thời điểm nào.
- Ngày nay, hầu hết mọi người đều có điện thoại di động của riêng mình và mang nó theo người hầu như cả ngày. Với một điện thoại di động, bạn có thể gửi và đọc các tin nhắn SMS bất cứ lúc nào bạn muốn, sẽ không gặp khó khăn gì khi bạn đang ở trong văn phòng hay trên xe bus hay ở nhà...
- Tin nhắn SMS có thể được gửi tới các điện thoại đang tắt nguồn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Nếu như không chắc chắn cho một cuộc gọi nào đó thì bạn có thể gửi một tin nhắn SMS đến bạn của bạn thậm chí khi người đó tắt nguồn máy điện thoại trong lúc bạn gửi tin nhắn đó. Hệ thống SMS của mạng điện thoại sẽ lưu trữ tin nhắn đó rồi sau đó gửi nó tới người bạn đó khi điện thoại của người bạn này mở nguồn.
- Các tin nhắn SMS ít gây phiền phức trong khi bạn vẫn có thể giữ liên lạc với người khác
- Việc đọc và viết các tin nhắn SMS không gây ra ồn ào. Trong khi đó, bạn phải chạy ra ngoài khỏi rạp hát, thư viện hay một nơi nào đó để thực hiện một cuộc điện thoại hay trả lời một cuộc gọi. Bạn không cần phải làm như vậy nếu như tin nhắn SMS được sử dụng.

❖ Cấu trúc tin nhắn SMS

Các tiêu chuẩn của tin nhắn SMS xác định những thông tin nào được gửi trong một tin nhắn, các bit của mã nhị phân tạo nên một lá thư và làm thế nào dữ liệu này được tổ chức, gửi và nhận qua lại giữa các thiết bị với nhau? Định dạng dữ liệu của một tin nhắn không chỉ có nội dung tin nhắn mà còn có thêm thời gian, số điện thoại gửi đến.

Chi tiết tin nhắn được mô tả từ các đơn vị giao thức PDU (Protocol Description Unit), trong các hình thức của một chuỗi hệ thập lục phân và bán số thập phân. Hệ thập lục phân là một hệ đếm có 16 ký tự, từ 0 đến 9 và từ A đến F để đại diện cho các giá trị từ 10 đến 15.

Định dạng PDU bao gồm các mảng thông tin sau, vài bit đầu tiên chứa thông tin về nơi gửi đến, trong đó có trung tâm tin nhắn, số của người gửi. Các bit tiếp theo là chuỗi tin nhắn.

Tiếp theo, thông tin người gửi và người nhận được chuyển thành một dạng giao thức và một thẻ để xác định chương trình mã hóa dữ liệu đã được sử dụng. Thẻ xác định mã nhằm giúp trung tâm nhận tin nhắn biết được tin nhắn đã sử dụng chương trình mã hóa nào để giải mã lại tin nhắn đó. Ngoài ra còn có nhãn thời gian và thông tin về độ dài của tin nhắn.

c. Giới thiệu Module SIM900A và tập lệnh AT

❖ Tổng quan về Module SIM900A

Một modem GSM là một modem wireless, nó làm việc cùng với một mạng wireless GSM. Một modem wireless thì cũng hoạt động giống như một modem quay số. Điểm khác nhau chính ở đây là modem quay số thì truyền và nhận dữ liệu thông qua một

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

đường dây điện thoại cố định trong khi đó một modem wireless thì việc gửi nhận dữ liệu thông qua sóng.

Giống như một điện thoại di động GSM, một modem GSM yêu cầu 1 thẻ sim với một mạng wireless để hoạt động.

Module SIM 900A là một trong những loại modem GSM. Nhưng Module SIM 900A được nâng cao hơn có tốc độ truyền dữ liệu nhanh hơn. Nó sử dụng công nghệ GSM/GPRS hoạt động ở băng tần EGSM 900Mhz VÀ DCS 1800 Mhz.



Hình 2.20: Module Sim900A

Chức năng các chân:

- Chân +5V: chân cung cấp nguồn dương 5VDC để module sim900A hoạt động.
- Chân DTR: Chân đầu cuối dữ liệu.
- Chân TX: chân truyền UART (TX) dùng để truyền dữ liệu.
- Chân RX: chân nhận UART (RX) dùng để nhận và xử lý dữ liệu.
- Chân Headphone: chân xuất dữ liệu âm thanh ra loa thoại.
- Chân Microphone: chân kết nối với mic nếu muốn đàm thoại.
- Chân Reset: chân reset module sim.
- Chân GND: chân nối mass.

❖ Khảo sát tập lệnh AT của Module SIM900A

Các lệnh AT là các hướng dẫn được sử dụng để điều khiển một modem. AT là một cách viết gọn của chữ Attention. Mỗi dòng lệnh của nó bắt đầu với “AT” hay “at”. Đó là lý do tại sao các lệnh modem được gọi là các lệnh AT. Nhiều lệnh của nó được sử dụng để điều khiển các modem quay số sử dụng dây mới (wired dial-up modems), chẳng hạn như ATD (Dial), ATA (Answer), ATH (Hoot control) và ATO (return to online data state) cũng được hỗ trợ bởi các modem GSM/GPRS và các điện thoại di động.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Bên cạnh bộ lệnh AT thông dụng này, các modem GSM/GPRS và các điện thoại di động còn được hỗ trợ bởi một bộ lệnh AT đặc biệt đối với công nghệ GSM. Nó bao gồm các lệnh liên quan tới SMS như AT+CMGS (gửi tin nhắn SMS), AT+CMSS (gửi tin nhắn SMS từ một vùng lưu trữ), AT+CMGL (chuỗi liệt kê các tin nhắn SMS) và AT+CMGR (đọc tin nhắn SMS). Ngoài ra, các modem GSM còn hỗ trợ một bộ lệnh AT mở rộng.

➤ Một số thuật ngữ

- Carriage return (được dịch từ mã ASCII là 0x0D).
- Line Feed (được dịch từ mã ASCII là 0x0A).
- MT: Mobile Terminal: Thiết bị đầu cuối mạng (chính là module).
- TE: Terminal Equipment: Thiết bị đầu cuối (chính là vi điều khiển).

➤ Cú pháp lệnh AT

- Lệnh khởi đầu: luôn là “AT” hoặc “at”.
- Lệnh kết thúc là: ký tự <CR>.
- Thông thường sau mỗi lệnh AT là một đáp ứng, cấu trúc của đáp ứng này là:
“<CR><LF><Response><CR><LF>”

Cú pháp chính của lệnh AT có thể được phân chia thành 3 loại: cú pháp có cấu trúc cơ bản, cú pháp có cấu trúc tham số S, cú pháp có cấu trúc mở rộng.

Với các cú pháp nêu trên thì các lệnh có thể hoạt động ở nhiều chế độ khác nhau. Các chế độ này được thống kê ở bảng bên dưới như sau:

Bảng 2.3: Các chế độ lệnh AT

<Lệnh kiểm tra>	AT+<x>=?	Thông kê lại các tham số trong câu lệnh và các giá trị có thể thiết lập cho tham số.
<Lệnh đọc>	AT+<x>?	Đọc nội dung tin nhắn được gửi đến, kiểm tra giá trị tin nhắn về mặt dữ liệu.
<Lệnh thiết lập>	AT+<x>=<...>	Được sử dụng để thiết lập các giá trị cho tham số.
<Lệnh thực thi>	AT+<x>	Thực thi nội dung tin nhắn được tiến hành bên trong của Module sim.

❖ Một số lệnh AT cơ bản

- Lệnh ATZ

Lệnh ATZ dùng thiết lập lại (reset) tất cả các tham số hiện tại theo mẫu được người dùng định nghĩa. Lệnh trả về của modem là lệnh OK. Mẫu người dùng định nghĩa trước đó được lưu trên bộ nhớ cố định. Nếu không thiết lập lại được theo mẫu của người dùng

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

định nghĩa thì nó sẽ reset lại theo đúng các tham số mặc định của nhà sản xuất. Bất cứ lệnh AT cộng thêm nào trên cùng một dòng với lệnh ATZ đều không được thực hiện.

- Lệnh AT+CMGR: đọc nội dung tin nhắn

Lệnh AT+CMGR được dùng để đọc tin nhắn trên một ngăn nào đó trên sim điện thoại. Cấu trúc lệnh như sau: AT+CMGR=i, với i là ngăn bộ nhớ chứa tin nhắn trong sim. Đáp ứng trả về là lệnh OK nếu ngăn i có chứa tin nhắn. Nếu ngăn i không chứa tin nhắn thì sẽ xuất hiện thông báo lỗi trả về ERROR. Ví dụ khi gõ lệnh AT+CMGR=1 thì sim900 sẽ đọc tin nhắn tại ngăn số 1 của bộ nhớ sim điện thoại gắn ngoài.

- Lệnh AT+CMGS: gửi tin nhắn SMS

Lệnh AT+CMGS dùng để gửi tin nhắn SMS tới một số điện thoại cho trước. Cú pháp gửi tin như sau:

- AT+CMGS=“số điện thoại cần gửi”<CR>
- Nội dung tin nhắn
- ESC/Ctrl Z

Số điện thoại cần gửi phải được đặt trong dấu ngoặc kép. Sau khi gõ xong số điện thoại thì cần thực hiện lệnh enter để xuống dòng và bắt đầu nội dung tin nhắn. Kết thúc lệnh này bằng việc thực hiện lệnh Cltr Z.

Ví dụ lệnh gửi tin nhắn tới số 0906600837 với nội dung “abcd” được thực hiện tại chế độ text trong phần mềm lập trình CCS như sau:

```
Printf("AT+CMGS=\\"0906600837\\r\\n");  
delay ms (500);  
printf("abcd"); // nội dung tin nhắn  
delay ms (500);  
putc(26); // ctrl +z
```

- AT+CMGD: xóa tin nhắn SMS

Lệnh AT+CMGD dùng để xóa tin nhắn SMS trên sim. Cấu trúc lệnh như sau: AT+CMGD=i

Với i là ngăn nhớ chứa tin nhắn cần xóa. Nếu ngăn i chứa tin nhắn thì đáp ứng trả về là OK, nếu việc thực hiện tin nhắn không thực hiện được như ngăn i không có tin nhắn, hoặc kết nối tới sim, lỗi sóng thì trả về sẽ là ERROR.

Ví dụ xóa tin nhắn từ ngăn số 1 của sim: AT+CMGD=1.

- Lệnh ATE : thiết lập chế độ lệnh phản hồi

Lệnh này dùng để thiết lập chế độ lệnh phản hồi trả lại. Đáp ứng trả lại OK.

Lệnh AT có hai tham số hoàn toàn khác nhau:

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- ATE0: tắt chế độ phản hồi.
- ATE1: bật chế độ phản hồi.

Khi giao tiếp Module sim900 với phần mềm putty trên máy tính, nếu ta dùng lệnh ATE0 thì khi gõ lệnh AT khác thì không nhìn thấy lệnh ta gõ mà chỉ nhìn thấy kết quả trả về của sim900. Ngược lại, khi dùng lệnh ATE1 thì sẽ nhìn thấy cả lệnh gõ lên và lệnh sim900 trả về.

- Lệnh AT&W: lưu các tham số hiện tại vào mẫu người dùng

Lệnh AT&W được dùng để lưu cấu hình cài đặt được thiết lập bởi các lệnh ATE và AT+CLIP vào bộ nhớ (Lệnh AT+CLIP để cài đặt cuộc gọi). Đáp ứng trả về khi thực hiện lệnh này là OK.

- Lệnh AT+CMGF: lựa chọn định dạng tin nhắn SMS

Lệnh AT+CMGF dùng để lựa chọn định dạng tin nhắn SMS, với hai chế độ là text và PDU [14], cụ thể như sau:

- AT+CMGF=1: lựa chọn sử dụng tin nhắn ở chế độ văn bản
- AT+CMGF=0: lựa chọn sử dụng tin nhắn ở chế độ PDU

Đáp ứng trả về là “OK” nếu như modem hỗ trợ, ngược lại, nếu modem không hỗ trợ chế độ định dạng tin nhắn là text hoặc PDU thì đáp ứng trả về sẽ là “ERROR”.

- Lệnh AT+CNMI : thông báo có tin nhắn mới đến

Lệnh này dùng để thông báo có tin nhắn mới đến. Với các tham số khác nhau thì mỗi khi có tin nhắn, đáp ứng trả về cũng sẽ khác nhau.

Ví dụ về các lệnh AT+CNMI khác nhau khi cùng nhận một tin nhắn SMS có nội dung giống nhau:

- AT+CNMI = 1, 1, 0, 0, 0 sẽ trả về:
+CMTI: "SM", 10
- AT+CNMI = 2, 2, 0, 0, 0 sẽ trả về:
+CMT: "+84906600837", "", "17/07/23, 14:21:42+28" Abcdef

Như vậy, chúng ta thấy rằng, với trường hợp 1 thì nội dung tin nhắn được lưu trực tiếp vào ngăn số 10 của sim. Nội dung của tin nhắn này chỉ được đọc bằng lệnh AT+CMGR=10. Còn trong trường hợp số 2, nội dung tin nhắn được hiển thị ra cùng với thời gian và số điện thoại.

- Lệnh AT+CSAS: lưu các thiết lập tin nhắn SMS

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Lệnh AT+CSAS dùng để lưu các thiết lập SMS do người dùng đã cài đặt trước đó. Lệnh này sẽ lưu trực tiếp các thông số đã cài đặt cho tin nhắn SMS như các lệnh AT+CMGF=1; AT+CNMI = 2, 2, 0, 0, 0...và còn nhiều lệnh khác liên quan tới tin nhắn SMS đều được lưu lại bởi lệnh AT+CSAS này.

2.2.6 Dòng chip Wi-Fi ESP8266

Chip ESP8266 được phát triển bởi Espressif để cung cấp giải pháp giao tiếp Wi-Fi cho các thiết bị IoT. Điểm đặc biệt của dòng ESP8266 là nó được tích hợp các mạch RF như balun, antenna switches, TX power amplifier và RX filter ngay bên trong chip với kích thước rất nhỏ chỉ 5x5mm nên các board sử dụng ESP8266 không cần kích thước board lớn cũng như không cần nhiều linh kiện xung quanh. Ngoài ra, giá thành của ESP8266 cũng rất thấp đủ để hấp dẫn các nhà phát triển sản phẩm IoT.

a. Cấu trúc phần cứng của dòng chip ESP8266

- Sử dụng 32-bit MCU core có tên là Tensilica
- Tốc độ system clock có thể set ở 80MHz hoặc 160MHz
- Không tích hợp bộ nhớ Flash để lưu chương trình
- Tích hợp 50KB RAM để lưu dữ liệu ứng dụng khi chạy
- Có đầy đủ các ngoại vi chuẩn để giao tiếp như 17 GPIO, 1 Slave SDIO, 3 SPI, 1 I2C, 1 I2S, 2 UART, 2 PWM.
- Tích hợp các mạch RF để truyền nhận dữ liệu ở tần số 2.4GHz
- Hỗ trợ các hoạt động truyền nhận các IP packages ở mức hardware như Acknowledgement, Fragmentation và Defragmentation, Aggregation, Frame, Encapsulation... (và phần stack TCP/IP sẽ được thực hiện trên firmware của ESP8266).
- Do không hỗ trợ bộ nhớ Flash nên các board sử dụng ESP8266 phải gắn thêm chip Flash bên ngoài và thường là Flash SPI để ESP8266 có thể đọc chương trình ứng dụng với chuẩn SDIO hoặc SPI.

b. Mạch nguyên lý đầy đủ của ESP8266

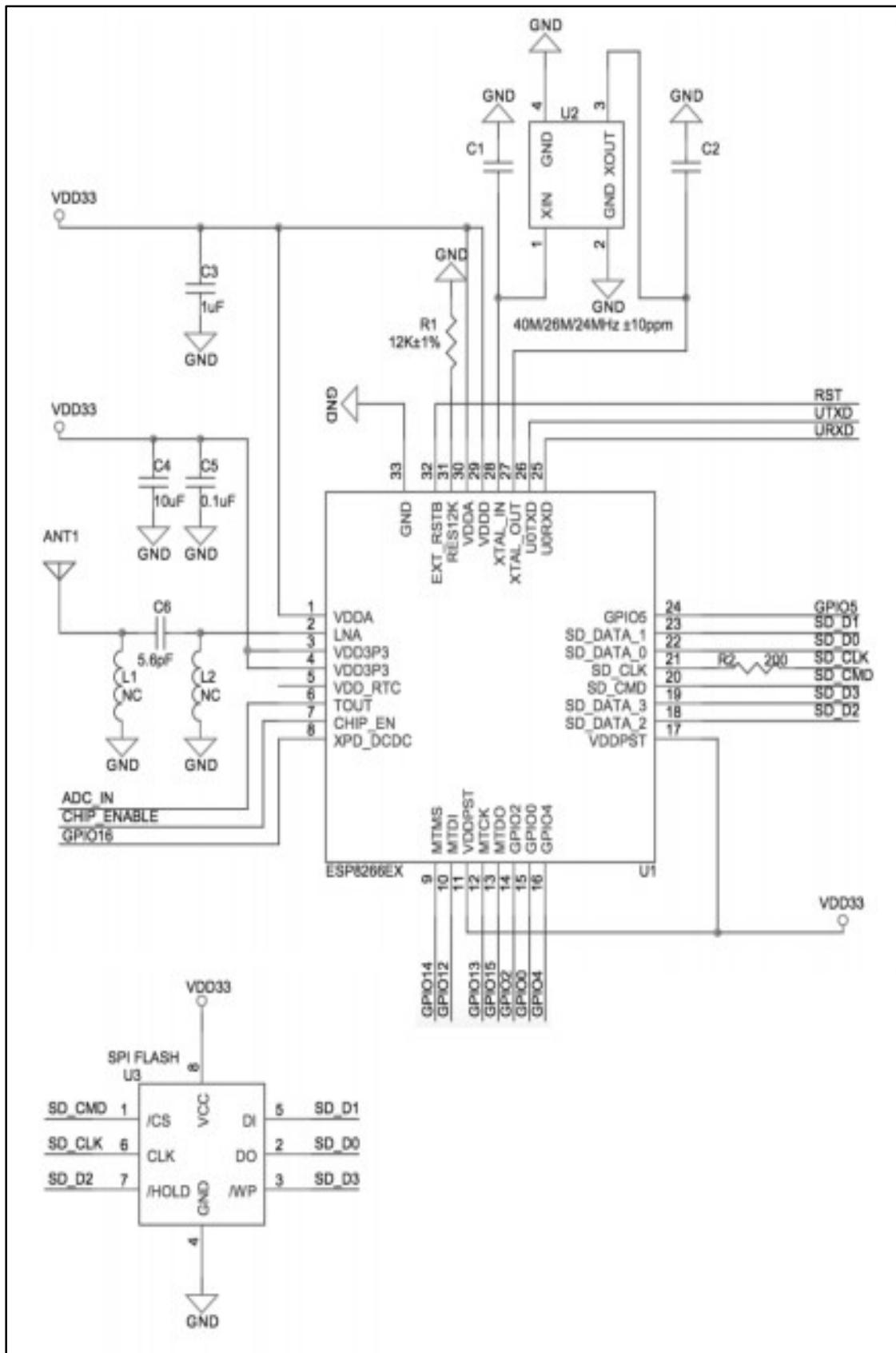
Chúng ta có thể thấy board ESP8266 chỉ cần thạch anh và SPI flash chip và vài linh kiện điện trở rất đơn giản. Do đó việc tích hợp giao tiếp Wi-Fi vào board ứng dụng với ESP8266 rất dễ dàng và nhanh chóng.

Mô hình lập trình ứng dụng với ESP8266 có thể chia làm 2 loại như sau:

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Sử dụng firmware được cung cấp bởi Espressif và giao tiếp thông qua AT commands.
- Lập trình firmware trực tiếp vào ESP8266 sử dụng bộ thư viện SDK cung cấp bởi Espressif.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.21: Sơ đồ nguyên lý ESP8266

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

c. Các loại module cho ESP8266 trên thị trường

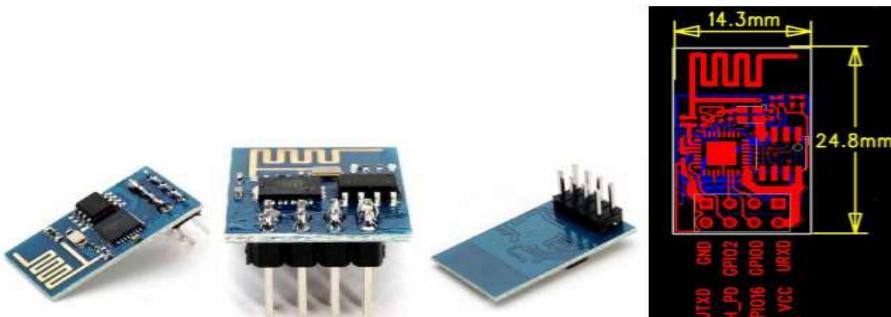
Ngoài trừ module ESP-WROOM-02 được phát triển bởi chính Espressif cho mục đích nghiên cứu các tính năng của ESP8266, các module ứng dụng phổ biến hiện nay của ESP8266 đều được phát triển bởi công ty AI-Thinker.

Hiện tại có khá nhiều module khác nhau cho ESP8266 được sản xuất bởi công ty AI-Thinker. Đặc điểm khác nhau giữa các module này bao gồm:

- Loại anten sử dụng (PCB anten, chip anten hoặc gắn anten ngoài)
- Dung lượng của chip Flash SPI trên board
- Kích thước board của module.
- Có gắn khung nhôm chống nhiễu hay không
- Số lượng pin GPIO đưa ra chân kết nối

Hiện tại AI-Thinker sản xuất 14 loại module cho ESP từ module ESP-01 đến ESP-14. Ở thị trường Việt Nam thì các module là ESP-01, ESP-07 và ESP-12, ESP NodeMCU khá phổ biến.

❖ ESP-01

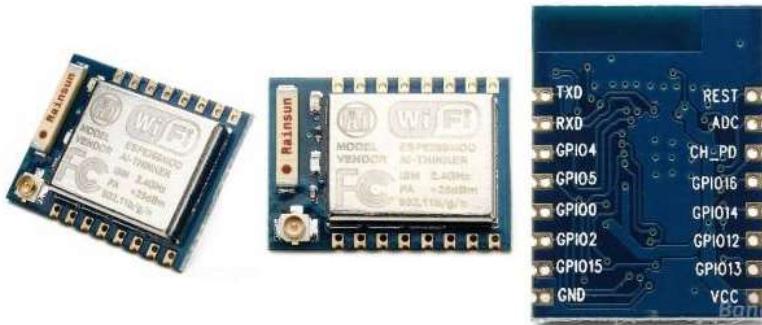


Hình 2.22: Hình ảnh ESP-01

- Sử dụng on-board PCB antenna.
- Có 2 LED trên board để báo nguồn và báo TX.
- Cung cấp 3 chân GPIO (GPIO0, GPIO2 và GPIO6) và 2 chân TXD/RXD cho UART.
- Dung lượng SPI Flash 4 MByte.
- Đưa chân ra jumper luôn nên có thể kết nối trực tiếp với các board khác 1 cách nhanh chóng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

❖ ESP-07

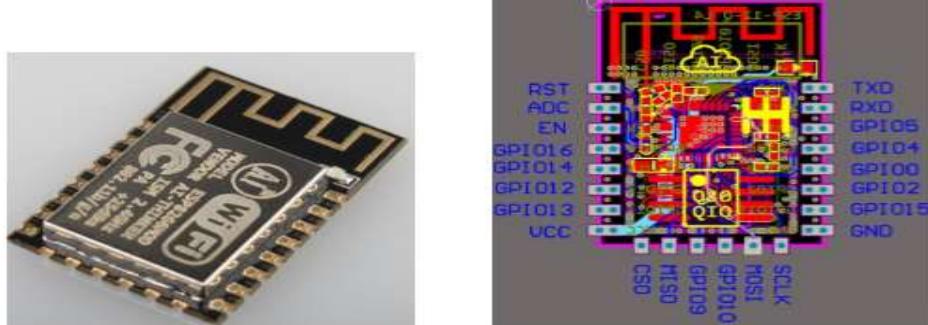


Hình 2.23: Hình ảnh ESP-07

Sử dụng chip anten on-board và có IPEX connector hỗ trợ gắn thêm anten ngoài để tăng khoảng cách truyền.

- Có 2 LED trên board để báo nguồn và báo TX.
- Đưa ra 9 chân GPIO, 2 chân TX/RX cho UART, 1 chân REST để reset chip, 1 chân ADC, 1 chân CH_PD để đưa chip vào chế độ low power.
- Dung lượng SPI Flash trên board là 4MByte.
- Có thẻ hàn thêm jumper để kết nối trực tiếp với board khác hoặc hàn trực tiếp lên board ứng dụng.

❖ ESP-12



Hình 2.24: Hình ảnh ESP-12

- Sử dụng PCB anten on-board.
- Đưa ra 11 chân GPIO, 2 chân TX/RX cho UART, các chân cho SPI, chân RST để reset chip, 1 chân ADC.
- Dung lượng SPI Flash là 4MByte.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Có thẻ hàn jumper để cắm dây vào các board khác hoặc hàn trực tiếp lên board ứng dụng.

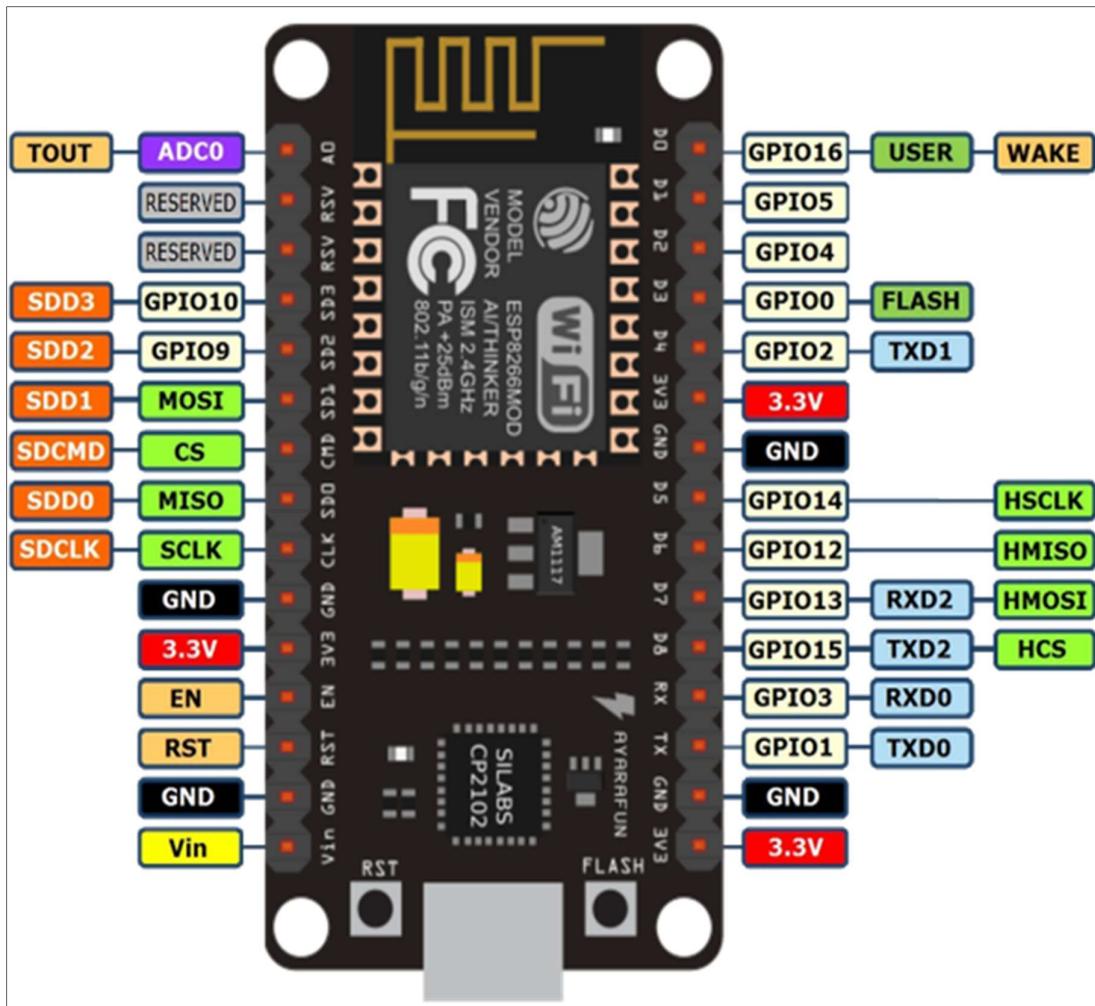
❖ ESP NodeMCU

Kit RF thu phát Wifi ESP8266 NodeMCU là phiên bản NodeMCU sử dụng IC nạp giá rẻ CP2102 từ Lolin với bộ xử lý trung tâm là module Wifi SoC ESP8266, kit có thiết kế dễ sử dụng và đặc biệt là có thẻ sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.

Thông số kỹ thuật:

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua.
- Chip nạp và giao tiếp UART: CH340.
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cáp nguồn: 5VDC MicroUSB hoặc Vin.
- GIPO giao tiếp mức 3.3 VDC.
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 59 x 32mm.



Hình 2.25: ESP8266 NodeMCU

2.2.7 Công nghệ RFID

a. Công nghệ RFID và quá trình phát triển

- ❖ Giới thiệu về công nghệ RFID

Công nghệ RFID (Radio Frequency Identification) cho phép một thiết bị đọc thông tin chứa trong chip không cần tiếp xúc trực tiếp ở khoảng cách xa, không thực hiện bất kỳ giao tiếp vật lý nào hoặc giữa hai vật không nhìn thấy. Công nghệ này cho ta phương pháp truyền, nhận dữ liệu từ một điểm đến một điểm khác.

Kỹ thuật RFID sử dụng truyền thông không dây trong dải tần sóng vô tuyến để truyền dữ liệu từ các tag (thẻ) đến các reader (bộ đọc). Tag có thể được đính kèm hoặc gắn vào đối tượng được nhận dạng chẳng hạn sản phẩm, hộp hoặc giá kệ (pallet). Reader scan dữ liệu của tag và gửi thông tin đến cơ sở dữ liệu có lưu trữ dữ liệu của tag. Ví dụ: các tag có thẻ được đặt trên kính chắn gió xe hơi để hệ thống thu phí đường có thể nhanh

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

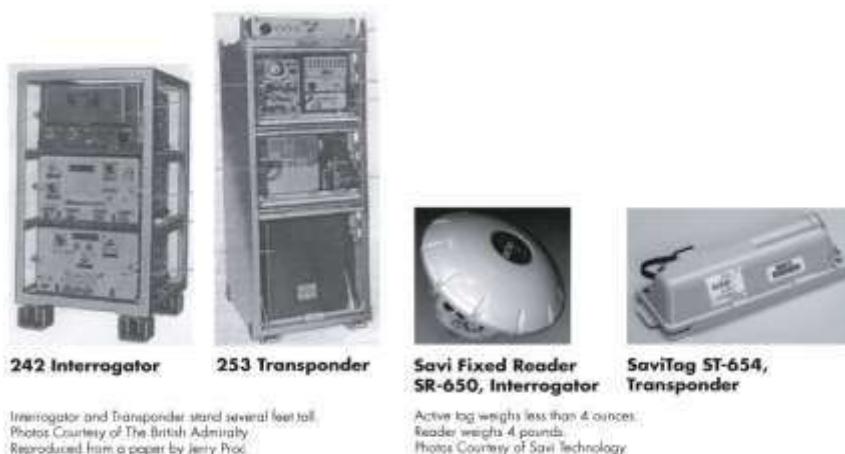
chóng nhận dạng và thu tiền trên các tuyến đường.

Dạng đơn giản nhất được sử dụng hiện nay là hệ thống RFID bị động làm việc như sau: reader truyền một tín hiệu tần số vô tuyến điện từ qua anten của nó đến một con chip. Reader nhận thông tin trả lại từ chip và gửi nó đến máy tính điều khiển đầu đọc và xử lý thông tin lấy được từ chip. Các chip không tiếp xúc không tích điện, chúng hoạt động bằng cách sử dụng năng lượng nhận từ tín hiệu được gửi bởi reader.

❖ Lịch sử và quá trình phát triển

Năm 1897: Guglielmo Marconi phát hiện ra sóng radio, tạo nền tảng để phát triển RFID.

Năm 1937: phòng thử nghiệm nghiên cứu Naval U.S phát triển hệ thống xác định Friend – or – Foe (IFF) cho phép những đối tượng thuộc về quân ta với quân địch



Hình 2.26: Thiết bị IFF và thiết bị RFID hiện đại ngày nay

Cuối thập kỉ 60 đầu thập kỉ 70: nhiều công ty như Sensormatic and Checkpoint Systems giới thiệu những sản phẩm mới ít phức tạp hơn và ứng dụng rộng rãi hơn do công nghệ được tích hợp trong IC, chip nhớ lập trình được. Các công ty bắt đầu phát triển thiết bị giám sát điện tử để bảo vệ và kiểm kê sản phẩm như quần áo trong cửa hàng, sách trong thư viện. Hệ thống RFID thương mại ban đầu này chỉ là hệ thống Tag 1 bit giá rẻ để xây dựng, thực hiện và bảo hành. Tag không đòi hỏi nguồn pin (thụ động) dễ dàng đặt vào sản phẩm và thiết kế để cảnh báo khi tag đến gần bộ đọc, thường đặt tại lối ra vào để phát hiện sự có mặt của tag.

Suốt thập kỉ 70: nghiên cứu và phát triển những dự án để tìm cách dùng IC dựa trên hệ thống RFID. Có nhiều ứng dụng trong công nghiệp tự động, xác định thú vật, theo

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

dõi lưu thông. Tag có đặc điểm: bộ nhớ ghi được, tốc độ đọc nhanh hơn và khoảng cách đọc xa hơn.

Đầu thập niên 80: được áp dụng trong nhiều ứng dụng: đặt tại đường ray ở Mỹ, đánh dấu thú vật trên nông trại ở châu Âu. Hệ thống RFID còn dùng trong nghiên cứu động vật hoang dã đánh dấu các loài thú quý và nguy hiểm.

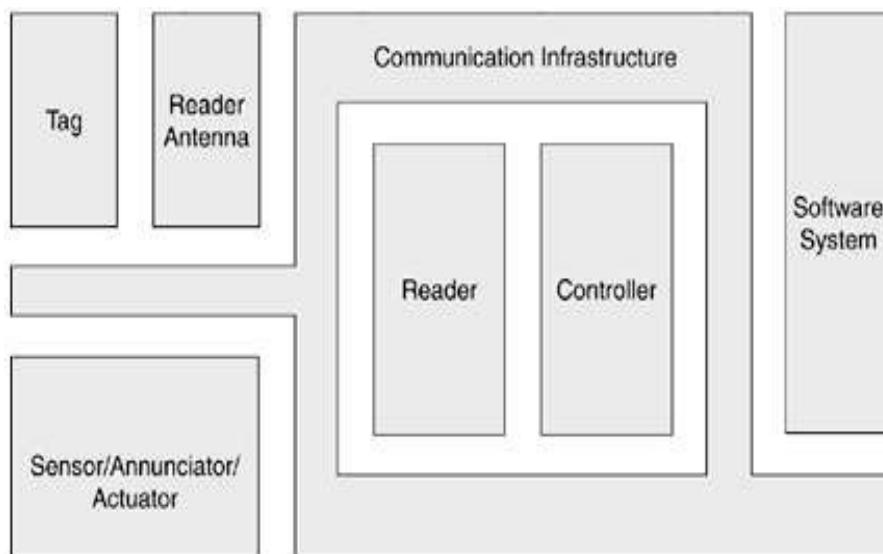
Đầu năm 1990: xuất hiện nhiều hệ thống thu phí điện tử, tiêu chuẩn hóa các đặc tính kỹ thuật như tần số hoạt động và giao thức giao tiếp phần cứng.

Cuối thế kỷ 20: phát triển nhanh trên phạm vi toàn cầu.

- Mỹ: tạo ra hệ thống xác nhận và đăng ký Texas instrument (TIRIS)
- Châu Âu: phát minh công nghệ liên quan đến việc xác định thẻ thông minh

Cuối những năm 90 đầu năm 2000: EPCglobal được thành lập và hỗ trợ hệ thống mã sản phẩm điện tử (Electronic Product Code Network EPC) và hệ thống này đã trở thành tiêu chuẩn cho xác nhận sản phẩm tự động.

b. Thành phần của một hệ thống RFID



Hình 2.27: Sơ đồ khái niệm về thành phần của một hệ thống RFID

Một hệ thống RFID là một tập hợp các thành phần mà nó thực thi giải pháp RFID.

Một hệ thống RFID bao gồm các thành phần sau:

Thẻ (Tags): là một thành phần bắt buộc đối với mọi hệ thống RFID. Bao gồm: chip bán dẫn nhỏ và anten thu nhỏ trong một số hình thức đóng gói.

Đầu đọc (Reader): là thành phần bắt buộc, thực hiện việc đọc, ghi dữ liệu lên Tag, giao tiếp với máy chủ.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Ăngten (Antena): làm nhiệm vụ bức xạ, thu sóng điện từ và gia công tín hiệu.

Mạch điều khiển (Controller): là thanh phần bắt buộc, tuy nhiên hầu hết reader mới đều có thành phần này gắn liền với chúng. Cho phép các thành phần bên ngoài như con người, chương trình máy tính giao tiếp điều khiển chức năng của reader, annunciator, cơ cấu chấp hành kết hợp với reader.

Cảm biến (sensor), cơ cấu chấp hành (actuator) và bảng tín hiệu điện báo (annunciator): hỗ trợ xuất và nhập của hệ thống.

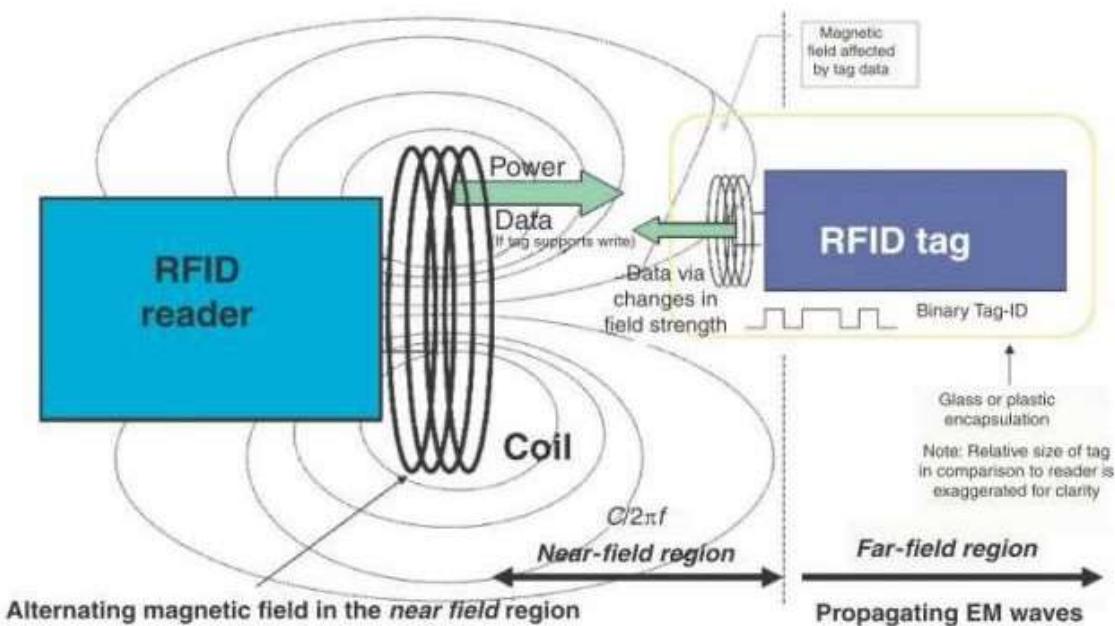
Máy chủ và hệ thống phần mềm: về mặt lí thuyết, một hệ thống RFID có thể hoạt động mà không cần thành phần này. Thực tế, một hệ thống RFID gần như không có ý nghĩa nếu không có thành phần này.

