

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**LUẬN VĂN TỐT NGHIỆP**



**Đề tài**

**CÂN BẰNG TẢI HỆ THỐNG WI-FI BẰNG BỘ ĐIỀU KHIỂN TẬP TRUNG**

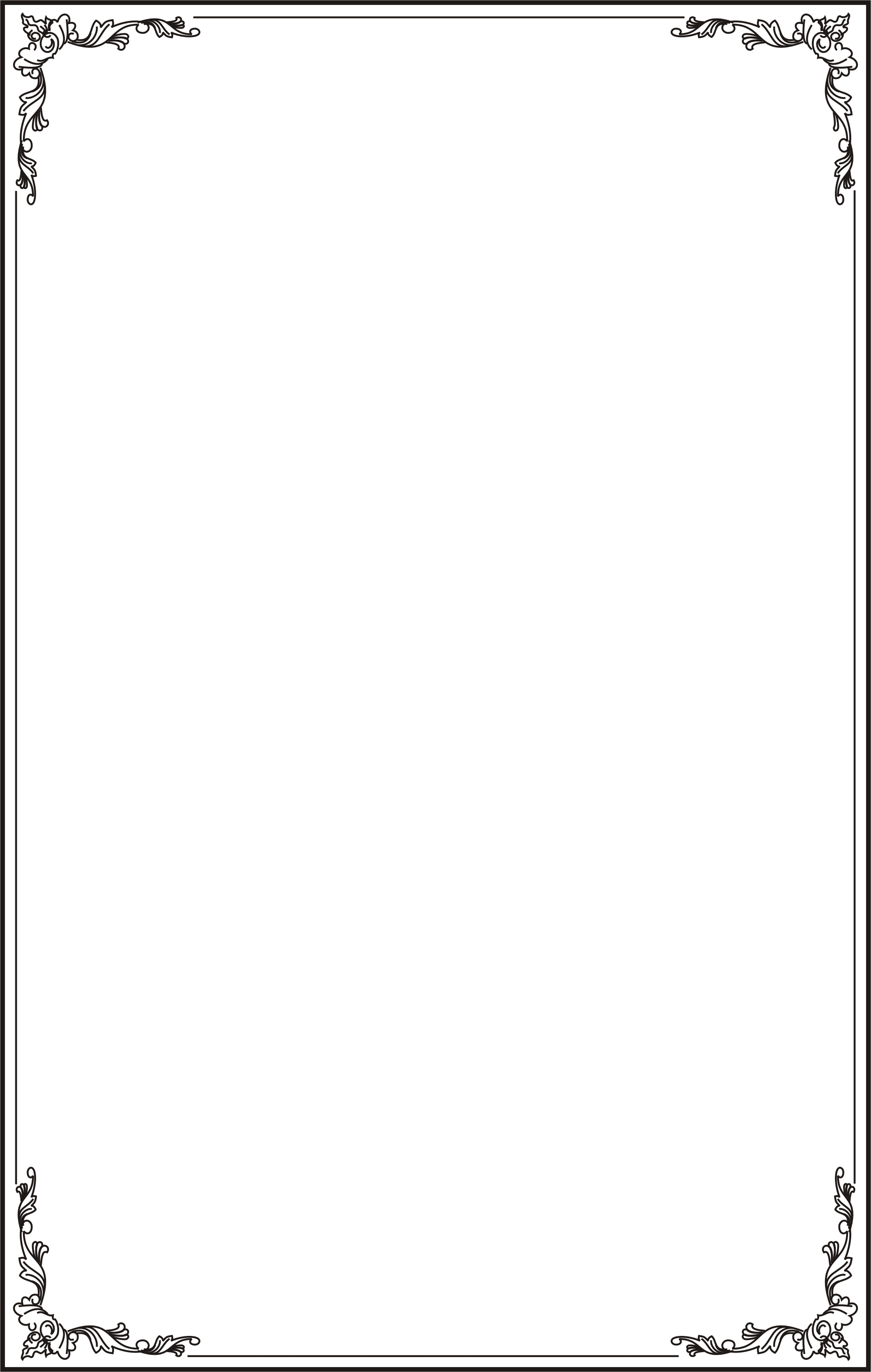
***Sinh viên thực hiện***

***Phạm Quốc Khải - B1401149***

***Giáo viên hướng dẫn***

***• TS. Thái Minh Tuấn***

**Cần Thơ, 12/2018**



**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**LUẬN VĂN TỐT NGHIỆP**



**Đề tài**

**CÂN BẰNG TẢI HỆ THỐNG WI-FI**

**BẰNG BỘ ĐIỀU KHIỂN TẬP TRUNG**

***Sinh viên thực hiện***

***Phạm Tuấn Khải - B1401149***

***Giáo viên hướng dẫn***

***• TS. Thái Minh Tuấn***

**Cần Thơ, 12/2018**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……….Ngày…...tháng.…..năm 2018

Giáo viên hướng dẫn

*(Kí tên)*

# LỜI CẢM ƠN

Để hoàn thành tốt luận văn này, tôi đã nhận được sự giúp đỡ, hỗ trợ và động viên của rất nhiều cá nhân cũng như tập thể. Nhân đây, chúng tôi xin tỏ lòng biết ơn đến gia đình đã quan tâm, ủng hộ về vật chất lẫn tinh thần trong quá trình tôi học tập và thực hiện luận văn.

Xin chân thành biết ơn sâu sắc đến thầy Thái Minh Tuấn đã rất tận tình hướng dẫn, góp ý, truyền đạt những kiến thức và kinh nghiệm quý báu để tôi hoàn thành tốt luận văn này.

