**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH**

**---------------------------------**



**ĐỒ ÁN TỐT NGHIỆP**

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

**ĐỀ TÀI:**

**NGHIÊN CỨU, THIẾT KẾ VÀ THI CÔNG CỬA NHẬN DẠNG KHUÔN MẶT VÀ ĐIỀU KHIỂN, GIÁM SÁT CÁC THIẾT BỊ TRONG NHÀ**

**GVHD: ThS. NGUYỄN ĐÌNH PHÚ  
SVTH:**

**1. PHẠM QUỐC BẢO 15141101**

**2. VÕ VĂN THÀNH**

**Tp. Hồ Chí Minh – 06/2019**

**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH**

**---------------------------------**

**ĐỒ ÁN TỐT NGHIỆP**

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

**ĐỀ TÀI:**

**NGHIÊN CỨU, THIẾT KẾ VÀ THI CÔNG CỬA NHẬN DẠNG KHUÔN MẶT VÀ ĐIỀU KHIỂN, GIÁM SÁT THIẾT BỊ**

**TRONG NHÀ**

**GVHD: ThS. NGUYỄN ĐÌNH PHÚ  
SVTH:**

**1. PHẠM QUỐC BẢO 15141101**

**2. VÕ VĂN THÀNH**

**Tp. Hồ Chí Minh – 06/2019**

|  |  |
| --- | --- |
| TRƯỜNG ĐH SPKT TP. HỒ CHÍ MINH  **KHOA ĐIỆN-ĐIỆN TỬ**  **BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP - TỰ DO - HẠNH PHÚC**  **----o0o----** |

Tp. HCM, ngày tháng năm 2019

# **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

|  |  |  |
| --- | --- | --- |
| Họ tên sinh viên: | Phạm Quốc Bảo  Võ Văn Thành | MSSV: 15141101  MSSV: 1514 |
| Chuyên ngành: | Kỹ thuật điện tử - truyền thông | Mã ngành: 141 |
| Hệ đào tạo: | Đại học chính quy | Mã hệ: 1 |
| Khóa: | 2015 | Lớp: 15141DT2A |

I. TÊN ĐỀ TÀI: **NGHIÊN CỨU, THIẾT KẾ VÀ THI CÔNG CỬA NHẬN DẠNG KHUÔN MẶT VÀ ĐIỀU KHIỂN, GIÁM SÁT THIẾT BỊ TRONG NHÀ**

II. NHIỆM VỤ

1. Các số liệu ban đầu:

(*ghi những thông số, tập tài liệu tín hiệu, hình ảnh…*)

2. Nội dung thực hiện:

(*ghi những nội dung chính* *cần thực hiện như trong phần tổng quan*)

III. NGÀY GIAO NHIỆM VỤ:

IV. NGÀY HOÀN THÀNH NHIỆM VỤ:

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

|  |  |
| --- | --- |
| TRƯỜNG ĐH SPKT TP. HỒ CHÍ MINH  **KHOA ĐIỆN-ĐIỆN TỬ**  **BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP - TỰ DO - HẠNH PHÚC**  **----o0o----** |

Tp. HCM, ngày tháng năm 2019

# **LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên 1: Phạm Quốc Bảo

Lớp: 15141DT2A MSSV: 15141101

Họ tên sinh viên 2: Võ Văn Thành

Lớp: 15141DT2A MSSV: 1514

Tên đề tài:

**NGHIÊN CỨU, THIẾT KẾ VÀ THI CÔNG CỬA NHẬN DẠNG KHUÔN MẶT VÀ ĐIỀU KHIỂN, GIÁM SÁT THIẾT BỊ TRONG NHÀ**

|  |  |  |
| --- | --- | --- |
| **Tuần/ngày** | **Nội dung** | **Xác nhận GVHD** |
| Tuần 1  (18/2 – 24/2) | - Gặp giáo viên hướng dẫn  - Chọn đề tài. |  |
| Tuần 2  (25/2 – 3/3) | - Báo cáo GVHD.  - Viết đề cương chi tiết cho đồ án. |  |
| Tuần 3  (4/3 – 10/3) | - Báo cáo GVHD.  - Trình bày phướng án thực hiện đề tài.  - Phân chia công việc cho từng thành viên. |  |
| Tuần 4  (11/3 – 17/3) | - Báo cáo GVHD.  - Tìm hiểu Raspberry Pi 3b và xửa lý ảnh |  |
| Tuần 5  (18/3 – 24/3) | - Báo cáo GVHD.  - Thiết kế sơ đồ nguyên lý toàn mạch |  |
| Tuần 6  (25/3 – 31/3) | - Lập danh sách, mua những linh kiện cần thiết.  - Tìm hiểu giao tiếp ngoại vi với STM32F407 |  |
| Tuần 7  (1/4 - 7/4) | - Báo cáo GVHD.  - Kết nối khối ngõ ra, hiển thị thông tin đo đạc ra màn hình. |  |
| Tuần 8  (8/4 – 14/4) | - Báo cáo GVHD.  - Tìm hiểu giao thức HTTP và thiết kế webserver |  |
| Tuần 9  (15/4 – 21/4) | - Báo cáo GVHD.  - Tiến hành thiết kế giao diện web và app. |  |
| Tuần 10  (22/4 – 28/4) | - Báo cáo GVHD.  - Viết chương trình cho toàn bộ hệ thống |  |
| Tuần 11  (29/4 – 5/5) | - Báo cáo GVHD.  - Vẽ mạch PCB, hàn mạch. |  |
| Tuần 12  (6/5 – 12/5) | - Báo cáo GVHD.  - Thi công hộp đựng |  |
| Tuần 13  (13/5 – 19/5) | - Báo cáo GVHD.  **-** Lắp mạch và thiết bị liên quan vào hộp.  - Chạy thử nghiệm |  |
| Tuần 14  (20/5 – 26/5) | - Báo cáo GVHD.  - Cân chỉnh, tối ưu sản phẩm. |  |
| Tuần 15  (27/5 – 2/6) | - Báo cáo GVHD.  - Chỉnh sửa và hoàn thiện báo cáo |  |
| Tuần 16  (18/6 – 22/6) | - Bảo vệ ĐATN |  |

**GV HƯỚNG DẪN**

**(Ký và ghi rõ họ và tên)**

# **LỜI CAM ĐOAN**

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

**Người thực hiện đề tài**

**Phạm Quốc Bảo Võ Văn Thành**

# **LỜI CẢM ƠN**

Lời đầu tiên, nhóm đề tài xin gởi lời cảm ơn chân thành đến Thầy Nguyễn Đình Phú - giảng viên khoa Điện-Điện tử, Trường ĐH Sư phạm Kỹ thuật Tp. HCM, đã theo sát hỗ trợ và hướng dẫn nhóm một cách chi tiết trong quá trình thực hiện đề tài. Thầy luôn hỗ trợ hết mình, giải đáp thắc mắc, chỉ ra sai sót cũng như gợi ý những phương án thực hiện sao cho khả thi và dễ tiếp cận nhất.

Nhóm cùng xin gửi lời cảm ơn đến quý thầy(cô) trong khoa Điện-Điện tử đã tận tình giải đáp thắc mắc, nguyện vọng trong quá trình thực hiện đề tài của nhóm. Sự hỗ trợ của quý thầy(cô) đóng góp một phần không nhỏ vào thành công ngày hôm nay.

Nhóm đề tài cũng xin cảm ơn các bạn sinh viên trong khoa Điện-Điện tử đã nhiệt tình giúp đỡ, hỗ trợ nhóm, đặc biệt là tập thể lớp 1514DT2A. Những đóng góp của các bạn luôn được nhóm tiếp nhận và xem xét kĩ lưỡng.

Cuối cùng, nhóm xin cảm ơn chân thành và sâu sắc đến Cha, Mẹ - những người luôn bên cạnh hỗ trợ hết mình về tài chính cũng như tinh thần trong suốt những năm tháng qua.

Thành công của đề tài ngày hôn nay chính là một phần đóng góp to lớn của mọi người. Một lần nữa, nhóm xin gởi lời cảm ơn chân thành nhất đến quý thầy cô, bạn bè và quý phụ huynh đã hỗ trợ nhóm thực hiện đề tài hoàn chỉnh.

Xin chân thành cảm ơn!

Người thực hiện đề tài

**Phạm Quốc Bảo Võ Văn Thành**

# **MỤC LỤC**

**[NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP](#_Toc10367551)** [i](#_Toc10367551)

[**LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP** ii](#_Toc10367552)

[**LỜI CAM ĐOAN** iv](#_Toc10367553)

[**LỜI CẢM ƠN** v](#_Toc10367554)

[**MỤC LỤC** vi](#_Toc10367555)

[**DANH MỤC HÌNH** viii](#_Toc10367556)

[**TÓM TẮT** x](#_Toc10367557)

[**CHƯƠNG 1: TỔNG QUAN** 1](#_Toc10367558)

[**1.1 ĐẶT VẤN ĐỀ** 1](#_Toc10367559)

[**1.2 MỤC TIÊU** 2](#_Toc10367560)

[**1.3 NỘI DUNG NGHIÊN CỨU** 2](#_Toc10367561)

[**1.4 GIỚI HẠN** 2](#_Toc10367562)

[**1.5 BỐ CỤC** 3](#_Toc10367563)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 4](#_Toc10367564)

[**2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH** 4](#_Toc10367565)

[**2.1.1 Giới thiệu xử lý ảnh** 4](#_Toc10367566)

[**2.1.2 Những vấn đề xử lý ảnh** 5](#_Toc10367567)

[**2.1.3 Ngôn ngữ Python và thư viện OpenCV** 9](#_Toc10367568)

[**2.2 Phương pháp Nhận dạng khuôn mặt** 10](#_Toc10367569)

[**2.3 Phương pháp HOG (Histogram of Oriented Gradient)** 12](#_Toc10367570)

[**2.3.1 Tiền xử lý** 13](#_Toc10367571)

[**2.3.2 Tính Gradient** 13](#_Toc10367572)

[**2.3.3 Tính vector đặc trưng cho từng ô** 14](#_Toc10367573)

[**2.3.4 Chuẩn hóa khối (blocks)** 18](#_Toc10367574)

[**2.3.5 Tính toán vector đặc trưng HOG** 22](#_Toc10367575)

[**2.4 Thư viện Dlib** 23](#_Toc10367576)

[**CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG** 24](#_Toc10367577)

[**3.1. THIẾT KẾ SƠ ĐỒ KHỐI** 24](#_Toc10367578)

[**3.2. RASPBERRY PI 3 MODEL B** 25](#_Toc10367579)

[**3.2.1 Giới thiệu Raspberry Pi 3** 25](#_Toc10367580)

[**3.2.2 Nội dung thêm vào** 27](#_Toc10367581)

[**3.3 STM32F407VG** 28](#_Toc10367582)

[**3.3.1 Bộ điều khiển reset** 29](#_Toc10367583)

[**3.3.2 Xung clock** 30](#_Toc10367584)

[**3.3.3 Gpio** 32](#_Toc10367585)

[**3.4 WEBCAM LOGITECH** 36](#_Toc10367586)

[**3.5 KHOÁ CHỐT ĐIỆN** 37](#_Toc10367587)

[**3.6 NODEMCU ESP8266** 38](#_Toc10367588)

[**3.6.1 Giới thiệu chung** 38](#_Toc10367589)

[**3.9 SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH** 39](#_Toc10367590)

[**CHƯƠNG 4: THI CÔNG HỆ THỐNG** 40](#_Toc10367591)

[**4.1 GIỚI THIỆU** 40](#_Toc10367592)

[**4.2 THI CÔNG HỆ THỐNG** 40](#_Toc10367593)

[**4.2.1 Thi công board mạch** 40](#_Toc10367594)

[**4.2.2 Lắp ráp và kiểm tra** 41](#_Toc10367595)

[**4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH** 41](#_Toc10367596)

[**4.4 LẬP TRÌNH HỆ THỐNG** 41](#_Toc10367597)

[**4.4.1 Lưu đồ giải thuật** 41](#_Toc10367598)

[**4.4.2 Giới thiệu hệ điều hành raspbian** 42](#_Toc10367599)

[**4.4.3 Giới thiệu phần mềm android** 42](#_Toc10367600)

[**4.3.4 Trình biên dịch Keil µVision 5** 48](#_Toc10367601)

[**4.3.5 Phần mềm viết giao diện màn hình nextion editor** 56](#_Toc10367602)

[**CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT – ĐÁNH GIÁ** 58](#_Toc10367603)

[**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 59](#_Toc10367604)

[**TÀI LIỆU THAM KHẢO** 60](#_Toc10367605)

# **DANH MỤC HÌNH**

[***Hình 2-1: Tổng quan quá trình xử lí ảnh 4***](#_Toc10293705)

[***Hình 2-2: Các bước xử lý ảnh 4***](#_Toc10293706)

[***Hình 2-3: Các lân cận ảnh 6***](#_Toc10293707)

[***Hình 2-4: Các trường hợp biên 8***](#_Toc10293708)

[***Hình 2-5: Tách biên là một trong những công đoạn trích chọn đặc trưng trong quá trình xử lí ảnh 9***](#_Toc10293709)

[***Hình 2-6: Ngôn ngữ lập trình Python 9***](#_Toc10293710)

[***Hình 2-7: Các bước tiến hành để xử lí ảnh 12***](#_Toc10293711)

[***Hình 2-8: Điều chỉnh kích thước ảnh cần xử lí 13***](#_Toc10293712)

[***Hình 2-9: Ví dụ điểm ảnh cần xử lí 14***](#_Toc10293713)

[***Hình 2-10: Giá trị tuyệt đối của x-gradient, y-gradient và cường độ gradient 14***](#_Toc10293714)

[***Hình 2-11: Chia hình ảnh thành các block 16***](#_Toc10293715)

[***Hình 2-12: Xây dựng cường độ biểu đồ gradient 17***](#_Toc10293716)

[***Hình 2-13: Histogram của gradient 18***](#_Toc10293717)

[***Hình 2-14: Ảnh hưởng của việc chuẩn hóa tới các vector gradient 19***](#_Toc10293718)

[***Hình 2-15: Các giá trị pixel thực tế và sự thay đổi của vector gradient 20***](#_Toc10293719)

[***Hình 2-16: Chia hình ảnh thành các block để tính toán vector đặc trưng HOG 22***](#_Toc10293720)

[***Hình 2-17: Thư viện Dlib 23***](#_Toc10293721)

[***Hình 3-1: Sơ đồ khối toàn bộ hệ thống 24***](#_Toc10293722)

[***Hình 3-2: Raspberry Pi 3B 26***](#_Toc10293723)

[***Hình 3-3: Sơ đồ chân của Raspberry Pi 3B 27***](#_Toc10293724)

[***Hình 3-4: Kit STM32F407VG 27***](#_Toc10293725)

[***Hình 3-5: Sơ đồ cây xung Clock 31***](#_Toc10293726)

[***Hình 3-6: Cấu trúc cơ bản của một chân I/O 32***](#_Toc10293727)

[***Hình 3-7: Thanh ghi GPIOx\_MODER 33***](#_Toc10293728)

[***Hình 3-8: Thanh ghi GPIOx\_OTYPER 33***](#_Toc10293729)

[***Hình 3-9: Thanh ghi GPIOx\_ OSPEEDR 34***](#_Toc10293730)

[***Hình 3-10: Thanh ghi GPIOx\_PUPDR 34***](#_Toc10293731)

[***Hình 3-11: Thanh ghi GPIOx\_IDR 35***](#_Toc10293732)

[***Hình 3-12: Thanh ghi GPIOx\_ODR 35***](#_Toc10293733)

[***Hình 3-13: Thanh ghi GPIOx\_BSRR 36***](#_Toc10293734)

[***Hình 3-14: Webcam Logitech C270 36***](#_Toc10293735)

[***Hình 3-15: Khoá chốt điện 37***](#_Toc10293736)

[***Hình 3-16: NodeMCU V1 39***](#_Toc10293737)

[***Hình 3-17: Sơ đồ chân NodeMCU V1 41***](#_Toc10293739)

[***Hình 3-18: Sơ đồ mạch nguyên lí của NodeMCU V1 43***](#_Toc10293741)

[***Hình 3-19: Phần mềm Android Studio 44***](#_Toc10293742)

[***Hình 3-20: Cửa sổ làm việc của Android Studio 45***](#_Toc10293743)

[***Hình 3-21: Giao diện ban đầu của Android Studio 47***](#_Toc10293744)

[***Hình 3-22: Chọn màn hình cho ứng dụng 48***](#_Toc10293745)

[***Hình 3-23: Thiết lập các thông tin ban đầu cho project 49***](#_Toc10293746)

[***Hình 3-24: Vị trí 2 file activity\_main.xml và MainActivity.java 50***](file:///C:\Users\Admin\Desktop\LUAN_VAN_DEMO.docx#_Toc10293747)

[***Hình 3-25: Giao diện người dùng 50***](file:///C:\Users\Admin\Desktop\LUAN_VAN_DEMO.docx#_Toc10293748)

# **TÓM TẮT**

Đề tài “Nghiên cứu, thiết kế và thi công hệ thống cửa nhận dạng khuôn mặt và điều khiển, giám sát thiết bị trong nhà” là mô hình phân loại sản phẩm, cụ thể ở đây được ứng dụng trong các hệ thống nhà thông minh. Hoạt động của mô hình dựa trên cơ chế xử lý tín hiệu tương tự (màu sắc, cân nặng) của kit Raspberry Pi Model 3b+, với ngôn ngữ lập trình Python và thư viện OpenCV, sau đó chuyển tín hiệu về cho trung tâm xử lý để được phép điều khiển thiết bị trong gia đình. Nhóm sử dụng những đặc điểm riêng về khuôn mặt của mỗi người để cho raspberry pi3b+ học được hình ảnh của chủ nhà. Kết quả thực hiện đề tài là một mô hình nhà thông minh với tiêu chí an toàn cho chủ nhà được ưu tiên đặt lên hàng đầu. Việc quản lí hoạt động của hệ thống, bao gồm theo dõi người ra vào nhà, quản lý, giám sát thiết bị trong gia đình…sẽ được thực hiện từ xa thông qua điện thoại thông minh hoặc thông qua webserver.