Cơ sở hạ tầng truyền thông: là thành phần bắt buộc, nó là một tập gồm cả 2 mạng có dây và không dây và các bộ phận kết nối tuân tự để kết nối các thành phần đã liệt kê ở trên với nhau để chúng truyền với nhau hiệu quả.

c. Phương thức làm việc của RFID

Một hệ thống RFID có ba thành phần cơ bản: tag, đầu đọc, và một máy chủ. Mỗi tag được lập trình với một nhận dạng duy nhất cho phép theo dõi không dây đối tượng hoặc con người đang gắn tag đó. Bởi vì các chip được sử dụng trong tag RFID có thể giữ một số lượng lớn dữ liệu, chúng có thể chứa thông tin như chuỗi số, hướng dẫn cấu hình, dữ liệu kỹ thuật, sổ sách y học, và lịch trình. Cũng như phát sóng tivi hay radio, hệ thống RFID cũng sử dụng bốn băng thông tần số chính: tần số thấp (LF), tần số cao (HF), siêu cao tần (UHF) hoặc sóng cực ngắn (viba). Các hệ thống trong siêu thị ngày nay hoạt động ở băng thông UHF, trong khi các hệ thống RFID cũ sử dụng băng thông LF và HF. Băng thông vibra đang được để dành cho các ứng dụng trong tương lai.

Các tag có thể được cấp nguồn bởi một bộ pin thu nhỏ trong tag (các tag tích cực) hoặc bởi reader mà nó “wake up” (đánh thức) tag để yêu cầu trả lời khi tag đang trong phạm vi (tag thụ động).



Hình 2.28: Hoạt động giữa tag và reader RFID

Tag tích cực đọc xa 100 feet tính từ reader và có thể là tag RW (với bộ nhớ được viết lên và xóa như một ổ cứng máy tính) hoặc là tag RO. Tag thụ động có thể được đọc xa reader 20 feet và có bộ nhớ RO. Kích thước tag, giá cả, dài đọc, độ chính xác đọc/ghi, tốc độ dữ liệu và chức năng hệ thống thay đổi theo đặc điểm nêu ra trong thiết kế và dải tần hệ thống RFID sử dụng.

Reader gồm một anten liên lạc với tag và một đơn vị đo điện tử học đã được nối mạng với máy chủ. Đơn vị đo tiếp sóng giữa máy chủ và tất cả các tag trong phạm vi đọc của anten, cho phép một đầu đọc liên lạc đồng thời với hàng trăm tag. Nó cũng thực thi các chức năng bảo mật như mã hóa, giải mã và xác thực người dùng. Reader có thể phát hiện tag ngay cả khi không nhìn thấy chúng.

Máy chủ xử lý dữ liệu mà các reader thu thập từ các tag và dịch nó giữa mạng RFID và các hệ thống công nghệ thông tin lớn hơn, mà nơi đó quản lý dây chuyền hoặc cơ sở dữ liệu quản lý có thể thực thi. Middleware là phần mềm nối hệ thống RFID với một hệ thống IT quản lý luồng dữ liệu.

d. Các ứng dụng của RFID

RFID được ứng dụng trong các lĩnh vực:

- Bảo mật, an ninh:
 - Điều khiển truy nhập: khóa và các thiết bị cố định.

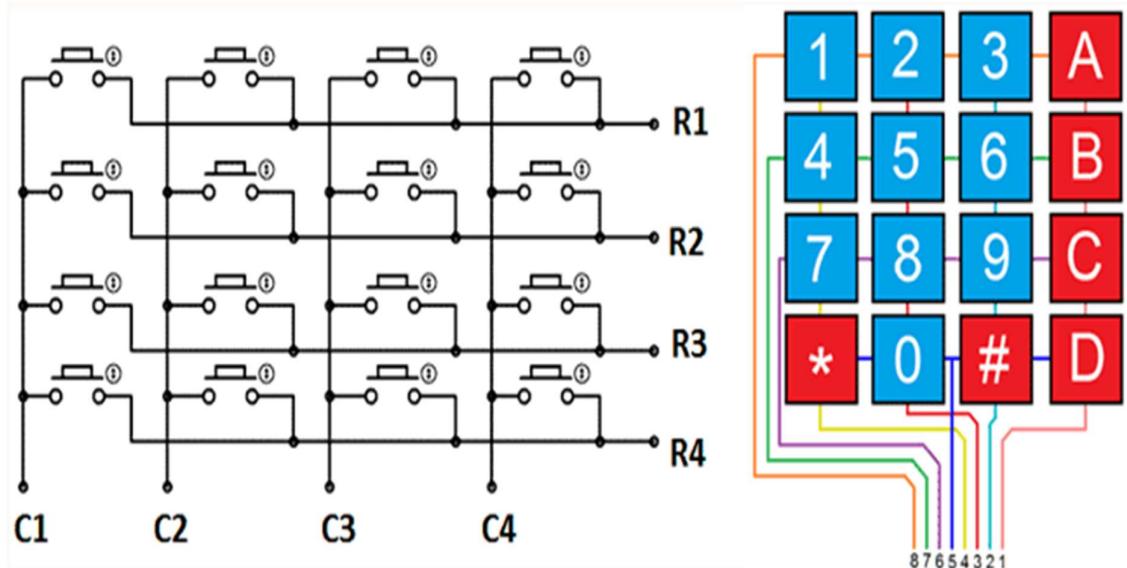
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Quy trình quản lý.
- Chống trộm: trong việc kinh doanh buôn bán.
- RFID trong việc xử phạt.
- Giám sát:
 - Dây truyền cung cấp: điều khiển kiểm soát trong các nhà kho.
 - Người hoặc súc vật: trẻ em, bệnh nhân, vận động viên, gia súc, thú kiêng.
 - Tài sản: hành lí trên máy bay, hàng hóa, thiết bị.
- Hệ thống thanh toán điện tử:
 - Lưu thông: hệ thống thu phí tự động.
 - Vé vào cổng.
 - Thẻ tín dụng.

2.2.8 Bàn phím ma trận

Bàn phím ma trận hay còn gọi là keypad là một bàn phím gồm nhiều nút nhấn, cho phép ta nhập các con số, chữ cái, ký tự vào bộ điều khiển. Số lượng các nút nhấn thay đổi phụ thuộc vào yêu cầu ứng dụng. Trong đề tài này dùng bàn phím ma trận để điều khiển khoá số điện tử nên ta sẽ tìm hiểu về bàn phím ma trận 4x4.

Bàn phím ma trận 4x4 gồm 16 nút nhấn được bố trí 4 hàng và 4 cột. Cách bố trí ma trận hàng và cột này tương tự như các ma trận LED, các nút nhấn cùng hàng và cùng cột được nối với nhau. Do đó bàn phím 4x4 sẽ có tổng cộng 8 ngõ ra (4 hàng và 4 cột).



Hình 2.29: Bàn phím ma trận 4x4

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Hoạt động của bàn phím ma trận : Giả sử phím “2” được nhấn, khi đó đường C2 và R1 được nối với nhau, nếu C2 nối với GND thì R1 cũng sẽ là GND. Vậy khi ta nối tắt cả các đường C1, C2, C3, C4 vào GND và nhấn phím “2” thì lúc này R1 cũng vẫn là GND nhưng ta sẽ không thể kết luận được phím “1”, phím “2”, phím “3” hay phím “A” được nhấn. Để khắc phục vấn đề này thì ta phải dùng phương pháp “quét” phím.

Phương pháp quét phím được trình bày theo thứ tự như sau:

- Đầu tiên các chân C1, C2, C3, C4 sẽ được thiết lập là các ngõ ra và giữ mức cao, còn các chân R1, R2, R3, R4 sẽ được thiết lập là ngõ vào và có điện trở kéo lên.
- Lần lượt kéo các chân C1, C2, C3, C4 xuống mức thấp.
- Đọc trạng thái các chân R1, R2, R3, R4 để kết luận phím nào được nhấn.

Ví dụ: Khi phím “2” được nhấn thì quá trình quét sẽ bao gồm 4 lần quét cho một chu kỳ đối với bàn phím 4x4.

Lần quét đầu: kéo chân C1 xuống mức thấp, các chân C2, C3, C4 vẫn ở mức cao. Kiểm tra 4 chân R1, R2, R3, R4 thu được kết quả R1=1, R2=1, R3=1, R4=1. Như vậy xét theo hàng dọc của C1 có các phím “1”, “4”, “7”, “*” không có phím nào được nhấn.

Lần quét thứ 2: kéo chân C2 xuống mức thấp, các chân C1, C3, C4 ở mức cao. Kiểm tra 4 chân R1, R2, R3, R4 thu được kết quả R1=0, R2=1, R3=1, R4=1. Ta thấy R1=0, xét trong hàng dọc của C2 thì tương ứng là vị trí của phím “2”. Như vậy ta kết luận được là phím “2” vừa được nhấn và có thể dừng quá trình quét tại đây.

2.2.9 Relay tiếp điểm cơ khí

Dùng điện áp 220VAC để cung cấp cho thiết bị công suất. Để cách ly tín hiệu điều khiển với phần thiết bị công suất có nhiều phương án thực hiện như relay tiếp điểm cơ khí, Solid State Relay, MOC + Triac...

Relay là một công tắc (khóa K) nhưng khác với công tắc thường là Relay được kích hoạt bằng điện. Trong mạch ta sử dụng relay với mục đích đóng ngắt các thiết bị điện theo yêu cầu.



Hình 2.30: Relay 5V

Thông số kỹ thuật:

- Điện áp điều khiển: 5V.
- Dòng tiêu thụ: 80mA.
- Dòng điện AC và điện áp AC cực đại: 10A → 250VAC.
- Thời gian tác động: 10ms.
- Thời gian nhả hẫm: 5ms.
- Nhiệt độ hoạt động: -45 → 75 °C.

2.2.10 Module hạ áp LM2596

Mạch giảm áp DC LM2596 4.5-50VDC 3A chịu đựng được điện áp đầu vào cao hơn so với dòng LM2596 thông thường. Mạch nhỏ gọn, dễ dàng sử dụng khi chỉ cần 2 chân ngõ vào và 2 chân ngõ ra. Chỉ cần cấp nguồn vào 2 chân input+, input- và nhận nguồn ra từ 2 chân output+, output- và có thể điều chỉnh điện áp đầu ra tùy biến qua biến trở tinh chỉnh.



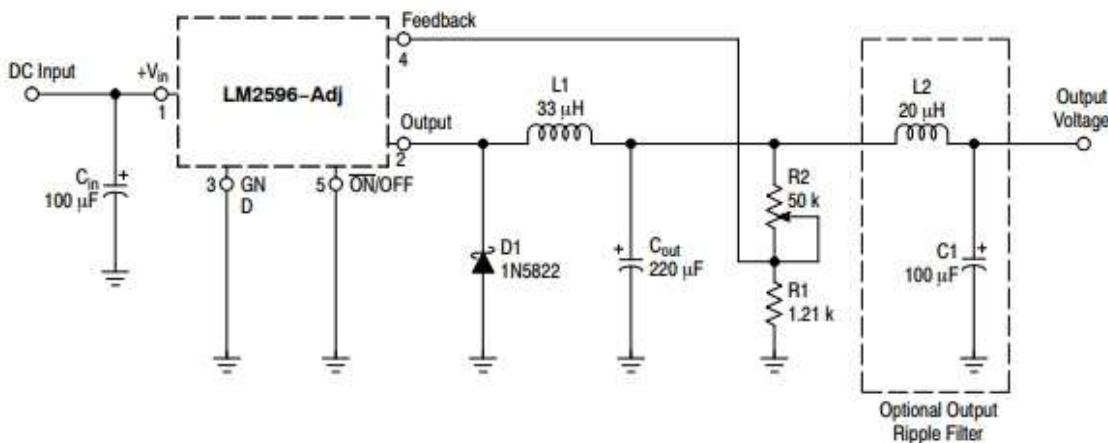
Hình 2.31: Module LM2596

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật:

- Điện áp đầu vào: 4.5 → 50V.
- Điện áp đầu ra: 3 → 35V.
- Dòng điện ngõ ra: max 3A.
- Công suất đầu ra: 15W.
- Hiệu suất đầu ra: 92%.
- Dòng tĩnh: 6mA.
- Kích thước: 43.6 x 21.5 x 13.5mm.

LM2596



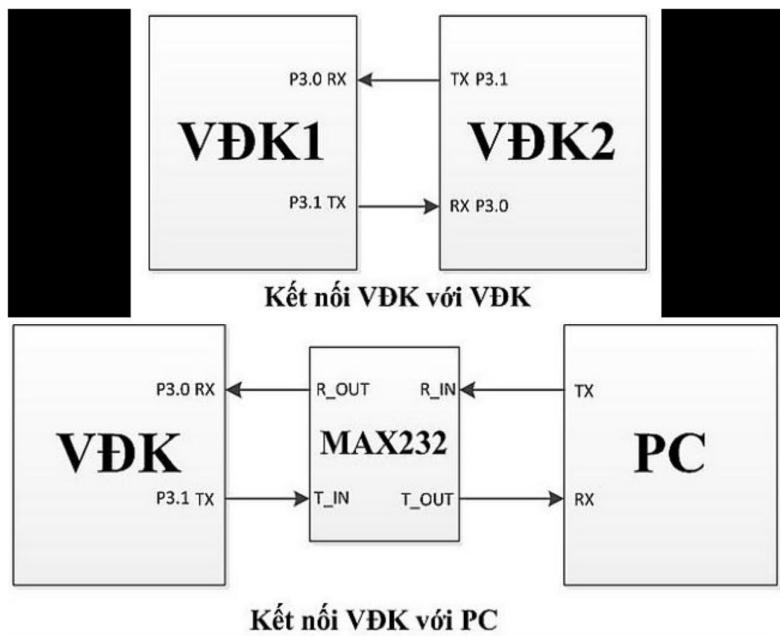
Hình 2.32: Sơ đồ nguyên lý module LM2596

2.3 CÁC CHUẨN TRUYỀN DỮ LIỆU

2.3.1 Chuẩn truyền thông UART

UART 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.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.33: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC

Một số thông số:

Baud rate (tốc độ Baud): Khi truyền nhận không đồng bộ để hai module hiểu được nhau thì cần quy định một khoảng thời gian cho 1 bit truyền nhận, nghĩa là trước khi truyền thì tốc độ phải được cài đặt đầu tiên. Theo định nghĩa thì tốc độ baud là số bit truyền trong một giây.

Frame (khung truyền): Do kiểu truyền thông nối tiếp này rất dễ mất dữ liệu nên ngoài tốc độ, khung truyền cũng được cài đặt từ ban đầu để tránh bớt sự mất mát dữ liệu này. Khung truyền quy định số bit trong mỗi lần truyền, các bit báo như start, stop, các bit kiểm tra như parity và số bit trong một data.

Bit Start: Là bit bắt đầu trong khung truyền bit này nhằm mục đích báo cho thiết bị nhận biết quá trình truyền bắt đầu trên AVR bit Start có trạng thái là 0.

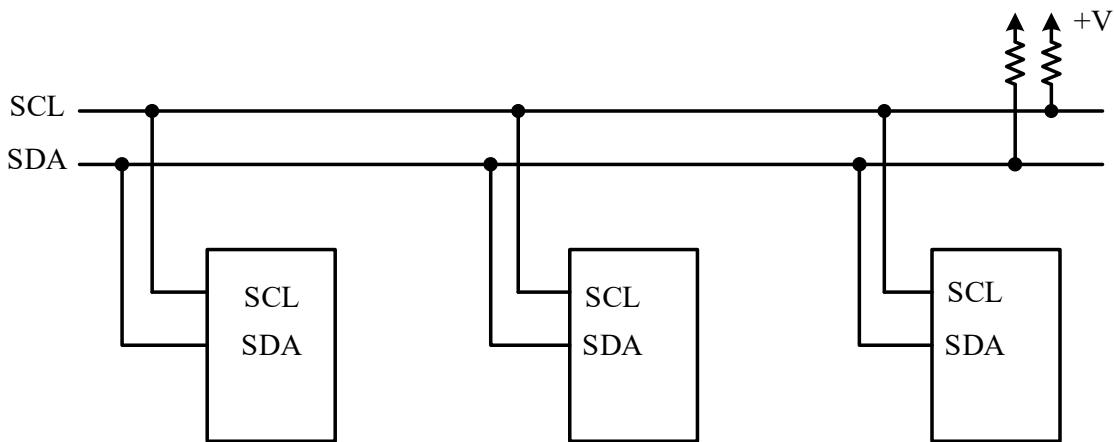
Data: Dữ liệu cần truyền Data không nhất thiết phải 8 bit có thể là 5, 6, 7, 8, 9. Trong UART bit LSB được truyền đi trước, Bit MSB được truyền đi sau.

Parity bit: Là bit kiểm tra dữ liệu đúng không, có 2 loại parity: chẵn (even parity), lẻ (odd parity). Parity chẵn là bit parity thêm vào để số số 1 trong data + parity = chẵn parity lẻ là bit parity thêm vào để số số 1 trong data + parity = lẻ. Bit Parity là không bắt buộc nên có thể dùng hoặc không.

Stop: là bit báo cáo kết thúc khung truyền, thường là mức 5V. Và có thể có 1 hoặc 2 stop.

2.3.2 Chuẩn giao tiếp I2C

I2C 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, AVR, ARM...



Hình 2.34: 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 vi kết nối vào đường bus I2C thì chân SDA của nó sẽ nối với dây SDA của bus, chân SCL sẽ nối với dây SCL.

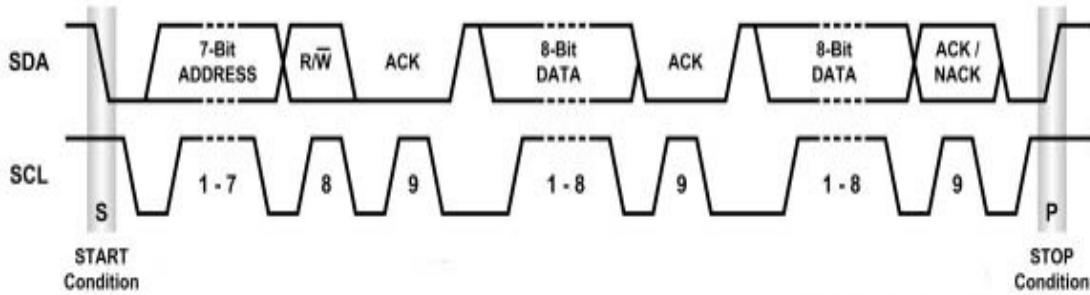
Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên (pullup resistor). Sự cần thiết của các điện trở này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực máng hở (opendrain hay opencollector). Giá trị của các điện trở này khác nhau tùy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

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.

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

Trình tự truyền bit trên đường truyền:



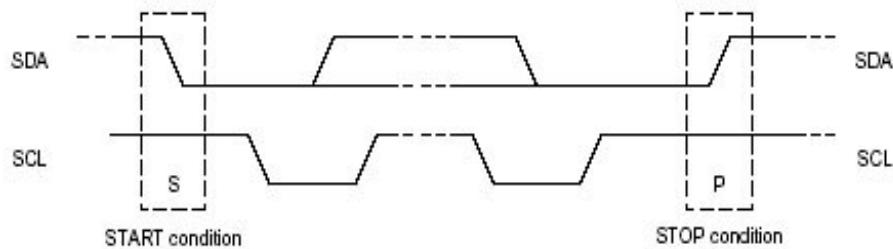
Hình 2.35: Trình tự truyền bit trên đường truyền

- Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tớ lắng nghe dữ liệu trên đường truyền.
- Thiết bị chủ gửi địa chỉ của thiết bị tớ 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 tớ đến thiết bị chủ, nếu cờ thiết lập xuống 0 thì byte tiếp theo được truyền từ thiết bị chủ đến thiết bị tớ).
- 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 8 bit 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.

Điều kiện START và STOP (START and STOP conditions)

START và STOP là những điều kiện bắt buộc phải có khi một thiết bị chủ muốn thiết lập giao tiếp với một thiết bị nào đó trên bus I2C. START là điều kiện khởi đầu, báo hiệu bắt đầu của giao tiếp, còn STOP báo hiệu kết thúc một giao tiếp. Hình dưới đây mô tả điều kiện START và STOP.

Ban đầu khi chưa thực hiện quá trình giao tiếp, cả hai đường SDA và SCL đều ở mức cao (**SDA = SCL = HIGH**). Lúc này bus I2C được coi là rỗng (“bus free”), sẵn sàng cho một giao tiếp. Hai điều kiện START và STOP là không thể thiếu trong việc giao tiếp giữa các thiết bị I2C với nhau.



Hình 2.36: Điều kiện start stop

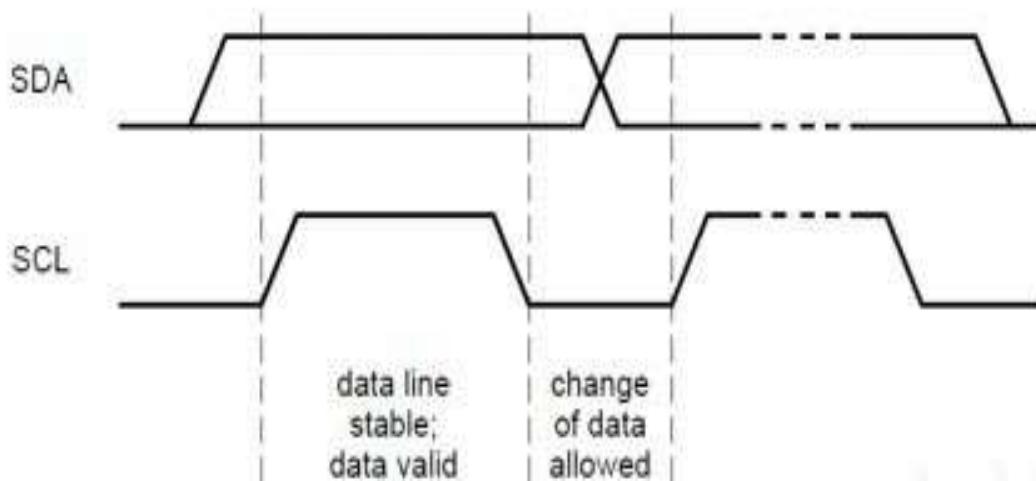
Điều kiện START: một sự chuyển đổi trạng thái từ cao xuống thấp trên đường SDA trong khi đường SCL đang ở mức cao (cao = 1; thấp = 0) báo hiệu một điều kiện START

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

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

Truyền dữ liệu:

Mỗi xung clock có một bit dữ liệu được truyền. Mức tín hiệu SDA chỉ được thay đổi khi xung clock đang ở mức thấp, và ổn định khi xung clock ở mức cao. Thiết bị nào có thể lấy mẫu dữ liệu khi xung clock ở mức cao.

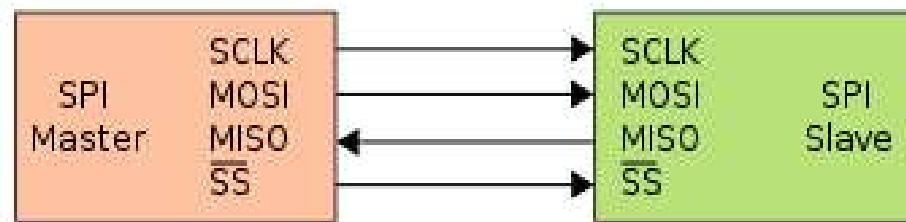


Hình 2.37: Truyền dữ liệu I2C

2.3.3 Chuẩn truyền thông SPI

SPI là phương pháp truyền song công (full duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI còn gọi là chuẩn truyền thông “4 dây” vì nó có 4 đường truyền giao tiếp, đó là SCK (Serial Clock), MISO (Master Input Slave Output), SS (Slave Select), MOSI (Master Output Slave Input). SPI là giao diện đồng bộ, bất cứ quá trình truyền nào cũng được đồng bộ hóa với tín hiệu clock chung và tín hiệu này sinh ra bởi master. Chức năng của 4 loại tín hiệu:

- 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 SCK báo 1 bit dữ liệu đến hoặc đi.
- SS – Slave Select: SS là đường chọn Slave cần giao tiếp, trên các Slave đường SS sẽ mức cao khi không làm việc. Nếu Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó.
- MISO - Master Input / Slave Output: nếu là Master thì đây là đường Input còn nếu là 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 Master thì đây là đường Output còn nếu là Slave thì MOSI lại là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.



Hình 2.38: Giao diện kết nối 4 dây chuẩn SPI

Chương 3: TÍNH TOÁN VÀ THIẾT KẾ

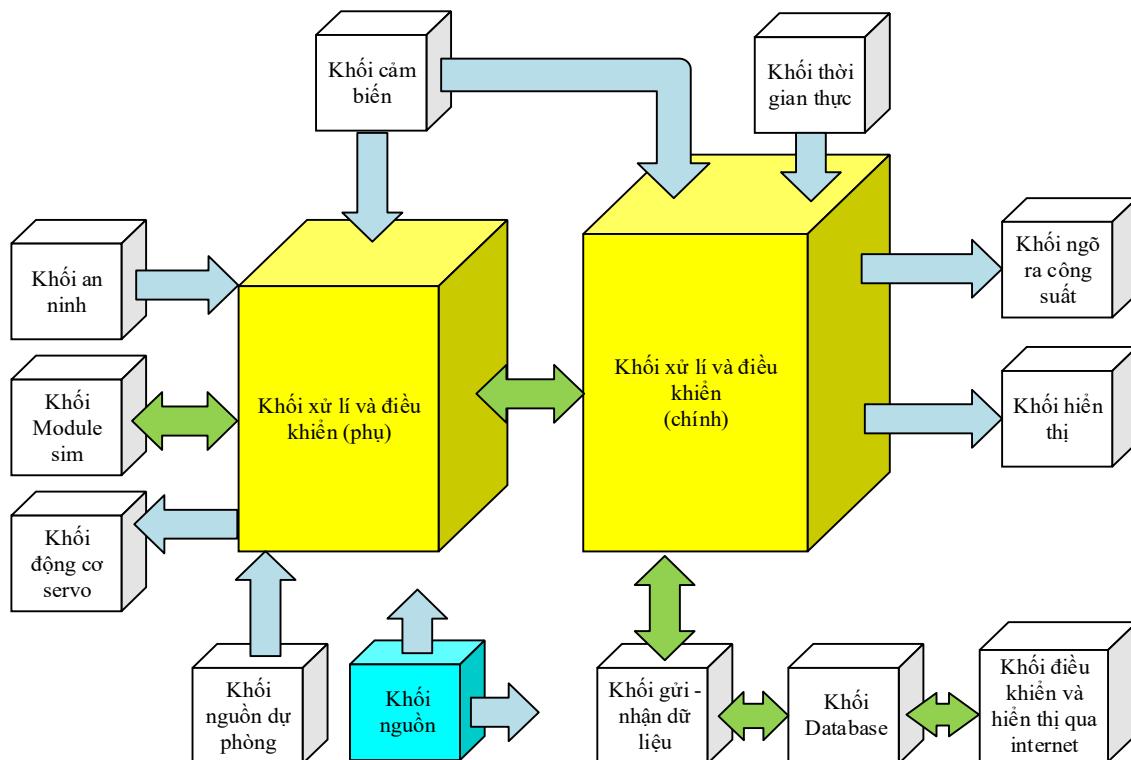
3.1 GIỚI THIỆU

Trong chương này, trình bày về cách tính toán, sơ đồ khói, sơ đồ nguyên lý của các board mạnh của hệ thống: mạch điều khiển các thiết bị ngõ ra có sử dụng relay, mạch nguồn cung cấp điện áp – dòng điện cho cả hệ thống.

3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2.1 Thiết kế sơ đồ khói hệ thống

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



Hình 3.1: Sơ đồ khói của hệ thống

Chức năng từng khói:

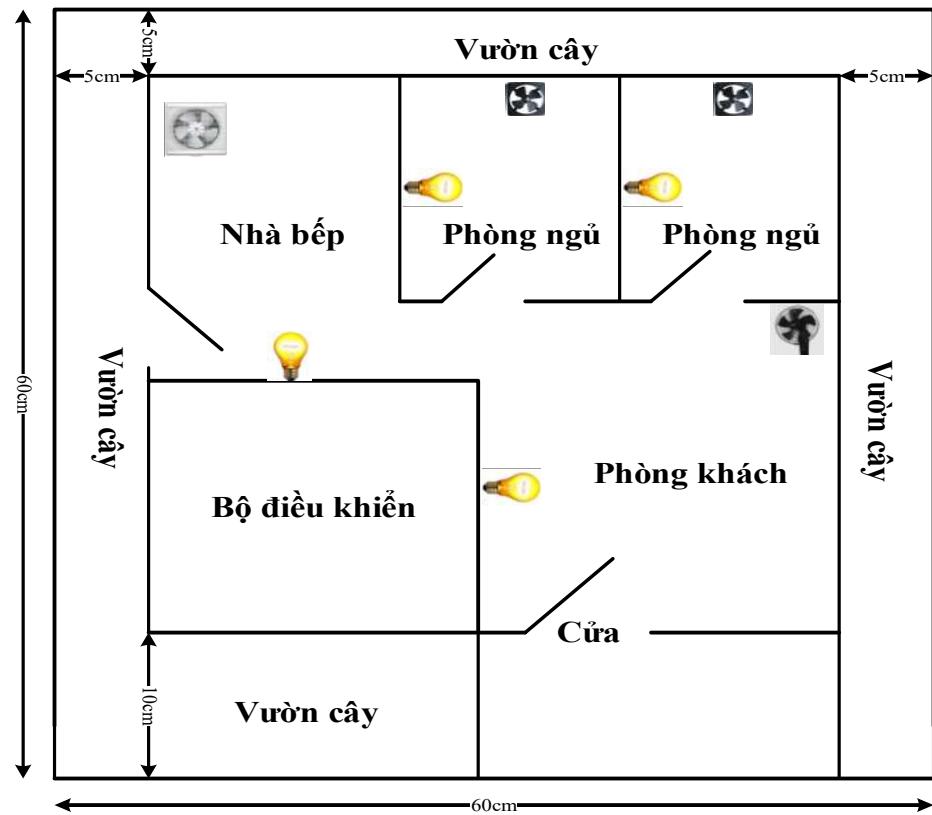
- ❖ **Khối xử lý và điều khiển (chính):** Tạo ra lệnh điều khiển, điều khiển hoạt động của hệ thống. Nhận tín hiệu từ cảm biến và từ khói xử lý và điều khiển (phụ) gửi về xử lý và gửi tín hiệu điều khiển sang khói khác.
- ❖ **Khối xử lý và điều khiển (phụ):** Điều khiển hoạt động của hệ thống như đóng mở cửa. Nhận tín hiệu từ cảm biến gửi về xử lý và gửi sang khói xử lý và điều khiển (chính).

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

- ❖ **Khối hiển thị:** Hiển thị thời gian, giá trị cảm biến, trạng thái thiết bị trên màn hình Oled.
- ❖ **Khối ngõ ra công suất:** Đóng ngắt các tiếp điểm theo sự điều khiển của ngõ ra STM32F407 để điều khiển thiết bị điện áp cao (220V). Cách ly giữa mạch công suất và mạch điều khiển.
- ❖ **Khối thời gian thực:** Cung cấp thời gian thực để hiển thị và đóng ngắt máy bơm nước tưới cây theo thời gian cài đặt.
- ❖ **Khối an ninh:** Đóng mở cửa khi người dùng quét thẻ, tạo sự tiện lợi cho người dùng khi không muốn nhập mật khẩu. Hiển thị các chức năng nhập mật khẩu mở cửa, đóng cửa hoặc đổi mật khẩu. Cảnh báo khi có sự cố rò rỉ khí gas hoặc có trộm.
- ❖ **Khối module sim:** Gửi tin nhắn hoặc gọi để điều khiển thiết bị, gửi tin nhắn và gọi chủ nhà khi có sự cố.
- ❖ **Khối gửi - nhận dữ liệu:** Gửi - nhận dữ liệu từ database.
- ❖ **Khối database:** Lưu trữ dữ liệu từ khối gửi - nhận.
- ❖ **Khối điều khiển và hiển thị qua internet:** Điều khiển thiết bị và hiển thị các thông số như trạng thái thiết bị, giá trị các cảm biến.
- ❖ **Khối cảm biến:** Đo giá trị cảm biến gửi về STM32F407 và Arduino xử lý.
- ❖ **Khối động cơ servo:** Kéo cửa khi người dùng nhập đúng mật khẩu hoặc nhấn nút đóng mở cửa.
- ❖ **Khối nguồn dự phòng:** Cung cấp nguồn tạm thời cho hệ thống an ninh có thể hoạt động thêm được một thời gian khi sự cố mất điện xảy ra.
- ❖ **Khối nguồn:** Cung cấp nguồn cho toàn mạch, trong đề tài này sử dụng 4 nguồn: 3.3VDC cấp cho cảm biến DS18B20, màn hình oled, 5VDC cấp mạch relay, module thời gian thực ds1307, ESP8266, 9VDC cấp cho Arduino mega, 12VDC cấp cho quạt và bơm nước, 220VAC cấp cho bóng đèn.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Để thiết kế ra một mô hình đáp ứng đầy đủ yêu cầu về công dụng, tính thẩm mỹ nhóm thực hiện đã hình thành ý tưởng và phác thảo ra mô hình sau đây:



Hình 3.2: Thiết kế mô hình căn hộ

3.2.2 Tính toán và thiết kế

a. Thiết kế khói cảm biến

❖ Khối cảm biến (STM32F407) đo các giá trị nhiệt độ.

• Điện áp hoạt động cảm biến:

- Cảm biến DS18B20: 3.3V

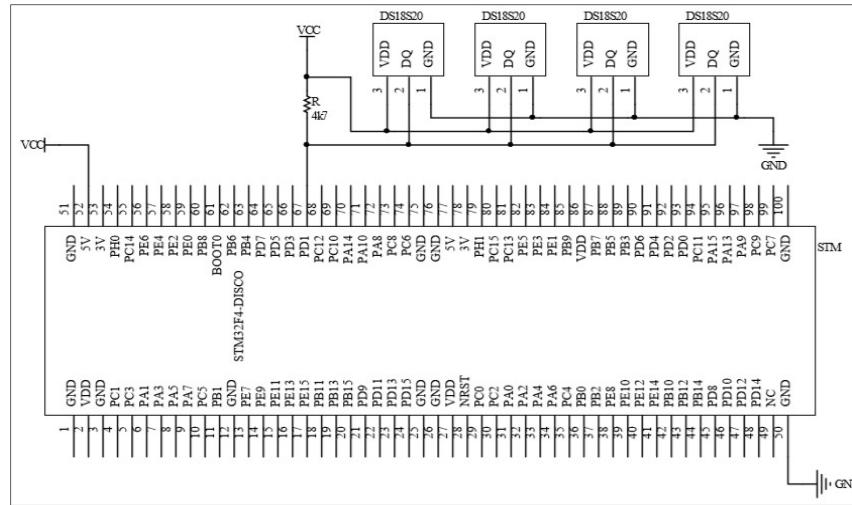
• Dòng tiêu thụ:

- Cảm biến DS18B20: 1.5mA

Từ thông số trên ta có:

$$I_{cb} = 4 \times I_{DS18B20} = 4 \times 1.5 = 6mA.$$

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



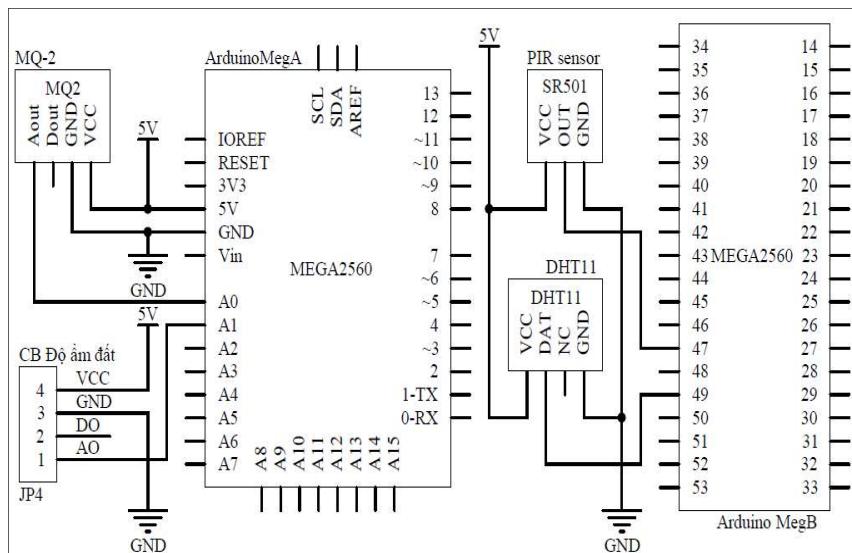
Hình 3.3: Sơ đồ nguyên lý kết nối cảm biến với STM32F407

- ❖ Khối cảm biến (Arduino) đo đạc các giá trị như: cảm biến báo trộm, nồng độ khí gas, nhiệt độ, độ ẩm không khí và độ ẩm đất vườn cây.

- Điện áp hoạt động của các cảm biến: 5V.
 - Dòng tiêu thụ:
 - Cảm biến DHT1: $I_{DHT} = 2.5\text{mA}$
 - Cảm biến khí gas: MQ-02: $I_{\text{gas}} = 180\text{mA}$
 - Cảm biến PIR: $I_{\text{PIR}} = 65\text{mA}$
 - Cảm biến độ ẩm đất: $I_{\text{đất}} = 35\text{mA}$

Từ các thông số trên ta có:

$$I_{CB} = I_{DHT} + I_{gas} + I_{PIR} + I_{đất} = 2.5 + 180 + 65 + 35 = 282.5 \text{ mA.}$$



Hình 3.4: Sơ đồ kết nối cảm biến với Arduino mega

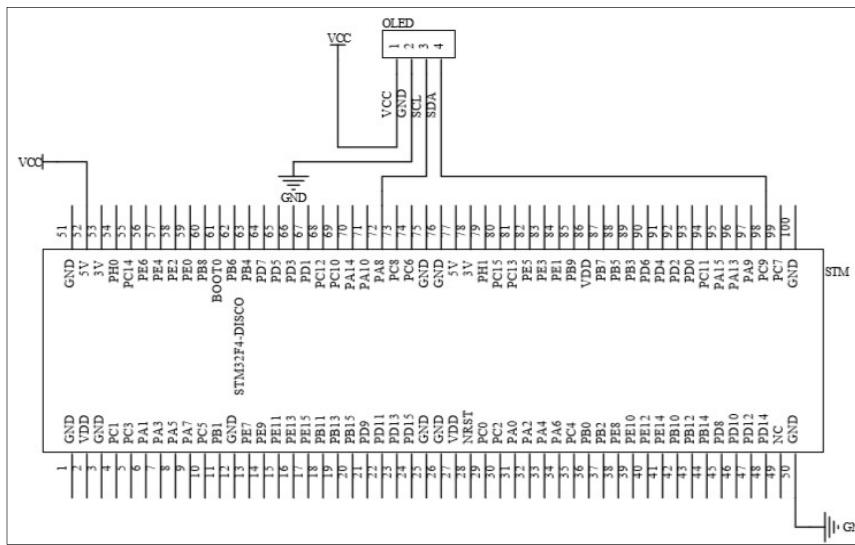
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

b. Thiết kế khối hiển thị

Oled được kết nối với STM32F407 theo chuẩn giao tiếp I2C với 2 chân Serial Data (SDA) và Serial Clock (SCL).

I2C được sử dụng là I2C3 của vi điều khiển. Chân SCL của Oled được kết nối với chân SCL (PA8) và chân SDA của oled được kết nối với chân SDA (PC9) của vi điều khiển.

Oled hoạt động với nguồn điện áp 3.3V, dòng hoạt động 12mA.

