

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA ĐIỆN - ĐIỆN TỬ**



**ĐỒ ÁN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - TRUYỀN THÔNG**

**THIẾT KẾ VÀ THI CÔNG HỘP ĐIỀU KHIỂN  
THIẾT BỊ ĐIỆN TRONG NHÀ**

**GVHD: ThS. PHAN VĂN HOÀN  
SVTH: LÊ TẤN ĐẠT  
MSSV: 12141047  
SVTH: LÊ VĂN HÙNG  
MSSV: 12141553**



**Tp. Hồ Chí Minh, tháng 7/2016**

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



# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**ĐỀ TÀI:**

**THIẾT KẾ VÀ THI CÔNG**

**HỘP ĐIỀU KHIỂN**

**THIẾT BỊ ĐIỆN TRONG NHÀ**

**GVHD: Th.s PHAN VĂN HOÀN**

**SVTH: LÊ TẤN ĐẠT**

**MSSV: 12141047**

**SVTH: LÊ VĂN HÙNG**

**MSSV: 12141553**

**Tp. Hồ Chí Minh - 7/2016**

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

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

*THIẾT KẾ VÀ THI CÔNG*  
*HỘP ĐIỀU KHIỂN*  
*THIẾT BỊ ĐIỆN TRONG NHÀ*

GVHD: Th.s PHAN VÂN HOÀN

SVTH: LÊ TẤN ĐẠT

MSSV: 12141047

SVTH: LÊ VĂN HÙNG

MSSV: 12141553

Tp. Hồ Chí Minh – 07/2016





# LỜI CAM ĐOAN

Chúng tôi cam đoan đây là công trình nghiên cứu của chúng tôi.

Các số liệu, kết quả nêu trong luận văn là hoàn toàn do chúng tôi thực hiện.

Nhóm thực hiện đề tài

# LỜI CẢM ƠN

Xin gửi lời cảm ơn chân thành đến quý thầy cô khoa Điện – Điện tử đã giảng dạy em trong suốt bốn năm học và tạo điều kiện thuận lợi để em thực hiện tốt đề tài.

Cảm ơn gia đình, cha mẹ đã là nguồn động viên to lớn về vật chất và tinh thần trong suốt thời gian học hành, để em có được tương lai, theo đuổi ước mơ và sự nghiệp.

Đặc biệt cảm ơn thầy **Phan Văn Hoàn** đã tạo điều kiện và hướng dẫn em cách học tập cũng như nghiên cứu để hoàn thành tốt đồ án tốt nghiệp.

Cuối cùng xin chúc gia đình, bạn bè và quý thầy, cô nhiều sức khỏe và thành công trong công việc.

Xin chân thành cảm ơn!

Người thực hiện đề tài

# MỤC LỤC

Trang bìa .....	i
Nhiệm vụ đồ án.....	ii
Lịch trình .....	iii
Cam đoan .....	iv
Lời cảm ơn .....	v
Mục lục .....	vi
Liệt kê hình vẽ .....	ix
Liệt kê bảng .....	xii
Tóm tắt .....	xiii
<b>CHƯƠNG 1. TỔNG QUAN .....</b>	<b>1</b>
1.1. Đặt vấn đề .....	1
1.2. Mục tiêu .....	2
1.3. Nội dung nghiên cứu.....	2
1.4. Giới hạn.....	3
1.6. Bố cục .....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>4</b>
2.1. Vi điều khiển .....	4
2.1.1. Bộ vi xử lý ARM CORTEX – M4 .....	4
2.1.2. Vi điều khiển STM32F407 .....	7
2.2. Module LCD TFT .....	24
2.3. Giao tiếp sdcard.....	25
2.4. Phương pháp điều chỉnh điện áp xoay chiều một pha.....	28
2.5. Tìm hiểu hệ điều hành Android.....	30
2.5.1. Khái niệm Android.....	30
2.5.2. Kiến trúc hệ điều hành Android .....	31
2.5.3. Sơ lược về lập trình ứng dụng cho android .....	34
2.5.4. Các thành phần cơ bản của một ứng dụng android .....	35
2.5.5. Chức năng giám sát bằng camera ip thông qua ứng dụng android .....	38

<b>CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ.....</b>	<b>39</b>
3.1. Giới thiệu.....	39
3.2. Tính toán và thiết kế hệ thống.....	39
3.2.1. Sơ đồ khối hệ thống.....	39
3.2.2. Thiết kế khối mạch công suất.....	40
3.2.3. Khối Điều Khiển.....	45
3.2.4. Khối hiển thị giao tiếp với người dùng.....	47
3.2.5. Khối giao tiếp giải mã âm thanh.....	48
3.2.6. Khối giao tiếp SD Card.....	51
3.2.7. Module wifi ESP8266 V12.....	51
3.2.8. Khối cảm biến.....	53
<b>CHƯƠNG 4. THI CÔNG HỆ THỐNG .....</b>	<b>56</b>
4.1. Giới thiệu.....	56
4.2. Thi công hệ thống .....	56
4.2.1. Thi công bo mạch .....	56
4.2.2. Lắp ráp và kiểm tra .....	59
4.3. Đóng gói và thi công mô hình .....	60
4.3.1. Đóng gói bộ điều khiển .....	60
4.3.2. Thi công mô hình .....	61
4.4. Lập trình hệ thống.....	62
4.4.1. Lưu đồ giải thuật.....	62
4.4.2. Phần mềm lập trình cho vi điều khiển.....	69
4.4.3. Phần mềm lập trình cho điện thoại android.....	74
4.5. Viết tài liệu hướng dẫn sử dụng,thao tác .....	81
4.5.1. Viết tài liệu hướng dẫn sử dụng.....	81
4.3.2. Quy trình thao tác .....	81
<b>CHƯƠNG 5. KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ .....</b>	<b>85</b>
5.1. Kết quả.....	85
5.1.1. Kết quả nghiên cứu.....	85
5.1.2. Kết quả thi công.....	86
5.2. Nhận xét – Đánh giá .....	91

<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.</b> .....	<b>92</b>
6.1. Kết luận .....	92
6.2. Hướng phát triển .....	92
<b>TÀI LIỆU THAM KHẢO</b> .....	<b>93</b>
<b>PHỤ LỤC</b> .....	<b>94</b>

# LIỆT KÊ HÌNH VẼ

Hình	Trang
Hình 2.1. Kiến trúc bộ vi xử lý ARM Cortex-M4 .....	5
Hình 2.2. Lõi ARM7TDMI của Cortex-M4. ....	6
Hình 2.3. Kiến trúc đường ống của ARM Cortex-M4 .....	6
Hình 2.4. Cấu trúc bus hệ thống.....	8
Hình 2.6. Sơ đồ khối SPI.....	13
Hình 2.7. Phương thức truyền I2C. ....	14
Hình 2.8. Sơ đồ khối I2C. ....	15
Hình 2.9. Sơ đồ khối FSMC.....	17
Hình 2.11. Sơ đồ khối Timer 1.....	20
Hình 2.12. Chế độ capture/compare kênh 1.....	21
Hình 2.13. Dạng sóng Edge-aligned PWM.....	22
Hình 2.14. Dạng sóng Center-aligned PWM .....	22
Hình 2.15. Chế độ half duplex của USART. ....	23
Hình 2.16. Chế độ hỗ trợ giao tiếp SPI của USART. ....	24
Hình 2.17. Màn hình LCD TFT. ....	24
Hình 2.18. Quá trình đọc dữ liệu của khối SDIO.....	27
Hình 2.20. Dạng sóng điện áp trong phương pháp điều chỉnh điện áp xoay chiều. ....	28
Hình 2.21. Mạch điều chỉnh độ sáng cơ bản dùng vi xử lý. ....	29
Hình 2.22. Kiến trúc hệ điều hành Android. ....	32
Hình 2.23. Vòng đời của Activity. ....	36
Hình 2.24. Sử dụng Intent để trao đổi thông tin giữa hai chương trình.....	37
Hình 3.1. Sơ đồ khối hệ thống. ....	39
Hình 3.2. Mạch nguồn 5V sử dụng LM2576.....	41
Hình 3.3. Mạch phát hiện điểm 0.....	42
Hình 3.4. Mạch chuyển đổi công suất.....	43
Hình 3.5. Mạch PWM 0-16VDC. ....	44
Hình 3.6. Sơ đồ kết nối vi điều khiển. ....	46
Hình 3.7. Sơ đồ kết nối chân GLCD và Touch với vi điều khiển.....	47
Hình 3.8. Sơ đồ khối VS1003B.....	48
Hình 3.9. Phổ ngõ ra tại chân LEFT hoặc RIGHT với trở ngõ ra là 30 $\Omega$ .....	49
Hình 3.10. Giảm đồ xung của quá trình đọc VS1003B .....	49

Hình 3.11. Giản đồ xung của quá trình ghi VS1003B. ....	50
Hình 3.12. Sơ đồ thứ tự giải mã dữ liệu của VS1003B. ....	50
Hình 3.13. Sơ đồ khối SD Card. ....	51
Hình 3.14. Module ESP8266 V12. ....	52
Hình 3.15. Sơ đồ nguyên lý giao tiếp module ESP8266 với vi điều khiển ....	53
Hình 3.16. Cảm biến DHT11. ....	54
Hình 3.17. Cảm biến ánh sáng BH1750.....	55
Hình 4.1. Lớp Top PCB mạch điều khiển.....	56
Hình 4.2. Lớp Bottom PCB mạch điều khiển ....	57
Hình 4.3. Lớp Top PCB mạch công suất ....	57
Hình 4.4. Lớp Bottom PCB mạch công suất.....	58
Hình 4.5. Kết quả thi công mạch công suất. ....	59
Hình 4.6. Kết quả thi công mạch điều khiển.....	59
Hình 4.7. Hộp điều khiển sau khi thi công.....	60
Hình 4.8. Mặt trên của hộp điều khiển.....	60
Hình 4.9. Vị trí công tắt nguồn. ....	61
Hình 4.10. Mặt dưới của hộp điều khiển.....	61
Hình 4.11. Mặt trên của hộp điều khiển.....	61
Hình 4.13. Lưu đồ giải thuật tab Home. ....	63
Hình 4.14. Lưu đồ giải thuật tab Setup. ....	64
Hình 4.15. Lưu đồ giải thuật tab Timer. ....	64
Hình 4.16. Lưu đồ giải thuật tab Music. ....	65
Hình 4.17. Lưu đồ Activity Main.....	66
Hình 4.18. Lưu đồ giải thuật hàm Run(). ....	67
Hình 4.19. Lưu đồ hàm onOptionsItemSelected(). ....	68
Hình 4.20. Hộp thoại cài đặt Keil C V5.....	69
Hình 4.21. Hộp thoại license cài đặt Keil C V5 ....	70
Hình 4.23. Thêm Pack STM32F4 cho Keil C V5.....	71
Hình 4.26. Chọn dòng vi điều khiển để lập trình.....	73
Hình 4.27. Hộp thoại cài đặt Android Studio. ....	74
Hình 4.28. Chọn các option của Android Studio. ....	75
Hình 4.29. Hộp thoại quản lý giấy phép Android Studio. ....	75
Hình 4.30. Hộp thoại chọn nơi lưu bản cài đặt Android Studio. ....	76
Hình 4.31. Hộp thoại Install Android Studio.....	77

Hình 4.32. Hộp thoại Quick Start của Android Studio. ....	78
Hình 4.33. Đặt tên cho project android. ....	78
Hình 4.34. Hộp thoại cho phép chọn bản SDK. ....	79
Hình 4.35. Hộp thoại chọn giao diện bắt đầu cho ứng dụng. ....	80
Hình 4.36. Quy trình thao tác trên màn hình cảm ứng. ....	82
Hình 4.37. Quy trình thao tác trên điện thoại. ....	84
Hình 5.1. Giao diện tab Home. ....	86
Hình 5.2. Giao diện tab Setup. ....	86
Hình 5.3 Điều chỉnh PWM. ....	87
Hình 5.4. Giao diện tab Mucsic. ....	87
Hình 5.5. Giao diện tab Timer. ....	88
Hình 5.6. Giao diện tab System. ....	88
Hình 5.7. Thiết bị được điều khiển ở chế độ tắt. ....	89
Hình 5.8. Thiết bị được điều khiển ở chế độ mở. ....	99
Hình 5.9. Giao diện chính của ứng dụng. ....	90
Hình 5.10. Thêm thiết bị. ....	90
Hình 5.11. Giao diện tùy chỉnh. ....	90
Hình 5.12. Giao diện hẹn giờ. ....	90
Hình 5.13. Giao diện giám sát bằng camera IP. ....	91



# LIỆT KÊ BẢNG

<b>Bảng</b>	<b>Trang</b>
Bảng 2.1 Các tín hiệu FSMC sử dụng để giao tiếp với NOR Flash. ....	19
Bảng 2.2 Chức năng chân SD Card. ....	26
Bảng 2.3 Các phiên bản Android. ....	31
Bảng 3.1 Thông số MOSFET IRF640. ....	44
Bảng 4.1 Danh sách linh kiện chính. ....	58

# TÓM TẮT

Ngày nay với sự phát triển mạnh mẽ của công nghệ IoTs (Internet of Things) cùng với các thiết bị thông minh như SmartPhone, SmartWatch... các sản phẩm trên đã được ứng dụng để tạo ra các sản phẩm thông minh nhằm phục vụ cuộc sống của con người ngày càng tốt hơn, tiện nghi hơn, đặc biệt là các sản phẩm điều khiển thông minh, chúng được ứng dụng rộng rãi trong đời sống và công nghiệp.

Đã có nhiều dự án về nhà thông minh được nghiên cứu và phát triển thành sản phẩm. Tuy nhiên các nhà thông minh chưa được triển khai nhiều do chi phí lắp đặt và vận hành cao, giao diện người dùng chưa thân thiện.

Vì vậy để khắc phục các nhược điểm trên nhóm nghiên cứu đã quyết định thiết kế, thi công hộp điều khiển thiết bị điện trong nhà, trong đó sử dụng vi điều khiển trung tâm là STM32F407VET6, giao diện người dùng được thiết kế trên cảm ứng điện dung GLCD 7". Bên cạnh đó thiết bị còn có thể điều khiển bằng điện thoại thông minh sử dụng hệ điều hành Android để điều khiển, phát nhạc mp3 phục vụ nhu cầu giải trí.

# Chương 1. TỔNG QUAN

## 1.1. ĐẶT VẤN ĐỀ

Hiện nay với sự phát triển vượt bậc của khoa học kỹ thuật. Con người ngày càng hiện đại hóa các trang thiết bị trong nhà nhằm nâng cao chất lượng cuộc sống. Nhà thông minh (smart home) đang trở thành xu hướng tất yếu và là tiêu chuẩn của những ngôi nhà hiện đại.

Việt Nam không nằm ngoài xu hướng đó, đã có rất nhiều trường đại học và công ty tại Việt Nam nghiên cứu và phát triển các thiết bị điều khiển thông minh. Tuy nhiên việc ứng dụng ngoài thực tế còn gặp nhiều khó khăn do chi phí đầu tư cao, chưa tiết kiệm về chi phí vận hành và giao diện điều khiển chưa thân thiện với người dùng. Hiện tại các đề tài cấp sinh viên thực hiện chỉ xoay quanh một số nội dung như: điều khiển bằng điện thoại, tin nhắn, sóng vô tuyến hay là Bluetooth và chỉ sử dụng các cảm biến thông dụng như nhiệt độ, cảm biến khí gas, ứng dụng công nghệ RFID, về lập trình thì lập trình vi điều khiển PIC, Arduino....

Ngày nay các hãng sản xuất thiết bị điện tử như Microchip, Atmel, Intel, STMicroelectronics, cho ra đời nhiều dòng vi điều khiển 32 bit và 64 bit được ứng dụng rộng rãi trong công nghiệp và đời sống. Hãng STMicroelectronics đã tung ra thị trường các dòng vi điều khiển 32 bit sử dụng core Arm cortex M4 với ưu điểm là giá thành rẻ, hiệu suất cao, dễ sử dụng và cộng đồng sử dụng khá mạnh trên khắp thế giới.

Để tiếp tục phát triển theo xu hướng của thế giới nhóm quyết định chọn đề tài “THIẾT KẾ VÀ THI CÔNG HỘP ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG NHÀ”. Và với những ưu điểm của dòng Arm cortex M4, chọn STM32F407VET6 làm vi điều khiển trung tâm.

### **1.2. MỤC TIÊU**

Mục tiêu là thiết kế và thi công hộp điều khiển thiết bị điện trong nhà. Gồm có mạch điều khiển và mạch công suất. Cho phép điều khiển, giám sát cũng như hẹn giờ bật tắt 5 thiết bị AC 220V trên màn hình cảm ứng 7” . Hiển thị Nhiệt độ, độ ẩm và thời gian thực. Cho phép người dùng điều khiển và giám sát thiết bị từ điện thoại thông qua wifi và camera ip. Ngoài ra còn có chức năng giải trí như nghe nhạc MP3.

### **1.3. NỘI DUNG NGHIÊN CỨU**

Các nội dung mà nhóm cần thực hiện bao gồm:

- NỘI DUNG 1: Tìm hiểu các hoạt động của các bộ điều khiển thông minh có trên thị trường.
- NỘI DUNG 2: Đưa ra các giải pháp hoạt động và thiết kế các khối.
- NỘI DUNG 3: Thiết kế và thi công mạch công suất.
- NỘI DUNG 4: Thiết kế và thi công mạch điều khiển giao tiếp giữa vi điều khiển STM32F407VET6 với cảm biến nhiệt độ- độ ẩm, cảm biến ánh sáng, màn hình GLCD, thẻ nhớ microSD, IC phát nhạc VS1003, module wifi esp8266 và mạch công suất.
- NỘI DUNG 5: Thiết giao diện người dùng trên màn hình GLCD 7” và giao diện trên điện thoại android bằng phần mềm Android Studio.
- NỘI DUNG 6: Viết chương trình cho vi điều khiển bằng phần mềm Keil  $\mu$ Vision 5.
- NỘI DUNG 7: Lắp ráp các khối và tiến hành điều khiển thử nghiệm các thiết bị.
- NỘI DUNG 8: Chỉnh sửa các lỗi xuất hiện.
- NỘI DUNG 9: Viết báo cáo luận văn.
- NỘI DUNG 10: Bảo vệ luận văn.

### 1.4. GIỚI HẠN

Các giới hạn của đề tài:

- Sản phẩm chỉ ứng dụng cho các thiết bị công suất nhỏ như đèn, quạt, TV, DVD...
- Chỉ sử dụng các cảm biến đơn giản như nhiệt độ độ ẩm, độ chính xác chưa cao.
- Hệ thống chỉ sử dụng trong nhà.

### 1.5. BỐ CỤC

- **Chương 1: Tổng quan.**

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

- **Chương 2: Cơ sở lý thuyết.**

Giới thiệu về Arm cortex M4 và vi điều khiển STM32F4 bên cạnh đó giới thiệu về các chuẩn giao tiếp và linh kiện chính được sử dụng trong đề tài.

- **Chương 3: Tính toán thiết kế.**

Chương này trình bày sơ đồ khối hệ thống, cách tính toán các thông số kỹ thuật của các khối sử dụng, thiết kế các khối với yêu cầu đặt ra ban đầu.

- **Chương 4: Thi công hệ thống.**

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

- **Chương 5: Kết quả - Nhận xét - Đánh giá**

Chương này trình bày các kết quả đạt được và chưa đạt.

- **Chương 6: Kết luận và hướng phát triển.**

Đưa ra đánh giá về những kết quả đạt được so với mục tiêu ban đầu, đề ra các phương án khắc phục và hướng phát triển của đề tài.

## **Chương 2. CƠ SỞ LÝ THUYẾT**

### **2.1. VI ĐIỀU KHIỂN**

#### **2.1.1 Bộ vi xử lý ARM CORTEX – M4**

Cấu trúc ARM (viết tắt từ tên gốc là Acorn RISC Machine) là một loại cấu trúc vi xử lý 32 – bit kiểu RISC (viết tắt của Reduced Instructions Set Computer - Máy tính với tập lệnh đơn giản hóa) được sử dụng rộng rãi trong các thiết kế nhúng. Do có đặc điểm tiết kiệm năng lượng, các bộ CPU ARM chiếm ưu thế trong các sản phẩm điện tử di động, mà với các sản phẩm này việc tiêu tán công suất thấp là một mục tiêu thiết kế quan trọng hàng đầu.[1]

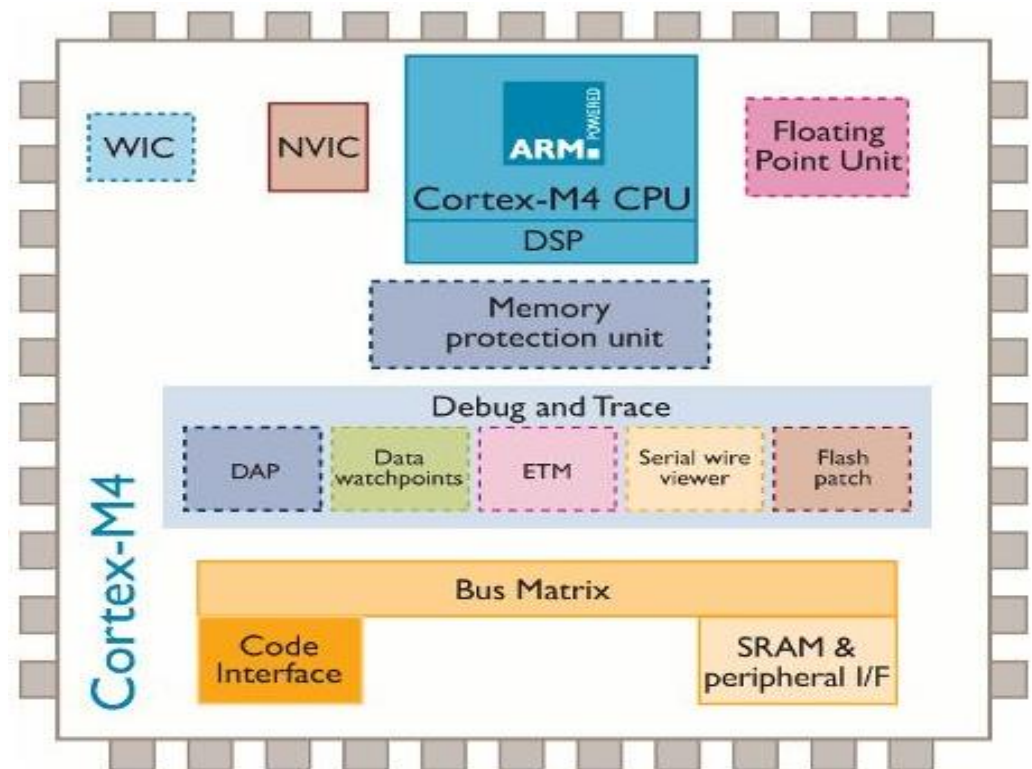
Arm cortex là một thế hệ lõi vi xử lý thế hệ mới đưa ra một kiến trúc chuẩn cho các nhu cầu công nghệ. Kiến trúc của nó là một lõi hoàn chỉnh bao gồm bộ vi xử lý cortex và một hệ thống các thiết bị ngoại vi được thiết kế theo giải pháp SoC.

Cortex cho phép truy cập dữ liệu không thẳng hàng, hỗ trợ đặt xóa các bit bên trong hai vùng 1Mbyte của bộ nhớ mà không cần xử lý luận lý (Boolean processor). Cortex gồm 3 dòng:

- **Cortex-A:** Bộ xử lý dành cho hệ điều hành và các ứng dụng phức tạp. Hỗ trợ tập lệnh ARM, thumb, và thumb-2.
- **Cortex-R:** Bộ xử lý dành cho hệ thống đòi hỏi khắc khe về đáp ứng thời gian thực. Hỗ trợ tập lệnh ARM, thumb, và thumb-2.
- **Cortex-M:** Bộ xử lý dành cho dòng vi điều khiển, được thiết kế để tối ưu về giá thành. Hỗ trợ tập lệnh Thumb-2. (Dòng ARM STM32 có lõi Cortex-M).

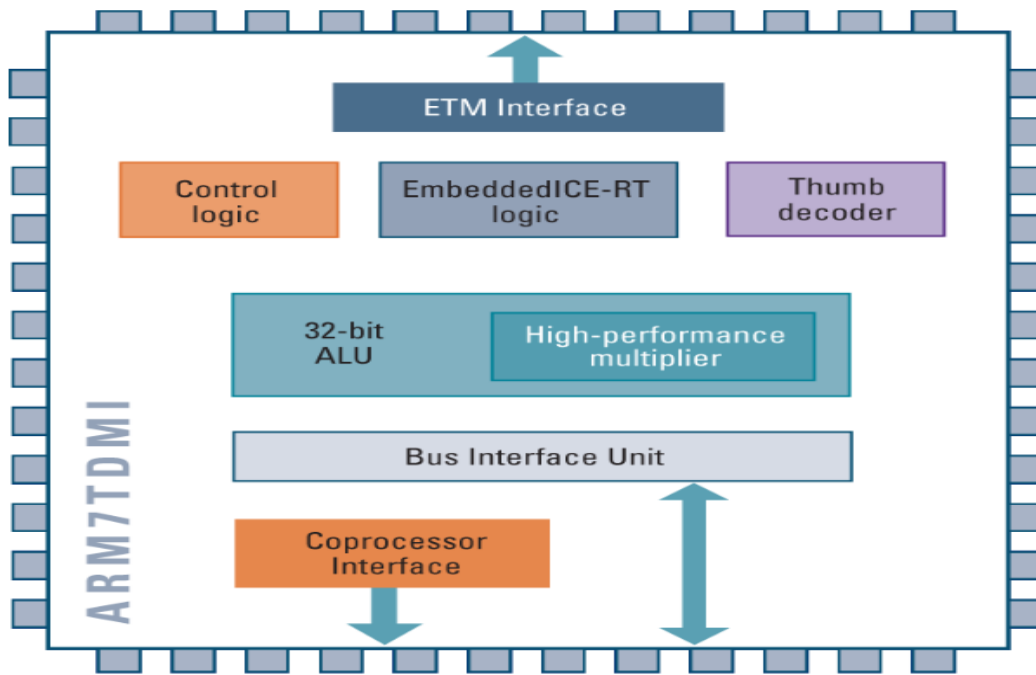
Bộ vi xử lý ARM Cortex-M4 là thế hệ vi xử lý thứ 2 của dòng Cortex-M dựa trên kiến trúc ARMv7-M, được giới thiệu vào năm 2010. Nó được mở rộng thêm về tập lệnh và kiến trúc mới.

Corex-M4 gồm một lõi CPU 32-bit, các thanh ghi 32-bit, đường dẫn dữ liệu nội bộ 32-bit và giao diện bus 32-bit, cùng một số thành phần khác. Ngoài ra, Cortex-M4 cũng đã hỗ trợ một số hoạt động liên quan đến dữ liệu 64-bit.



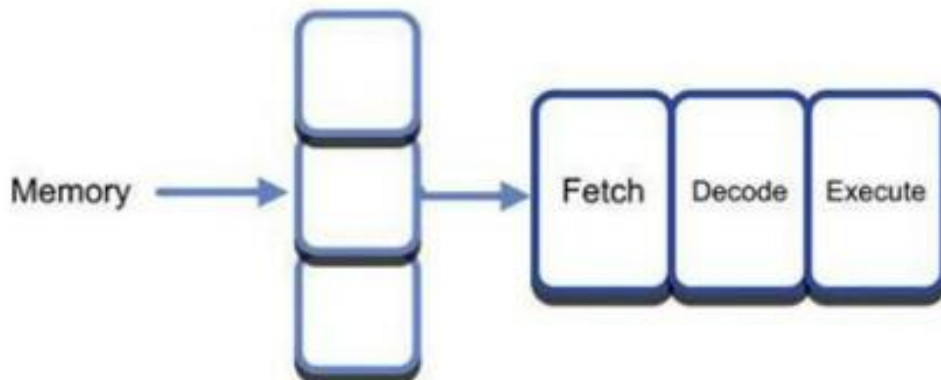
**Hình 2.1.** Kiến trúc bộ vi xử lý ARM Cortex-M4

Lõi của Cortex-M4 là một CPU RISC 32-bit được thiết kế dựa trên kiến trúc Harvard (đặc trưng bằng sự tách biệt giữa vùng nhớ chứa dữ liệu và chương trình). Nó là phiên bản đơn giản hóa từ mô hình lập trình của ARM7/9 nhưng có một tập lệnh phong phú và tối ưu hơn, hỗ trợ tốt cho các phép toán số nguyên, khả năng thao tác với bit và đáp ứng thời gian thực tốt hơn.



**Hình 2.2.** Lõi ARM7TDMI của Cortex-M4.

CPU Cortex-M4 có cấu trúc đường ống 3 đoạn (three-stage pipeline). Với kiến trúc này thời gian rỗi của một chu kỳ sẽ được hạn chế tối đa nên hiệu suất tổng thể của nó được cải thiện rõ rệt.



**Hình 2.3.** Kiến trúc đường ống của ARM Cortex-M4

Bình thường Cortex-M4 có thể thực thi hầu hết lệnh trong một chu trình đơn, nhưng với cấu trúc pipeline này Cortex-M4 có thêm khả năng dự đoán rẽ nhánh nên lệnh có thể được thực thi nhanh hơn. Tức là nếu với một lệnh thường A đang được thực thi thì một lệnh B khác sẽ được giải mã và một lệnh C khác nữa sẽ được lấy về bộ nhớ. Còn đối với lệnh rẽ nhánh (if ... else chẳng hạn), trong khi lệnh điều kiện được giải mã ở tầng decode thì bộ xử lý sẽ nạp lệnh dự định rẽ nhánh trong else, khi



lệnh rẽ được thực thi, bộ xử lý sẽ phân tích xem đâu là lệnh tiếp theo. Nếu lệnh rẽ nhánh được chọn thì nó sẽ được thực thi ngay (vì nó đang ở tầng decode), còn nếu không thì bộ vi xử lý sẽ thực hiện tiếp lệnh tiếp theo (đã được lấy ở tầng fetch).

Lỗi Cortex-M4 chỉ hỗ trợ tập lệnh Thumb-2. So với kiến trúc tập lệnh ARM hoặc Thumb thì Thumb-2 tối ưu hơn về hiệu suất và kích thước mã, bao gồm cả bộ phận phần cứng, nhân chu trình đơn, và thao tác bit-lĩnh vực.

Ngoài ra, Cortex-M4 còn được tích hợp chế độ ngủ (sleep mode) và tùy chọn khả năng duy trì trạng thái (optional state retention capabilities) cho phép đạt hiệu suất cao với mức tiêu thụ điện năng thấp. [2]

### 2.1.2 Vi điều khiển STM32F407

#### a . Cấu trúc bộ nhớ và bus.

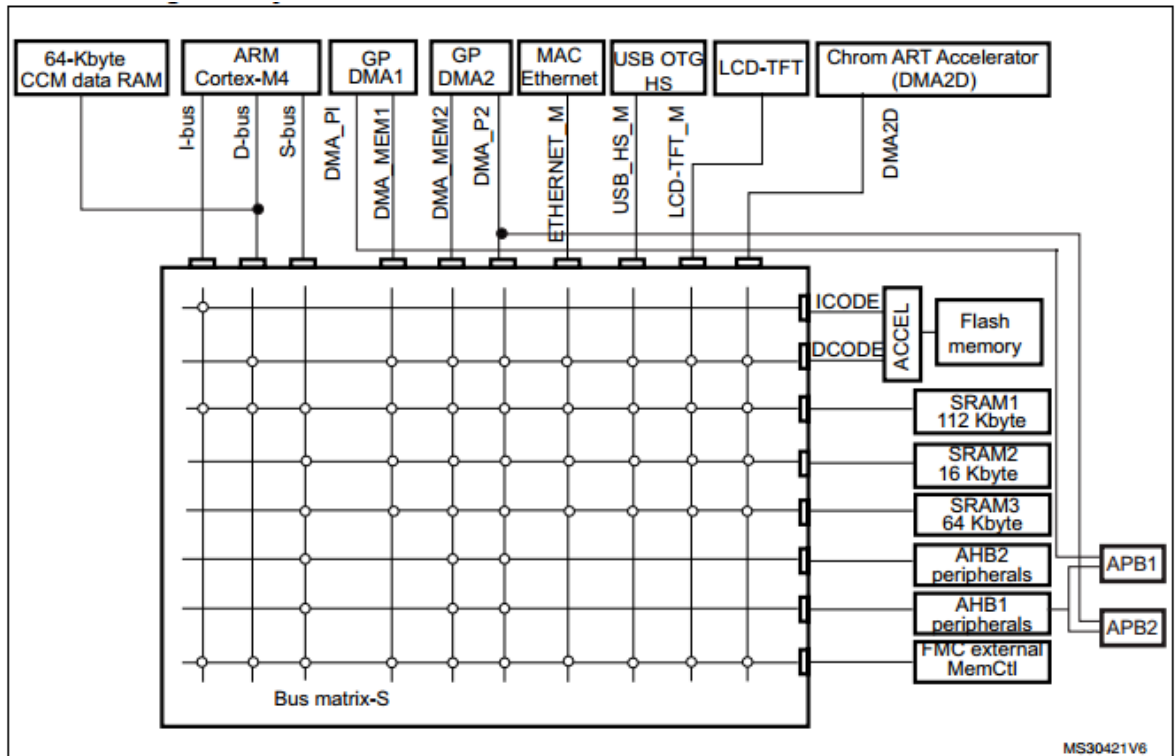
##### ❖ Cấu trúc hệ thống bus:

##### - Tám bus chủ:

- Cortex®-M4 với lõi FPU I-bus, D-bus và S-bus
- DMA1 memory bus
- DMA2 memory bus
- DMA2 peripheral bus
- Ethernet DMA bus
- USB OTG HS DMA bus

##### - Bảy bus tớ:

- Bộ nhớ flash nội Icode bus
- Bộ nhớ flash nội Dcode bus
- Bộ nhớ nội SRAM(bộ nhớ truy xuất ngẫu nhiên tĩnh ) 1 là 112KB (chính)
- Bộ nhớ nội SRAM2 16KB (phụ)
- AHB1 peripherals
- AHB2 peripherals
- FMC



**Hình 2.4.** Cấu trúc bus hệ thống

### ❖ Cấu trúc bộ nhớ.

Bộ nhớ chương trình, bộ nhớ dữ liệu, thanh ghi và cổng I/O được tổ chức trong phạm vi 4 GB. Bộ nhớ bắt đầu từ địa chỉ 0x00000000. SRAM bắt đầu từ địa chỉ 0x20000000 và tất cả SRAM nội đều được bố trí ở điểm bắt đầu vùng bit band. Vùng nhớ thiết bị ngoại vi bắt đầu từ địa chỉ 0x40000000 và ở vùng nhớ bit band. Các thanh ghi điều khiển của nhân Cortex bắt đầu từ địa chỉ 0xE0000000.

Vùng nhớ dành cho FLASH được chia nhỏ thành ba vùng. Vùng thứ nhất gọi là User Flash bắt đầu từ địa chỉ 0x00000000. Kế tiếp là System Memory hay còn gọi là vùng nhớ lớn. Vùng này có độ lớn 4Kbytes thông thường sẽ được nhà sản xuất cài đặt bootloader. Cuối cùng là vùng nhớ nhỏ bắt đầu từ địa chỉ 0x1FFFFFF80 chứa thông tin cấu hình dành cho STM32. Bootloader thường được dùng để tải chương trình thông qua cổng USART1 và chứa ở vùng User Flash.