# **CHƯƠNG 1: TỔNG QUAN**

## **1.1 ĐẶT VẤN ĐỀ**

Trong các ngôi nhà hiện đại ngày nay, số lượng trang thiết bị điện - điện tử đang không ngừng gia tăng. Tuy nhiên, do khác nhau về kiến trúc, việc điều khiển các thiết bị đôi khi bất cập. Thêm vào đó, điều khiển các thiết bị một cách thủ công với khoảng cách xa cũng gây nhiều trở ngại. Vì vậy, việc áp dụng các công nghệ điều khiển từ xa nhằm giải quyết tương tác giữa môi trường và các thiết bị trong nhà một cách linh hoạt, dễ dàng là điều tất yếu và cần thiết.

Xã hội ngày càng hiện đại, nhu cầu về sự tiện nghi của con người ngày càng được nâng cao. Một căn nhà không còn đơn thuần là nơi che mưa che nắng nữa, mà ở đó chúng ta có thể điều khiển, theo dõi và giám sát các thiết bị trong nhà một cách tự động và tiện lợi chỉ với một vài thao tác. Ở đó, tất cả các thiết bị điện tử gia dụng đều được liên kết với thiết bị điều khiển trung tâm và có thế phối hợp với nhau để cùng thực hiện một chức năng.

Nhiều công nghệ đã được áp dụng khi xây dựng nhà thông minh, trong đó công nghệ Internet of Things (IoT) đang trở lên thông dụng và phổ biến, biến những món đồ điện tử bình thường trong ngôi nhà trở nên thân thiện và gần gũi với người dùng hơn.

Ngoài ra hệ thống mà nhóm đang phát triển còn giúp người dùng giám sát người ra vào nhà nên cũng tích hợp thêm xử lý ảnh để giúp kiểm soát được người vào ra ngôi nhà từ đó xử lý được những tình huống không mong muốn khi bạn đi vắng.

Với mục đích tạo ra một sản phầm giúp mọi người dễ dàng điều khiển, quản lý thiết bị trong gia đình một cách dễ dàng, cảnh báo sớm cho người dùng cùng với khả năng tùy biến cao, giá cả phù hợp, nhóm xin giới thiệu mô hình: “*Khóa cửa nhận dạng khuôn mặt và điều khiển, giám sát thiết bị trong nhà”* giúp giám sát ngôi nhà của bạn mọi lúc, mọi nơi chỉ cần có internet.

## **1.2 MỤC TIÊU**

Mục tiêu nghiên cứu của đề tài “Nghiên cứu, thiết kế và thi công mô hình cửa nhận dạng khuôn mặt và điều khiển, giám sát thiết bị trong nhà” sử dụng Kit Raspberry cùng với ngôn ngữ lập trình Python và thư viện OpenCV xử lý nhận dạng được chủ nhà thông qua đó có thể điều khiển, giám sát thiết bị trong nhà.

## **1.3 NỘI DUNG NGHIÊN CỨU**

Để thực hiện đề tài: ***“Nghiên cứu thiết kế thi công cửa nhận dạng khuôn mặt và điều khiển, giám sát thiết bị trong gia đình”***, cần nghiên cứu những nội dung sau đây:

* NỘI DUNG 1: Tìm hiểu cách thức hoạt động của các module liên quan.
* NỘI DUNG 2: Nghiên cứu thuật toán xử lý ảnh ứng dụng phân loại khuôn mặt.
* NỘI DUNG 3: Giải pháp thiết kế, xử lý trên mô hình thực tế.
* NỘI DUNG 4: Giải pháp thiết kế giao diện điện thoại, server giám sát hoạt động của hệ thống.
* NỘI DUNG 5: Hoàn thiện hệ thống điều khiển và mô hình, tiến hành chạy thử nghiệm.
* NỘI DUNG 6: Đánh giá kết quả thực hiện.

## **1.4 GIỚI HẠN**

* Thiết kế mô hình cửa ra vào và các thiết bị trong gia đình.
* Dùng Raspberry Pi 3 B để xử lý hình ảnh và gửi dữ liệu lên môi trường web trung gian.
* Giám sát hoạt động của hệ thống qua màn hình cảm ứng, ứng dụng điều khiển trên điện thoại Android thông qua môi trường web trung gian.
* Ứng dụng điều khiển hệ thống trên điện thoại Android bằng app.
* Sử dụng vi điều khiển Arm (32bit) để điều khiển các thiết bị trong nhà

## **1.5 BỐ CỤC**

* **Chương 1:** Tổng quan
* **Chương 2:** Cơ sở lý thuyết
* **Chương 3:** Tính toán và thiết kế hệ thống
* **Chương 4:** Thi công hệ thống
* **Chương 5:** Kết quả, đánh giá
* **Chương 6:** Kết luận và hướng phát triển
* **Chương 1: Tổng quan.**

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

* **Chương 2: Cơ sở lý thuyết.**

Nội dung chương sẽ bao gồm lý thuyết cơ bản về xử lý ảnh, phương pháp cơ bản để nhận dạng và phân loại ảnh, giới thiệu cơ bản về Raspberry và thư viện liên quan.

* **Chương 3: Thiết kế và thi công**

Chương này sẽ đi tìm hiểu sơ đồ và cách kết nối cho hệ thống.

* **Chương 4: Thi công hệ thống**

Nội dụng chương này là quá trình thực thi chương trình nhận dạng khuôn mặt, điều khiển, giám sát thiết bị trong nhà

* **Chương 5: Nhận xét và đánh giá**

Nội dụng chương này là tổng hợp các kết quả đạt được sau khi thi công mô hình cùng với nhận xét và đánh giá quá trình thực hiện.

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

Nội dụng chương này là tổng hợp các kết quả đạt được sau khi thi công mô hình cùng với nhận xét và đánh giá quá trình thực hiện.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH**

### **2.1.1 Giới thiệu xử lý ảnh**

Xử lý ảnh là quá trình xử lý số tín hiệu với đối tượng xử lý là tín hiệu hình ảnh. Trong đó, ảnh đầu vào sẽ được xử lý sao cho ảnh sau khi xử lý đạt kết quả mong muốn. Kết quả của Xử lý ảnh là một ảnh khác có đặc điểm khác với ảnh ban đầu hoặc một kết luận (hình 2.1). Xử lý ảnh phát triển rất mạnh mẽ trong thời gian gần đây, bao gồm ở rất nhiều lĩnh vực: kinh thế, văn hóa, y tế, quân sự… Sự bùng nổ khoa học kỹ thuật trong những năm gần đây, tiêu biểu là cuộc Cách mạng Công nghiệp 4.0 đã tạo cho con người thêm nhiều công cụ và tiện ích, phục vụ mọi mặt đời sống của họ. Hiện nay, có bốn lĩnh vực chính liên quan đến sử lý ảnh: xử lý và nâng cao chất lượng ảnh, nhận dạng ảnh, truy vấn ảnh và nén ảnh. Ở phạm vi đề tài này sẽ tìm hiểu về lĩnh vực nhận dạng ảnh.



Hình 2-1: Tổng quan quá trình xử lí ảnh

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P (c1, c2..,cn). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

A close up of a logo

Description automatically generated

Hình 2-2: Các bước xử lý ảnh

### **2.1.2 Những vấn đề xử lý ảnh**

**a. Điểm ảnh**

Ảnh số (digital image) là một tập hợp của nhiều điểm ảnh, hay còn gọi là pixel. Mỗi điểm ảnh biểu diễn một màu sắc nhất định (hay độ sáng với ảnh đen trắng) tại một điểm duy nhất, có thể xem một điểm ảnh giống như một chấm nhỏ trong một tấm ảnh màu.

Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặt điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn.

**b. Mức xám của ảnh**

Mức xám của ảnh (grayscale) là một trong những giá trị số của điểm ảnh biểu diễn mức độ ánh sáng (light intensity) tại điểm ảnh đấy. Thông thường, trong xử lý ảnh hiện tại, mức xám hay sử dụng nhất là mức 256 (có giá trị mức xám từ 0 -> 255).

**c. Phân loại ảnh**

Có hai dạng quan trọng trong ảnh số được dùng với nhiều mục đích khác nhau là ảnh màu và ảnh đen trắng (hay còn gọi là ảnh xám). Trong đó, ảnh màu được cấu trúc từ các pixel màu trong khi ảnh đen trắng được xây dựng từ các pixel có giá trị mức xám khác nhau.

*Ảnh đen trắng:* với một ảnh đen trắng được xây dựng từ nhiều pixel mà tại đó biểu diễn một giá trị nhất định tương ứng với một mức xám.

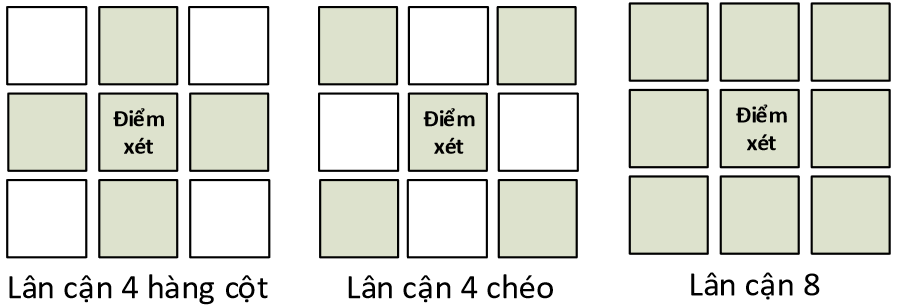
*Ảnh màu:* Một ảnh màu thường được tạo thành từ nhiều pixel mà trong đó mỗi pixel được biểu diễn bởi ba giá trị tương ứng với các mức trong các kênh màu đỏ (Red), xanh lá (Green) và xanh dương (Blue) tại một vị trí cụ thể. Các kênh màu Red, Green và Blue (trong không gian màu RGB) là những màu cơ bản mà từ đó có thể tạo ra các màu khác nhau bằng phương pháp pha trộn. Cùng một ảnh có kích thước nhất định, dung lượng của ảnh màu lưu trên bộ nhớ luôn lớn hơn gấp ba lần dung lượng dành cho ảnh đen trắng trong trường hợp không sử dụng các kỹ thuật nén ảnh.

*Ảnh nhị phân:* chỉ sử dụng duy nhất một bit để biểu diễn một pixel. Do một bit chỉ có thể xác lập hai trạng thái là đóng và mở hay 1 và 0 tương ứng với hai màu đen và trắng.

*Ảnh chỉ số:* Một vài ảnh màu (hay đen trắng) được tạo thành từ một bảng màu có sẵn bị giới hạn, điển hình thường dùng là tập 256 màu khác nhau. Những ảnh này được gọi là ảnh màu chỉ số hóa do dữ liệu dành cho mỗi pixel bao gồm chỉ số có sẵn chỉ rõ màu trong tập có sẵn ứng với pixel đang xem xét.

**d. Quan hệ giữa các điểm ảnh**

Lân cận ảnh là tập hợp những điểm ảnh có xung quanh điểm ảnh đang xét. Có 2 loại lân cận ảnh: lân cận 4 và lân cận 8.



Hình 2-3: Các lân cận ảnh

4 điểm ảnh lân cận 4 theo cột và hàng với tọa độ lần lượt là (x+1, y), (x-1, y), (x, y+1), (x, y-1) ký hiệu là tập N4 (p). 4 điểm ảnh lân cận 4 theo đường chéo có tọa độ lần lượt là (x+1, y+1), (x+1, y+1), (x-1, y+1), (x-1, y-1) ký hiệu là tập ND (p). Tập 8 điểm ảnh lân cận 8 là hợp của 2 tập trên:

N8 (p) = N4 (p) + ND (p)

Liên kết ảnh: Các mối liên kết của ảnh được xem như là mối liên kiết của 2 điểm ảnh gần nhau, có 3 loại liên kết: liên kết 4, liên kết 8, lên kết m (liên kết hỗn hợp). Trong ảnh đa mức xám, ta có thể đặt V chứa nhiều giá trị như V= {tập con}. Cho p có tọa độ (x, y).

Liên kết 4: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 4 của nhau nếu q thuộc về tập N4 (p).

Liên kết 8: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 8 của nhau nếu q thuộc về tập N8 (p).

Liên kết m: hai điểm ảnh p và q có giá trị thuộc tập V được gọi là Liên kết M của nhau nếu thỏa 1 trong 2 điều kiện: q thuộc về tập N4 (p), q thuộc về tập ND (p) và giao của hai tập N4 (p), N4 (q) không chứa điểm ảnh nào có giá trị thuộc V.

**e. Lọc nhiễu**

Ảnh thu nhận được thường sẽ bị nhiễu nên cần phải loại bỏ nhiễu. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân nhóm theo công dụng: làm trơn nhiễu, nổi biên. Để làm trơn nhiễu hay tách nhiễu, người ta sử dụng các bộ lọc tuyến tính (lọc trung bình, thông thấp) hoặc lọc phi tuyến (trung vị, giả trung vị, lọc đồng hình). Từ bản chất của nhiễu (thường tương ứng với tần số cao) và từ cơ sở lý thuyết lọc là: bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua, để lọc nhiễu người ta thường dùng lọc thông thấp (theo quan điểm tần số không gian) hay lấy tổ hợp tuyến tính để san bằng (lọc trung bình). Để làm nổi cạnh (ứng với tần số cao), người ta dùng các bộ lọc thông cao, lọc Laplace. Phương pháp lọc nhiễu. Chia làm 2 loại: lọc tuyến tính, lọc phi tuyến.

**Làm trơn nhiễu bằng lọc tuyến tính**: Khi chụp ảnh có thể xuất hiện nhiều loại nhiễu vào quá trình xử lý ảnh, nên ta cần phải lọc nhiễu. Gồm các phương pháp cơ bản lọc trung bình, lọc thông thấp, …Ví dụ lọc trung bình: Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận.

**Làm trơn nhiễu bằng lọc phi tuyến**[2]: Các bộ lọc phi tuyến cũng hay được dùng trong kỹ thuật tăng cường ảnh. Một số phương pháp lọc cơ bản bộ lọc trung vị, lọc ngoài… Với lọc trung vị, điểm ảnh đầu vào sẽ được thay thế bởi trung vị các điểm ảnh còn lọc giả trung vị sẽ dùng trung bình cộng của hai giá trị “trung vị” (trung bình cộng của max và min).

Lọc trung vị: Kỹ thuật này đòi hỏi giá trị các điểm ảnh trong cửa sổ phải xếp theo thứ tự tăng hay giảm dần so với giá trị trung vị. Kích thước cửa sổ thường được chọn sao cho số điểm ảnh trong cửa sổ là lẻ.

Lọc ngoài: Giả thiết có ngưỡng nào đó cho các mức nhiễu (có thể dựa vào lược đồ xám). Tiến hành so sánh giá trị độ xám của một điểm ảnh với trung bình số học 8 lân cận của nó. Nếu sai lệch lớn hơn ngưỡng, điểm ảnh này được coi như nhiễu. Trong trường hợp đó, thay thế giá trị của điểm ảnh bằng giá trị trung bình 8 lân cận vừa tính được.