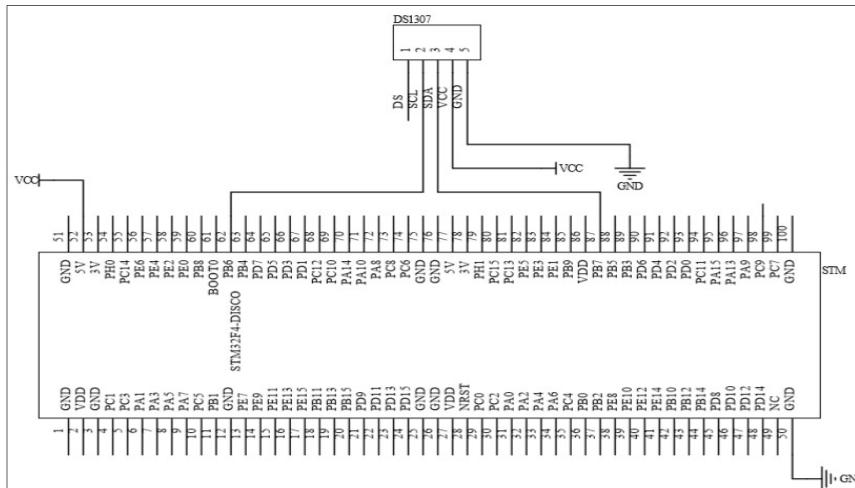


Hình 3.5: Sơ đồ nguyên lý kết nối Oled với STM32F407

c. Thiết kế khối thời gian thực

Cũng như màn hình Oled, module thời gian thực DS1307 cũng giao tiếp I2C với STM32F407 và I2C1 được sử dụng với chân SCL (PB6) và SDA (PB7).

DS1307 hoạt động với nguồn điện áp 5V.



Hình 3.6: Sơ đồ nguyên lý kết nối DS1307 với STM32F407

d. Thiết kế khôi ngõ ra công suất

Chọn relay 5 Vdc, loại này chỉ cần đặt áp 5 Vdc vào 2 đầu cuộn dây và dòng trên cuộn dây đảm bảo trên 70mA là các tiếp điểm hoạt động. Mặt khác, 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.

Sử dụng Transistor đóng ngắt loại NPN – D468, vì loại này cò dòng lên tới 1A để đóng ngắt cho relay.

Gắn thêm 1 Diode giữa 2 đầu cuộn dây mỗi Relay, để chống quá áp ngược từ cuộn dây, tránh hỏng các transistor. Chọn diode loại 1N4007, loại này chịu được áp ngược tối đa 700V.

Để đảm bảo an toàn cho mạch điều khiển thì nhóm sử dụng thêm Opto PC817 (hay còn gọi là cách ly quang). Nó là linh kiện tích hợp có cấu tạo gồm 1 Led và 1 photodiot hay 1 phototransistor. Được sử dụng để cách ly giữa khôi điều khiển với khôi ngõ ra công suất.

Led xanh lá cần dòng 10mA và áp roi là 1.9 – 4V. Từ đó ta suy ra giá trị:

$$R_{ledmax} = \frac{V_{cc} - V_{led} - 0.2}{I_{led}} = \frac{5 - 1.9 - 0.2}{10} = 290 \Omega$$

$$R_{ledmin} = \frac{V_{cc} - V_{led} - 0.2}{I_{led}} = \frac{5 - 4 - 0.2}{10} = 80 \Omega$$

Chọn $R_{led} = 220 \Omega$.

Dòng tối thiểu qua cuộn dây:

$$I_{c.dây} = \frac{U_2 \text{ đầu cuộn dây}}{\text{Nội trở cuộn dây}} = \frac{5}{70} \simeq 70 \text{ mA}$$

$$I_c = I_{c.dây} = 70 \text{ mA}$$

$$\text{Mà } I_c = \beta \times I_B$$

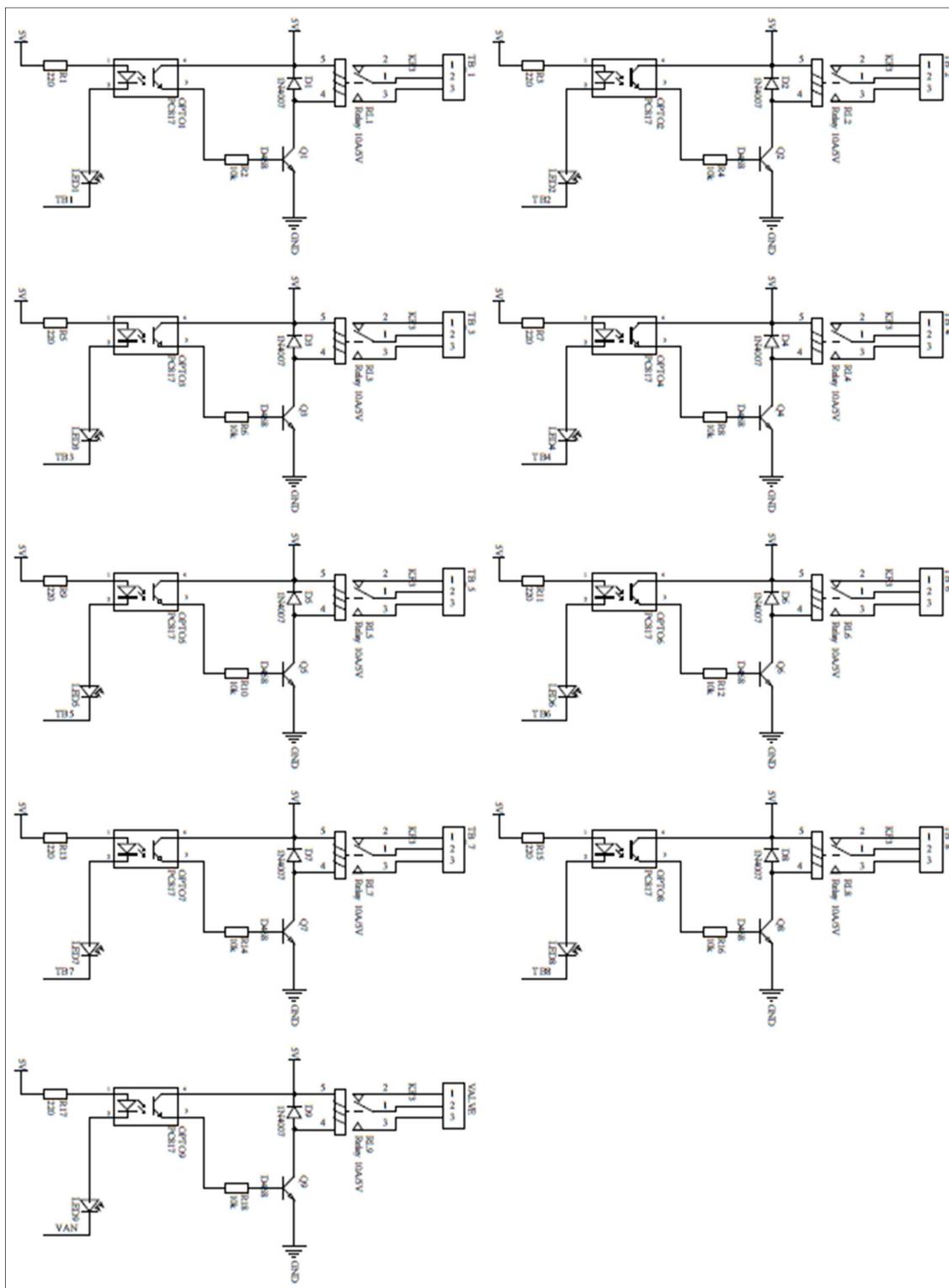
Transistor D468 có hệ số β thuộc khoảng từ 85 – 240, chọn hệ số $\beta = 100$.

$$I_B = \frac{70}{100} = 0.7 \text{ mA}$$

$$R_B = \frac{5 - 0.2}{0.7} = 6.9 \text{ k}\Omega$$

Chọn $R = 10 \text{ k}\Omega$.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



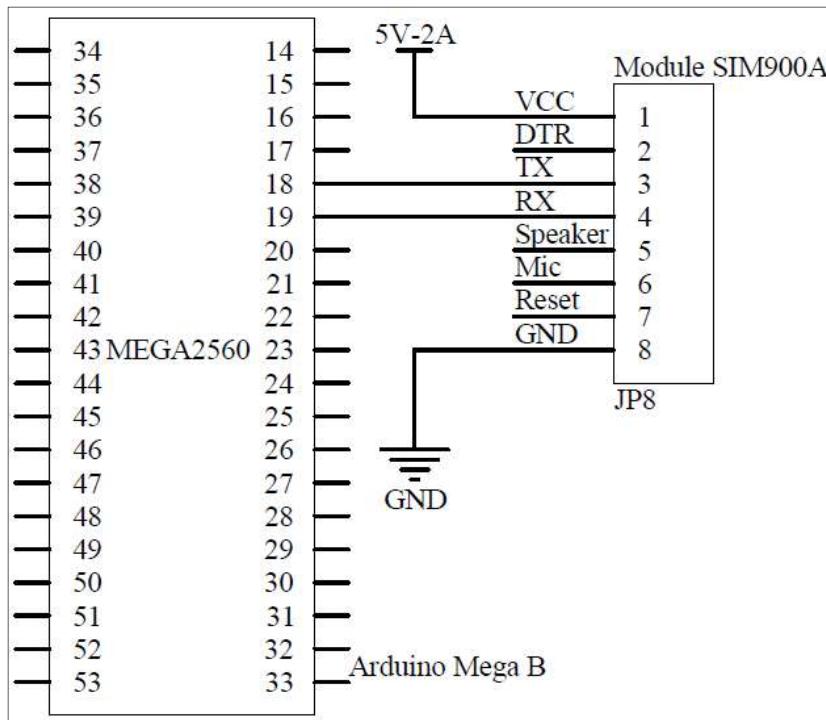
Hình 3.7: Sơ đồ nguyên lý kết nối công suất với STM32F407

e. Thiết kế khôi module sim

Khôi module SIM gồm một module SIM được kết nối vào cổng UART1 của Arduino Mega. Điện áp hoạt động của module SIM là 4.5V nhưng dòng khởi động của

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

nó là 2A, lớn hơn dòng ra tối đa của Arduino nên ta phải dùng nguồn ngoài để module có thể hoạt động tốt.



Hình 3.8: Sơ đồ nguyên lý kết nối module SIM với Arduino

f. Thiết kế khối an ninh

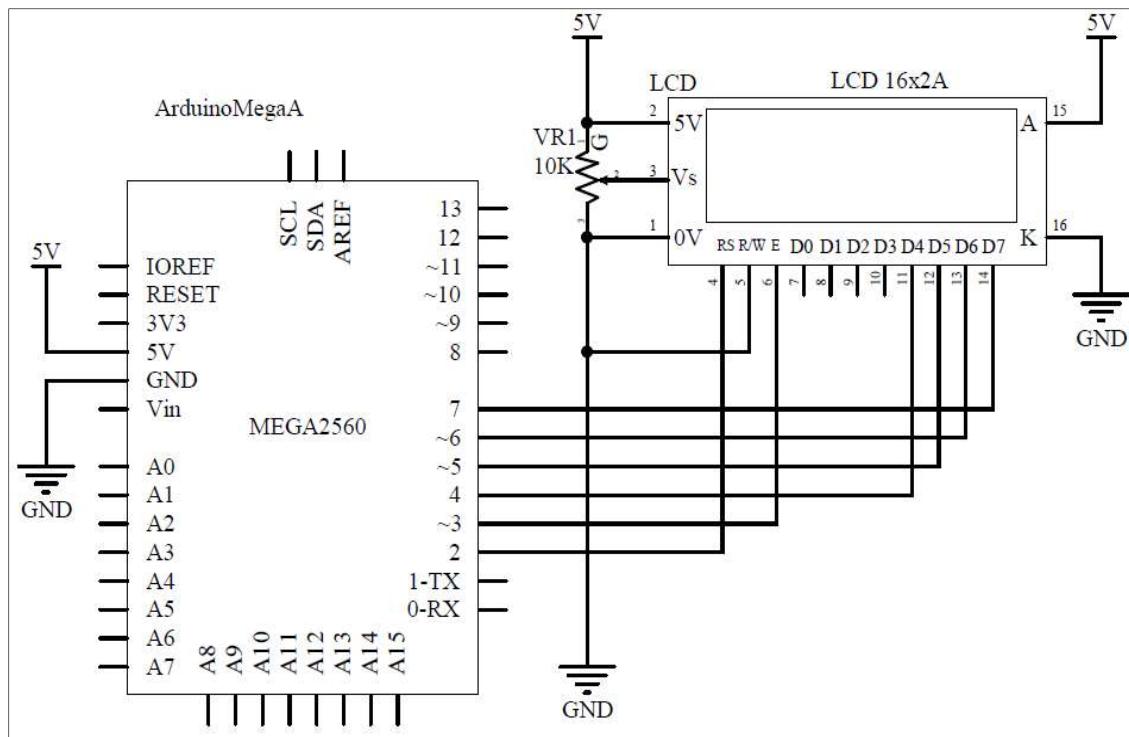
Khối an ninh bao gồm một LCD 16x2 dùng để hiển thị và bàn phím (keypad) 4x4 để người dùng chọn các chức năng hoặc nhập mật khẩu và đầu đọc thẻ RC522.

LCD 16x2 được điều khiển theo kiểu 4 bit, tức là chỉ dùng 4 chân data từ D4 đến D7 để truyền dữ liệu và 2 dây điều khiển từ vi điều khiển được kết nối với chân E và RS của LCD. Để điều chỉnh độ tương phản của LCD thì ta kết nối một biến trở 10k Ohm vào chân V0 của LCD. LCD 16x2 hoạt động tốt ở điện áp 5V và dòng tiêu thụ là 160mA nên có thể dùng nguồn từ Arduino.

Đầu đọc thẻ RFID RC522 hoạt động ở mức điện áp 3.3V, dòng hoạt động 100mA. Hỗ trợ chuẩn giao tiếp SPI với tốc độ 10Mbit/s.

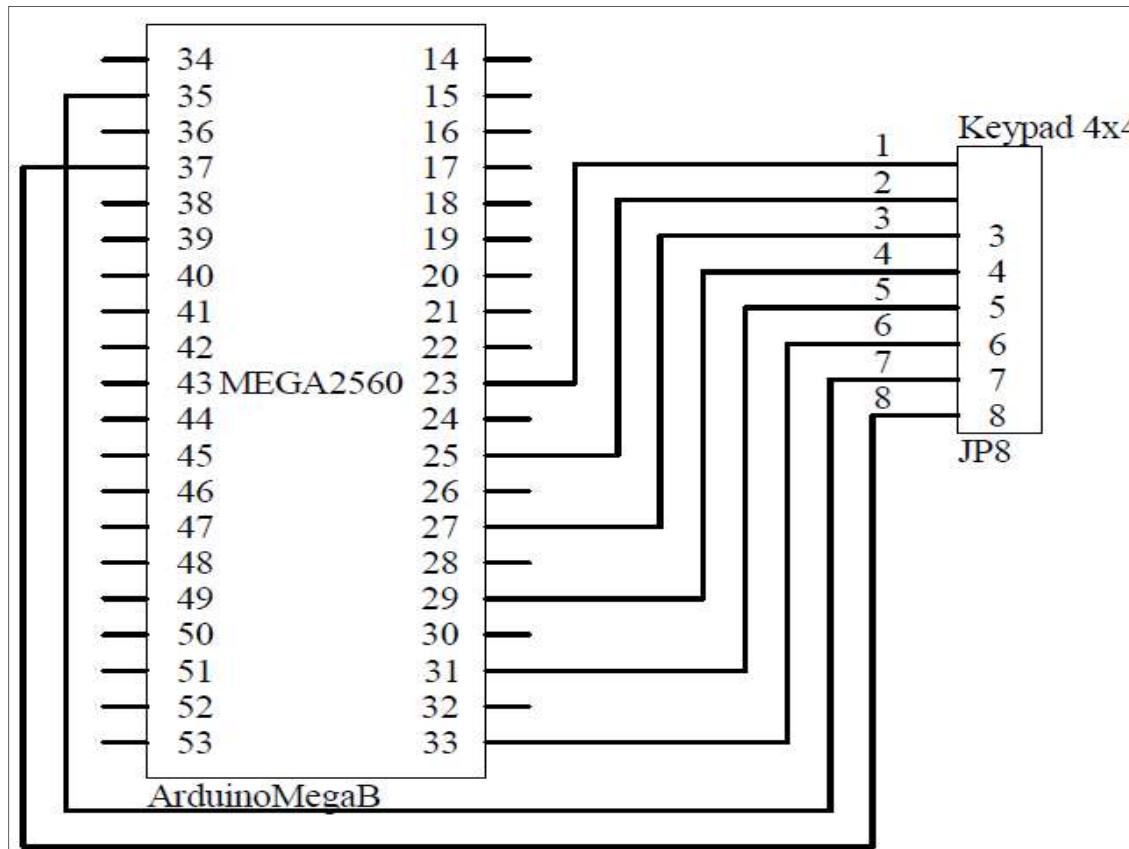
Một còi báo được kết nối bằng cách nối chân dương của còi vào chân pin 13 của Arduino Mega, chân âm của còi nối vào GND. Khi xảy ra sự cố, Arduino sẽ phát một xung PWM vào cực dương của còi, làm còi kêu lên báo động.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



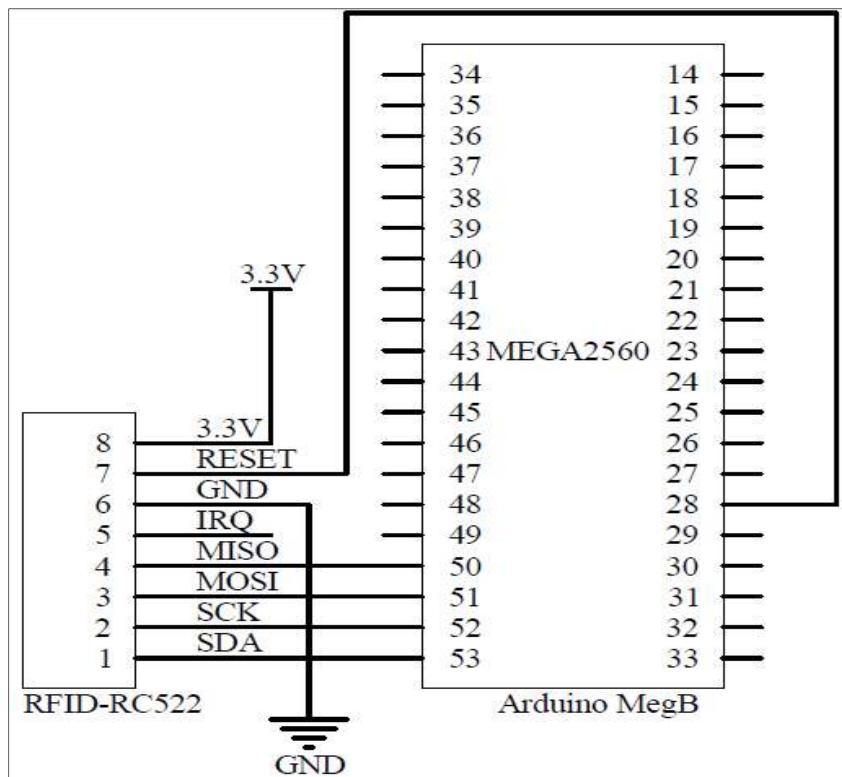
Hình 3.9: Sơ đồ nguyên lý kết nối LCD với Arduino

Bàn phím 4x4 gồm 8 chân sẽ được kết nối trực tiếp vào Arduino.

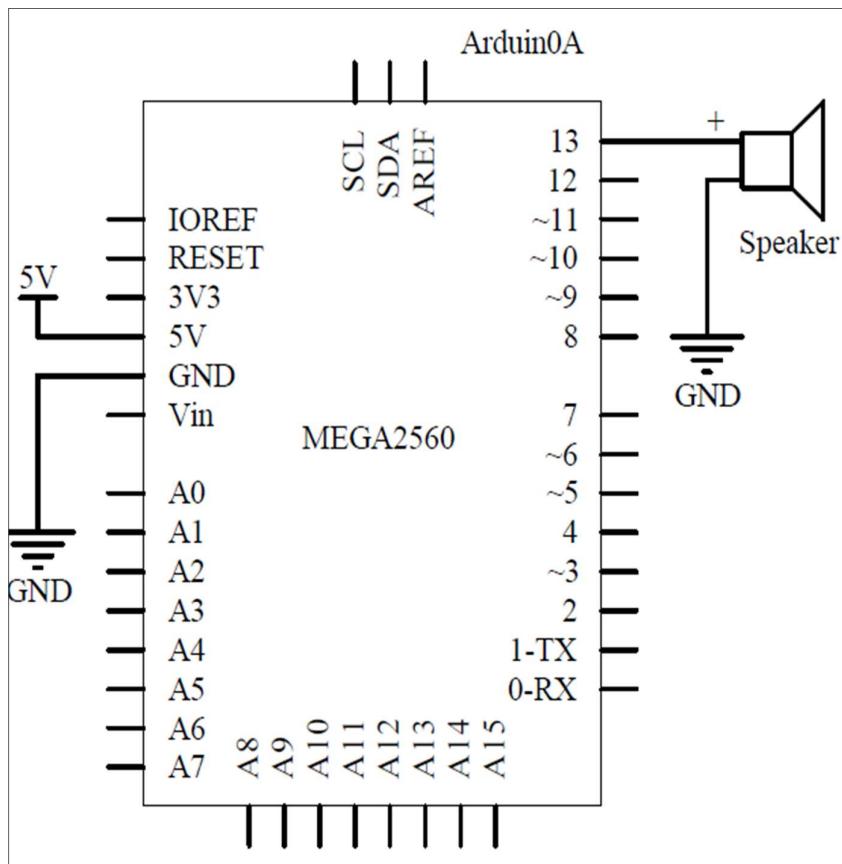


Hình 3.10: Sơ đồ nguyên lý kết nối bàn phím 4x4 với Arduino

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



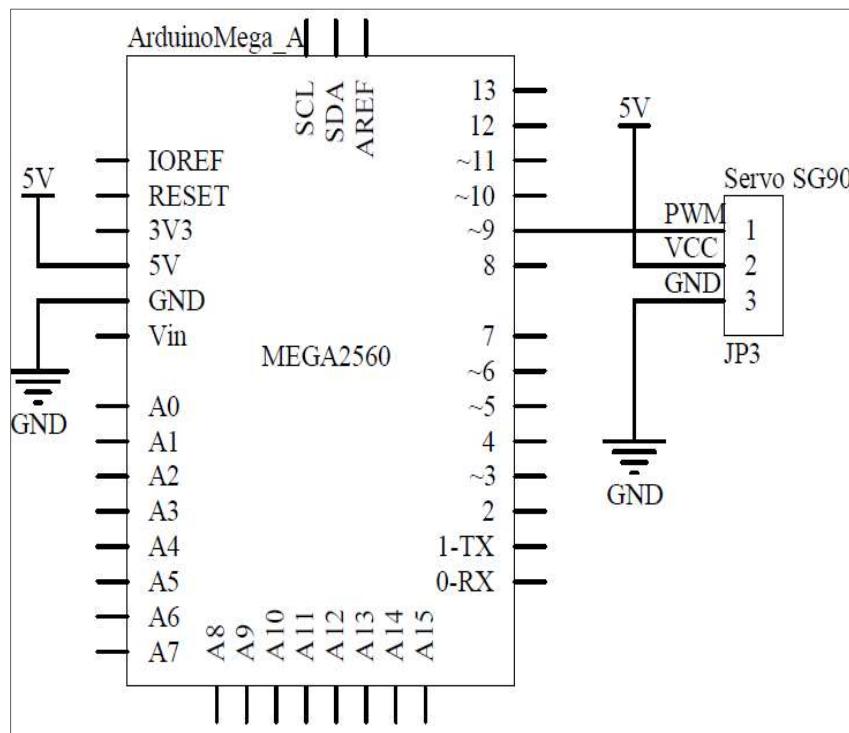
Hình 3.11: Sơ đồ nguyên lý kết nối đầu đọc RC522 với Arduino



Hình 3.12: Sơ đồ nguyên lý kết nối còi báo với Arduino

g. Thiết kế khối động cơ servo

Khối động cơ bao gồm một động cơ servo SG90 hoạt động với điện áp 5V.

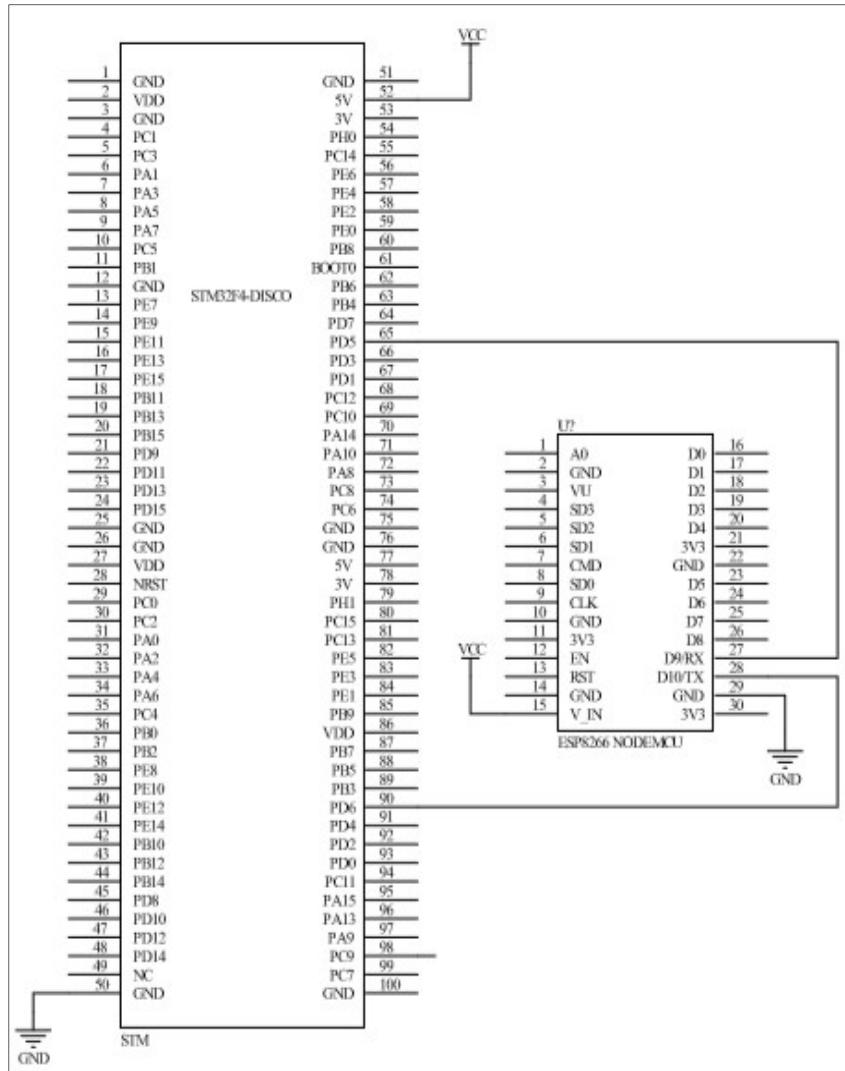


Hình 3.13: Sơ đồ kết nối khối động cơ với Arduino

h. Thiết kế khối gửi – nhận dữ liệu

Khối gửi – nhận dữ liệu là module ESP8266 NodeMCU hỗ trợ chuẩn giao tiếp UART.

Điện áp hoạt động của ESP8266 là 3V3 nhưng có thể cấp 5V vào chân 5V của ESP và dòng hoạt động là 170 mA khi kết nối tới wifi.



Hình 3.14: Sơ đồ nguyên lý kết nối esp8266 với STM32F407

i. Thiết kế khối database

Chức năng khói này là lưu trữ giá trị trạng thái các thiết bị, giá trị nhiệt độ được gửi lên.

Firebase là một cơ sở dữ liệu miễn phí và có tính phí của Google. Firebase có thể lưu trữ dữ liệu do người dùng gửi lên theo nhiều dạng như văn bản, hình ảnh, danh bạ. Firebase và Android là của Google nên việc đồng bộ và nhúng sẽ trở nên dễ dàng hơn.

j. Thiết kế khói điều khiển và hiển thị qua internet

Khối điều khiển và hiển thị qua Internet là tổng hợp của các thiết bị có thể kết nối với Internet mà người dùng có thể sử dụng một cách dễ dàng. Ở trong đề tài này nhóm sử dụng điện thoại thông minh chạy hệ điều hành Android, máy tính để bàn hoặc máy tính xách tay.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

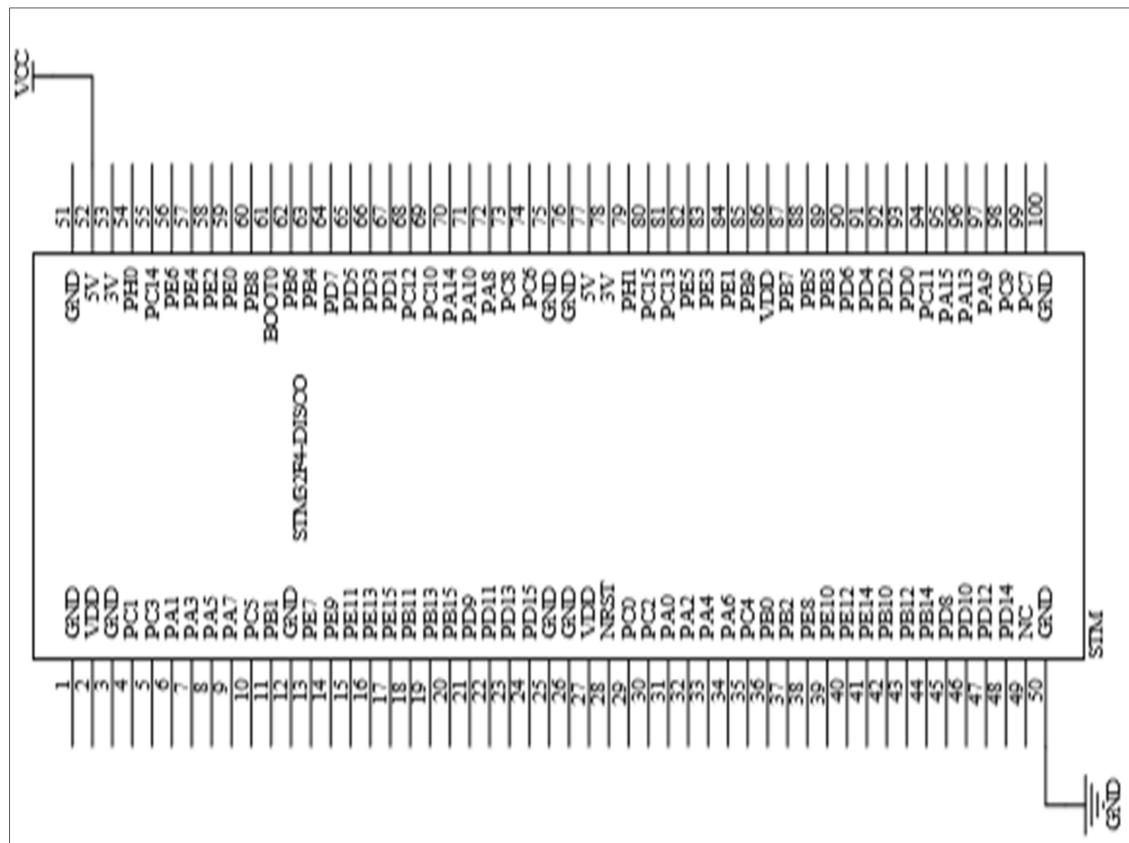
Khối này có chức năng điều khiển thiết bị qua mạng Internet bằng cách thay đổi dữ liệu trên khối database.



Hình 3.15: Khối điều khiển và hiển thị qua internet

k. Thiết kế khói xử lý và điều khiển (chính)

Dòng STM32F407xx được dựa trên lõi RISC 32-bit ARM®Cortex®-M4 hiệu suất cao hoạt động ở tần số lên đến 168 MHz. STM32F407 VGT6 có tất cả 100 chân với 4 USART và 2 UART, 3 giao diện I2C, 3SPI và 2 CAN giao diện. Nó chứa tất cả mọi thứ cần thiết để tạo thành khói xử lý và điều khiển với đầy đủ các port.

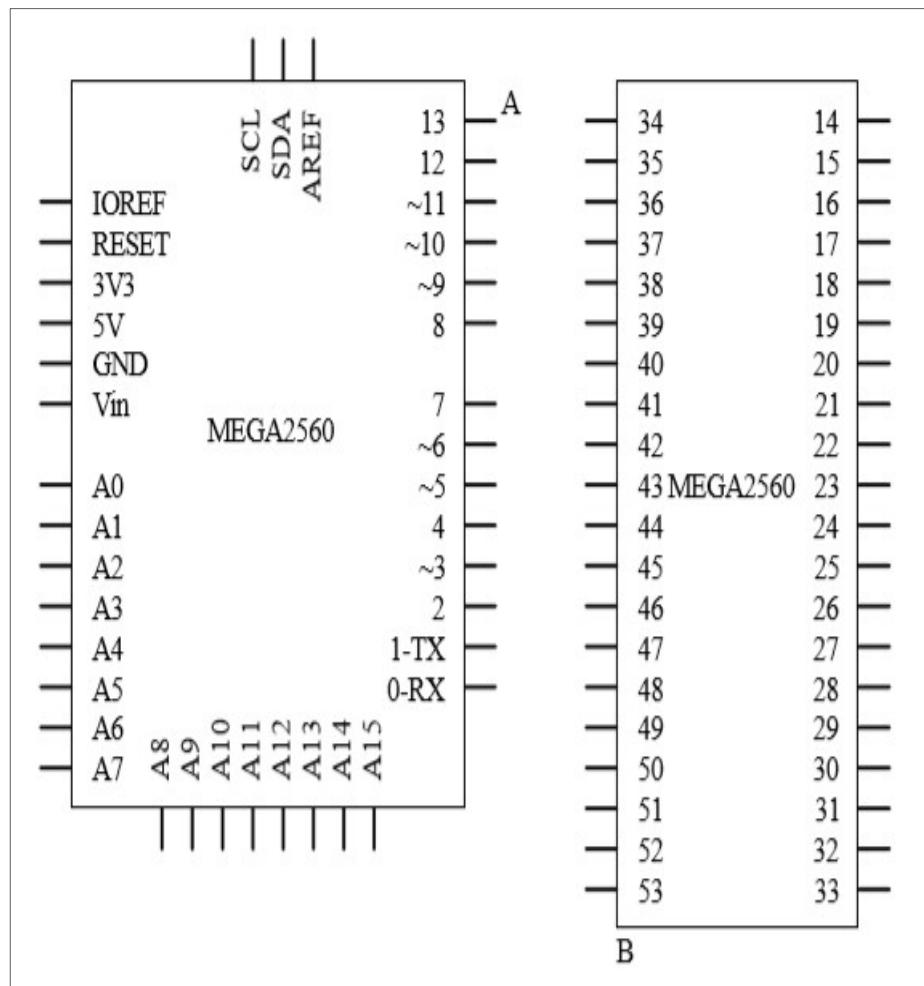


Hình 3.16: Khối xử lý và điều khiển (chính)

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

I. Thiết kế khối xử lý và điều khiển (phụ)

Vì Arduino Mega sử dụng chip ATmega2560. Nó có 54 chân digital I/O 16 chân đầu vào tương tự (Analog Inputs), 4 UARTs (cổng nối tiếp phần cứng), một thạch anh dao động 16 MHz, két nối USB, một jack cảm điện, một đầu ICSP và một nút reset. Nó chứa tất cả mọi thứ cần thiết để tạo thành khối xử lý và điều khiển với đầy đủ các port.



Hình 3.17: Khối xử lý và điều khiển (phụ)

m. Thiết kế khối nguồn dự phòng

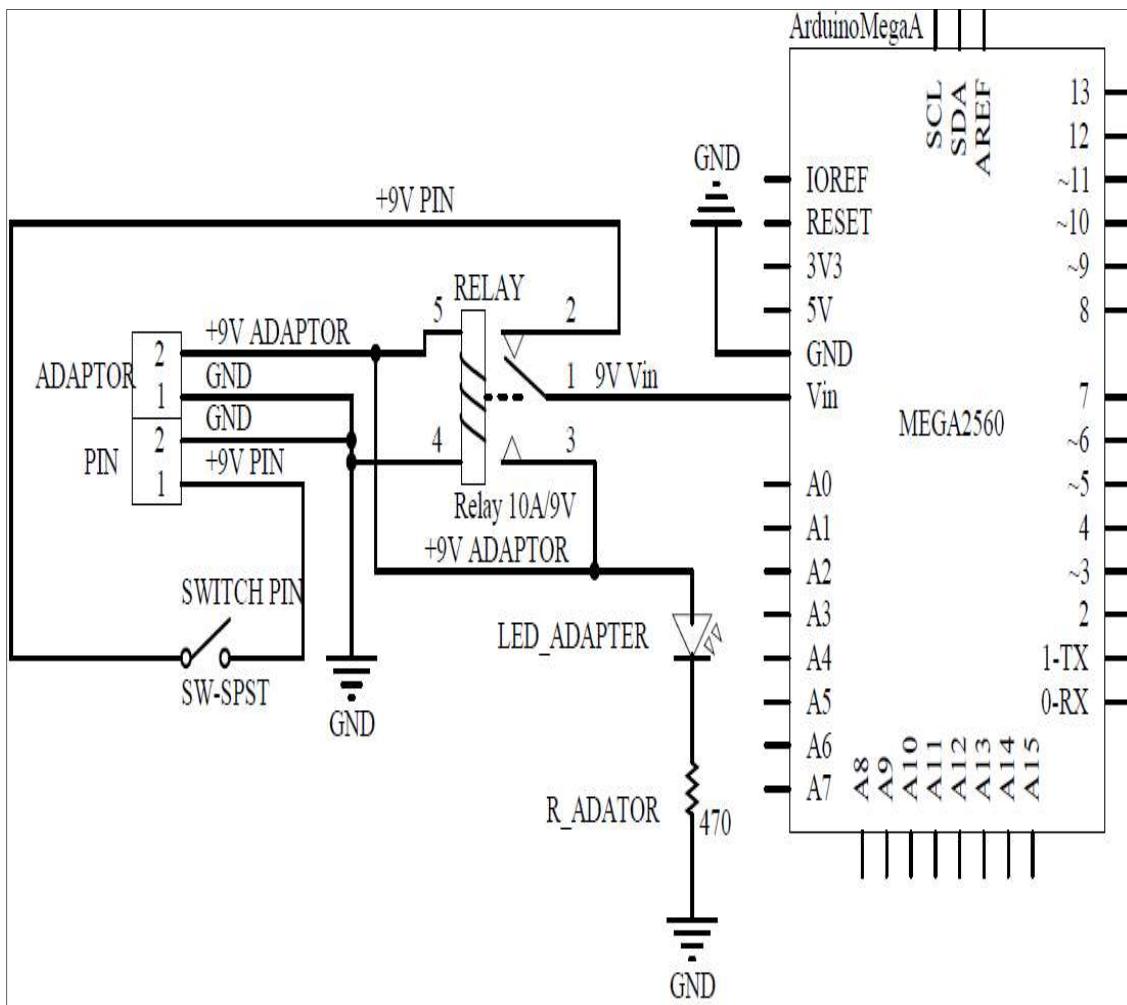
Khối nguồn dự phòng sẽ cấp nguồn cho vi điều khiển hoạt động trong một khoảng thời gian, tránh trường hợp không thể mở hoặc đóng cửa khi có sự cố mất điện. Ta cần thiết kế một cơ chế tự thay đổi nguồn điện chính và nguồn dự phòng bằng cách sử dụng một relay để chọn nguồn. Hai đầu cuộn dây relay sẽ nối với 2 cực của nguồn điện chính, tiếp điểm thường đóng của relay sẽ nối với cực dương của nguồn dự phòng, tiếp điểm thường hở nối với cực dương của nguồn điện chính và đầu còn lại của relay sẽ nối với ngõ vào điện áp của Arduino. Cực âm của 2 nguồn và mạch đều được nối chung.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Mô tả hoạt động:

Bình thường khi mạch hoạt động thì có dòng điện chạy qua cuộn dây relay tạo ra một từ trường hút lõi sắt non làm thay đổi công tắc chuyển mạch, tiếp điểm thường hở sẽ nối với ngõ vào điện áp của Arduino. Như vậy là mạch đang dùng nguồn điện chính để hoạt động.