Chân thành cảm ơn quý thầy cô bộ môn Công nghệ thông tin - Đại học Cần Thơ đã tận tình chỉ bảo, giúp đỡ trong thời gian tôi thực hiện luận văn. Cuối cùng chúng tôi tin cảm ơn bạn bè, anh chị khoa Công nghệ thông tin và Truyền thông đã cho chúng tôi ý tưởng, hỗ trợ, động viên tôi trong quá trình thực hiện luận văn. Mặc dù cố gắng hoàn thiện luận văn này nhưng do kiến thức còn hạn chế nên không thể tránh những sai sót, vì thế chúng tôi mong nhận được những góp ý của thầy cô và các bạn.

**PHẠM QUỐC KHẢI**

# 

# MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN i](#_Toc2434)

[LỜI CẢM ƠN ii](#_Toc21194)

[MỤC LỤC iii](#_Toc12265)

[DANH SÁCH CÁC TỪ VIẾT TẮT v](#_Toc15133)

[DANH SÁCH HÌNH vi](#_Toc2426)

[DANH SÁCH BẢNG vii](#_Toc26563)

[TÓM TẮT viii](#_Toc11924)

[ABSTRACT ix](#_Toc5838)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc13488)

[1.1 ĐỘNG CƠ NGHIÊN CỨU 1](#_Toc31429)

[1.2 MỤC TIÊU 1](#_Toc7944)

[1.3 NHỮNG NGHIÊN CỨU LIÊN QUAN 1](#_Toc3495)

[1.4 PHƯƠNG PHÁP NGHIÊN CỨU 2](#_Toc12997)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc9962)

[2.1 MẠNG WI-FI 3](#_Toc15533)

[2.1.1 Mạng Wi-Fi là gì? 3](#_Toc3776)

[2.1.2 Nguyên tắc hoạt động của mạng Wi-Fi 4](#_Toc24743)

[2.1.3 Roaming trong mạng Wi-Fi 4](#_Toc19459)

[2.2 FIRMWARE OPENWRT 5](#_Toc31338)

[2.2.1 Firmware OpenWrt là gì? 5](#_Toc8880)

[2.2.2 Những tính năng của OpenWrt 6](#_Toc16541)

[2.3 TFTP 10](#_Toc20230)

[2.4 TIẾN TRÌNH KHỞI ĐỘNG ROUTER 10](#_Toc9709)

[2.5 GIỚI THIỆU CÔNG NGHỆ SDN 10](#_Toc10942)

[2.5.1 SDN là gì? 10](#_Toc4073)

[2.5.2 Giao thức OpenFlow 11](#_Toc5377)

[2.6 CÔNG NGHỆ WEB 11](#_Toc31136)

[CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU 15](#_Toc31487)

[3.1 MÔ TẢ CHẾ ĐỘ HOẠT ĐỘNG 15](#_Toc2137)

[3.2 YÊU CẦU KỸ THUẬT 15](#_Toc3675)

[3.3 THIẾT KẾ TƯƠNG TÁC 16](#_Toc2777)

[3.4 THIẾT KẾ GIẢI THUẬT 17](#_Toc24511)

[3.5 LẮP MÔ HÌNH 18](#_Toc24601)

[3.6 KIỂM THỬ 20](#_Toc1884)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 21](#_Toc26706)

[4.1 KẾT QUẢ ĐẠT ĐƯỢC 21](#_Toc4642)

[4.2 HƯỚNG PHÁT TRIỂN 21](#_Toc31691)

[PHỤ LỤC 1 22](#_Toc25071)

[PHỤ LỤC 2 26](#_Toc31960)

[TÀI LIỆU THAM KHẢO 29](#_Toc24587)

# DANH SÁCH CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ viết đầy đủ |
| 1 | COM | Communication |
| 2 | DS | Data serial |
| 3 | E | Enable |
| 4 | GND | Ground |
| 5 | I2C | Inter-integrated Circuit |
| 6 | IC | Integrated Circuit |
| 7 | IDE | Integrated Development Environment |
| 8 | IoT | Internet of Things |
| 9 | LCD | Liquid Crystal Display |
| 10 | LED | Light Emitting Diode |
| 11 | MR | Master Reclear |
| 12 | OE | Output enable |
| 13 | RAM | Random Access Memory |
| 14 | RS | Register Select |
| 15 | RW | Read/Write |
| 16 | SCL | Serial Clock |
| 17 | SDA | Serial Data |
| 18 | SHCP | Shift register clock pin |
| 19 | STCP | Storage register clock pin |
| 20 | VE | Constrast Voltage |

# DANH SÁCH HÌNH

[Hình 1. Vi điều khiển 2](#_Toc530323041)

[Hình 2. Vị trí các chân vào/ra trên Arduino Uno R3 3](#_Toc696342875)

[Hình 3. Cổng giao tiếp và nút Reset trên Arduino Uno R3 4](#_Toc803028864)

[Hình 4. Cấu tạo đèn LED 5](#_Toc597718683)

[Hình 5. Sơ đồ cấu tạo LED 7 đoạn 5](#_Toc719177390)

[Hình 6. Các chân IC 74HC595 6](#_Toc161981184)

[Hình 7. Sơ đồ bàn phím 4x4 8](#_Toc494323212)

[Hình 8. Màn hình LCD 16x2 8](#_Toc1916427093)

[Hình 9. Module I2C 10](#_Toc1457749224)

[Hình 10. Đường I2C truyền tín hiệu. 10](#_Toc687556408)

[Hình 11. Giao diện chính của Arduino IDE 11](#_Toc823014825)

[Hình 12. Sử dụng thư viện Keypad. 12](#_Toc6484751)