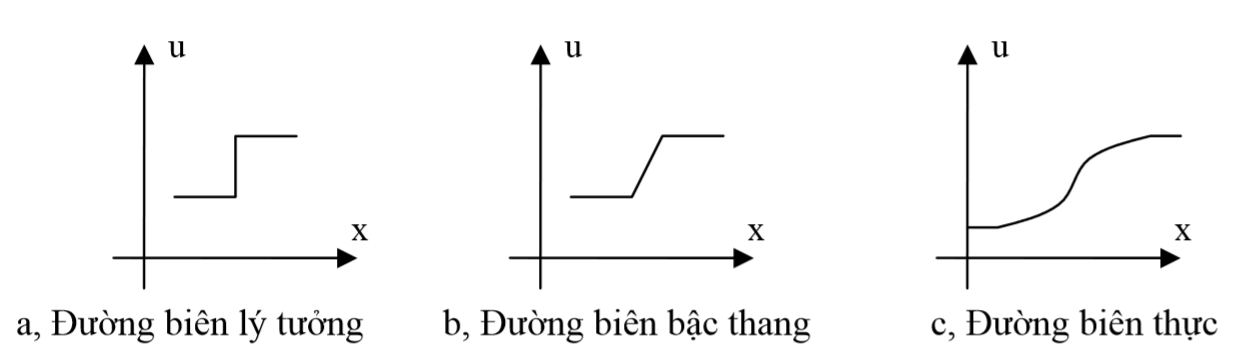
**f. Phương pháp phát hiện biên**

*Điểm biên*: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.

*Đường biên (đường bao: boundary):* tập hợp các điểm biên liên tiếp tạo thành một đường biên hay đường bao.

*Ý nghĩa của đường biên trong xử lý:* ý nghĩa đầu tiên: đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.

*Tầm quan trọng của biên*: để thấy rõ tầm quan trọng của biên, xét ví dụ sau: khi người họa sĩ muốn vẽ một danh nhân, họa sĩ chỉ cần vẽ vài đường nứt tốc họa mà không cần vẽ một cách đầy đủ. Mô hình biểu diễn đường biên, theo toán học: điểm ảnh có sự biến đổi mức xám u(x) một cách đột ngột theo hình dưới.



Hình 2-4: Các trường hợp biên

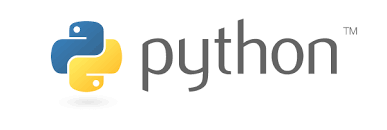
**Chú ý:** Phát hiện biên là một phần trong phân tích ảnh, sau khi đã lọc ảnh (hay tiền xử lý ảnh). Các bước của phân tích ảnh có thể mô tả theo sơ đồ dưới đây. Việc dò và tìm biên ảnh là một trong các đặc trưng thuộc khối trích chọn đặc trưng.



Hình 2-5: Tách biên là một trong những công đoạn trích chọn đặc trưng trong quá trình xử lí ảnh

### **2.1.3 Ngôn ngữ Python và thư viện OpenCV**

**a. Ngôn ngữ Python**

****

Hình 2-6: Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng và là ngôn ngữ bậc cao ngữ nghĩa động.Python hỗ trọ các module và các gói package, khuyến khích chương trình module hoá và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Những đặc điểm của ngôn ngữ Python:

* Ngữ pháp đơn giản, dễ .
* Vừa hướng đối tượng, vừa hướng thủ tục.
* Hỗ trợ module và hỗ trợ gói.
* Xử lý lỗi bằng ngoại lệ.
* Kiểu dữ liệu động ở mức cao.
* Các bộ thư viện chuẩn và các module ngoài đáp ứng được tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên C/C++
* Có thể Nhúng vào ứng dụng như một giao tiếp kịch bản

Sau khi xem xét các tính năng nhóm quyết định sử dụng Python để thực hiện đề tài.

**b. Thư viện Open CV**

OpenCV (OpenSource Computer Vision) là một thư viện mã nguồn mở. OpenCV chính thức ra mắt vaò năm 1999 được phát triển đầu tiên bởi Intel. Thư viện đa nền tảng và miễn phí sử dụng theo giấy phép BSD nguồn mở. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực.OpenCV có các giao diện cho C/C++, Python, Java vì vậy hỗ trợ được cho Windowns, Linux,MacOs lẫn Android

Các tính năng của thư viện OpenCV:

* Bộ công cụ 2D và 3D.
* Nhận diện khuôn mặt.
* Nhận diện cử chỉ.
* Nhận dạng chuyển động, đối tượng, hành vi.
* Tương tác giữa con người và máy tính.
* Điều khiển Robot.
* Hỗ trợ thực tế tăng cường.

Chính vì những lý do trên, nhóm đề tài quyết định dùng thư viện OpenCV làm thư viện chính, viết trên ngôn ngứ Python để thực hiện đề tài

## **2.2 Phương pháp Nhận dạng khuôn mặt**

Nhận dạng mặt người (Face recognition) là một lĩnh vực nghiên cứu của ngành Computer Vision, và cũng được xem là một lĩnh vực nghiên cứu của ngành Biometrics (tương tự như nhận dạng vân tay – Fingerprint recognition, hay nhận dạng mống mắt – Iris recognition). Xét về nguyên tắc chung, nhận dạng mặt có sự tương đồng rất lớn với nhận dạng vân tay và nhận dạng mống mắt, tuy nhiên sự khác biệt nằm ở bước trích chọn đặt trưng (feature extraction) của mỗi lĩnh vực. Trong khi nhận dạng vân tay và mống mắt đã đạt tới độ chín, tức là có thể áp dụng trên thực tế một cách rộng rãi thì nhận dạng mặt người vẫn còn nhiều thách thức và vẫn là một lĩnh vực nghiên cứu thú vị với nhiều người. So với nhận dạng vân tay và mống mắt, nhận dạng mặt có nguồn dữ liệu phong phú hơn (bạn có thể nhìn thấy mặt người ở bất cứ tấm ảnh, video clip nào liên quan tới con người trên mạng) và ít đòi hỏi sự tương tác có kiểm soát hơn (để thực hiện nhận dạng vân tay hay mống mắt, dữ liệu input lấy từ con người đòi hỏi có sự hợp tác trong môi trường có kiểm soát). Hiện nay các phương pháp nhận dạng mặt được chia thành nhiều hướng theo các tiêu chí khác nhau: nhận dạng với dữ liệu đầu vào là ảnh tĩnh 2D (still image based FR) là phổ biến nhất, tuy nhiên tương lai có lẽ sẽ là 3D FR (vì việc bố trí nhiều camera 2D sẽ cho dữ liệu 3D và đem lại kết quả tốt hơn, đáng tin cậy hơn), cũng có thể chia thành 2 hướng là: làm với dữ liệu ảnh và làm với dữ liệu video. Trên thực tế người ta hay chia các phương pháp nhận dạng mặt ra làm 3 loại: phương pháp tiếp cận toàn cục (global, như Eigenfaces-PCA, Fisherfaces-LDA), phương pháp tiếp cận dựa trên các đặc điểm cục bộ (local feature based, như LBP, Gabor wavelets) và phương pháp lai (hybrid, là sự kết hợp của hai phương pháp toàn cục và local feature). Phương pháp dựa trên các đặc điểm cục bộ đã được chứng minh là ưu việt hơn khi làm việc trong các điều kiện không có kiểm soát và có thể nói rằng lịch sử phát triển của nhận dạng mặt (A never ending story) là sự phát triển của các phương pháp trích chọn đặc trưng (feature extractrion methods) được sử dụng trong các hệ thống dựa trên feature based. Các ứng dụng cụ thể của nhận dạng mặt dựa trên 2 mô hình nhận dạng: identification (xác định danh tính, bài toán 1-N), và verification (xác thực danh tính, bài toán 1-1). Trong bài toán identification, ta cần xác định danh tính của ảnh kiểm tra, còn trong bài toán verification ta cần xác định 2 ảnh có cùng thuộc về một người hay không.

Để xây dựng một hệ thống nhận dạng khuôn mặt bước đầu tiên phải phát hiện được phần ảnh của mặt trong dữ liệu đưa vào và cắt phần ảnh đó để thực hiện nhận dạng. Bước thứ hai tiền xử lý bao gồm các bước căn chỉnh ảnh và chuẩn hoá ánh sáng. Tiếp theo là trích chọn điểm ảnh ở đây sử dụng HOG và SVM để trích xuất các thông tin đặc trưng cho ảnh. Mỗi bức ảnh sẽ được biểu diễn dưới dạng các vector. Tiếp theo là nhận dạng hay so sánh với các ảnh đã có ở thư viện mà chúng ta cần so sánh.

A screenshot of a cell phone

Description automatically generated

Hình 2-7: Các bước tiến hành để xử lí ảnh

## **2.3 Phương pháp HOG (Histogram of Oriented Gradient)**

HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “featura descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh. Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là “tế bào” (cells) và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Có 5 bước cơ bản để xây dựng một vector HOG cho hình ảnh, bao gồm:

* + Tiền xử lý
  + Tính gradient
  + Tính vector đặc trưng cho từng ô (cells)
  + Chuẩn hoá khối (blocks)
  + Tính toán vector (HOG)

### **2.3.1 Tiền xử lý**

A group of people on a court

Description automatically generated

Hình 2-8: Điều chỉnh kích thước ảnh cần xử lí

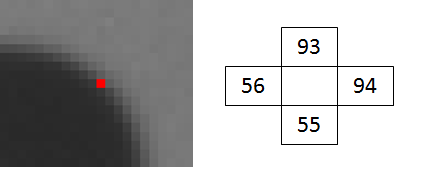
Ví dụ như bài toán trong hình việc đầu tiên chúng ta cần phải chỉnh lại kích thước của ảnh mà mình cần xử lý.

### **2.3.2 Tính Gradient**

Thực hiện bằng phép nhân chập ảnh gốc 2 chiều, tương ứng với các bài toán lấy đạo hàm theo hai hướng Ox và Oy. Tyrong đó, 2 hướng tương ứng đó là:

Dx =. [-1 0 1] và Dy = [1 0 -1]

Ví dụ ta có một điểm ảnh như sau:



Hình 2-9: Ví dụ điểm ảnh cần xử lí

Áp dụng công thức như trình bày ở trên để tính được gradient của điểm ảnh này:

A screenshot of a cell phone

Description automatically generated

Tương tự với bức ảnh đã trình bày ở mục một ta sẽ thu được:

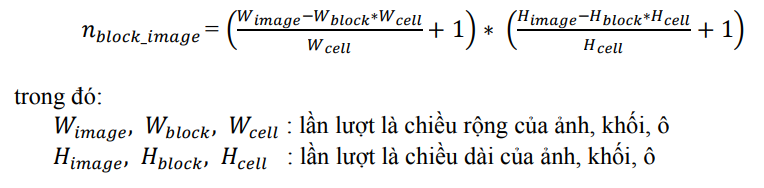
A picture containing photo

Description automatically generated

Hình 2-10: Giá trị tuyệt đối của x-gradient, y-gradient và cường độ gradient

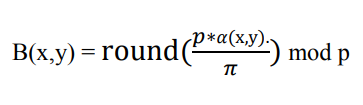
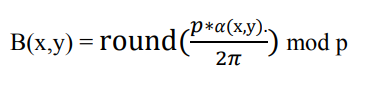
**2.3.3 Tính vector đặc trưng cho từng ô**

Để tính toán vector đặc trưng cho từng ô (cell), chúng ta cần chia hình ảnh thành các block, mỗi block lại chia đều thành các cell. Để xác định được số block, chúng ta sẽ sử dụng công thức sau:

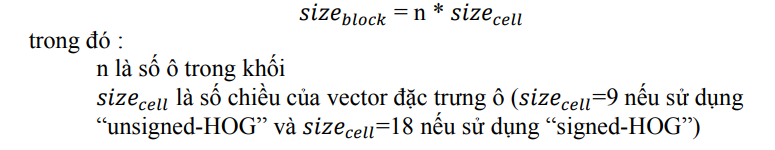
  
Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

1. Chia không gian hướng thành p bin (số chiều vector đặc trưng của ô).
2. Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin.

Giả sử góc hướng nghiêng tại pixel ở vị trí (x,y) có độ lớn là alpha(x,y)

* Trường hợp rời rạc hóa unsigned-HOG với p=9: 
* Trường hợp rời rạc hóa signed-HOG với p=18:

Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối. Số chiều vector đặc trưng khối tính theo công thức:



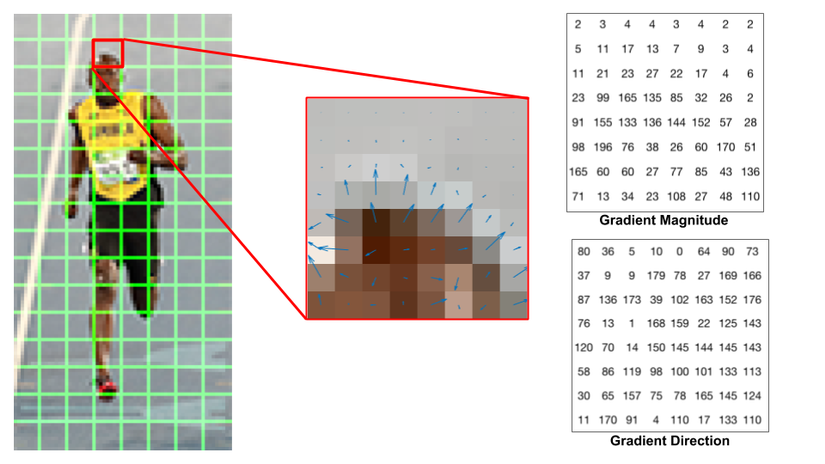
**Ví dụ:** Trong trường hợp này, hình ảnh của chúng ta có kích thước là 64x128, ta sẽ chia mỗi hình ảnh thành các block có kích thước 16x16. Mỗi block sẽ bao gồm 4 cell, mỗi cell có kích thước là 8x8.

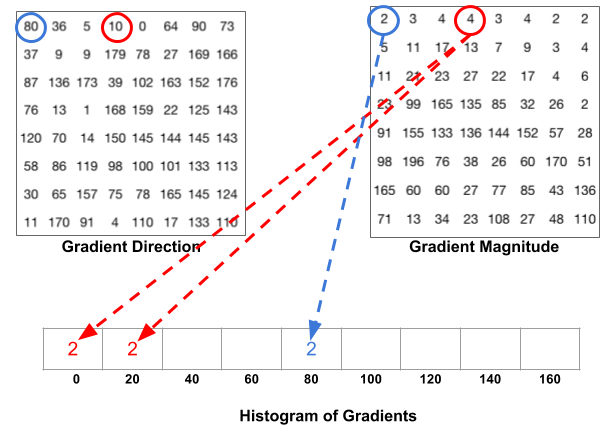


Hình 2-11: Chia hình ảnh thành các block

Tiếp theo, tiến hành tính toán đặc trưng HOG tại mỗi cell sử dụng không gian hướng 9 bin, trường hợp “**unsigned-HOG**”. Hướng gradient sẽ chạy trong khoảng 0 độ đến 180 độ, trung bình 20 độ mỗi bin.