Ngược lại khi nguồn điện chính không có hoặc gặp sự cố thì tiếp điểm thường đóng sẽ nối với ngõ vào điện áp của Arduino, mạch sử dụng nguồn điện dự phòng.



Hình 3.18: Sơ đồ nguyên lý khối nguồn dự phòng

Nguồn điện dự phòng được lấy từ pin sạc loại 18650 có điện áp 4.2V, dung lượng 3800mAh. Nếu chỉ dùng một viên pin thì mạch sẽ không đủ điện áp để hoạt động, do đó ta sẽ mắc nối tiếp 2 viên PIN vào để được một nguồn điện dự phòng có áp là 8.4V, dung lượng vẫn là 3800mAh.

Tính toán thời gian hoạt động của nguồn dự phòng trước khi hết:

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

$$T = I_{pin}/(I_{Arduino} + I_{SG90} + I_{CB} + I_{LCD} + I_{RFID}) = 3800/(200+60+282.5+160+100) = 4.7(\text{h})$$

Như vậy nguồn dự phòng có thể giúp mạch hoạt động thêm được gần 4.7 giờ sau khi mất nguồn điện chính.

n. Thiết kế khối nguồn

Để thiết kế được khối nguồn, nhóm tiến hành liệt kê các bảng tính toán dòng điện và các cấp điện áp sử dụng trong mạch.

Bảng 3.1: Dòng điện của các linh kiện sử dụng trong mạch (STM32F407)

STT	Tên linh kiện	Điện áp(V)	Dòng tiêu thụ(mA)	Số lượng	Tổng dòng điện (A)
1	STM32F407	5	160	1	0.16
2	ESP8266	5	170	1	0.17
3	DS1307	5	1.5	1	0.0015
4	Relay	5	80	9	0.72
5	Transistor D468	5	40	9	0.36
6	Opto	5	5	9	0.045
7	DS18B20	3.3	1.5	4	0.006
8	Oled	3.3	12	1	0.012
9	Led đơn	5	20	9	0.18
				Tổng	1.65A

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Bảng 3.2: Dòng các linh kiện sử dụng trong mạch (Arduino)

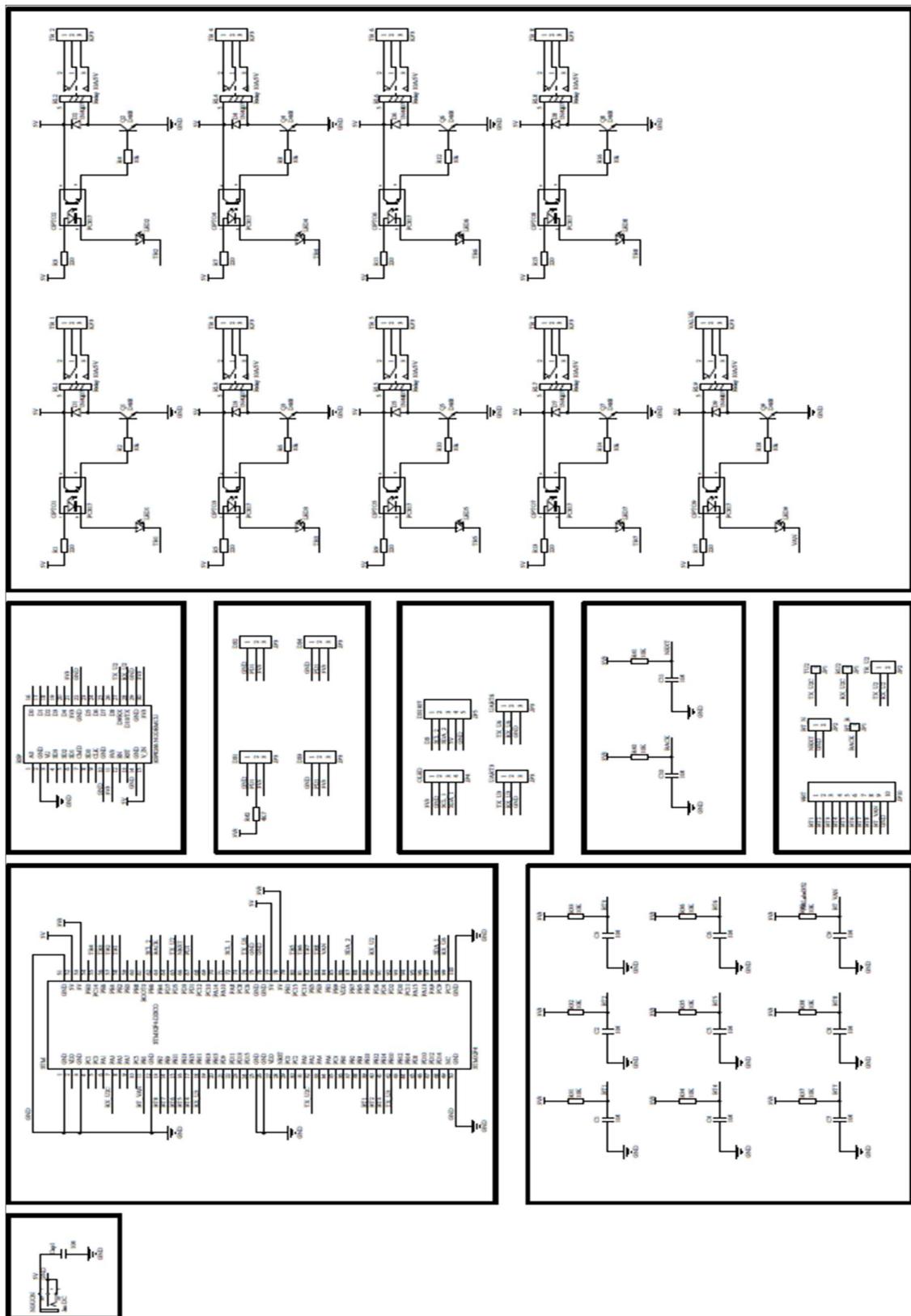
STT	Tên linh kiện	Điện áp (V)	Dòng tiêu thụ (mA)	Số lượng	Tổng dòng điện (A)
1	Arduino Mega	9	200	1	0.2
2	RFID RC522	3.3	100	1	0.1
3	Module SIM	5	500	1	0.5
4	Servo SG90	5	60	1	0.05
5	CB Gas	5	180	1	0.18
6	CB DHT11	5	2.5	1	0.0025
7	CB SR501 (PIR)	5	65	1	0.065
8	CB độ ẩm đất	5	35	1	0.035
9	LCD 16x2	5	160	1	0.16
10	Relay 5V	5	80	1	0.08
11	Led đơn	5	20	3	0.06
				Tổng	1.45A

Bảng 3.3: Dòng điện của các linh kiện sử dụng trong mạch 12V

STT	Tên linh kiện	Dòng tiêu thụ(mA)	Số lượng	Tổng dòng điện (A)
1	Quạt	100	4	0.4
2	Bơm nước	250	1	0.25
			Tổng	0.65A

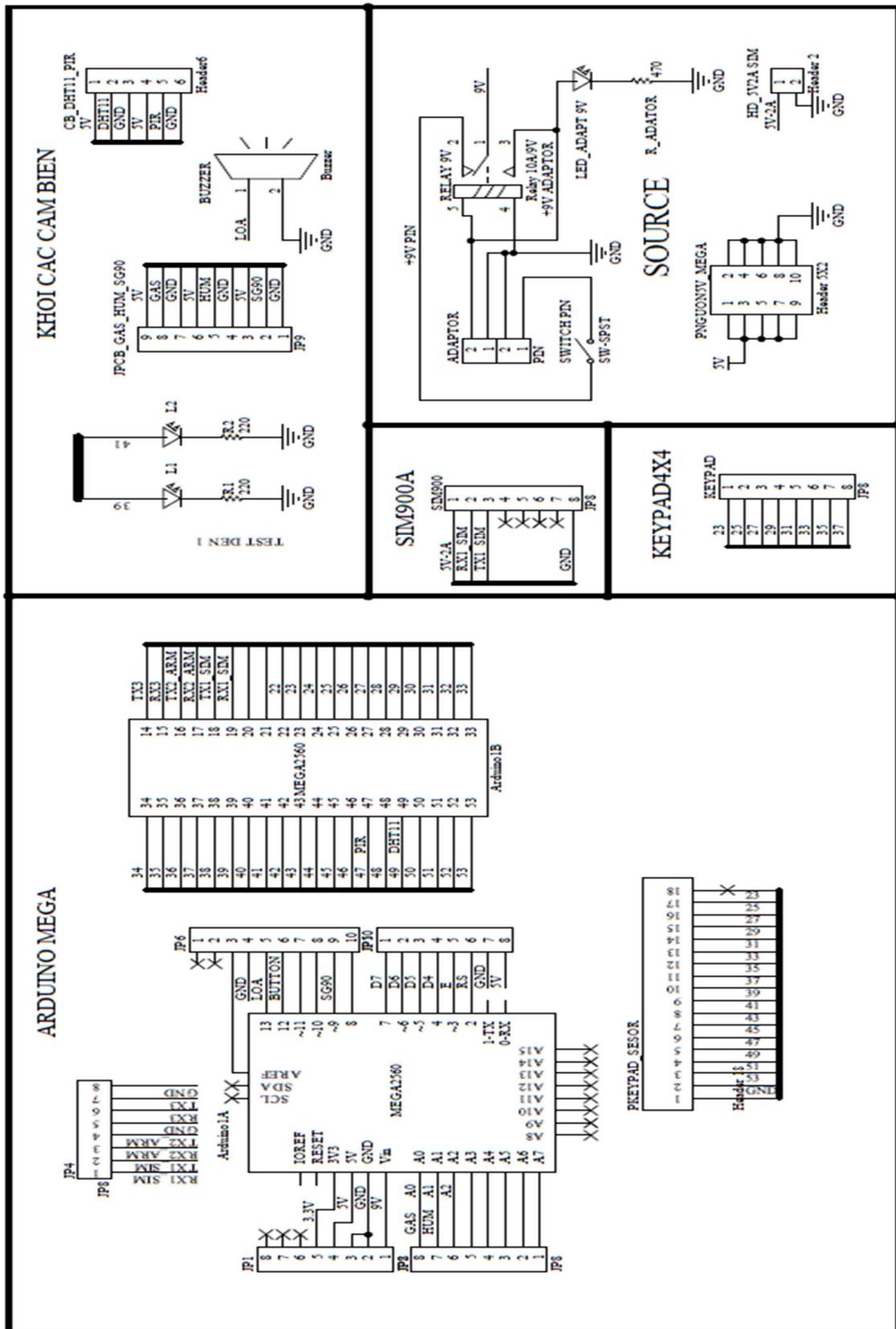
Từ các bảng trên, nhóm sử dụng khói nguồn để cung cấp cho toàn hệ thống bao gồm 1 nguồn tổ ong chuyển đổi điện áp từ 220VAC sang 12VDC, dòng ra tối đa 10A. Sử dụng module LM2596 để hạ áp từ 12V xuống 5V cấp cho mạch điều khiển và một module LM2596 hạ áp từ 12V xuống 9V để cấp cho Arduino.

3.2.3 Sơ đồ nguyên lý toàn mạch



Hình 3.19: Sơ đồ nguyên lý toàn mạch (chính)

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



Hình 3.20: Sơ đồ nguyên lý toàn mạch (phụ)

Chương 4: THI CÔNG HỆ THỐNG

4.1 GIỚI THIỆU

Trong chương này là quá trình thi công PCB, lập trình, lắp ráp phần cứng và test mạch. Bên cạnh đó là hình vẽ được chụp từ mô hình thực tế của hệ thống, hình chụp các kết quả chạy của hệ thống.

4.2 THI CÔNG HỆ THỐNG

4.2.1 Thi công board mạch

Chuẩn bị vật liệu: Một cây thước, một cây kéo, một dao cắt mạch, một bàn ủi quần áo, giấy nhám làm nhám board đồng để mực dính vào, một cây bút vẽ mạch để vẽ lại đường mạch nếu trong quá trình ủi bị đứt đường mạch, chì hàn, một mỏ hàn, một bộ khoan, thuốc rửa mạch (axit), nhựa thông để làm cho mạch in bóng và chống oxi hóa cho lớp đồng, chậu rửa, một kìm bấm chân linh kiện, một VOM để đo thông mạch và một số dụng cụ cần thiết khác.

Vẽ mạch in bằng phần mềm Altium, xuất file PDF.

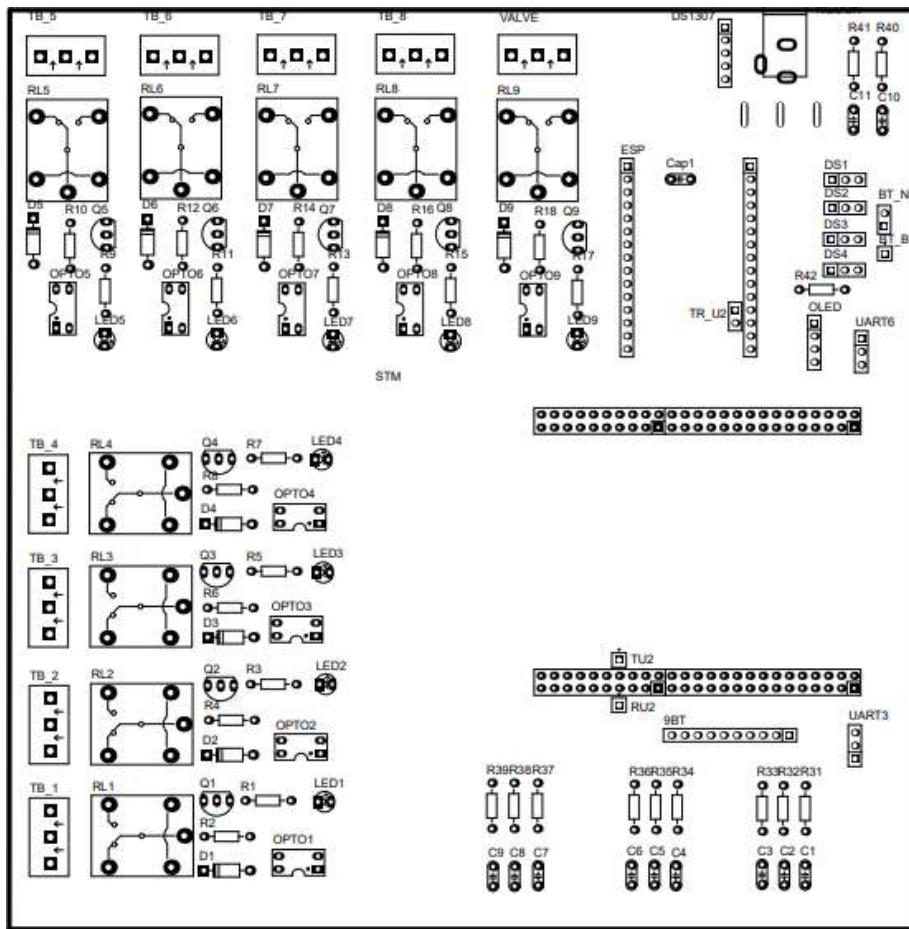
In mạch vào giấy in chuyên dụng, sau đó cắt board đồng vừa với mạch in áp vào mặt có mực in của giấy, giữ thật chặt, để bàn ủi ở chế độ nóng nhất rồi ủi lên mặt sau của tờ giấy in cho đến khi mực in thấm ra mặt sau của giấy thì dừng. (*Lưu ý: tránh trường hợp ủi quá lâu sẽ làm hỏng mạch*).

Sau đó để 2-3 phút cho người rời bóc lớp giấy in ra một cách nhẹ nhàng để tránh bị đứt các đường mạch, nếu đường mạch bị đứt, dùng bút vẽ mạch vẽ lại những đường mạch bị đứt. Tiếp theo, ngâm với thuốc rửa (axit) cho đến khi lớp đồng còn lại bị mát hết. Sau đó rửa sạch lớp mực, ta được board mạch hoàn chỉnh. Sau đó quét một lớp nhựa thông lỏng lên bề mặt đồng vừa rửa để bảo vệ khỏi oxi hóa. Dùng VOM đo xem mạch có thông nhau không, nếu có lỗi thì hàn lại cho thông mạch.

a. Mạch điều khiển trung tâm (chính)

- Sơ đồ bố trí linh kiện của mạch điều khiển trung tâm

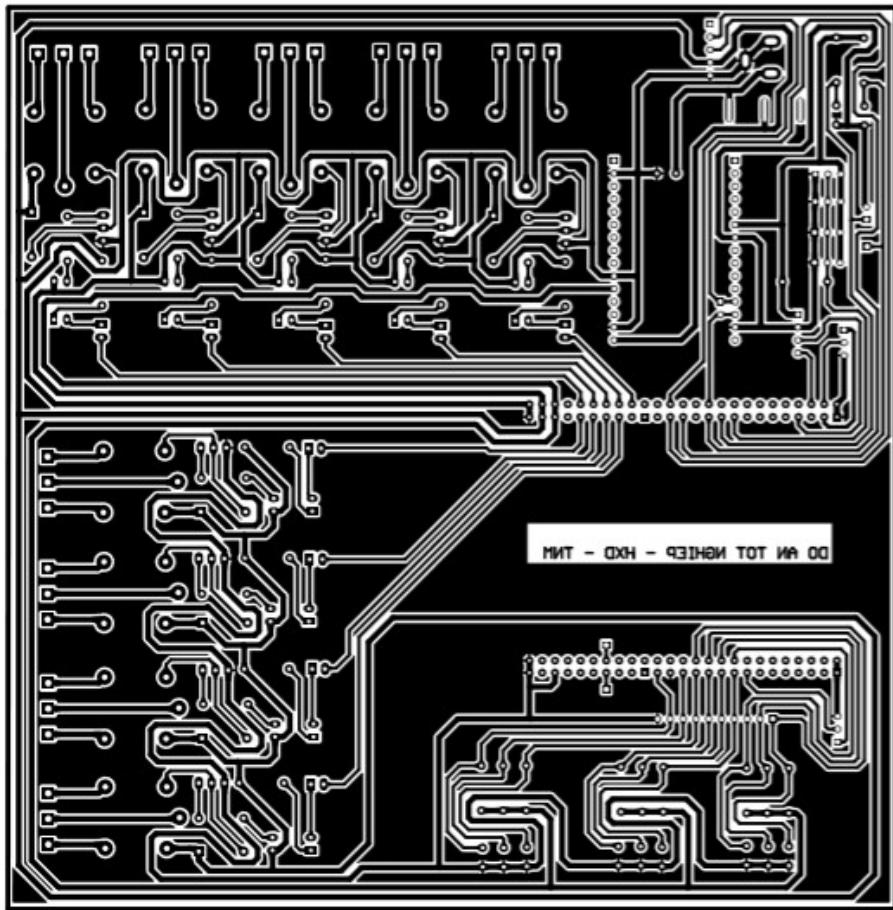
CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.1: Sơ đồ bố trí linh kiện mạch điều khiển trung tâm (chính)

Với sơ đồ bố trí linh kiện như trên, trung tâm của board mạch là vị trí gắn STM32F407. Phía trên STM32F407 là vị trí gắn hàng ESP8266. Phía trên ESP8266 là vị trí gắn module DS1307, Jack nguồn và công tắc nguồn. Bên phải ESP8266 là vị trí gắn 4 cảm biến DS18B20, màn hình oled được kéo dây ra cùng 2 nút nhấn Back, Neck cho Oled. Phía dưới STM32F407 là vị trí cảm dây kết nối ra nút nhấn điều khiển 8 thiết bị và bơm nước cùng với USART3 kết nối với mạch điều khiển trung tâm (phụ). Phía bên tay trái là mạch module relay.

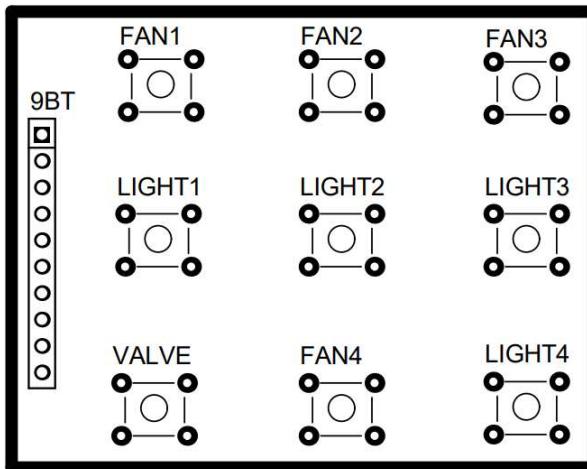
- Sơ đồ đi dây của mạch điều khiển trung tâm (chính)



Hình 4.2: Sơ đồ đi dây mạch điều khiển trung tâm (chính)

b. Mạch nút nhấn điều khiển thiết bị

- Sơ đồ bố trí linh kiện của mạch nút nhấn điều khiển thiết bị



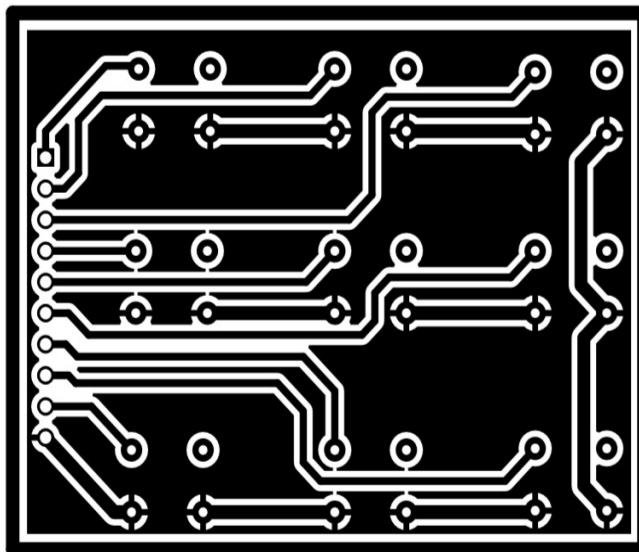
Hình 4.3: Sơ đồ bố trí linh kiện mạch nút nhấn điều khiển thiết bị

Với sơ đồ bố trí linh kiện như trên, 4 nút nhấn điều khiển đèn đặt ở các vị trí light1, light2, light3, light4, 4 nút nhấn điều khiển quạt đặt ở các vị trí fan1, fan2, fan3, fan4 và

CHƯƠNG 4: THI CÔNG HỆ THỐNG

1 nút nhấn điều khiển bơm nước đặt ở vị trí valve. Ở vị trí 9BT là vị trí gắn dây nối cho 9 nút nhấn và mass từ mạch điều khiển trung tâm qua. Do trỏ kéo lên VCC của nút nhấn được đặt trong mạch điều khiển trung tâm (chính) nên ở mạch này không cần nối VCC.

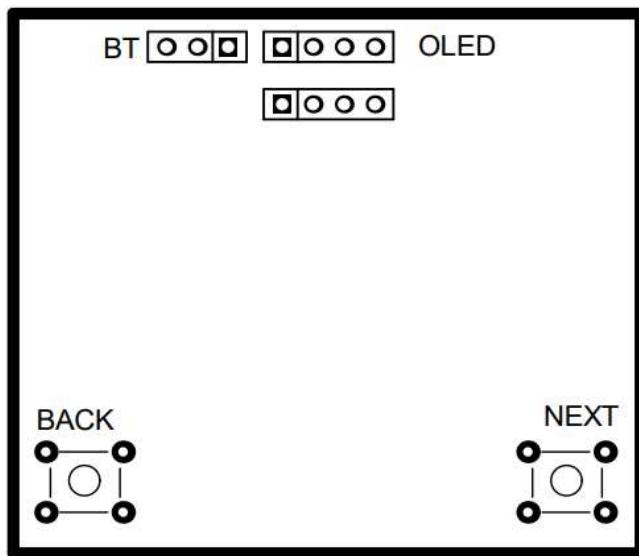
- Sơ đồ đi dây của mạch nút nhấn điều khiển thiết bị



Hình 4.4: Sơ đồ đi dây mạch điều khiển thiết bị

c. Mạch màn hình hiển thị oled

- Sơ đồ bố trí linh kiện của mạch màn hình hiển thị oled

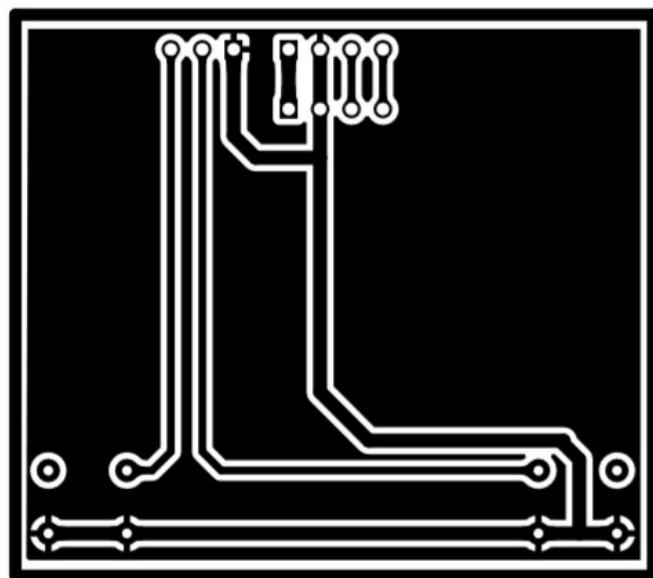


Hình 4.5: Sơ đồ bố trí linh kiện mạch màn hình hiển thị oled

Với sơ đồ bố trí linh kiện như trên, 2 nút nhấn chuyển màn hình hiển thị cho Oled đặt ở 2 vị trí back và next. Phía trên là vị trí gắn Oled và dây nối từ mạch điều khiển trung tâm qua cho Oled và 2 nút nhấn.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

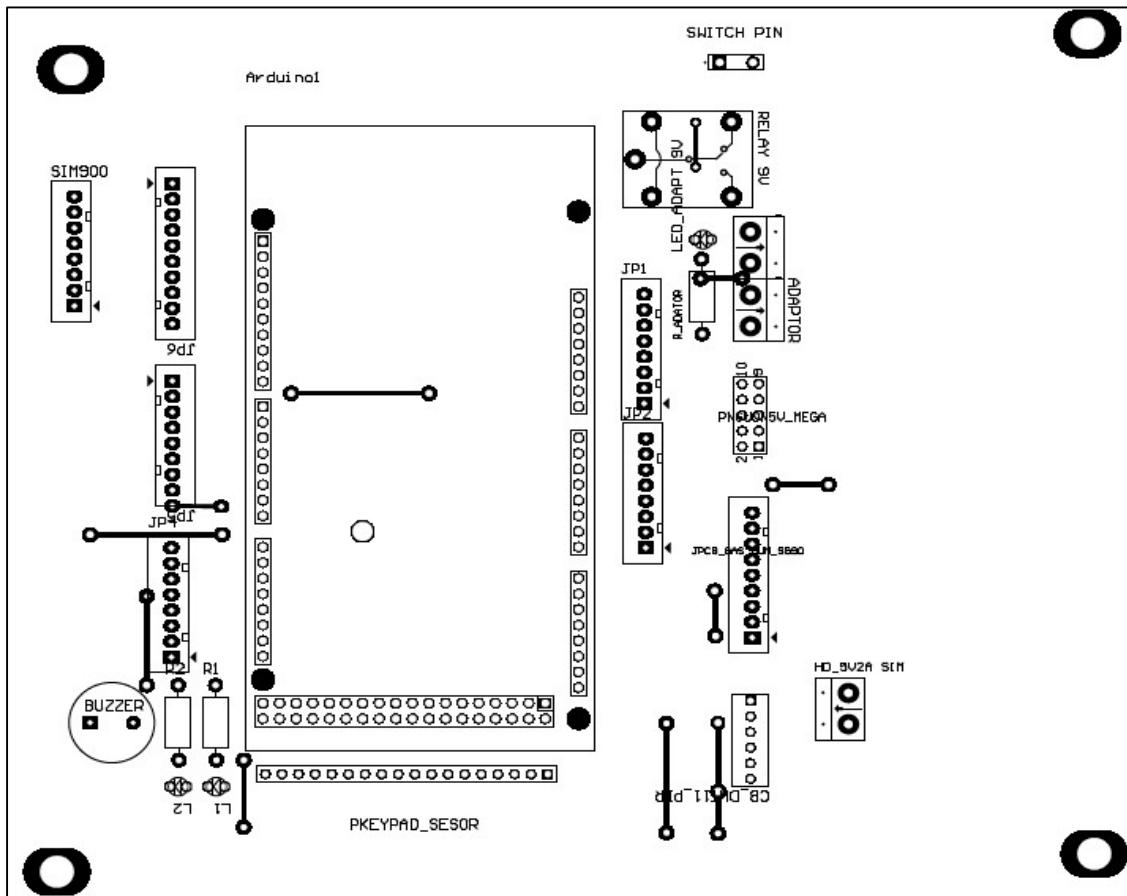
- Sơ đồ đi dây của mạch màn hình hiển thị oled



Hình 4.6: Sơ đồ đi dây mạch màn hình hiển thị oled

d. Mạch điều khiển (phụ)

- Sơ đồ bố trí mạch điều khiển phụ

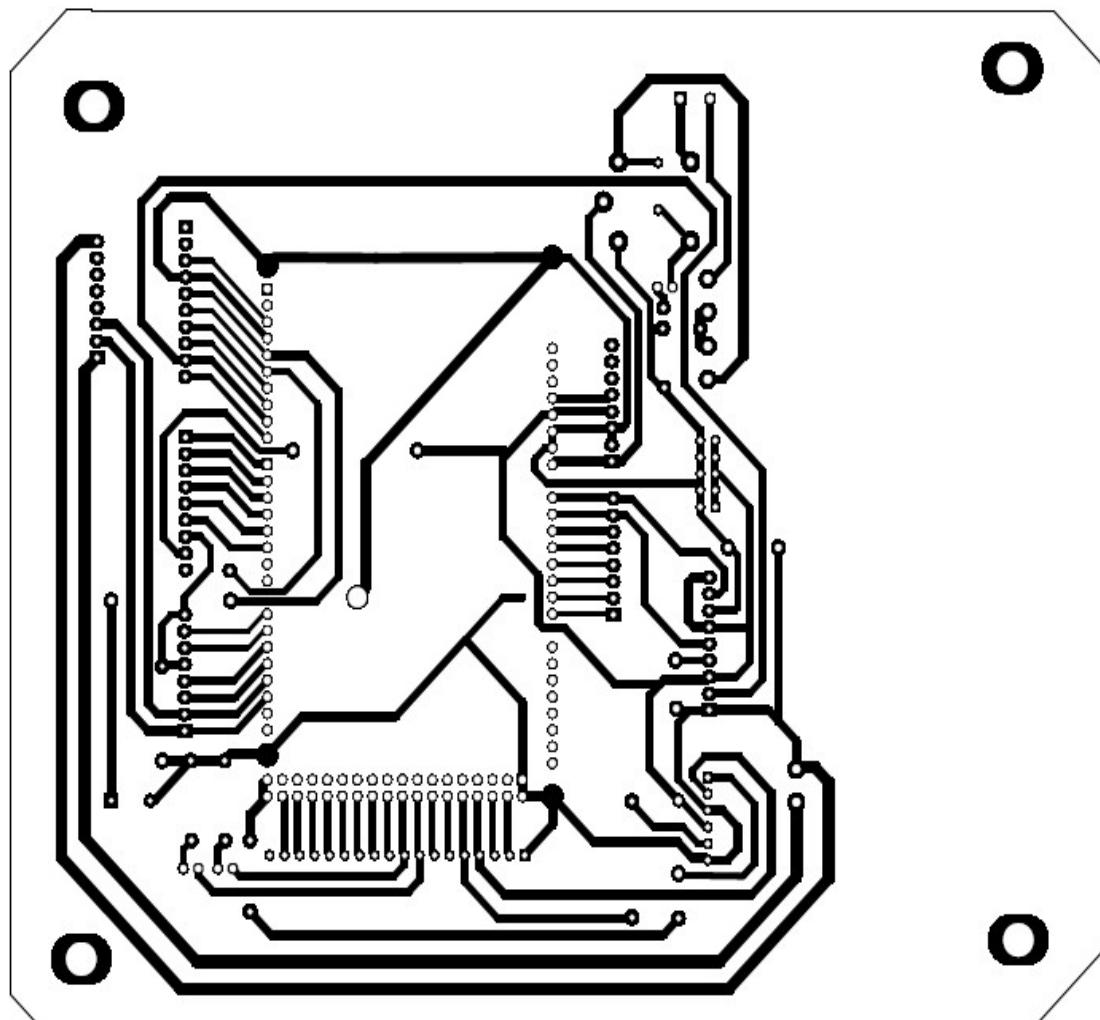


Hình 4.7: Sơ đồ bố trí linh kiện mạch điều khiển (phụ)

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Theo như sơ đồ bố trí như trên thì ở trung tâm board mạch điều khiển phụ là vị trí gắn Arduino Mega 2560. Phía ngoài cùng bên trái là vị trí gắn hàng domino để cắm module SIM (cổng UART1 của Arduino), kế bên là hàng domino của các chân inout (các chân từ 2 đến 13), phía bên dưới nữa là nơi đặt hàng domino kết nối cổng UART1, UART2, UART3, còi báo, 2 điện trở và 2 LED. Phía bên phải mạch theo thứ tự từ trên xuống là vị trí đặt 1 công tắc ngắt nguồn pin, 1 relay, 2 domino 2 đầu ra, 1 LED báo nguồn, hàng domino kết nối các chân nguồn của Arduino (3.3V, 5V, GND, Vin). Tiếp đến là domino để cắm các linh kiện ngoại vi (cảm biến khí gas, độ ẩm đất, động cơ servo SG90, cảm biến DHT11 và cảm biến PIR), kế bên là vị trí gắn domino kết nối nguồn 5V-2A để cấp cho module SIM. Hàng dưới cùng là vị trí đặt domino kết nối các chân inout DIGITAL (các chân số lẻ từ 23 đến 53).

- ❖ Sơ đồ đi dây của mạch điều khiển (phụ):



Hình 4.8: Sơ đồ đi dây của mạch điều khiển (phụ)

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Bảng 4.1: Bảng thông kê linh kiện

STT	Tên linh kiện	Giá trị	Số lượng	Chú thích
1	Module nguồn	12V-10A	1	Nguồn tổ ong
2	Nguồn 5V	5V-3A	1	Hạ áp từ nguồn tổ ong
3	Nguồn 12V	12V-1A	1	Nguồn cấp cho bơm nước
4	LM2596		3	Module hạ áp
5	Diode N4007	5V	9	
6	ESP8266 NodeMCU	3.3V	1	Wifi
7	STM32F407	5V	1	Ví điều khiển
8	Arduino Mega	5V	1	Atmega2560
9	DS1307	5V	1	Module thời gian thực
10	DS18B20	3.3V	4	Cảm biến nhiệt độ
11	DHT11	5V	1	Cảm biến nhiệt độ độ ẩm
12	MQ_02	5V	1	Cảm biến khí gas
13	HC_SR501	5V	1	Cảm biến chuyển động
14	Cảm biến đất	5V	1	Cảm biến độ ẩm đất
15	Module SIM	5V	1	
16	RFID	3V3	1	
17	Servo SG90	5V	1	Động cơ servo
18	Buzzer	5V	1	Còi chíp
19	Bàn phím 4x4	5V	1	
20	LCD	5V	1	16x2
21	Đèn pin		1	Đèn 2 pin 18650
22	Oled	3.3V	1	Màn hình hiển thị
23	Điện trở	1kΩ	9	
24	Jac DC	5V	1	JAC nguồn
25	PC817	5V	9	Opto
26	Transistor D468	5V	9	

CHƯƠNG 4: THI CÔNG HỆ THỐNG

27	Led đơn	5V	11	2 led (Arduino), 9 led (STM)
28	Điện trở	220Ω	9	
29	Điện trở	10kΩ	11	
30	Điện trở	4.7kΩ	1	
31	Relay	5V-10A	10	1 (Arduino), 9 (STM)
32	Domino 3 chân		9	
33	Domino 2 chân		1	
34	Công tắc nguồn	5V	1	
35	Nút nhấn	3.3V	11	
36	Switch		1	
37	Hàng rào cái đôi		2	
38	Hàng rào cái đơn		2	
39	Hàng rào đực đơn		2	
40	Tụ 104		12	Tụ gốm
41	Quạt	12V	4	
42	Bơm nước	12V	1	
43	Đèn	220VAC	4	

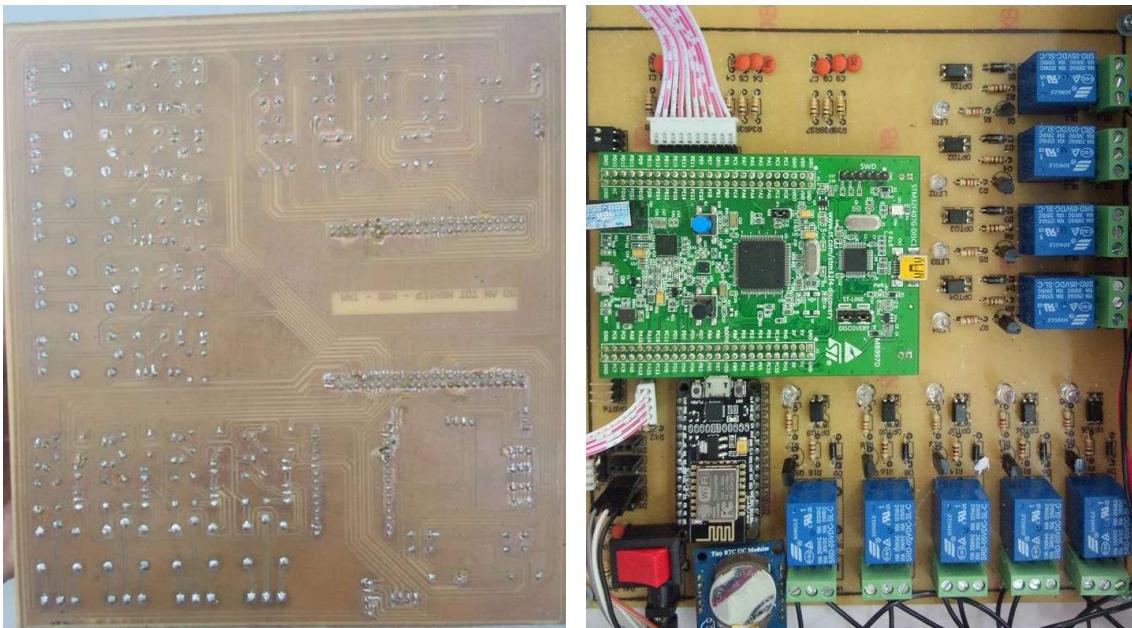
4.2.2 Lắp ráp và kiểm tra

Sau khi đã hoàn thành phần mạch in ta bắt tay vào khoan lỗ, lắp linh kiện để hàn. Các mũi khoan để khoan: mũi 0.8 ly dùng để khoan các con trỏ và IC, mũi 1 ly dùng để khoan lỗ Jump, mũi 3 ly dùng để khoan lỗ bắt ốc. Khi in mạch ra để ý vào tâm của chân linh kiện có các lỗ trống, khoan vào những lỗ đó để thê hàn linh kiện được đẹp.