[Hình 13. Sử dụng thư viện LiquidCrystal\_I2C. 13](#_Toc953982140)

[Hình 14. Quá trình thực thi chương trình trong Arduino. 14](#_Toc767880226)

[Hình 15. Sơ đồ tương tác của người dùng và mô hình 16](#_Toc670945539)

[Hình 16. Giải thuật chọn chế độ. 17](#_Toc128839418)

[Hình 17. Giải thuật vận hành chế độ ban ngày 18](#_Toc2040063492)

[Hình 18. Mô hình đèn giao thông. 20](#_Toc1406825293)

[Hình 19: Giao diện tải IDE dành cho Arduino 22](#_Toc11097061)

[Hình 20: Giao diện chỉ tải Arduino IDE về hoặc đóng góp kinh phí 22](#_Toc1728912840)

[Hình 21: Chấp nhận điều khảng cài đặt IDE 23](#_Toc1832746430)

[Hình 22: Chọn phần cài đặt của gói phần mềm 23](#_Toc221154673)

[Hình 23: Chọn thư mục cài đặt IDE 24](#_Toc8262338)

[Hình 24: Hoàn thành cài đặt IDE 24](#_Toc2100835573)

[Hình 25: Giao diện Arduino IDE 25](#_Toc1427094991)

[Hình 26. Kết nối Arduino với máy tính 26](#_Toc2135818359)

[Hình 27. Viết mã lệnh cho chương trình 27](#_Toc1940413734)

[Hình 28. Chọn bo mạch 27](#_Toc829878453)

[Hình 29. Chọn cổng giao tiếp 28](#_Toc1193786090)

[Hình 30. Nạp chương trình cho Arduino 28](#_Toc1503549760)

# DANH SÁCH BẢNG

[Bảng 1: Ý nghĩa các chân IC 74HC595 7](#_Toc2030665099)

[Bảng 2. Các chân của LCD 16x2 9](#_Toc2011741105)

# TÓM TẮT

Mạng không dây Wi-Fi đang là loại mạng phổ biến đối với người dùng hiện nay. Đối với các doanh nghiệp hay cá nhân thì việc triển khai mạng không dây là vô cùng quan trọng bởi vì nó cung cấp tiện ích rất lớn đó là kết nối các thiết bị di động mà không cần sử dụng cáp nối nhờ vào các điểm truy cập không dây. Tuy nhiên, đối với các điểm truy cập đông người thì nó lại phát sinh những khó khăn vì lượng lớn người truy cập gây ra sự quá tải ở điểm truy cập.

Trên thực tế, hệ thống cân bằng tải cho Wi-Fi đã có nhưng phải đầu tư với chi phí khá cao nên ở luận văn này, tôi muốn hướng đến việc phát triển một hệ thống Wi-Fi cân bằng nhưng với một chi phí hợp lí.

Chúng tôi đưa ra những thuật toán, những thiết bị để hoàn thành tốt dự án được đưa ra, góp phần nghiên cứu để cải thiện vào hệ thống mạng Wi-Fi được tốt hơn.

Từ khóa: *Cân bằng tải. Wi-fi.*

# ABSTRACT

Wi-Fi is becoming more common among today's users. For businesses or individuals, deploying a wireless network is extremely important because it provides a large utility that connects mobile devices without the need for a cable connection using the access points. However, for crowded places, it is difficult because of the large number of clients causes overload at the access point.

In fact, the load balancing system for Wi-Fi is available, but it costs a lot to invest. In this thesis, I want to develop a balanced Wi-Fi system but at a reasonable cost.

We provide algorithms, devices to successfully complete the proposed project, contributing to improved research into the Wi-Fi network.  
Keywords: *load balancing. WIFI.*

# CHƯƠNG 1: TỔNG QUAN

## **ĐỘNG CƠ NGHIÊN CỨU**

Wi-Fi đã trở thành thiết bị quan trọng hàng đầu trong việc sử dụng mạng internet. Wi-Fi được cài đặt ở hầu hết các địa điểm công cộng và ở cà nhà riêng nó đang phát triển với một tốc độ nhanh chóng. Sử dụng Wi-Fi mang lại sự thuận tiện dễ dàng cho mọi người trong việc truy cập internet tuy nhiên việc sự dụng mạng Wi-Fi vào các điểm truy cập công cộng có quy mô lớn như khuôn viên trường đại học, quán cafe hay các công ty, xí nghiệp, siêu thị... đang gặp một số vấn đề nghiêm trọng cần giải quyết như:

Nhiễu kênh truyền: Các router Wi-Fi do không “biết” được kênh hoạt động của nhau nên thường xuyên xảy ra tình trạng các router khác nhau hoạt động trên cùng một kênh truyền làm ảnh hưởng nghiêm trọng đến hiệu suất và thông lượng của Wi-Fi.

Tải Wi-Fi không cân bằng: Thông thường đa số các thiết bị sử dụng Wi-Fi có xu hướng kết nối với router Wi-Fi đầu tiên, do các router có sóng khá mạnh nên sau khi kết nối, dù có đi khá xa các thiết bị vẫn không ngừng kết nối với router đầu tiên. Điều này dẫn đến một số Wi-Fi sẽ quá tải do số lượng nối kết quá nhiều còn một số khác thì dư tải và ít nối kết việc này ảnh hưởng đến trải nghiệm truy cập của mọi người và băng thông chung của hệ thống.

Để giải quyết các vấn đề trên đã có một số nghiên cứu cân bằng tải cho Wi-Fi cũng như có một số nhà sản xuất tạo ra các router cân bằng tải dành riêng cho các điểm truy cập Wi-Fi quy mô lớn tuy nhiên thông thường giá của các router này khá đắt đỏ.