Tại mỗi cell, xây dựng một biểu đồ cường độ gradient bằng cách vote các pixel vào biểu đồ. Trọng số vote của mỗi pixel phụ thuộc hướng và cường độ gradient (được tính toán từ bước 2) của pixel đó. Ví dụ:





Hình 2-12: Xây dựng cường độ biểu đồ gradient

Như trong hình ảnh trên, đầu tiên là pixel có bao quanh màu xanh lam. Nó có hướng 80 độ và cường độ là 2, vì vậy ta thêm 2 vào bin thứ 5 (hướng 80 độ). Tiếp theo là pixel có bao quanh màu đỏ. Nó có hướng 10 độ và cường độ 4. Vì không có bin 10 độ, nên ta vote cho bin 0 độ và 20 độ, mỗi bin thêm 2 đơn vị. Sau khi vote hết các pixel trong một cell kích thước 8x8 vào 9 bin, ta có thể thu được kết quả như sau:

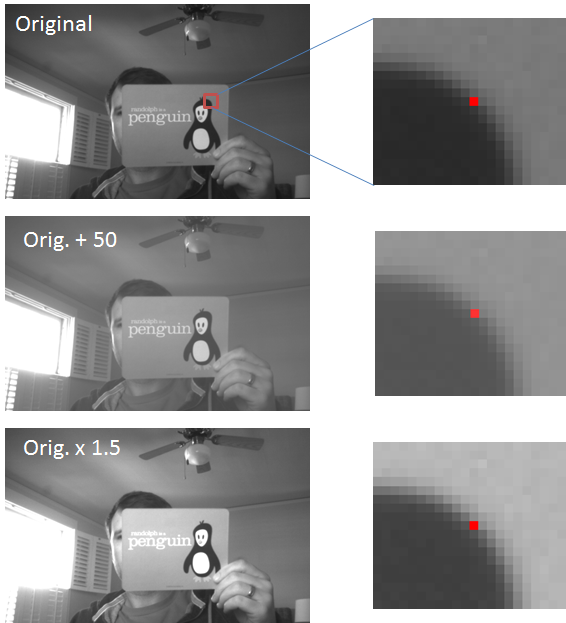


Hình 2-13: Histogram của gradient

### **2.3.4 Chuẩn hóa khối (blocks)**

Để tăng cường hiệu năng nhận dạng, các histogram cục bộ sẽ được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một khối và sử dụng giá trị đó để chuẩn hóa tất cả các ô trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

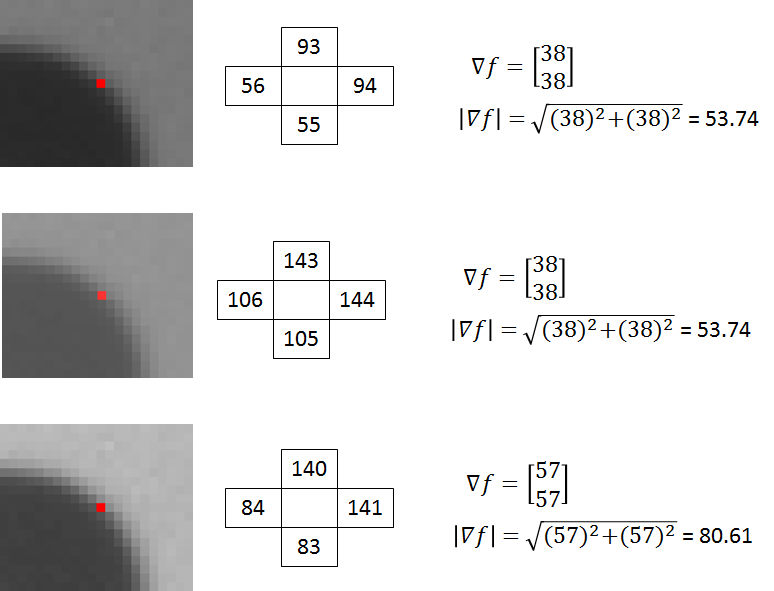
Đầu tiên, hãy xem xét ảnh hưởng của việc chuẩn hóa tới các vector gradient trong ví dụ sau:



Hình 2-14: Ảnh hưởng của việc chuẩn hóa tới các vector gradient

Trong hình ảnh trên, trường hợp đầu tiên là một ô của hình ảnh ban đầu. Trường hợp thứ hai, tất cả các giá trị pixel đã được tăng lên 50. Trong trường hợp thứ ba, tất cả các giá trị pixel được nhân với 1.5. Dễ dàng thấy được, trường hợp thứ ba hiển thị độ tương phản gia tăng. Ảnh hưởng của phép nhân là làm các điểm ảnh sáng trở nên sáng hơn nhiều, trong khi các điểm ảnh tối chỉ sáng hơn một chút, do đó làm tăng độ tương phản giữa các phần sáng và tối của hình ảnh.

Hãy nhìn vào các giá trị pixel thực tế và sự thay đổi của vector gradient của ba trường hợp trên trong hình ảnh sau:



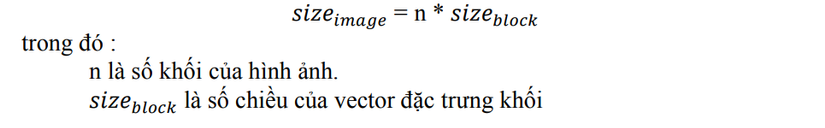
Hình 2-15: Các giá trị pixel thực tế và sự thay đổi của vector gradient

Trong trường hợp một và hai, giá trị cường độ vector gradient của chúng tương đương nhau, nhưng trong trường hợp thứ ba, cường độ vector gradient đã tăng lên 1.5 lần. Nếu chia ba vector bằng độ lớn tương ứng, ta sẽ nhận được các kết quả tương đương cho cả ba trưởng hợp. Vì vậy, trong ví dụ trên, chúng ta thấy rằng bằng cách chia các vector gradient theo độ lớn của chúng, chúng ta có thể biến chúng thành bất biến để thay đổi độ trương phản.

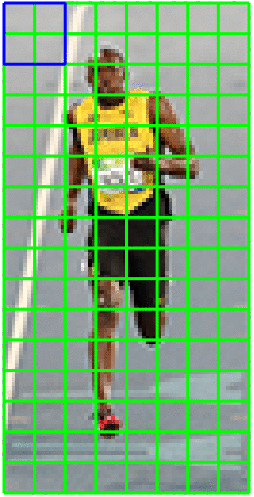
Có nhiều phương pháp có thể được dùng để chuẩn hóa khối. Gọi v là vector cần chuẩn hóa chứa tất cả các histogram của mội khối. ‖v(k)‖ là giá trị chuẩn hóa của v theo các chuẩn k=1, 3 và e là một hằng số nhỏ. Khi đó, các giá trị chuẩn hóa có thể tính bằng một trong những công thức sau:



Ghép các vector đặc trưng khối sẽ thu được vector đặc trưng R-HOG cho ảnh. Số chiều vector đặc trưng ảnh tính theo công thức :



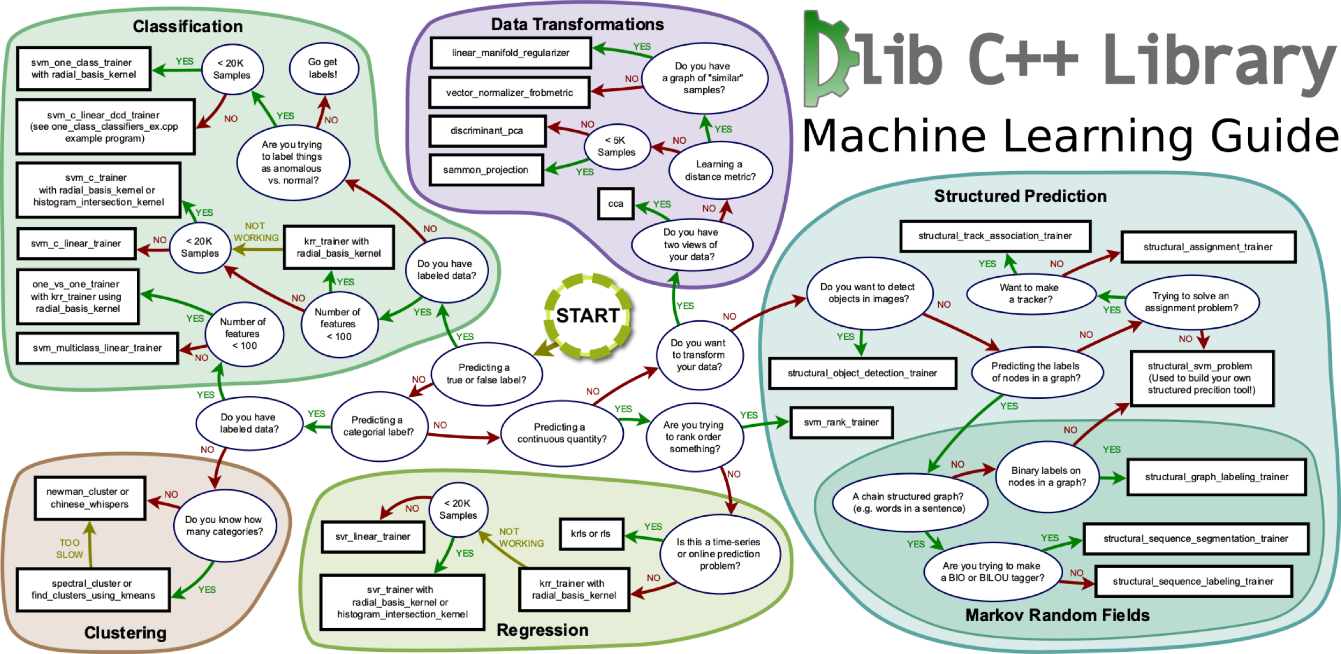
### **2.3.5 Tính toán vector đặc trưng HOG**



Hình 2-16: Chia hình ảnh thành các block để tính toán vector đặc trưng HOG

* Với mỗi hình ảnh kích thước 64x128, chia thành các block 16x16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có 7x15=105 blocks.
* Mỗi block gồm 4 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước 36x1.
* Vì vậy, khi nối tất cả các vector trong một block lại với nhau, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước 36x1x105 = 3780x1.

## **2.4 Thư viện Dlib**



Hình 2-17: Thư viện Dlib

Ngoài thư viện OpenCV, chúng tôi còn sử dụng dlib, một thư viện mã nguồn mở khác cho việc cài đặt hệ thống. Dlib được tạo ra từ 2002 bởi tác giả Davis King, được viết trên ngôn ngữ lập trình C++. Khác với mục đích của OpenCV là cung cấp hạ tầng thuật toán cho các ứng dụng xử lý ảnh và thị giác máy tính, dlib được thiết kế cho các ứng dụng máy học và trí tuệ nhân tạo với các thư viện con chính như sau: - Classification: các kỹ thuật phân lớp chủ yếu dựa trên hai phương pháp cơ sở là kNN và SVM.

Data transformation: các thuật toán biến đổi dữ liệu nhằm giảm số chiều, loại bỏ

các dữ liệu dư thừa và tăng cường tính khác biệt (discriminant) của các đặc điểm

được giữ lại

- Clustering: các kỹ thuật phân cụm

- Regression: các kỹ thuật hồi qui

- Structure prediction: các thuật toán dự đoán có cấu trúc

- Markov Random Fields: các thuật toán dựa trên các trường Markov ngẫu nhiên

Cụ thể trong bài này, phần phát hiện khuôn mặt người một cách tự động sẽ được lập trình bằng cách sử dụng thư viện dlib với dữ liệu huấn luyện và kỹ thuật hồi qui đã được cung cấp trước.

# **CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG**

## **3.1. THIẾT KẾ SƠ ĐỒ KHỐI**

Dựa vào nguyên lý hoạt động của hệ thống mà chúng ta có thể xây dựng được sơ đồ khối tổng quan cho cả hệ thống:





Hình 3-1: Sơ đồ khối toàn bộ hệ thống

* Khối Web: có chức năng điều khiển thiết bị qua Web.
* Khối MCU: xử lý tín hiệu từ các khối ngõ vào và xuất tín hiệu điều khiển các thiết bị ngõ ra.
* Khối điều khiển: Điều khiển thiết bị qua màn hình cảm ứng
* Khối wifi: tạo môi trường internet để cập nhật và theo dõi dữ liệu trên webserver.
* Khối nguồn: cấp nguồn cho toàn bộ mạch hoạt động.
* Khối tải: nhận lệnh từ khối điều khiển, thực thi, gửi phản hồi về khối điều khiển
* Khối App: Điều khiển thiết bị qua App Android
* Khối camera: Cung cấp hình ảnh đầu vào để nhận dạng
* Khối Raspberry Pi 3b: xử lý hình nảh sau khi nhận được từ camera
* Khối điều khiển: điều khiển cửa sau khi nhận dạng đúng người dùng
* Khối Lưu kết quả: lưu kết quả torng lúc nhận dạng

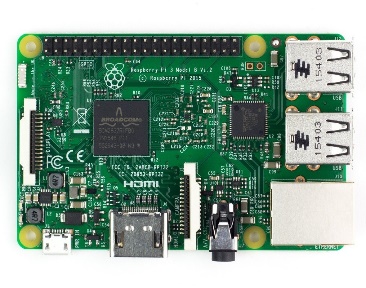
## **3.2. RASPBERRY PI 3 MODEL B**

### **3.2.1 Giới thiệu Raspberry Pi 3**

Nội dung viết lại

Raspberry Pi được phát triển đầu tiên vào năm 2012. Raspberry Pi ban đầu là như một thẻ card được cắm trên bo mạch máy tính được phát triển bởi các nhà phát triển ở Anh. Sau đó Raspberry Pi đã được phát triển thành một bo mạch đơn có chức năng như một máy tính mini dùng để giảng dạy ở các trường trung học. Được phát triển bởi Raspberry Pi Foundation – là tổ chức phi lợi nhuận với tiêu chí xây dựng hệ thống mà nhiều người có thể sử dụng được trong những công việc tùy biến khác nhau.Raspberry Pi sản xuất bởi 3 OEM: Sony,Qsida, Egoman. Và được phân phối chính bởi Element14, RS Components và Egoman. Mặc dù chậm hơn so với các dòng laptop, máy tính hiện đại nhưng Raspberry Pi vẫn được xem là máy tính Linux hoàn chỉnh và có thể cung cấp tất cả các khả năng mà mọi người mong đợi, với mức tiêu thụ năng lượng thấp.

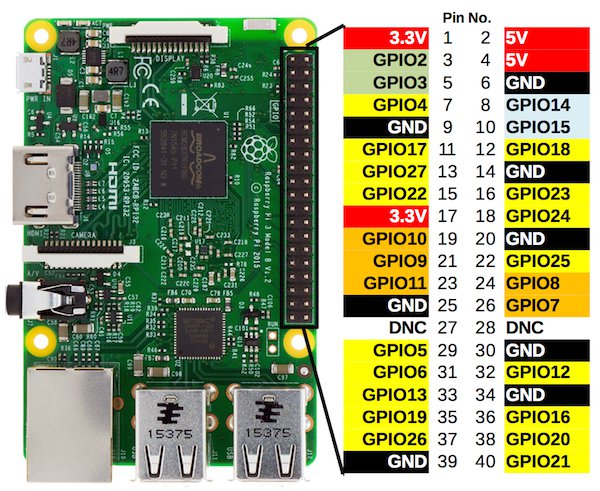
Nội dung viết lại



Hình 3-2: Raspberry Pi 3B

* Cấu hình của Raspberry Pi 3B
  + Bộ xử lí Broadcom BCM2835 tốc độ xử lí 1.2ghz 64-bit quad-core ARM CortexA53
  + Mạng Wireless LAN chuẩn 802.11 b/g/n
  + Bộ xử lý đa phương tiện Videocore IV® Dual Core
  + Bộ nhớ Ram 1GB
  + Hỗ trợ tất các bản phân phối ARM GNU/Linux mới nhất và Windows10
  + Đầu nối microusb cho nguồn điện 2,5A 5VDC
  + Cổng mạng 1 x 10/100
  + Đầu nối video/âm thanh 1 x HDMI
  + Đầu nối video/âm thanh 1 x RCA
  + 4 x USB 2.0 ports
  + 40 GPIO pins
  + Chip antenna
  + Kết nối hiển thị DSI
  + Khe cắm thẻ nhớ MicroSD
  + Kích thước: 85 x 56 x 17 mm

### **3.2.2 Nội dung thêm vào**



Hình 3-3: Sơ đồ chân của Raspberry Pi 3B

## **3.3 STM32F407VG**



Hình 3-4: Kit STM32F407VG

Kit STM32F407VGT6 Discovery được thiết kế phục vụ nghiên cứu họ STM32F4 và phát triển các ứng dụng nhúng. STM32 là chip của ST, dựa trên nền lõi ARM Cortex-M. Dòng ARM CortexTM – M là dòng thế hệ mới, thiết lập các tiêu chuẩn mới về hiệu suất, chi phí, ứng dụng cho các thiết bị cần tiêu thụ năng lượng thấp và đáp ứng yêu cầu thời gian thực khắc khe.

**Các tính năng chính**

- Vi điều khiển 32-bit ARM Cortex®-M4 STM32F407VGT6 với lõi FPU hỗ trợ xử lý tính toán dấu phẩy động, 1-MB bộ nhớ Flash, 192 Kbyte RAM.

- On-board ST-LINK/V2 trên STM32F4-DISCOVERY giúp nạp chương trình, gỡ lỗi

- Nguồn điện cung cấp cho kit: thông qua cổng USB hoặc từ một nguồn cung cấp điện áp 5V bên ngoài.

- Từ kit, có thể cấp nguồn 3,3 V và 5 V cho các ứng dụng.

- Cảm biến chuyển động LIS302DL, ST MEMS 3 trục gia tốc.

- Cảm biến âm thanh MP45DT02 ST-MEMS, mic cảm biến âm thanh vô hướng kỹ thuật số.

- Bộ chuyển đổi DAC âm thanh CS43L22.

- Tám đèn LED:

+ LD1 (đỏ /xanh lá cây) để giao tiếp USB

+ LD2 (màu đỏ) báo hiệu nguồn 3,3 V

+ Bốn đèn LED màu: LD3 (cam), LD4 (xanh lá cây), LD5 (đỏ) và LD6 (xanh dương).