Kiểm tra mạch xem hệ thống bị đứt hay chập mạch hay không nhò vào đồng hồ VOM. Kiểm tra chân nguồn, chân điều khiển. Chú ý, quan sát kĩ các đường mạch có bị sát vào nhau hay không, nếu có thì dùng mũi khoan cỡ 0.8 ly tách nhẹ đường mạch bị sát vào nhau. Kiểm tra nếu có bị đứt dây thì ta dùng dây điện nối lại.

a. Lắp ráp mạch điều khiển trung tâm (chính)

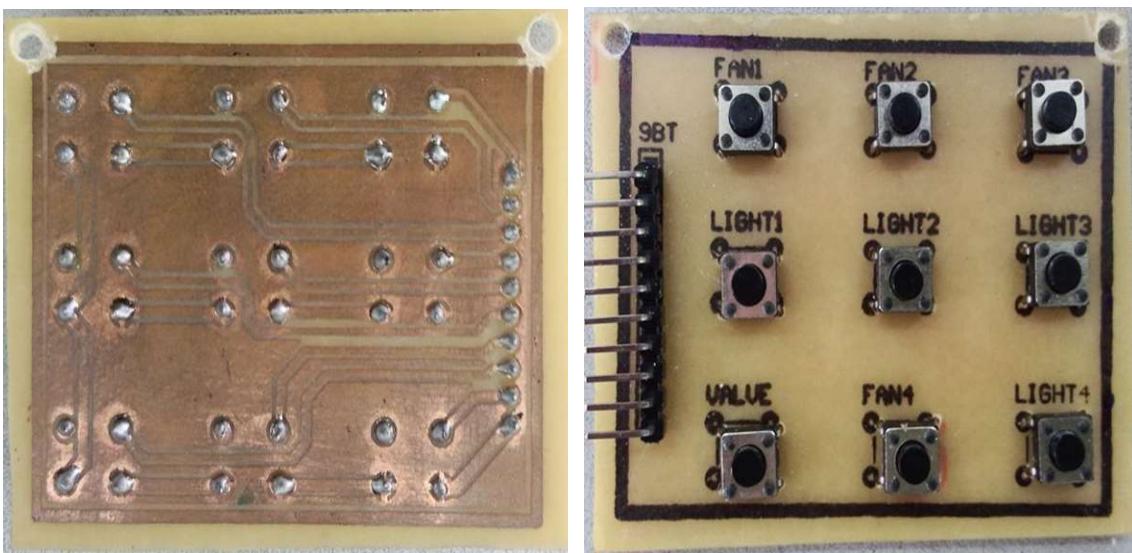
Sau khi đã ủi mạch, tiến hành gắn và hàn linh kiện theo sơ đồ mạch nguyên lý vào mạch, mạch sau khi thi công xong có hình ảnh thực tế như hình 4.9.



Hình 4.9: Hình ảnh thực tế mạch điều khiển trung tâm (chính)

b. Lắp ráp mạch nút nhấn điều khiển thiết bị

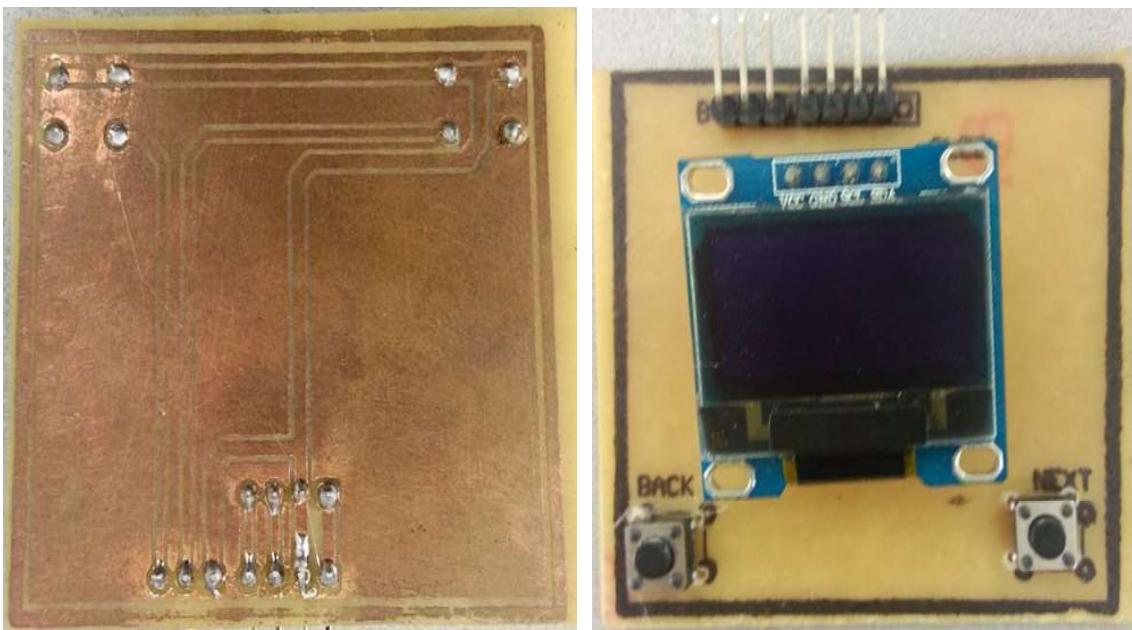
Mạch nút nhấn sau khi ráp sẽ có 1 hàng rào để kết nối vào mạch điều khiển trung tâm và 9 nút nhấn để điều khiển 8 thiết bị và 1 bơm nước. Mạch sau khi thi công có hình ảnh thực tế như hình 4.10.



Hình 4.10: Hình ảnh thực tế mạch nút nhấn điều khiển thiết bị

c. Lắp ráp mạch màn hình hiển thị oled

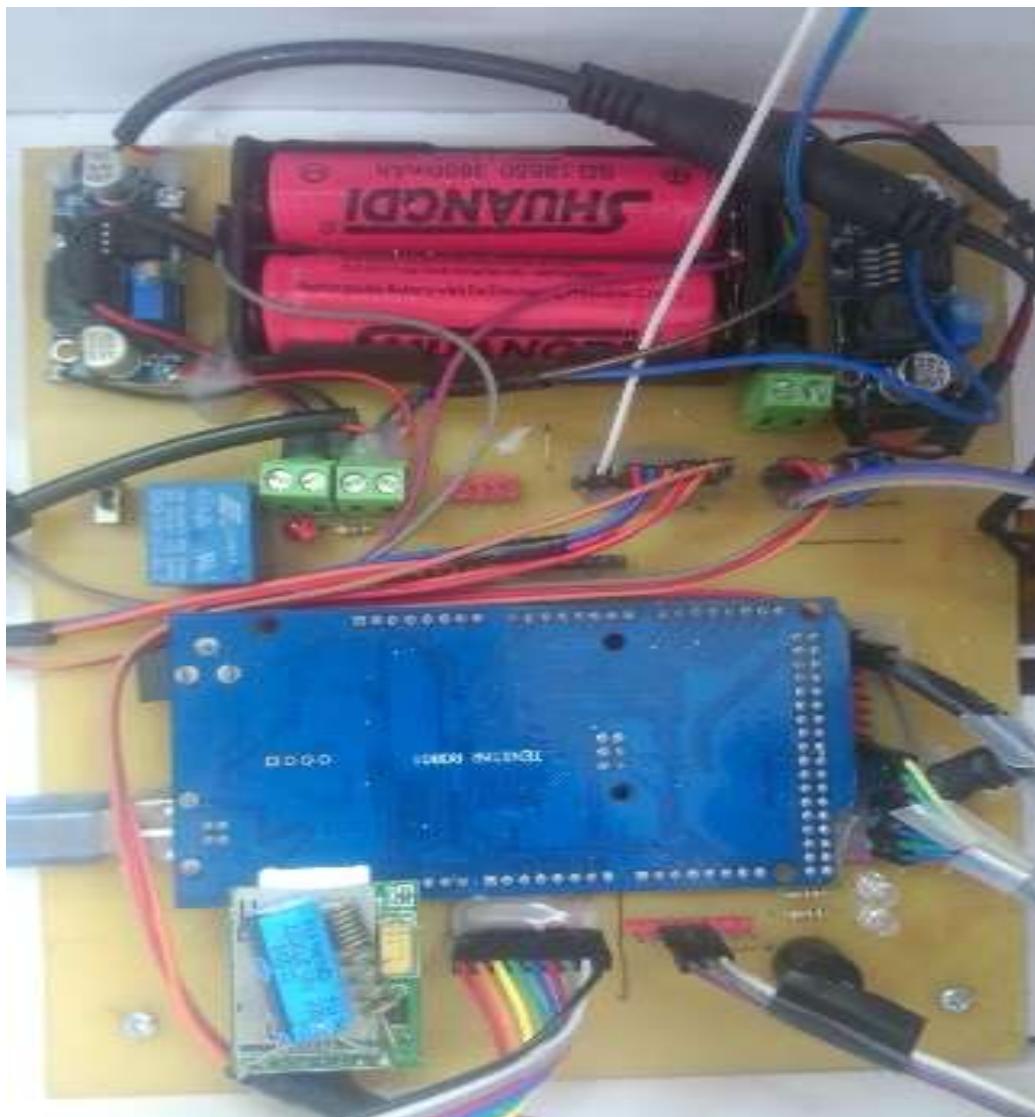
Mạch màn hình hiển thị Oled khá đơn giản nên việc ráp mạch vào thi công khá dễ dàng, mạch có thêm 2 nút nhấn để chuyển chế độ hiển thị. Sau khi thi công mạch có hình ảnh thực tế như hình 4.11.



Hình 4.11: Hình ảnh thực tế mạch màn hình hiển thị Oled

d. Lắp ráp mạch điều khiển trung tâm (phụ)

Mạch điều khiển trung tâm phụ với khá nhiều linh kiện nên việc thi công khá khó khăn. Cũng như các mạch trên, tiến hành gắn linh kiện vào và hàn linh kiện. Sau khi thi công xong mạch có hình ảnh thực tế như hình 4.12.



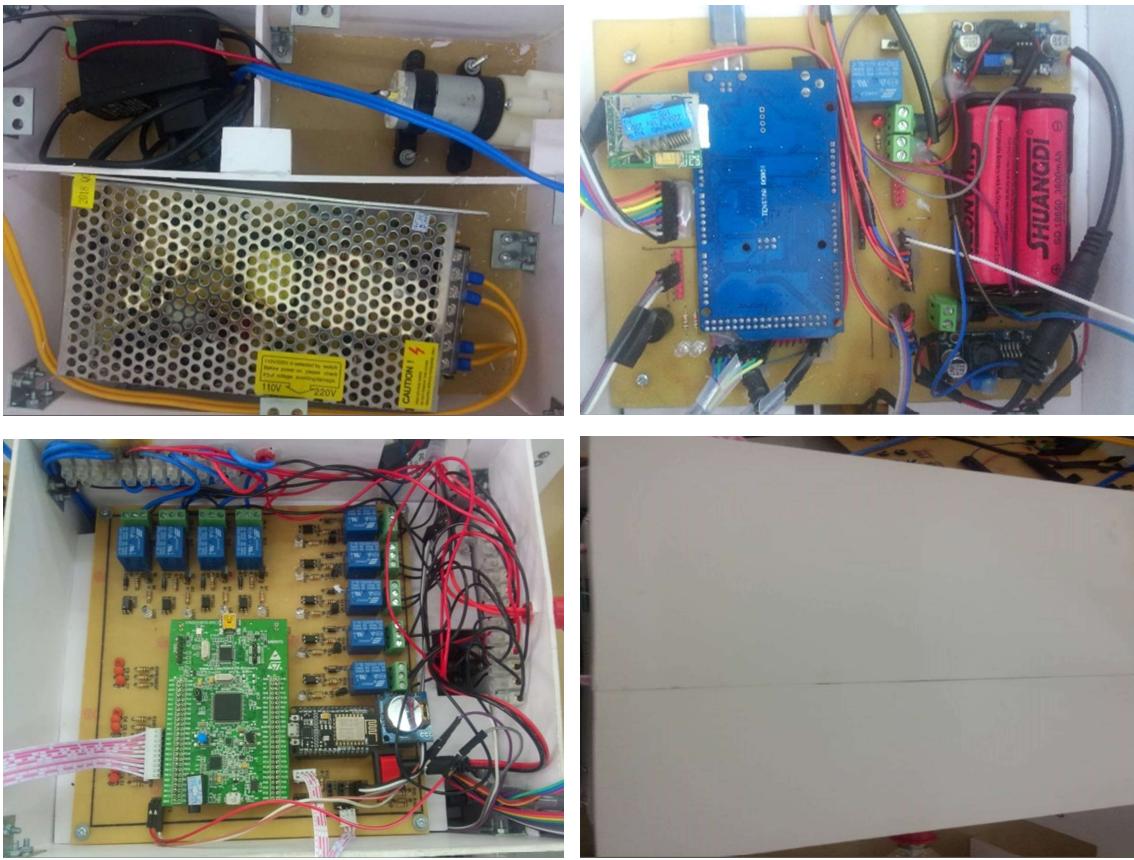
Hình 4.12: Hình ảnh thực tế mạch điều khiển trung tâm (phụ)

4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

4.3.1 Đóng gói bộ điều khiển

Sau khi thi công, kiểm tra và rắp ráp xong các board mạch hệ thống, nhóm tiến hành đóng gói bộ điều khiển như hình:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.13: Hộp bảo vệ bộ điều khiển của hệ thống

Bộ điều khiển được đóng gói trong một hộp hình chữ nhật được làm từ bìa trắng dày 3mm. Hộp có chiều dài là 25 cm, chiều rộng 20 cm và chiều cao là 20 cm. Hộp gồm 3 lớp, dưới cùng là lớp để nguồn cấp cho toàn bộ hệ thống, lớp giữa để mạch điều khiển (phụ) và lớp trên cùng để mạch điều khiển (chính) để tiện đi dây kết nối ra công suất. Dùng mũi khoan khoan lỗ để bắt ốc bìa trắng với nhau và gắn vào mica để cố định.

4.3.2 Thi công mô hình

Mô hình được làm bằng chất liệu foam – một dạng bọt xốp thường được sử dụng để làm mô hình trong ngành kiến trúc. Loại nguyên liệu này rất dễ mua, thể tìm thấy ở các cửa hàng bán vật liệu quảng cáo.... Tùy theo nhu cầu sử dụng mà có thể mua loại cứng hay mềm, khổ to hay nhỏ. Đặc điểm của foam là tuy nhẹ nhưng rất dẽ vỡ.

Để cắt được foam thì ta dùng dao rọc giấy. Cách này khó khi muốn cắt những thành phần cần độ chính xác và thẩm mỹ cao, do đó cần đo đạc, thao tác chính xác và khéo léo để đạt yêu cầu về thẩm mĩ. Để dán mảnh ghép sau khi cắt, ta sử dụng óc hoặc keo nến để liên kết và keo chuyên dụng để dán foam.

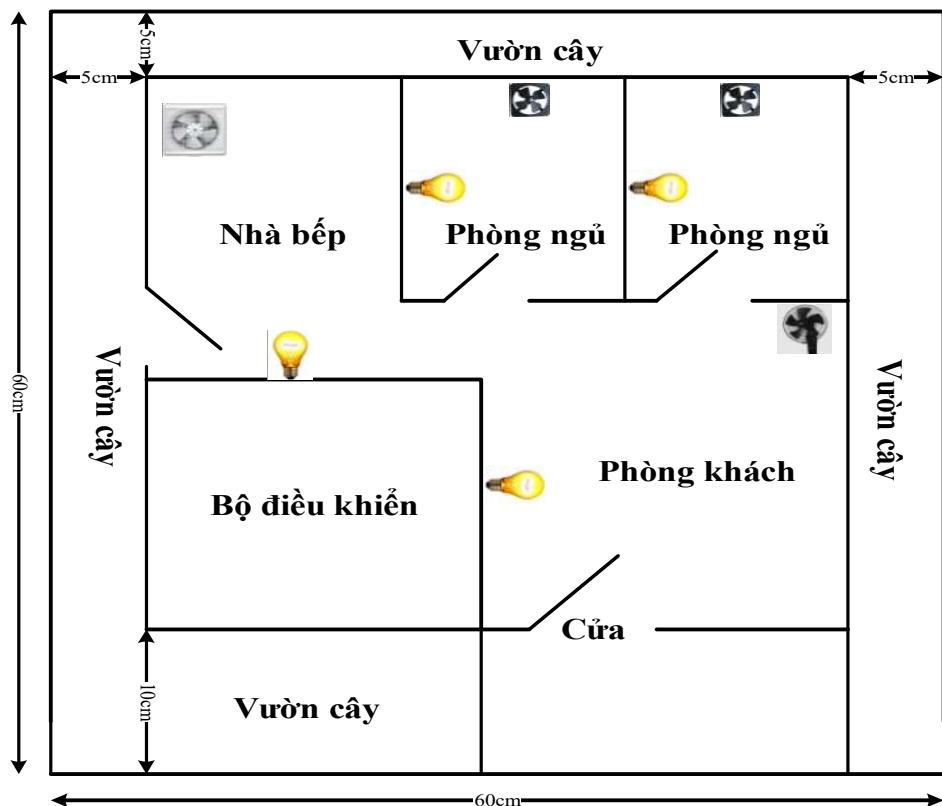
a. Chuẩn bị vật liệu cần thiết

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- 1 tấm foam: dày 4.5mm, kích thước 60x60 cm.
- 3 tấm foam: dày 3mm, kích thước 60x60 cm.
- 1 tấm nhựa mica, cỏ, keo nén dán foam, ốc vít, dao rọc giấy, 1 cây thước dài 50 cm, 1 cây thước dài 20 cm, bút chì + tẩy, kéo, mũi khoan, súng bắn keo.

b. Thi công

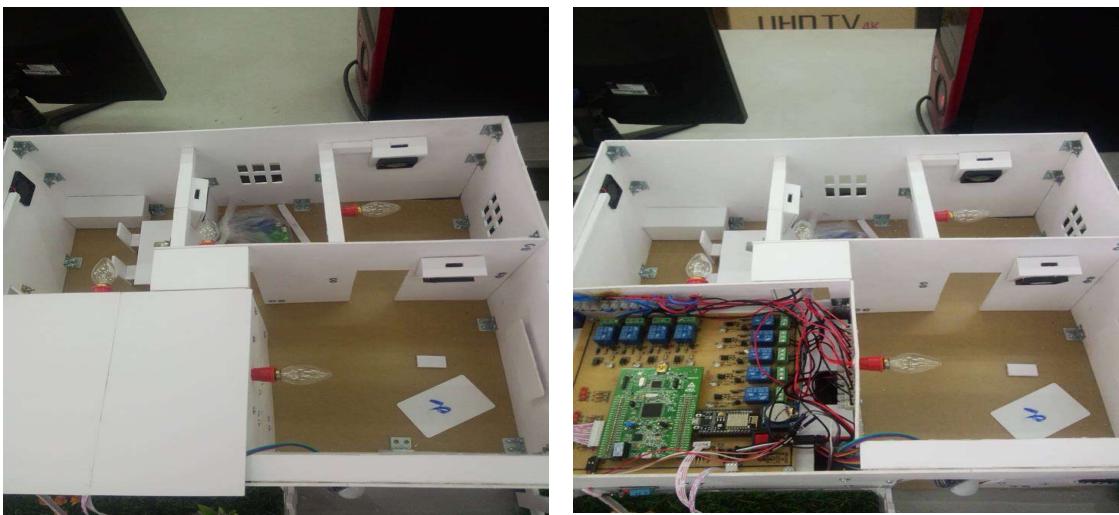
Chuẩn bị 4 tấm foam và 1 tấm mica kích thước 60x60 cm để làm nguyên liệu làm mô hình.



Hình 4.14: Hình thành ý tưởng thiết kế mô hình

Đầu tiên, hình thành ý tưởng về tỷ lệ, bố cục và vẽ thiết kế mô hình trên giấy. Sau đó tiến hành đo đạc và cắt foam.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.15: Hoàn chỉnh bộ khung và tiến hành đi dây.

Sau khi đi dây thì tiến hành dán dây để mô hình được đẹp và an toàn hơn.



Hình 4.16: Trang trí và lắp ráp hoàn thiện mô hình

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Trang trí thêm cây cỏ cho vườn, hàng rào, mái nhà, cho mô hình thêm sinh động, thẩm mĩ.

- Tiến hành thi công, lắp ráp, hoàn thiện LCD, RFID, động cơ servo đóng mở cửa vào mặt trước của mô hình.

4.4 LẬP TRÌNH HỆ THỐNG

4.4.1 Lưu đồ giải thuật

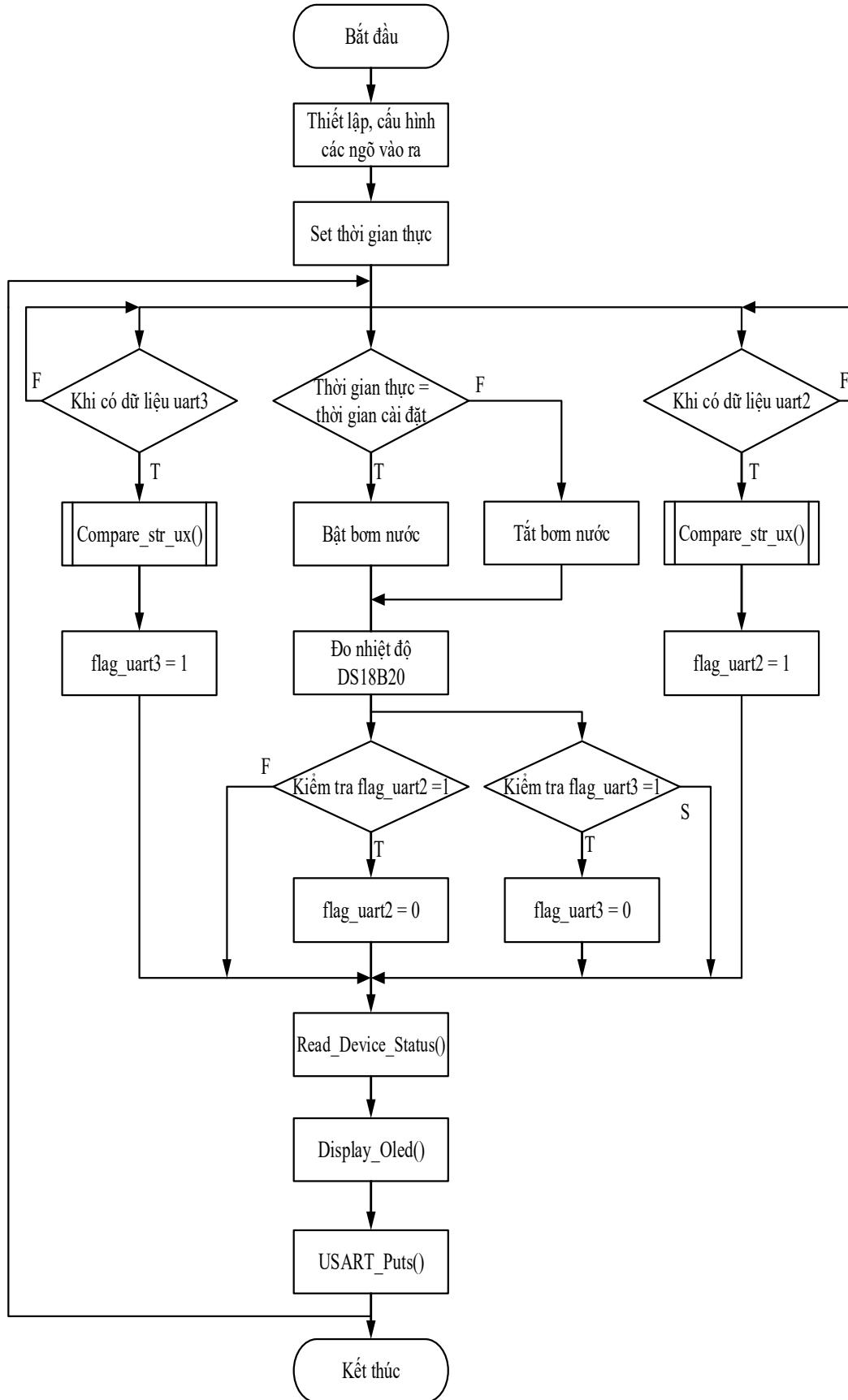
Yêu cầu điều khiển: ban đầu khi cấp nguồn, quét thẻ hay nhập mật khẩu thì động cơ servo được gắn ngay cửa để đóng mở cửa vào nhà.

Sau khi kết nối thành công Wifi, ta truy cập trang <http://iot-datn.firebaseio.com>, mở app android hay WPF ở giao diện đăng nhập ta tiến hành nhập tài khoản trên để đăng nhập được vào trang chủ. Sau đó điều khiển bật tắt thiết bị hay bơm nước tùy ý và giám sát hệ thống.

a. Lưu đồ chương trình STM32F407

❖ Lưu đồ chương trình chính STM32F407

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.17: Lưu đồ chương trình STM32F407

CHƯƠNG 4: THI CÔNG HỆ THỐNG

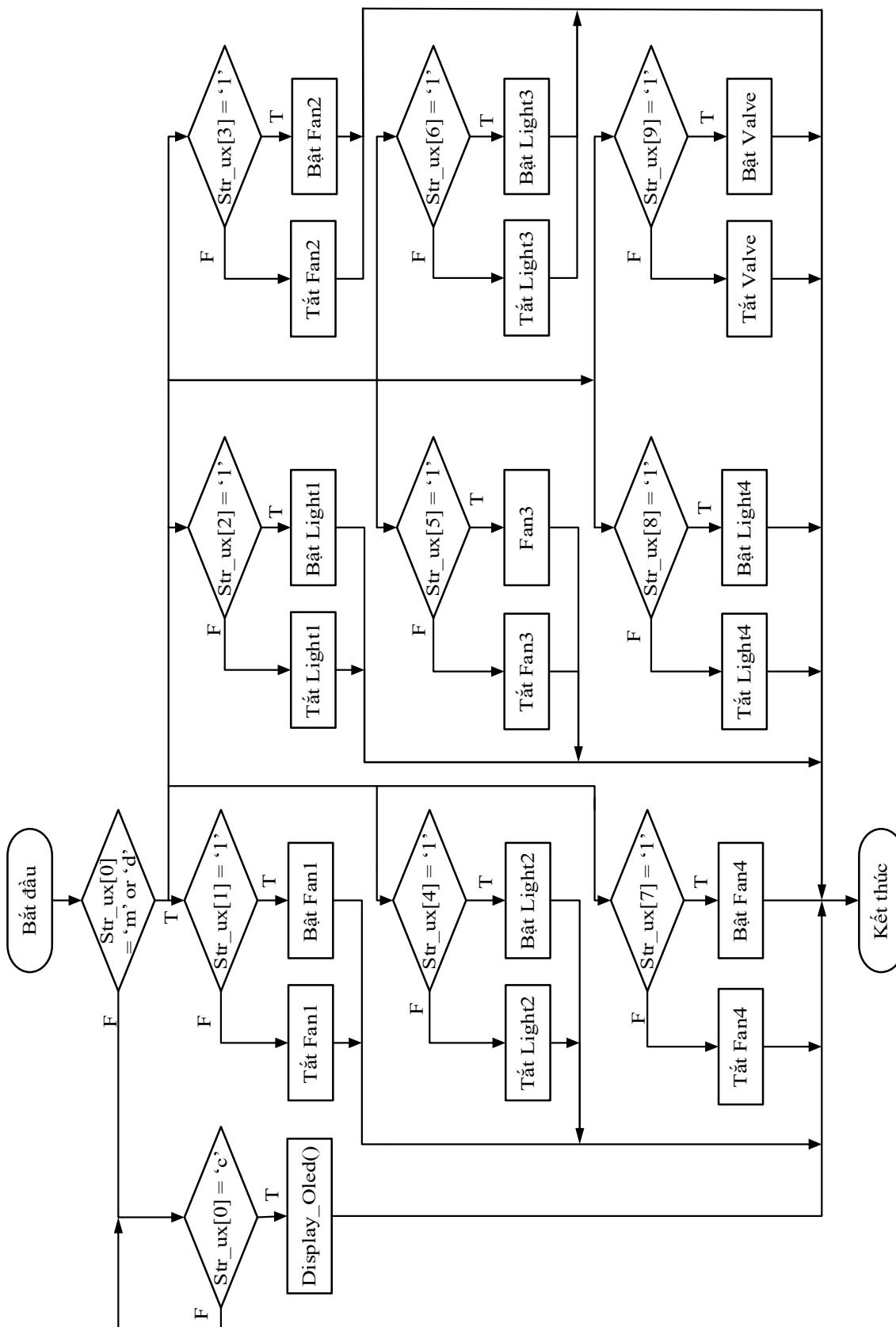
- Giải thích lưu đồ:

Ban đầu khởi tạo các giá trị bạn đầu, khai báo các chân vào ra, khai báo cáo biến dùng trong chương trình sau đó set thời gian thực cho ds1307, nếu thời gian thực của ds1307 bằng thời gian cài đặt thì bật bơm nước tưới cây theo lịch trình đã cài đặt và nếu không bằng nhau thì tắt bơm nước.

Tiếp theo, đọc bốn cảm biến ds18b20 cho bốn phòng theo thời gian của ds1307, cứ 5 giây cho cảm biến đọc một lần.

Kiểm tra cờ uart2 và uart3, nếu bằng 1 thì cho cờ về 0. Khi có dữ liệu uart2 hoặc dữ liệu uart3 thì gọi hàm *Compare_str_ux()*. Sau đó, đọc trạng thái các thiết bị và van nước (hàm *Read_Device_Status()*) hiển thị lên màn hình oled (hàm *Display_Oled()*) rồi đóng gói tất cả dữ liệu ghép thành chuỗi (“nhiệt độ + độ ẩm + độ ẩm đất + trạng thái bơm nước + trạng thái cửa + khí gas + trạng thái quạt phòng 1 + trạng thái quạt phòng 1 + nhiệt độ phòng 1 + trạng thái quạt phòng 2 + trạng thái đèn phòng 2 + nhiệt độ phòng 2 + trạng thái quạt phòng 3 + trạng thái đèn phòng 3 + nhiệt độ phòng 3 + trạng thái quạt phòng 4 + trạng thái đèn phòng 4 + nhiệt độ phòng 4”) gửi qua uart2 (hàm *USART_Puts()*) để cập nhật lên web, android hay WPF.

- **Lưu đồ chương trình con Compare_str_ux()**



Hình 4.18: Lưu đồ chương trình con điều khiển thiết bị

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Giải thích lưu đồ:

Khi có dữ liệu uart, so sánh chuỗi nhận được Str_ux[]. Nếu Str_ux[0] = ‘c’ thì dữ liệu nhận được là dữ liệu các cảm biến nhận từ uart3 kết nối với Arduino sau đó cho hiển thị lên Oled.

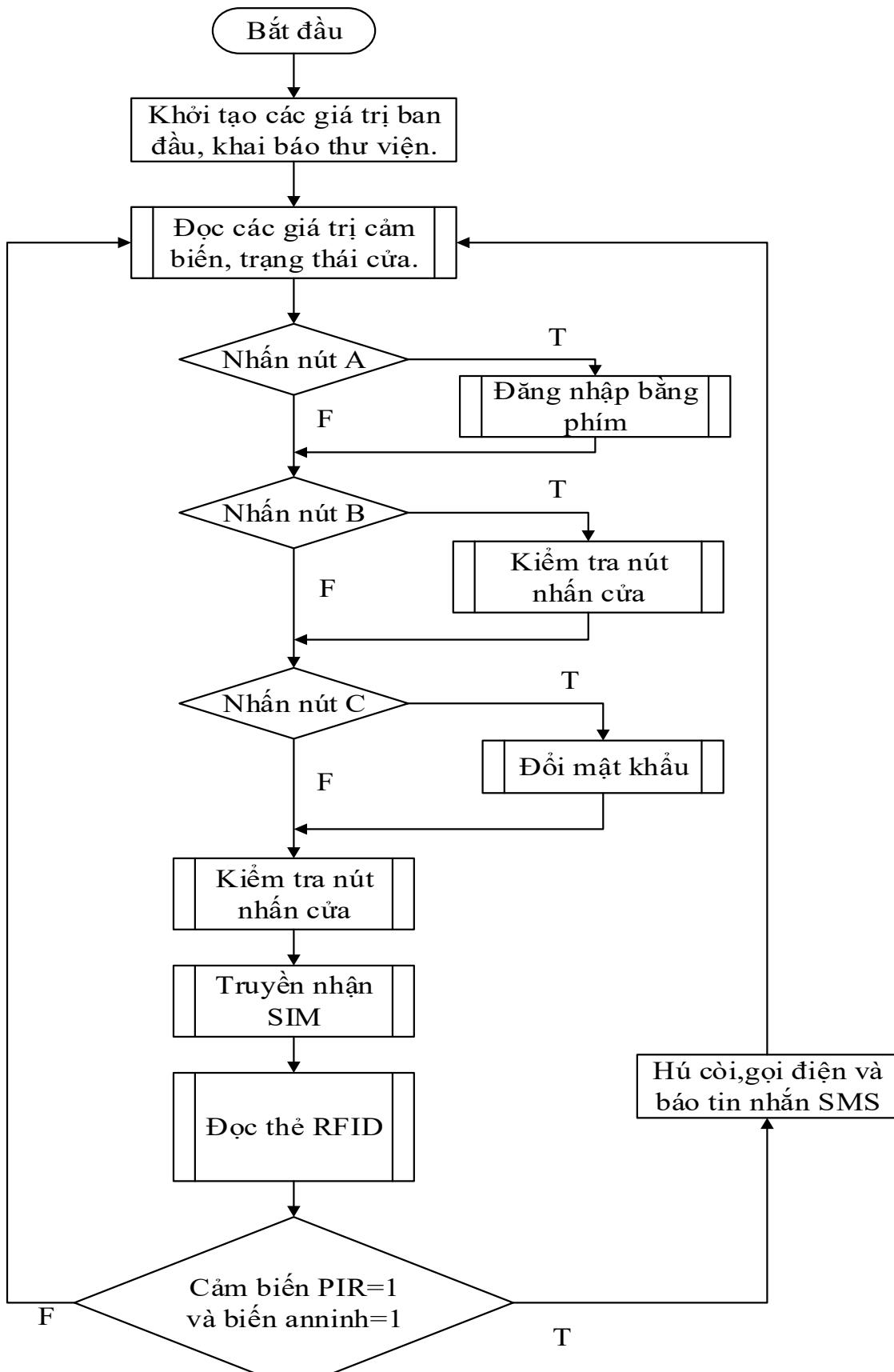
Nếu Str_ux[0] = ‘d’ hoặc ‘m’ thì bắt đầu điều khiển thiết bị. Str_ux[1] = ‘1’ thì bật Fan1 ngược lại str_ux[1] = ‘0’ thì tắt Fan1, str_ux[2] = ‘1’ thì bật Light1 ngược lại str_ux[2] = ‘0’ thì tắt Light1.

Tương tự cho các phần tử khác trong chuỗi, nhưng ở đây có sự khác nhau khi nhận Str_ux[0], nếu là ‘d’ thì điều khiển thêm hệ thống bơm nước ở str_ux[10] còn nếu nhận str_ux[0] là ‘m’ thì chỉ điều khiển 8 thiết bị và không điều khiển bơm nước.

b. Lưu đồ chương trình Arduino

- **Lưu đồ chương trình chính Aduino**

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.19: Lưu đồ chương trình chính Arduino

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Giải thích lưu đồ chương trình:

Khai báo các thư viện sử dụng trong chương trình như <LiquidCrystal.h>, <Servo.h>, <DHT.h>, <SPI.h>, <MFRC522.h>, "sim.h", "KeyPad4x4.h", định nghĩa các chân sử dụng, sau đó khởi tạo trạng thái của các chân INPUT, OUTPUT, PWM.

Nếu A nhấn → Vào login.

Nếu B nhấn → Vào kiểm tra nút nhấn cửa.

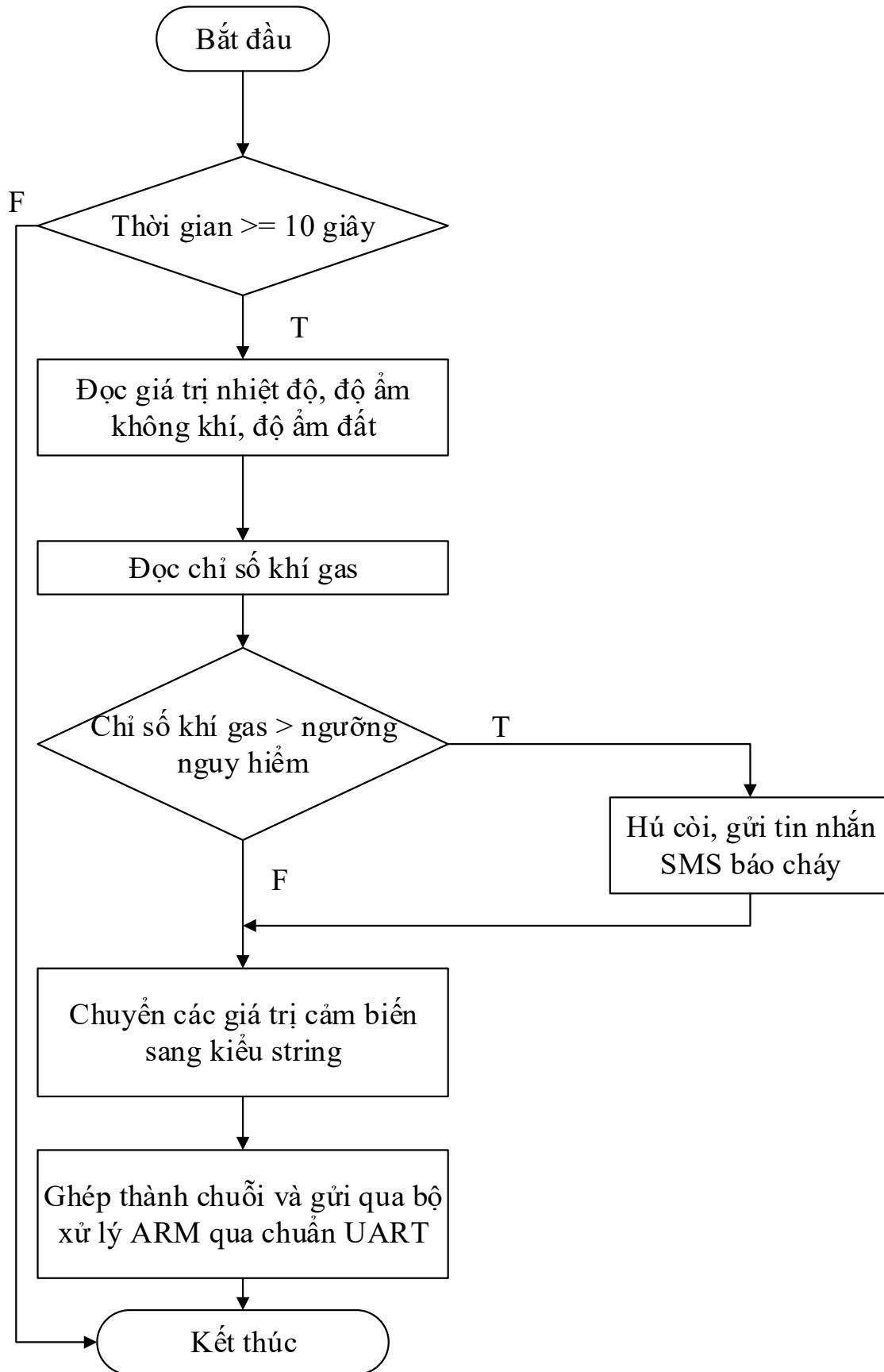
Nếu C nhấn → Vào chương trình đổi mật khẩu.

- Chương trình con đọc giá trị cảm biến.
- Chương trình con đăng nhập bằng phím.
- Chương trình con truyền nhận SIM.
- Chương trình con đọc thẻ RFID.

Chương trình con kiểm tra trạng thái an ninh và xử lý nếu có bật an ninh và có trộm thì chuông kêu, nhấn tin và gọi điện về điện thoại.