Vì các lý do kể trên nhóm quyết định thực hiện đề tài “cân bằng tải Wi-Fi bằng hệ thống điều khiển tập trung” sử dụng các tiện ích trên firmware OpenWrt cũng như bộ điều khiển để tạo ra một mạng Wi-Fi có thể điều khiển bằng phần mềm, có thể “giao tiếp được với nhau” và có giá cả hợp lý.

### **MỤC TIÊU**

Xây dựng hệ thống Wi-Fi giá rẻ có khả năng điều khiển bằng bộ điều khiển tập trung. Đồng thời dựng trang web theo dõi hiệu năng, danh sách các thiết bị nối kết của router.

Đề xuất một giải thuật để giải quyết các vấn đề tải không cân bằng nêu trên để đảm bảo được thông lượng chung của hệ thống và trải nghiệm mạng một cách tốt nhất của người dùng.

### **NHỮNG NGHIÊN CỨU LIÊN QUAN**

Dựa theo bài báo khoa học tự đề “SAMF: An SDN-Based Framework for Access Point Management in Large-scale Wi-Fi Networks”[1] nghiên cứu về việc quản trị mạng Wi-Fi mạng cở lớn.

### **PHƯƠNG PHÁP NGHIÊN CỨU**

Đề tài là một phạm vi khá rộng và có những hướng đi phát triển tốt trong tương lai. Người thực hiện cần nắm kiến thức cơ bản về mạng không dây, xử lý ngôn ngữ lập trình cũng như những vấn đề xấu phát sinh mắc phải, không những thế người thực hiện cần hiểu rõ những khái niệm, những đặc điểm của cái thiết bị mạng không dây,… Để hoàn thành tốt việc nghiên cứu đề tài, phương pháp được chia ra như sau:

* Nghiên cứu về firmware OpenWrt, các API và các gói tiện ích trên OpenWrt.
* Nghiên cứu về kiến trúc Restful, cách sử dụng truy vấn.
* Xây dựng nhánh mạng
* Nghiên cứu công nghệ SDN

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## MẠNG WI-FI

### Mạng Wi-Fi là gì?

Wi-Fi viết tắt từ Wireless Fidelity hay mạng 802.11 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống như điện thoại di động, truyền hình và radio. Hầu hết các thiết bị điện tử ngày nay như điện thoại di động, laptop… đều có thể kết nối Wi-Fi[2]

Hệ thống Wi-Fi hiện nay đang trở nên rất phổ biến ở các sân bay, quán cafe, thư viện khách sạn. Hệ thông cho phép người sử dụng truy cập internet ở các khu vực có sóng và hoàn toàn không cần dùng đến cáp nối. Ngoài các điểm kết nối hostposts Wi-Fi hiện nay có thể thiết lập tại nhà riêng tạo sự thuận tiện dễ dàng cho người sử dụng.

Hệ thống Wi-Fi truyền và phát tín hiệu ở tần số 2.4 GHz, 5 GHz và 60Ghz . Tần số này cao hơn so với các tần số sử dụng cho điện thoại di động, các thiết bị cầm tay và truyền hình. Tần số cao hơn cho phép tín hiệu mang theo nhiều dữ liệu hơn Hiện nay hệ thống mạng Wi-Fi đang có 5 chuẩn thông dụng là a,b,g,n,ac mỗi chuẩn hoạt động ở một băng tần nhất định và hỗ trợ tốc độ riêng.

**Chuẩn 802.11**:Năm 1997, IEEE đã giới thiệu chuẩn mạng không dây đầu tiên và đặt tên nó là 802.11. Tuy nhiên chuẩn 802.11 chỉ hỗ trợ tốc độ mạng tối đa 2 Mbps với băng tần 2.4 GHz, rất chậm so với ngày nay và không được áp dụng rộng rãi trên thị trường

**Chuẩn 802.11b**: Tháng 7 năm 1999, chuẩn 802.11b ra đời và hỗ trợ tốc độ lên đến 11Mpbs thay vì 2Mbps như trước kia. Tương tự thế hệ đầu tiên, chuẩn kết nối 802.11b cũng sử dụng băng tần 2.4 GHz rất dễ bị gây nhiễu từ các thiết bị điện tử khác như điện thoại di động, lò vi sóng,...

**Chuẩn 802.11a**: Được phát triển song song với chuẩn 802.11b, tuy nhiên chuẩn a thường được sử dụng trong các mạng của doanh nghiệp thay vì gia đình như chuẩn b vì giá thành khá cao. So với chuẩn 802.11b, chuẩn này hỗ trợ tốc độ tối đa gần gấp 5 lần, lên đến 54 Mpbs và sử dụng băng tần vô tuyến 5 GHz có thể tránh tình trạng bị nhiễu do các thiết bị khác. Tuy nhiên do tần số cao hơn nên phạm vi hoạt động của chuẩn 802.11a có phần hẹp hơn (40-100m) và khó xuyên qua các vật cản, vách tường.

**Chuẩn 802.11g**: Năm 2003, chuẩn Wi-Fi thế hệ thứ 3 ra đời được đặt là chuẩn 802.11g, chuẩn Wi-Fi này thậm chí còn được sử dụng ở nhiều mạng Wi-Fi các gia đình hiện nay. Chuẩn 802.11g được xem là kết hợp giữa chuẩn a và b trước kia, với giá thành khá rẻ (tuy có phần đắt hơn chuẩn b). Chuẩn 802.11g hỗ trợ tốc độ đến 54 Mpbs như chuẩn a nhưng sử dụng băng tần 2.4 GHz như chuẩn b, vì vậy chuẩn này có tốc độ cao, phạm vi tín hiệu tốt (80-200m).Và tất nhiên chuẩn này cũng có nhược điểm như chuẩn b là dễ bị nhiễu từ các thiết bị phát sóng khác. Do sự giống nhau về nhiều thông số, chuẩn kết nối 802.11g có khả năng tương thích ngược với chuẩn 802.11b và ngược lại.