+ Hai USB OTG LED LD7 (xanh lá cây) VBUS và LD8 (màu đỏ).

- Hai nút bấm (nút bấm User màu xanh, nút bấm Reset màu đen).

- OTG FS USB với cổng nối micro-AB.

- Header mở rộng cho tất cả LQFP100 I/O.

- Phần mềm miễn phí bao gồm một loạt các ví dụ, sử dụng thư viện chuẩn của ST

### **3.3.1 Bộ điều khiển reset**

Có 3 loại reset là:

* System Reset
* Power Reset
* Backup Domain Reset.

**a. System Reset**

System Reset sẽ reset giá trị của tất cả các thanh ghi ngoại trừ các cờ trong thanh ghi điều khiển xung clock và các thanh ghi của Backup Domain.

System Reset xảy ra khi một trong những sự kiện sau phát sinh:

* Tác động vào chân NRST bằng điện áp mức thấp (reset ngoài).
* Window Watchdog đếm hết giá trị đặt (WWDG reset).
* Independent Watchdog đếm hết giá trị đặt (IWDG reset).
* Reset bằng phần mềm (SW reset).
* Reset do việc điều khiển các chế độ tiết kiệm năng lượng.
* Reset xảy ra khi đưa ARM vào chế độ Standby.
* Reset xảy ra khi đưa ARM vào chế độ Stop

Nguyên nhân của Reset có thể biết được thông qua việc kiểm tra các cờ báo Reset trong thanh ghi Control/Status (RCC\_CSR ở địa chỉ 0x24 – các cờ báo reset trong thanh ghi này chỉ có thể được reset bằng Power Reset).

**b. Power reset**

Power reset sẽ reset giá trị của tất cả các thanh ghi về trạng thái mặc định của nó ngoại trừ các thanh ghi của Backup Domain.

Power Reset xảy ra khi một trong những sự kiện sau phát sinh:

* Mở nguồn/ Sụt áp (POR/PDR reset)
* Thoát khỏi chế độ Standby.

**c. Backup domain reset**

Backup Domain Reset xảy ra khi một trong những sự kiện sau phát sinh:

* Reset bằng phần mềm, kích hoạt khi đặt bit BDRST (bit thứ 16) trong thanh ghi Backup domain controll (RCC\_BDCR địa chỉ 0x20) mức ‘1’.
* Khi cấp nguồn VDD hay VBAT mà cả hai nguồn này trước đó đều tắt (người ta thường dùng để kiểm tra việc hết Pin).

### **3.3.2 Xung clock**

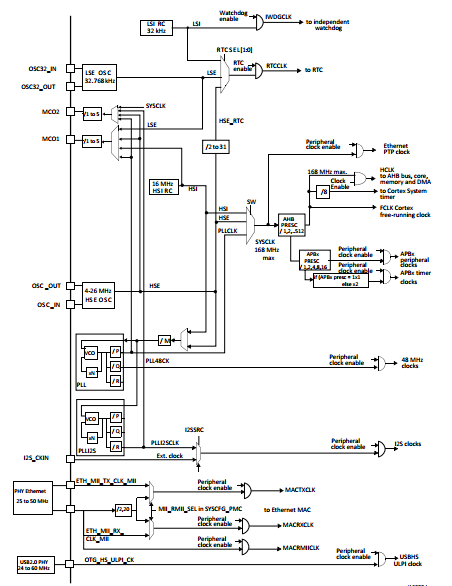
Có 3 loại xung clock có thể được dùng làm xung clock hệ thống (SYSCLK - xung clock cấp cho khối xử lý):

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

Ngoài ra ARM cũng còn có 2 nguồn xung clock phụ sau:

* LSI (Low Speed Internal) nguồn xung clock tốc độ chậm 32 kHz ở bên trong ARM nguồn xung clock này được dùng để cấp cho independent watchdog và có thể được dùng để cấp cho RTC trong việc định thời gian tự động đánh thức CPU thoát khỏi chế độ Sop hoặc Stanbly.
* 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.

Mỗi nguồn xung clock có thể được bật tắt độc lập nhằm tiết kiệm năng lượng.

****

Hình 3-5: Sơ đồ cây xung Clock

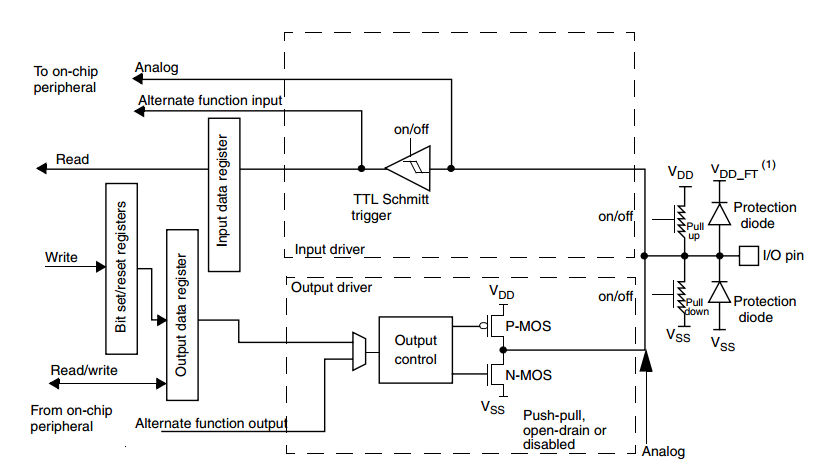
### **3.3.3 Gpio**

**a. Giới thiệu GPIO**

GPIO (General - purpose input/output): ngõ vào - ngõ ra sử dụng chung.

* Đối với các dòng STM32 thì mỗi PORT có 16 chân I/O.
* Ngoài chức năng là I/O nếu muốn sử dụng các chân này làm ngõ vào hoặc ngõ ra của ngoại vi ta phải thiết lập chân này là chức năng thay thế (Alternate Function) hoặcAnalog.
* Để tiện cho thiết kế phần cứng nhà sản xuất xây dựng chức năng remap cho phép người dùng có thể thay đổi vị trí I/O của ngoại vi trong một phạm vi nhất định.
* Mỗi chân I/O bên trong chip đều được gắn điện trở nội kéo lên và kéo xuống.

**b. Cấu trúc cơ bản của 1 chân I/O**

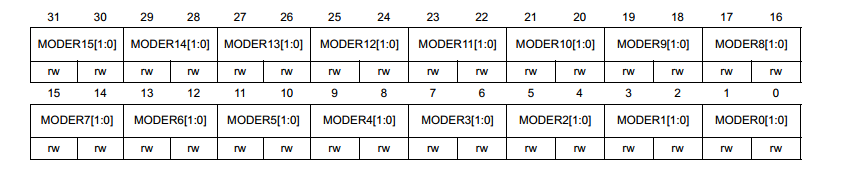
****

Hình 3-6: Cấu trúc cơ bản của một chân I/O

**c. Các thanh ghi và lệnh liên quan đến GPIO**

Thanh ghi GPIOx\_MODER (General Purpose I/O Mode Register)

(Địa chỉ: 0x00**,** giá trị mặc định sau khi reset:0xA800 0000 cho port A**,** 0x0000 0280 cho port B**,** 0x0000 0000 cho những port khác**)**

****

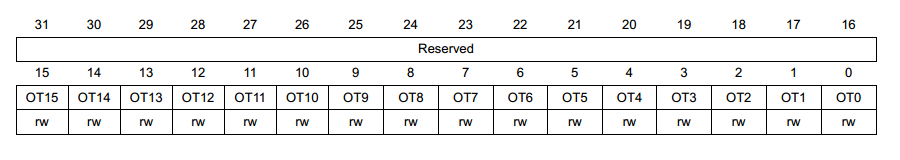
Hình 3-7: Thanh ghi GPIOx\_MODER

Thanh ghi GPIOx\_MODER dùng để cấu hình các chân từ 0 đến 15 của PORTx (2 bit cấu hình cho 1 chân). Chân thứ “y” có thể được cấu hình bằng cách thay đổi giá trị bit MODERy[1:0]

* 00: input (trạng thái reset)
* 01: chế độ ngõ ra sử dụng chung
* 10: chế độ chức năng thay thế
* 11: chế độ Analog

**Thanh ghi GPIOx\_OTYPER (G**eneral **P**urpose **I/O O**utput **T**ype **R**egister**)**

(Địa chỉ: 0x00, giá trị mặc định sau khi reset: 0x0000 0000)

****

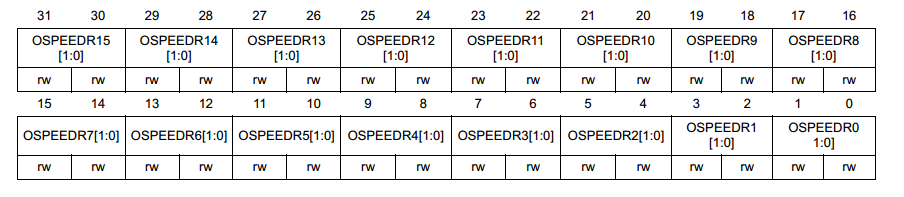
Hình 3-8: Thanh ghi GPIOx\_OTYPER

Thanh ghi GPIOx\_OTYPER dùng để cấu hình các chân từ 0 đến 15 của PORTx (mỗi bit cấu hình cho 1 chân). Chân thứ “y” có thể được cấu hình bằng cách thay đổi giá trị bit OTy.

* Bit 31:16 được đặt trước, phải được giữ ở giá trị mặc định sau khi reset.
* Bit 15:0 là các bit cấu hình của Port x (y=0…15)
* 0: Output push-pull (Reset state)
* 1: Output open-drain

**Thanh ghi GPIOx\_OSPEEDR (G**eneral **P**urpose **I/O O**utput **S**peed **R**egister**)**

(Địa chỉ: 0x80, giá trị mặc định sau khi reset: 0x0000 0000)



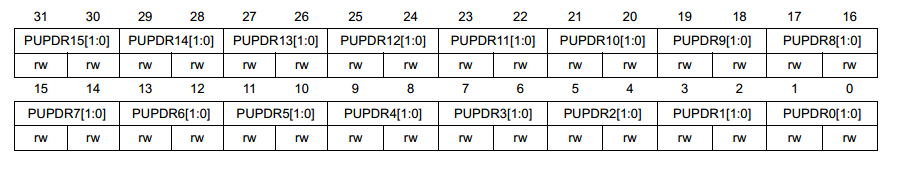
Hình 3-9: Thanh ghi GPIOx\_ OSPEEDR

Thanh ghi GPIOx\_ OSPEEDR dùng để cấu hình các chân từ 0 đến 15 của PORTx (2 bit cấu hình cho 1 chân). Chân thứ “y” có thể được cấu hình bằng cách thay đổi giá trị bit OSPEEDRy[1:0].

* 00: Low speed
* 01: Medium speed
* 10: High speed
* 11: Very high speed

**Thanh ghi GPIOx\_PUPDR** (**G**eneral **P**urpose **I/O** **P**ull-**U**p/**P**ull-**D**own **R**egister)

(Địa chỉ: 0x0C, giá trị mặc định sau khi reset: 0x6400 0000 cho port A, 0x0000 0100 cho port B, 0x0000 0000 cho những port khác).

****

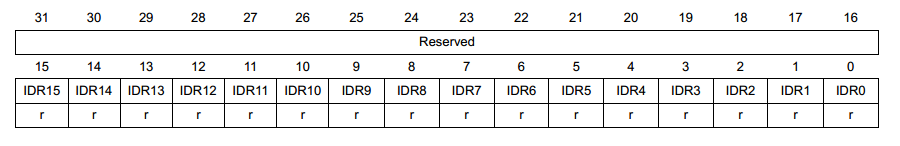
Hình 3-10: Thanh ghi GPIOx\_PUPDR

Thanh ghi GPIOx\_PUPDR dùng để cấu hình các chân từ 0 đến 15 của PORTx (2 bit cấu hình cho 1 chân). Chân thứ “y” có thể được cấu hình bằng cách thay đổi giá trị bit PUPDRy[1:0].

* 00: No pull-up, pull\_down
* 01: Pull-up
* 10: Pull-down
* 11: Reserved

Thanh ghi GPIOx\_IDR (General Purpose I/O Input Data Register)

(Địa chỉ 0x10, giá trị mặc định sau khi reset 0x0000 XXXX)

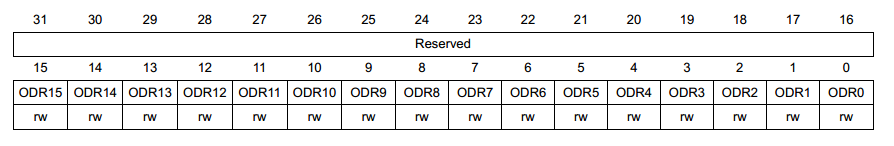
****

Hình 3-11: Thanh ghi GPIOx\_IDR

Thanh ghi GPIOx\_IDR chứa trạng thái logic đọc về từ các chân của PORTx.

**Thanh ghi GPIOx\_ODR (G**eneral **P**urpose **I/O O**uput **D**ata **R**egister**)**

(Địa chỉ 0x14, giá trị mặc định sau khi reset 0x0000 0000)



Hình 3-12: Thanh ghi GPIOx\_ODR

Thanh ghi GPIOx\_ODR chứa trạng thái logic xuất ra các chân của PORTx.

**Thanh ghi GPIOx\_BSRR (G**eneral **P**urpose **I**/**O B**it **S**et **R**eset **R**egister)

(Địa chỉ 0x18, giá trị mặc định sau khi reset 0x0000 0000)



Hình 3-13: Thanh ghi GPIOx\_BSRR

Thanh ghi GPIOx\_BSRR được dùng để xuất mức ‘1’ (điều khiển các bit từ 0 đến 15) hoặc xuất mức “0-reset” (điều khiển các bit từ 16 đến 31) ra các chân của PORTx.

Ví dụ điều khiển chân số 2 và chân số 5 của PORTx xuất mức ‘1’, chân số 0 và chân số 12 của PORTx xuất mức ‘0’. Ta phân tích yêu cầu như sau:

Do yêu cầu chân số 2 và chân số 5 xuất mức ‘1’nên tại vị trí bit số 2 (ứng với BS2 – set chân 2) và bit số 5 (ứng với BS5 – set chân 5) của thanh ghi GPIOx\_BSRR ta điền số 1, mặt khác chân số 0 và chân 12 xuất mức ‘0’nên tại vị trí bit số 16 (ứng với BR0 – reset chân 0) và bit thứ 28 (ứng với BR12 – reset chân 12) ta cũng điền số 1( số 1 là cho phép chứ không phải là mức ‘1’). Các vị trí còn lại ta điền số 0.

## **3.4 WEBCAM LOGITECH**

Webcam Logitech C270 là module được sản xuất bởi hãng công nghệ hàng đầu thế giới về camera đó là Logitech. Webcam có độ phân giải 3MP đảm bảo bức ảnh thu nhận được đảm bảo được độ sắc nét. Cùng với tính năng tự động điều chỉnh ánh sáng cho bạn có những bức ảnh đẹp trong điều kiện thiếu sáng.



Hình 3-14: Webcam Logitech C270

Webcam sử dụng giao tiếp USB 2.0 với thiết kế chắn chắn, cứng cáp giúp phát triển ứng dụng dễ dàng phù hợp với nhiều mục đích khác nhau.

Thông số Webcam Logitech C270:

* Độ phân giải 720p/30fps
* Tích hợp công nghệ Logitech RightSound
* Hình ảnh lên đến 3Mpx
* Quay video: 120 x 720 pixel
* Hi Speed USB 2.0
* Tự động cân bằng sáng
* Tự động lấy nét.

Với những tính năng như trên nhóm quyết định chọn Webcam Logitech C270 để thực hiện đề tài.

## **3.5 KHOÁ CHỐT ĐIỆN**

Khóa chốt điện Solenoid Lock LY-03 đi kèm gá chốt, có chức năng hoạt động như một ổ khóa cửa sử dụng Solenoid để kích đóng mở bằng điện, được sử dụng nhiều trong nhà thông minh hoặc các loại tủ, cửa điện,..., khóa sử dụng điện áp 12 / 24VDC, là loại thường đóng với chất lượng tốt, độ bền cao.



Hình 3-15: Khoá chốt điện

Thông số kĩ thuật:

* + Vật Liệu: Thép không gỉ
  + Nguồn điện 12VDC
  + Dòng điện làm việc 0.8A
  + Công Suất 9.6W
  + Yêu cầu nguồn cấp 12VDC/1A
  + Kích thước L54xD38xH2

Đề tài “Giám sát năng lượng điện tiêu thụ hộ gia đình” bao gồm bộ cảm biến đo thông số của thiết bị điện, truyền dữ liệu lên internet thông qua giao thức HTTP, dữ liệu sẽ được cập nhật và theo dõi liên tục qua giao diện web được thiết kế tiện lợi cho người giám sát. Để hệ thống hoạt động đạt hiệu quả cao thì phần tính toán và thiết kế rất quan trọng, trong phần này chúng ta cùng tính toán và thiết kế cụ thể từng phần.