Để kích hoạt bootloader của STM32 người dùng phải thiết lập các chân BOOT0 và BOOT1 ở mức điện áp thấp và cao tương ứng. Khi đó, sau khi STM32 được khởi động chương trình sẽ đặt bootloader vào địa chỉ 0x00000000 và thực thi nó thay thực thi chương trình của người dùng ở User Flash. Để giao tiếp với bootloader, ST cung

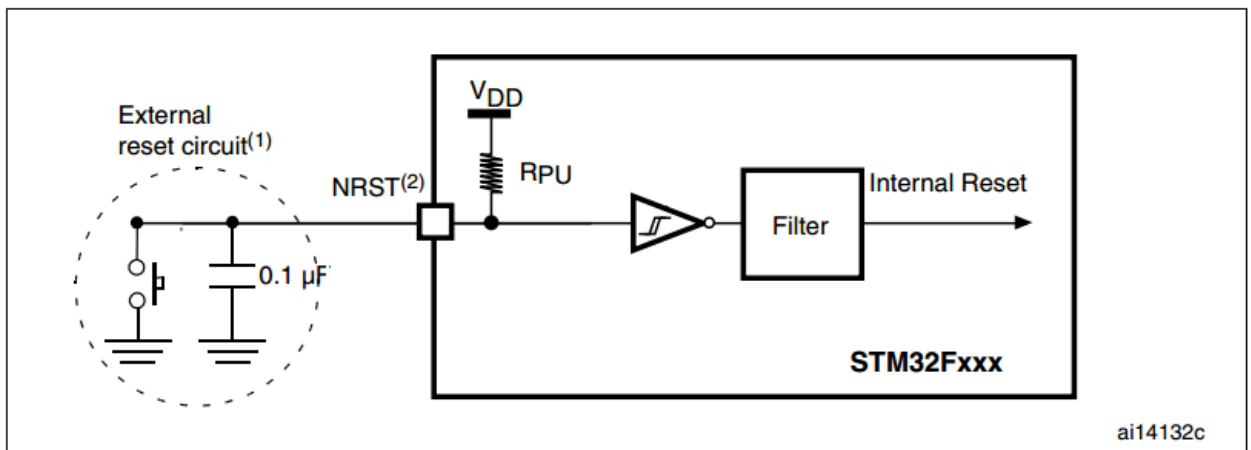
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

cấp một chương trình chạy ở PC, chương trình này có khả năng ghi, xóa vùng nhớ ở User Flash. Ngoài ra chúng ta có thể cấu hình các chân bootpins đưa SRAM nội vào địa chỉ 0x00000000, cho phép tải xuống và thực thi chương trình ngay tại SRAM. Điều này làm tăng tốc độ tải chương trình và hạn chế số lần ghi vào Flash .

### b . Reset And Clock Control

#### ❖ Bộ reset của vi điều khiển.

STM32 có nhiều nguồn Reset khác nhau ngoài đường Reset bên ngoài. STM32 có thể bị buộc khởi động lại từ: các bộ Watchdogs nội, một Reset mềm thông qua NVIC, bộ Reset mở / tắt nguồn nội và mạch phát hiện điện áp nguồn thấp. Nếu một tín hiệu Reset xuất hiện, một bộ cờ trong thanh ghi kiểm soát và trạng thái RCC có thể được đọc để xác định nguyên nhân gây ra Reset. Trạng thái của những cờ này vẫn còn tồn tại cho đến khi hệ thống được cấp nguồn trở lại hoặc cho đến khi người dùng thiết lập bit Remove Reset.



Hình 2.5. Mạch reset.

#### ❖ Xung clock của vi điều khiển STM32F407.

Có 3 loại xung clock có thể được sử dụng để làm xung clock hệ thống (SYSCLK-xung clock cho khối xử lý).

- HSI( High Speed Internal-16MHZ) nguồn xung clock tốc độ cao bên trong Arm.
- HSE( High Speed External) nguồn xung clock tốc độ cao bên ngoài Arm.
- PLL(Phase Locked Loop) nguồn xung clock lấy từ bộ nhân tần số( hay sử dụng).

Ngoài ra Arm còn có các nguồn clock phụ khác như:

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

- LSI( Low speed internal) 32Khz dùng để cấp cho independent watchdog hay RTC dùng để định thời.

- LSE(Low Speed External) nguồn xung clock tốc độ chậm thường được nối với thạch anh 32.768 kHz từ bên ngoài, xung clock này có thể được dùng để cấp cho RTC.

Trong hầu hết các ứng dụng, xung clock dùng cho bộ xử lý Cortex và các thiết bị ngoại vi được lấy từ nguồn dao động bên ngoài (HSE và LSE). Nếu thạch anh ngoại có vấn đề nó sẽ sử dụng bộ dao động nội (LSI) 16Mhz.

### c . GPIO và AFIO.

#### ❖ GPIO của vi điều khiển.

STM32F407VET6 có 5 cổng đa dụng với 80 chân điều khiển. Mỗi chân điều khiển có thể cấu hình như là GPIO hoặc có chức năng thay thế khác, hoặc mỗi chân có thể cùng lúc là nguồn ngắt ngoại.

Các cổng GPIO được đánh số A->E và mức tiêu thụ ở 5V. Nhiều chân có thể được cấu hình như là input/output tương tác với các thiết bị ngoại vi riêng của người dùng như USART hoặc I2C... Mỗi port I/O có hai thanh ghi cấu hình 32 bit (GPIOx\_CRL,GPIOx\_CRH),có hai thanh ghi dữ liệu 32 bit (GPIOx\_IDR, GPIOx\_ODR), thanh ghi set / reset 32 bit GPIOx\_BSRR và thanh ghi reset 16 bit GPIOx\_BRR. Tùy vào cách sử dụng mà GPIO được cấu hình là ngõ vào ,ngõ ra , ngõ vào tương tự ,hoặc vừa ngõ vào vừa ngõ ra... Dưới đây là bảng cấu hình của GPIO.

#### ❖ AFIO của vi điều khiển

Chức năng thay thế cho phép người dùng sử dụng các cổng GPIO với các ngoại vi khác. Để thuận tiện cho thiết kế phần cứng một thiết bị ngoại vi có thể được ánh xạ tới một hay nhiều chân của vi xử lý.

Sử dụng các tính năng thay thế của STM32 được điều khiển bởi các thanh ghi “Remap & Debug I/O”. Mỗi thiết bị ngoại vi (USART, CAN, Timers, I2C và SPI) có 1 hoặc 2 trường bit điều khiển ánh xạ đến các chân của vi điều khiển. Một khi các chân được cấu hình sử dụng chức năng thay thế, các thanh ghi điều khiển GPIO sẽ được sử dụng để điều khiển chức năng thay thế thay vì tác vụ I/O. Các thanh ghi

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Remap có điều khiển bộ JTAG. Khi hệ thống khởi động, cổng JTAG được kích hoạt tuy nhiên chức năng theo dõi dữ liệu (data trace) vẫn chưa khởi động. JTAG có thể chuyển qua chế độ Debug, xuất dữ liệu theo dõi ra ngoài hoặc đơn giản chỉ sử dụng cổng GPIO.

### d . Ngắt.

Ngắt là việc vi điều khiển dừng tạm thời chương trình đang thực thi mà đi thực thi một chương trình khác có yêu cầu cấp thiết hơn. Sau khi thực thi xong CPU sẽ quay trở về thực hiện lại chương trình cũ.

NVIC (Nested Vector Interrupt Controller – Bộ điều khiển vector ngắt lồng nhau) là một khối nằm trong lõi Cortex có nhiệm vụ quản lý các nguồn ngắt, do có rất nhiều nguồn có khả năng yêu cầu CPU ngắt nên chúng cần có một “**người quản lý**” chung để sắp xếp, điều khiển mọi hoạt động ngắt được diễn ra đúng theo ý muốn tránh trường hợp chồng chéo, xung đột giữa các yêu cầu

Để sử dụng NVIC cần phải qua ba bước cấu hình. Đầu tiên cấu hình bảng vector cho các nguồn ngắt mà ta muốn sử dụng. Tiếp theo cấu hình bảng thanh ghi NVIC để cho phép và thiết lập các mức ưu tiên của các ngắt trong NVIC và cuối cùng cần phải cấu hình các thiết bị ngoại vi và cho phép ngắt tương ứng. Bảng vector ngắt của Cortex bắt đầu ở vùng dưới của bảng địa chỉ. Tuy nhiên bảng vector bắt đầu từ địa chỉ 0x00000004 thay vì là 0x00000000 như ARM7 và ARM9, bốn byte đầu tiên được sử dụng để lưu trữ địa chỉ bắt đầu của con trỏ ngăn xếp.

Mỗi vector ngắt có độ rộng 4 byte và giữ địa chỉ bắt đầu của chương trình phục vụ ngắt tương ứng, 15 vector ngắt đầu tiên là các ngắt đặt biệt chỉ xảy ra trong lõi Cortex. Các ngắt ngoại vi người dùng bắt đầu từ vector 16, được định nghĩa bởi nhà sản xuất và được liên kết đến thiết bị ngoại vi. Trong phần mềm, bảng vector thường được giữ trong chương trình khởi động bằng cách định vị các địa chỉ trình phục vụ ngắt tại địa chỉ nền của bộ nhớ.

### e . Giao diện ngoại vi nối tiếp(SPI).

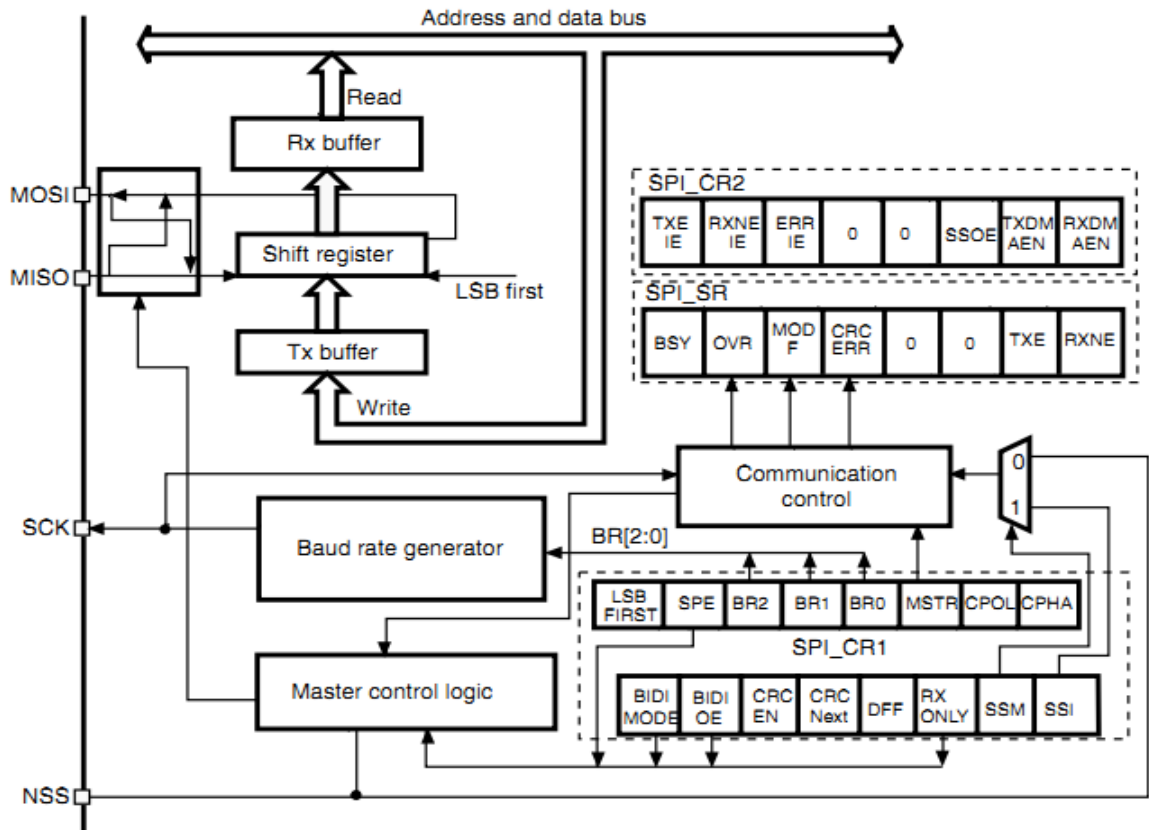
Giao diện ngoại vi nối tiếp (SPI) là một chuẩn đồng bộ nối tiếp để truyền dữ liệu ở chế độ song công toàn phần full-duplex (hai chiều, hai phía), do công ty Motorola thiết kế nhằm đảm bảo sự liên hợp giữa các vi điều khiển và thiết bị ngoại

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

---

vi một cách đơn giản và giá rẻ. Trong giao diện SPI sử dụng truyền dẫn 4 dây MOSI, MISO, CLK, CS.

Khối SPI tốc độ cao kết nối với Bus ngoại vi APB2, khối SPI tốc độ thấp kết nối với Bus ngoại vi APB1 . Mỗi SPI có thanh ghi cấu hình độc lập, dữ liệu truyền có thể là 8 bit hoặc 16 bit, thứ tự hỗ trợ MSB hay LSB. Có thể cấu hình nó đóng vai trò là Master hay Slave. Để hỗ trợ truyền dữ liệu tốc độ cao, mỗi khối SPI có hỗ trợ thêm 2 kênh DMA dành cho gửi và truyền nhận dữ liệu. Thêm vào khối CRC dành cho cả nhận và truyền nhận dữ liệu. Khối kiểm tra lỗi CRC có thể hỗ trợ CRC8 hoặc CRC16 ...



**Hình 2.6.** Sơ đồ khối SPI.

Trong giao diện SPI có bốn tín hiệu số:

MOSI hay SI – cổng ra của bên chủ ( Master Out Slave IN). Đây là chân dành cho việc truyền tín hiệu từ thiết bị chủ động đến thiết bị bị động.

MISO hay SO – Cổng ra bên tớ (Master IN Slave Out). Đây là chân dành cho việc truyền dữ liệu từ tớ đến chủ.

SCLK hay SCK là tín hiệu clock đồng bộ (Serial Clock). Xung nhịp chỉ được tạo bởi Master.

NSS: đây là một chân tùy chọn để chọn chế độ chủ / tớ. Chân này hoạt động như một "chip chọn" để cho các SPI chủ giao tiếp với các thiết bị tớ riêng và để tránh chồng dữ liệu. Đầu vào NSS của thiết bị tớ có thể được điều khiển bởi các cổng I/O trên thiết bị chủ. Chân NSS cũng có thể được sử dụng như một đầu ra nếu được kích hoạt (SSOE bit) và xuống mức thấp nếu SPI là trong cấu hình chính. Theo cách này, tất cả các chân NSS từ thiết bị tớ được kết nối với chân NSS của thiết bị chủ đều

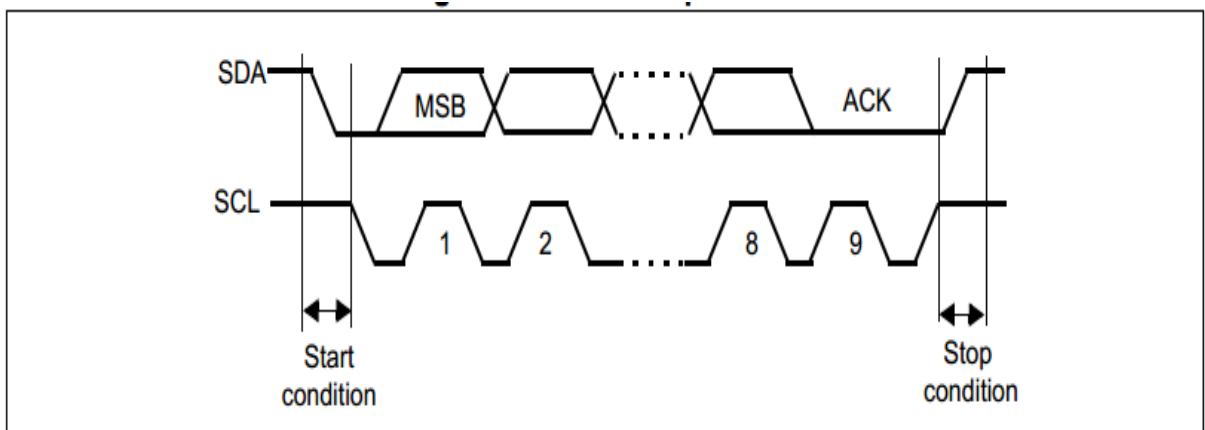
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

được đưa xuống mức thấp và trở thành thiết bị tớ khi chúng được cấu hình trong chế độ điều khiển bằng phần cứng.

### f. I2C

Giao tiếp I2C là giao thức ưu tiên truyền thông nối tiếp được phát triển bởi Philips Semiconductor. Một giao tiếp I2C gồm 2 dây: SDA là đường truyền dữ liệu hai hướng, SCL là đường truyền xung đồng bộ. Truyền với tốc độ tiêu chuẩn là 100KHz và tốc độ cao là 400KHz. Giao tiếp I2C hoạt động chế độ một chủ một tớ hoặc một chủ nhiều tớ

Bên chủ sẽ truyền tín hiệu Start đến bên tớ, sau đó nó sẽ gửi 7 bit dữ liệu chứa địa chỉ của bên tớ mà nó muốn giao tiếp, theo thứ tự các bit có trọng số lớn nhất MSB sẽ truyền trước, bit thứ tám để quy định là nhận bit "0" hay gửi dữ liệu bit "1" của thiết bị tớ, tiếp theo là bit ACK để xác nhận thiết bị có nhận được 1 byte hay không, cuối cùng là tín hiệu Stop để kết thúc quá trình truyền.



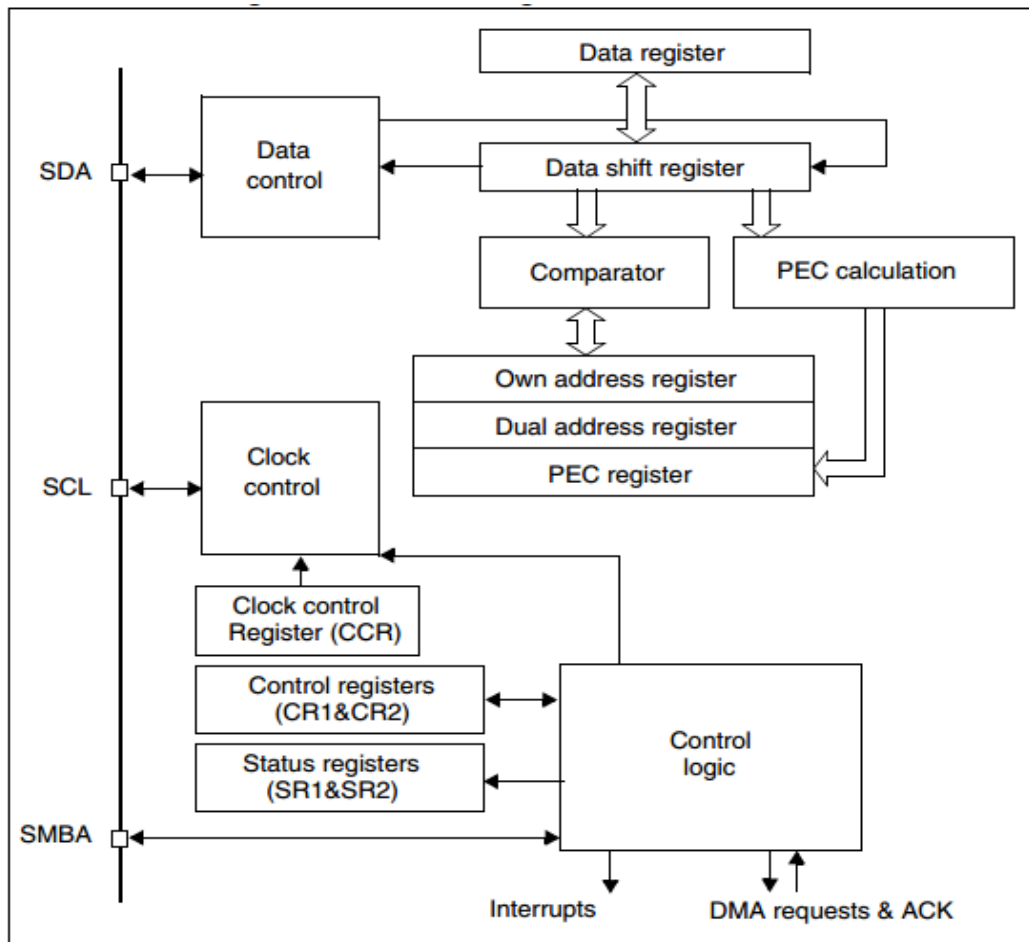
Hình 2.7. Phương thức truyền I2C.

I2C trong STM32 hỗ trợ nhằm giao tiếp với các mạch tích hợp ngoài. Giao diện I2C có thể được cấu hình hoạt động ở chế độ slave, master hay đóng vai trò bộ phân xử đường trong hệ thống multi-master. Dao diện I2C hỗ trợ tốc độ truyền chuẩn 100kHz hay tốc độ cao 400kHz. Ngoài ra còn hỗ trợ 7 hoặc 10 bit địa chỉ. Được thiết kế nhằm đơn giản hóa quá trình trao đổi với 2 kênh DMA cho truyền và nhận dữ liệu. Hai ngắt cho 1 nhân Cortex, một cho địa chỉ và truyền nhận. Thêm nữa, để đảm bảo tính chính xác dữ liệu truyền, khối kiểm tra lỗi dữ liệu (PAC- packet error checking)



## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

được tính hợp thêm vào dao điện I2C cho phép kiểm tra mã CRC- 8 bit. Thao tác này được thực hiện hoàn toàn bởi phần cứng.



Hình 2.8. Sơ đồ khối I2C.

### g . Bộ điều khiển cho phép truy xuất linh hoạt bộ nhớ tĩnh (FSMC).

FSMC là giao diện truyền dữ liệu 16 bit đồng bộ và bất đồng bộ giữa vi điều khiển với các loại bộ nhớ bên ngoài.[3]

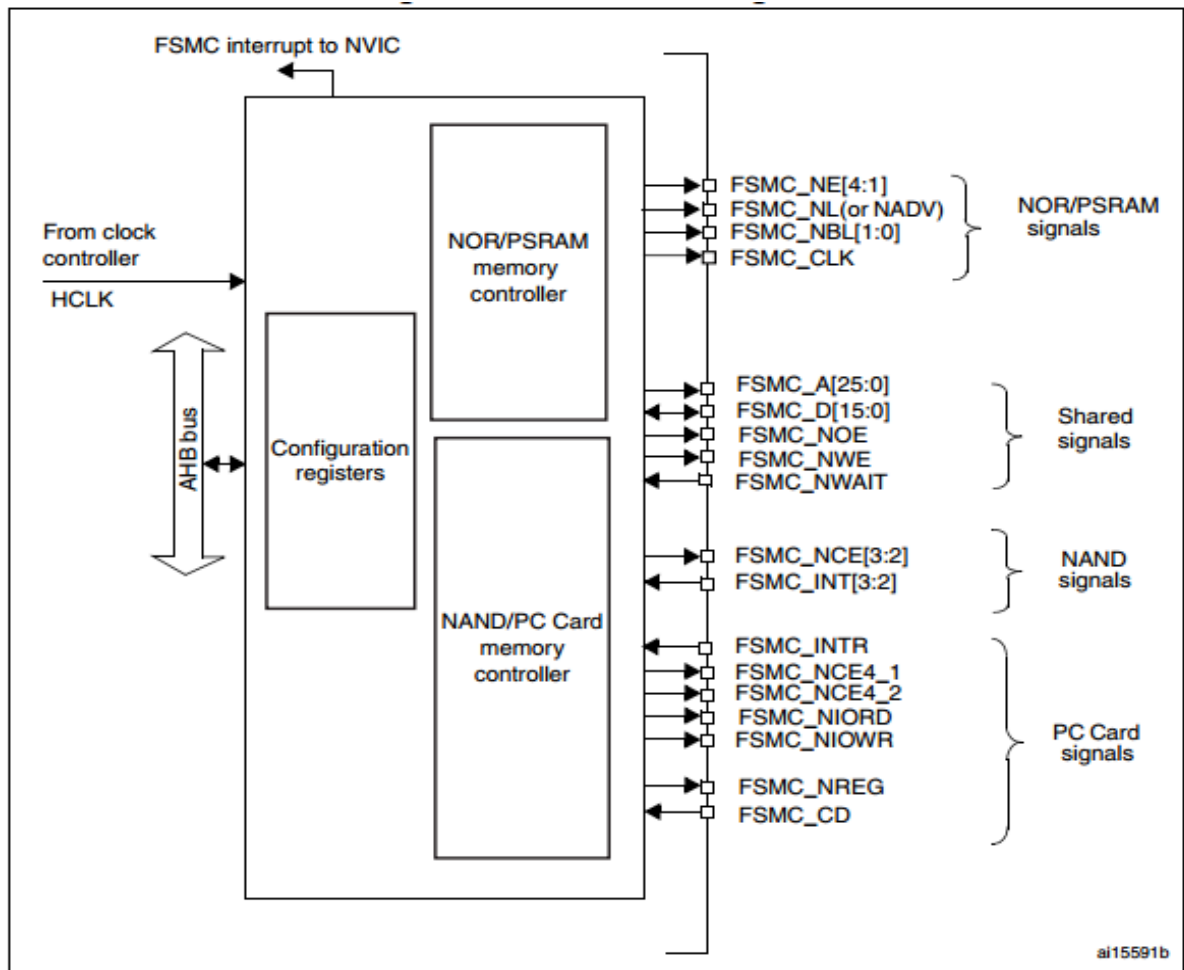
FSMC có những đặc điểm sau:

- Giao tiếp truyền dữ liệu giữa AHB với các loại bộ nhớ ngoại có giao thức phù hợp.
- Đáp ứng được các yêu cầu về thời gian truy xuất của các loại bộ nhớ ngoại.
- Tất cả các bộ nhớ ngoại chia sẻ địa chỉ, dữ liệu và tín hiệu điều khiển. Do đó để chọn bộ nhớ cần truy xuất ta phải sử dụng thêm tín hiệu chọn chip(

chip select) riêng biệt cho từng bộ nhớ. Tại một thời điểm khối FSMC chỉ có thể giao tiếp với một bộ nhớ và chỉ cho phép một truy xuất xảy ra.

Các đặc tính FSMC của dòng vi điều khiển STM32F40X.

- Hỗ trợ giao tiếp với các loại bộ nhớ tĩnh:
  - Ram tĩnh (SRAM – static random access memory).
  - NOR Flash
  - PSRAM ( 4 bank bộ nhớ).
- Hai bank NAND Flash hỗ trợ ECC có thể kiểm tra tới 8 Kbyte dữ liệu.
- Hỗ trợ giao tiếp PC Card 16 bit.
- Hỗ trợ chế độ truy xuất burst cho phép đồng bộ các thiết bị( ví dụ giữa NOR Flash và PSRAM)
- Cho phép cài đặt chế độ 8 hoặc 16 bit dữ liệu.
- Các tín hiệu điều khiển chip select được thiết kế độc lập cho từng bank bộ nhớ.
- Cho phép lập trình thời gian trễ cho phù hợp với nhiều loại bộ nhớ như là:
  - Thời gian ở trạng thái chờ ( tối đa là 15)
  - Thời gian quay vòng của chu kỳ bus ( tối đa 15)
  - Thời gian trễ trước khi cho phép xuất dữ liệu và ghi dữ liệu( tối đa 15)
  - Thời gian trễ cho việc đọc, ghi và giao thức truyền nhận.
- Cho phép truy xuất dữ liệu theo byte khi dùng PSRAM và SRAM.
- Chuyển đổi từ bus 32 bit AHB sang 8 hoặc 16 bit nối tiếp để có thể truy xuất được các loại bộ nhớ ngoại 8 hoặc 16 bit.
- Hỗ trợ FIFO( First In First Out) dài 2 word mỗi word gồm 32 bit làm bộ đệm giúp giải phóng AHB khi truy xuất bộ nhớ tốc độ thấp.
- Hỗ trợ giao thức truyền bất đồng bộ.



**Hình 2.9.** Sơ đồ khối FSMC.

FSMC chia vùng địa chỉ giao tiếp với các bộ nhớ ngoài ra làm 4 bank mỗi bank có kích thước 256Mbyte.

Bank 1 được dùng để giao tiếp tới 4 bộ nhớ NOR Flash hay PSRAM. Bank này được chia ra thành 4 bank nhỏ mỗi bank nhỏ có kích thước 64 Mbyte như sau:

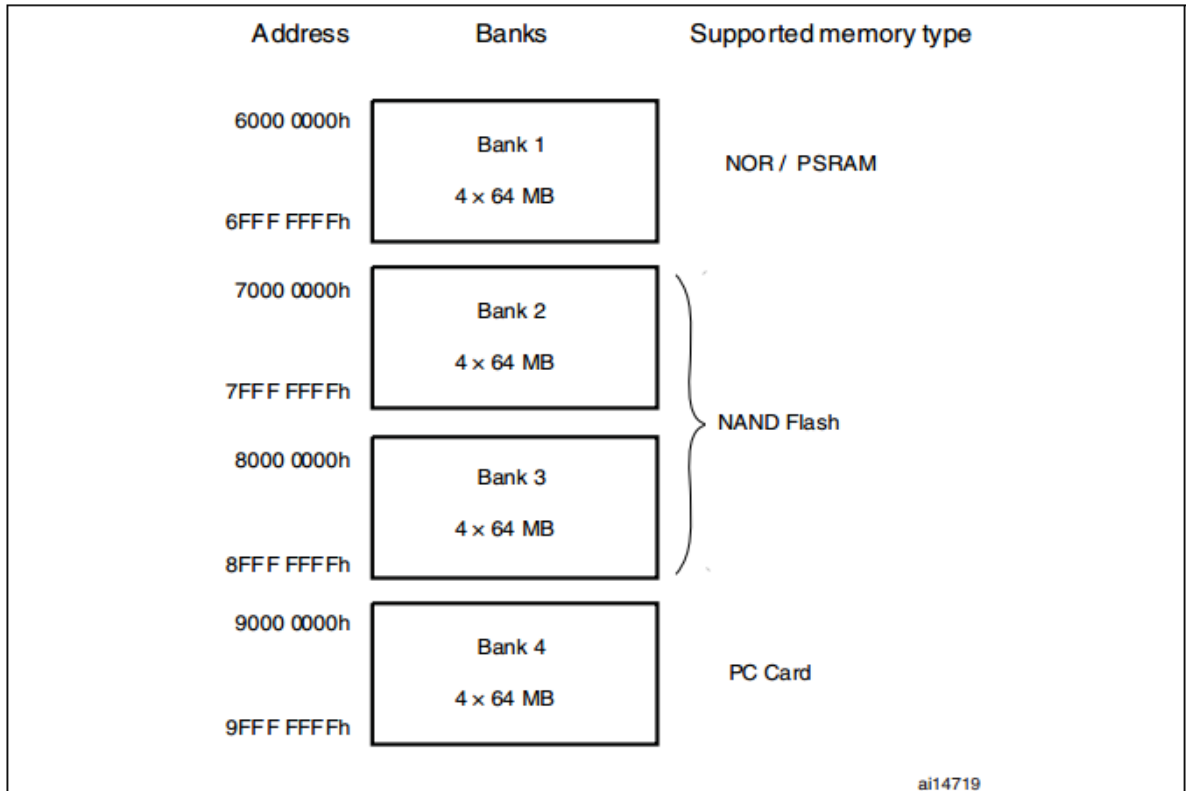
- Bank 1 – NOR/PSRAM1 – Địa chỉ nền 6000 0000H – Được chọn bởi NE1.
- Bank 1 – NOR/PSRAM2 – Địa chỉ nền 6400 0000H – Được chọn bởi NE2.
- Bank 1 – NOR/PSRAM3 – Địa chỉ nền 6800 0000H – Được chọn bởi NE3.
- Bank 1 – NOR/PSRAM4 – Địa chỉ nền 6C00 0000H – Được chọn bởi NE4.

Bank 2 và 3 được dùng để giao tiếp với bộ nhớ NAND Flash (mỗi bộ nhớ dùng 1 bank).

Bank 4 được dùng để giao tiếp với PC Card

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

**Chú ý:** Khi sử dụng mỗi bank ta phải khai báo loại bộ nhớ mà ta đang sử dụng thông qua thanh ghi cấu hình. Khi giao tiếp với bộ nhớ có bề rộng dữ liệu là 16 bit thì FSMC sẽ sử dụng các đường địa chỉ nội của AHB là HADDR[25:1] để tạo thành các đường địa chỉ để giao tiếp với bộ nhớ ngoại là FSMC\_A[24:0]. Bất chấp bề rộng dữ liệu của bộ nhớ là 8 bit hay 16 bit thì khi kết nối ta luôn phải kết nối FSMC\_A[0] với đường địa chỉ A[0] của bộ nhớ.



**Hình 2.10.** Tổ chức địa chỉ của FSMC.

**Bảng 2.1.** Các tín hiệu FSMC sử dụng để giao tiếp với NOR Flash.

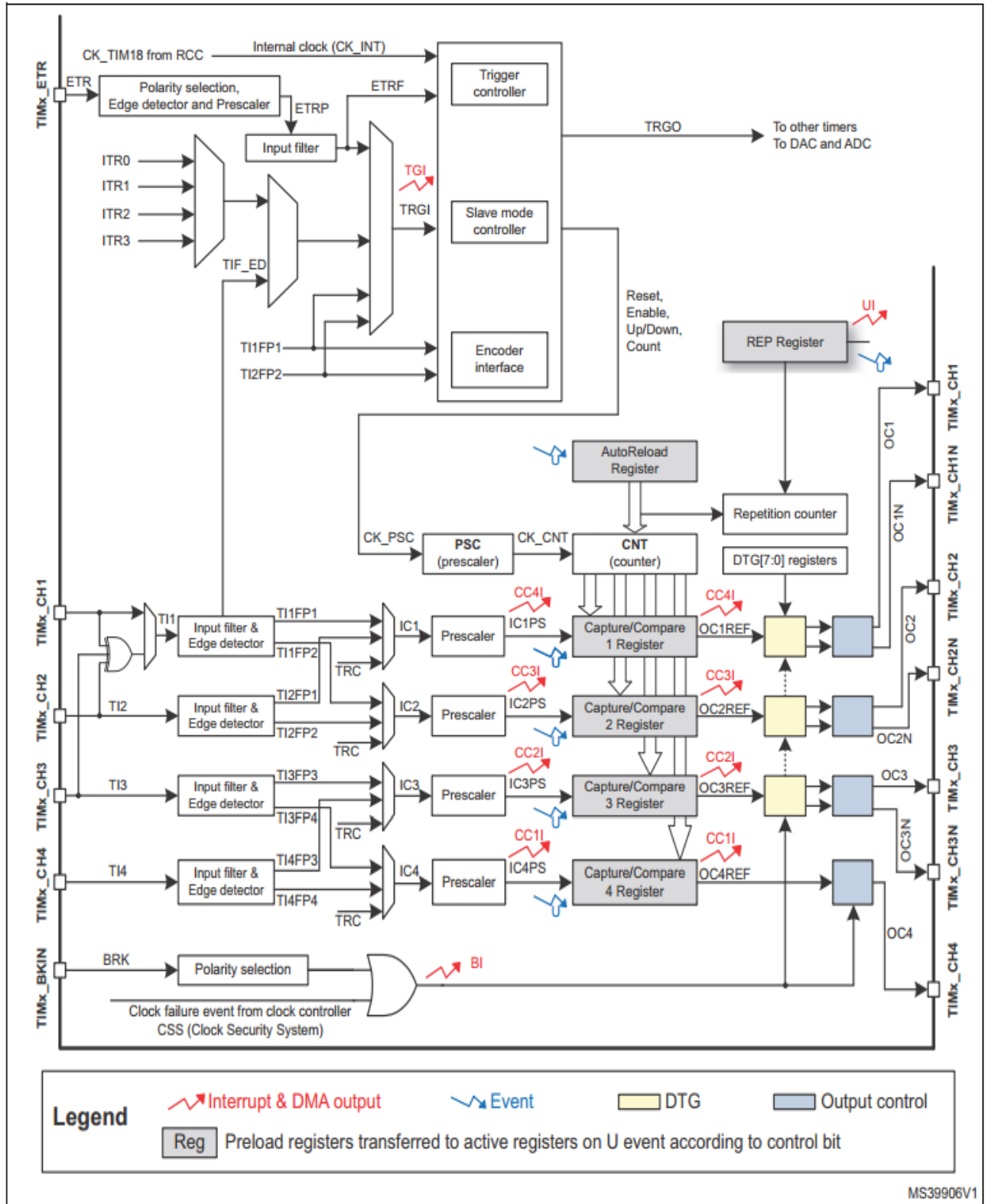
Tên các tín hiệu FSMC	I/O	Chức năng
CLK	O	Cấp xung clock cho chế độ truy suất đồng bộ
A[25:16]	O	Bus địa chỉ
AD[15:0]	I/O	16 đường dữ liệu khi hoạt động ở chế độ không đa hợp.