**Chuẩn 802.11n**: Đây là chuẩn tương đối mới và đang sử dụng khá phổ biến hiện nay. Chuẩn Wi-Fi 802.11n được đưa ra nhằm cải thiện chuẩn 802.11g bằng cách sử dụng công nghệ MIMO (Multiple-Input Multiple-Output) tận dụng nhiều anten hơn. Chuẩn kết nối 802.11n hỗ trợ tốc độ tối đa lên đến 600 Mpbs, có thể hoạt động trên cả băng tần 2,4 GHz và 5 GHz, nếu router hỗ trợ thì hai băng tần này có thể cùng phát sóng song song.Chuẩn kết nối này đã và đang dần thay thế chuẩn 802.11g với tốc độ cao, phạm vi tín hiệu rất tốt (từ 100-250m) và giá thành đang ngày càng phù hợp với túi tiền người tiêu dùng.

**Chuẩn 802.11ac**: Chuẩn 802.11ac là chuẩn mới Wi-Fi nhất của IEEE đã được tung ra thị trường, áp dụng công nghệ đa anten đã có trên chuẩn 802.11n, với băng tần 5 GHz và tốc độ tối đa lên đến 1,3 Gigabit/giây người dùng sẽ trải nghiệm tốc độ mạng ở mức cao nhất.

**Chuẩn 802.11ad**: chuẩn 802.11ad phát ở tần số 60 GHz nhanh hơn so với chuẩn 802.11ac, tốc độ truyền dữ liệu tối đa đạt đến 4,6 Gigabit/giây

### Nguyên tắc hoạt động của mạng Wi-Fi

Bộ chuyển tín hiệu không dây (adapter) của thiết bị sử dụng mạng không dây sẽ chuyển đổi dữ liệu của nó sang tín hiệu vô tuyến và phát tín hiệu này đi bằng một ăng-ten.

Thiết bị Router Wi-Fi nhận những tín hiệu không dây của thiết bị sử dụng Wi-Fi gửi và giải mã chúng sau đó gửi qua mạng bằng các kết nối có dây (ethernet).

Hai quá trình trên sẽ hoạt động với chiều ngược lại để truyền tín hiệu từ mạng xuống thiết bị sử dụng Wi-Fi.[3].

### Roaming trong mạng Wi-Fi

Roaming là quá trình xử lý, đảm bảo kết nối Wi-Fi của client khi di chuyển từ AP này sang AP khác. Khi ta di chuyển giữa các AP, kết nối phải được thiết lập trên AP mới. Vì vậy bất kỳ dữ liệu nào của ta đang gửi trước khi roaming sẽ được gửi lại từ AP cũ đến AP mới. Điều này giúp việc gửi nhận dữ liệu trong và sau roaming không bị mất. Khoảng thời gian trì hoãn là không đáng kể.

Để có thể tiến hành roaming, các AP cần phải cùng chuẩn, SSID, chế độ mã hóa/ xác thực nhưng không được trùng kênh. Đầu tiên, router Wi-Fi thứ nhất phải được cấu hình không trùng kênh với router Wi-Fi thứ hai. Điều này giúp máy tính của ta phân biệt được tín hiệu của router Wi-Fi thứ nhất và router Wi-Fi thứ hai. Chẳng hạn, 2 router Wi-Fi cùng hỗ trợ chuẩn 802.11g, ta có thể cấu hình router Wi-Fi thứ nhất sử dụng kênh số 1 và router Wi-Fi thứ hai sử dụng kênh 6.

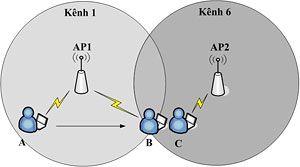
Việc xử lý roaming sẽ do trình điều khiển card Wi-Fi của client đảm nhận (AP không tham gia vào quá trình này). Client có thể thực hiện một trong hai cách để chuẩn bị tiến hành roaming:

* Cách 1: khi client nhận thấy tín hiệu từ router giảm đi, nó sẽ thực hiện tìm kiếm router khác gần nhất.
* Cách 2: khi client di chuyển ra cận vùng phủ sóng router, tín hiệu yếu, không đảm bảo kết nối, nó sẽ tiến hành tìm kiếm router khác gần nhất.

Trình điều khiển card Wi-Fi của client sẽ quyết định lúc nào cần thực hiện roaming. Giải thuật roaming là các giải thuật bí mật của nhà sản xuất card Wi-Fi, ta không thể nhìn thấy thời điểm chính xác khi nào diễn ra tiến trình roaming. Giải thuật roaming thường căn cứ vào cường độ tín hiệu, chất lượng tín hiệu...

Với mỗi thiết bị di động khác nhau sẽ có thời điểm roaming khác nhau. Một số thì tiến hành roaming ngay khi vào vùng tín hiệu mới; một số thì lại roaming khi nhận thấy tín hiệu AP mới tốt hơn. Khi client quyết định tiến hành roaming thì bước tiếp theo là nó phải tìm kiếm, dò kênh AP mới. Có 02 cách:

* Dò tìm thụ động: client vừa dò kênh, vừa lắng nghe báo hiệu 802.11 từ router mới.
* Dò tìm chủ động: client vừa dò kênh, vừa gửi frame thăm dò 802.11 để tìm kiếm router mới.