## **3.6 NODEMCU ESP8266**

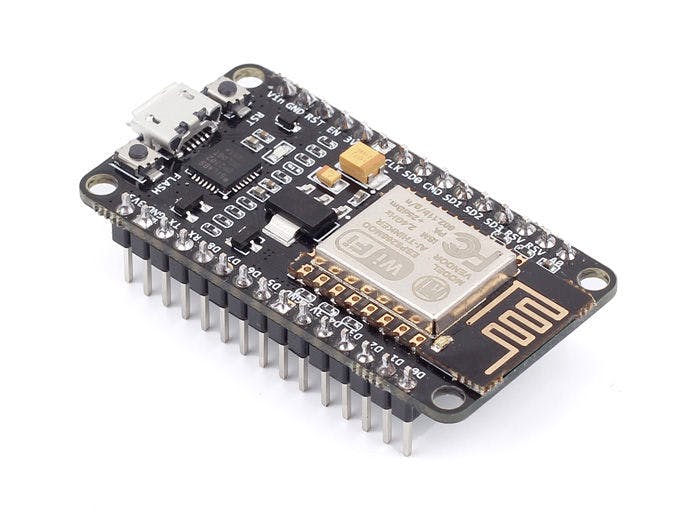
### **3.6.1 Giới thiệu chung**

Hiện nay có nhiều loại Kit wifi khác nhau như ESP8266 – V1, ESP8266 – 12, ESP8266 – 12E, Wemos D1… nhưng ESP8266 – 12E được sử dụng rộng rãi và phổ biến nhất. ESP8266 – 12E còn có tên là NodeMCU ESP8266 V3, được phát triển dựa trên nền tảng chip ESP8266 của Expressif. Trên Kit còn tích hợp IC CP2102 TTL giúp dễ dàng giao tiếp với máy tính thông qua cổng Micro-USB để lập trình và gỡ lỗi.

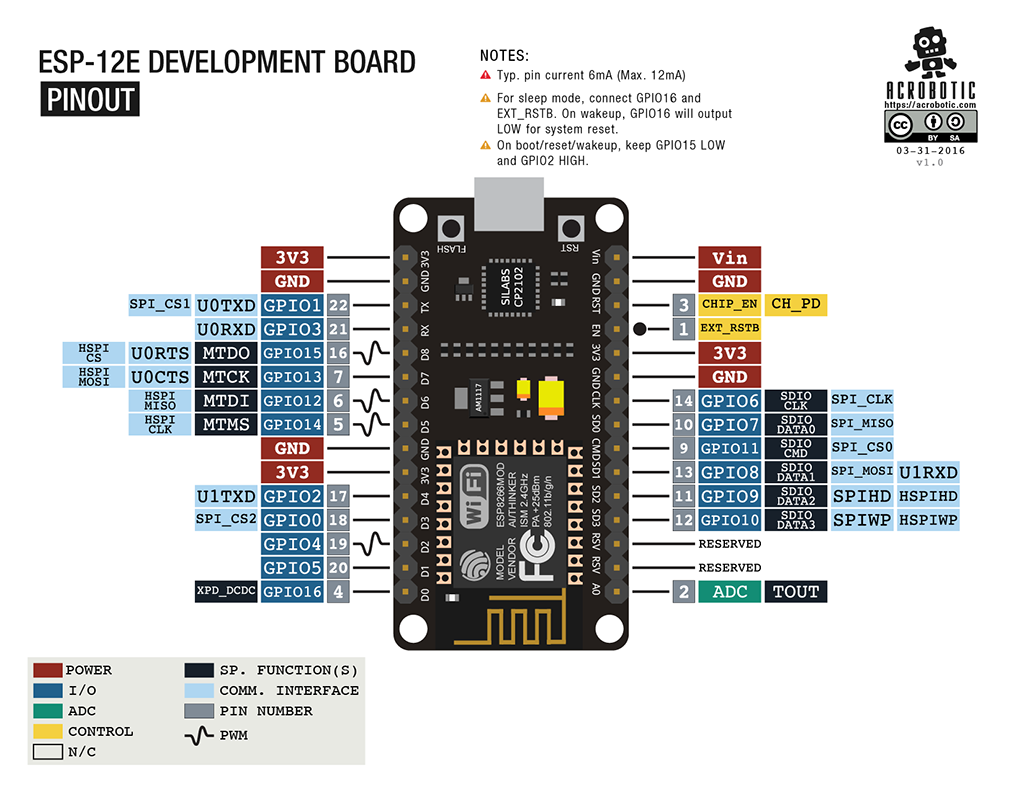
Hiện tại có hai ngôn ngữ có thể lập trình cho ESP8266, sử dụng trực tiếp phần mềm IDE của Arduino để lập trình với bộ thư viện riêng hoặc sử dụng phần mềm node MCU. Kit thích hợp và dễ dàng thực hiện các ứng dụng thu thập dữ liệu và điều khiển qua Wifi.

Thông số kỹ thuật:

* Điện áp cung cấp: DC 5 ~ 9V
* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Hỗ trợ bảo mật: WPA/WPA2
* Bộ nhớ Flash: 32MB
* Cổng kết nối: hỗ trợ USB-TTL CP2102 với cổng Micro-USB
* Giao tiếp dữ liệu: UART / HSPI / I2C / I2S /GPIO / PWM



Hình 3-16: NodeMCU V3



Hình 3-17: NodeMCU V1

## **3.9 SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH**

# **CHƯƠNG 4: THI CÔNG HỆ THỐNG**

## **4.1 GIỚI THIỆU**

## **4.2 THI CÔNG HỆ THỐNG**

### **4.2.1 Thi công board mạch**

Bảng 3-1: Thông tin các thiết bị, linh kiện sử dụng cho mạch

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên linh kiện | Thông số, giá trị | Kích thước, dạng vỏ |
| 1 | U7A | STM32F407VG | LQFP 100 14x14x1.4 |
| 2 | Nextion | Màn hình TFT Nextion | 3.2 inch, 95mm x 47.6mm |
| 3 | RL | Relay 5V 1 kênh | 1R5V Module,50x26x18.5mm |
| 4 | LS1 | Buzzer 5V | 12x6mm |
| 5 | U3 | Cảm biến DHT11 | Module 3 chân |
| 6 | Q (1,2,3,4,5) | Transistor C1815 | TO – 92 |
| 7 | R (5,6…14) | Điện trở 330Ω/0.5W/5% | Carbon Film |
| 8 | R (1,2,3,4) | Điện trở 10kΩ/0.5W/5% | Carbon Film |
| 9 | BT (1,2) | Nút nhấn đơn | Nút nhấn tròn, 2 chân, 5x4mm. |
| 10 | SW1 | Switch gạt đơn | DIP, màu đỏ |
| 11 | L (1,2,3,4) | Quang trở GL5528 | 5mm |
| 12 | VR (1,2,3,4) | Biến trở 5k | Biến trở vuông 3362P-103 |
| 13 | RL (1,2,3,4) | Relay 5VDC 5PIN | YL303H-S5VDC-1Z |
| 14 | U (1,2) | LM358 | 8-DIP |
| 15 | HD (1,2,3,4) | Domino 2 | Xanh dương |
| 16 | LED (5,6…9) | Led siêu sáng đỏ | 3mm |

Bảng 3-2: Thông tin các linh kiện sử dụng cho khối Wifi

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên linh kiện | Thông số, giá trị | Kích thước, dạng vỏ |
| 1 | LED1 | Led siêu sáng đỏ | 3mm |
| 2 | R | Điện trở 330Ω/0.5W/5% | Carbon Film |
| 3 | RL | Relay 5V 1 kênh | 1R5V Module, 50x26x18.5mm |
| 4 | Q1 | Transistor C1815 | TO – 92 |
| 5 | M1 | ESP8266 V1 | Module |
| 6 | C1, C4 | Hàng rào đực 2.54mm | Đen |
| 7 | ESP | NodeMCU ESP8266 | Module LoLin Lua |

Bảng 3-3: Thông tin các linh kiện sử dụng cho khối nguồn

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên linh kiện | Thông số, giá trị | Kích thước, dạng vỏ |
| 1 | LM2596 | LM2596 | Module |
| 2 | USB | Đế ra chân USB 2.0 4 PIN | Module |
| 3 | JDC | Jack cái DC 3.5x1.3mm | DC1135 DC Socket |
| 4 | LED | Led siêu sáng đỏ | 3mm |
| 5 | R | Điện trở 330Ω/0.5W/5% | Carbon Film |
| 6 | Header 4,5,6 | Hàng rào đực 2.54mm | Đen |

Hình ảnh mạch in

### **4.2.2 Lắp ráp và kiểm tra**

## **4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH**

## **4.4 LẬP TRÌNH HỆ THỐNG**

### **4.4.1 Lưu đồ giải thuật**

Phần này bỏ lưu đồ vào

### **4.4.2 Giới thiệu hệ điều hành raspbian**

### **4.4.3 Giới thiệu phần mềm android**

#### **4.4.3.1 Giới thiệu phần mềm Android Studio**



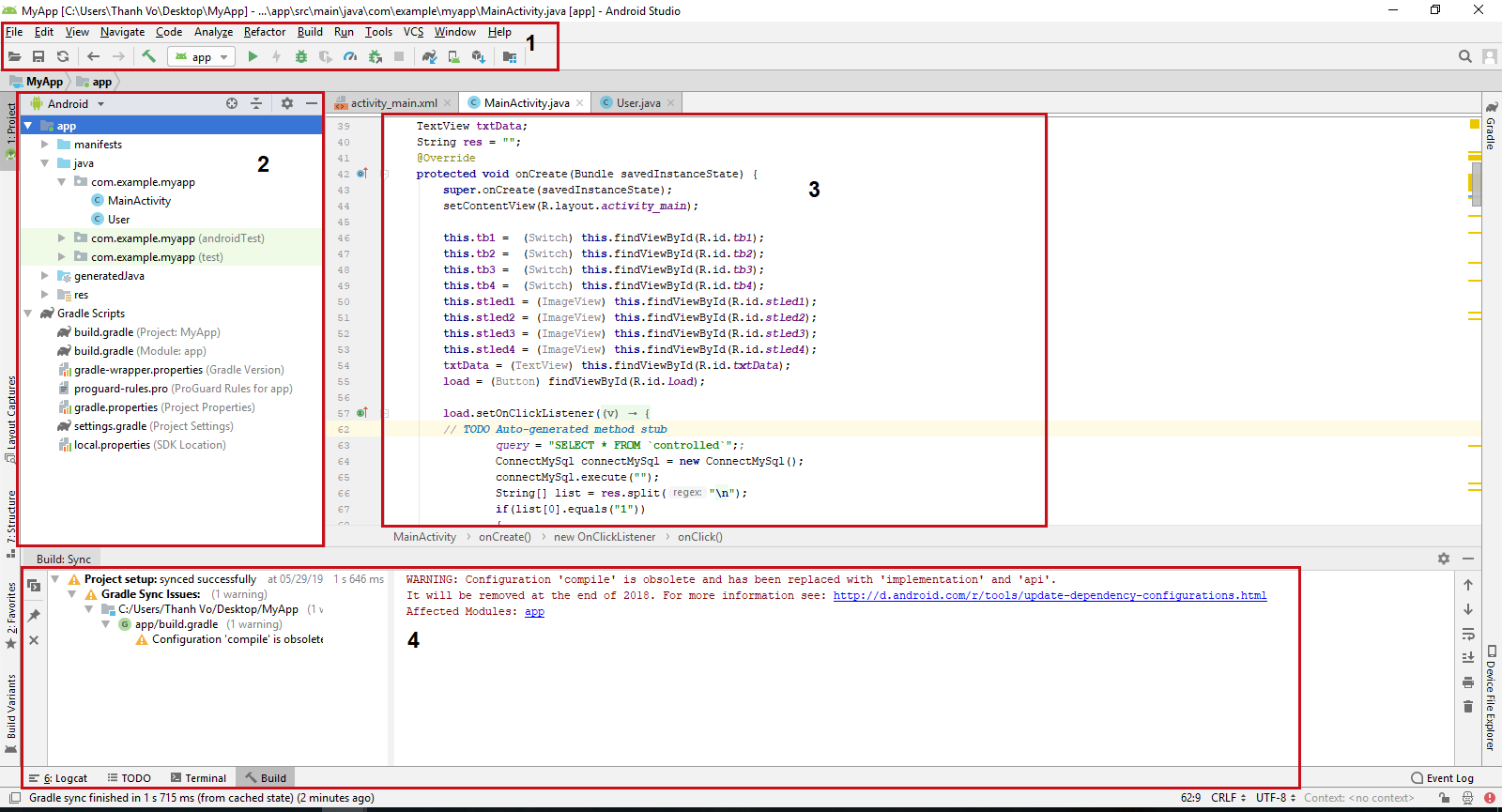
Hình 3-: **Phần mềm Android Studio**

**Hệ điều hành Android** đang chiếm hơn 80% thị trường thiết bị di động hiện đại ngày nay. Nhu cầu sử dụng ứng dụng di động cũng ngày càng cao. Nhu cầu việc làm về lập trình trên hệ điều hành mobile phổ biến nhất thế giới này cũng vì thế mà tăng mạnh do tính mở và dễ tiếp cận của nó. Trong đó ***Android Studio*** là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiét bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức. Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

Cửa sổ làm việc của Andoid được chia thành 4 vùng chính hỗ trợ lập trình như sau:

* Vùng 1: Thanh công cụ, thanh công cụ chứa nhiều nút chức năng cơ bản hỗ trợ lập trình.
* Vùng 2: Chứa cấu trúc thư mục của project: cho phép ta quản lý các tập tin trong Project, và cửa sổ Function, cho phép ta quản lý các chương trình con.
* Vùng 3: Vùng soạn thảo chương trình.
* Vùng 4: Cửa sổ Build: giúp chúng ta có thể kiểm tra tình trạng của project.