		Khi hoạt động ở chế độ đa hợp thì đây là 16 đường địa chỉ/ dữ liệu
NE[x]	O	Tín hiệu chọn chip(Ship select),x=1..4
NOE	O	Cho phép ngõ ra(Output enable)
NWE	O	Cho phép ghi( write enable)
NL(=NADV)	O	Cho phép chốt( Latch enable), tín hiệu này được dùng để báo là địa chỉ đã sẵn sàng đối với một số loại NOR Flash
NWAIT	I	Tín hiệu yêu cầu chờ từ NOR Flash.

**h . Bộ định thời (Timer)**

STM32 có bốn khối định thời. Timer 1 là khối nâng cao dành cho điều khiển động cơ, 3 khối còn lại đảm nhiệm chức năng đa nhiệm. Tất cả chúng đều có chung kiến trúc, khối nâng cao sẽ có thêm các đặc tính phần cứng riêng.

Tất cả các bộ TIMER đều gồm bộ đếm 16-bit với thanh ghi chia tần số dao động 16-bit (prescaler) và thanh ghi tự nạp (auto-reload). Bộ đếm của khối định thời có thể được cấu hình để đếm lên, đếm xuống hay trung tính (lên xuống xen kẽ nhau). Xung nhịp cho đồng hồ có thể được lựa chọn dựa trên 8 nguồn khác nhau: từ đồng hồ chuyên biệt được lấy từ đồng hồ hệ thống, từ xung nhịp chân ra lấy từ khối định thời khác, hoặc từ nguồn xung nhịp ngoại. Khối định thời sử dụng cổng chọn để lấy xung nhịp đầu vào thích hợp, người dùng có thể sử dụng chân ETR để điều khiển cổng chọn này.

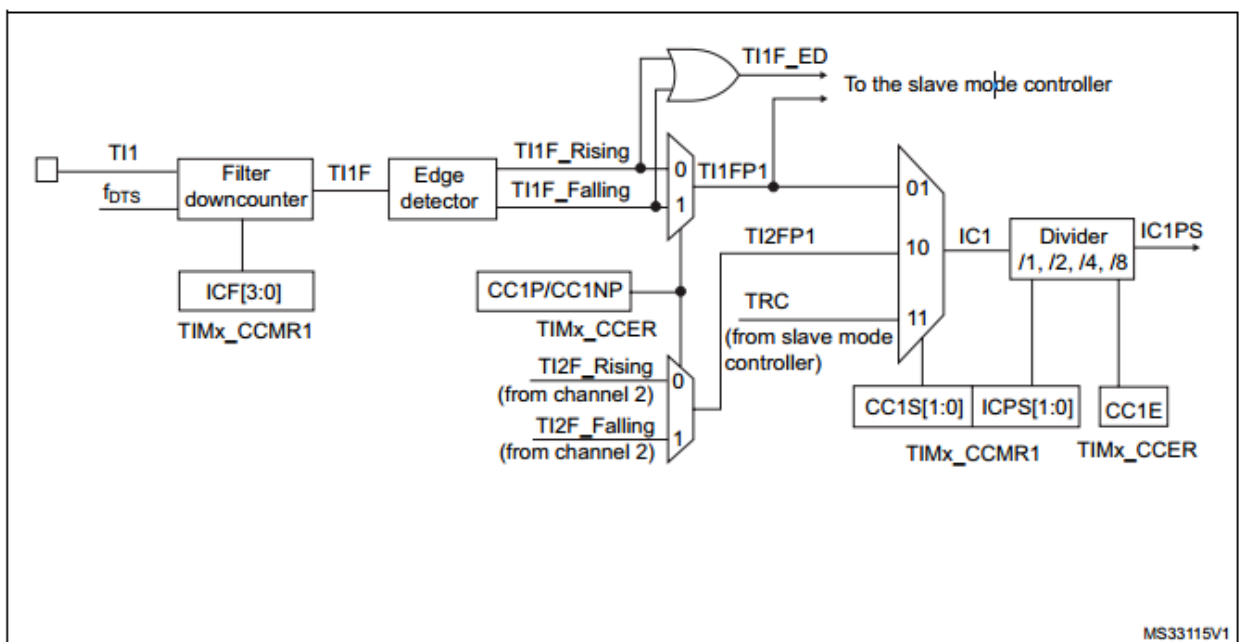


Hình 2.11. Sơ đồ khối Timer 1.

❖ **Kênh capture/compare**

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

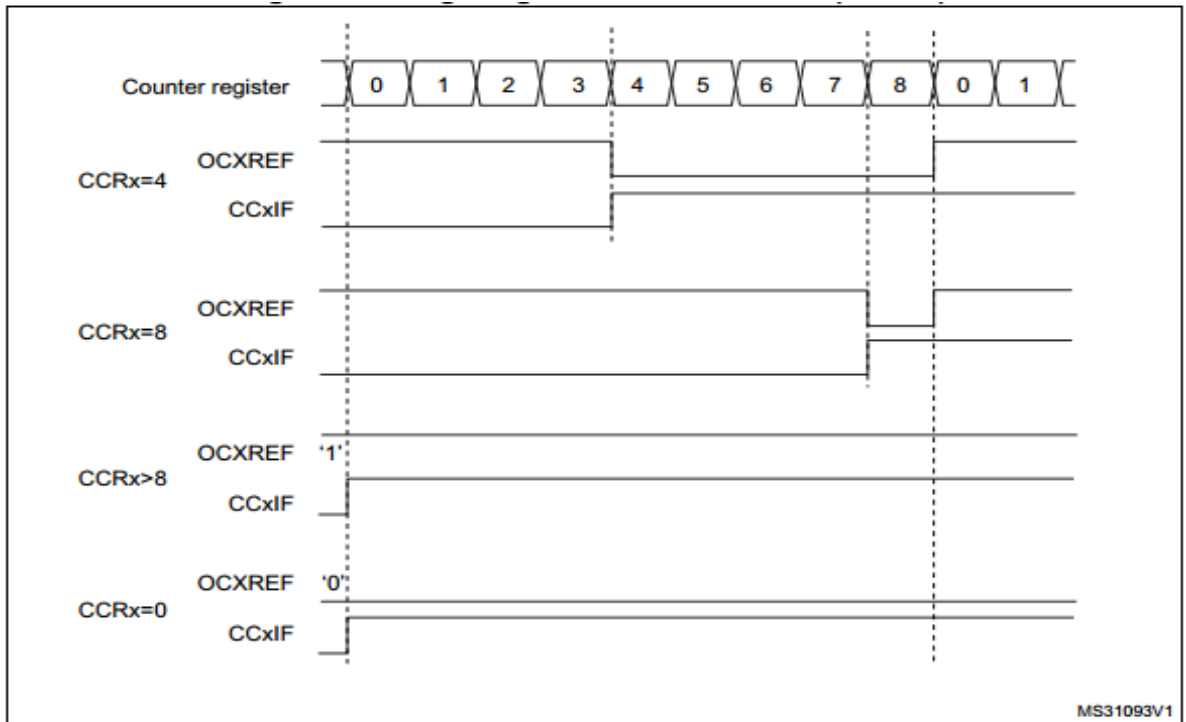
Mỗi kênh Capture/ Compare được điều khiển bởi duy nhất một thanh ghi. Chức năng của thanh ghi này có thể thay đổi tùy thuộc cấu hình. Ở chế độ Capture, thanh ghi này có nhóm các bit đảm nhiệm thiết bị lọc dữ liệu đầu vào và chế độ đánh giá các ngõ PWM. Một khối Capture cơ bản gồm có bốn kênh vào để cấu hình bộ phát hiện xung (Edge Detector). Khi một xung cạnh lên hay xung cạnh xuống được phát hiện, bộ đếm hiện thời sẽ được cập nhập vào các thanh ghi 16 bit Capture/ Compare. Khi sự kiện Capture xảy ra bộ đếm có thể được khởi động lại hay tạm ngưng. Một ngắt DMA có thể được sử dụng trong trường hợp này. Ở chế độ Compare, STM32 cung cấp hàm chuẩn so sánh và bộ tạo xung PWM.



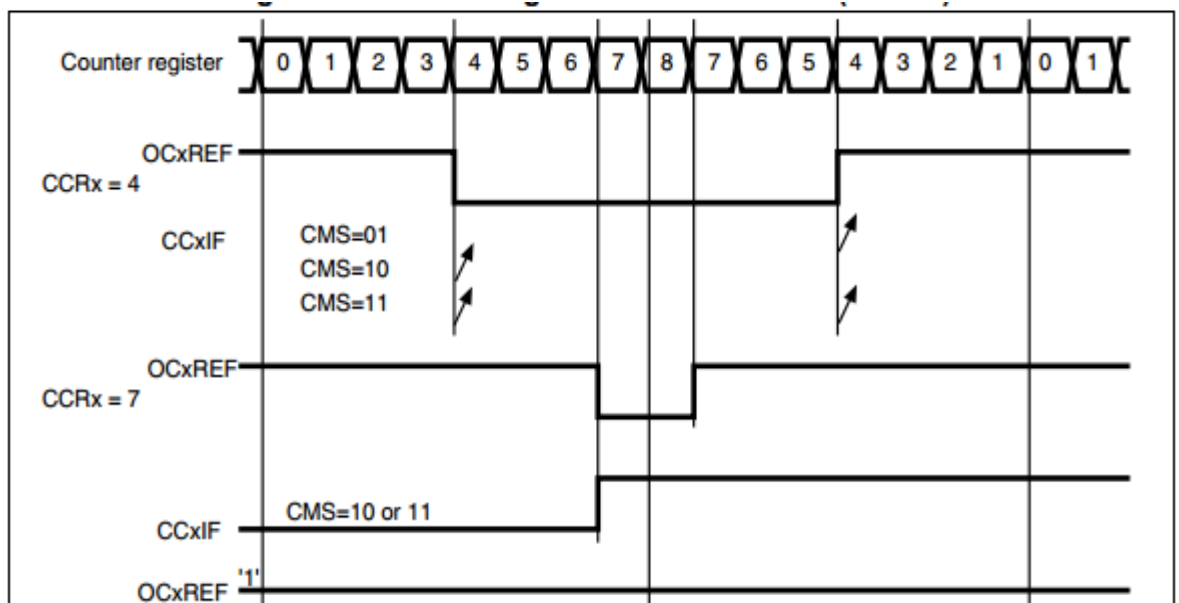
Hình 2.12. Chế độ capture/compare kênh 1.

### ❖ Chế độ điều khiển độ rộng xung (PWM).

Mỗi khối Timer đều có khả năng tạo xung nhịp PWM. Ở chế độ xung PWM, giá trị Period được lưu vào thanh ghi Auto Reload còn giá trị Duty được lưu trong thanh ghi Capture/Compare. Có 2 kiểu tạo PWM, một là canh lề (edge-aligned) và canh lề giữa(center-aligned). Với Edge-aligned cạnh xuống của tín hiệu trùng với thời điểm thanh ghi reload cập nhập lại giá trị. Với center-aligned thời điểm thanh ghi reload cập nhập lại là khoảng giữa của chu kỳ Duty.



Hình 2.13. Dạng sóng Edge-aligned PWM.



Hình 2.14. Dạng sóng Center-aligned PWM.

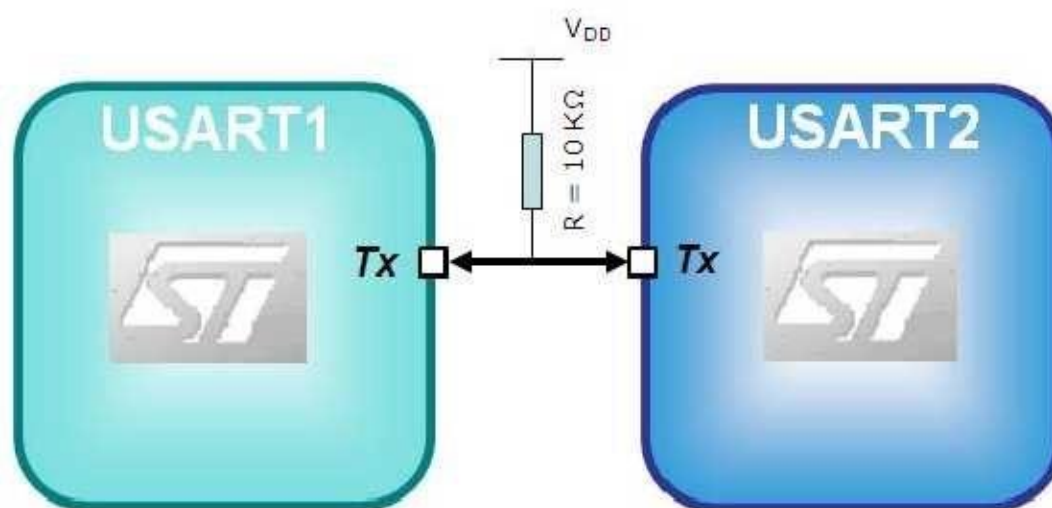


### i. USART

USART(Universal synchronous asynchronous receiver transmitter) là chuẩn truyền nhận nối tiếp đồng bộ. STM32F4 có ba khối USART, mỗi khối có thể hoạt động ở tốc độ 4.5Mbps. Một khối nằm trên APB1 với xung nhịp hoạt động 72MHz, hai khối còn lại nằm trên APB2.

Mạch tích hợp cho phép chia nhỏ tốc độ Baud thành nhiều tốc độ khác nhau thích hợp với nhiều kiểu trao đổi dữ liệu khác nhau. Mỗi khối USART có hai khối DMA dành cho truyền nhận dữ liệu.

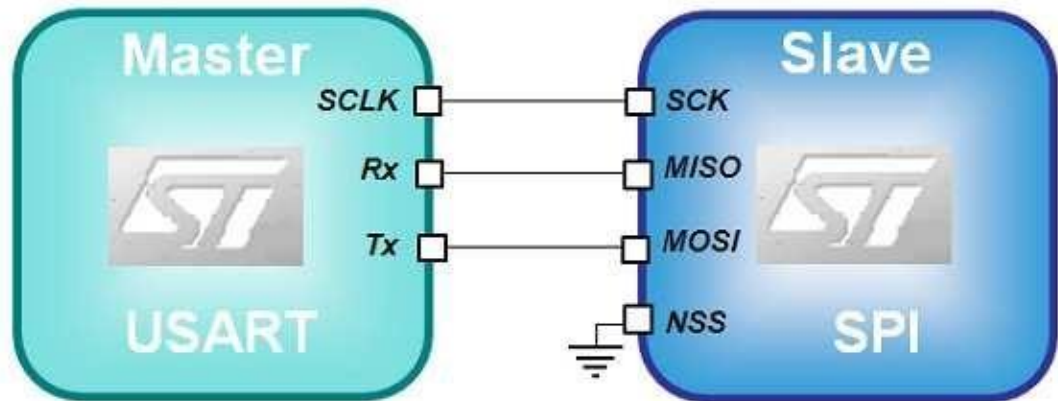
Dữ liệu có thể trao đổi theo chế độ half duplex trên đường trên Tx. Khi hỗ trợ giao tiếp modem và giao tiếp nhiều luồng USART còn cung cấp các tín hiệu điều khiển CTS và RTS.



[www.mbedvn.com]

**Hình 2.15.** Chế độ half duplex của USART.

Người dùng có thể cấu hình khối USART cho các giao tiếp đồng bộ tốc độ cao dựa trên 3 đường tín hiệu riêng biệt như SPI. Khi hoạt động ở chế độ này, khối USART sẽ đóng vai trò là SPI master và có khả năng cấu hình Clock Polarity/Phase nên hoàn toàn có thể giao tiếp với các SPI slave khác.



[www.mbedvn.com]

**Hình 2.16.** Chế độ hỗ trợ giao tiếp SPI của USART.

### 2.2. Module LCD TFT

Module được sử dụng có tên ER-TFTM070-4V2.1, là phiên bản cập nhật của module ER-TFTM070-4. Module kết hợp hiển thị trên màn hình LCD 7 inch và tấm cảm ứng điện dung, với độ phân giải 800x480 được điều khiển bởi vi mạch SSD1963 mang lại hình ảnh sắc nét rõ ràng, góc nhìn rộng.



**Hình 2.17.** Màn hình LCD TFT.

Module hỗ trợ nhiều chuẩn giao tiếp song song 6800, 8080 8 bit /9 bit/16 bit/18 bit/24 bit, tích hợp khe cắm thẻ Micro SD. Tấm cảm ứng được điều khiển bởi IC FT5206. Hoạt động sử dụng nguồn áp thấp từ 1,5V đến 5,25V. Tiêu thụ điện năng thấp dưới 0,75 mW tại 2,7V tỉ lệ lấy mẫu lên đến 125 Khz. Dòng tiêu thụ ở mức điện áp 3.3V là 480mA, ở 5V là 300mA. Là mạch lý tưởng khi sử dụng các nguồn pin. Được ứng dụng phổ biến trong điều khiển giám sát màn hình cảm ứng hay trong các thiết bị đo lường, thiết bị đầu cuối...

### 2.3. Giao tiếp sdcard

#### a. Thẻ nhớ SD

Thẻ nhớ chuẩn SD (Secure Digital) nói chung là định dạng thẻ được phát minh bởi Panasonic và được phát triển bởi liên minh giữa Panasonic, SanDisk và Toshiba vào năm 2000. Hiệp hội này được thành lập nhằm mục tiêu đưa ra tiêu chuẩn thẻ nhớ tốt hơn cho thiết bị cầm tay. Sử dụng dễ dàng trong lưu trữ dữ liệu. Hiện tại có 3 tiêu chuẩn thẻ SD phổ biến là microSD, SDHC và SDXC tương thích với hầu hết thiết bị.

Thẻ SDHC “High Capacity” (dung lượng lớn) được giới thiệu vào năm 2008, đánh dấu bước phát triển mới cho dòng thẻ nhớ SD. Mức dung lượng lưu trữ có thể từ 4 MB đến 64 GB.. Thẻ SDXC “Extended Capacity” (dung lượng mở rộng). Giới hạn lưu trữ được mở rộng từ 64 GB lên 2000 GB (2 TB).

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

**Bảng 2.2 .** Chức năng chân SDCard.

SD Pin	MC Pin	Mini SD Pin	Micro SD Pin	Name	I/O	Logic	Description
1	1	1	2	nCS	I	PP	SPI Card Select [CS]
2	2	2	3	DI	I	PP	SPI Serial Data in [MOSI]
3	3	3		VSS	S	S	Ground
4	4	4	4	VDD	S	S	Power
5	5	5	5	CLK	I	PP	SPI Serial Clock [SCK]
6	6	6	6	VSS	S	S	Ground
7	7	7	7	DO	O	PP	SPI Serial Data Output [MISO]
	8	8	8	NC nIRQ	O	OD	Unused (memory card) Interrupt (SDIO card)
	9	9	1	NC			Unused
		10		NC			Reserved
		11		NC			Reserved

### b. SDIO.

SDIO là chuẩn giao tiếp thẻ nhớ SD tốc độ cao. Khối SDIO của vi điều khiển STM32F4 gồm 2 thành phần chính:

- Khối chuyển đổi SDIO cung cấp đầy đủ chức năng cho các MMC/SD/SD I/O như là xung clock( SDIO\_CK), lệnh ( SDIO\_CMD), dữ liệu(SDIO\_D[0..7]).
- Giao diện APB2 truy cập các thanh ghi SDIO và các tín hiệu yêu cầu ngắt hay DMA.

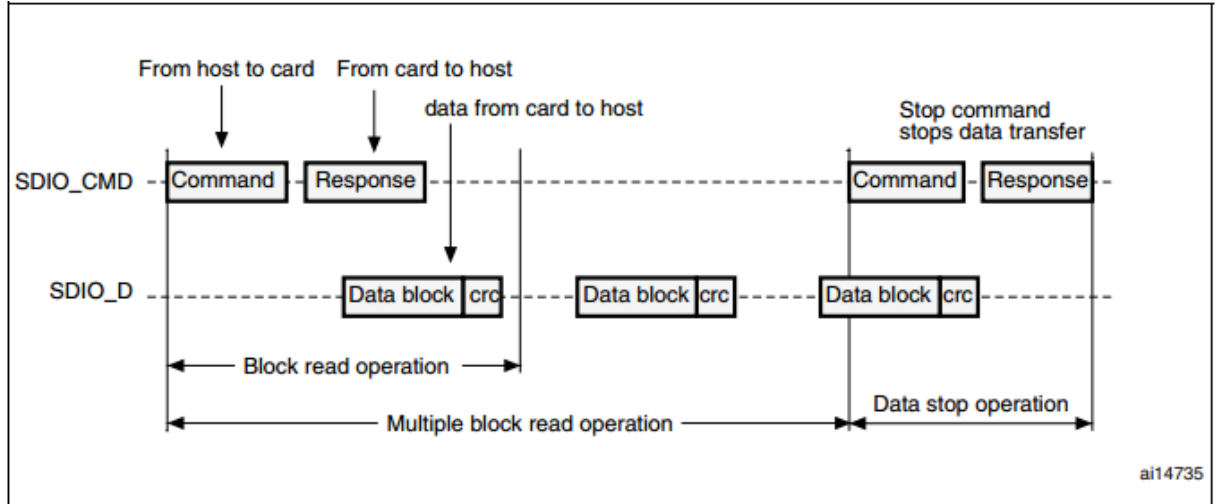
Các đặc điểm của chuẩn giao tiếp SDIO:

- Hỗ trợ 3 chế độ kết nối đường dữ liệu khác nhau theo chuẩn MultiMediaCard System Specification Version 4.2 là: 1 bit, 4 bit và 8 bit.
- Hoàn toàn tương thích với các phiên bản trước của MultiMediaCards.
- Thích hợp với SD Memory Card Specification Version 2.0.
- Hỗ trợ các tính năng đọc CE-ATA.

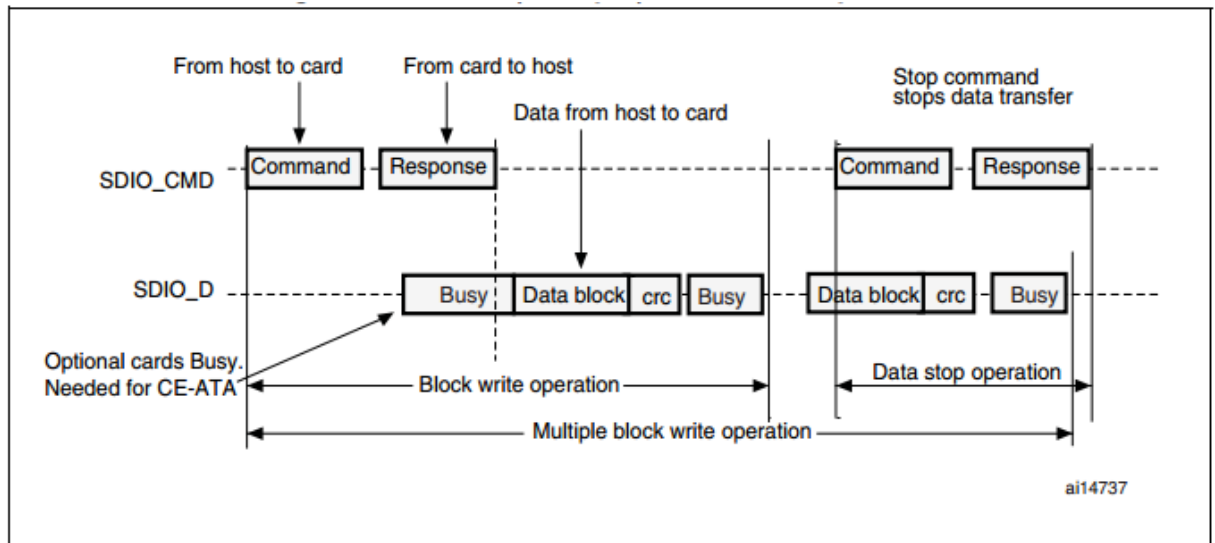
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Tốc độ truyền dữ liệu có thể lên tới 48Mhz với chế độ 8bit.

Chế độ kết nối dữ liệu mặc định của SDIO là mode 1 bit với chân dữ liệu là SDIO\_D0. Chúng ta có thể thay đổi chế độ kết nối trong phần cấu hình SDIO.



**Hình 2.18.** Quá trình đọc dữ liệu của khối SDIO.



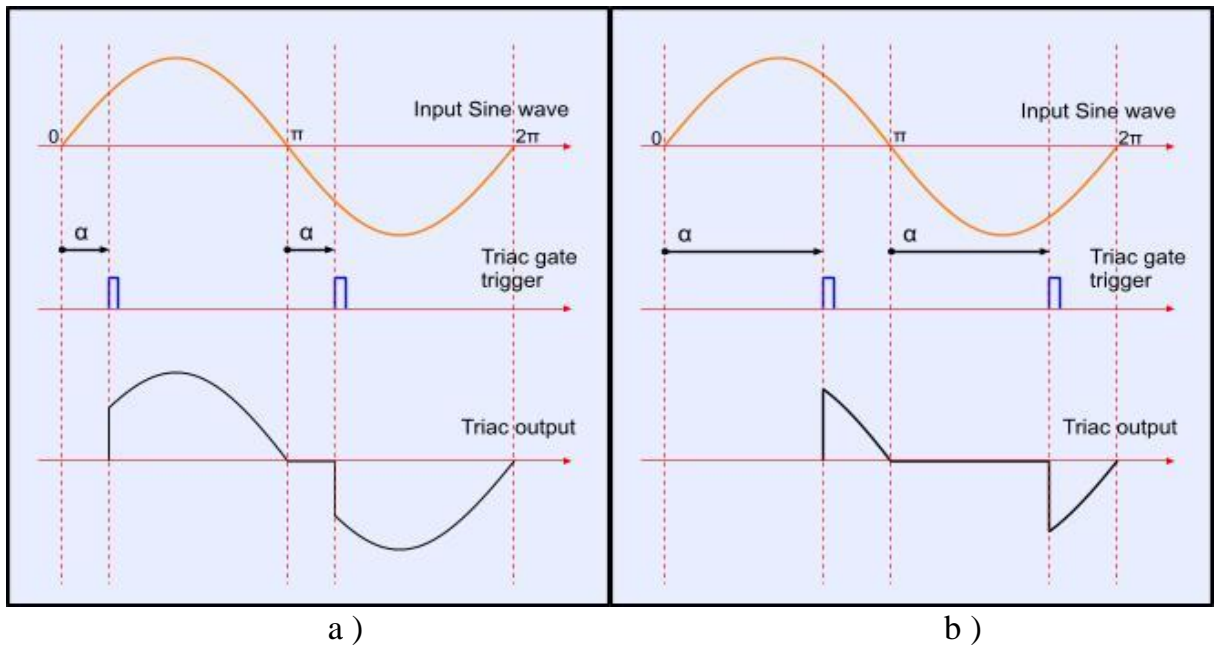
**Hình 2.19.** Quá trình ghi dữ liệu của khối SDIO.

### 2.4. Phương pháp điều chỉnh điện áp xoay chiều một pha.

Phương pháp này có thể giúp ta điều khiển độ sáng của bóng đèn hay tốc độ của quạt bằng cách thay đổi độ trị hiệu dụng của điện áp ngõ ra thông qua việc đóng mở góc kích của triac kết hợp với việc điều chỉnh độ rộng xung của vi điều khiển.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Trong một chu kỳ điện áp xoay chiều 1 pha thì triac chỉ cho phép điện áp đi qua khi được kích dẫn, như hình bên dưới.



**Hình 2.20.** Dạng sóng điện áp trong phương pháp điều chỉnh điện áp xoay chiều.

Hai hình trên là dạng sóng của cùng 1 mạch, khác nhau là với hình a) thì đèn sẽ sáng hơn hình b) do được kích sớm hơn hình b). Thời điểm triac dẫn gọi là góc kích alpha  $\alpha$  và được tính từ điểm 0 của điện áp AC. Góc kích  $\alpha$  càng nhỏ thì triac được dẫn càng sớm do đó điện áp hiệu dụng ngõ ra sẽ càng lớn làm cho đèn sáng hơn và ngược lại. Điểm 0V sẽ xuất hiện 2 lần trong một chu kỳ vì vậy góc kích  $\alpha$  sẽ thay đổi từ  $0^\circ$ - $180^\circ$  ( $0$ - $\pi$ ). Khi  $\alpha=0$  thì toàn bộ điện áp sẽ qua tải và khi  $\alpha=180^\circ$  thì sẽ không có điện áp qua tải.

Công thức tính điện áp hiệu dụng của tải ngõ ra như sau:

$$U_T = U \sqrt{1 - \frac{\alpha}{\pi} + \frac{\sin \alpha}{2\pi}} \quad (2.1)$$

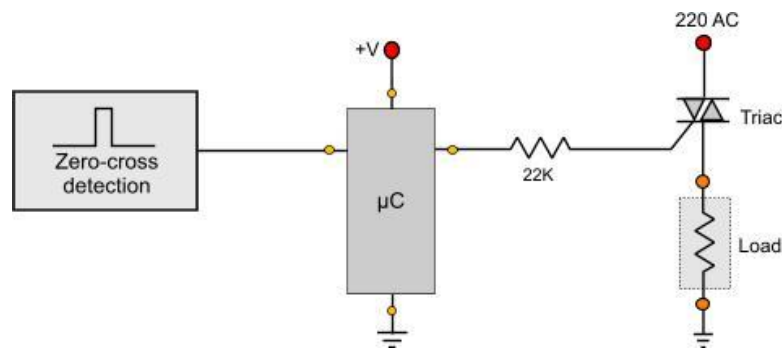
Trong đó:  $U_T$ : Điện áp hiệu dụng ngõ ra.

$U$  : Điện áp hiệu dụng ngõ vào

$\alpha$  : Góc kích.

Mạch phát hiện điểm 0 là một phần khá quan trọng trong việc điều khiển góc kích của triac. Mạch này có nhiệm vụ phát hiện thời điểm sóng điện áp xoay chiều xuống mức 0V, lúc này mạch sẽ tạo ra 1 xung gửi về vi điều khiển, sau đó vi điều khiển sẽ xử lý và tính toán gửi xung tạo góc kích cho triac dẫn.

Giả sử điện áp AC có tần số 50Hz thì có nghĩa là mỗi chu kỳ điện áp mất  $1/50=20$  ms, đồng nghĩa với việc mỗi 20 ms sẽ có 2 lần dạng sóng điện áp về 0V ở thời điểm bắt đầu và thời điểm sau đó 10ms. Khoảng thời gian từ 0-10 ms sẽ quyết định đến điện áp ngõ ra, ví dụ ta muốn giá trị điện áp hiệu dụng ngõ ra bằng một nửa ngõ vào thì vi điều khiển cần gửi 1 xung kích triac vào thời điểm 5ms của chu kỳ dạng sóng điện áp.



Hình 2.21. Mạch điều chỉnh độ sáng cơ bản dùng vi xử lý.

## 2.5. TÌM HIỂU VỀ HỆ ĐIỀU HÀNH ANDROID.

### 2.5.1 Khái niệm Android.

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Ngoài các phiên bản android mới hiện nay: ngày 3/11/2014 Android 5.0 Lollipop đã ra mắt và chính thức được cập nhật cho các thiết bị. Và hiện nay phiên bản mới nhất của android là Android 6.0 Marshmallow được tung ra thị trường vào ngày 5/10/2015.

**Bảng 2.3.** Các phiên bản Android.

<b>Version</b>	<b>Code Name</b>	<b>Release date</b>	<b>API</b>
6.0 - 6.0.1	Marshmallow	October 5, 2015	23
5.0–5.1.1	Lollipop	November 12, 2014	21–22
4.4.x	KitKat	October 31, 2013	19–20
4.1–4.3.1	Jelly Bean	July 9, 2012	16–18
4.0–4.0.4	Ice Cream Sandwich	October 18, 2011	14–15
3.0–3.2.6	Honeycomb	February 22, 2011	11–13
2.3–2.3.7	Gingerbread	December 6, 2010	9–10



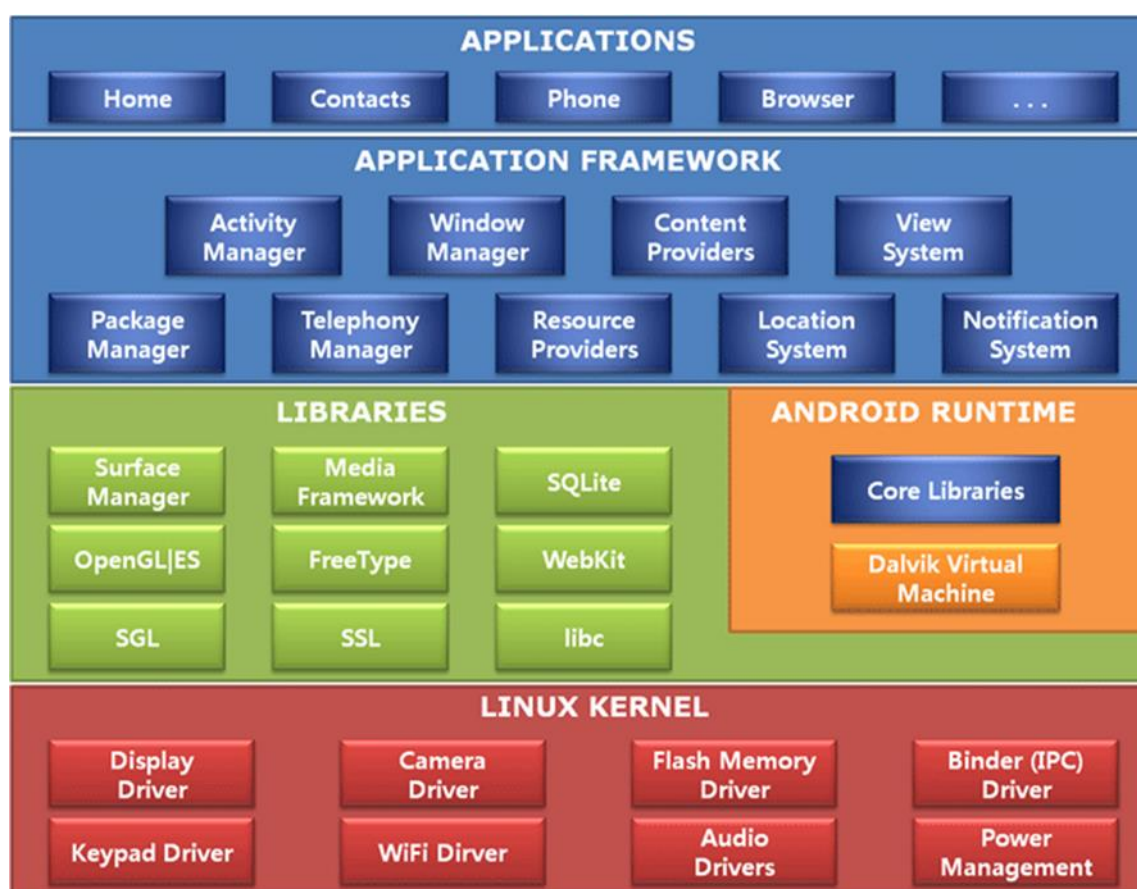
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.2–2.2.3	Froyo	May 20, 2010	8
2.0–2.1	Eclair	October 26, 2009	5–7
1.6	Donut	September 15, 2009	4
1.5	Cupcake	April 27, 2009	3

### 2.5.2 Kiến trúc hệ điều hành Android.

Kiến trúc hệ điều hành chính là các thành phần (component), các lớp (layer) cấu nên một hệ điều hành và các thành phần hay các lớp này phải có mối quan hệ với nhau về cấp bậc và chúng phải được thể hiện rõ trong sơ đồ kiến trúc.