Khi client dò tìm thụ động, nó chỉ chờ nhận báo hiệu từ AP. Dò tìm chủ động, client phải gửi thăm dò và chờ nhận tín hiệu trả lời của AP. Dò tìm chủ động sẽ hiệu quả hơn vì client trực tiếp tìm kiếm AP, giúp rút ngắn thời gian trì hoãn việc gửi nhận dữ liệu.

Quá trình roaming.

Vì client không thể kết nối với 2 router cùng lúc nên sau khi tiến hành roaming, dò tìm kênh, client sẽ ngắt kết nối với router thứ nhất và chuyển qua với router thứ hai. Kết nối thành công, client sẽ sử dụng router thứ hai làm điểm truy cập mới. [2]

## FIRMWARE OPENWRT

### Firmware OpenWrt là gì?

OpenWrt là một hệ điều hành nhúng dựa trên nhân Linux, và thường được nhúng trên các thiết bị nhúng vào mạng lưới định tuyến đường truyền. Thành phần chính của OpenWrt là nhân Linux, util-linux, uClibc và BusyBox. Các thành phần của OpenWrt đã được tối ưu kích thước để có thể nhúng vào bộ nhớ có hạn của thiết bị router dùng trong gia đình. Vì được xây dựng dựa trên nhân Linux nên OpenWrt có thể biến một router của nhà cung cấp với ít khả năng thao tác của người dùng trở nên linh hoạt hơn trong việc quản lý và điều khiển

OpenWrt được cấu hình bằng cách sử dụng giao diện dòng lệnh (ash Shell), hoặc một giao diện Web ([Luci](http://wiki.openwrt.org/doc/techref/luci)). Có khoảng 3.500 gói phần mềm tùy chọn có sẵn để cài đặt qua hệ thống quản lý gói [opkg.](https://vi.wikipedia.org/w/index.php?title=Opkg.&action=edit&redlink=1" \o "Opkg. (trang chưa được viết))

OpenWrt có thể chạy trên các bộ định tuyến CPE, hệ thống mạng nhỏ, điện thoại thông minh (ví dụ: Neo FreeRunner), máy tính bỏ túi (ví dụ: Ben NanoNote), thậm chí cả máy tính xách tay (ví dụ: One Laptop per Child (OLPC)). Ngoài ra, OpenWrt có thể chạy trên các máy tính thông thường (ví dụ như kiến trúc [x86).](https://vi.wikipedia.org/wiki/X86" \o "X86) Nhiều bản vá lỗi từ các mã nguồn cơ sở của OpenWrt đã được đưa và đóng góp ngược lại cho nhân Linux bằng đường chính.

### Những tính năng của OpenWrt

OpenWrt có sự mở rộng rất linh hoạt và đa dạng tính năng. Các tính năng chính:

* OpenWrt cũng cấp một hệ thống tập tin gốc cho phép người dùng có thể thêm, xóa hoặc sửa đổi bất kỳ tập tin. Điều này có thể thực hiện bằng việc sử dụng overlayfs có thệ thống tập tin chỉ đọc được nén với SquashFS và JFFS2 có thể ghi Copy-on-Write.
* Quản lý gói opkg, cho phép người dùng cài đặt và gỡ bỏ phần mềm. Trong các kho có chứa khoảng 3500 gói.
* Một tập hợp các mã gọi là UCI (giao diện cấu hình Unified) Dự kiến để thống nhất và đơn giản hóa cấu hình của toàn bộ hệ thống[[22]](https://vi.wikipedia.org/wiki/OpenWrt" \l "cite_note-22)
* Cấu hình mở rộng của toàn bộ trình điều khiển phần cứng, ví dụ như xây dựng tích hợp switch mạng và [VLAN,](https://vi.wikipedia.org/wiki/Virtual_LAN" \o "Virtual LAN) WNIC s, modem DSL s, FX, các nút phần cứng có sẵn, vv
* Khả năng toàn diện để cấu hình các tính năng mạng như:
* Định tuyến thông qua iproute2, Quagga, BIRD, vv
* Chức năng [không dây,](https://vi.wikipedia.org/w/index.php?title=Kh%C3%B4ng_d%C3%A2y,&action=edit&redlink=1" \o "Không dây, (trang chưa được viết)) ví dụ như làm những hành động thiết bị như một repeater không dây, một điểm truy cập không dây, một cầu nối không dây, một cổng khóa,…
* Bảo mật không dây: Packet injection, ví dụ: [Airpwn](http://airpwn.sourceforge.net/Airpwn.html), lorcon,…
* Tường lửa, NAT và port forwarding thông qua netfilter; PeerGuardian có sẵn
* Tự động được cấu hình cổng giao thức chuyển tiếp.
* IP tunneling
* Domain Name System (DNS) và [DHCP](https://vi.wikipedia.org/wiki/DHCP" \o "DHCP) qua dnsmasq, MaraDNS, vv
* Dịch vụ DNS động để duy trì một tên miền cố định với một ISP không cung cấp một địa chỉ IP tĩnh
* hệ thống phân phối không dây (WDS) bao gồm WPA-PSK, WPA2-PSK, WPA-PSK / WPA2-PSK Mixed-Mode chế độ mã hóa
* OpenWrt hỗ trợ bất kỳ phần cứng có hỗ trợ Linux như máy in, webcam, card âm thanh
* Mở rộng một giao diện Web [Ajax,](https://vi.wikipedia.org/w/index.php?title=Ajax,&action=edit&redlink=1" \o "Ajax, (trang chưa được viết)) nhờ dự án Luci
* Sửa lỗi và cập nhật thường xuyên, ngay cả đối với các thiết bị không còn được hỗ trợ bởi các nhà sản xuất của họ.