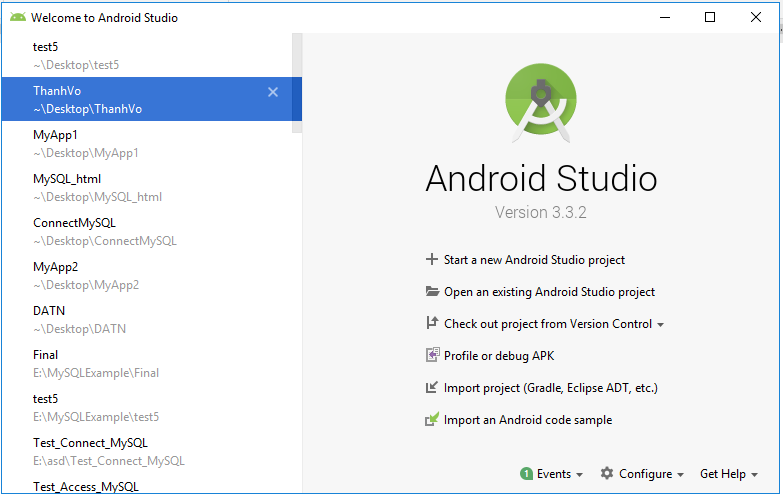
Hình 3-: Cửa sổ làm việc của Android Studio

Một project bao gồm các file cơ bản sau:

|  |  |
| --- | --- |
| File | Mô tả |
| AndroidManifest.xml | Đây là file manifest mô tả các đặc điểm cơ bản của ứng dụng và xác định từng thành phần của nó. |
| java | Thư mục này có chứa các file nguồn java cho dự án của bạn. Theo mặc định, nó bao gồm một tập tin nguồn MainActivity.java một lớp hoạt động (activity) chạy khi ứng dụng của bạn được khởi động |
| res/drawable | Các phiên bản Android trước đây sử dụng thư mục này để chứa ảnh, các phiên bản hiện tại sử dụng thư mục mipmap thay thế làm nơi chứa ảnh. Thư mục này gần như không còn sử dụng. |
| res/layout | Thư mục này chứa các file định nghĩa giao diện người dùng. |
| res/menu | Thư mục này chứa các file xml, định nghĩa các menu sẽ hiển thị trên Action Bar. |
| res/mipmap | Chứa các ảnh 'mipmap'. |
| res/values | Đây là một thư mục cho các tập tin XML khác nhau có chứa một tập hợp các nguồn, chẳng hạn như các chuỗi (String) và các định nghĩa màu sắc. |

#### **4.4.3.2 Các bước tạo một project với Android Studio**

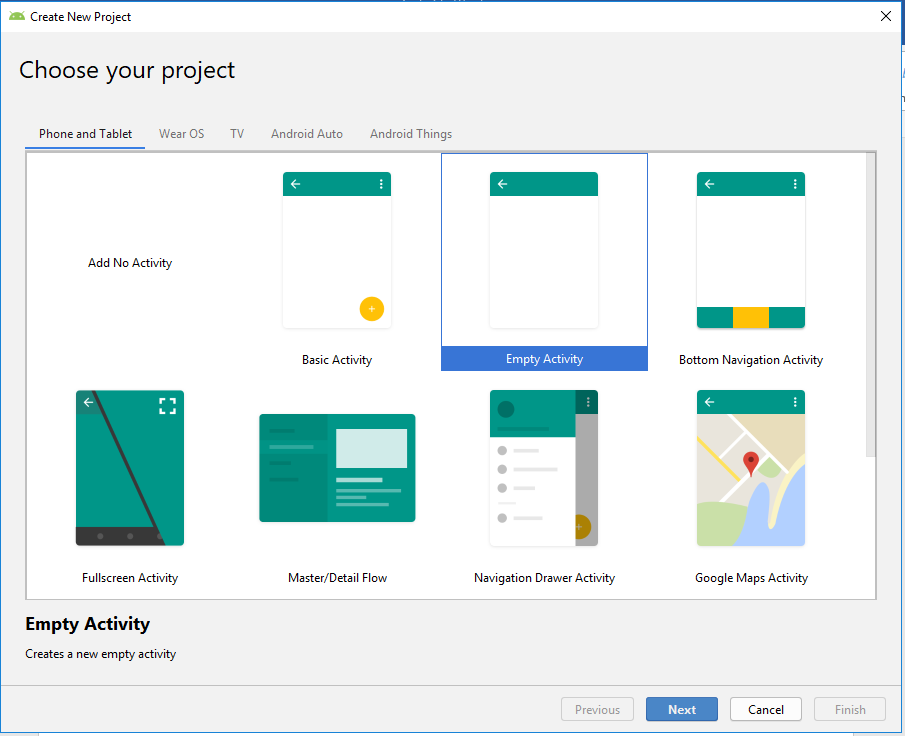
Click đúp vào biểu tượng ***Android Studio***trên màn hình, giao diện mới đầu vào sẽ có dạng như hình:



Hình 3-: Giao diện ban đầu của Android Studio

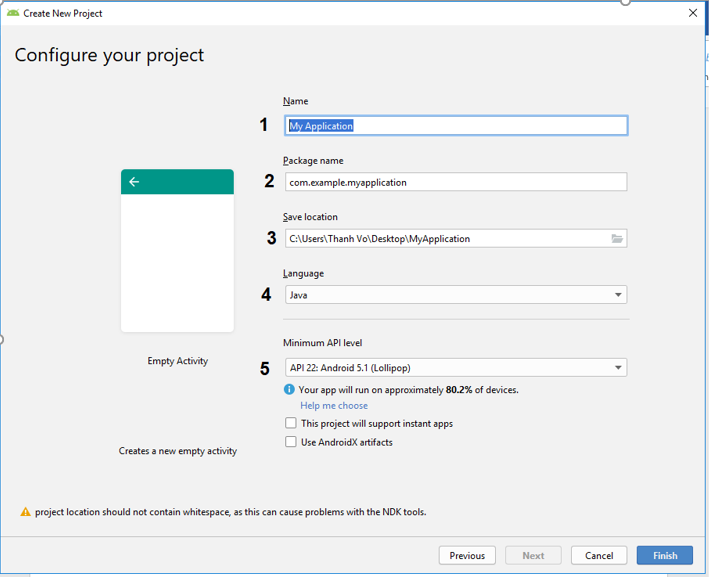
Click chọn Start a new Android Studio project để tạo mới 1 project hay chọn Open an existing Android Studio project để mở 1 project sẵn có.

Sau đó chọn thiết bị và mẫu màn hình của ứng dụng mà Android Studio đã cung cấp. Ứng dụng này được viết cho điện thoại android nên chọn Phone and Table và chọn mẫu màn hình Empty Activity. Nhấn Next để lưu thiết lập.



Hình 3-: Chọn màn hình cho ứng dụng

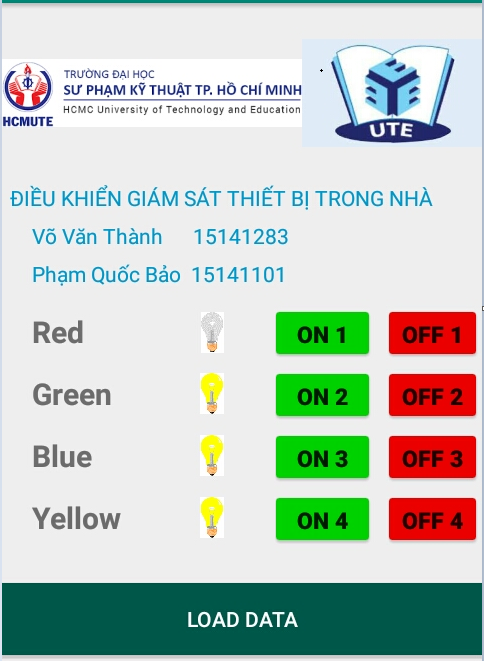
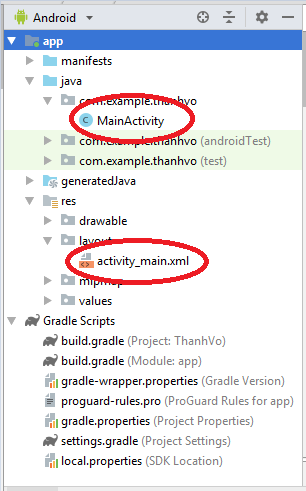
Sau đó xuất hiện 1 cửa sổ mới, của số này chứa đựng các thông tin: tên project ở mục số 1, tên package ở mục số 2, vị trí lưu của project, ngôn ngữ lập trình, mức API và các nền tảng ở mục số 5. Các thông tin này có thể được tùy chỉnh cho phù hợp với project của mình. Cuối cùng nhấn Finish để hoàn tất cài đặt ban đầu và bắt đầu làm việc với project của mình.



Hình 3-: Thiết lập các thông tin ban đầu cho project

#### **4.4.3.3 Ứng dụng Android kết nối với MySQL**

Ứng dụng android được xây dựng để có thể kết nối tới database của MySQL trên server nhằm điều khiển và giám sát hoạt động của các thiết bị. Giao diện của ứng dụng bao gồm cá nút nhấn On/Off để điều khiển các đèn. Các TextView hiển thị tên các thiết bị và các ImageView để hiển thị các trạng thái của thiết bị. Ngoài ra có 1 Button “Load Data” để có thể cập nhật trạng thái các thiết bị.



Hình 3-23: Vị trí 2 file activity\_main.xml và MainActivity.java

Hình 3-24: Giao diện người dùng

Để ứng dụng có thể kết nối tới MySQL chúng ta cần import thư viện hỗ trợ là “mysql-connector” vào project. Sau đó, ứng dụng cần truy cập vào internet để có thể kết nối đến server nên cần cấp quyền truy cập internet cho ứng dụng trong thư mục “AndroidManifest.xml”. Tiếp theo là thiết lập giao diện người dùng trên file “layout/activity\_main.xml” và lập trình ứng dụng trên file “MainActivity.java” như hình 8.

### **4.3.4 Trình biên dịch Keil µVision 5**

#### **4.3.4.1 Giới thiệu trình biên dịch Keil µVision 5**

**Keil µVision 5** là một phần mềm hỗ trợ cho người dùng trong việc lập trình cho vi điều khiển các dòng khác nhau**. Keil µVision 5** giúp người dùng soạn thảo và biên dịch chương trình C hay cả ASM thành ngôn ngữ máy để nạp vào vi điều khiển giúp chúng ta tương tác giữa vi điều khiển và người lập trình.



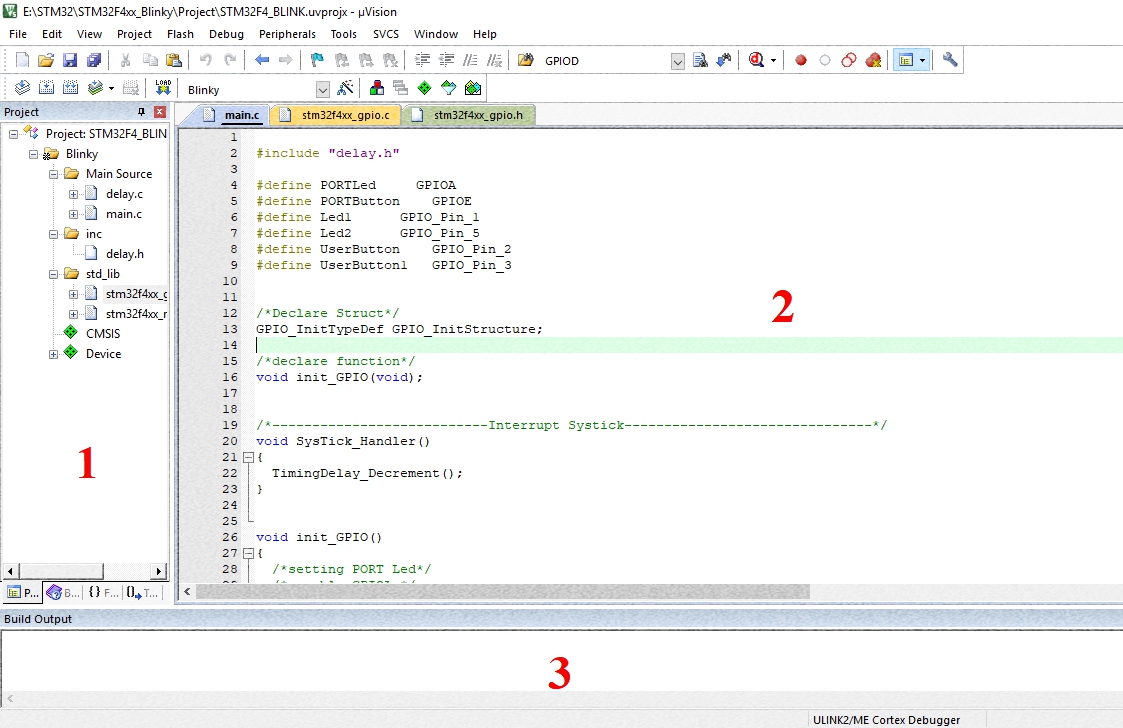
Hình 3-9: Biểu tượng trình biên dịch Keil µVision 5

Màn hình làm việc của Keil C bao gồm các thanh công cụ ở phía trên cùng và 3 vùng chính như trong hình 1.1

Vùng 1: Là vị trí của các cửa sổ Project, cho phép ta quản lý các tập tin trong Project, và cửa sổ Function, cho phép ta quản lý các chương trình con.

Vùng 2: Vùng soạn thảo. Đây là nơi ta sẽ soạn thảo chương trình.

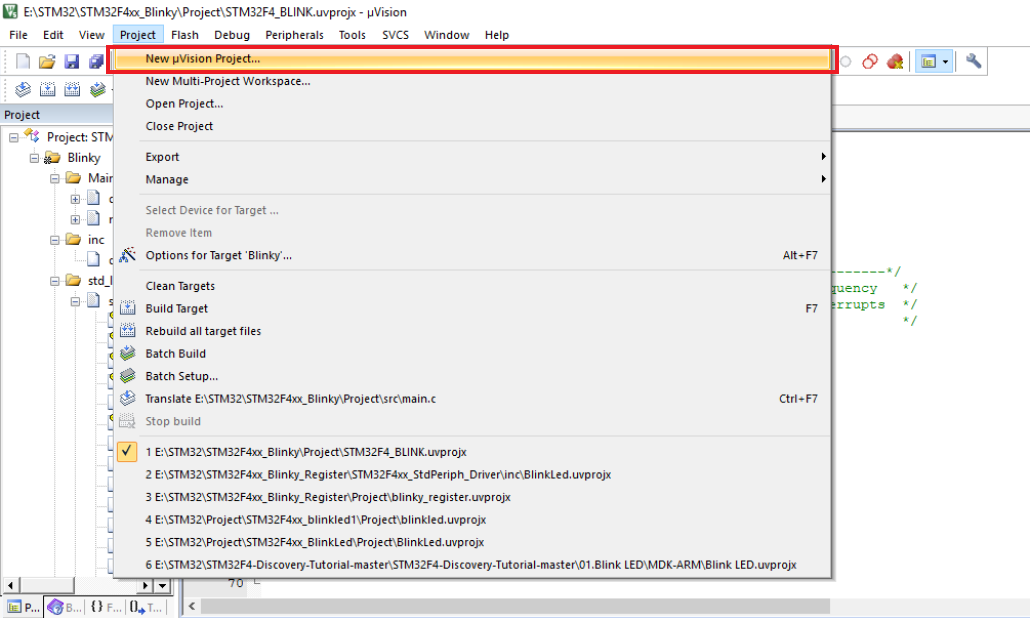
Vùng 3: Cửa sổ Build Output. Hiển thị các thông tin trong quá trình biên dịch chương trình.



Hình 3-10: Giao diện của trình biên dịch Keil µVision 5

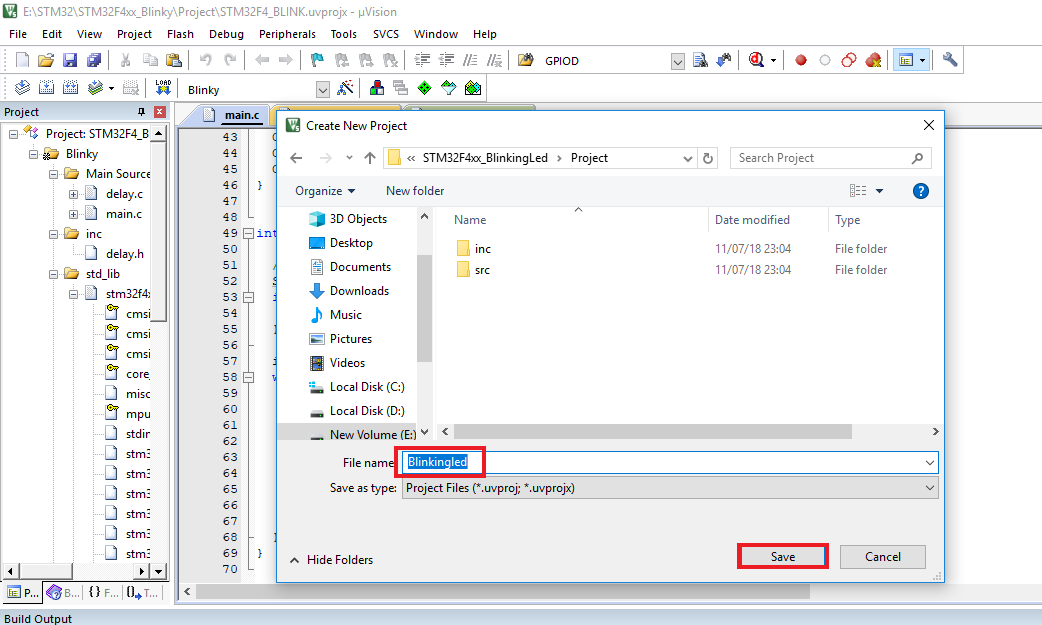
#### **4.3.4.2 CÁC BƯỚC TẠO MỘT PROJECT MỚI VỚI Keil µVision 5**

Để tạo một Project mới, đầu tiên khởi động phần mềm Keil µVision5 🡪 New µVision Project.