Thì kiến trúc hệ điều hành chia làm 5 lớp chính là lớp ứng dụng (Application), lớp khung ứng dụng (Application Framework), lớp thư viện (Libraries), lớp thực thi Android (Android Runtime), lõi Linux (Linux Kernel).



Hình 2.22. Kiến trúc hệ điều hành Android.

### a . Lớp ứng dụng (Application).

Android đã bao gồm một bộ các ứng dụng chính như trình nhận email, trình nhận tin SMS, lịch, bản đồ, trình duyệt web, danh bạ và các chương trình phụ trợ khác. Tất cả các ứng dụng sử dụng ngôn ngữ lập trình Java.

### b . Lớp khung ứng dụng (Application Framework).

Nhờ Android cung cấp nền tảng phát triển mở, các nhà phát triển có thể xây dựng các ứng dụng đa dạng và sáng tạo. Các nhà phát triển có thể dùng các phần cứng thiết bị, lấy thông tin vị trí, chạy ngầm các dịch vụ, thiết lập cảnh báo, thêm thông báo vào thanh trạng thái và nhiều cái khác nữa.

Các nhà phát triển có thể tác động sâu vào khung API mà các ứng dụng chính đang dùng. Kiến trúc ứng dụng được thiết kế đơn giản để tái sử dụng lại các thành phần trong ứng dụng đó.

Tất cả các ứng dụng nền là một bộ các dịch vụ và hệ thống, bao gồm các thành phần chính:

- View System: đa dạng và có thể tùy biến bao gồm: danh sách (listView), lưới (gridView), textbox, nút (Button), và các View khác.
- Content Provider: cho phép ứng dụng lấy dữ liệu từ ứng dụng khác hay chúng chia sẻ dữ liệu với nhau, ...
- Resource Manager: cho phép truy xuất các nguồn tài nguyên cục bộ như chuỗi tĩnh, hình ảnh và tập tin khai báo giao diện, ...
- Notification Manager: cho phép ứng dụng trình bày cảnh báo ở thanh trạng thái.
- Activity Manager: quản lý vòng đời ứng dụng.

### c . Lớp thư viện (Libraries).

Tầng này chứa tất cả các thư viện Java, được viết đặt biệt cho Android như ác thư viện framework, các thư viện xây dựng giao diện, đồ họa và cơ sở dữ liệu. Một số

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

có thể kể đến như là bộ máy trình duyệt web mã nguồn mở WebKit, thư viện libc, cơ sở dữ liệu SQLite tiện lợi cho việc lưu trữ và chia sẻ dữ liệu, thư viện hỗ trợ thu phát âm thanh và video, thư viện SSL giúp bảo mật mạng...

Thư viện bao gồm các thành phần đáng quan tâm:

- Surface Manager: quản lý giao diện bao gồm lớp giao diện 2D và 3D.
- Media Framework: một bộ thư viện hỗ trợ đọc, ghi lại nhiều định dạng âm thanh, video, hình ảnh như: MPEG4, MP3, AAC, JPG, PNG..v..v...
- SQLite: hệ thống cơ sở dữ liệu có kích thước nhỏ nhưng mạnh mẽ cho tất cả ứng dụng.
- FreeType: liên quan trình bày kiểu chữ và hình ảnh.
- SGL: hệ thống đồ họa 2D cơ sở.
- Libc: thư viện hệ thống C được rút ra từ thư viện hệ thống C chuẩn.

### **d . Lớp thực thi Android(Android Runtime).**

Android sử dụng máy ảo Dalvik để chạy các ứng dụng viết bằng ngôn ngữ Java. Máy ảo Dalvik là là máy ảo được tinh chỉnh dành riêng cho thiết bị di động chạy trên nền Android. Khác với máy ảo Java chạy trên các thiết bị di động có hỗ trợ Java đời cũ, máy ảo Dalvik có thể giúp thiết bị chạy nhiều máy ảo cùng lúc (thể hiện tính Đa Nhiệm).

Máy ảo Dalvik chạy nhiều tập tin định dạng .dex được tối ưu hóa cho máy có bộ nhớ thấp. Máy ảo Dalvik lệ thuộc vào lõi Linux cho các tính năng cơ bản như là luồng và quản lý bộ nhớ mức độ thấp.

### **e . Lõi Linux (Linux Kernel).**

Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, ... Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

### 2.5.3 Sơ lược về lập trình ứng dụng cho android.

#### a . Ngôn ngữ lập trình.

Ngôn ngữ lập trình chính thức của Android là Java. Mặc dù các ứng dụng trên Android được phát triển dựa trên nền tảng Java, nhưng Android không hỗ trợ J2ME và J2SE, là hai ngôn ngữ lập trình phổ dụng cho các thiết bị di động dựa trên máy ảo Java của Sun.

#### b . Môi trường lập trình cho Android.

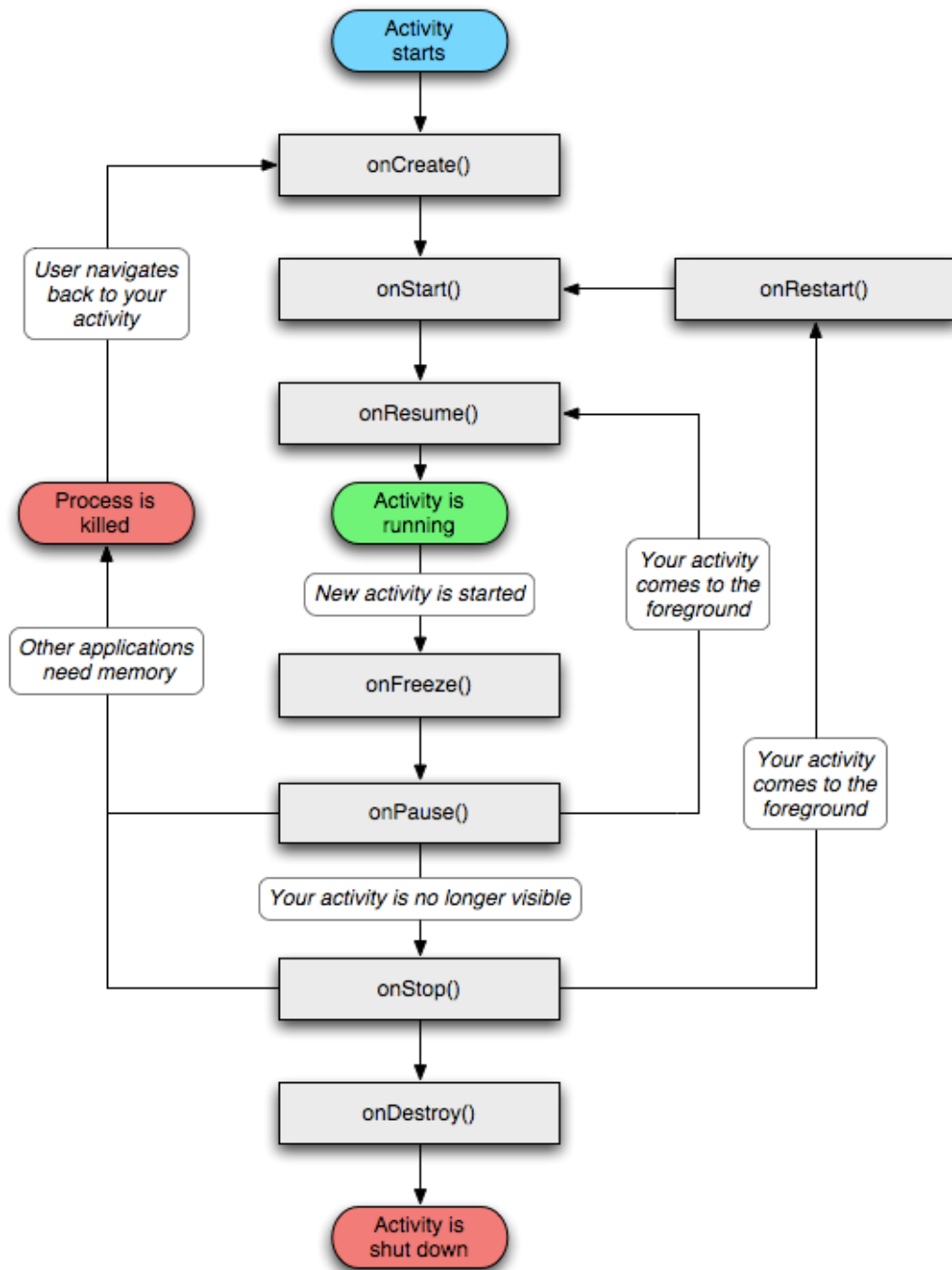
Trước đây, lập trình với android sử dụng Eclipse và cài đặt plugin vào để code. Nhưng hiện nay Google đã chính thức release bản Android Studio riêng (giống như Visual Studio của Microsoft đó), tất cả những gì cần lập trình cho android cũng như các plug-in đều sẽ được tích hợp và có thể cài mở rộng.

Có thể nói Android Studio là một môi trường phát triển tích hợp (IDE - Integrated Development Environment) tuyệt vời, được dựa trên một tên tuổi nổi tiếng IntelliJ IDE. Như tên gọi, Android Studio là một môi trường thiết kế và phát triển ứng dụng cho nền tảng Android. Môi trường phát triển này rất dễ cài đặt, thiết lập và có thể tạo ra một dự án (project) mới chỉ sau vài giây.

### 2.5.4 Các thành phần cơ bản của một ứng dụng android.

#### a . Activity.

Là thành phần quan trọng nhất của bất kỳ một ứng dụng Android nào. Thuật ngữ Activity chỉ một việc mà người dùng có thể thực hiện trong một ứng dụng Android. Do gần như mọi activity đều tương tác với người dùng, lớp Activity đảm nhận việc tạo ra một cửa sổ (window) để người lập trình đặt lên đó một giao diện người dùng với `setContentView(View)`. Một activity có thể mang nhiều dạng khác nhau: Một cửa sổ toàn màn hình (full screen window), một cửa sổ floating (với `windowsIsFloating`) hay nằm lồng bên trong 1 activity khác (với `ActivityGroup`).



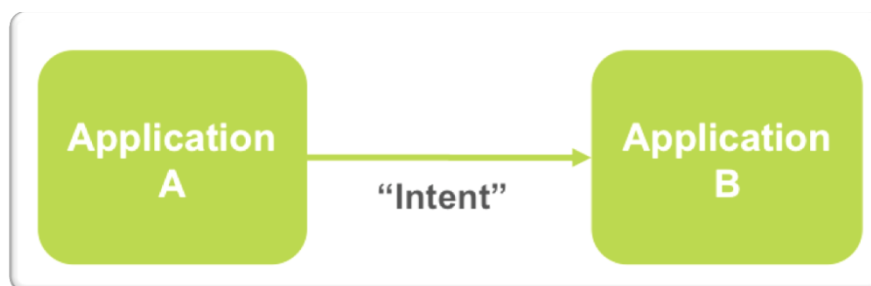
[<http://expressmagazine.net/>]

Hình 2.23. Vòng đời của Activity.

### b . Intent.

Intent được sử dụng với phương thức `startActivity()` để mở một Activity, và dùng với `broadcastIntent` để gửi nó đến bất kì `BroadcastReceiver` liên quan nào, và dùng với `startService(Intent)`, `bindService(Intent, ServiceConnection, int)` để giao tiếp với các Service chạy dưới nền.

Intent cung cấp một chức năng cho phép kết nối hai chương trình khác nhau trong quá trình thực thi (runtime) (Cung cấp khả năng cho phép hai chương trình khác nhau giao tiếp với nhau). Chức năng quan trọng và được sử dụng nhiều nhất của một Intent là mở một Activity, nơi mà nó có thể được dùng như một vật kết nối các Activity lại với nhau (Truyền thông tin giữa hai Activity khác nhau).[4]



[<http://expressmagazine.net/>]

**Hình 2.24.** Sử dụng Intent để trao đổi thông tin giữa hai chương trình.

### c . Service.

Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó. Mỗi class Service phải chứa thẻ `<service>` được khai báo trong file `AndroidManifest.xml`. Services có thể được bắt đầu bởi `Context.startService()` và `Context.bindService()`

Cũng như các đối tượng của ứng dụng khác, services chạy trên luồng (Thread) của tiến trình chính. Có nghĩa là nếu service dự định chạy các hành vi có cường độ lớn hoặc các hành vi có thể gây nghẽn mạch, nó sẽ tự sinh ra luồng (Thread) của chính nó để làm thực hiện các tác vụ được mô tả trong Service.

### d . View.

View là thành phần cơ bản để xây dựng giao diện người dùng cho 1 ứng dụng Android. View là 1 lớp căn bản của widgets (widgets được dùng để tạo các nút nhấn, text fields,...).

### 2.5.5 Chức năng giám sát bằng camera ip thông qua ứng dụng android.

Ngày nay yêu cầu về an ninh và bảo mật ngành càng cấp thiết và quan trọng. Ngoài lực lượng an ninh thì camera IP là một công cụ hỗ trợ đắc lực trong việc giám sát từ xa.

Camera IP sử dụng giao thức TCP/IP để truyền tải dữ liệu hình ảnh, âm thanh, mỗi camera IP có 1 địa chỉ IP riêng biệt như 1 máy tính nối mạng.

Để có thể giám sát từ xa bằng camera IP thì cần phải đăng ký một tên miền ddns miễn phí. Ứng dụng android sẽ truy cập vào tên miền của camera IP và cho phép hiển thị hình ảnh, âm thanh camera IP thu được trên màn hình của điện thoại.

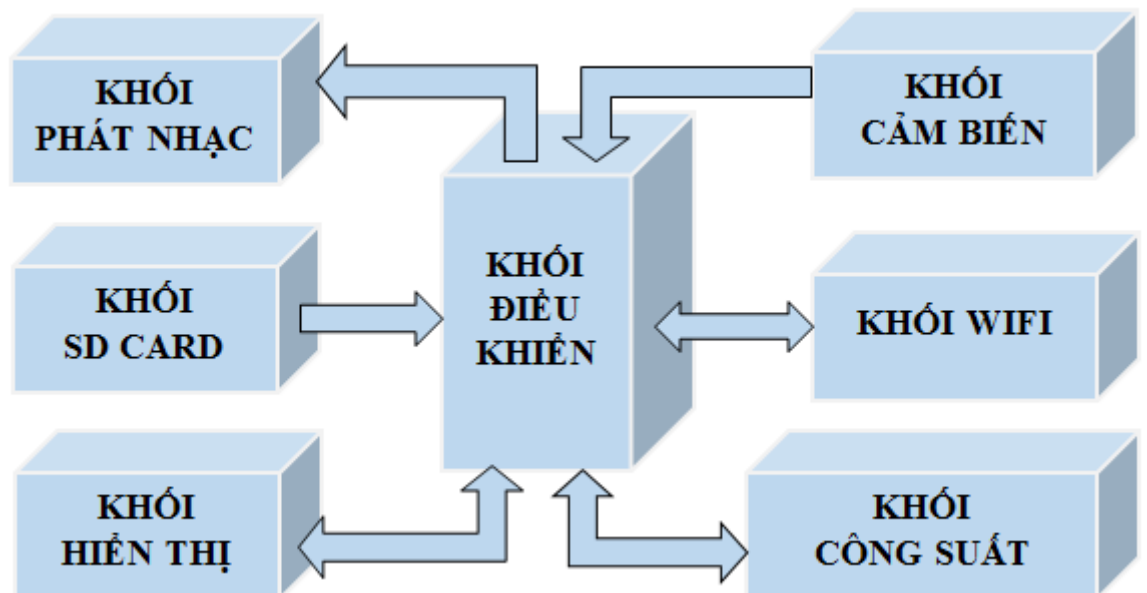
## **Chương 3. TÍNH TOÁN VÀ THIẾT KẾ**

### **3.1. GIỚI THIỆU**

Với mục đích tạo ra hộp điều khiển có thể điều khiển các thiết bị trong nhà công suất nhỏ nên hệ thống cần phải có khối công suất, khối điều khiển, nhóm lựa chọn phương pháp điều khiển thông qua điện thoại và màn hình cảm ứng nên phải có khối hiển thị và khối wifi, ngoài ra nhóm còn có các khối khác như khối SD Card, khối cảm biến và khối phát nhạc phục vụ nhu cầu giải trí.

### **3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG.**

#### **3.2.1. Sơ đồ khối hệ thống**



**Hình 3.1.** Sơ đồ khối hệ thống.

Hệ thống gồm 7 khối:

- Khối điều khiển STM32F407VET6 có chức năng nhận và xử lý dữ liệu từ cảm biến, SD card, module wifi, điều khiển màn hình LCD ở khối hiển thị, khối phát nhạc MP3 và điều khiển các ngõ ra ở mạch công suất.
- Khối đọc SD CARD: đọc, giải mã file hình ảnh, âm thanh mp3.
- Khối hiển thị người dùng GLCD 7" hiển thị nhiệt độ, thời gian, trạng thái thiết bị, giao tiếp với người dùng.



- Khối giải mã và phát nhạc Mp3 VS1003B.
- Khối công suất điều khiển các thiết bị điện 220V, đồng thời cung cấp nguồn cho GLCD và khối điều khiển.
- Khối cảm biến nhiệt độ, độ ẩm.
- Khối giao tiếp wifi với điện thoại Android dùng module ESP8266.

### 3.2.2. Thiết kế khối mạch công suất.

Với mục đích cung cấp nguồn cho khối vi điều khiển và màn hình LCD ngoài ra còn có 5 ngõ ra để điều khiển các thiết bị 220VAC và 1 khối ngõ ra 0-16VDC. Vì vậy mạch công suất bao gồm các khối:

- Khối ngõ vào 12VAC cấp từ adapter.
- Khối chỉnh lưu và ổn áp 5VDC/3A.
- Khối phát hiện điểm không.
- Khối các ngõ ra điều khiển thiết bị 220VAC.
- Khối ngõ ra 0 -> 16VDC.

#### a. Khối ngõ vào 12VAC.

Để đáp ứng về yêu cầu thẩm mỹ và nhỏ gọn nên nhóm quyết định chọn khối nguồn ngõ vào từ Adapter 12VAC mà không làm mạch nguồn xung 220VAC nhằm giảm diện tích cho mạch công suất. Các thông số của Adapter 12VAC:

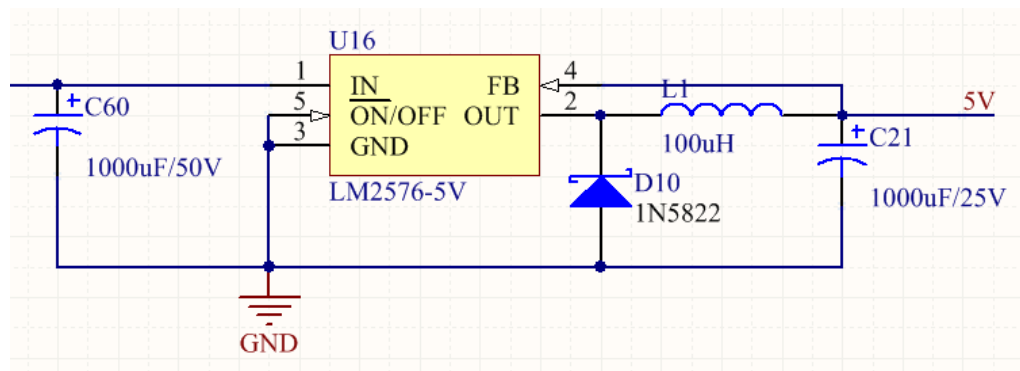
- Điện áp ngõ vào: 220VAC
- Tần số: 50Hz
- Điện áp ngõ ra: 12VAC
- Dòng tối đa: 1A

### b . Khối chỉnh lưu điện áp và nguồn ổn áp 5V/3A.

#### ❖ Khối chỉnh lưu.

Do nguồn ngõ vào là 12VAC nên cần phải chỉnh lưu điện áp AC thành DC để cấp cho khối vi điều khiển và LCD. Hiện nay trên thị trường có nhiều loại IC chỉnh lưu khác nhau, sau khi xem xét nhóm quyết định chọn IC chỉnh lưu cầu diode KPB307 do giá thành rẻ và nhiều người sử dụng.

#### ❖ Khối nguồn ổn áp 5VDC/3A.



Hình 3.2. Mạch nguồn 5V sử dụng LM2576.

Với yêu cầu ngõ ra là  $V_{out} = 5VDC$  và  $I_{load} = 3A$ . Mạch sử dụng IC LM2576-5V với ngõ vào lấy từ ngõ ra của IC chỉnh lưu.

LM2576 là loại IC nguồn switching rất thông dụng trên thị trường và có khả năng hoạt động được với điện áp vào lên tới 45V đối với các loại thông thường hoặc tới 63V đối với loại LM7526HV.

Họ LM2756 có ưu điểm là cho dòng ra lên tới 3A dòng tải, lớn hơn nhiều so với các IC ổn áp tuyến tính thông thường như KA7805,KA7812,...LM2576 là lựa chọn thích hợp với các mạch ứng dụng cần dòng lớn như Led driver, Motor DC....

Các thông số của LM2576:

- Điện áp làm việc: 4→45V.
- Điện áp ngõ ra: 5V
- Hiệu suất: 77%
- Tần số dao động: 52Khz

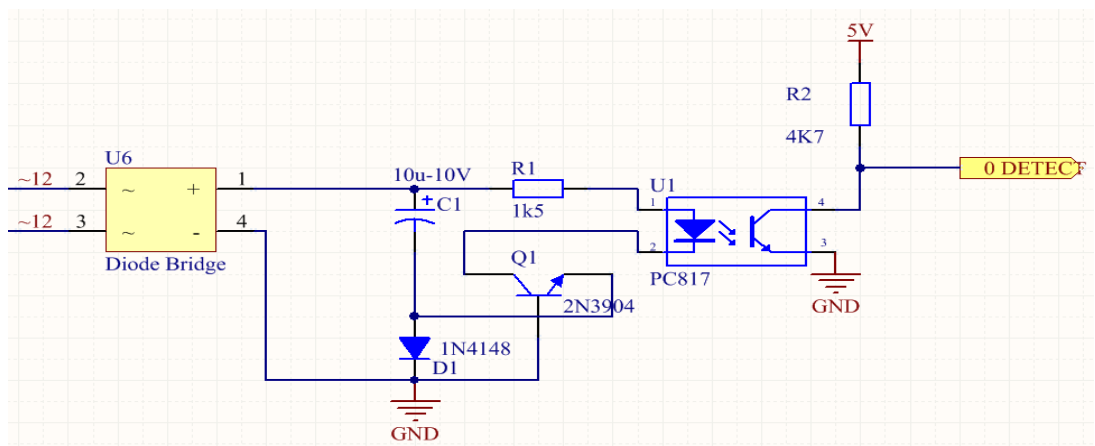
## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

- Dòng giới hạn: 5.8A
- Nhiệt độ làm việc:  $-40 \rightarrow 125^{\circ}\text{C}$

### Tính toán các thông số.

- Tính  $L_1$ : dựa vào đặc tính điện áp dòng điện của LM2576, chọn giá trị  $L_1=100\mu\text{H}$ .
- Tính  $C_{\text{OUT}}$ : giá trị tụ lọc ngõ ra có thể lấy từ  $680\mu\text{F}$  đến  $2000\mu\text{F}$ . Điện áp gấp 1.5 lần điện áp ra  $5\text{V} \rightarrow$  chọn  $C_{\text{OUT}} = 1000\mu\text{F}/50\text{V}$ .
- Chọn diode kẹp: dựa vào bảng hướng dẫn với điện áp đầu ra  $5\text{V}-3\text{A}$  có thể chọn diode schottky 1N5820.
- Tụ ngõ vào  $C_{\text{IN}}$  chọn  $100\mu\text{F}-35\text{V}$ .

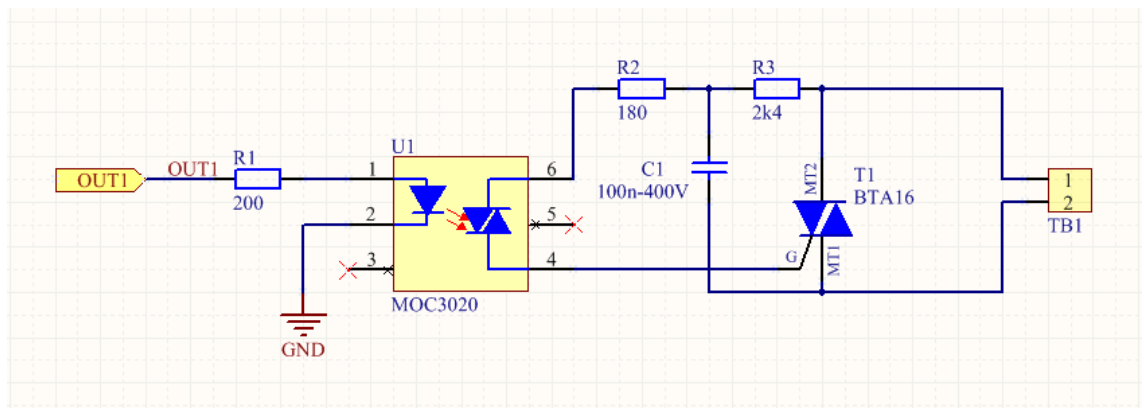
### c . Mạch phát hiện điểm 0.



Hình 3.3. Mạch phát hiện điểm 0.

Mạch tạo một xung ngắn khi điện áp vào bằng 0. Khi điện áp chính lưu vào giảm xuống thấp hơn C1, Q1 bắt đầu dẫn và tạo ra một xung vài trăm micro giây, IC PC817 sẽ làm việc theo xung Q1. Điện trở R2 giới hạn dòng qua LED phát của PC817.

### d . Mạch chuyển đổi công suất.



**Hình 3.4.** Mạch chuyển đổi công suất.

BTA12 là triac điều khiển toàn sóng với hiệu suất cao, chống nhiễu tốt và chuyển mạch nhanh.

Đặc tính:

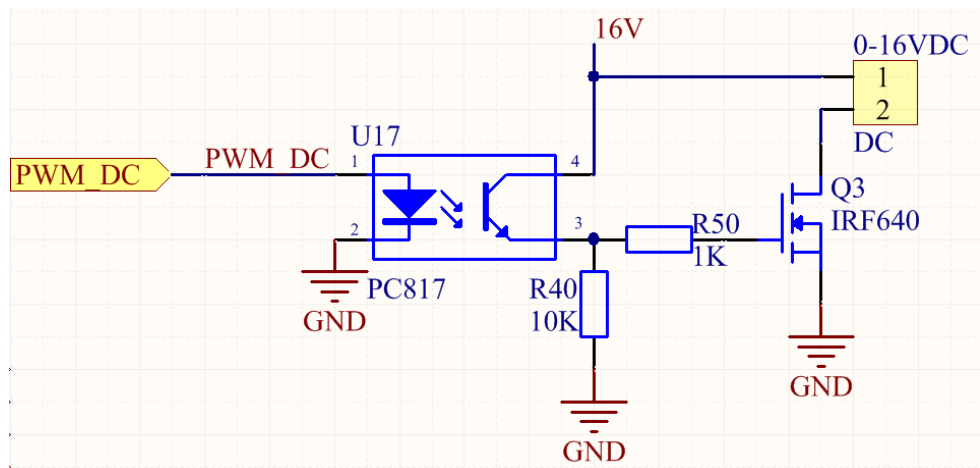
- Điện áp làm việc tới 800V.
- Dòng ở trạng thái dẫn là 12A RMS ở 25°C.
- Chuyển mạch nhanh 1.5mA/ms.
- Cách ly nội (2500V<sub>RMS</sub>)

MOC3020 là opto cách ly quang dùng để điều khiển triac. MOC3020 bao gồm một LED phát hồng ngoại GaAs và một chuyển mạch hai chiều bằng ánh sáng (hoạt động như một triac). MOC3020 dùng để giao tiếp giữa các điều khiển điện tử và triac công suất để điều khiển các tải trở và tải cảm. Điện áp cách ly cao (5300VAC RMS) và chuyển mạch nhanh.

Khi ngõ ra OUT1 của vi điều khiển lên mức 1, LED phát của MOC sáng, triac quang dẫn kích chân G của triac công suất, điện trở R2 giới hạn dòng kích, lúc này triac công suất dẫn. Tụ C1 được sử dụng khi tải có tính chất cảm.

## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

### e . Mạch PWM 0-16VDC.



Hình 3.5. Mạch PWM 0-16VDC.

IRF640 là MOSFET kênh N, chuyển mạch với tần số rất cao, dùng trong các mạch điều chỉnh điện áp DC, điều chỉnh tốc độ động cơ DC.

Bảng 3.1 . Thông số MOSFET IRF640

Thông số	Giải thích	Min	Max	Đơn vị	Điều kiện
$V_{DSS}$	Điện áp đánh thủng	200		V	$V_{GS}=0V, I_D=250\mu A$
$R_{DS}$	Điện trở cực DS		0.18	$\Omega$	$V_{GS}=10V, I_D=11A$
$V_{SD}$	Điện áp trước diode		2	V	$T_J=25^\circ C, I_S=18A, V_{GS}=0V$
$V_{GS}$	Điện áp trước cực GS	2	4	V	$V_{DS}=V_{GS}, I_D=250\mu A$
$I_D$	Dòng tải tối đa		18	A	$V_{GS}=10V, T_C=25^\circ C$
P	Công suất cực đại		50	W	

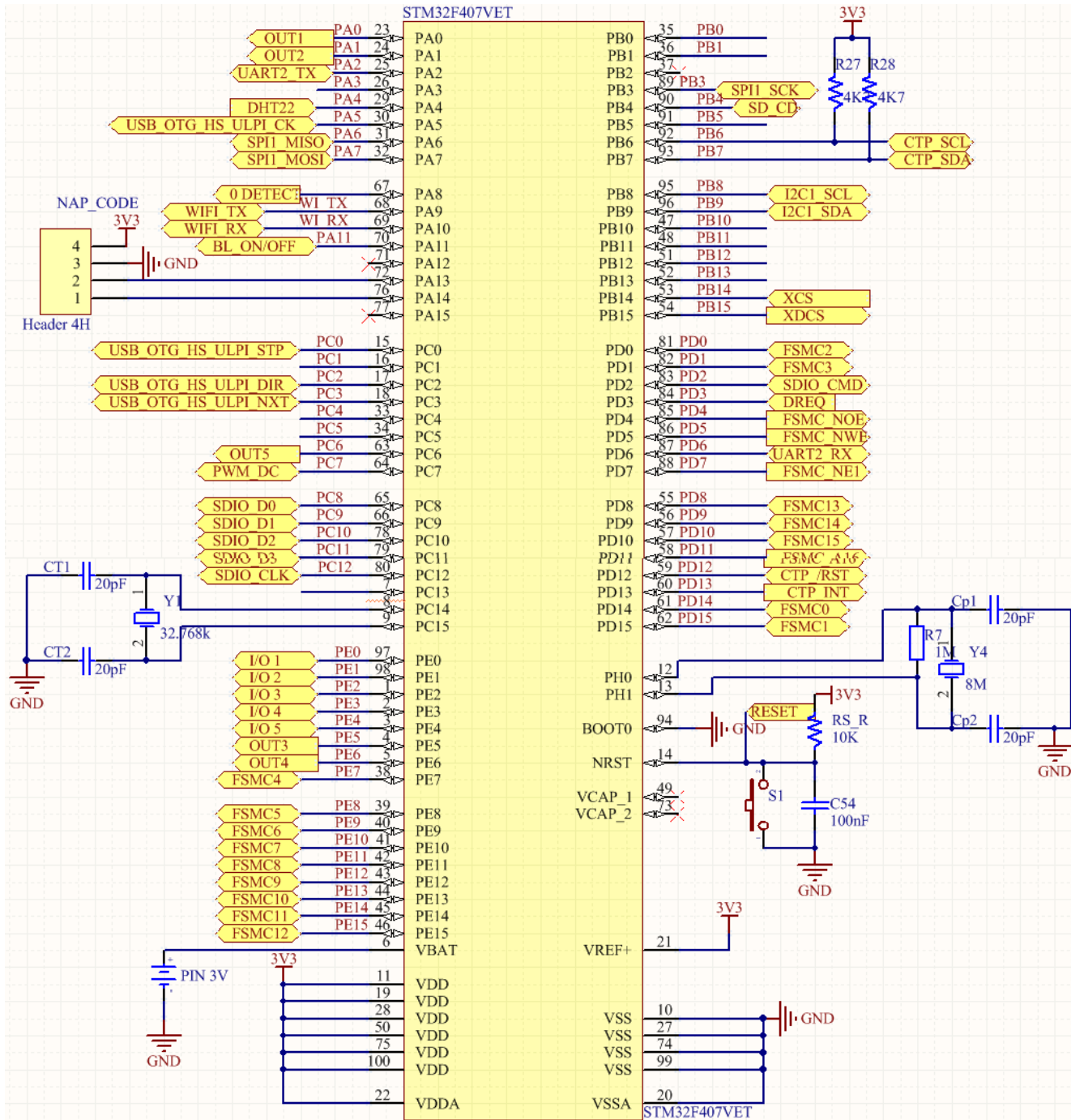
Xung điện áp đưa vào chân G của IRF, giả sử ban đầu xung điện áp  $> 0V$ , IRF dẫn, có điện áp đầu ra, tại thời điểm thứ hai xung điện áp  $= 0$ , IRF khóa, điện áp đầu ra  $= 0$  và cứ như vậy tạo ra điện áp đầu ra có dạng xung vuông, tần số phụ thuộc vào

tần số ngõ ra PWM của vi xử lý, ở đây là 10Khz. Opto PC817 dùng như công tắc chuyển mạch đóng ngắt IRF theo tần số ngõ ra của vi xử lý, đồng thời cách ly mạch IRF với mạch điều khiển.

### **3.2.3. Khối Điều Khiển**

Với việc điều khiển thiết bị điện 220V dùng PWM của timer 1, phát nhạc dùng chip VS1003B với SD Card để giải mã mp3 theo chuẩn SDIO, điều khiển GLCD 7” dùng chuẩn FSMC và giao tiếp với cảm biến nhiệt độ độ ẩm, cảm biến ánh sáng và module Wifi. Với những yêu cầu trên nhóm quyết định chọn chip STM32F407VET6. Vừa đáp ứng đủ các ngõ IO để điều khiển vừa tiết kiệm chi phí nhưng vẫn đáp ứng đủ hiệu năng. Chip xử lý 32 bit ARM Cortex M4 STM32F407VET6. Với bộ nhớ Flash lên đến 512Kb, bộ nhớ RAM 192 Kbytes và 80 ngõ IO.

# CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ



Hình 3.6. Sơ đồ kết nối vi điều khiển.

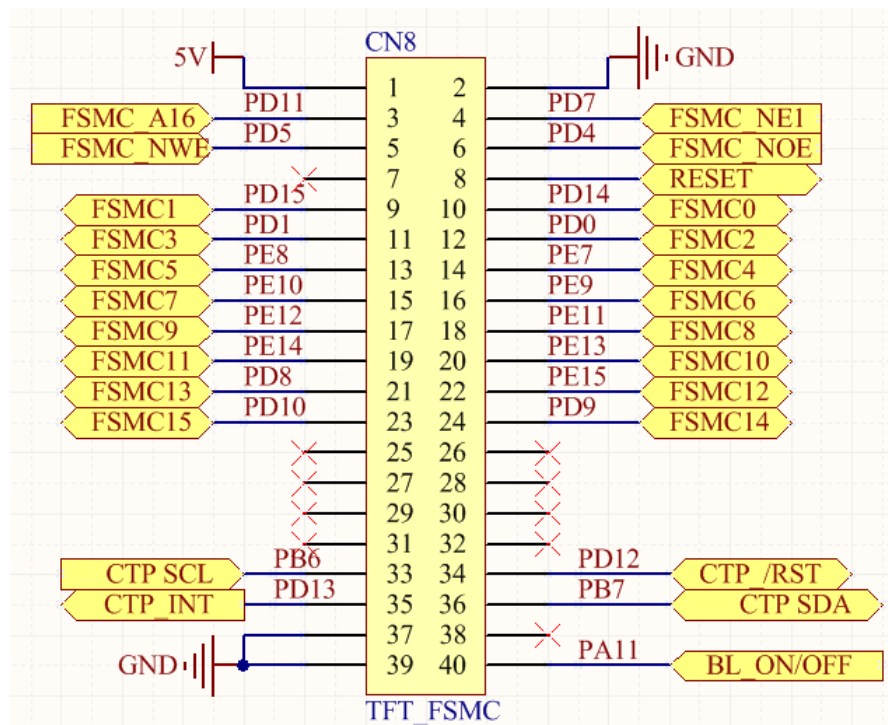
### 3.2.4. Khối hiển thị giao tiếp với người dùng

Sử dụng màn hình màu cảm ứng 7" để hiển thị và giao tiếp với người dùng chip điều khiển được tích hợp sẵn trên LCD là chip SSD1963. Màn hình 7 inch độ phân giải 800x480 pixel:

- Hiển thị giao diện nhiệt độ, thời gian, trạng thái các thiết bị và giao diện nghe nhạc
- Giao tiếp với người dùng , thao tác lệnh ngay trên màn hình .

Sơ đồ kết nối cảm ứng được trình bày ở hình 3.7, các tín hiệu giao tiếp với vi điều khiển gồm có : TP\_/RST dùng chân PD12, TP\_INT dùng chân PD13, TP\_SCL dùng chân PB6, TP\_SDA dùng chân PB7.

CON40 là connector vừa giao tiếp cảm ứng vừa giao tiếp hiển thị Graphic LCD, vi điều khiển sử dụng chuẩn giao tiếp FSMC với 16 bit để giao tiếp, các bit điều khiển LCD\_CS, LCD\_RS, LCD\_WR, LCD\_RD được kết nối với các chân PD7, PD11, PD5, PD4 .

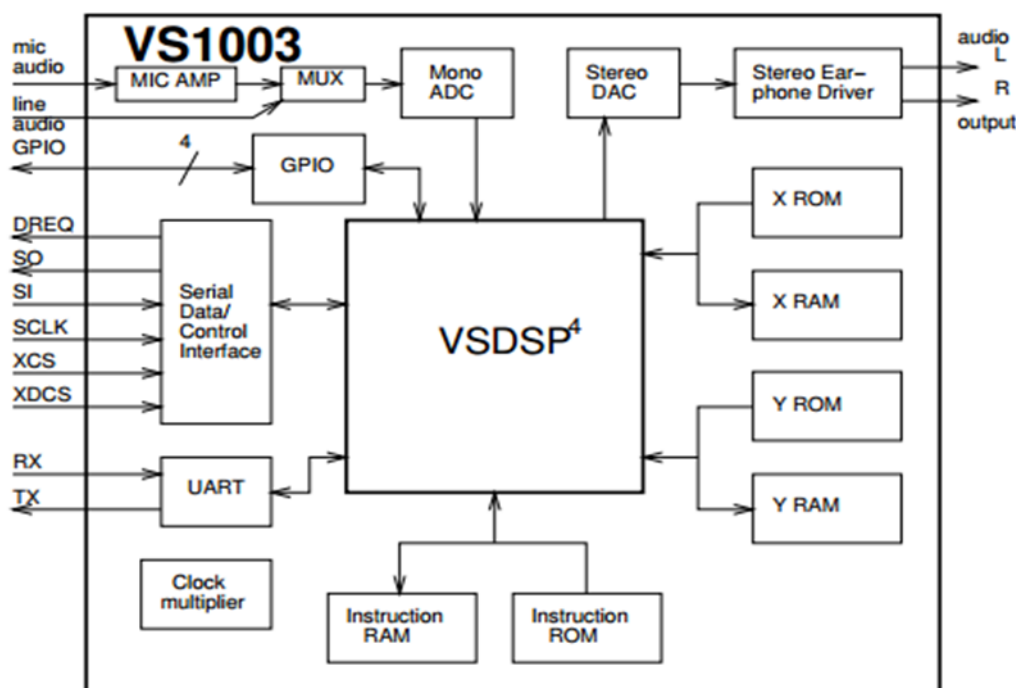


Hình 3.7 . Sơ đồ kết nối chân GLCD và Touch với vi điều khiển



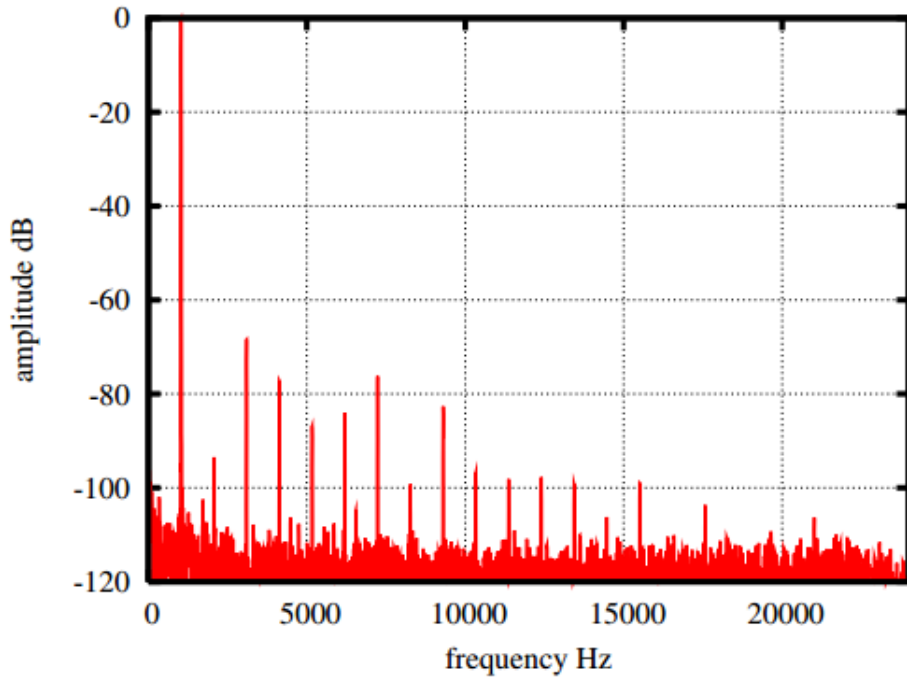
### 3.2.5. Khối giao tiếp giải mã âm thanh.

Khối này sử dụng IC giải mã âm thanh VS1003b dùng để giải mã âm thanh cho các chuẩn định dạng nhạc MP3, WMA, WAV, MIDI, thu nhạc định dạng IMA, ADPCM (âm thanh mono). Hỗ trợ bộ chuyển đổi DAC cho MP3 và WAV dạng stereo chất lượng cao. Có bộ điều khiển tai nghe âm thanh, giao tiếp MP3 dùng chuẩn SPI1 .VS1003B có 5.5Kb RAM trên chip sử dụng cho code/data.



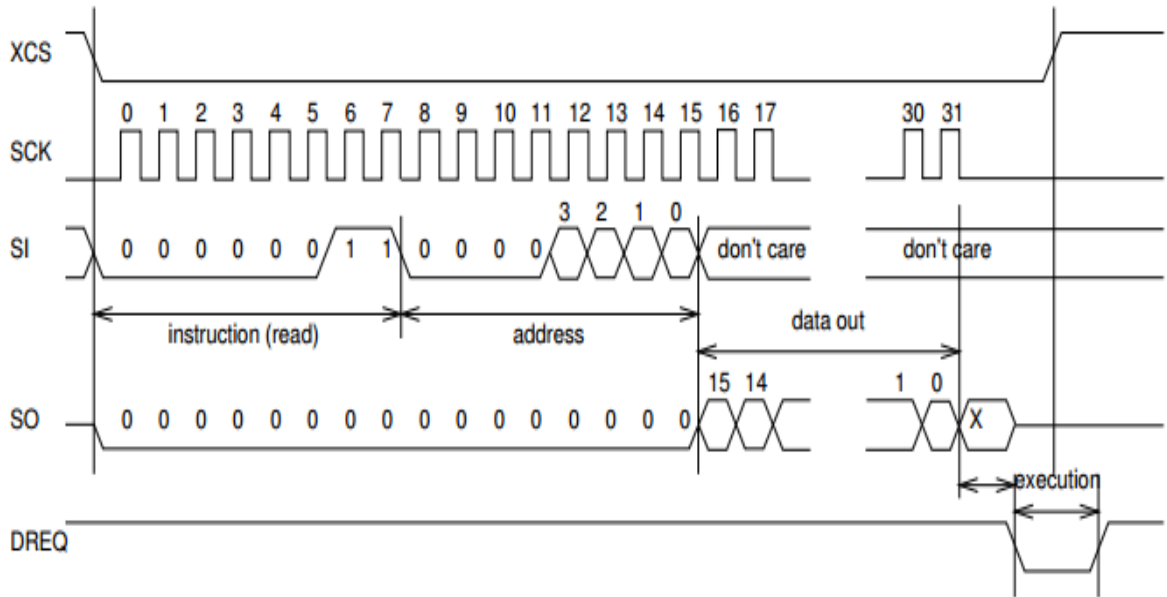
Hình 3.8. Sơ đồ khối VS1003B

Ngõ ra ra tại chân LEFT hoặc RIGHT sẽ tạo ra sóng sine ngõ ra tần số 1 kHz với bộ điều chế tần số 48 kHz (với thạch anh cung cấp xung ngoài 12.228 MHz).



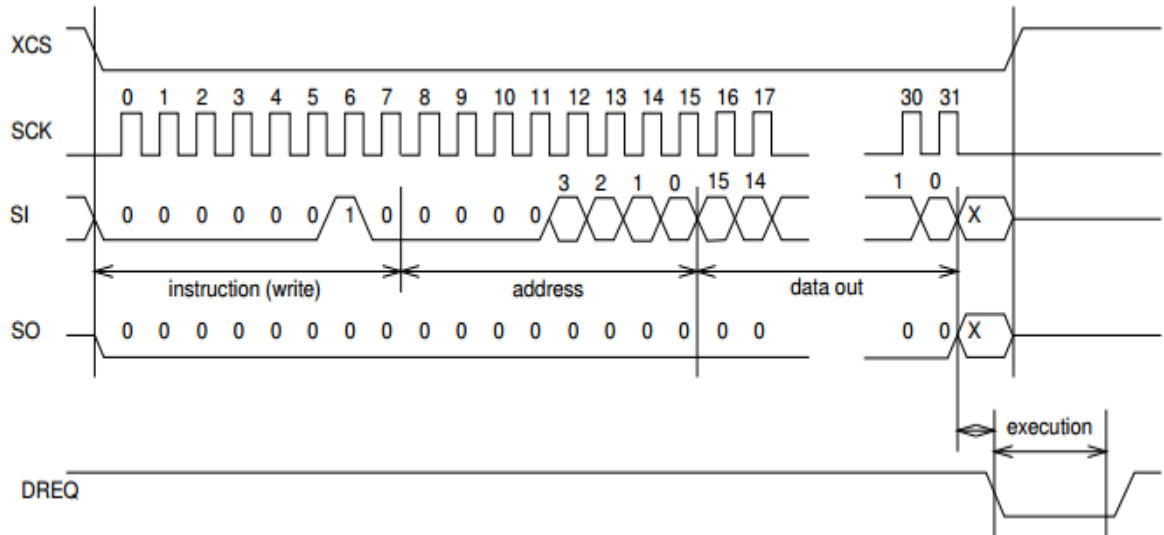
**Hình 3.9.** Phổ ngõ ra tại chân LEFT hoặc RIGHT với trở ngõ ra là 30 Ω

Chế độ SCI Read của VS1003B( mã lệnh 0x03):



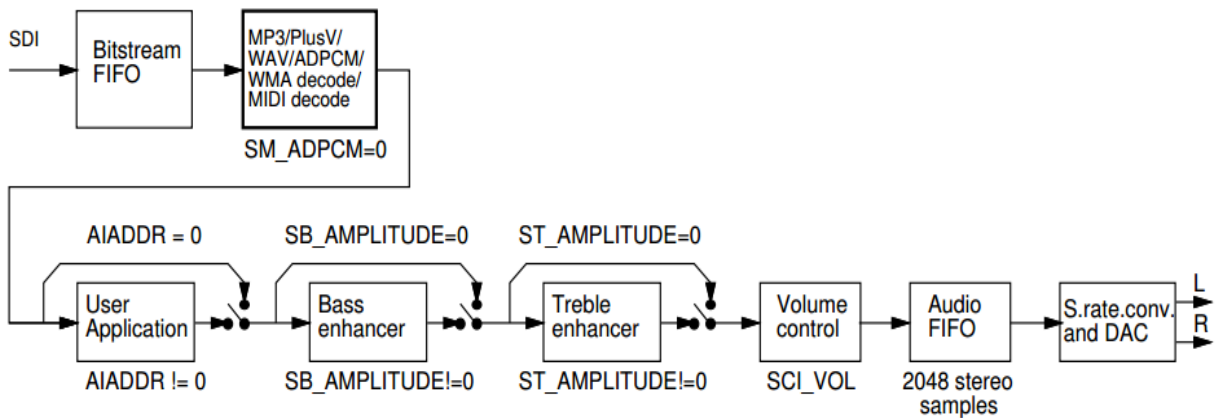
**Hình 3.10.** Giải đồ xung của quá trình đọc VS1003B

Chế độ SCI Write của VS1003B( mã lệnh 0x02):



**Hình 3.11.** Giải đồ xung của quá trình ghi VS1003B.

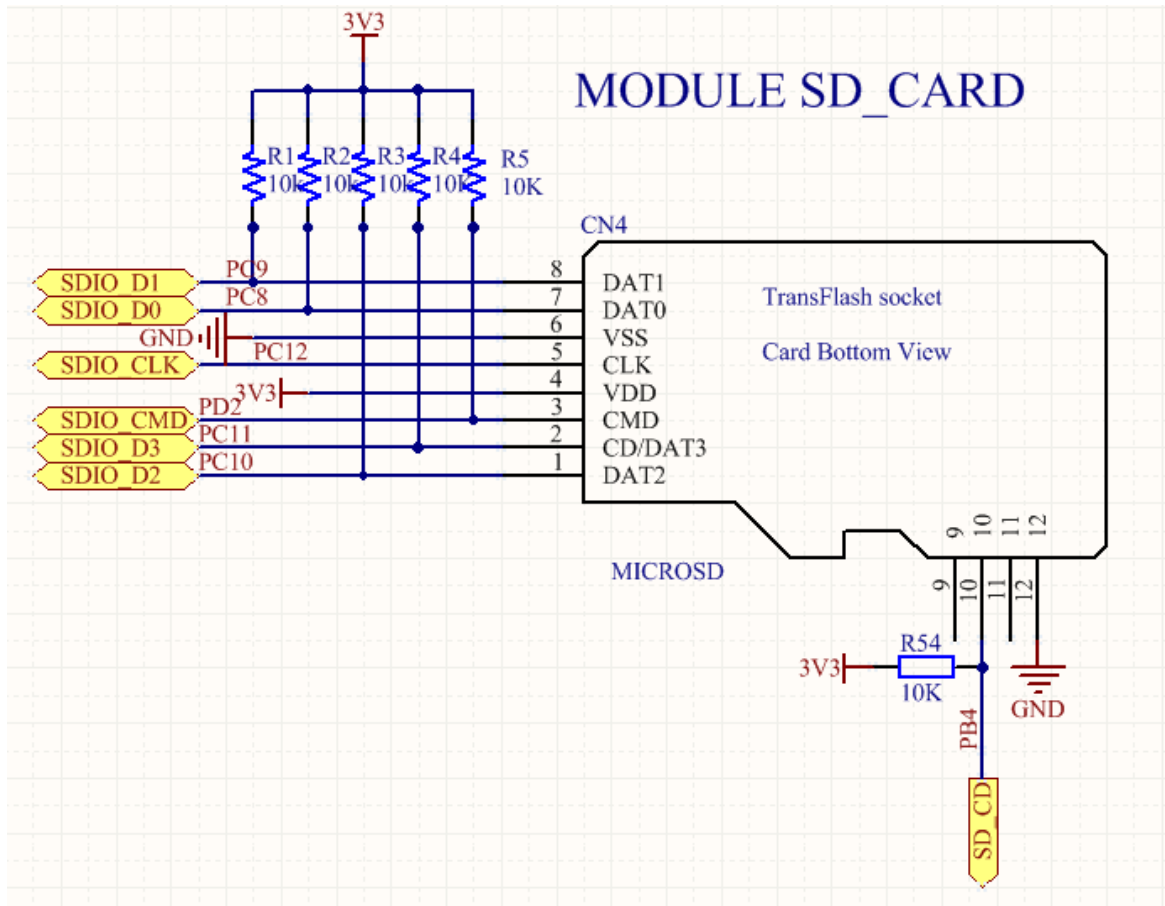
Sơ đồ giải mã Mp3 của VS1003B:



**Hình 3.12.** Sơ đồ thứ tự giải mã dữ liệu của VS1003B.

### 3.2.6. Khối giao tiếp SD Card.

Khối giao tiếp với SD card giao tiếp qua chuẩn SDIO 4 bit. Các chân tín hiệu lần SDIO\_D1 đến SDIO\_D3 được kết nối tương ứng với các chân PB8, PB9, PB10 và PB11, chân SDIO\_CLK nối với chân PC12, chân SDIO\_CMD nối với chân PD2 và chân SD\_CD nối với chân PB4.



Hình 3.13. Sơ đồ khối SD Card.

### 3.2.7. Module wifi ESP8266 V12.

#### a. Giới thiệu

Module ESP8266 là module wifi giá rẻ và được sử dụng rộng rãi trong các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác.

ESP8266 là một chip tích hợp cao, được thiết kế cho nhu cầu của một thế giới kết nối mới, thế giới Internet of thing (IOT). ESP8266 cung cấp một giải pháp kết nối

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

mạng Wi-Fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

ESP8266 có xử lý và khả năng lưu trữ mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với một chi phí tối thiểu và một PCB tối thiểu.



Hình 3.14. Module ESP8266 V12.

#### b. Các thông số kỹ thuật.

ESP8266 V12 có các thông số kỹ thuật như sau:

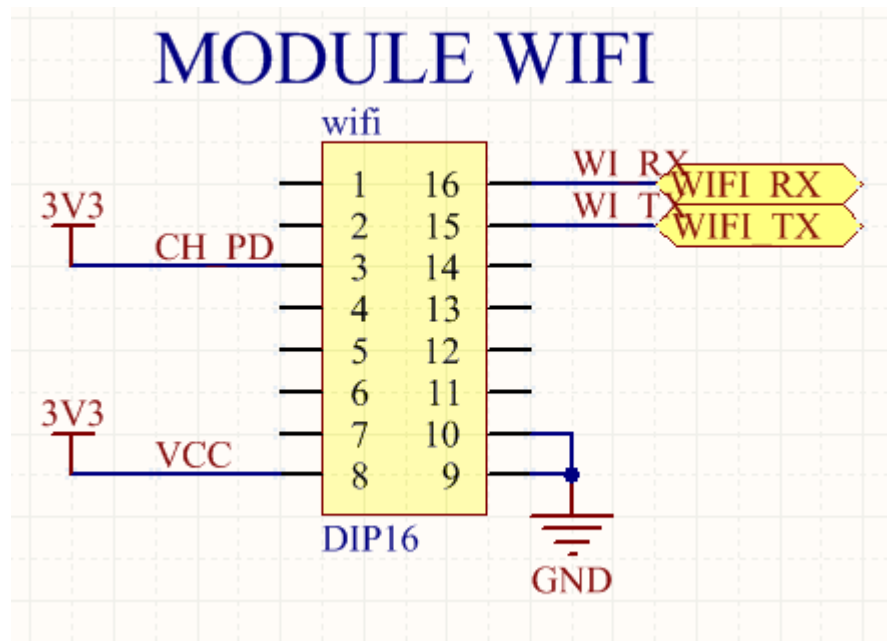
- Tương thích các chuẩn wifi: 802.11 b/g/n
- Wi-Fi 2.4 GHz, hỗ trợ WPA/WPA2.
- Chuẩn điện áp hoạt động: 3.3V.
- Chuẩn giao tiếp nối tiếp UART với tốc độ Baud lên đến 115200
- Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.
- Hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA\_PSK, WPA2\_PSK, WPA\_WPA2\_PSK.
- Hỗ trợ: Wi-Fi Direct (P2P), soft-AP.
- Tích hợp TCP/IP protocol stack.
- Tích hợp TR switch, balun, LNA, power amplifier and matching network
- Tích hợp bộ nhân tần số, ổn áp..

## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

- LED chỉ báo truyền nhận TX / RX

### c. Sơ đồ kết nối module wifi.

Chân VCC và chân CH\_PD (chân kích hoạt chip) kết nối với nguồn 3.3V của mạch điều khiển. Chân GND và chân GPIO15 nối với GND của mạch, chân TXD nối với chân RX của vi điều khiển, chân RXD nối với chân TX của vi điều khiển.



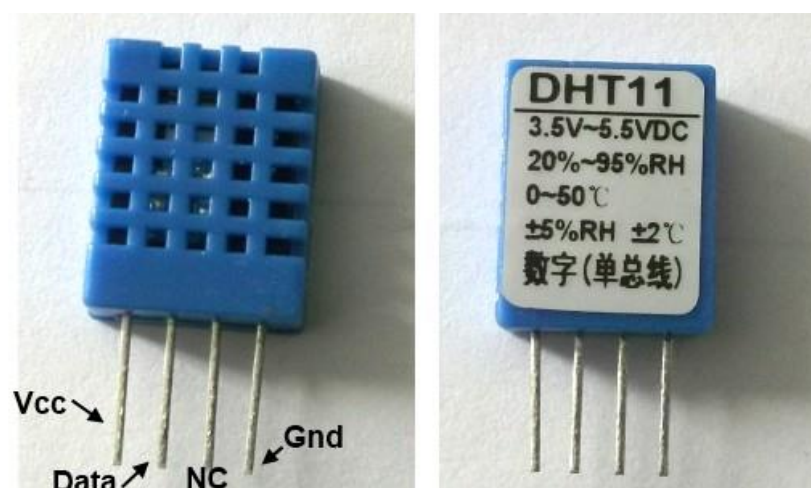
Hình 3.15. Sơ đồ nguyên lý giao tiếp module ESP8266 với vi điều khiển

### 3.2.8. Khối cảm biến.

#### a. Cảm biến nhiệt độ, độ ẩm DHT11

##### ❖ Giới thiệu.

DHT11 là cảm biến nhiệt độ và độ ẩm, ra đời sau và được sử dụng thay thế cho dòng SHT1x ở những nơi không cần độ chính xác cao về nhiệt độ và độ ẩm. Hiện nay nó được sử dụng rộng rãi do chi phí rẻ và dễ lấy dữ liệu thông qua giao tiếp one wire.



[www.hshop.vn]

Hình 3.16. Cảm biến DHT11.

### ❖ Thông số kỹ thuật và nguyên lý hoạt động.

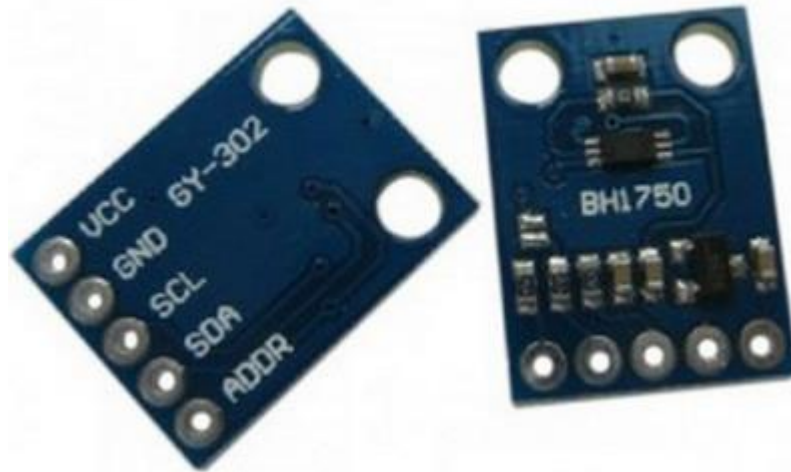
DHT11 có cấu tạo 4 chân như hình 3.16, nó có các thông số kỹ thuật sau:

- Nguồn cấp: 3-5V
- Dòng sử dụng: 2.5 mA max
- Đo tốt ở độ ẩm 20% - 80% với sai số  $\pm 5\%$
- Đo tốt ở nhiệt độ  $0^{\circ}\text{C}$  đến  $50^{\circ}\text{C}$  sai số  $\pm 2^{\circ}\text{C}$
- Tần số lấy mẫu tối đa 1Hz

### b . Cảm biến ánh sáng BH1750.

Thông thường với các đề tài đơn giản thì để đo cường độ ánh sáng ta có thể sử dụng quang trở để đọc giá trị ADC rồi tính toán. Nhưng với yêu cầu về độ chính xác cao nên nhóm em quyết định chọn cảm biến ánh sáng BH1750.

Cảm biến ánh sáng BH1750 dùng để đo cường độ ánh sáng giao tiếp theo chuẩn I2C . Cảm biến có bộ ADC nội và bộ tiền sử lý nên kết quả là giá trị cường độ ánh sáng lux mà không cần phải xử lý tính toán chuyển đổi nào nữa.



[www.hshop.vn]

**Hình 3.17.** Cảm biến ánh sáng BH1750.

Thông số kỹ thuật:

- Nguồn cung cấp: 3-5V.
- Độ phân giải 1 – 65535 lux.
- Giao tiếp theo chuẩn truyền I2C.
- Khả năng chống nhiễu sáng ở tần số 50 Hz/60 Hz.
- Sự biến đổi ánh sáng nhỏ (+/- 20%).
- Tiêu hao nguồn ít.

Một vài ví dụ về độ rọi của ánh sáng:

- Ban đêm: 0.001 - 0.02 lx.
- Trời sáng trắng: 0.02 - 0.3 lx.
- Trời mây trong nhà: 5 - 50 lx.
- Trời mây ngoài trời: 50 - 500 lx.
- Trời nắng trong nhà: 100- 1000 lx.



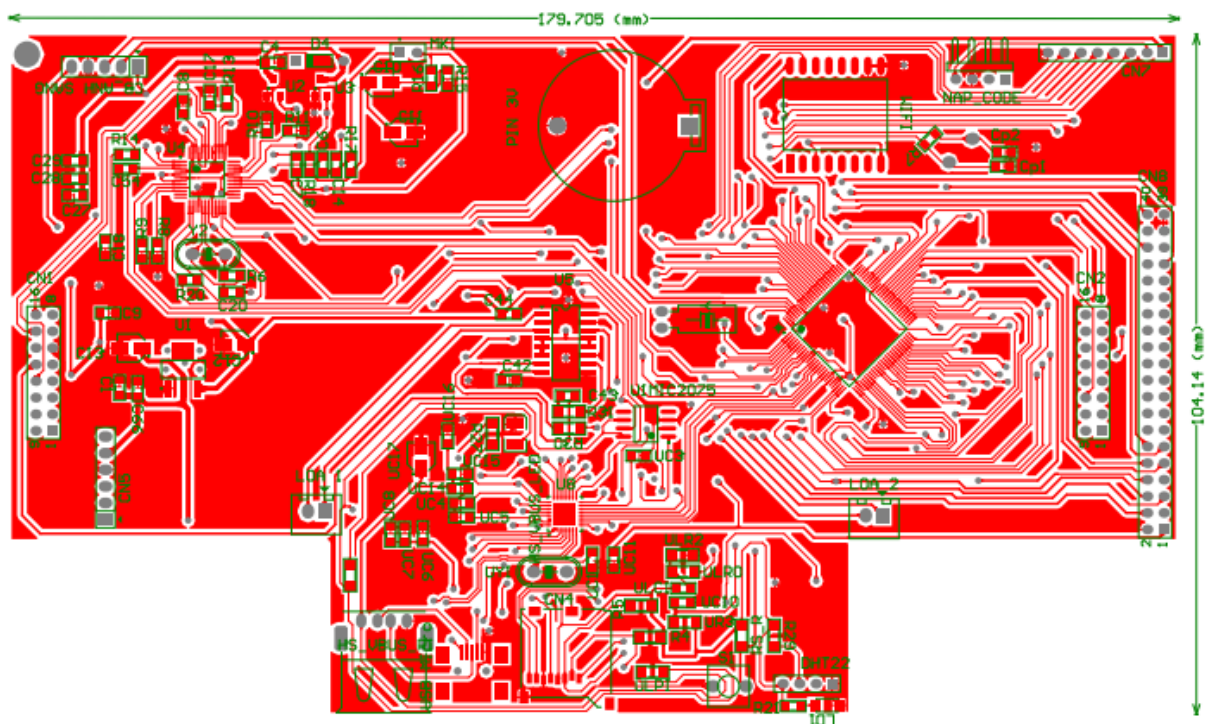
## **Chương 4. THI CÔNG HỆ THỐNG**

### **4.1 GIỚI THIỆU**

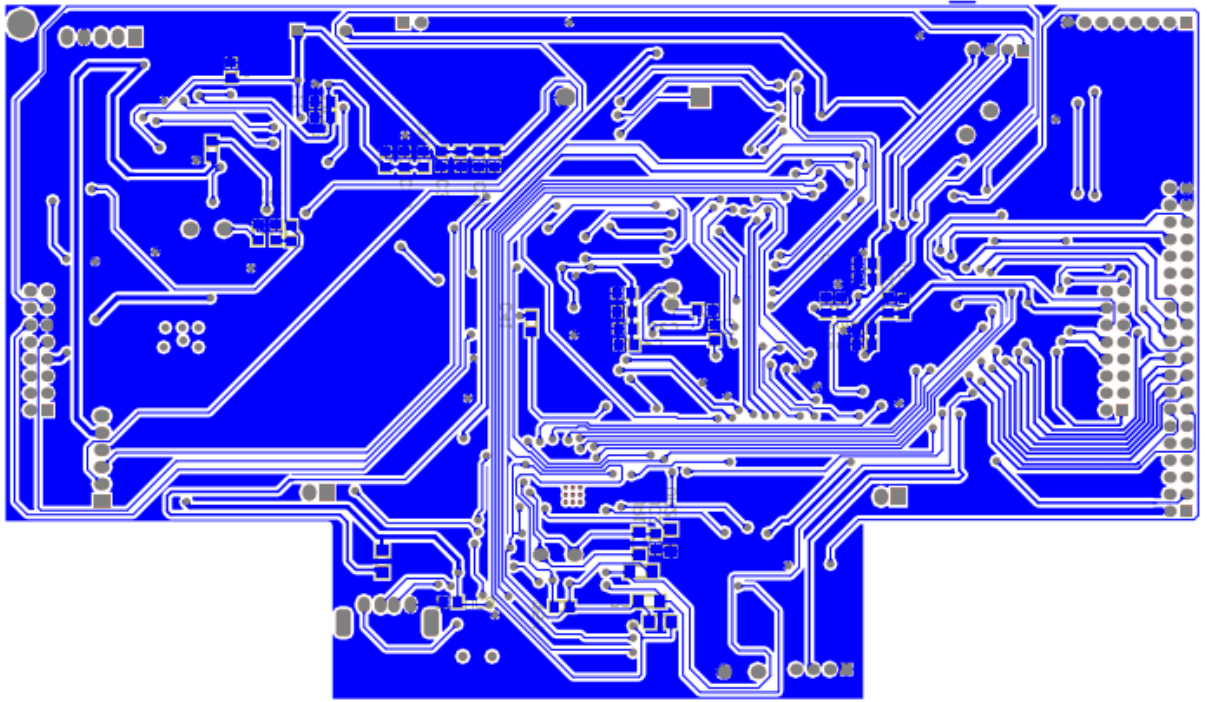
Sau khi tính toán và thiết kế nhóm đã thi công phần mạch công suất và mạch điều khiển. Mạch công suất bao gồm khối nguồn, khối công suất, mạch dò điểm không. Mạch điều khiển gồm các khối vi điều khiển, khối mp3, khối SD card, khối cảm biến và giao tiếp với module wifi.