**Dự án Luci**

Luci được tạo ra vào tháng 3 năm 2018 với tên gọi là FFLuci với mục tiêu là tạo ra giao diện web thân thiện, có thể mở rộng và bảo trì cho các thiết bị nhúng sử dụng firmware OpenWrt. Trong khi các giao diện cấu hình khác sử dụng những ngôn ngữ kịch bản nặng nề thì Luci sử dụng ngôn ngữ lập trình là lua và chia giao diện thành các phần logic (model và view) sử dụng các thư viện hướng đối tượng để đảm bảo hiệu suất cao hơn, kích thước cài đặt nhỏ hơn, thời gian chạy nhanh hơn và khả năng bảo trì tốt hơn.

**JSON RPC API**

JSON-RPC hoạt động bằng cách gửi yêu cầu tới máy chủ triển khai giao thức này. Các client trong trường hợp này thường là phần mềm có ý định gọi một phương pháp duy nhất của một hệ thống từ xa. Nhiều tham số đầu vào có thể được chuyển tới phương thức từ xa như một mảng hoặc đối tượng, trong khi chính phương thức đó cũng có thể trả về nhiều dữ liệu đầu ra. (Điều này tùy thuộc vào phiên bản được triển khai.)

Tất cả các kiểu truyền là các đối tượng đơn lẻ, được tuần tự hóa bằng JSON. Một yêu cầu là một cuộc gọi đến một phương pháp cụ thể được cung cấp bởi một hệ thống từ xa. Nó phải chứa ba thuộc tính nhất định:

* method - Một chuỗi với tên của phương thức được gọi.
* params - Một đối tượng hoặc mảng của các giá trị được chuyển thành các tham số cho phương thức đã định nghĩa.
* id - Giá trị của bất kỳ loại nào được sử dụng để khớp với phản hồi với yêu cầu mà nó đang trả lời.

Người nhận yêu cầu phải trả lời bằng phản hồi hợp lệ cho tất cả các yêu cầu đã nhận. Phản hồi phải chứa các thuộc tính :

* result - Dữ liệu được trả về bởi phương thức được dẫn. Nếu xảy ra lỗi khi gọi phương thức, giá trị này phải là rỗng.
* error - Một mã lỗi được chỉ định nếu có lỗi khi gọi phương thức, nếu không thì sẽ là null.
* id - Id của yêu cầu mà nó đang phản hồi.

OpenWrt cung cấp JSON-RPC-API cho phép lập trình viên có thể thông qua kiến trúc Restful đưa ra các lời gọi hàm và nhận về các dữ liệu, thông tin mong muốn.

Dựa trên tài liệu cung cấp ở site: [https://github.com/OpenWrt/luci/wiki/JsonRpcHowTo[5],](https://github.com/OpenWrt/luci/wiki/JsonRpcHowTo,) Lập trình viên có thể xây dựng các function để theo dõi thông số của các router, theo dõi phiên kết nối của các client cũng như việc cập nhật cấu hình bằng những lời gọi hàm thích hợp.

Tùy mục đích các API cung cấp bởi OpenWrt được chia theo kiến trúc như sau:

Mỗi một thư viện của JSON-RPC-API sẽ được đưa vào một đường dẫn Restful riêng với những chức năng được cung cấp sẵn.

VD: Để thực hiện lệnh khởi động lại router bằng thư viện luci.sys ta có thể thực hiện với kiến trúc restful sau:

curl http://<hostname>/cgi-bin/luci/rpc/sys?auth=CHUOITOKEN '

{

"id": 1,

"jsonrpc": "2.0",

"method": "reboot",

"params": []

}'

chú thích:CHUOITOKEN

Chuỗi token là chuỗi xác thực được cung cấp trong một phiên làm việc để lập trình viên có quyền kiểm soát đối với toàn bộ các thư viện Json-Rpc. Nếu không có chuỗi này các lời gọi hàm Restful sẽ không thực thi được.

Để lấy chuỗi token ta có thể sử dụng kiến trúc Restful sau:

curl http://<hostname>/cgi-bin/luci/rpc/auth --data '

{

"id": 1,

"method": "login",

"params": [

"youruser",

"somepassword"

]

}'

**UCI**

UCI là viết tắt của Unified Configuration Interface và là một hệ thống tập trung cấu hình các dịch vụ OpenWrt.

UCI kế thừa việc cấu hình dựa trên NVRAM được tìm thấy trong loạt OpenWrt trắng của Nga và là giao diện người dùng cấu hình chính cho các thiết lập hệ thống quan trọng nhất. Ví dụ như cấu hình giao diện mạng chính, cài đặt không dây, chức năng ghi nhật ký và cấu hình truy cập từ xa. Ngoài ra, nhiều gói trong kho lưu trữ OpenWrt đã được thực hiện tương thích với hệ thống UCI.

Các ứng dụng được tạo tương thích UCI bằng cách chỉ cần viết tập tin cấu hình ban đầu (được chương trình đọc) theo cài đặt đã chọn trong tập tin UCI tương ứng. Điều này được thực hiện khi chạy các kịch bản khởi tạo trong /etc/init.d/. Do đó, khi bắt đầu một daemon với tập lệnh khởi tạo tương thích UCI, nên biết rằng tập tin cấu hình ban đầu của chương trình bị ghi đè. Ngoài ra, tập tin cấu hình của ứng dụng thường được lưu trữ trong RAM thay vì trong flash, bởi vì nó không cần phải được lưu trữ trong bộ nhớ không mất và nó được viết lại sau mỗi thay đổi, dựa trên tập tin UCI. Có nhiều cách để vô hiệu hóa UCI trong trường hợp ta muốn điều chỉnh các thiết lập trong tập tin cấu hình ban đầu không có sẵn thông qua UCI, trong cifs.server, có thể xem cách tắt UCI cho samba chẳng hạn.