- **Lưu đồ chương trình con “Đọc cảm biến”:**

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.20: Lưu đồ chương trình con đọc cảm biến

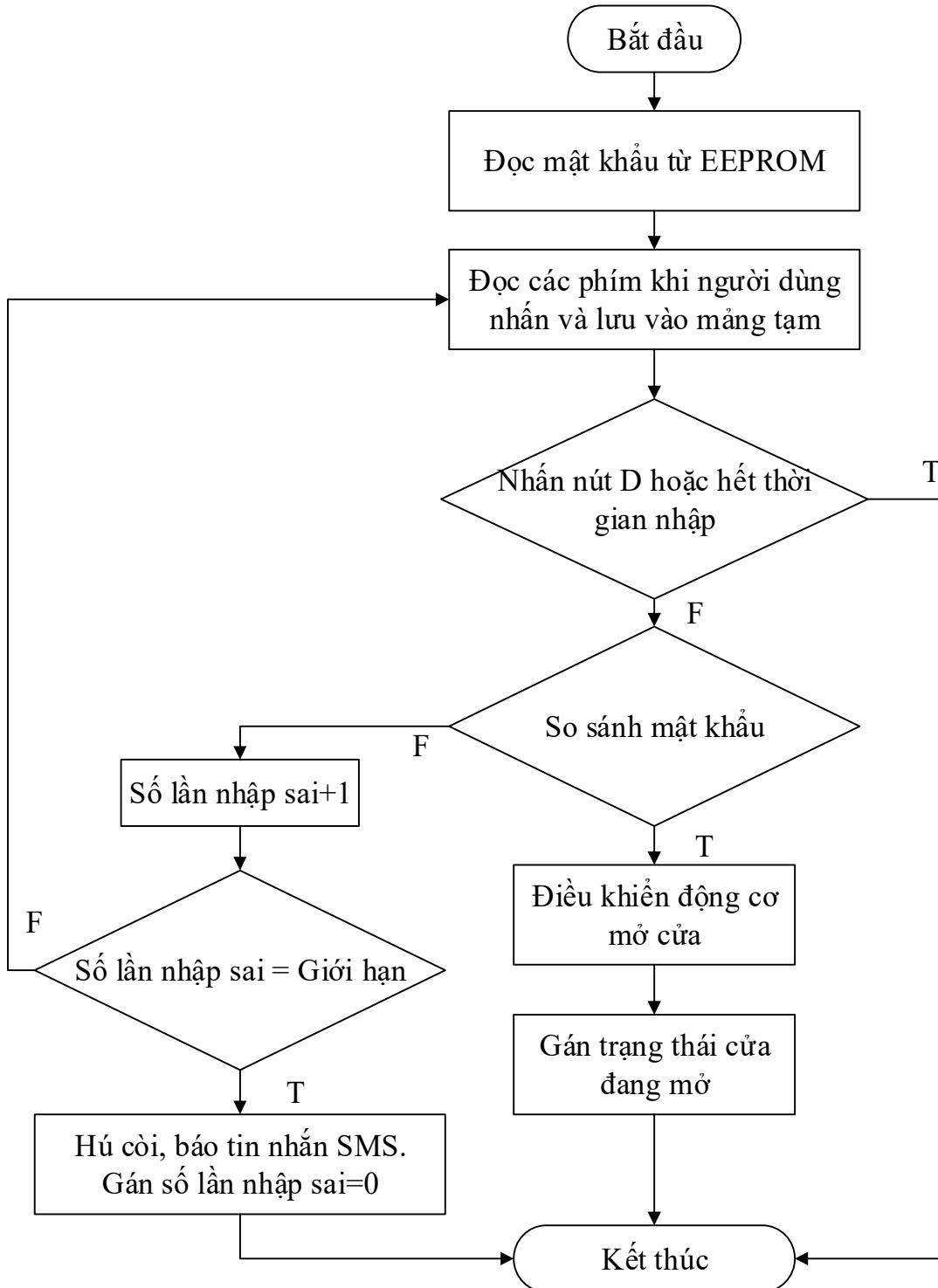
CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Giải thích lưu đồ:

Ban đầu kiểm tra thời gian nếu thời gian hiện tại bằng với thời gian đặt trước, giả sử thời gian đặt trước bây giờ là 10s. Nếu đúng bằng 10s thì chúng ta tiến hành đọc nhiệt độ, độ ẩm không khí từ cảm biến DHT11 và đọc cảm biến độ ẩm đất của vườn cây. Tiến hành đọc cảm biến khí gas. Nếu giá trị của cảm biến khí gas lớn hơn giá trị đặt trước (lớn nhất của cảm biến này là 1023) thì sẽ có chuông báo động thời sẽ gửi tin nhắn và gọi điện về cho điện thoại báo trạng thái khí gas cao và có thể có cháy trong nhà. Sau khi đọc hết tất cả cảm biến thì sẽ ép kiểu dữ liệu sang kiểu chuỗi, sau đó ghép chuỗi dữ liệu lại với nhau theo thứ tự nhất định ('c' + độ ẩm không khí + nhiệt độ không khí + độ ẩm đất + trạng thái cửa + trạng thái khí gas) để truyền qua ARM thông qua chuẩn truyền thông UART.

- **Lưu đồ chương trình con “Đăng nhập bằng phím”:**

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.21: Lưu đồ chương trình con đăng nhập bằng phím

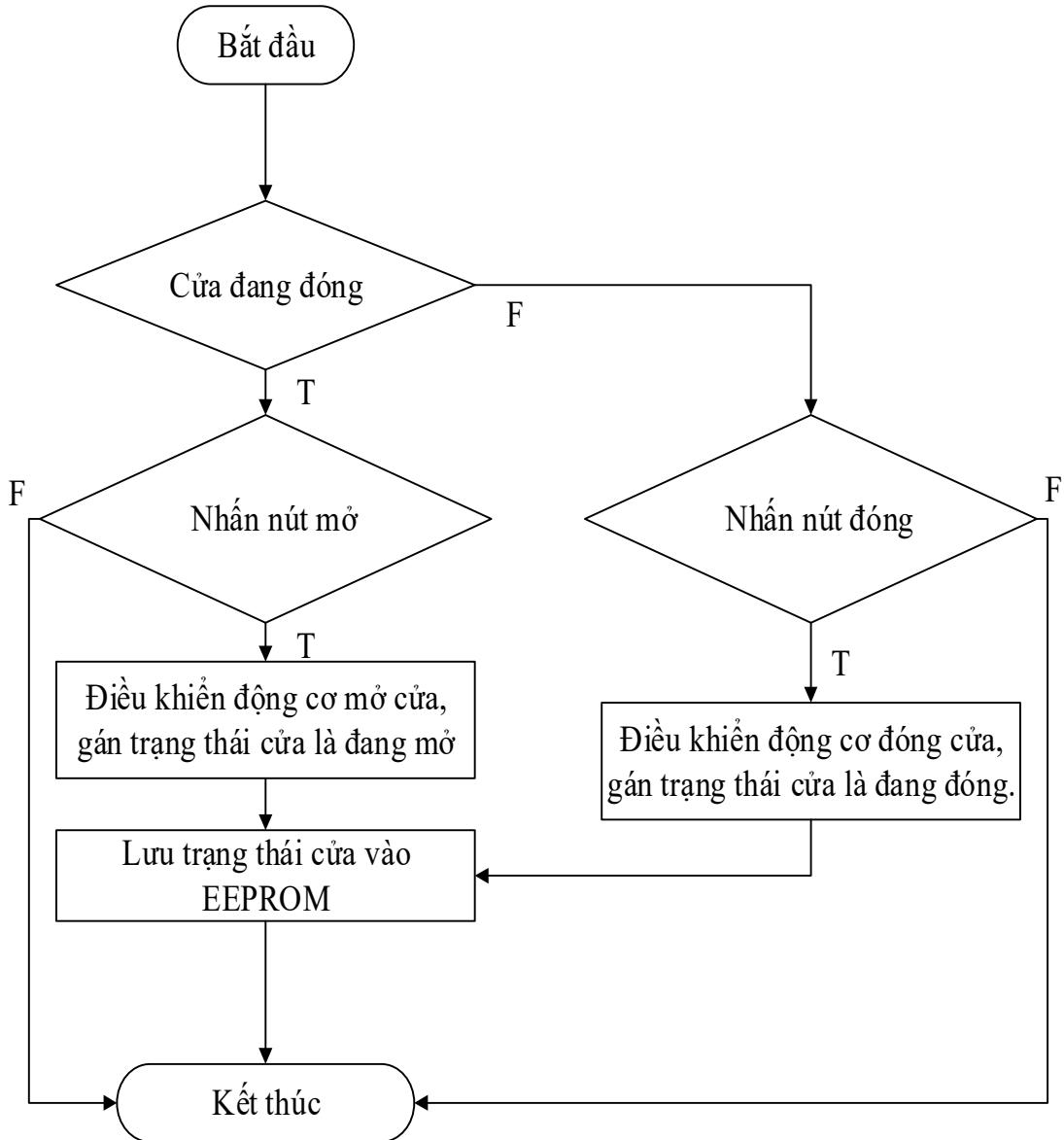
- Giải thích lưu đồ chương trình:

Ban đầu đọc giá trị mật khẩu hiện tại từ Eeprom, đọc giá trị phím nhấn người dùng nhập vào và lưu lần lượt vào mảng tạm. Trong thời gian này ta có thể nhấn nút D để thoát hoặc chương trình tự động thoát khi hết thời gian chờ là 27s. Tiếp tục nếu mật

CHƯƠNG 4: THI CÔNG HỆ THỐNG

khẩu đủ 6 ký tự thì so sánh với mật khẩu hệ thống Eeprom nếu bằng thì tiến hành mở cửa bằng động cơ servo sau đó hiển thị trạng thái ra màn hình LCD, lưu giá trị trạng thái cửa hiện tại vào Eeprom. Nếu mật khẩu đăng nhập không đúng quá 3 lần thì sẽ có chuông báo động và có tin nhắn kèm theo cuộc gọi về cho số điện thoại được cài đặt.

- **Lưu đồ chương trình con “Kiểm tra nút nhấn cửa”:**



Hình 4.22: Lưu đồ chương trình con kiểm tra nút nhấn cửa

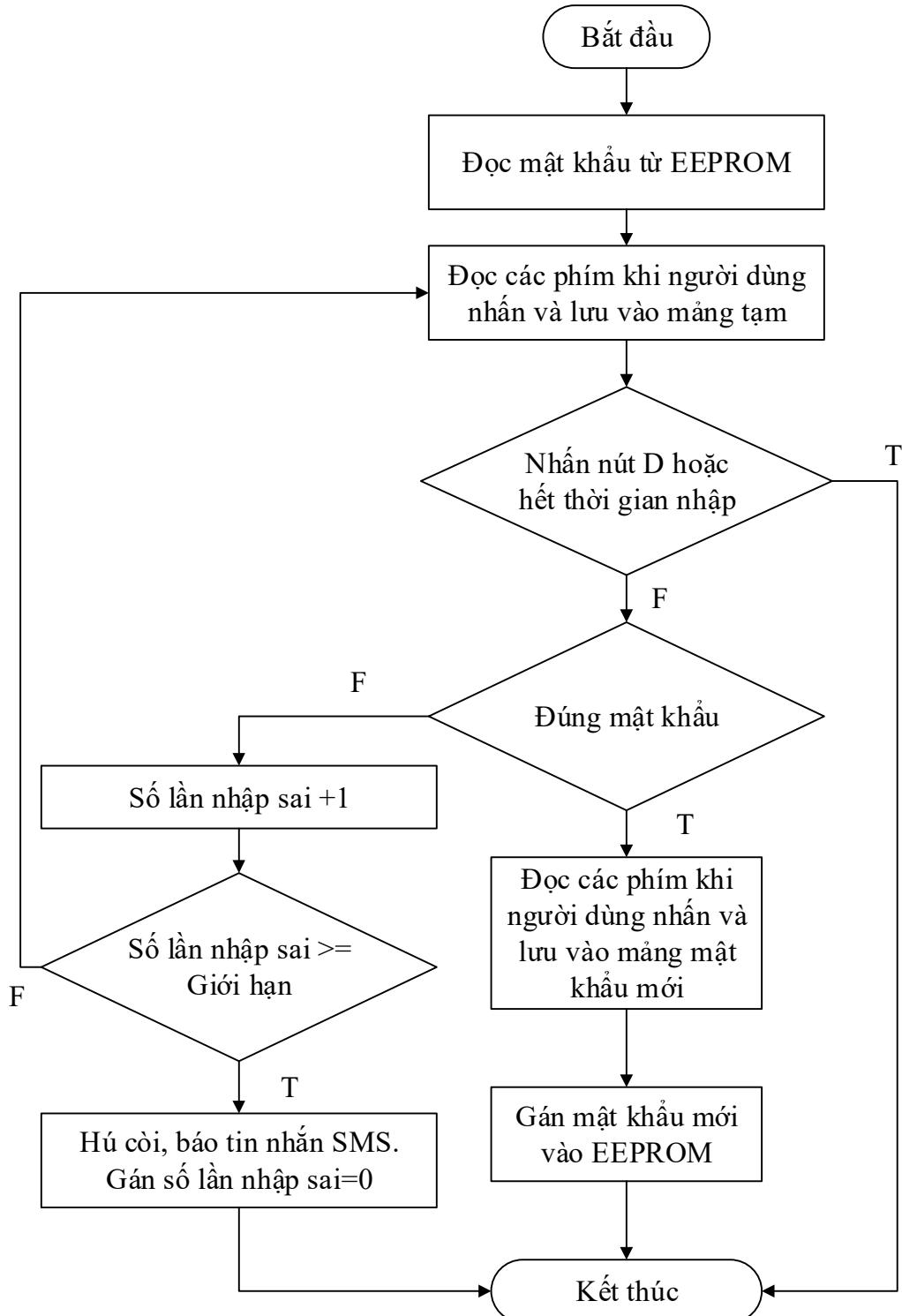
- Giải thích lưu đồ chương trình:

Kiểm tra trạng thái cửa, nếu cửa đang đóng và nhấn nút cửa thì điều khiển động cơ mở cửa và gán trạng thái cửa là đang mở ngược lại cửa đang mở và nhấn nút đóng

CHƯƠNG 4: THI CÔNG HỆ THỐNG

cửa thì điều khiển động cơ đóng cửa và gán trạng thái cửa là đang đóng. Hiển thị ra LCD, đồng thời lưu giá trị trạng thái cửa vào Eeprom.

- **Lưu đồ chương trình con “Đổi mật khẩu”:**



Hình 4.23: Lưu đồ chương trình con đổi mật khẩu

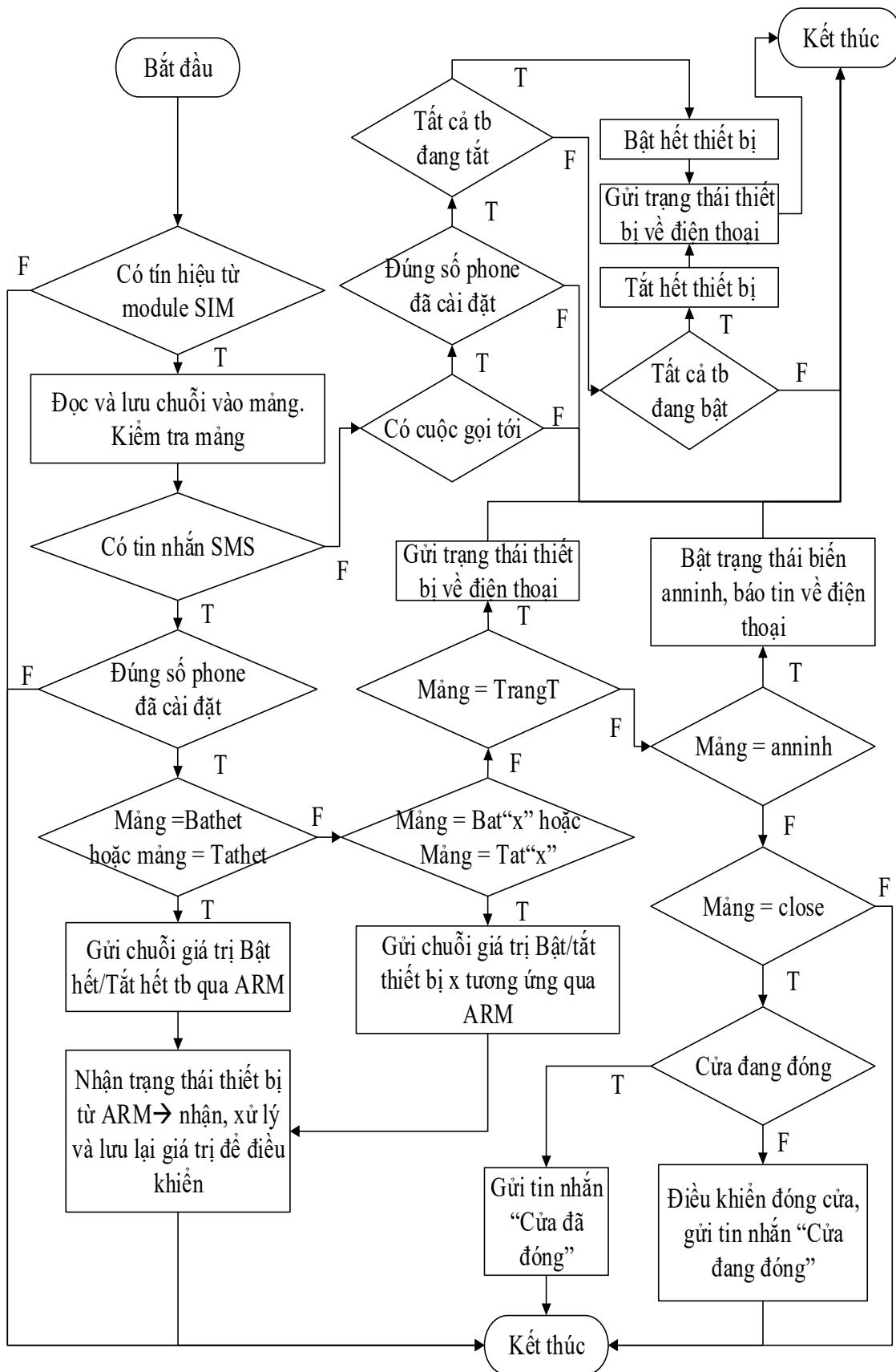
CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Giải thích lưu đồ:

Ban đầu đọc giá trị mật khẩu hiện tại từ Eeprom, đọc giá trị phím nhấn người dùng nhập vào và lưu lần lượt vào mảng tạm. Trong thời gian này ta có thể nhấn nút D để thoát hoặc chương trình tự động thoát khi hết thời gian chờ là 27s. Tiếp tục nếu mật khẩu đủ 6 ký tự thì so sánh với mật khẩu hệ thống Eeprom nếu bằng thì tiến hành nhập mật khẩu mới vào và lưu mật khẩu mới vào lại Eeprom, hiển thị trạng thái ra màn hình LCD. Nếu mật khẩu đăng nhập để kiểm tra mật khẩu cũ không đúng quá 3 lần thì sẽ có chuông báo động và có tin nhắn kèm theo cuộc gọi về cho số điện thoại được cài đặt.

- **Lưu đồ chương trình con “Truyền nhận SIM”:**

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.24: Lưu đồ chương trình con truyền nhận SIM

CHƯƠNG 4: THI CÔNG HỆ THỐNG

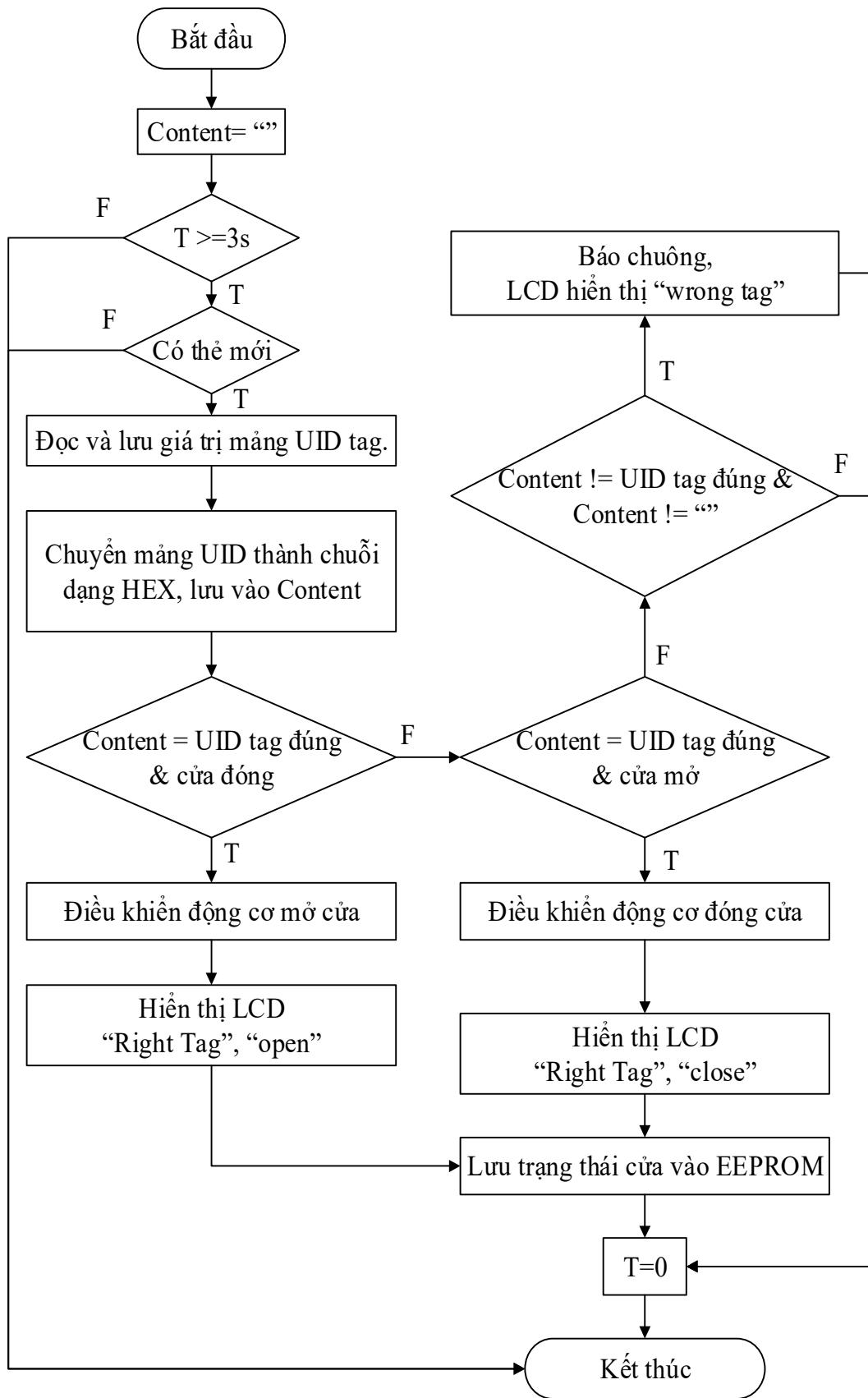
- Giải thích lưu đồ:

Kiểm tra có chuỗi dữ liệu nối tiếp UART từ SIM900 tới MCU, nếu có thì tiến hành lưu tất cả dữ liệu của vào mảng. Kiểm tra xem đó là cuộc gọi đến hay là tin nhắn và số điện thoại có giống như số điện thoại đã cài đặt trước. Kiểm tra các trường hợp:

- Nếu là cuộc gọi đến thì kiểm tra xem trạng thái tắt cả các thiết bị bây giờ nếu đang bật hết thì sẽ tắt hết và ngược lại, cập nhật tắt cả các giá trị trạng thái của thiết bị.
- Nếu kiểm tra mà là tin nhắn đến thì lần lượt kiểm tra xem mảng vừa nhận có chuỗi như là “Bathet”, “Tathet” thì lần lượt bật hết hoặc tắt hết tất cả 8 thiết bị hoặc là chuỗi “Batx”, “Tatx” thì bật và tắt thiết bị tương ứng (x từ [1:8]) đồng thời cập nhật trạng thái thiết bị vào lại để kiểm tra trạng thái thiết bị trong những lần sau.
- Nếu chuỗi là “Trangt” thì gửi trạng thái thiết bị về điện thoại. Nếu chuỗi là “Batan” thì bật hệ thống an ninh của cảm biến PIR lên mức cho phép.
- Nếu chuỗi là “close” thì sẽ kiểm tra trạng thái cửa sẽ đóng cửa nếu cửa đang mở đồng thời gửi tin nhắn cửa đang closing và sẽ gửi closed nếu cửa đã đóng trước đó nhằm đảm bảo an ninh nếu đi xa quên đóng cửa nhà.

Tất cả các chuỗi kiểm tra nếu đúng thì đều có phản hồi thông tin qua SMS đến người dùng.

- ❖ **Lưu đồ chương trình con “đọc thẻ RFID”:**



Hình 4.25: Lưu đồ chương trình con đọc thẻ RFID

CHƯƠNG 4: THI CÔNG HỆ THỐNG

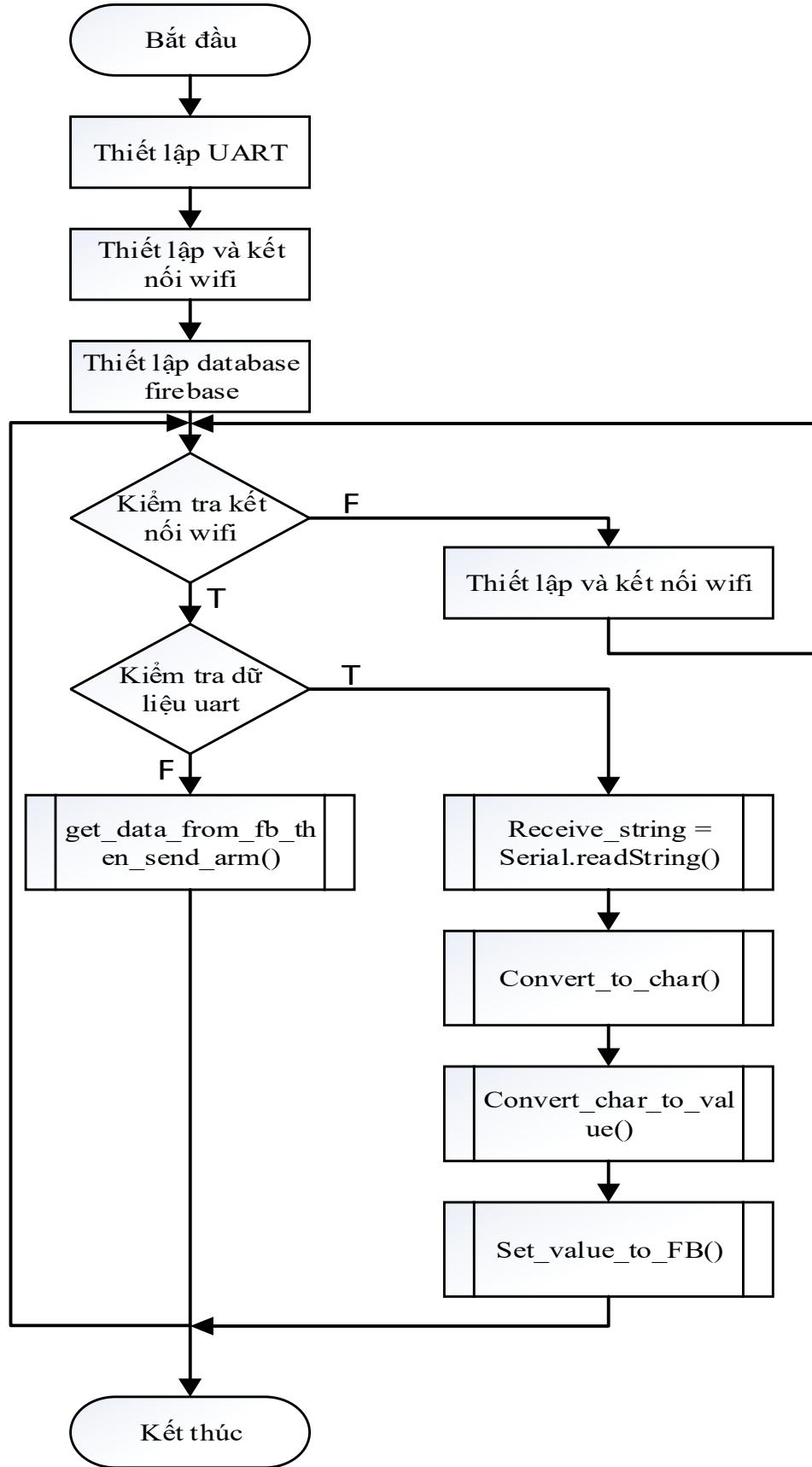
- Giải thích lưu đồ:

Ban đầu kiểm tra thời gian nếu lớn hơn cài đặt giả sử là 3s và mặc định Content = “”. Tiếp đến, nếu kiểm tra có quét thẻ (có thẻ mới) thì tiến hành đọc thẻ và lưu giá trị của mảng UID tag sau đó chuyển mảng UID tag thành kiểu chuỗi ở dạng HEX, lưu vào content và hiển thị giá trị ra Serial port arduino quan sát. Kiểm tra các trường hợp:

- Nếu Content == UID tag mặc định và cửa đang đóng: thì chuông báo, hiển thị open LCD, cửa mở và lưu lại trạng thái cửa vào Eeprom, reset biến đếm thời gian và thoát chương trình.
- Nếu Content == UID tag mặc định và cửa đang mở: thì chuông báo, hiển thị close LCD, cửa đóng và lưu lại trạng thái cửa vào Eeprom, reset biến đếm thời gian và thoát chương trình.
- Nếu Content != UID tag mặc định và Content != “”: thì chuông báo, hiển thị trên LCD : “wrong tag”, reset biến đếm thời gian và thoát chương trình.

c. Lưu đồ chương trình ESP8266

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.26: Lưu đồ chương trình esp8266

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Giải thích lưu đồ:

Bắt đầu, hệ thống sẽ thiết lập UART của Esp8266, thiết lập và kết nối đến điểm wifi chỉ định, thiết lập database firebase.

Tiếp theo, kiểm tra kết nối wifi, nếu có kết nối thì sẽ kiểm tra dữ liệu UART và ngược lại thì sẽ thiết lập và kết nối lại với wifi.

Ở bước kiểm tra dữ liệu UART, khi có dữ liệu thì sẽ bắt đầu gọi hàm *Serial.readString()* để đọc chuỗi nhận được và lưu vào biến *receive_string*. Tiếp theo sẽ gọi hàm *convert_to_char()*, hàm này có chức năng sẽ tách chuỗi vừa nhận được ra thành một mảng gồm những phần tử kiểu char. Tiếp theo là hàm *convert_char_to_value()*, hàm này có chức năng chuyển đổi các biến là kiểu dữ liệu char ở bước trước thành kiểu integer. Sau đó sẽ gửi các giá trị đã được chuyển đổi xong lên database Firebase.

Trong thời gian không có nhận dữ liệu UART từ STM32F407 thì ESP sẽ gọi hàm *get_data_from_fb_then_send_arm()* ra, hàm này có chức năng liên tục đọc giá trị điều khiển thiết từ trên database firebase về và kiểm tra khi nào có sự khác nhau của giá trị trước và sau. Khi có sự khác nhau thì ESP sẽ gửi giá trị đã thay đổi đó qua vi điều khiển để xử lý.

Kết thúc các quá trình trên thì sẽ quay lại bước kiểm tra kết nối wifi và lặp lại.

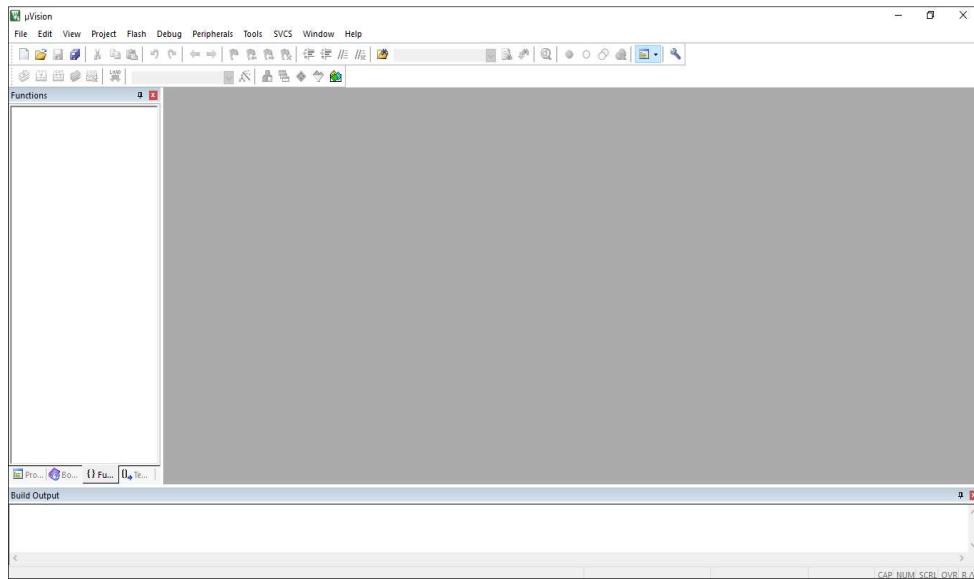
4.4.2 Phần mềm lập trình cho vi điều khiển

a. Phần mềm lập trình KeilC

❖ Giới thiệu

Để lập trình với dòng vi điều khiển STM32 có thể sử dụng IAR hoặc Keil C. Ở đây nhóm chúng tôi sử dụng phần mềm Keil C v5. Keil C v5 được sử dụng để soạn thảo và biên dịch code sang mã hex.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.27: Giao diện phần mềm KeilC

Tiếp theo, để việc lập trình trên dòng STM32 được dễ dàng kiểm soát hơn, nhóm sử dụng thư viện CMSIS của hãng. Thư viện này hỗ trợ các hàm để bạn không phải trực tiếp thao tác với các thanh ghi của Arm. Sử dụng thư viện này sẽ giảm đi những phức tạp khi lập trình và dễ kiểm soát code hơn là việc bạn sử dụng thư viện HAL và phần mềm CubeMX.

Phần mềm nạp chip: STM32 có hỗ trợ 2 chuẩn nạp của nhà sản xuất JTAG, SWD một số loại mạch nạp thông dụng như J-LINK, ST-LINK, U-LINK. Tùy vào loại mạch nạp sử dụng mà có thể nạp chip trực tiếp từ Keil C hoặc phải dùng phần mềm khác. Ở đây nhóm sử dụng mạch nạp ST-LINK.

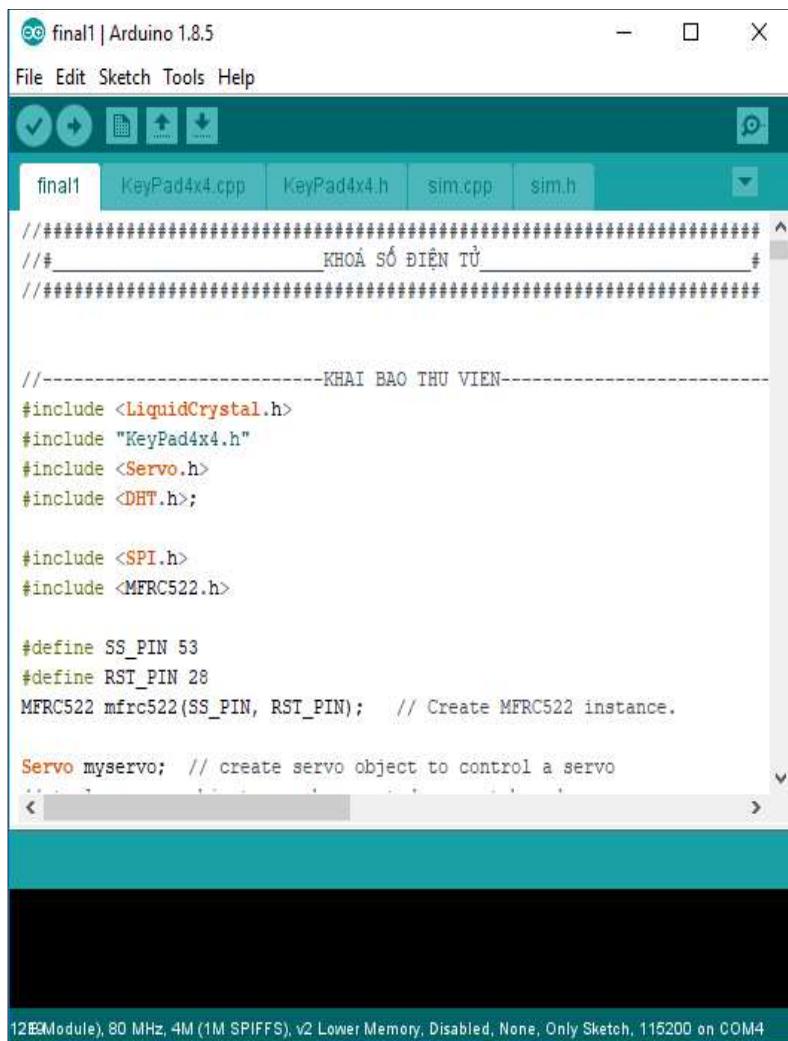
- ❖ **Viết chương trình**
- b. **Giới thiệu phần mềm lập trình Arduino IDE**
- ❖ **Giới thiệu**



Hình 4.28: Logo phần mềm Arduino IDE

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Môi trường phát triển tích hợp (IDE) của Arduino là một ứng dụng đa nền tảng được viết bằng Java, và từ IDE này sẽ được sử dụng cho ngôn ngữ lập trình xử lý và Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Arduino IDE bao gồm một chương trình code editor với các chức năng như đánh dấu cú pháp, và tự động canh lề, cũng như biên dịch và upload chương trình lên board chỉ với 1 cú nhấp chuột. Một chương trình hoặc code viết cho Arduino được gọi là một sketch. Arduino IDE có thể tải về miễn phí tại trang chủ của Arduino với phiên bản mới nhất là 1.8.5 hoặc người dùng có thể dùng trực tiếp phần mềm Arduino IDE trên giao diện web với tên gọi Arduino Web Editor.



Hình 4.29: Giao diện lập trình của phần mềm Arduino IDE

❖ Viết chương trình

4.4.3 Phần mềm lập trình giao diện điều khiển

a. Phần mềm Microsoft Visual Studio

❖ Giới thiệu



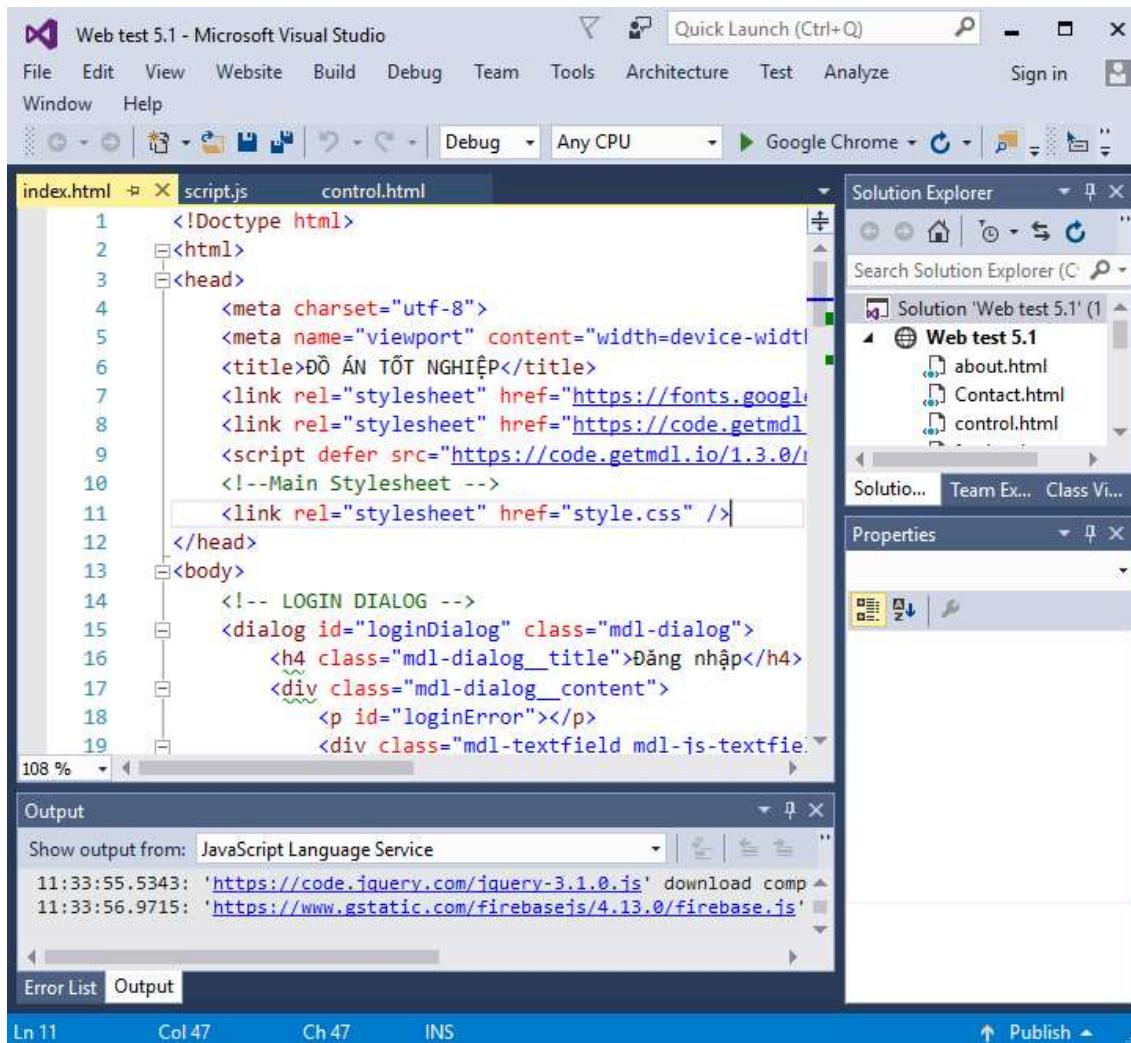
Hình 4.30: Logo phần mềm Visual Studio

Microsoft Visual Studio là môi trường phát triển tích hợp của hãng Microsoft. Nó được sử dụng để phát triển các chương trình máy tính, các trang web, ứng dụng web và các dịch vụ web.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cài tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép biên tập mã và hỗ trợ gỡ lỗi hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ lập trình được tích hợp gồm có C, C++, VB.NET, C#, F#, J++/J#, Python, Ruby và các ngôn ngữ hỗ trợ nền tảng web như XML/XSLT, HTML/XHTML, Javascript và CSS.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.31: Giao diện lập trình web của phần mềm Visual Studio

- ❖ **Viết chương trình**
- b. **Phần mềm Android Studio**
- ❖ **Giới thiệu**

Hiện nay, ứng dụng Android có thể được viết bằng nhiều phần mềm như Visual Studio, Android Studio, MIT AppInventor, Eclipse, ... Trong số đó thì Android Studio là công cụ được Google hỗ trợ mạnh mẽ và nhóm chọn công cụ này để viết chương trình ứng dụng điều khiển.