### **4.2 THI CÔNG HỆ THỐNG**

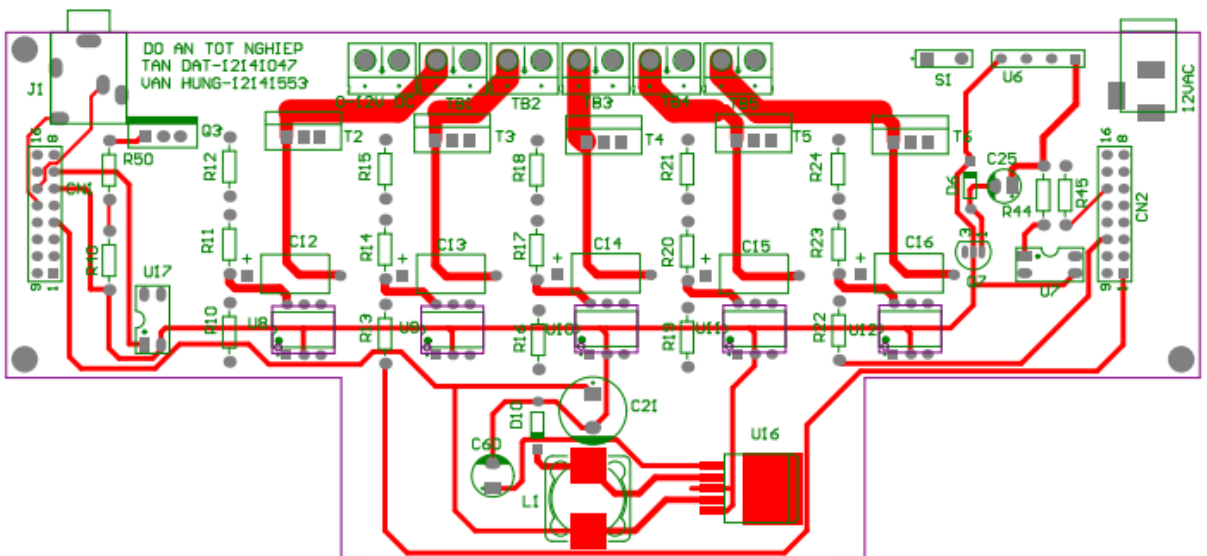
#### **4.2.1 Thi công bo mạch**



**Hình 4.1.** Lớp Top PCB mạch điều khiển.

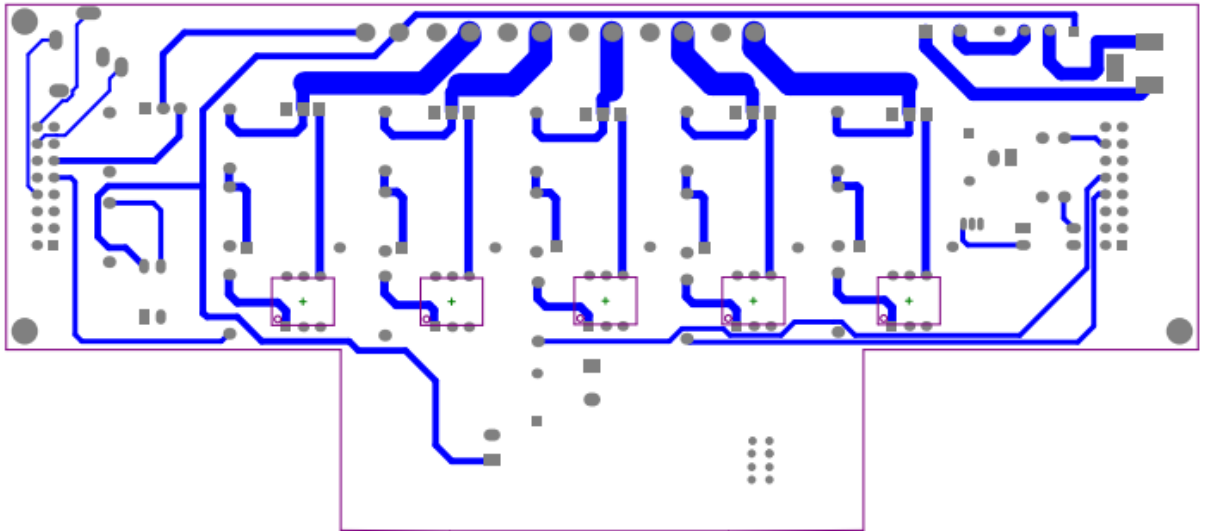


**Hình 4.2.** Lớp Bottom PCB mạch điều khiển.



**Hình 4.3.** Lớp Top PCB mạch công suất

## CHƯƠNG 4. THI CÔNG HỆ THỐNG



**Hình 4.4.** Lớp Bottom PCB mạch công suất.

**Bảng 4.1.** Danh sách các linh kiện chính.

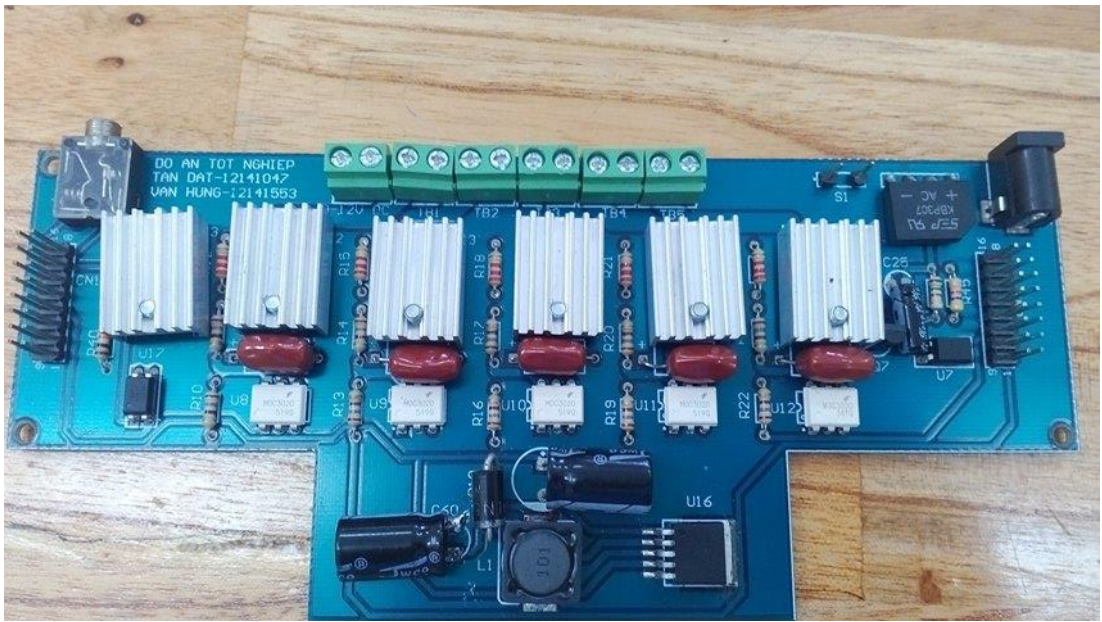
STT	Tên linh kiện	Giá trị	Dạng vỏ	Chú thích
1	IC LM2576	5V-3A	TO-263	
2	TRIAC BTA16	600V-16A	TO-220	Có tản nhiệt
3	MOC3020		DIP-6	
4	Tụ áp cao 100nF	100nF		
5	PC817		DIP-4	
6	IRF640		TO-220	
7	KBP307	700V-3A		
8	ASM1117-3.3V	3.3V-1A	SOT-223	
9	RT9166-25PXL	2.5V-0.3A	SOT-23	
10	RT9166-28PXL	2.8V-0.3A	SOT-23	
11	VS1003		LQFP-48	
12	STM32F407VET6		LQFP-100	
13	PAM8403	5V-3W	SOP-16	



## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.2.2 Lắp ráp và kiểm tra

#### a. Lắp ráp mạch công suất.



**Hình 4.5.** Kết quả thi công mạch công suất.

Mạch công suất bao gồm khối công suất, khối nguồn và khối dò điểm không. Hình là kết quả thi công của mạch công suất.

#### b. Lắp ráp khối điều khiển.

Kết quả thi công lắp ráp của mạch điều khiển như hình



**Hình 4.6.** Kết quả thi công mạch điều khiển.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

#### 4.3.1 Đóng gói bộ điều khiển

Hộp điều khiển được tính toán và thiết kế dự trên kích thước thực tế sau khi thi công khối công suất và khối nguồn, hộp có một ngõ cắm nguồn DC kết nối với mạch công suất, một ngõ ra jack 3.5 của mạch mp3, ngõ ra để gắn thẻ nhớ SD và công tắc nguồn.



**Hình 4.7.** Hộp điều khiển sau khi thi công.



**Hình 4.8.** Mặt trên của hộp điều khiển.



### 4.3.2 Thi công mô hình.



**Hình 4.9.** Vị trí công tắc nguồn.

Công tắc nguồn của bộ điều khiển được thiết kế nằm bên mép phải của hộp điều khiển nhằm giúp người dùng dễ thao tác.



**Hình 4.10.** Mặt dưới của hộp điều khiển.

Mặt dưới hộp điều khiển có ngõ cắm thẻ nhớ ở giữa và hai bên là các lỗ loa.



**Hình 4.11.** Mặt trên của hộp điều khiển.

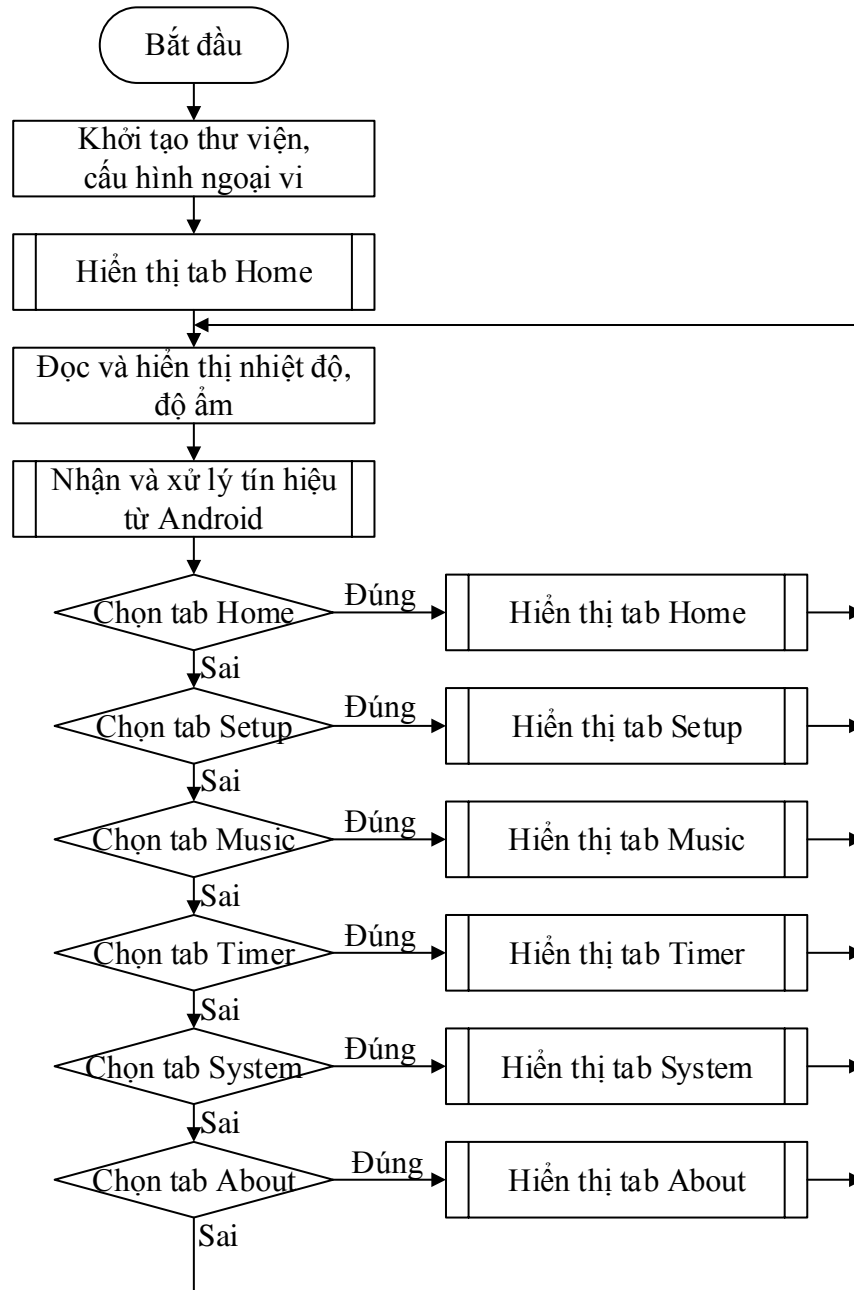
Mặt trên của hộp điều khiển gồm có ngõ cấp nguồn DC, các cổng kết nối thiết bị và ngõ cắm tai nghe.

### 4.4 LẬP TRÌNH HỆ THỐNG

#### 4.4.1 Lưu đồ giải thuật

a . Lưu đồ giải thuật cho vi điều khiển.

- *Lưu đồ chương trình chính.*



**Hình 4.12.** Lưu đồ giải thuật chương trình chính của vi điều khiển.

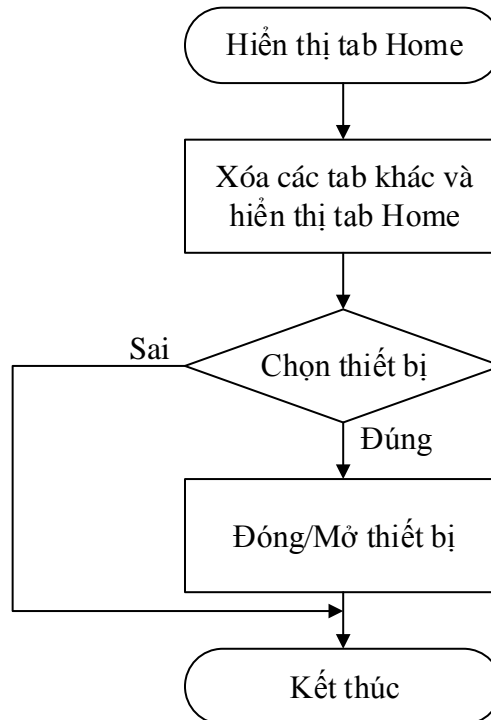
Khi được cấp nguồn vi điều khiển sẽ khởi tạo các biến và thư viện liên quan. Sau đó vi điều khiển tải giao diện tab Home. Tiếp theo đó là sẽ đọc các giá trị nhiệt độ, độ ẩm, cường độ ánh sáng để hiển thị lên màn hình và chờ tín hiệu điều khiển từ màn

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

---

hình cảm ứng hay người dùng nhấn vào các tab để chuyển qua các tab tương ứng. Quá trình này sẽ lặp đi lặp lại.

- **Lưu đồ giải thuật của chương trình con tab Home.**



**Hình 4.13.** Lưu đồ giải thuật tab Home.

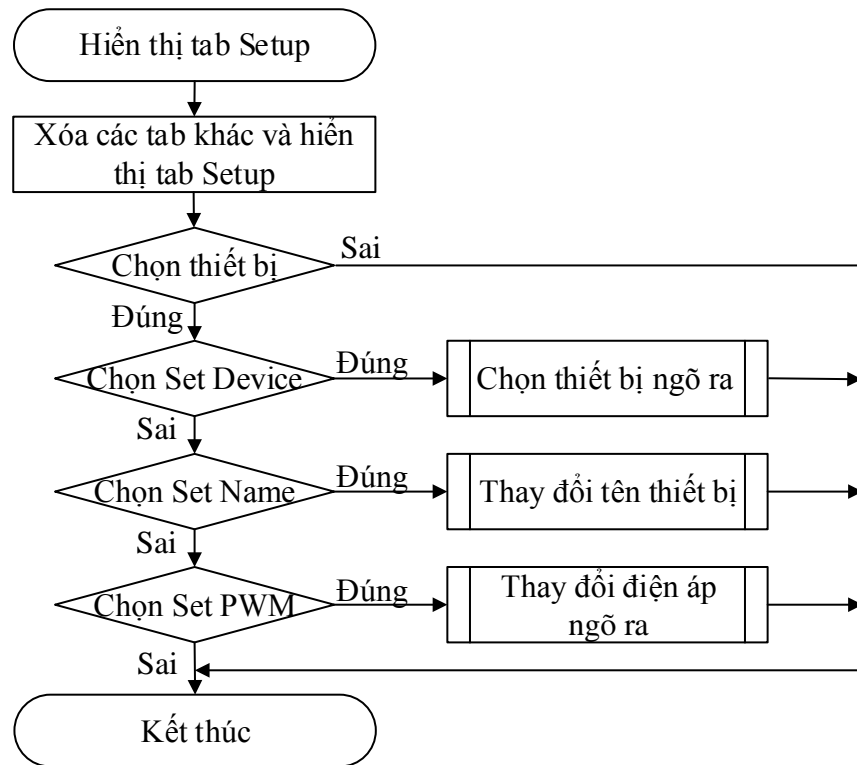
Tab này có chức năng đảo trạng thái của nút được nhấn.

- **Lưu đồ chương trình con tab Setup.**

Khi nhấn chọn vào một thiết bị thì bảng tùy chọn điều khiển sẽ hiện ra bao gồm các chức năng:

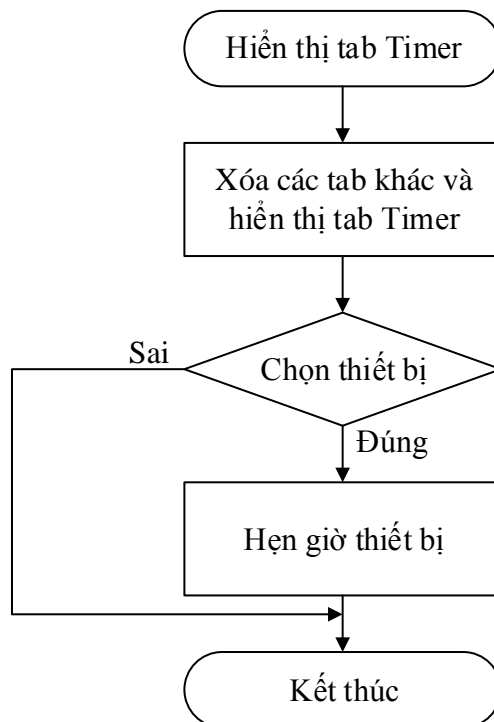
- Chọn Set Device cho phép thêm thiết bị mới hoặc xóa thiết bị hiện tại
- Chọn Set Name cho phép thay đổi tên của thiết bị.
- Chọn Set PWM cho phép thay đổi điện áp hiệu dụng của thiết bị để điều khiển tốc độ của quạt và độ sáng của bóng đèn.





**Hình 4.14.** Lưu đồ giải thuật tab Setup.

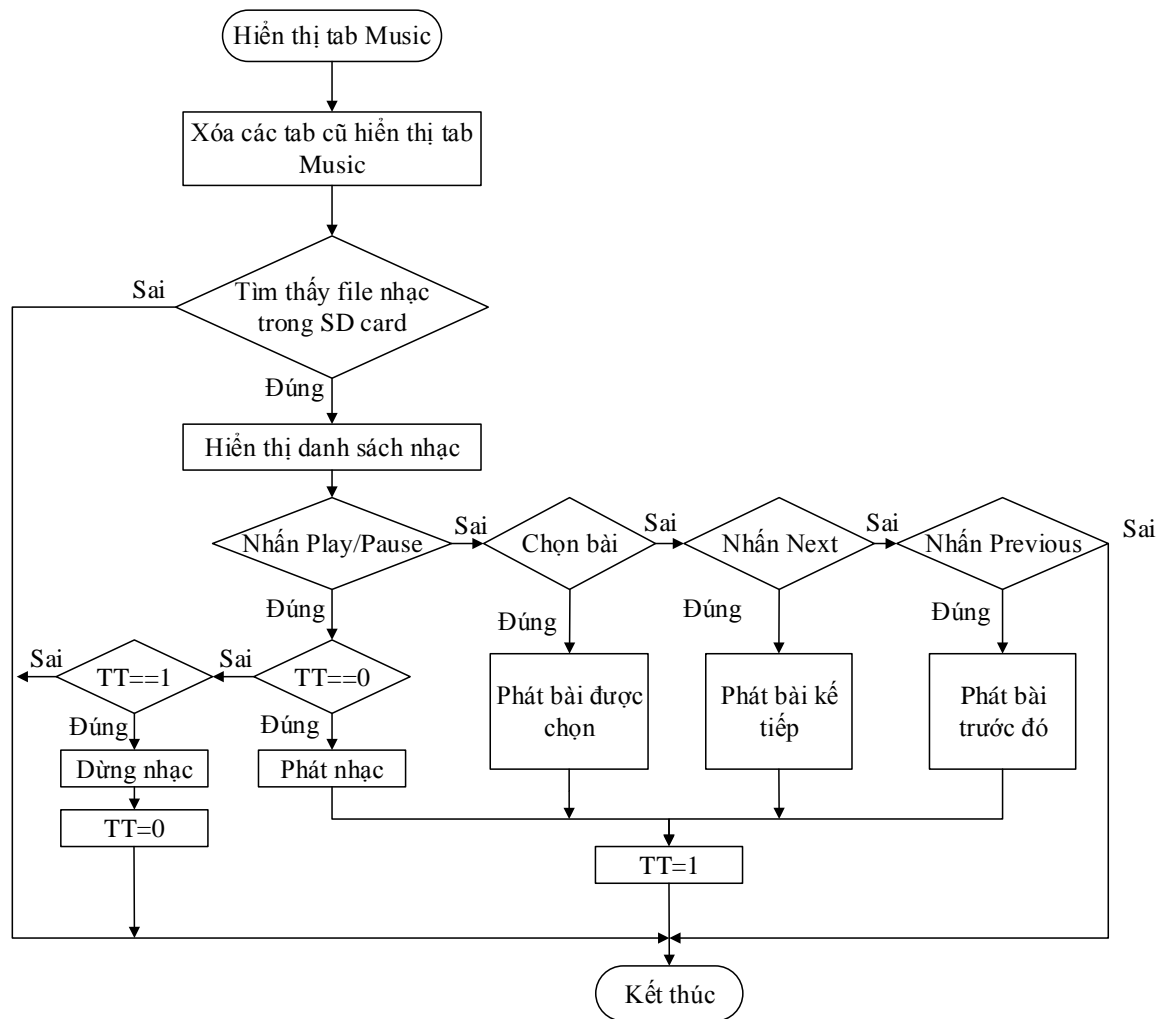
- *Lưu đồ chương trình con Timer.*



**Hình 4.15.** Lưu đồ giải thuật tab Timer.

Khi nhấn vào nút trong tab Timer, sẽ hiện ra một bảng cho phép đặt giờ bật, tắt thiết bị, thêm, xóa lịch trình hẹn giờ.

### - Lưu đồ chương trình con tab Music.

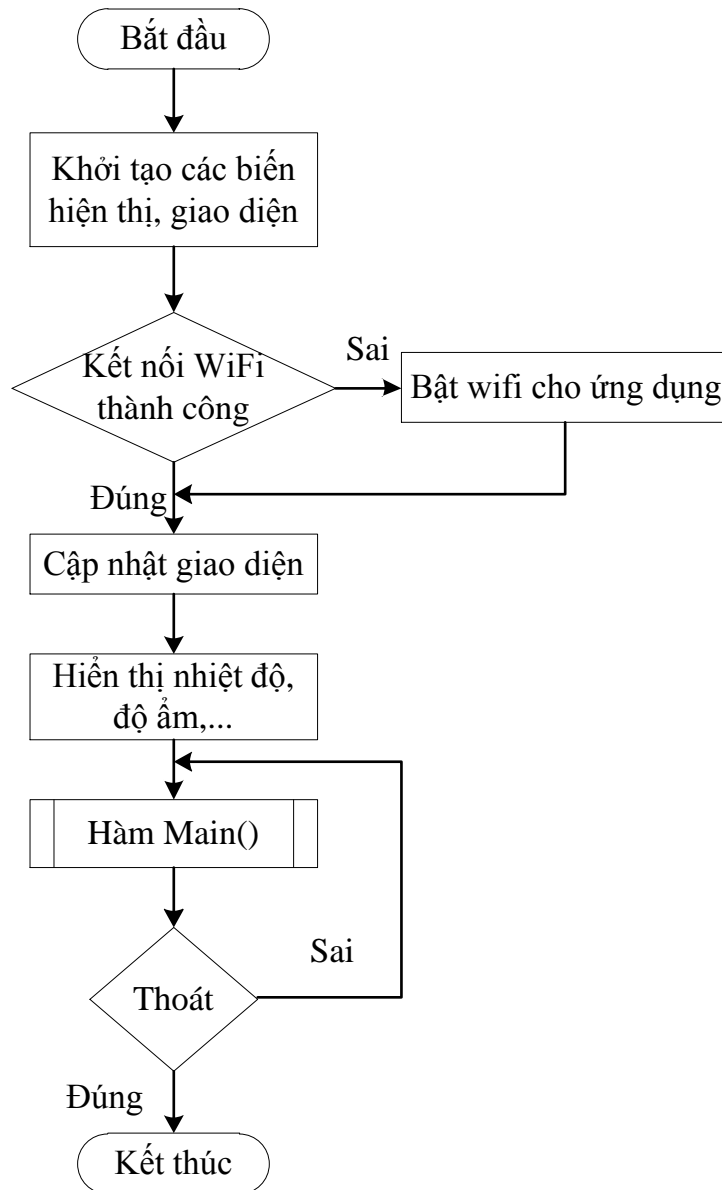


**Hình 4.16.** Lưu đồ giải thuật tab Music.

Khi tab Music được chọn lần đầu, vi xử lý đọc trong thẻ SD tìm file âm thanh, nếu không có thì thông báo và thoát. Nếu có tiến hành hiển thị danh sách bài hát. Chọn vào các nút Play, Pause, Next, Previous sẽ xử lý phát nhạc ứng với các nút.

### b . Lưu đồ giải thuật cho ứng dụng Android.

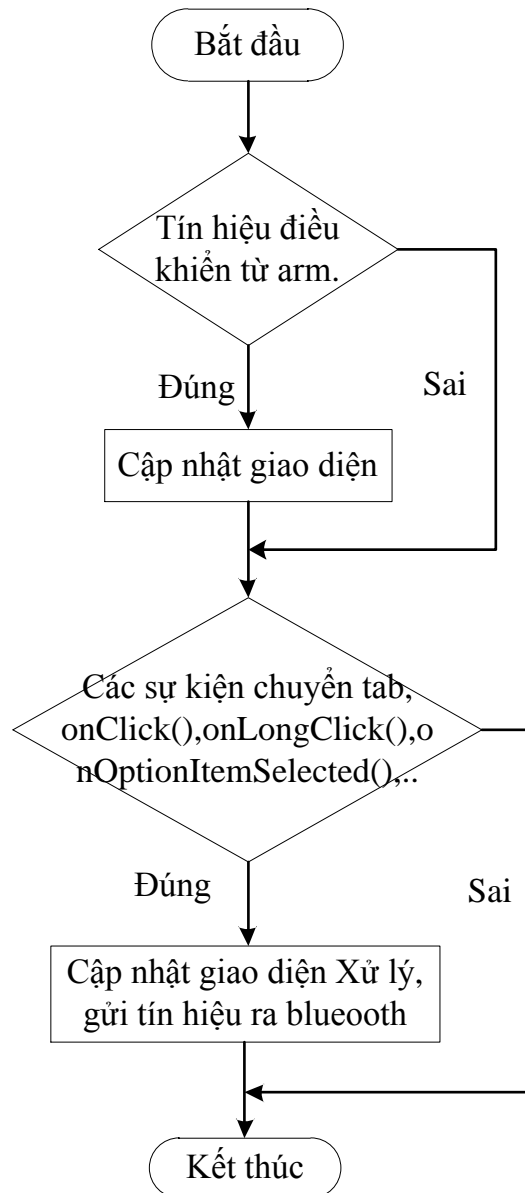
#### - Lưu đồ chương trình Android.



**Hình 4.17.** Lưu đồ Activity Main.

Ban đầu khi khởi động, ứng dụng sẽ khởi tạo các biến, tab và cập nhật giao diện. Sau đó ứng dụng sẽ kiểm tra thiết bị đã bật wifi chưa, nếu chưa thì ứng dụng sẽ tự động bật wifi, sau khi kết nối với wifi ứng dụng sẽ gửi lệnh yêu cầu cập nhật giao diện, lúc này vi điều khiển sẽ gửi lệnh đồng bộ cho giao diện cho ứng dụng. Người dùng có thể điều khiển thiết bị thông qua giao diện của ứng dụng.

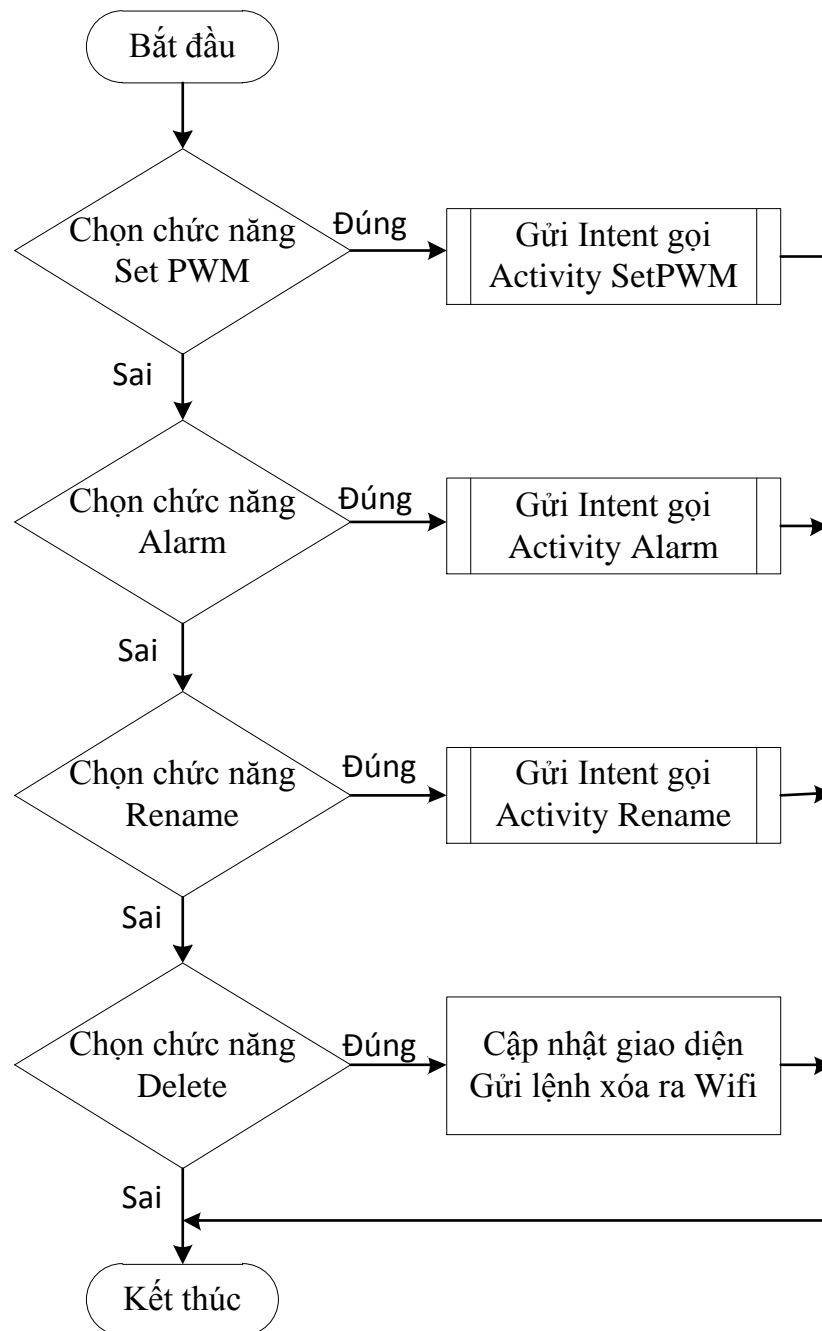
#### - Lưu đồ hàm Run.



**Hình 4.18.** Lưu đồ giải thuật hàm Run().

Trong hàm Run() này chương trình sẽ kiểm tra phía dưới vi điều khiển có gửi tín hiệu đồng bộ hay không, nếu có thì ứng dụng sẽ cập nhật giao diện, nếu không thì ứng dụng sẽ kiểm tra người dùng có tạo ra các sự kiện chuyển tab hay nhấn vào các nút trên màn hình ứng dụng hay không nếu có thì sẽ nhảy vào các chương trình con của các sự kiện.

- Lưu đồ giải thuật sự kiện *onOptionsItemSelected*



**Hình 4.19.** Lưu đồ hàm *onOptionsItemSelected()*.

Hàm này nằm trong *Activity List\_Control*, có chức năng hiển thị danh sách tùy chỉnh khi xảy ra sự kiện *onLongClick()* trên các *Item*. Khi nhấn vào các tùy chỉnh thì ứng dụng sẽ Intent tới *Activity* tương ứng.

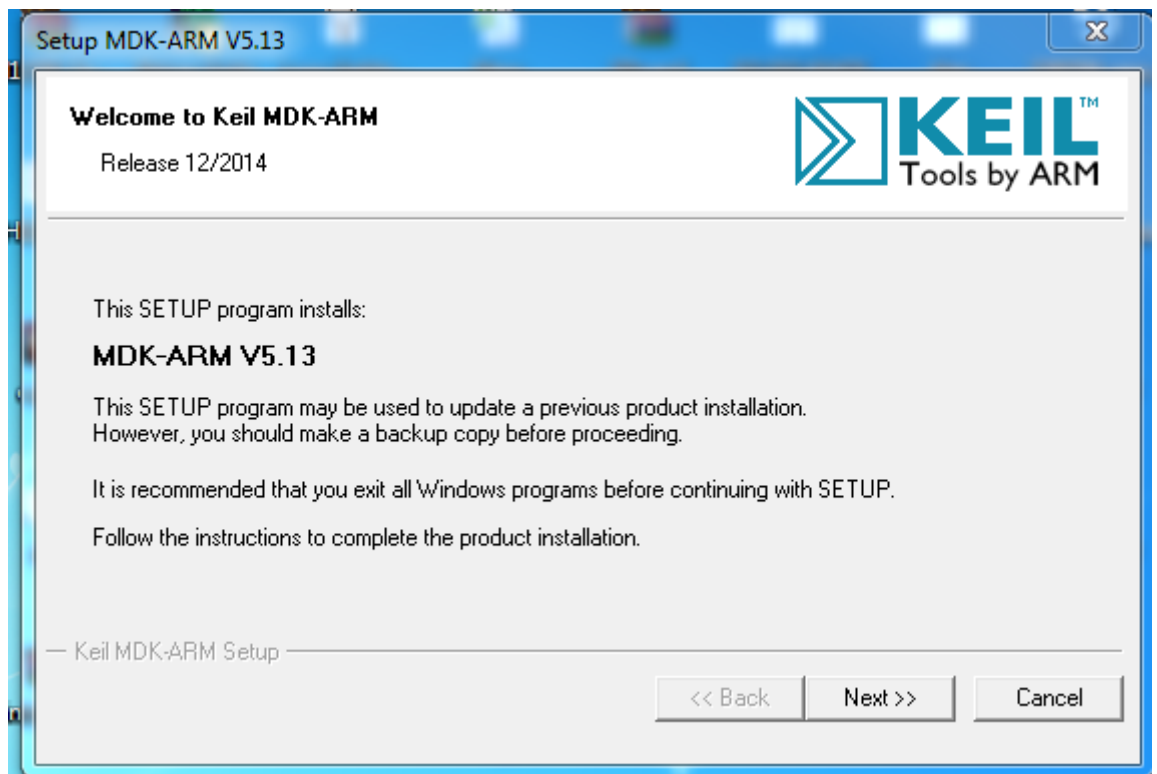
### 4.4.2 Phần mềm lập trình cho vi điều khiển.

#### a. Giới thiệu phần mềm lập trình

Hiện nay có rất nhiều phần mềm phục vụ cho việc lập trình vi điều khiển như Keil C, IAR, CCS, .... Sau khi tìm hiểu các phần mềm lập trình cho STM32 thì nhóm em quyết định chọn phần mềm Keil C V5 do phần mềm này được khá là nhiều người sử dụng, các bản Keil C sau này hỗ trợ nhiều bộ thư viện ngoại vi phát triển và hỗ trợ debug tốt.