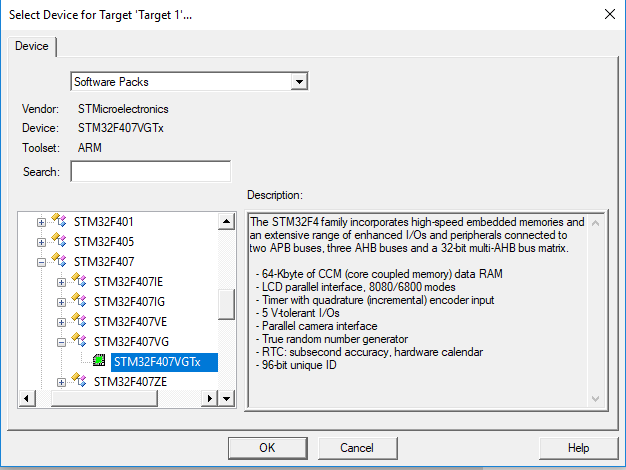
Hình 3-11: Tạo Project mới

Sau đó đặt tên cho Project



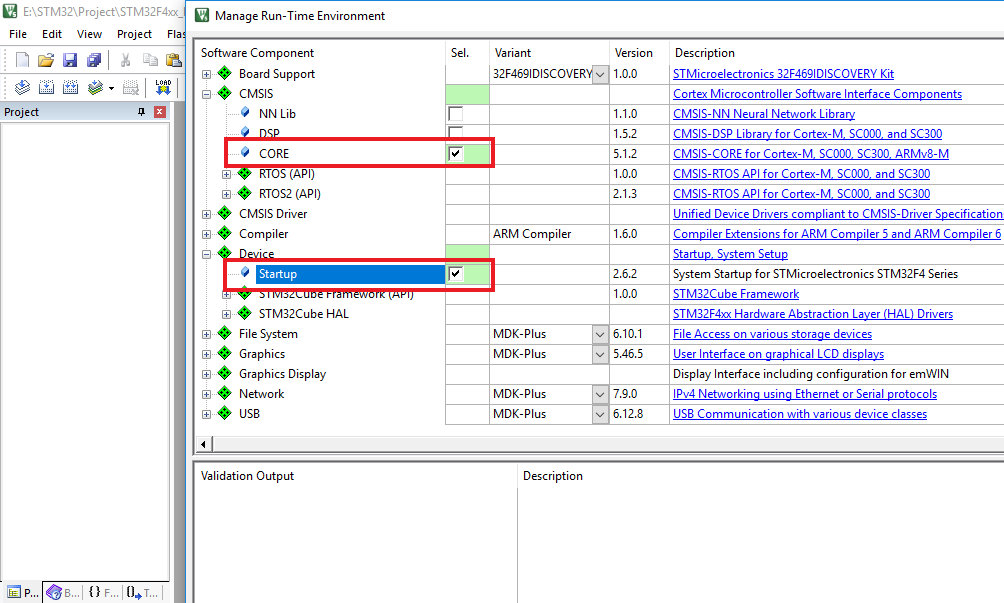
Hình 3-12: Menu để nhập tên cho Project

Và chọn chip chúng ta sử dụng



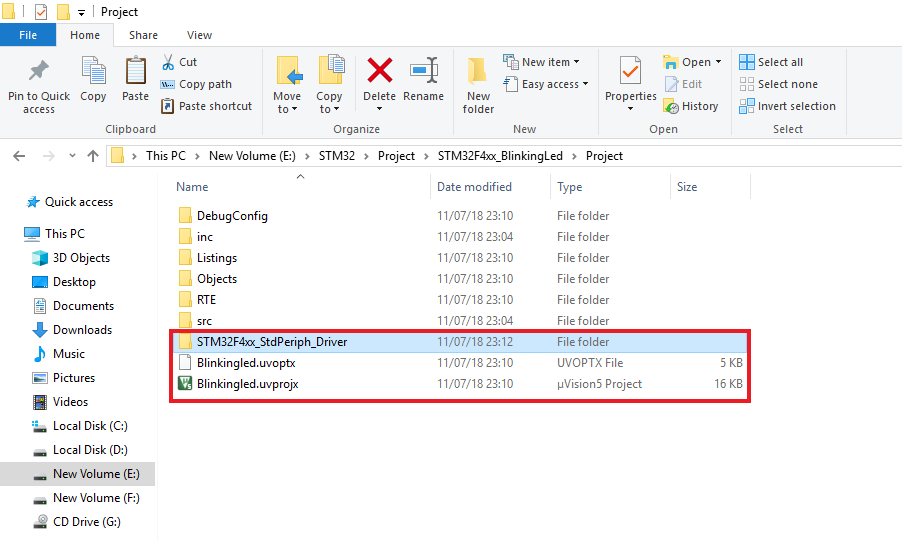
Hình 3-13: Menu dùng để chọn chip STM32F407VG

Tiếp theo, trong bảng Manage Run-Time Enviroment, chúng ta sử dụng thư viện CMCIS kèm theo thư viện Peripheral từ ST.



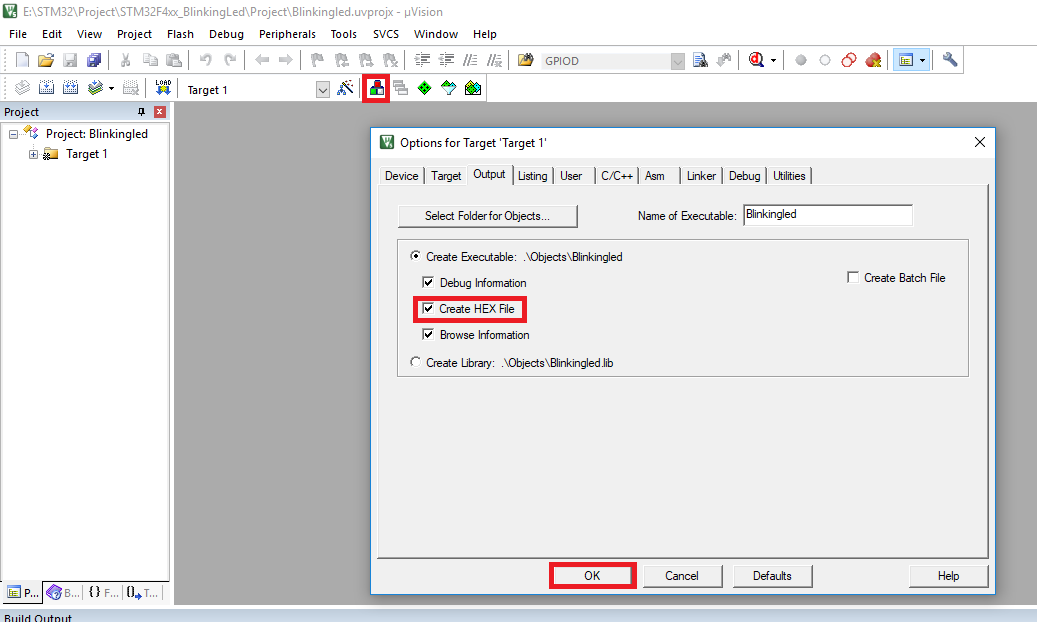
Hình 3-14: Màn hình quản lí môi trường làm việc

Lưu ý: chúng ta phải copy thư viện StdPeriph\_Driver vào chung thư mục với Project vừa tạo.



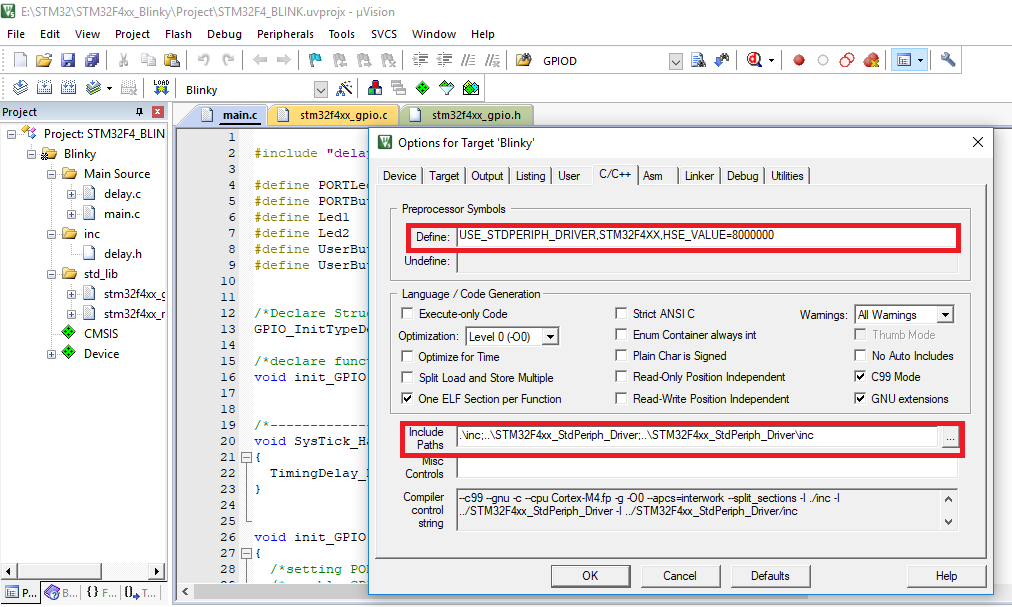
Hình 3-15: Màn hình thư mục chứa Project và thư viện

Tiếp đến, click vào mục Option for Target ‘Target 1’, tại tab Output, đánh dấu vào mục Create File Hex, đó là file mà chúng ta dùng để nạp chương trình.



Hình 3-16: Màn hình lựa chọn xuất file Hex tại tab Output

Chuyển sang tab C/C++, ở đây chúng ta sẽ điền thông tin vào những mục:

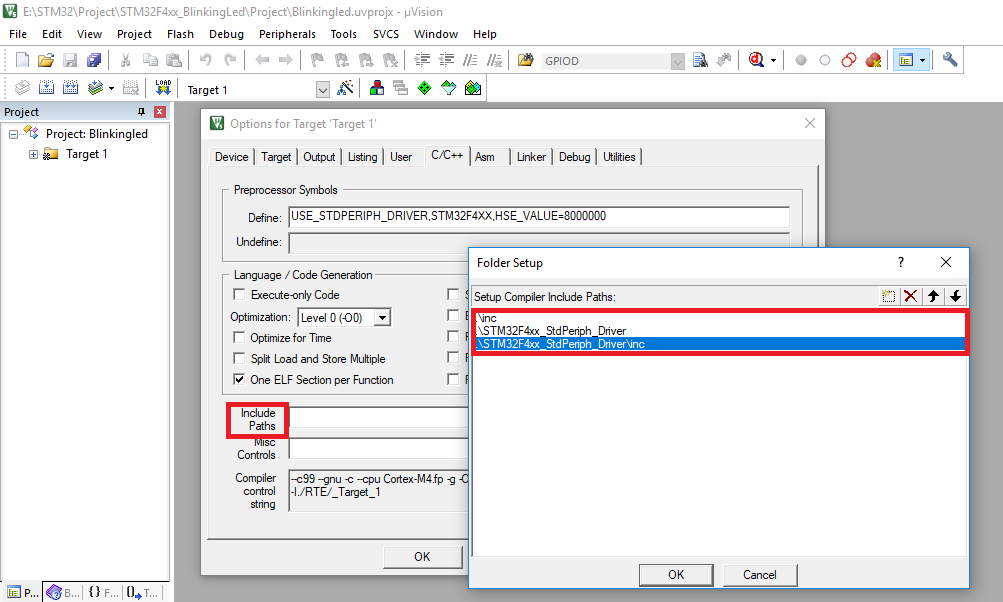


Hình 3-17: Màn hình cấu hình cho tab C/C++

Tại ô Define điền USE\_STDPERIPH\_DRIVER, STM32F4XX, HSE\_VALUE = 8000000

Nghĩa là sử dụng thư viện Peripheral của ST, dòng chip STM32F4, sử dụng thạch anh ngoại 8MHz.

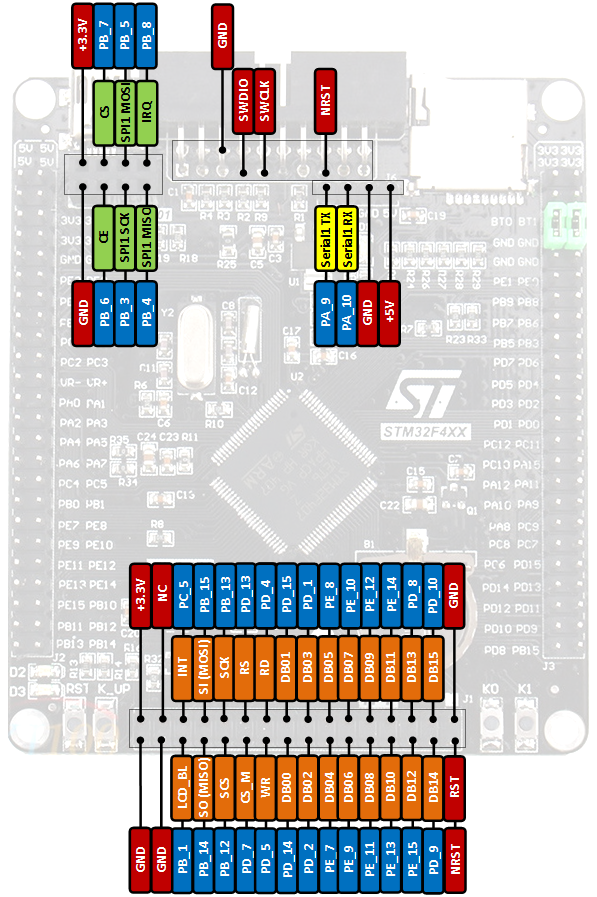
Vẫn trong tab C/C++ chúng ta cần trỏ tất cả đường dẫn tới folder chứa file thư viện cần cho project.



Hình 3-18: Màn hình các đường dẫn chưa thư viện cần cho Project

#### **4.3.4.3 NẠP CHƯƠNG TRÌNH CHO KIT STM32F407VG**

Để nạp code cho Kit, nhóm thực hiện sử dụng mạch nạp ST-Link V2. Mạch nạp ST-Link V2 hổ trợ nạp STM8 và STM32 chuẩn SWD, giao tiếp dễ dàng, nhanh và ổn định. Hỗ trợ phần mềm ST-Link Utility 2.0 hoặc phiên bản cao hơn.



Hình 3-19: Cách kết nối dây sử dụng mạch nạp ST-Link V2



Hình 3-20: Mạch nạp ST-Link V2

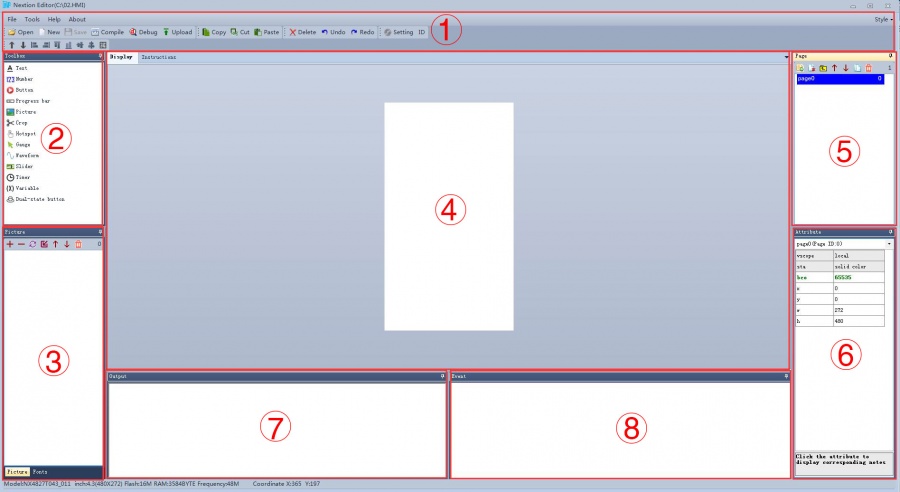
### **4.3.5 Phần mềm viết giao diện màn hình nextion editor**

Nextion Editor là một phần mềm phát triển được sử dụng để xây dựng hình ảnh giao diện đồ họa cho các thiết bị nhúng chuyên sâu - GUI với nhiều loại màn hình TFT và Touch Panels khác nhau. Sử dụng công cụ này, người dùng có thể tạo các thiết bị dựa trên TFT theo cách nhanh hơn và dễ dàng hơn.

Chúng ta có thể tải trực tiếp chương trình này trên trang chủ của Nextion.



Hình 3-21: Biểu tượng phần mềm Nextion Editor



Hình 3-22: Giao diện phần mềm Nextion Editor

**1**. Menu chính

**2**. Thành phần

**3**. Thư viện ảnh: nhập vào những hình ảnh chúng ta cần.

Thư viện phông chữ: lưu trữ phông chữ đã tạo bằng trình chỉnh sửa phông chữ.

**4**. Khu vực hiển thị

**5**. Khu vực trang: dùng đê chỉnh sửa trang

**6**. Khu vực chỉnh sửa thuộc tính: chỉnh sửa các thuộc tính của các thành phần và các trang, chẳng hạn như giá trị, màu sắc của một thành phần văn bản.

**7**. Cửa sổ đầu ra của trình biên dịch: nếu xảy ra lỗi, chúng sẽ được hiển thị trong vùng này.

**8**. Khu vực sự kiện: là nơi để viết và tổng hợp code.

# **CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT – ĐÁNH GIÁ**

# **CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

# **TÀI LIỆU THAM KHẢO**