Android Studio là một phần mềm bao gồm các công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như Smartphone, tablet, ... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để

CHƯƠNG 4: THI CÔNG HỆ THỐNG

giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển ứng dụng từ đơn giản đến phức tạp.



Hình 4.32: Giao diện phần mềm Android studio

❖ Viết chương trình

4.5 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

4.5.1 Viết tài liệu hướng dẫn sử dụng

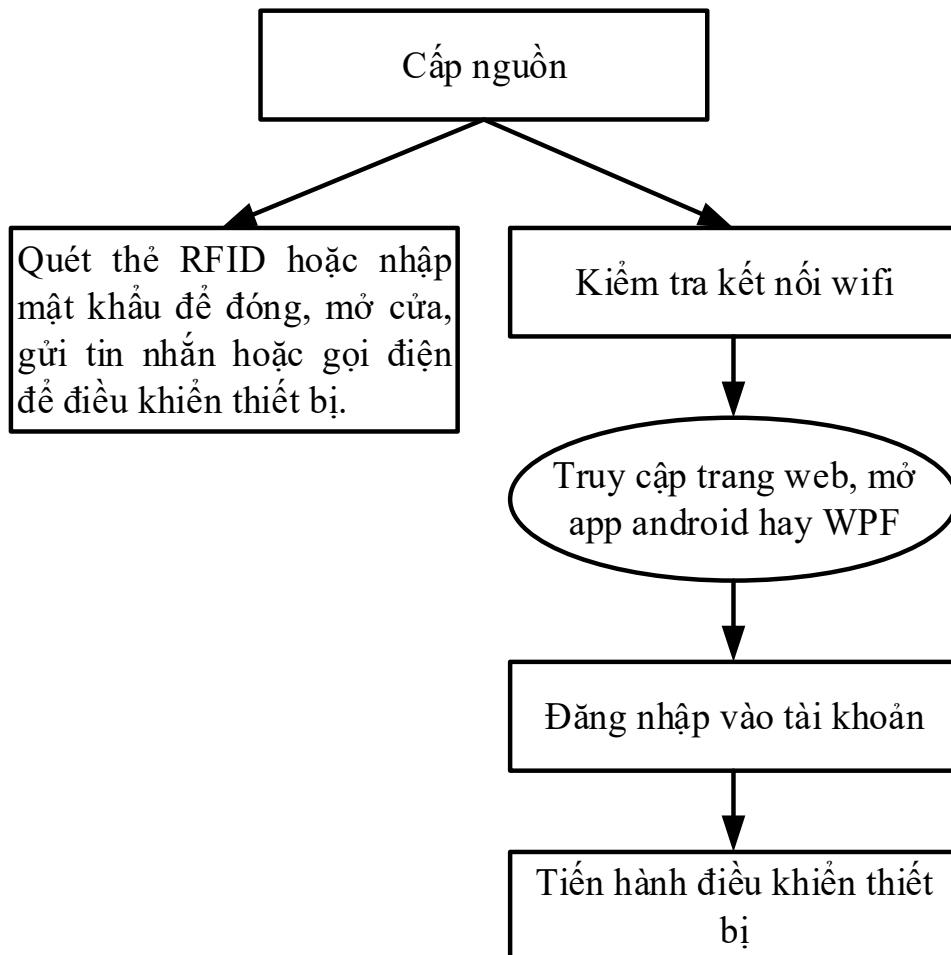
Bước 1: Cáp nguồn cho hệ thống, hệ thống sử dụng nguồn 220VAC để cấp cho ngõ ra công suất, mạch điều khiển cung cấp 220VAC qua nguồn tổ ong để cấp cho mạch.

Bước 2: Chờ cho ESP8266 kết nối wifi và bắt đầu tiến hành đăng nhập vào trang web: iot-datn.firebaseio.com, app Android hay WPF. Thực hiện đăng nhập để có thể điều khiển và giám sát các cảm biến, các thiết bị, cửa đang mở hay đóng.

Trên web để khiều khiển và giám sát thiết bị vào mục điều khiển, để giám sát cảm biến ta vào mục giám sát. Trên app android và WPF thì sau khi mở giao diện sẽ thấy cập nhật các giá trị cảm biến và trạng thái thiết bị, đồng thời có các nút điều khiển thiết bị. Gửi tin nhắn hoặc gọi để điều khiển thiết bị.

Quét thẻ đóng mở cửa dùng module RFID hay nhập mật khẩu bằng bàn phím. Quét thẻ đúng sẽ hiện "...RIGHT TAGS..." lên LCD và đóng mở cửa bằng cách quay servo. Thẻ sai (trộm) quét sẽ hiện "...WRONG TAGS..." và kêu buzzer.

4.5.2 Quy trình thao tác



Hình 4.33: Lưu đồ quy trình thao tác

❖ Giải thích:

Ban đầu sẽ cấp nguồn cho toàn bộ hệ thống

- Hệ thống mở cửa:

Sau khi cấp nguồn, quét thẻ ngay đầu đọc thẻ được bố trí ngay bên cạnh cửa hay nhập mật khẩu để mở cửa vào nhà, khi vào nhà sẽ có 1 nút bấm để đóng cửa.

Đầu tiên màn hình LCD 16x2 hiển thị.

A: LOGIN	B: CLOSE
C: CHANGE PASS [0→9]	

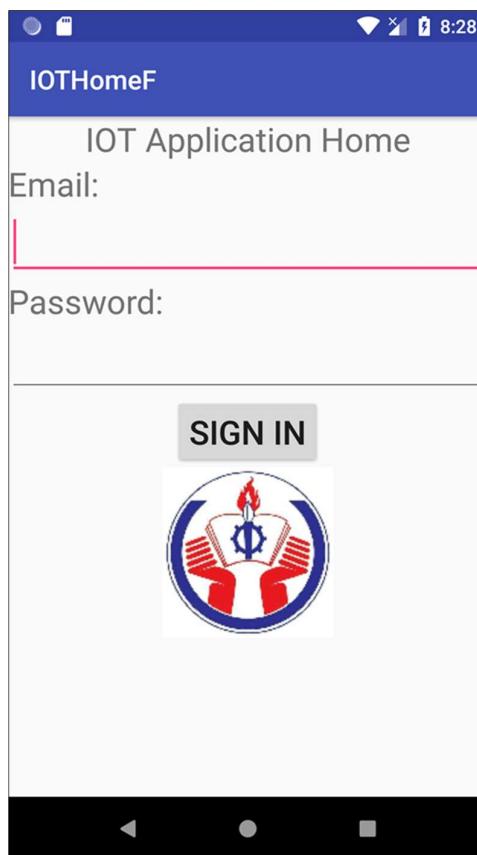
Nếu nhấn phím A thì sẽ đăng nhập vào, nếu cửa đang mở thì LCD hiển thị “CUA DANG MO”—“NHAN NUT DE DONG”. Khi vào đăng nhập thì sẽ nhập mật khẩu đã được cài đặt nếu đăng nhập đúng thì LCD sẽ thông báo “~~DANG MO CUA~~” và cửa

CHƯƠNG 4: THI CÔNG HỆ THỐNG

sẽ mở ra. Nếu đăng nhập mật khẩu sai quá 3 lần thì chuông kêu 1 hồi bíp và có tin nhắn báo về điện thoại đồng thời có cuộc gọi. Nhấn B thì sẽ đóng cửa nếu cửa đang mở. Nhấn C để thay đổi mật khẩu, LCD khi đó hiển thị “*: NO #: YES”. Nếu nhấn '#' thì tiến hành đổi, nhấn '*' thì thoát ra màn hình chính. Khi vào đổi mật khẩu thì đầu tiên chúng ta đăng nhập lại mật khẩu cũ, khi mật khẩu cũ đúng thì sẽ tiến hành nhập lại mật khẩu mới. Thê là đổi được mật khẩu.

- App android:

Ứng dụng Android FireSmartF có giao diện điều khiển như sau:



Hình 4.34: Giao diện app android

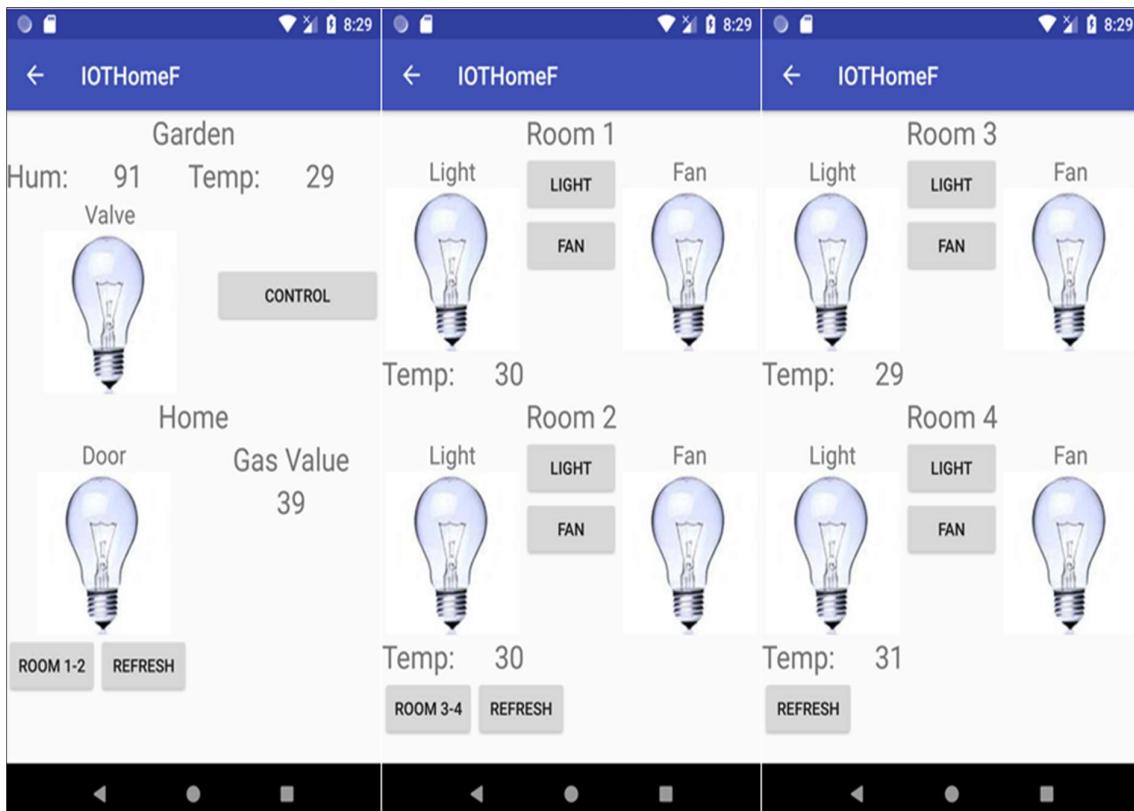
Giao diện khi khởi động ứng dụng:

- Ứng dụng sẽ kiểm tra kết nối internet (Wifi/3G/4G) nếu không có thì ứng dụng sẽ yêu cầu kết nối hoặc thoát ứng dụng.
- Hai trường Edit Text của giao diện: dùng để nhập email và mật khẩu được cung cấp bởi admin.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Khi nhấn nút “SIGN IN” nếu tài khoản hợp lệ thì sẽ chuyển qua giao diện tiếp theo. Ngược lại, người dùng sẽ được thông báo liên hệ với admin để được cấp tài khoản.

Các giao diện sau khi đăng nhập thành công:



Hình 4.35: Các giao diện app android sau khi đăng nhập

Đầu tiên, ở giao diện bên trái:

- Hiển thị giá trị nhiệt độ, độ ẩm của vườn.
- Hiển thị trạng thái đóng/mở của valve nước.
- Nút nhấn “CONTROL” điều khiển valve nước.
- Nút Refresh để cập nhật lại các giá trị mới và bắt đầu điều khiển thiết bị.
- Hiển thị giá trị Gas trong nhà.
- Hiển thị trạng thái đóng/mở cửa.
- Nút nhấn “ROOM 1-2” sẽ đưa người dùng chuyển qua giao diện tiếp theo.

Tiếp theo, ở giao diện giữa:

- Hiển thị giá trị nhiệt độ phòng 1 và phòng 2.
- Hiển thị trạng thái thiết bị đèn và quạt của cả hai phòng 1 và 2.
- Các nút nhấn để điều khiển việc đóng/mở của các thiết bị.

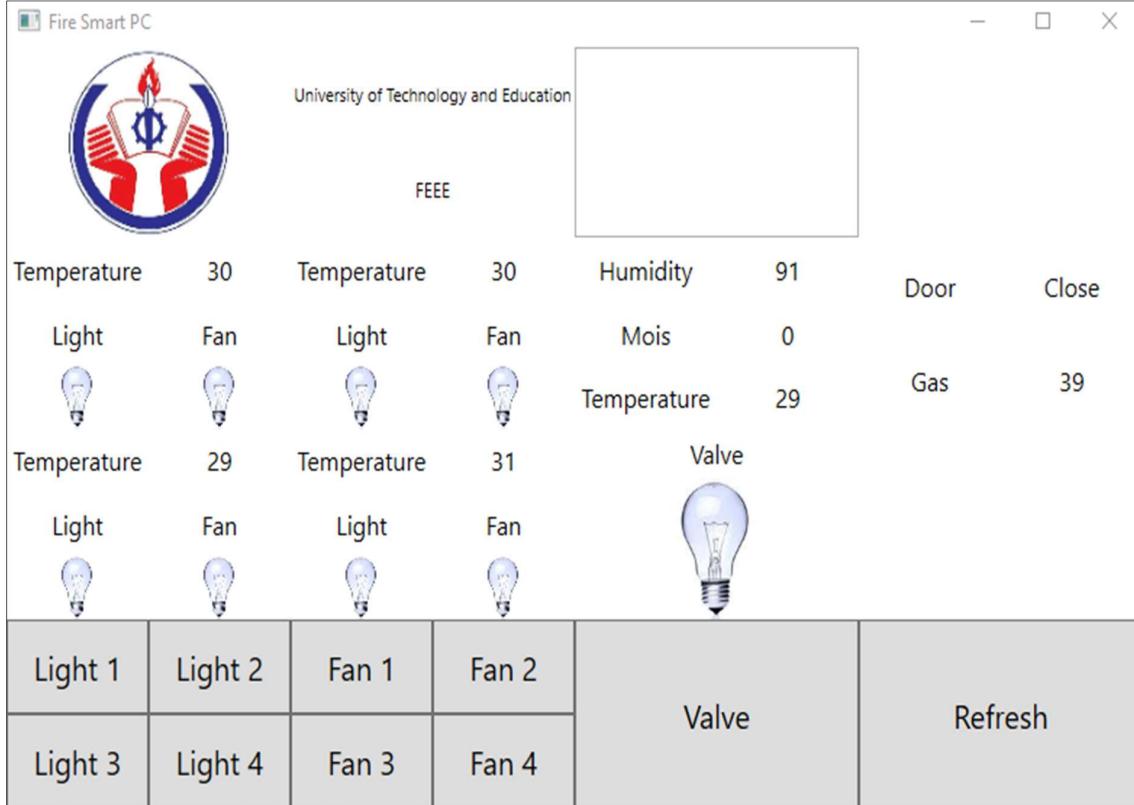
CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Nút nhấn “ROOM 3-4” sẽ đưa người dùng chuyển qua giao diện tiếp theo.

Cuối cùng, ở giao diện bên phải:

- Hiển thị giá trị nhiệt độ phòng 3 và phòng 4.
- Hiển thị trạng thái thiết bị đèn và quạt của cả hai phòng 3 và 4.
- Các nút nhấn để điều khiển việc đóng/mở của các thiết bị.
- Ngoài ra, nút mũi tên phía trên cùng bên trái sẽ đưa người dùng trở về giao diện trước đó.
- WPF (Windows Presentation Foundation)

Giao diện ứng dụng:



Hình 4.36: Giao diện WPF

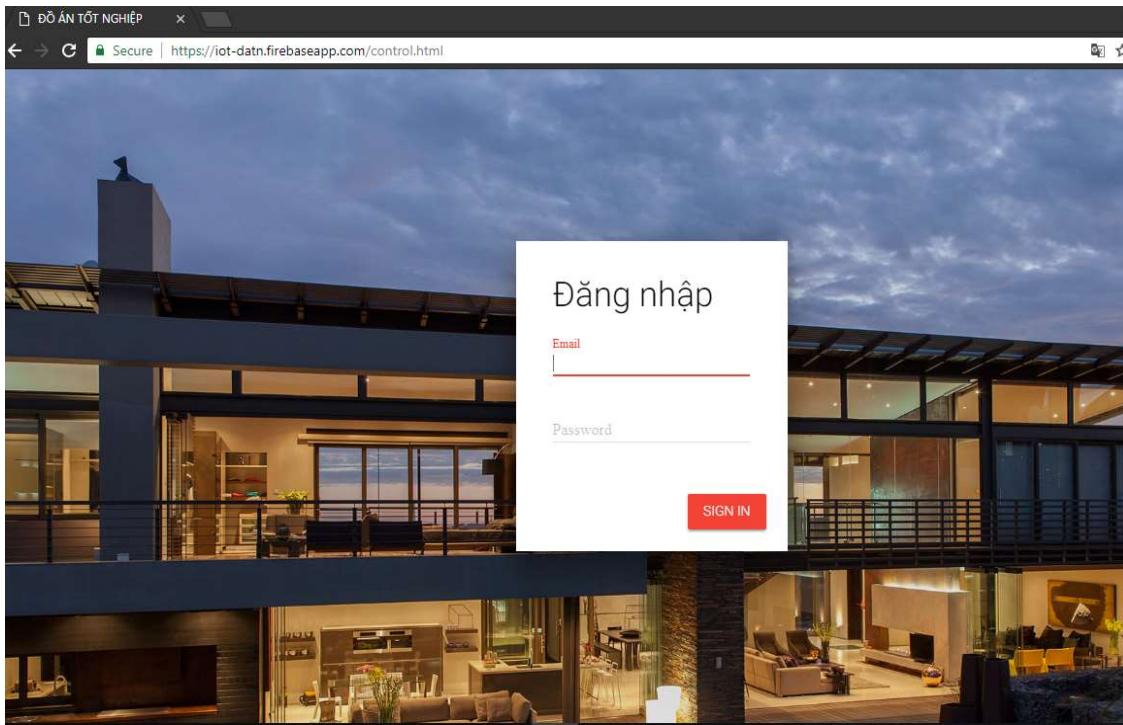
Ứng dụng trên máy tính chạy hệ điều hành Windows này có tên là Fire Smart PC gồm có các tính năng:

- Hiển thị tất cả trạng thái thiết bị.
- Hiển thị nhiệt độ, độ ẩm, khí gas từ các cảm biến.
- Các nút nhấn dùng để điều khiển các thiết bị chỉ định.
- Nút Refresh để cập nhật lại các giá trị mới và bắt đầu điều khiển thiết bị.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Khi người dùng muốn điều khiển thiết bị, người dùng chỉ cần nhấn vào các nút nhấn.
- Web server

Giao diện của web khi lần đầu truy cập:



Hình 4.37: Giao diện lần đầu đăng nhập web

Trong giao diện đăng nhập có 2 trường text để người dùng nhập vào là Email và Password. Đây là tài khoản được cung cấp bởi admin quản trị hệ thống. Nếu người dùng nhập sai một trong 2 trường text thì website sẽ báo lỗi đăng nhập và cần nhập lại tài khoản. Ngược lại nếu người dùng nhập đúng tài khoản thì website sẽ chuyển đến trang chủ của website.

Giao diện sau khi đăng nhập thành công:

CHƯƠNG 4: THI CÔNG HỆ THỐNG

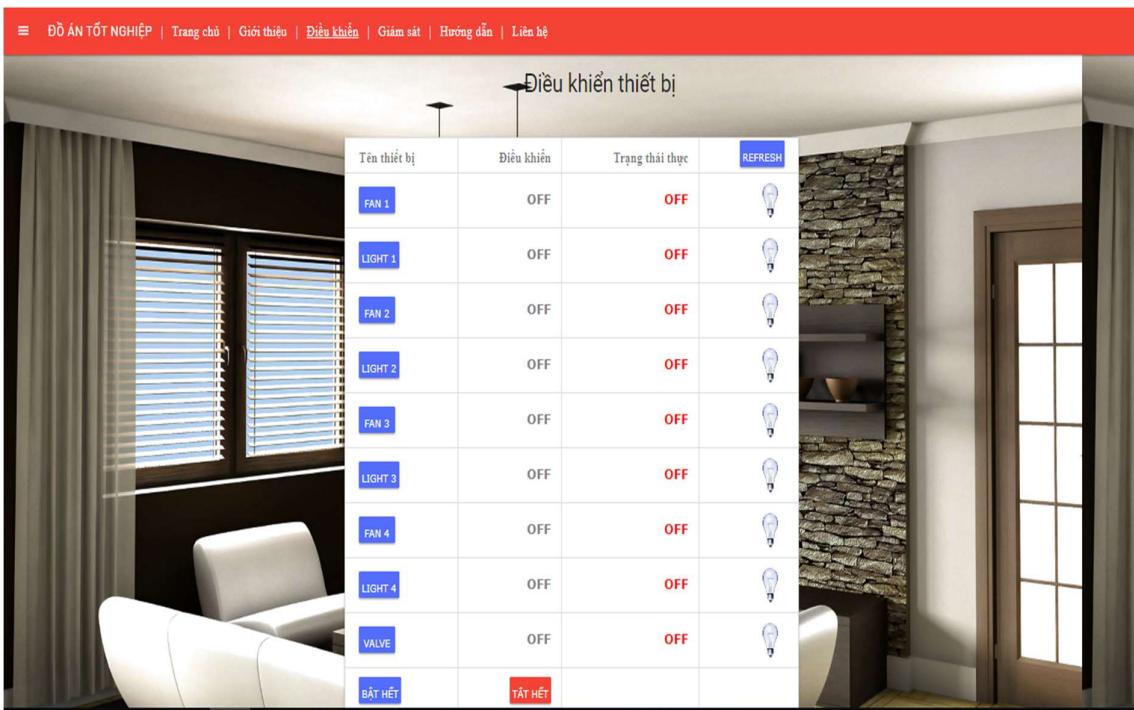
The screenshot shows the homepage of the HCMUTE website. At the top left is the university's logo and name: "TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH" and "HCMC University of Technology and Education". Below the logo is the acronym "HCMUTE". A red navigation bar contains the text "ĐỒ ÁN TỐT NGHIỆP" and links to "Trang chủ", "Giới thiệu", "Điều khiển", "Giám sát", "Hướng dẫn", and "Liên hệ". The main title "ĐỒ ÁN TỐT NGHIỆP" is centered above the subtitle "Đề tài: Hệ thống IoT điều khiển và giám sát ngôi nhà". Below the subtitle, it says "GVHD: Th.s NGUYỄN ĐÌNH PHÚ". To the right, there is a list of names under "SVTH": Trần Nhật Minh, Huỳnh Xuân Dũng, Nguyễn Quốc Phát, Nguyễn Bá Linh. At the bottom of the page, a dark footer bar includes the text "Copyright © nguoicuat@gmail.com Help Contact".

Hình 4.38: Giao diện web sau khi đăng nhập thành công

Theo bố cục của trang web trên thì ở vị trí trên cùng là hình logo của trường Đại Học Sư Phạm Kỹ Thuật TP.HCM, bên dưới là menu hàng ngang của website có chứa các link dẫn đến các trang nội bộ khác như trang “Giới thiệu”, “Điều khiển”, “Giám sát”, “Hướng dẫn” và “Liên hệ”. Khi người dùng nhấp vào nút có hình 3 dấu gạch ngang ở menu thì bên trái website cũng hiện ra menu theo kiểu hàng dọc. Ở giữa website là vị trí hiển thị nội dung chính của trang đó, ở dưới cùng là vị trí hiển thị footer của website. Đây là giao diện bố cục trang chủ và cũng là bố cục chung của website.

Giao diện trang “Điều khiển”:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



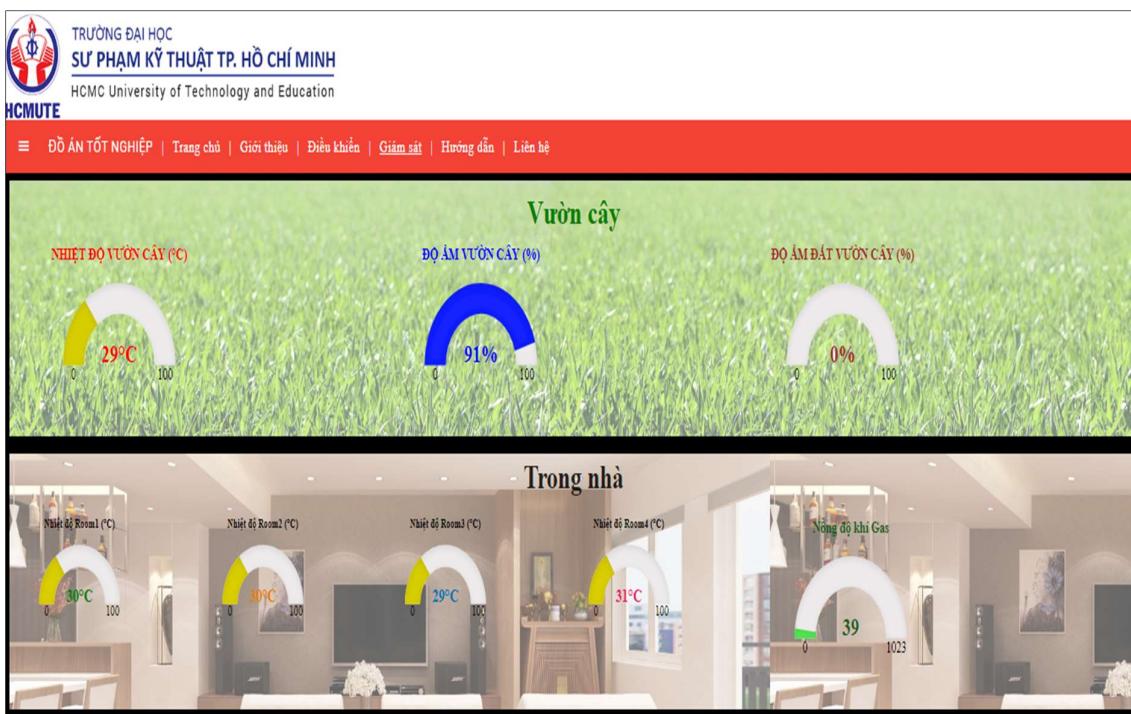
Hình 4.39: Giao diện trang điều khiển

Trang điều khiển thiết bị có nội dung là một bảng gồm có 4 cột:

- Cột đầu tiên là 9 nút nhấn cho 9 thiết bị trong hệ thống, mỗi khi người dùng nhấn nút thì trạng thái điều khiển thiết bị sẽ chuyển đổi.
- Cột thứ hai hiển thị trạng thái người dùng muốn điều khiển.
- Cột thứ ba hiển thị trạng thái thực của thiết bị trong hệ thống.
- Cột thứ 4 hiển thị các hình ảnh bóng đèn minh họa cho trạng thái thiết bị.
- Nút Refresh để cập nhật lại các giá trị mới và bắt đầu điều khiển thiết bị.
- Hàng cuối cùng là 2 nút nhấn để điều khiển bật hết hoặc tắt thiết bị cùng lúc.

Giao diện trang “Giám sát”:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.40: Giao diện trang giám sát

Nội dung trang “Giám sát” chia làm 2 phần:

- Phần trên hiển thị các thông số nhiệt độ, độ ẩm không khí, độ ẩm đất của vườn cây.
- Phần dưới hiển thị các thông số nhiệt độ của 4 phòng và nồng độ khí gas.
- Sử dụng module sim900a điều khiển và quản lý thiết bị:

Module sim900A hoạt động được theo tập lệnh AT. Ban đầu khởi động SIM khoảng 15s để SIM hoạt động ổn định. Sau đó khi có tín hiệu tới module sim qua UART thì tiến hành xử lý và lưu giá trị. Nếu là cuộc gọi đến và các thiết bị đang bật hết thì sẽ tắt hết và ngược lại (*lưu ý là các thiết bị đang ở trạng thái bật hết hoặc tắt hết*). Nếu là 1 tin nhắn đến thì tiến hành so sánh giá trị “Bathet”, “Tathet”, “Batx”, “Tatx” (với x là thiết bị từ 1→8) thì tiến hành bật tắt thiết bị tương ứng. Nếu tin nhắn là “Trangt” thì sẽ kiểm tra trạng thái thiết bị hiện tại để gửi tin nhắn. Nếu tin nhắn là “Batan” thì bật chế độ an ninh cho phép cảm biến PIR hoạt động. Nếu tin nhắn là “close” thì kiểm tra trạng thái cửa và tiến hành đóng cửa nếu như cửa mở trước đó và gửi thông báo về điện thoại chủ. Nếu có trộm hoặc khí gas trong nhà tăng cao thì SIM cũng sẽ gửi tin nhắn và cuộc gọi để cảnh báo cho chủ nhà biết.

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 thời gian 15 tuần. Bên cạnh đó là nhận xét, đánh giá và đề xuất hướng phát triển của sản phẩm mô hình để hoàn thiện và có thể đi vào thực tế.

5.2 KẾT QUẢ ĐẠT ĐƯỢC

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm nhận thấy được đề tài có khả năng ứng dụng vào thực tiễn cao, đáp ứng được nhu cầu cuộc sống hiện đại ngày nay và đi đúng xu hướng của thời đại. Đồng thời đề tài này cũng là một nguồn tài liệu có giá trị cho các bạn sinh viên những khóa tiếp theo có thể tham khảo khi nghiên cứu những đề tài có liên quan. Bên cạnh đó, nhóm cũng bổ sung cho mình những kiến thức hay và bổ ích.

5.2.1 Biết cách sử dụng các cảm biến

Các cảm biến được sử dụng trong đề tài này (DHT11, MQ2, DS18B20, độ ẩm đất, cảm biến thân nhiệt chuyển động) là những cảm biến cần thiết trong cuộc sống. Có thể nói, các cảm biến ngày nay được sản xuất ngày một đơn giản và nhỏ gọn hơn, nhằm đáp ứng nhu cầu của người tiêu dùng. Quá trình nghiên cứu đề tài này đã giúp nhóm thực hiện nâng cao kỹ năng đọc – hiểu cảm biến, trau dồi thêm kiến thức về các loại cảm biến. Đặc biệt là, tăng khả năng vận dụng lý thuyết vào thực tế, có khả năng lựa chọn được loại cảm biến phù hợp với từng yêu cầu của thực tiễn đời sống.

5.2.2 Biết cách lập trình STM32F407

STM32F407 là vi điều khiển được sử dụng rộng rãi hiện nay, nó có rất nhiều chức năng. Tuy nhiên, để hiểu về các chức năng của nó và lập trình thành thạo trên board STM32F407 là điều không dễ dàng. Và qua quá trình tìm hiểu và nghiên cứu đề tài thì nhóm đã biết được cách lập trình cho vi điều khiển này, nó có khả năng đọc các cảm biến cụ thể trong đề tài này nhóm thực hiện đọc 4 con cảm biến ds18b20, sử dụng chuẩn giao tiếp I2C để kết nối với ds1307 và màn hình Oled.

Từ đó làm nền tảng để nhóm có thể phát triển ý tưởng điều khiển được mọi thứ xung quanh, biến chúng trở nên thông minh hơn.

5.2.3 Biết cách lập trình Arduino Mega

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Arduino Mega là một trong những board mạch thông minh được sử dụng rộng rãi. Tuy nhiên, để sử dụng thành thạo và đầy đủ các chức năng của nó thì không phải dễ. Và đó là kết quả của quá trình nghiên cứu đề tài này mà nhóm thực hiện có được. Nó có khả năng đọc giá trị của các loại cảm biến khác nhau. Trong đề tài này, nhóm thực hiện có khả năng hiển thị thông tin theo yêu cầu lên LCD 16x2, truyền – nhận dữ liệu qua uart để điều khiển qua sim.

5.2.4 Biết cách sử dụng module wifi ESP8266

ESP8266 đang được sử dụng rất phổ biến trong các thiết bị giao tiếp với smartphone hay webserver thông qua Wifi nhờ giá thành rẻ, module nhỏ gọn và đặc biệt dễ sử dụng. Đây là công cụ cải tiến quá trình giám sát từ không dây sang có dây, giúp ta hội nhập vào cuộc sống hiện đại ngày nay. Qua quá trình nghiên cứu, nhóm đã biết cách sử dụng module này để giao tiếp truyền nhận dữ liệu với webserver, app android và WPF.

5.2.5 Biết cách viết app android, web và WPF

Sau thời gian nghiên cứu, nhóm đã biết cách viết ứng dụng trên Android, trang web server và WPF để điều khiển, giám sát thiết bị qua wifi. App android được viết bằng phần mềm Android studio. Còn web server và WPF được thiết kế bằng phần mềm Visual studio và sau đó được đưa lên firebase.

5.2.6 Biết cách lập trình đóng mở cửa bằng RFID

Hệ thống RFID được ứng dụng rất rộng rãi vào rất nhiều lĩnh vực, đời sống như: kiểm soát xe cộ, thu phí đường bộ, kiểm soát hành lý hành khách, quản lý nhân sự và chấm công nhân viên, có thể thay thế công nghệ mã vạch, giúp lưu giữ nhiều thông tin về sản phẩm hơn. Khi đưa thẻ RFID vào vùng hoạt động của đầu đọc RFID, sóng vô tuyến phát ra từ đầu đọc sẽ cung cấp cho thẻ RFID một dòng điện đủ nhỏ để kích hoạt hệ thống mạch điện nằm trong thẻ giúp nó gửi lại tín hiệu hồi đáp và thực hiện trao đổi dữ liệu theo yêu cầu của bộ điều khiển kết nối với đầu đọc RFID. Sau khi nhận được dữ liệu từ thẻ bộ điều khiển sẽ đưa ra yêu cầu điều khiển tùy vào từng ứng dụng cụ thể.

5.2.7 Biết cách truyền nhận dữ liệu giữa các vi điều khiển

Qua đề tài này nhóm đã biết cách truyền nhận dữ liệu giữa các vi điều khiển, dùng Arduino để đọc các cảm biến hay đóng mở cửa rồi gửi dữ liệu qua và nhận hồi đáp từ STM32F407. Truyền nhận dữ liệu giữa STM32F407 với Esp8266 qua giao tiếp UART.

5.2.8 Biết cách truyền nhận dữ liệu qua wifi

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Qua đề tài này, nhóm đã biết giám sát – điều khiển thiết bị qua wifi và tạo lập trang web, app android hay WPF riêng cho hệ thống. Cả ba phần mềm điều khiển này đều được thông qua firebase để điều khiển và giám sát.

5.3 KẾT QUẢ THỰC NGHIỆM

Cấp nguồn cho hệ thống: 2 dây nguồn AC 220V.

5.3.1 Quét thẻ RFID hoặc nhập mật khẩu đóng mở cửa

a. Quét thẻ đúng



Hình 5.1: Quét thẻ đúng và cửa mở ra, chuông kêu

Khi người dùng quét thẻ đúng với thẻ đã được lưu thì cửa sẽ mở đồng thời chuông kêu 1 tiếng bíp và LCD sẽ hiển thị “...RIGHT TAGS...^open^”. Sau khi vào nhà sẽ có 1 nút nhấn để đóng cửa.



Hình 5.2: Quét thẻ đúng và cửa đóng lại

Khi người dùng quét thẻ đúng thêm 1 lần nữa cửa sẽ đóng đồng thời LCD sẽ hiển thị "...RIGHT TAGS... ^^^close^'^^".

b. Quét thẻ sai



Hình 5.3: Quét thẻ sai, cửa không mở, chuông kêu

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Nếu quét thẻ sai, chuông kêu 1 hồi bíp để cảnh báo cho người dùng biết. LCD hiển thị “...WRONG TAGS...”

c. Nhập đúng mật khẩu



Hình 5.4: Nhập đúng mật khẩu và cửa mở ra

Khi nhập mật khẩu, nếu đúng mật khẩu là “111111” thì cử sẽ được mở ra và màn hình LCD hiển thị “MAT KHAU DUNG →→DANG MO CUA→→”

d. Nhập sai mật khẩu

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ



Hình 5.5: Nhập sai mật khẩu, cửa không mở, chuông kêu

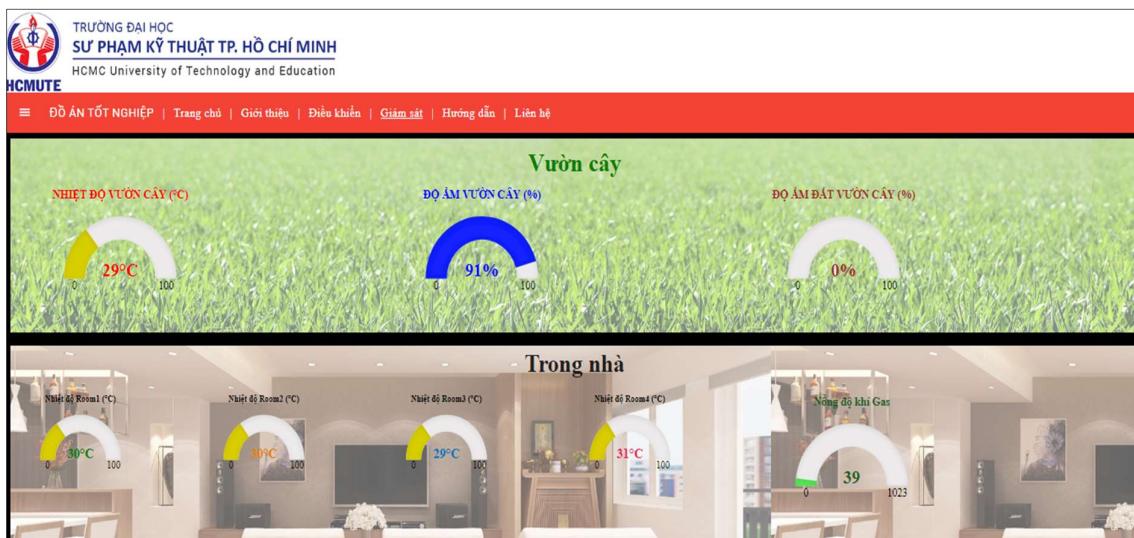
Khi nhập sai mật khẩu, cửa sẽ không mở và chuông kêu 1 tiếng bíp, nếu nhập sai mật khẩu 3 lần thì chuông sẽ kêu 1 thời gian dài.