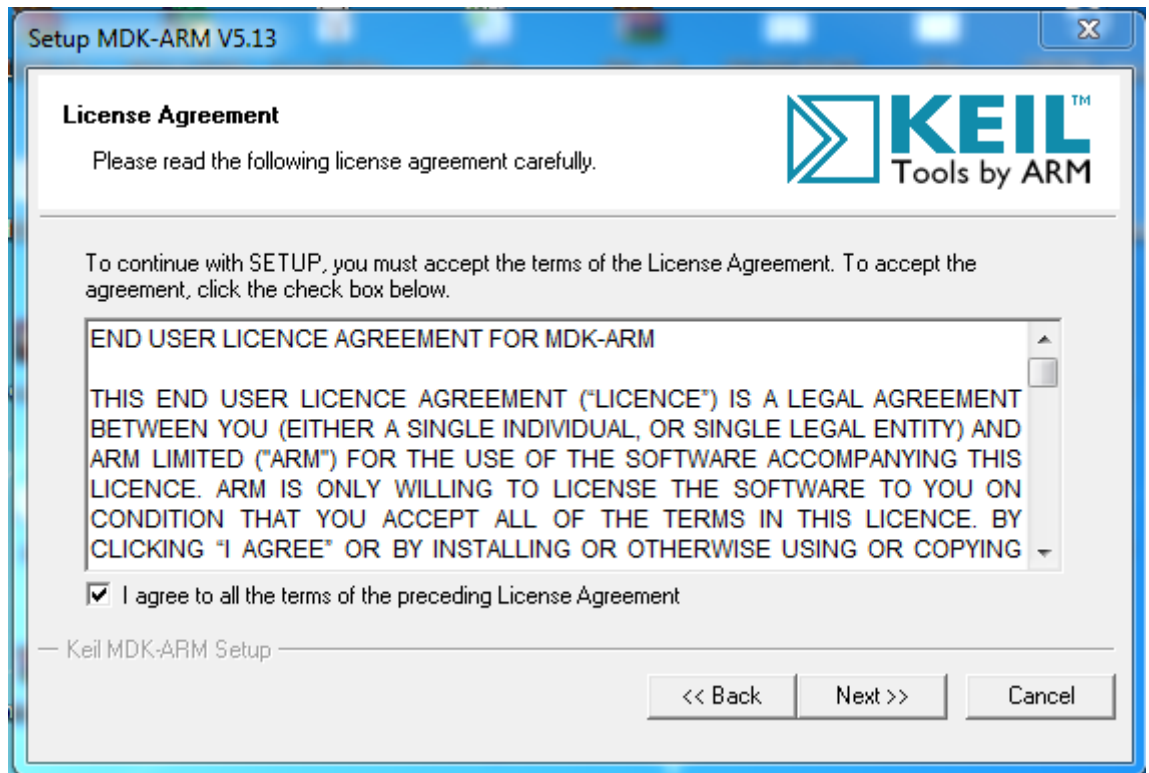
Các bước cài đặt phần mềm Keil C V5 như sau:

- Đầu tiên các bạn vào trang web của hãng Keil C để tải phần mềm Keil C V5 về. Tiếp theo chạy file mdk511.exe. Hộp thoại cài đặt xuất hiện và chọn Next.



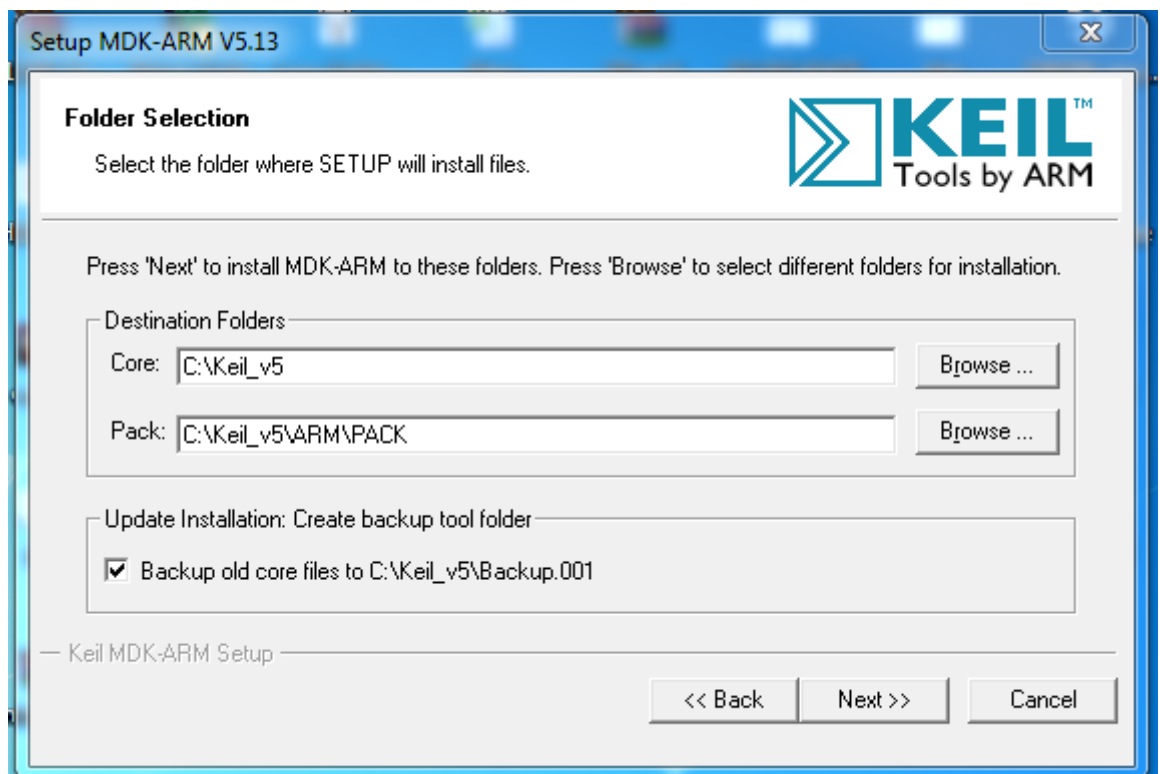
**Hình 4.20.** Hộp thoại cài đặt Keil C V5.

- Sau đó tích vào ô “*I agree to all the terms of the preceding License Agreement*” tiếp theo chọn Next.



**Hình 4.21.** Hộp thoại license cài đặt Keil C V5 .

- Tiếp theo bạn sẽ chọn đường dẫn lưu thư mục cài đặt ở mục Core, sau đó chọn Next.



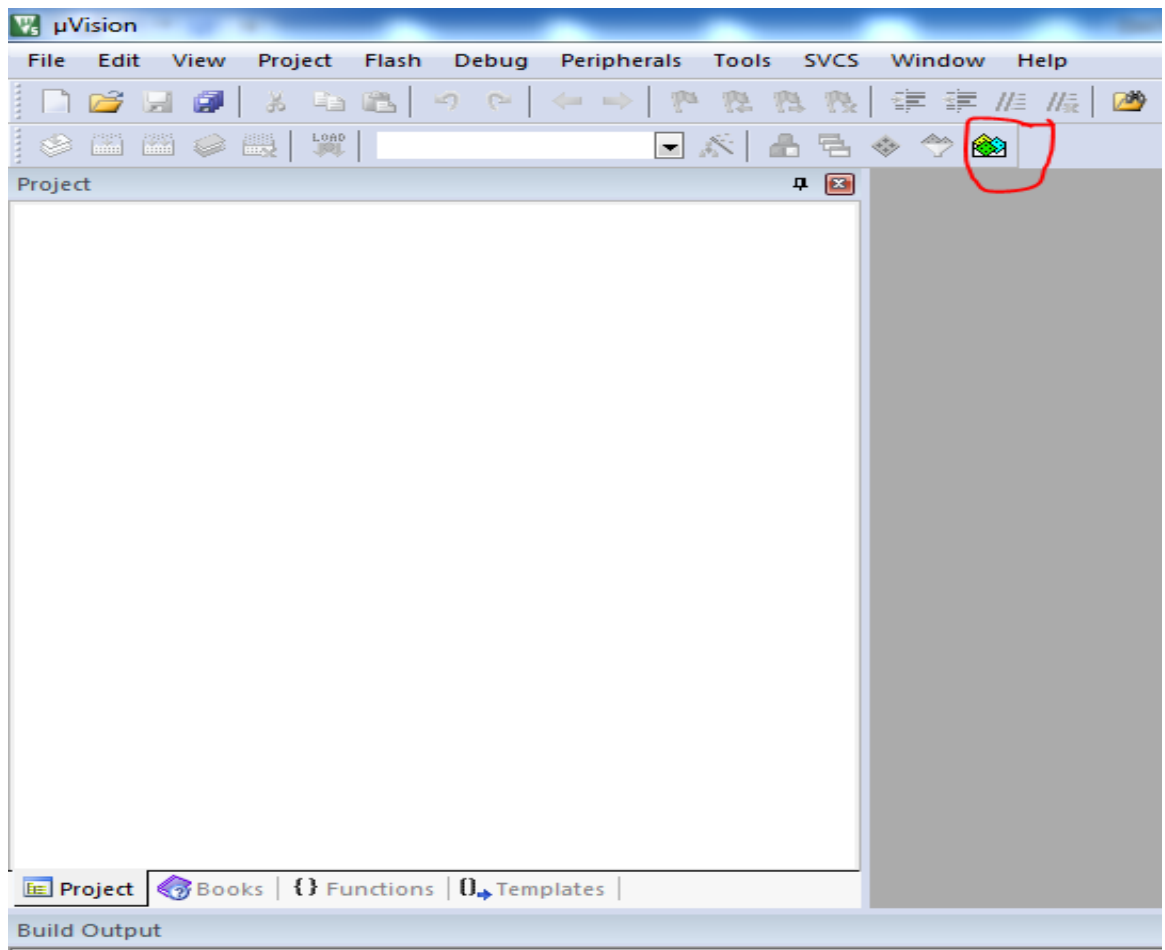
**Hình 4.22.** Hộp thoại chọn nơi lưu thư mục cài đặt Keil C.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Sau đó hộp thoại “ Customer Information “ xuất hiện, các bạn điền đầy đủ thông tin và chọn Next. Sau đó các bạn chờ máy tiến hành cài đặt và chọn Finish sau khi đã cài đặt xong.

Để lập trình cho vi điều khiển STM32F4, chúng ta cần phải tải bản Pack của STM32F4 sau đó tiến hành add bản Pack đó vào Keil C . Việc thêm Pack tiến hành như sau:

- Đầu tiên các bạn mở bản Keil C V5 lên sau đó chọn vào mục Pack Installer như trong hình:

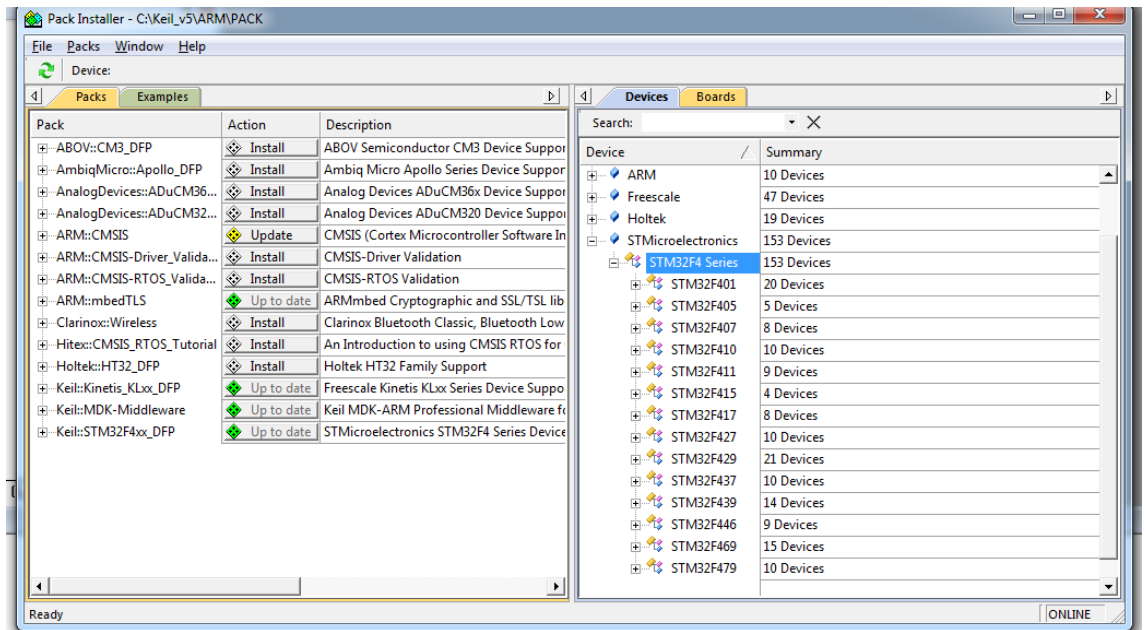


**Hình 4.23.** Thêm Pack STM32F4 cho Keil C V5.

- Sau đó các bạn chọn File-> Import và chọn vào bản Pack mà bạn muốn cài đặt. Sau khi cài đặt xong ta sẽ được như hình sau:



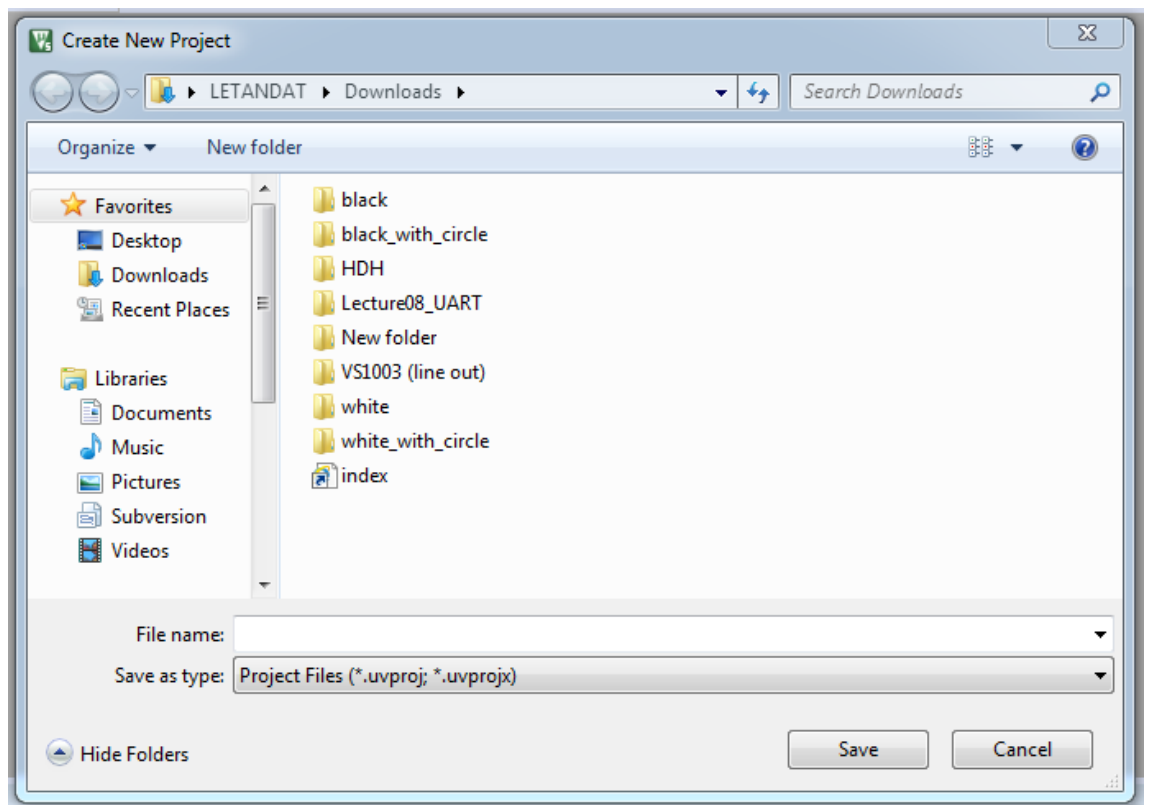
## CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.24. Bản Pack STM32F4.

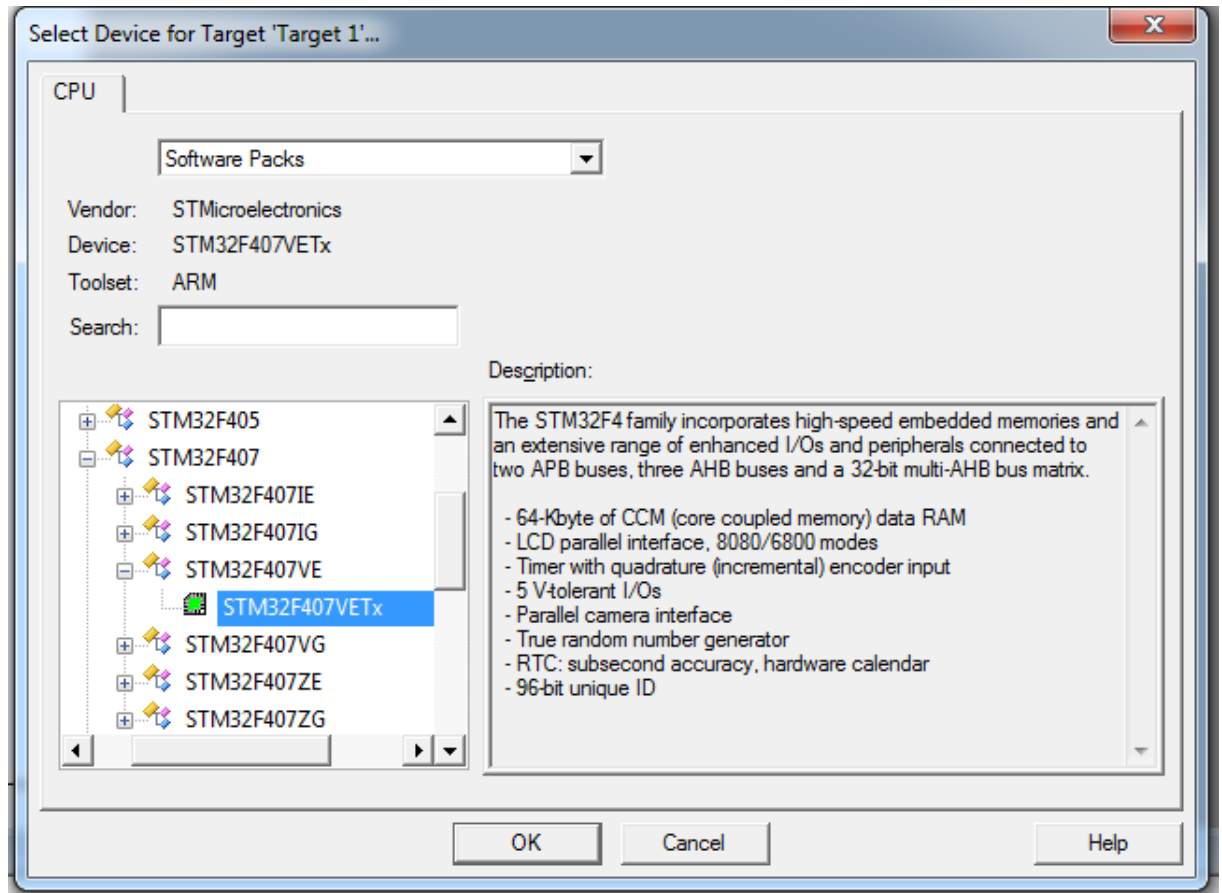
Sau khi thêm bản pack cho STM32F4 thì chúng ta có thể viết code cho dòng vi điều khiển STM32F407VET6. Sau đây nhóm sẽ trình bày các bước tạo một project mẫu như sau:

- Sau khi mở phần mềm Keil C lên ta chọn **Project -> New uVision Project** sau đó sẽ xuất hiện hộp thoại như hình:



Hình 4.25. Hộp thoại chọn nơi lưu project.

- Sau đó chọn thư mục lưu project -> save. Sau khi nhấn save sẽ xuất hiện 1 hộp thoại cho phép ta chọn dòng vi điều khiển ta muốn lập trình. Ở đây mình chọn dòng STM32F407VETx của hãng STMicroelectronics.



**Hình 4.26.** Chọn dòng vi điều khiển để lập trình.

- Sau khi thực hiện bước trên thì ta đã tạo được một project cơ bản.

### **b. Viết chương trình hệ thống**

Do chương trình tương đối dài nên nhóm xin phép không trình bày mà sẽ để trong đĩa CD.

### 4.4.3 Phần mềm lập trình cho điện thoại android.

Trước đây, để lập trình ứng dụng android thì các lập trình viên thường sử dụng phần mềm eclipse. Nhưng hiện nay Google đã cho ra đời phần mềm chính thức dùng để lập trình ứng dụng android đó là Android Studio. Với những ưu điểm như: hỗ trợ tốt hơn việc bố cục ứng dụng cho nhiều thiết bị khác nhau, đảm bảo rằng app viết ra có thể tương thích với màn hình của cả smartphone lẫn tablet, hỗ trợ công cụ mô phỏng, ... đó là lý do nhóm quyết định sử dụng phần mềm Android Studio để lập trình ứng dụng android cho người dùng.

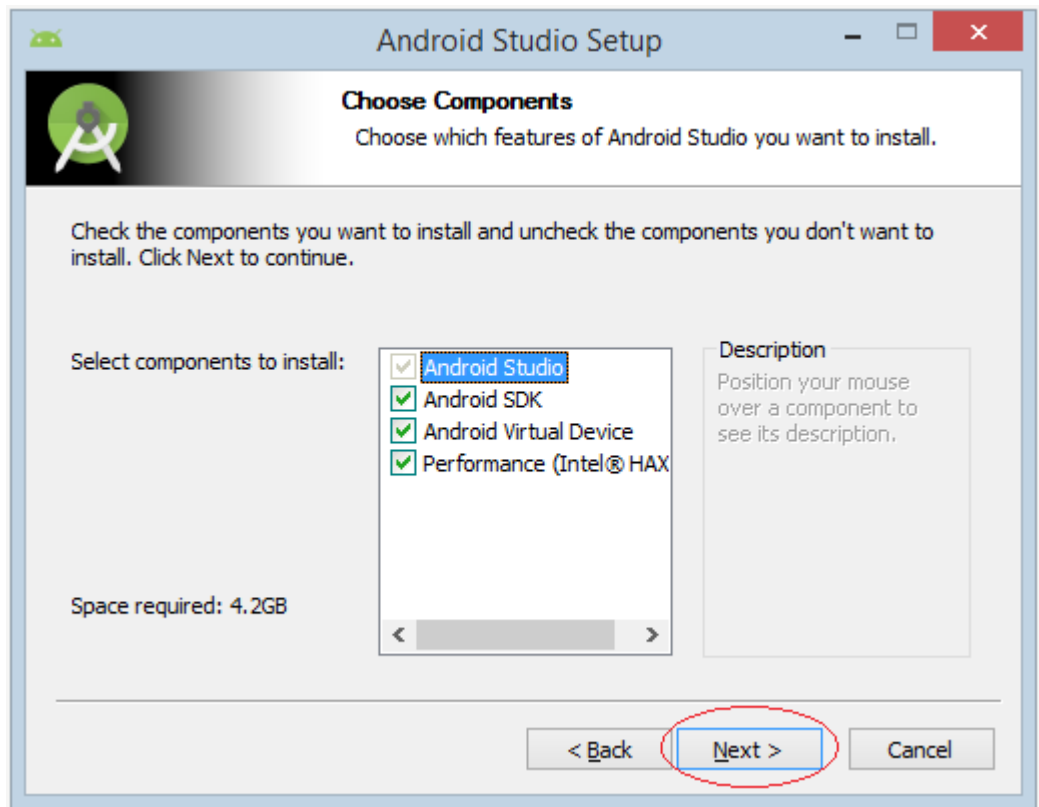
Các bước cài đặt Android studio:

- Đầu tiên các bạn cần phải cài đặt bản Java phù hợp với phiên bản Android Studio mình sử dụng.
- Sau đó vào trong web <https://developer.android.com> để tải Android Studio về.
- Tiếp theo Run file.exe mới tải về. Sau đó sẽ xuất hiện hộp thoại Setup, bạn chọn Next.



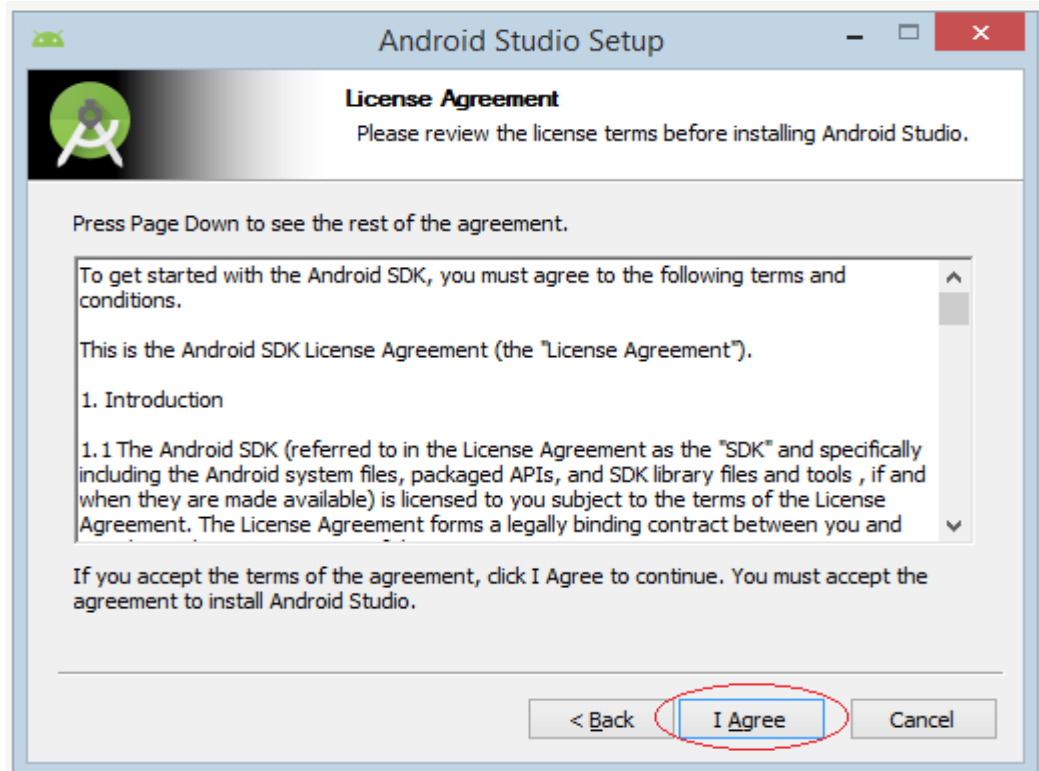
**Hình 4.27.** Hộp thoại cài đặt Android Studio.

- Tiếp theo chọn các option và chọn Next.



**Hình 4.28.** Chọn các option của Android Studio.

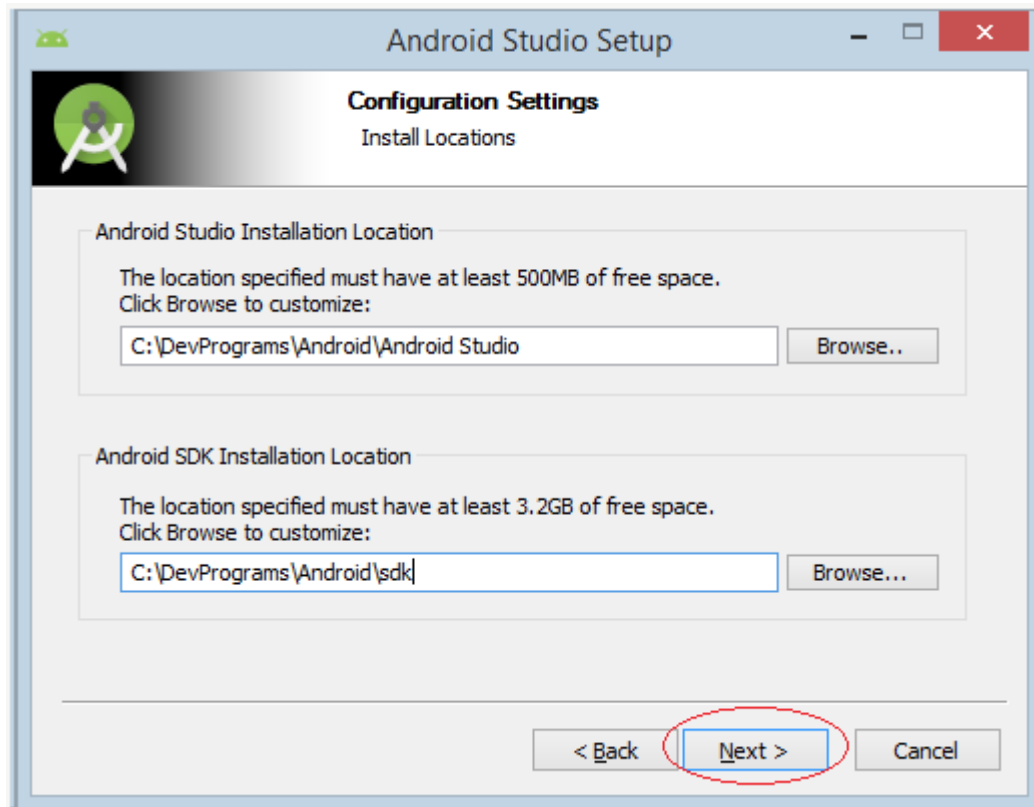
- Sau đó hộp thoại quản lý giấy phép sẽ xuất hiện. Các bạn chọn “ I Agree”.



**Hình 4.29.** Hộp thoại quản lý giấy phép Android Studio.

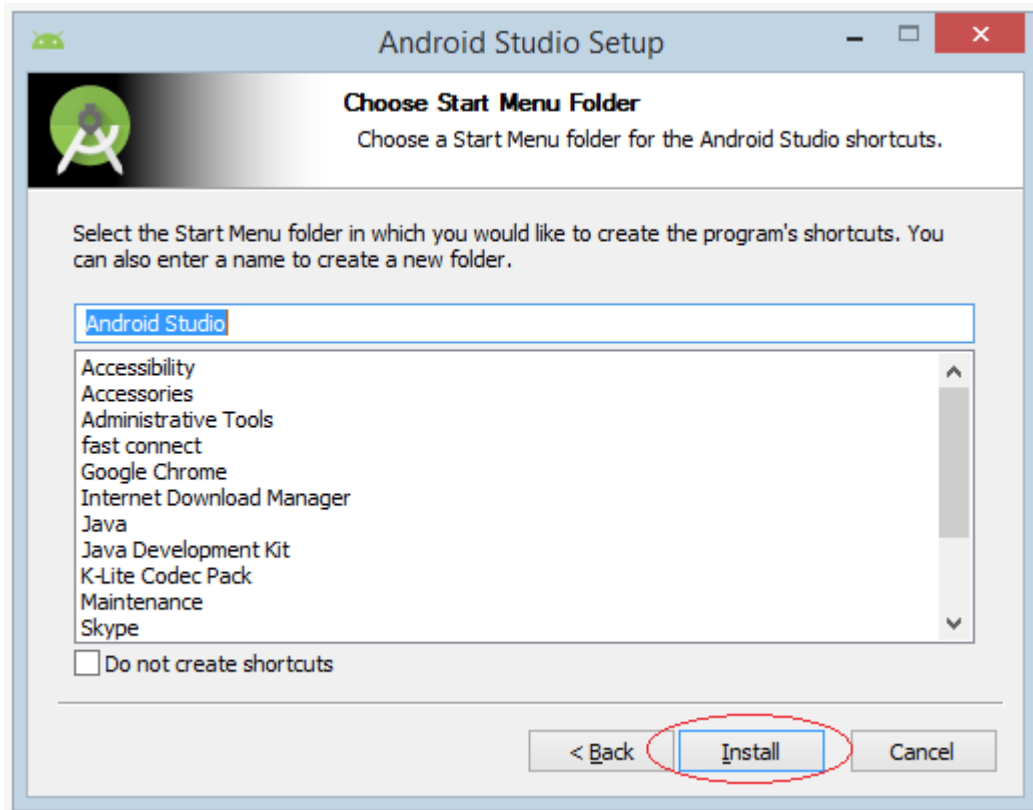
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Sau đó các bạn chọn thư mục lưu bản cài đặt và bản SDK. Tiếp theo chọn Next.



**Hình 4.30.** Hộp thoại chọn nơi lưu bản cài đặt Android Studio.

- Tiếp theo bạn chọn cấu hình RAM dành cho bộ quản lý tang tốc phần cứng (Nên chọn khoảng 2GB) và chọn Next.
- Sau đó hộp thoại Install xuất hiện và chọn Install. Sau đó bạn chờ máy tính cài đặt xong và chọn Finish.



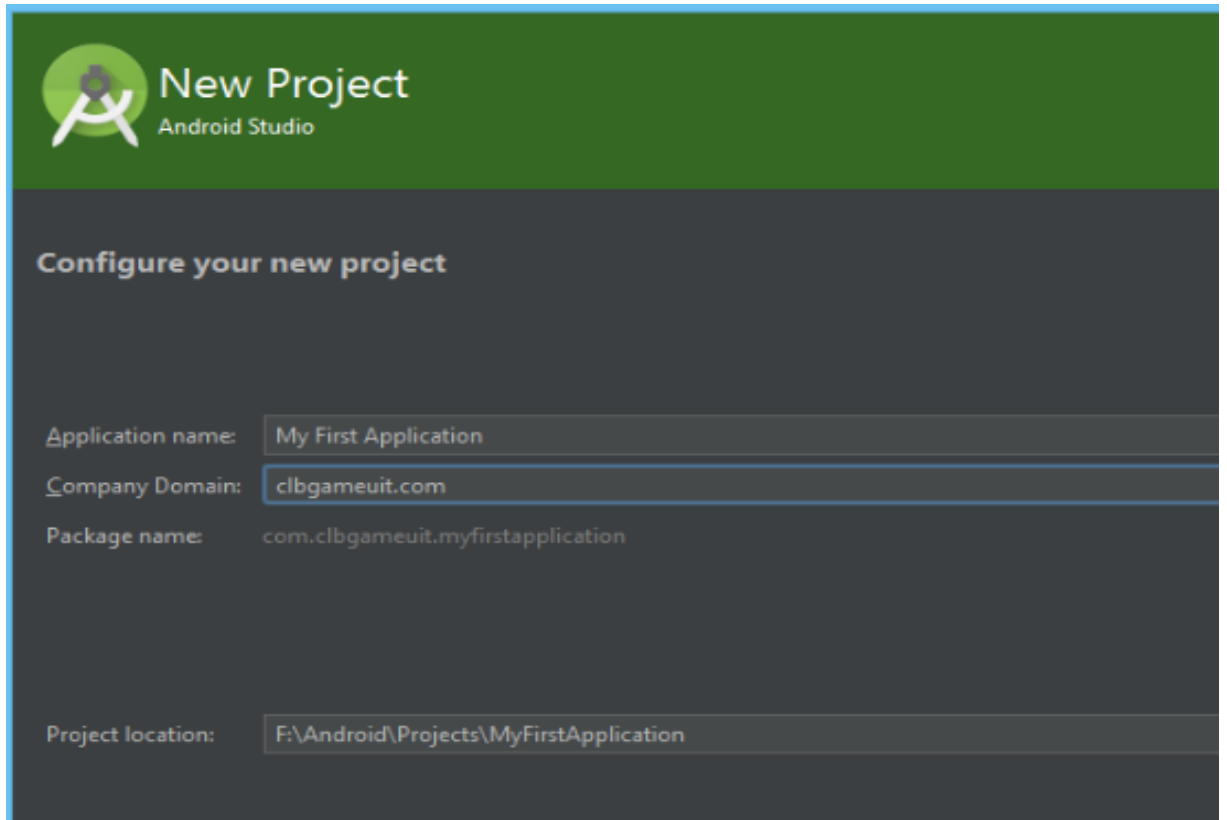
**Hình 4.31.** Hộp thoại Install Android Studio.

Để tạo một project Anroid cơ bản thì ta cần làm các bước sau:

- Đầu tiên nhấp đúp vào biểu tượng của Android Studio, sau đó sẽ xuất hiện hộp thoại như hình 4.32, chọn “Start a new Android Studio Project”.
- Tiếp theo điền tên ứng dụng vào khung “Application name” và thư mục lưu project ở khung “Project location” như hình 4.33. Sau đó chọn Next.