5.3.2 Điều khiển và giám sát thiết bị, cảm biến thông qua internet

a. Web server

❖ Giám sát

Vào giao diện web và đăng nhập, nhấp vào mục “giám sát”



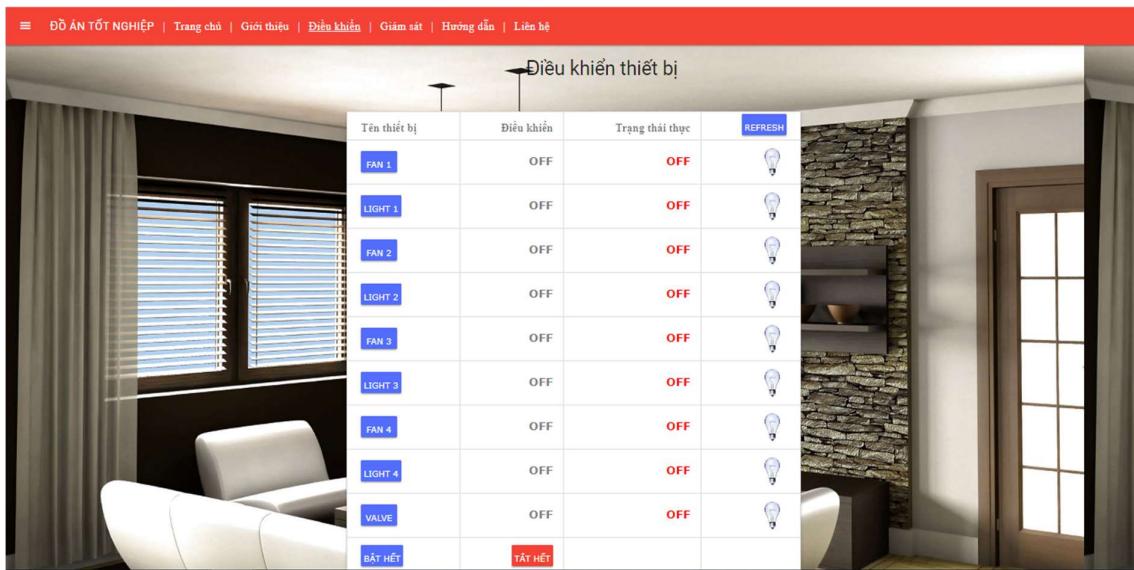
Hình 5.6: Giao diện trang giám sát

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau khi vào mục “giám sát” ta thấy các thông số như: nhiệt độ, độ ẩm vườn cây, độ ẩm đất vườn cây, nhiệt độ của 4 phòng và nồng độ khí gas. Các thông số này sau khi được đọc từ vi điều khiển sẽ xử lí và gửi qua Esp để đưa dữ liệu lên web. Cứ 10s sẽ cập nhật lại các thông số.

❖ Điều khiển

Sau khi nhấp vào mục “điều khiển”



Hình 5.7: Tất cả thiết bị đều tắt khi chưa điều khiển

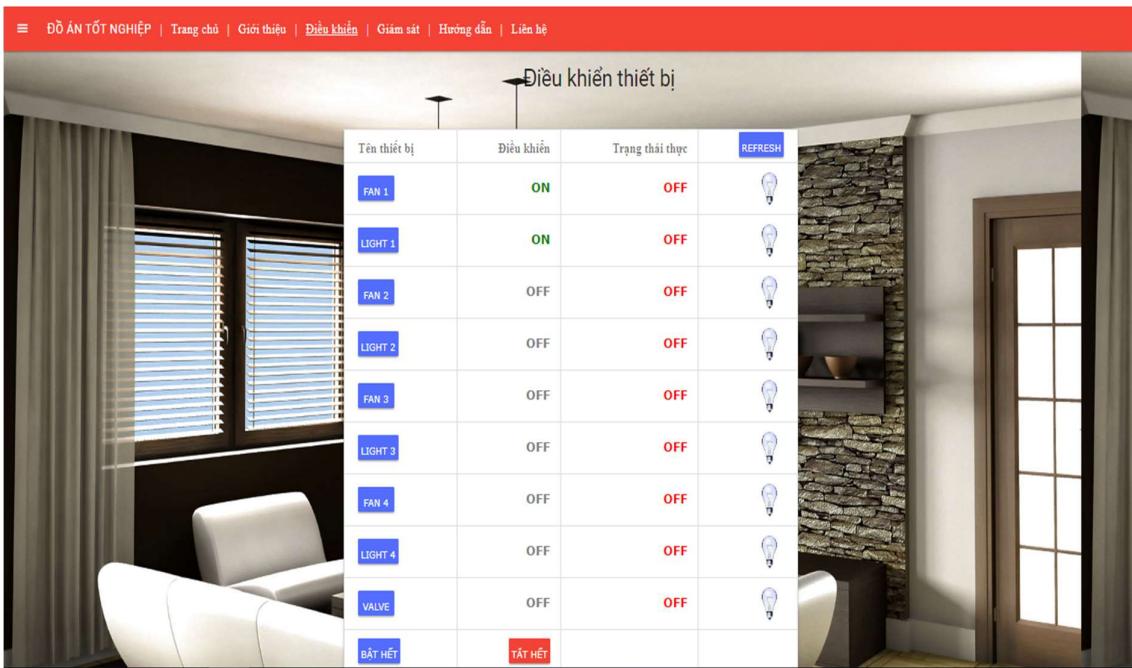
Ban đầu, khi chưa được điều khiển, trạng thái các thiết bị đều tắt.



Hình 5.8: Tất cả các thiết bị đều tắt

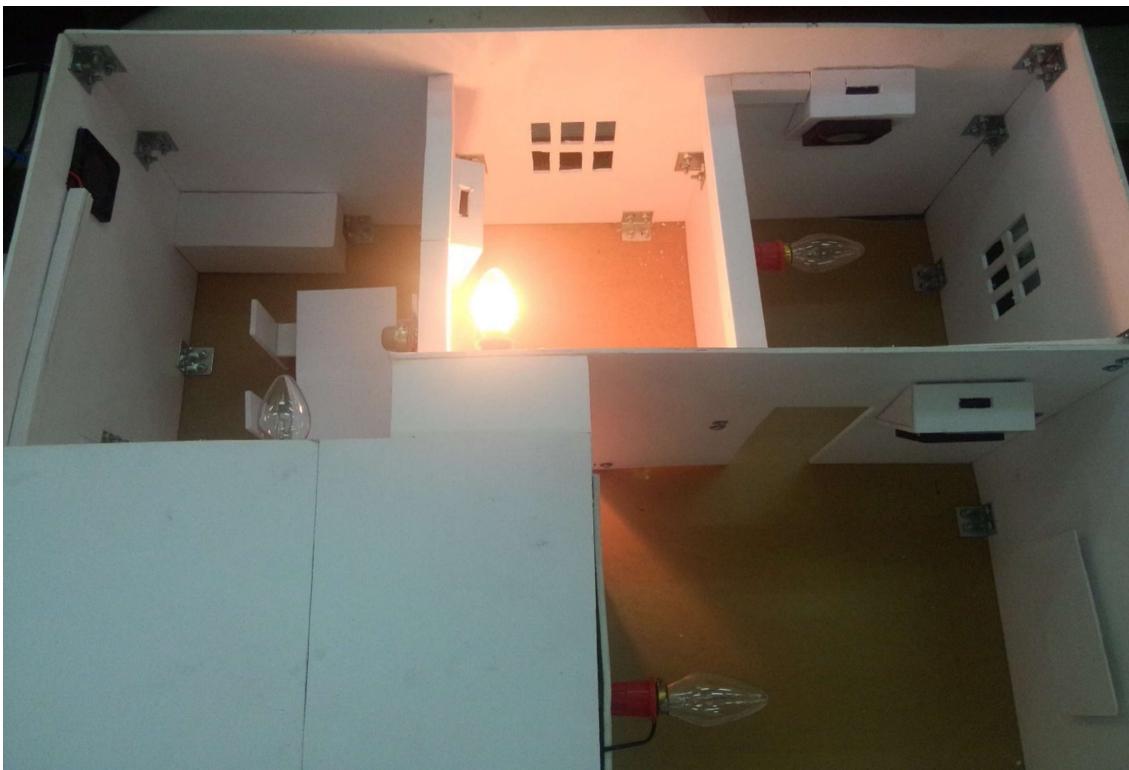
CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau đó nhấn vào Light1, Fan1 để bật đèn và quạt phòng 1.



Hình 5.9: Nhấn nút Light1, Fan1 để bật đèn và quạt phòng 1

Sau khi nhấn 2 nút “Light1” và “Fan1” thì chờ Esp8266 nhận và gửi dữ liệu điều khiển xuống STM32F407.



Hình 5.10: Đèn và quạt phòng 1 được bật lên

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

STM32F407 phân tích và điều khiển relay bật đèn và quạt phòng 1.

Tên thiết bị	Điều khiển	Trạng thái thực	REFRESH
FAN 1	ON	ON	
LIGHT 1	ON	ON	
FAN 2	OFF	OFF	
LIGHT 2	OFF	OFF	
FAN 3	OFF	OFF	
LIGHT 3	OFF	OFF	
FAN 4	OFF	OFF	
LIGHT 4	OFF	OFF	
VALVE	OFF	OFF	
BẤT HẾT			

Hình 5.11: Trạng thái thiết bị được cập nhật lên web

Sau khi STM32F407 điều khiển relay bật đèn và quạt phòng 1, nó sẽ đọc trạng thái các thiết bị và cập nhật lên web.

Điều khiển tương tự cho các thiết bị còn lại.



Hình 5.12: Bật tắt cả đèn và quạt

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

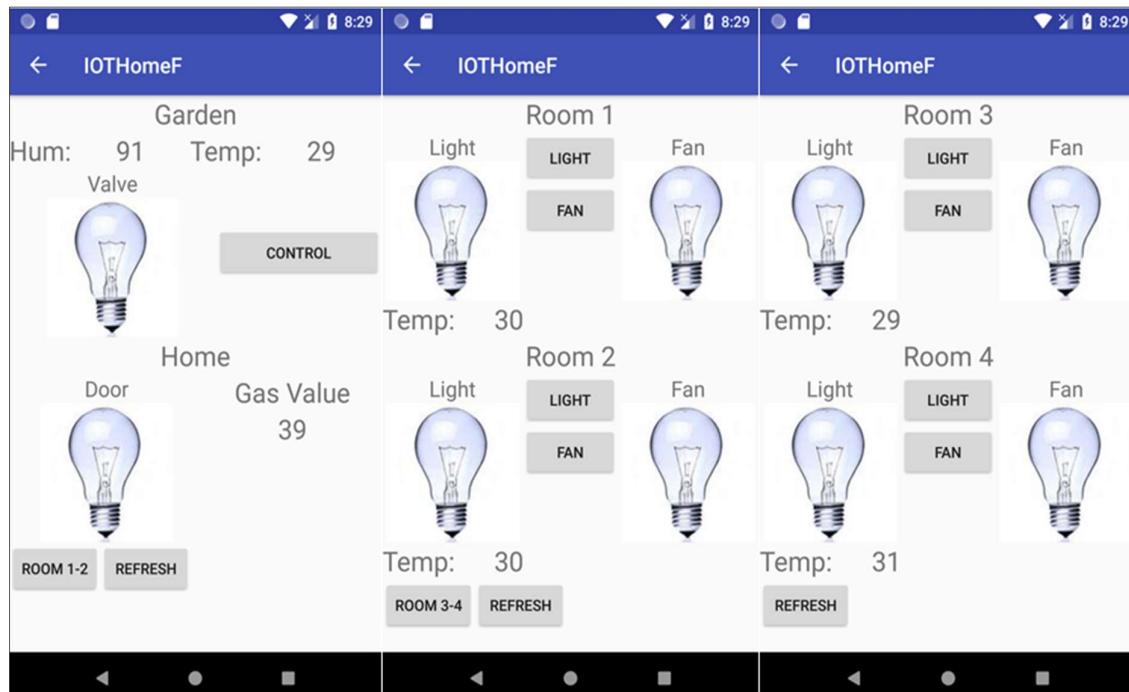


Hình 5.13: Tắt bơm nước

Trong thực tế, đôi khi chúng ta cần bật tất cả hoặc tắt tất cả các thiết bị cùng 1 lúc. Để làm được điều này, ta nhấp vào nút “Bật hết” hoặc “Tắt hết”.

b. App android

Sau khi đăng nhập vào app android các thông số cảm biến được cập nhật vào mục garden, riêng 4 cảm biến ds18b20 được cập nhật ở 4 phòng.



Hình 5.14: Giao diện app khi đăng nhập

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Ban đầu, khi chưa được điều khiển, trạng thái các thiết bị đều tắt.



Hình 5.15: Tất cả các thiết bị đều tắt

Sau đó nhấn vào Light3, Fan3 để bật đèn và quạt phòng 3.



Hình 5.16: Nhấn nút Light3, Fan3 để bật đèn và quạt phòng 3

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau khi nhấn 2 nút “Light3” và “Fan3” thì chờ Esp8266 nhận và gửi dữ liệu điều khiển xuống STM32F407.



Hình 5.17: Đèn và quạt phòng 3 được bật lên

STM32F407 phân tích và điều khiển relay bật đèn và quạt phòng 3.



Hình 5.18: Trạng thái thiết bị được cập nhật lên app android

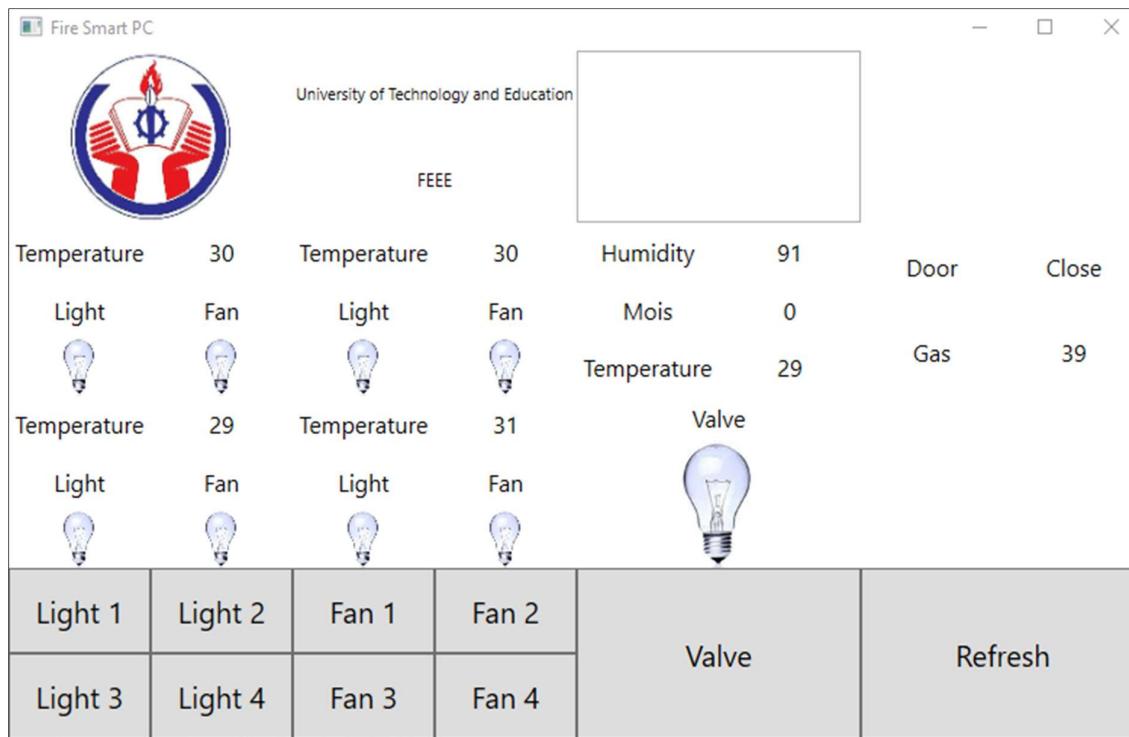
CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau khi STM32F407 điều khiển relay bật đèn và quạt phòng 3, nó sẽ đọc trạng thái các thiết bị và cập nhật lên app android.

Điều khiển tương tự cho các thiết bị còn lại.

c. WPF

Sau khi mở giao diện WPF



Hình 5.19: Giao diện WPF

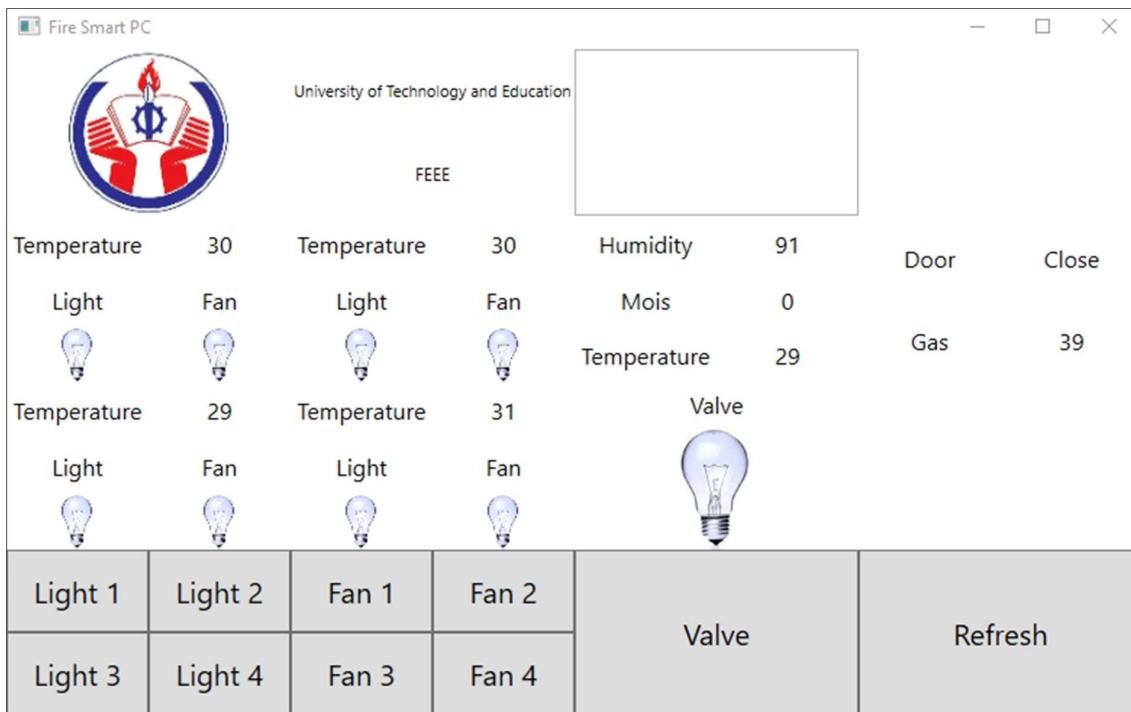
Ban đầu khi mở giao diện, các thông số cảm biến như: nhiệt độ, độ ẩm vườn cây, độ ẩm đất, nhiệt độ các phòng, nồng độ khí gas, trạng thái cửa được cập nhật, các thiết bị đều đang tắt khi chưa được điều khiển.

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ



Hình 5.20: Tất cả các thiết bị đều tắt

Sau đó nhấn vào tất cả các nút Light1, Fan1, Light2, Fan2, Light3, Fan3, Light4, Fan4, Valve để bật tất cả các thiết bị.



Hình 5.21: Nhấn tất cả các nút để bật hết thiết bị

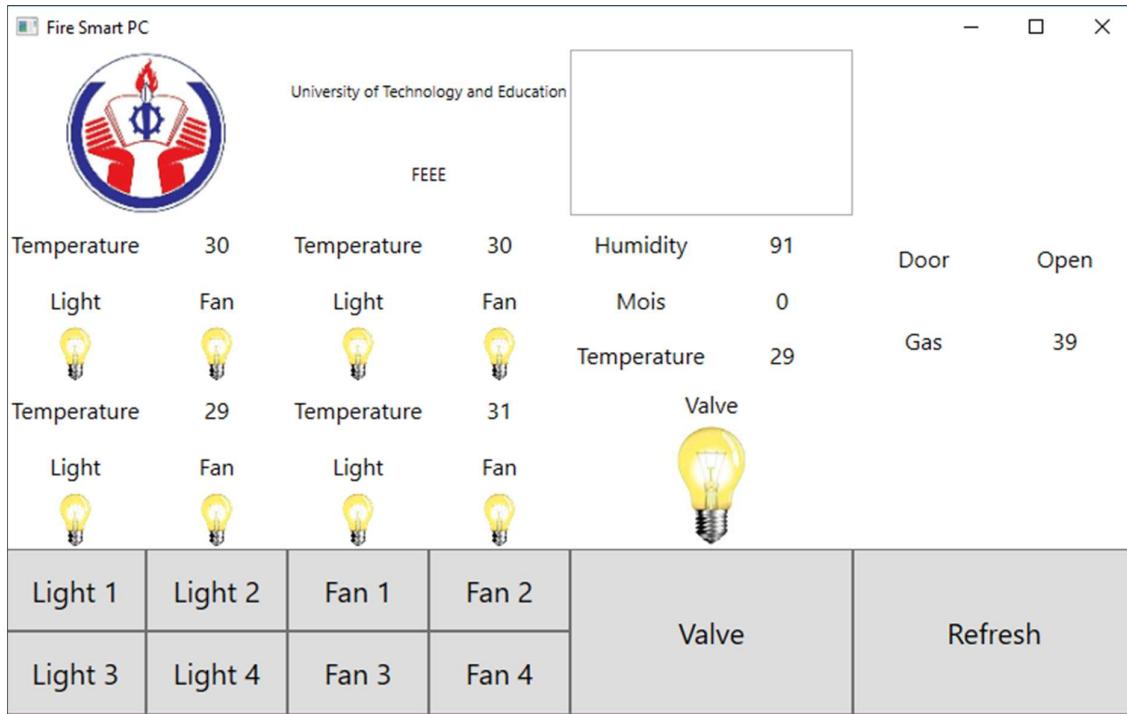
Sau khi nhấn tất cả các nút thì chờ Esp8266 nhận và gửi dữ liệu điều khiển xuống STM32F407.

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ



Hình 5.22: Bật hết tất cả các thiết bị

STM32F407 phân tích và điều khiển relay bật đèn và quạt phòng 3.



Hình 5.23: Trạng thái thiết bị được cập nhật lên WPF

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau khi STM32F407 điều khiển relay bật tất cả các thiết bị, nó sẽ đọc trạng thái các thiết bị và cập nhật lên WPF.

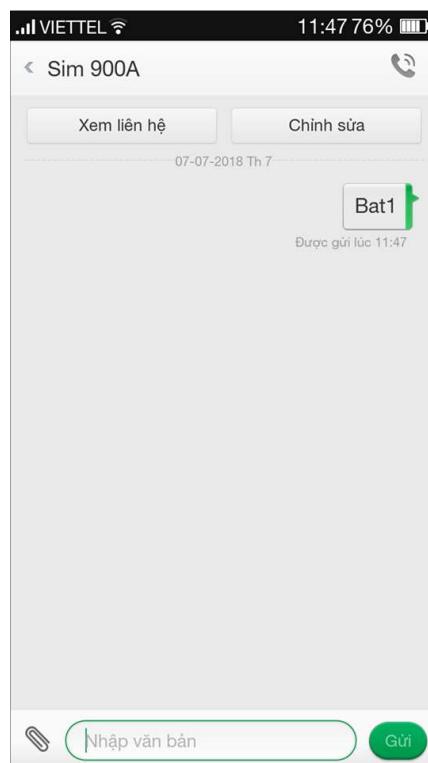
Điều khiển tương tự cho từng thiết bị.

5.3.3 Điều khiển thiết bị và giám sát an ninh qua Sim

a. Điều khiển và giám sát thiết bị

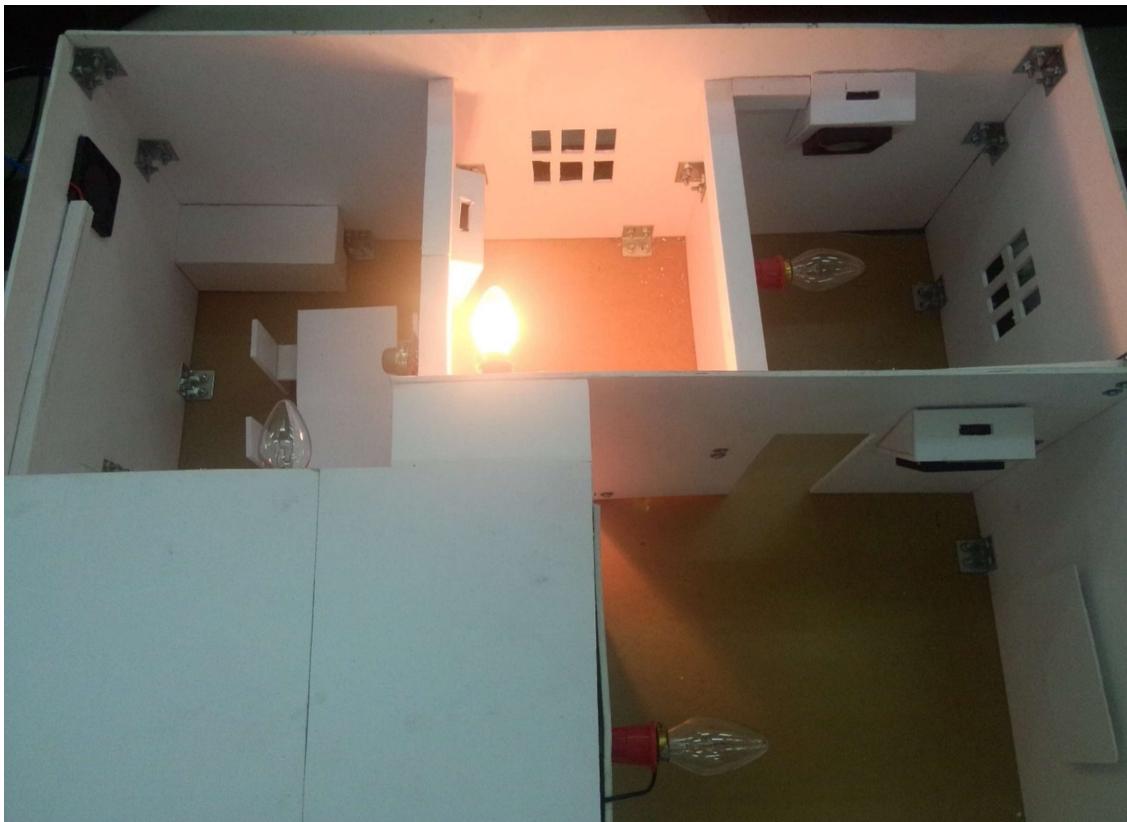
Ngoài việc điều khiển qua internet thì hệ thống còn được điều khiển qua sim bằng tin nhắn.

Tiến hành gửi tin nhắn cho Sim với cú pháp “Bat1”



Hình 5.24: Gửi tin nhắn Bat1 cho sim để bật Light1

Sau khi gửi tin nhắn thì chờ Arduino xử lí và gửi qua cho STM32F407.



Hình 5.25: Bật đèn phòng 1

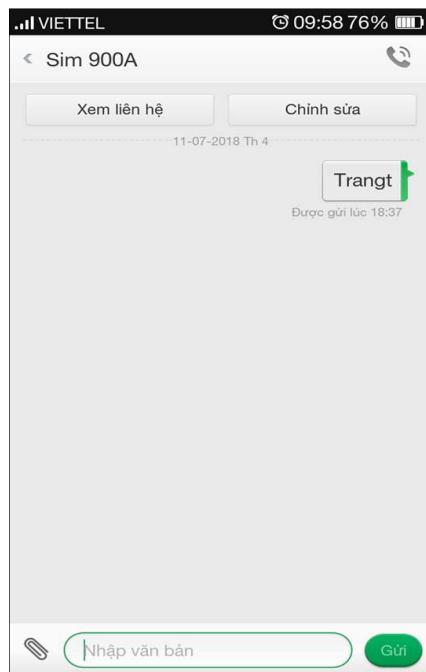
Sau đó, STM32F407 phân tích và điều khiển relay bật đèn phòng 1.

Điều khiển tương tự các thiết bị khác với tin nhắn “Batx” nếu muốn bật thiết bị và nếu muốn tắt thì nhắn tin “Tatx” với x từ 1 tới 8 tương ứng với 8 thiết bị. Ngoài ra nếu muốn bật hết hoặc tắt hết ta nhắn tin “Bathet” hoặc “Tathet”.

b. Giám sát thiết bị

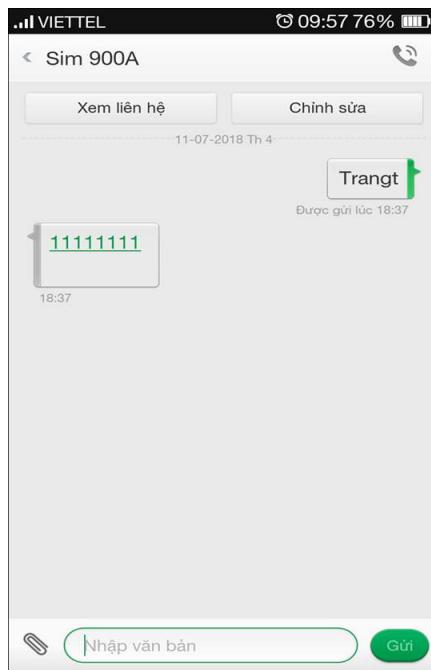
Nếu muốn kiểm tra trạng thái hiện tại của thiết bị, tiến hành nhắn tin “Trangt” gửi cho sim.

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ



Hình 5.26: Gửi tin nhắn "Trangt" để kiểm tra trạng thái thiết bị

Sau khi gửi tin nhắn thì chờ Arduino gửi qua cho STM32F407 xử lý.



Hình 5.27: Trạng thái thiết bị được gửi lại qua tin nhắn

Sau khi STM32F407 xử lý dữ liệu thì gửi lại trạng thái thiết bị cho Arduino và sim sẽ gửi lại trạng thái thiết bị qua tin nhắn. Ở trường hợp này, trạng thái thiết bị đang bật hết nên sẽ nhận được tin nhắn “11111111”.

c. Giám sát an ninh

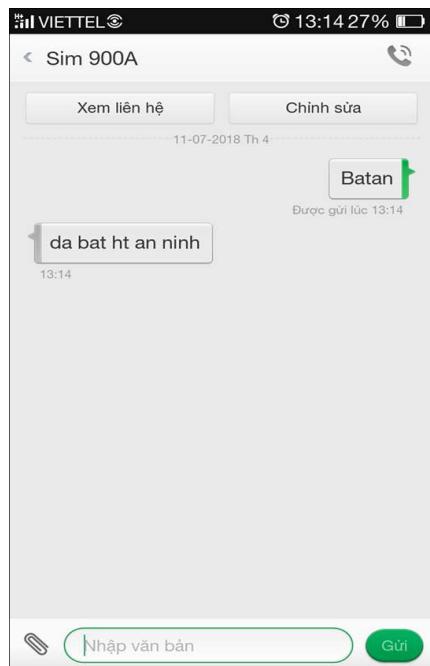
CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Để giám sát an ninh ngôi nhà, tiến hành nhắn tin “Batan” gửi cho sim.



Hình 5.28: Gửi tin nhắn "Batan" để bật an ninh

Sau khi gửi tin nhắn thì chờ Arduino gửi qua cho STM32F407 xử lý.



Hình 5.29: Trạng thái an ninh được gửi lại qua tin nhắn

Sau khi STM32F407 xử lý dữ liệu thì gửi lại trạng thái an ninh cho Arduino và sim sẽ gửi lại tin nhắn “da bat ht an ninh”.

Tương tự, nếu muốn tắt an ninh thì tiến hành gửi tin nhắn “Tatan”.

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Sau khi hệ thống an ninh được bật lên cho phép cảm biến PIR hoạt động.



Hình 5.30: Hệ thống an ninh gửi tin nhắn khi có trộm vào nhà

Lúc này khi cảm biến phát hiện người vào nhà Arduino sẽ xử lý và gửi lại cho sim tin nhắn “CO TROM VAO NHA” để thông báo cho chủ nhà.



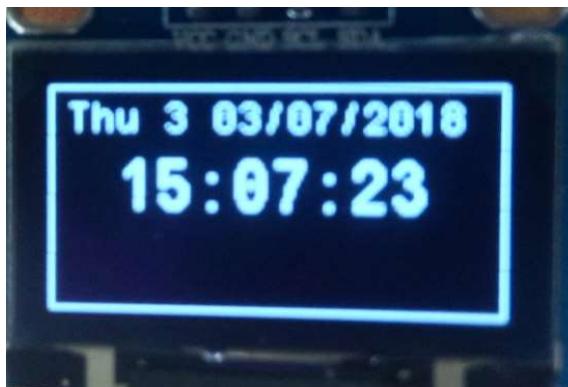
Hình 5.31: Hệ thống an ninh gọi điện khi có trộm vào nhà

Đồng thời hệ thống sẽ gọi điện để thông báo cho chủ nhà.

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

5.3.4 Hiển thị màn hình oled

Sau khi cấp nguồn, STM32F407 sẽ đọc thời gian thực từ module ds1307 và hiển thị lên màn hình Oled.



Hình 5.32: Hiển thị thời gian thực trên oled

Ở chế độ tiếp theo, sau khi nhấn phím “next”, màn hình oled sẽ hiện thị thông số 4 cảm biến nhiệt độ ds18b20.



Hình 5.33: Hiện thị nhiệt độ phòng trên oled

Tiếp theo, tiếp tục nhấn “next” màn hình oled sẽ chuyển sang chế độ hiển thị nhiệt độ, độ ẩm vườn cây, độ ẩm đất, nồng độ khí gas và trạng thái cửa.



Hình 5.34: Hiển thị các thông số cảm biến trên oled

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT – ĐÁNH GIÁ

Cuối cùng, nhấn “next” màn hình oled sẽ chuyển sang chế độ hiển thị trạng thái tất cả các thiết bị.



Hình 5.35: Hiển thị trạng thái thiết bị trên oled

Nếu tiếp tục nhấn “next” thì chế độ hiện thì sẽ quay lại hiển thị thời gian thực, nếu nhấn “back” thì Oled sẽ hiện thì chế độ trước đó.

5.4 NHẬN XÉT – ĐÁNH GIÁ

5.4.1 Nhận xét

Sau thời gian 15 tuần nghiên cứu và thực hiện đề tài, mạch cơ bản đáp ứng được yêu cầu thiết kế ban đầu. Dưới đây là một số nhận xét:

a. Ưu điểm

- Hệ thống có hồi tiếp trạng thái của thiết bị.
- Hệ thống có thể được điều khiển bất kì nơi đâu có internet.
- Hệ thống tự động đóng cửa theo lịch trình, khi khí gas rò rỉ sẽ cảnh báo cho chủ căn hộ bằng cách nhắn tin và gọi điện.
- Ứng dụng RFID để đóng mở cửa. Có cảnh báo trộm khi nhập sai mật khẩu 3 lần.
- Giao diện thiết kế dễ sử dụng và đẹp mắt.

b. Hạn chế

- Hệ thống phụ thuộc vào tốc độ của wifi hoặc 3G.
- Chưa hệ thống dập lửa khi có cháy.

5.4.2 Đánh giá

Sau quá trình vận hành thử hệ thống, nhóm thực hiện đã thu về những số liệu như bảng sau:

CHƯƠNG 5: KẾT QUẢ NHẬN XÉT ĐÁNH GIÁ

Bảng 5.1: Số liệu thực nghiệm

	Web server	App android	WPF	Đánh giá	Ghi chú
Điều khiển và giám sát thiết bị (50 lần)	46 lần thành công	46 lần thành công	46 lần thành công	Đạt	
Giám sát các cảm biến (2 tiếng)	Hoạt động ổn định	Hoạt động ổn định	Hoạt động ổn định	Đạt	
Quét thẻ hay RFID nhập mật khẩu (50 lần)	50 lần thành công			Đạt	
Đánh giá chung				Đạt	

Qua những số liệu bảng trên, nhóm đánh giá hệ thống đạt yêu cầu với những mục tiêu đã đề ra. Mô hình có tính thẩm mỹ, an toàn, bảo mật và dễ sử dụng. Sau thời gian test thử, mạch cho thấy sự ổn định. Tuy nhiên vẫn còn một số hạn chế cần khắc phục nếu muốn đưa vào thực tế đời sống như: tốc độ điều khiển và phản hồi còn chậm, chưa có chức năng nhận dạng chủ nhà, chưa có chức năng báo trộm khi có trộm đột nhập từ trên mái nhà, chưa hệ thống dập lửa, hiện tượng nhiễu do sai sót trong quá trình thi công mạch in và hàn linh kiện.

Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

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

- Sản phẩm đạt yêu cầu điều khiển và giám sát trạng thái của các thiết bị, các cảm biến qua mạng Internet.
- Kết quả điều khiển hồi tiếp được trạng thái hiện tại của thiết bị và thông số của cảm biến.
- Mạch thực hiện được chức năng mở cửa bằng RFID và nhập mật khẩu.
- Có hệ thống cảnh báo cho chủ căn hộ biết khi có trộm hoặc rò rỉ khí gas.
- Mô hình thi công có tính thẩm mỹ, an toàn, dễ thao tác.

6.2 HƯỚNG PHÁT TRIỂN

- Mở rộng số thiết bị cần điều khiển với công suất lớn.
- Mở rộng thêm nhiều cửa đóng mở tự động bằng cách nhận dạng dấu vân tay hoặc khuôn mặt.
- Tự động ngắt thiết bị khi chủ nhà quên tắt.
- Phát triển thêm hệ thống dập lửa khi phát hiện có cháy xảy ra.
- Thêm camera giám sát ngôi nhà thông qua web hoặc android.
- Phát triển thêm hệ thống đọc điện năng tiêu thụ của các thiết bị.

TÀI LIỆU THAM KHẢO

1. Sách tham khảo

- [1] Trần Thu Hà – Trương Thị Bích Ngà – Nguyễn Thị Lưỡng – Bùi Thị Tuyết Đan – Phù Thị Ngọc Hiếu – Dương Thị Cẩm Tú, Giáo trình *Điện tử cơ bản*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 3013.
- [2] Nguyễn Đình Phú, Giáo trình *Vi điều khiển*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.
- [3] Nguyễn Đình Phú, Giáo trình *Vi xử lý nâng cao*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.
- [4] Nguyễn Đình Phú – Nguyễn Trường Duy, Giáo trình *Kỹ thuật số*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2013.
- [5] Nguyễn Việt Hùng – Nguyễn Ngô Lâm – Nguyễn Văn Phúc, Giáo trình *Kỹ thuật truyền số liệu*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, 2011.
- [6] Hoàng Ngọc Văn, Giáo trình *Điện tử công suất*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, 2007.
- [7] Nguyễn Văn Hiệp – Đinh Quang Hiệp, Giáo trình *Lập trình Android cơ bản*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2015.
- [8] Nguyễn Văn Hiệp, Giáo trình *Lập trình Android trong ứng dụng điều khiển*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2016.
- [9] Nguyễn Văn Hiệp, Giáo trình *Công nghệ nhận dạng bằng sóng vô tuyến*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2014.

2. Trang web tham khảo

- [1] Github.com
- [2] Stackoverflow.com
- [3] Hocarm.org
- [4] Avislab.com
- [5] Embeddedsystemengineering.blogspot.com
- [6] Youtube Channel: Firebase
- [7] Arduino.vn