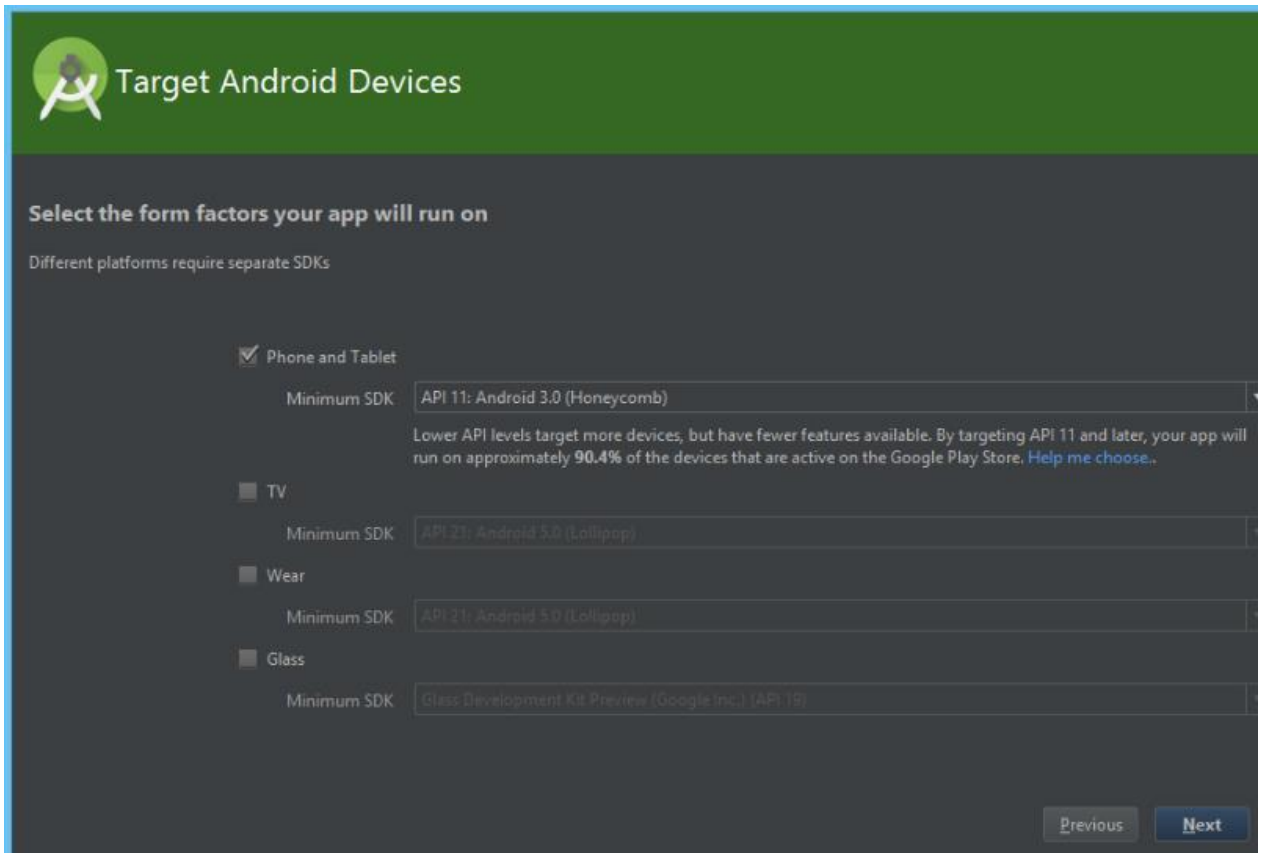
**Hình 4.32.** Hộp thoại Quick Start của Android Studio.



**Hình 4.33.** Đặt tên cho project android.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

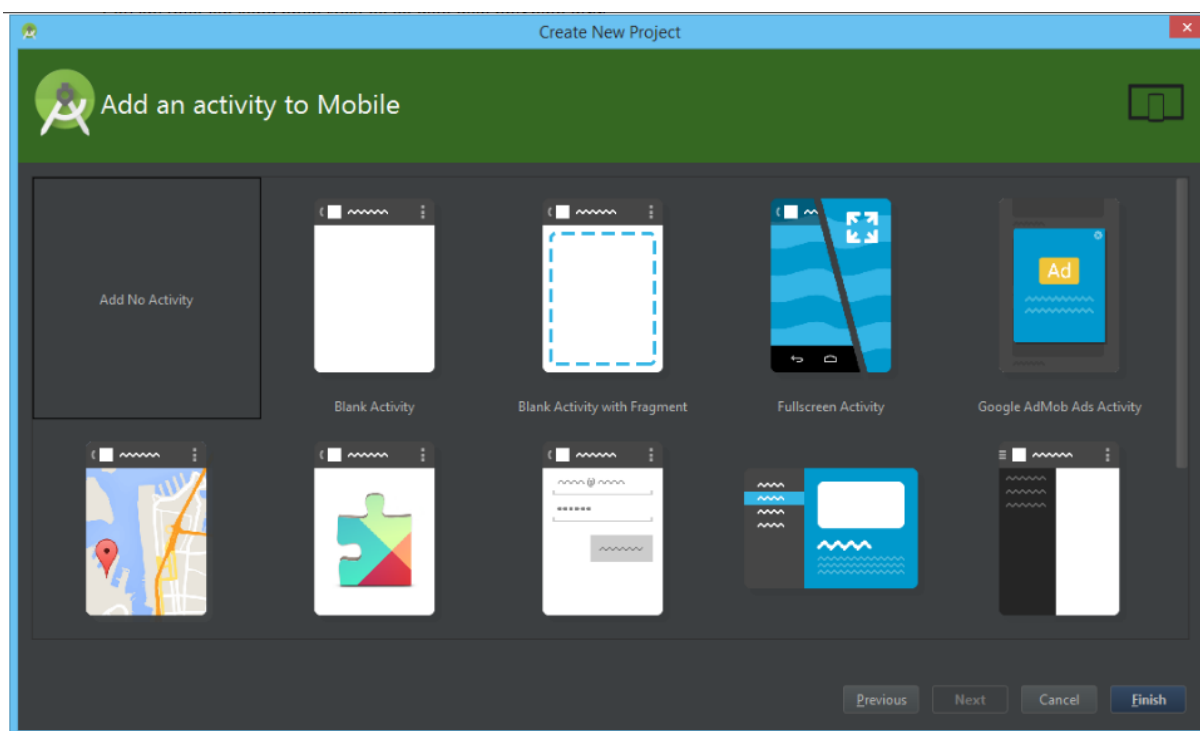
- Sau đó sẽ xuất hiện hộp thoại cho phép chọn bản SDK và thiết bị chạy ứng dụng. Bản SDK càng nhỏ thì hỗ trợ các nhiều thiết bị nhưng ích chức năng hơn. Chọn Next.



**Hình 4.34.** Hộp thoại cho phép chọn bản SDK.

- Tiếp theo sẽ xuất hiện 1 hộp thoại cho phép chọn giao diện màn hình bắt đầu của ứng dụng. Sau khi chọn xong bấm Finish như hình 4.35.





**Hình 4.35.** Hộp thoại chọn giao diện bắt đầu cho ứng dụng

### 4.5 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

#### 4.5.1 Viết tài liệu hướng dẫn sử dụng

Để sử dụng được thiết bị cần phải thực hiện các bước sau:

**Bước 1:** Lắp các thiết bị cần điều khiển vào mạch công suất. Sau đó cấp nguồn cho thiết bị

**Bước 2:** Cấp nguồn cho mạch điều khiển và chờ thiết bị setup.

**Bước 3:** Vào tab Setup chọn hoặc thay đổi thiết bị điều khiển cho phù hợp với thực tế cho từng button.

**Bước 4:** Quay về tab Home để thực hiện điều khiển.

**Bước 5:** Để thực hiện điều khiển bằng ứng dụng android. Các bạn cần phải bật chế độ wifi của android và kết nối vào mạng wifi của nơi bạn sử dụng thiết bị. Sau đó giao diện sẽ được cập nhật lên ứng dụng và các bạn có thể điều khiển như màn hình cảm ứng.

Ngoài ra để hẹn giờ bật tắt thiết bị thì vào tab Timer để thiết lập hoặc muốn nghe nhạc thì vào tab Music.

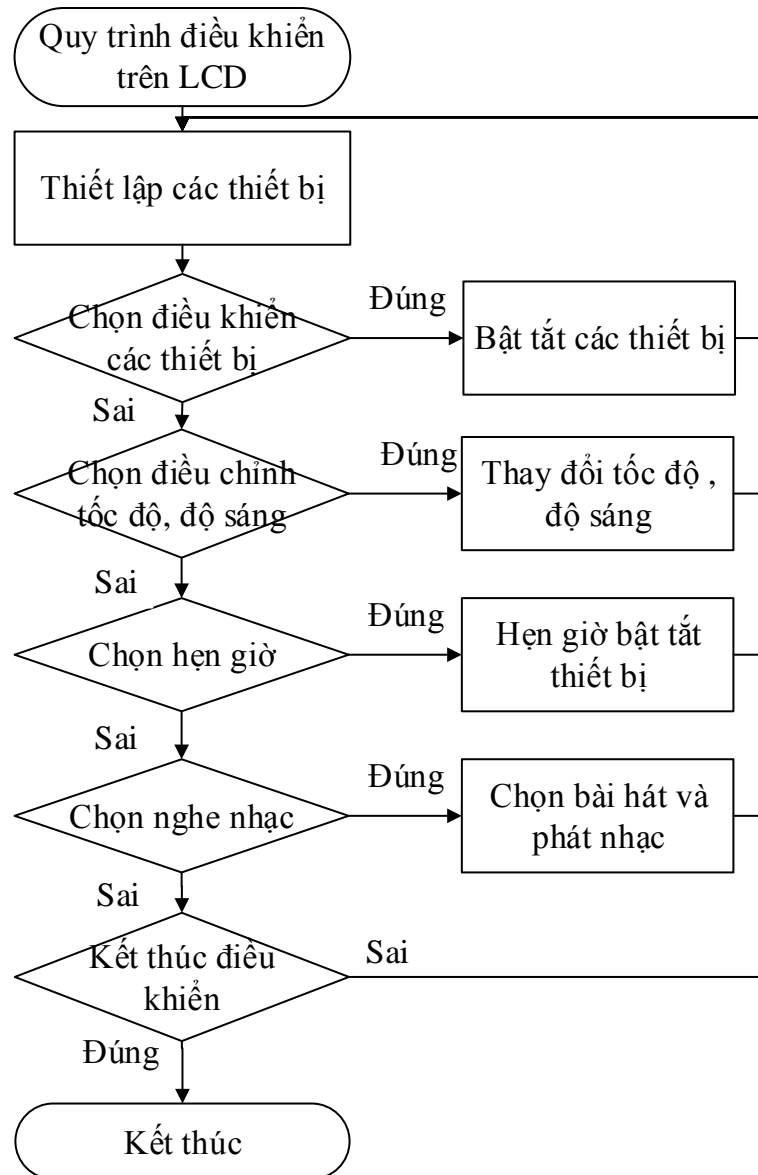
#### 4.5.2 Quy trình thao tác

Quy trình thao tác trên màn hình cảm ứng 7”:

- Điều khiển bật/tắt thiết bị: chọn tab Setup→tiến hành chọn thiết bị→quay lại tab Home→ bật/tắt thiết bị.
- Hỗ trợ thay đổi tốc độ, độ sáng cho một số thiết bị (Quạt, bóng đèn sợi đốt, van...): Chọn tab Setup→Set PWM→Thay đổi giá trị tùy chỉnh từ 0-100→Nhấn OK.
- Thay đổi tên thiết bị theo ý thích: Chọn tab Setup→Set Name→Nhập tên→Nhấn OK.
- Nghe nhạc: Chọn tab Music→ Chọn bài hát→ Play.
- Chọn bài hát: Trong tab Music chọn bài hát từ list music hoặc ấn Next/Previous.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Thay đổi độ sáng đèn nền cho LCD: vào tab System chỉnh độ sáng từ thanh trượt.



**Hình 4.36.** Quy trình thao tác trên màn hình cảm ứng.

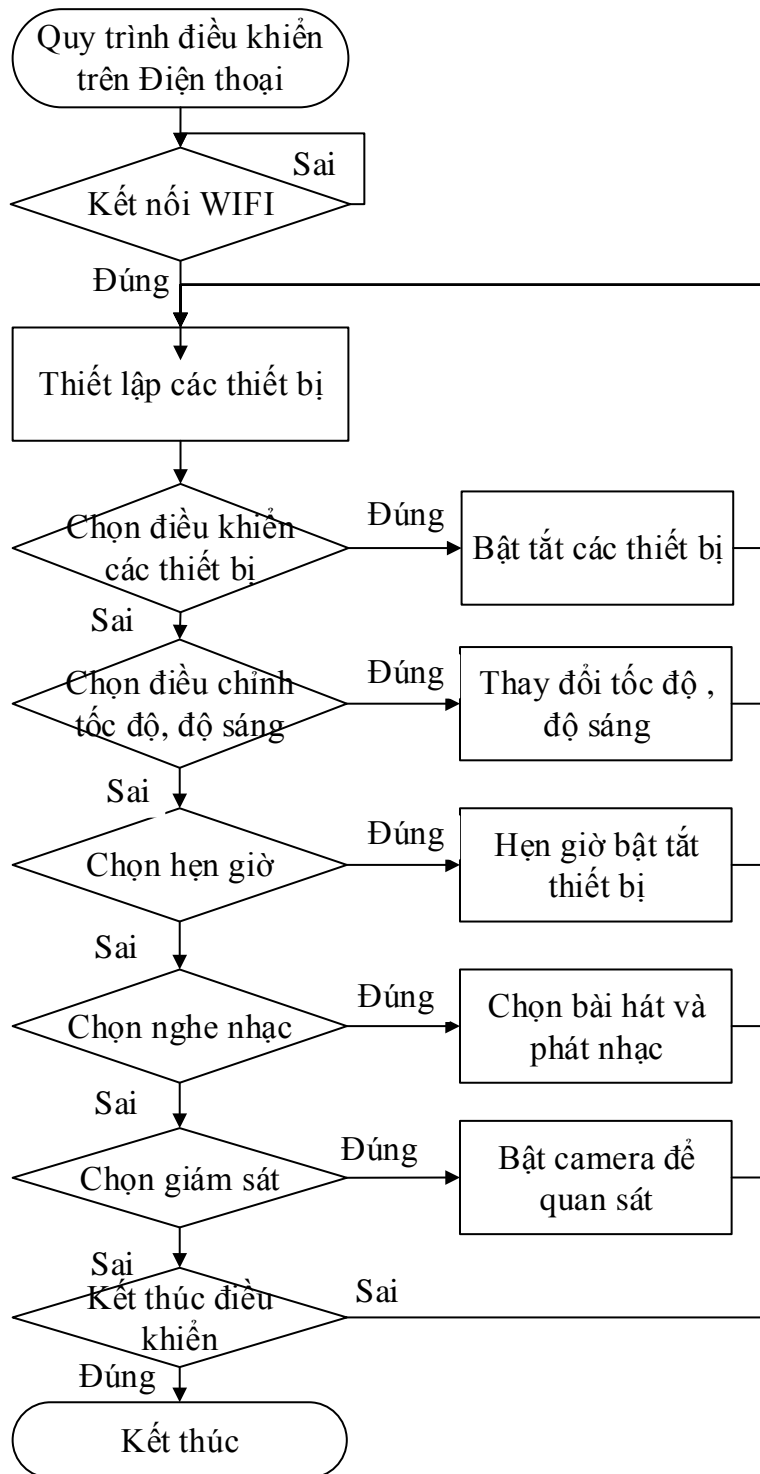
Quy trình điều khiển trên điện thoại:

- Kết nối với bộ điều khiển: chọn tab Setting → Nhập IP → Nhập Port → Chọn Connect.
- Điều khiển bật/tắt thiết bị: Chọn tab Home → tiến hành chọn thiết bị → bật/tắt thiết bị.

## **CHƯƠNG 4. THI CÔNG HỆ THỐNG**

---

- Hỗ trợ thay đổi tốc độ, độ sáng cho một số thiết bị (Quạt, bóng đèn sợi đốt, van....): Chọn tab Home → Nhấn giữ thiết bị → Set PWM → Thay đổi giá trị tùy chỉnh từ 0-100 → Nhấn OK.
- Thay đổi tên thiết bị theo ý thích: Chọn tab Home → Nhấn giữ thiết bị → Set Name → Nhập tên → Nhấn OK.
- Nghe nhạc: Chọn tab Music → Chọn bài hát → Play.
- Chọn bài hát: Trong tab Music chọn bài hát từ list music hoặc ấn Next/Previous.
- Chọn giám sát: Chọn tab Camera → tiến thành giám sát.



**Hình 4.37.** Quy trình thao tác trên điện thoại.

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

### **5.1 KẾT QUẢ**

#### **5.1.1 Kết quả nghiên cứu.**

Trong vòng 12 tuần làm đồ án nhóm nghiên cứu đã đạt được các kết quả như sau:

- Nghiên cứu sâu hơn về vi điều khiển STM32F407VET6 và dòng tương tự của STM32F4.
- Giao tiếp được với màn hình cảm ứng điện dung 7’’ theo chuẩn giao tiếp FSMC NOR FLASH.
- Nghiên cứu sử dụng được module wifi eps8266 và giao tiếp được với vi điều khiển và điện thoại.
- Tìm hiểu và nghiên cứu giao thức socket của hệ điều hành android để giao tiếp truyền dữ liệu giữa điện thoại và vi điều khiển.
- Biết cách giao tiếp với thẻ nhớ theo chuẩn SDIO và IC giải mã âm thanh VS1003.
- Tìm hiểu được cách thiết kế và sử dụng các linh kiện trong mạch công suất và mạch điều khiển.
- Tìm hiểu và biết các sử dụng phần mềm như phần mềm thiết kế mạch chuyên nghiệp Altium Designer, phần mềm lập trình vi điều khiển KeilC và phần mềm Android Studio để lập trình ứng dụng cho android.
- Thi công thành công mạch công suất và mạch điều khiển.

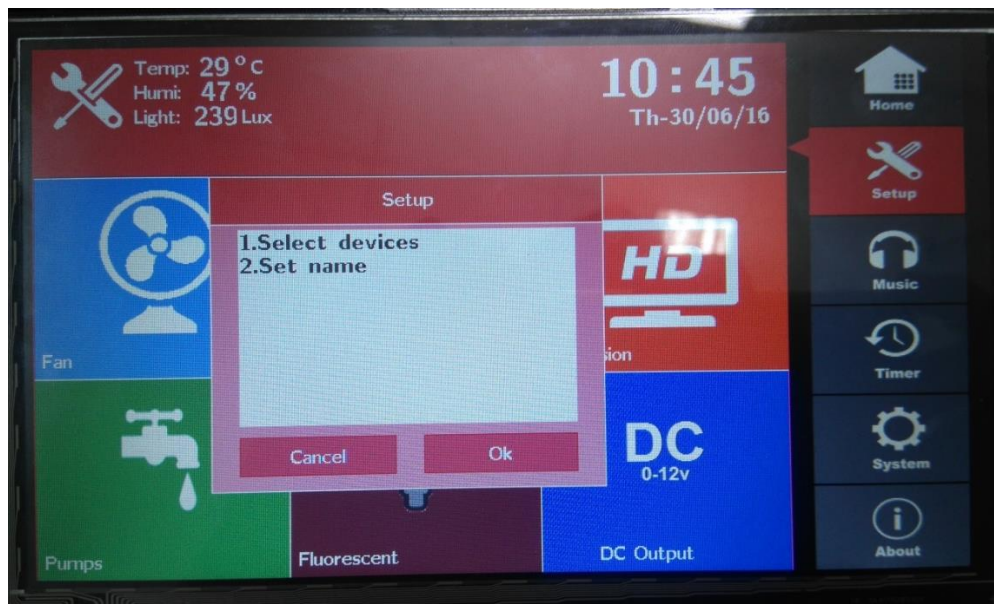
### 5.1.2 Kết quả thi công.

- ❖ Kết quả thi công giao diện người dùng trên hộp điều khiển.



**Hình 5.1.** Giao diện tab Home.

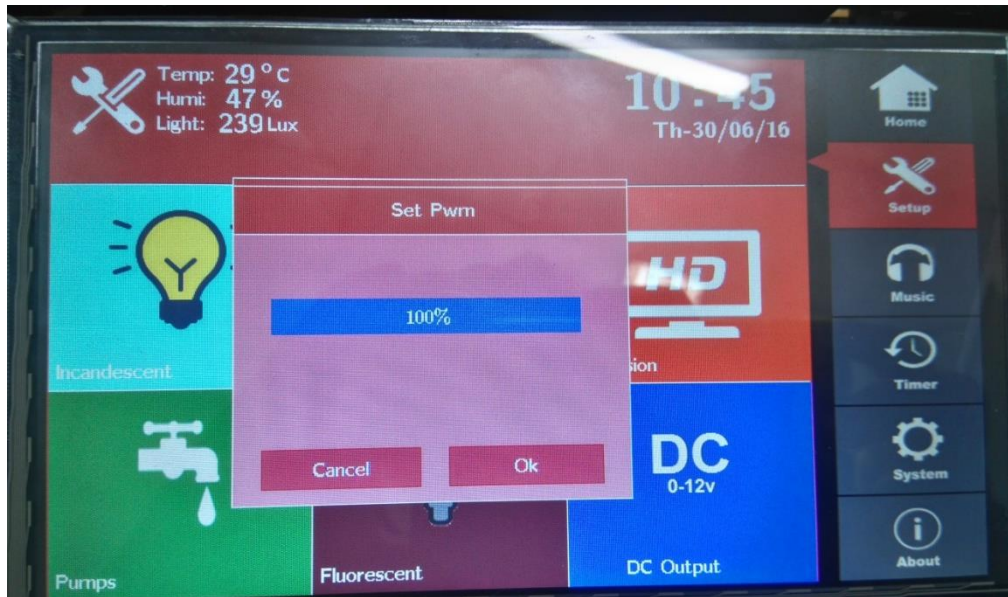
Tab Home là giao diện chính, hiển thị ngày giờ nhiệt độ, độ ẩm, cường độ sáng và trạng thái bật, tắt của 5 thiết bị AC và 1 thiết bị DC, trạng thái hẹn giờ. Trên giao diện này người dùng bật tắt các thiết bị thông qua việc chạm tay vào các nút tương ứng với mỗi nút.



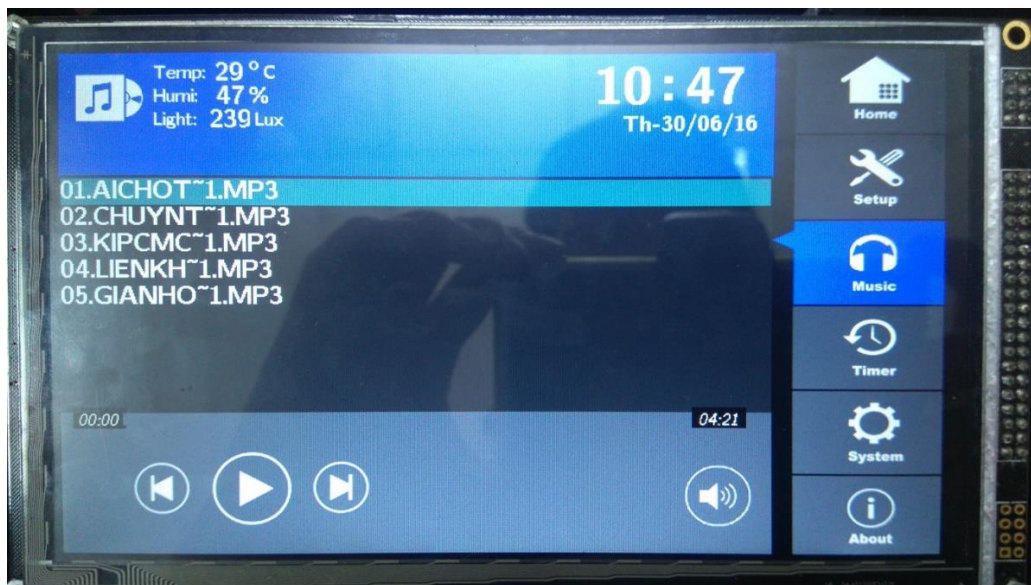
**Hình 5.2.** Giao diện tab Setup

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

Tab Setup cho phép người dùng tùy chọn điều khiển như: thêm, xóa một thiết bị, chỉnh độ sáng đèn sợi tóc, chỉnh tốc độ quạt, đặt tên cho thiết bị.



Hình 5.3 Điều chỉnh PWM



Hình 5.4. Giao diện tab Mucsic.

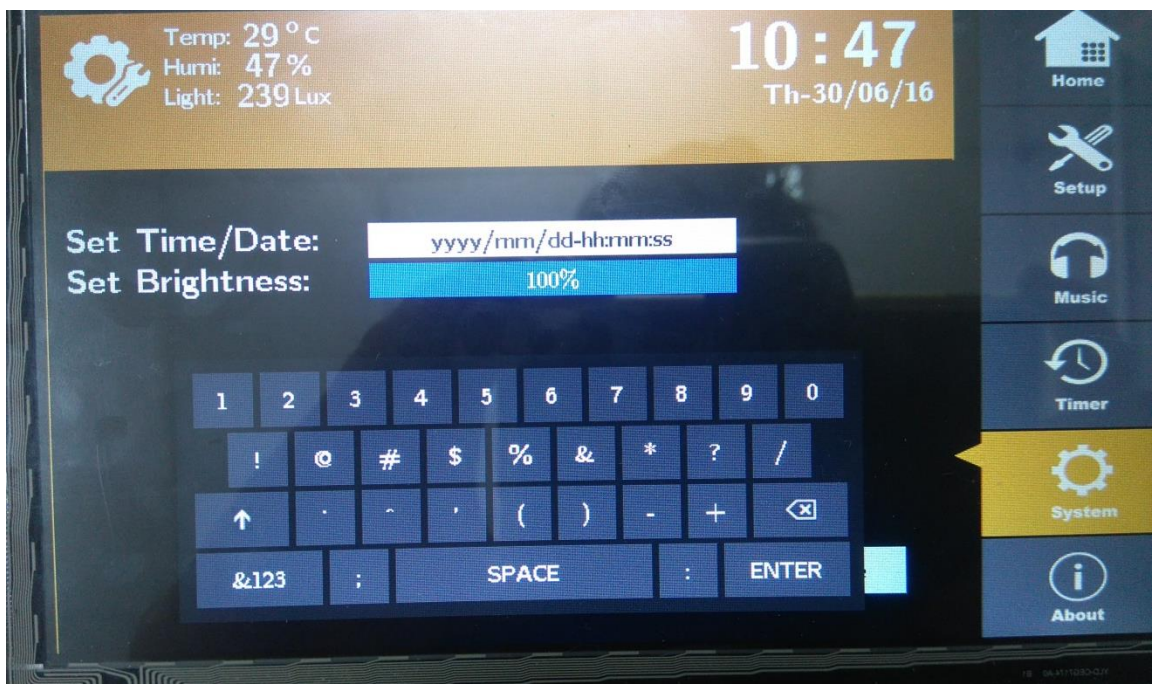
Tab Music hiển thị giao diện nghe nhạc, phát nhạc bằng cách chọn vào tên bài hát.





**Hình 5.5.** Giao diện tab Timer.

Tab Timer cho phép người dùng cài đặt thời gian bật, tắt thiết bị, có thể lưu thành một danh sách lịch trình bật, tắt.



**Hình 5.6.** Giao diện tab System.

Tab System cho phép người dùng điều chỉnh thời gian của hệ thống, chỉnh đèn nền màn hình và cho phép lưu trạng thái thiết bị hiện tại vào Flash.



**Hình 5.7.** Thiết bị được điều khiển ở chế độ tắt.



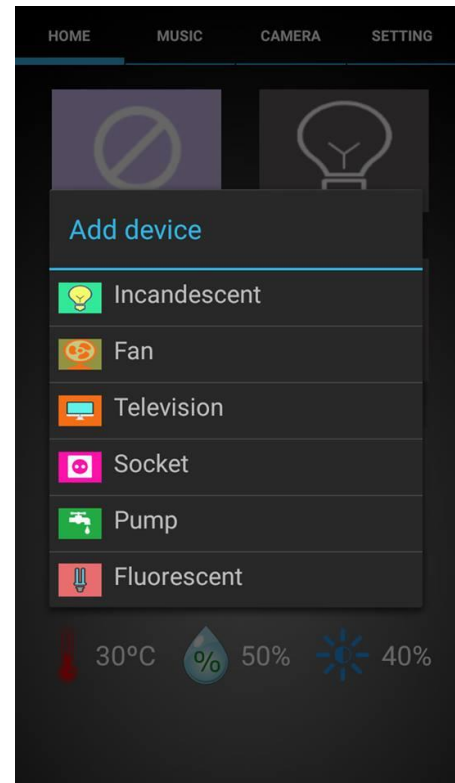
**Hình 5.8.** Thiết bị được điều khiển ở chế độ mở.

❖ Kết quả thi công ứng dụng Android.

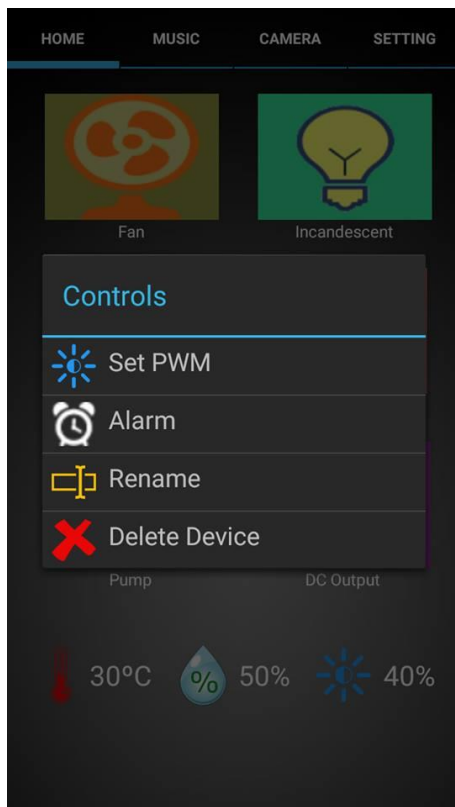
Để sử dụng ứng dụng trên android cần phải kết nối với wifi giao diện điện thoại sẽ đồng bộ với màn hình GLCD.



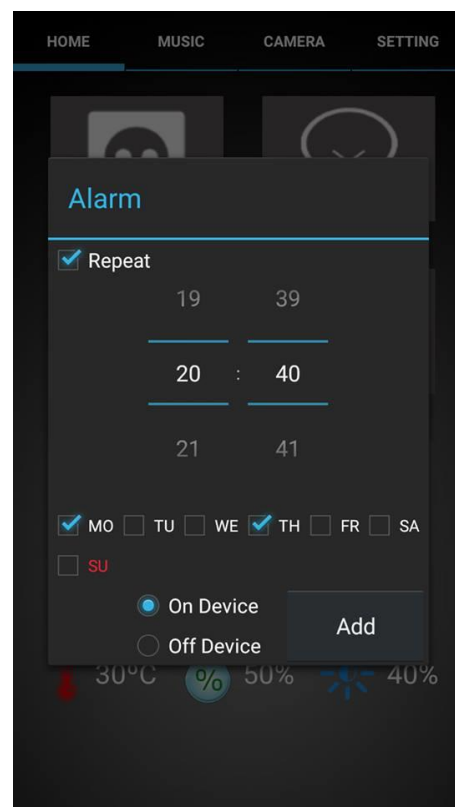
Hình 5.9. Giao diện chính của ứng dụng.



Hình 5.10. Thêm thiết bị.



Hình 5.11. Giao diện tùy chỉnh.



Hình 5.12. Giao diện hẹn giờ





**Hình 5.13.** Giao diện giám sát bằng camera IP.

### 5.2 NHẬN XÉT – ĐÁNH GIÁ.

Các nội dung đạt được so với yêu cầu đặt ra:

- Điều khiển bật tắt, điều chỉnh độ sáng hẹn giờ bật tắt thiết bị bằng cảm ứng của GLCD hoặc điện thoại thông qua wifi.
- Hiện thị nhiệt độ, thời gian và trạng thái của thiết bị trên màn hình GLCD và điện thoại.
- Phát nhạc MP3 từ thẻ nhớ.
- Giám sát từ xa bằng điện thoại.

Tuy nhiên hệ thống vẫn còn một số hạn chế như:

- Tính toán bố trí mạch công suất chưa hợp lý làm cho việc kết nối thiết bị gặp khó khăn.

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

### **6.1 KẾT LUẬN**

Chương kết quả đã phản ánh sinh viên chỉ thực hiện được khoảng 85% so với mục tiêu đặt ra ở chương 1. Các nội dung mà nhóm đã thực hiện được đó là thiết kế và thi công được một sản phẩm có thể điều khiển các thiết bị trong nhà, tạo được ứng dụng điều khiển và giao tiếp được giữa các khối với nhau. Tuy nhiên nhóm vẫn chưa thể tạo ra một hệ thống ổn định do phần tính toán thiết kế còn nhiều sai sót. Nhìn chung đề tài chỉ mới hoàn thành ở mức khá.

Trong quá trình làm đồ án, sinh viên đã rút ra được nhiều kinh nghiệm để tạo ra một sản phẩm hoàn thiện như: đầu tư thời gian, linh kiện trên thị trường, hiểu biết về các linh kiện và thiết kế board mạch, ...

### **6.2 HƯỚNG PHÁT TRIỂN**

Một số hướng phát triển để hoàn thiện đề tài như là:

- Kết nối internet, điều khiển kết hợp giữa Bluetooth, wifi và web server.
- Giao tiếp với USB theo chuẩn USB OTG để tăng tốc độ truyền dữ liệu.
- Thêm mạch công suất lớn để có thể điều khiển nhiều tải công suất lớn.

# TÀI LIỆU THAM KHẢO

## Sách tham khảo

- [1] ARM Việt Nam, *Cấu trúc cơ bản của STM32\_ARM Cortex M3*, 18/3/2010.
- [2] Tăng Quang Khải, *Báo cáo bài tập lớn lập trình nhúng cơ bản*, Trường ĐH Công Nghiệp Hà Nội, 2014.
- [3] Phan Văn Hoàn, *Giáo trình vi xử lý nâng cao*, Trường ĐH Sư Phạm Kỹ Thuật Tp.HCM, 2015.

## Website:

- [4] Các thành phần cơ bản của một ứng dụng android,  
<http://expressmagazine.net/development/1311/cac-thanh-phan-co-ban-cua-mot-ung-dung-android>
- [5] <http://www.arm.vn/TinChiTiet/tabid/105/id/148/Default.aspx>
- [6] <http://www.android.vn/>
- [7] <https://duythanhcse.wordpress.com/lap-trinh-di-dong/android/>

## PHỤ LỤC

[8] STM32F407xx Datasheet, [www.st.com/resource/en/datasheet/stm32f405rg.pdf](http://www.st.com/resource/en/datasheet/stm32f405rg.pdf)

[9] ESP8266 Datasheet, [www.electroschematics.com/11276/esp8266-datasheet](http://www.electroschematics.com/11276/esp8266-datasheet)

[10] VS1003B Datasheet, [www.vlsi.fi/fileadmin/datasheets/vs1003.pdf](http://www.vlsi.fi/fileadmin/datasheets/vs1003.pdf)