Đối với những chương trình không tương thích UCI, có một danh sách tiện lợi của một số tập tin cấu hình không phải UCI mà ta có thể muốn. Lưu ý rằng, đối với hầu hết các chương trình của bên thứ ba, ta nên tham khảo tài liệu riêng của chương trình.  
Nguyên tắc chung.

Cấu hình trung tâm của OpenWrt được chia thành nhiều tập tin nằm trong thư mục /etc/config/. Mỗi tập tin liên quan đến một phần của hệ thống mà nó cấu hình. Ta có thể chỉnh sửa các tập tin cấu hình bằng trình soạn thảo văn bản hoặc sửa đổi chúng bằng chương trình tiện ích dòng lệnh uci. Các tập tin cấu hình UCI cũng có thể sửa đổi thông qua các API lập trình khác nhau (như Shell, Lua và C), cũng là cách các giao diện web như LuCI thay đổi các tập tin UCI.

Khi thay đổi tập cấu hình UCI, cho dù thông qua trình soạn thảo văn bản hoặc dòng lệnh, dịch vụ hoặc tập thực thi bị ảnh hưởng phải được khởi động lại (hoặc, trong một số trường hợp) bằng lệnh init.d, sao cho Cấu hình UCI cập nhật được áp dụng cho chúng. Nhiều chương trình được thực hiện tương thích với UCI theo cách này bằng cách làm cho init.d script của họ viết các tập cấu hình phần mềm tiêu chuẩn cụ thể của họ. Tập lệnh init.d đầu tiên viết đúng một tập tin cấu hình vào vị trí dự kiến của phần mềm và nó được đọc lại bằng cách khởi động lại tập tin thực thi. Lưu ý rằng chỉ (re) bắt đầu thực thi trực tiếp, không có các cuộc gọi init.d, sẽ không dẫn đến một bản cập nhật UCI để relegate cấu hình UCI thành tập tin cấu hình dự kiến của chương trình. Những thay đổi trong các tập tin trong /etc/config/ sau đó không có hiệu lực.

**Hostpad**

Hostapd là một daemon không gian người dùng cho các máy chủ xác thực và điểm truy cập. Nó thực hiện quản lý điểm truy cập IEEE 802.11, Trình xác thực IEEE 802.1X / WPA / WPA2 / EAP, máy khách RADIUS, máy chủ EAP và máy chủ xác thực RADIUS. Phiên bản hiện tại hỗ trợ Linux (Trình điều khiển máy chủ AP, madwifi, trình điều khiển dựa trên mac80211) và FreeBSD (net80211).

Hostapd được thiết kế để trở thành một chương trình "daemon" chạy trong nền và hoạt động như thành phần phụ trợ kiểm soát xác thực. hostapd hỗ trợ các chương trình giao diện người dùng riêng biệt và một giao diện văn bản ví dụ, hostapd\_cli, được bao gồm với hostapd

**Ubus**

Để cung cấp thông tin liên lạc giữa các trình tiện ích và ứng dụng khác nhau trong OpenWrt, một dự án được gọi là ubus đã được phát triển. Nó bao gồm một số phần bao gồm daemon, thư viện và một số người trợ giúp thêm.

Trung tâm của dự án này là daemon ubusd. Nó cung cấp một giao diện cho các trình tiện ích khác để tự đăng ký cũng như gửi tin nhắn. Đối với những người tò mò, giao diện này được thực hiện bằng cách sử dụng socket Unix và nó sử dụng thông điệp TLV(type-length-value) .

Để đơn giản hóa việc phát triển phần mềm bằng cách sử dụng ubus (kết nối với nó) một thư viện có tên libubus đã được tạo ra.

Mỗi daemon đăng ký một tập hợp các đường dẫn dưới một không gian tên cụ thể. Mỗi đường dẫn có thể cung cấp nhiều thủ tục với bất kỳ số lượng đối số nào. Thủ tục có thể trả lời bằng tin nhắn.

## TFTP

TFTP là viết tắt của Trivial File Transfer Protocol là một giao thức truyền tập tin nhỏ gọn cho phép client lấy tập tin hoặc đặt tập tin lên máy chủ từ xa. Một trong những ứng dụng chính của nó là trong giai đoạn đầu của các nút khởi động từ một mạng cục bộ. TFTP đã được sử dụng cho ứng dụng này bởi vì nó rất đơn giản để thực hiện.

TFTP được chuẩn hóa lần đầu tiên vào năm 1981 và đặc điểm kỹ thuật hiện tại cho giao thức có thể được tìm thấy trong RFC 1350.

## TIẾN TRÌNH KHỞI ĐỘNG ROUTER

Khi khởi động router, nó sẽ trải qua tiến trình khở động gồm 4 bước:

Bước 1 - POST (Power on self test): Router thực kiểm tra các module phần cứng (Hardware diagnostic)

Bước 2: Sau khi kiểm tra xong và đảm bảo mọi module hoạt động hợp lệ, router sẽ chạy chương trình bootstrap nằm trong bộ nhớ ROM để load chương trình bootstrap từ ROM vào bộ nhớ RAM

Bước 3: Sau khi đã load chương trình bootstrap vào RAM, chương trình sẽ load hệ điều hành IOS cho router để chạy theo tiến trình sau:

Đầu tiên, nếu trình flash có tồn tại IOS, router sẽ giải nén và load IOS đầu tiên để vận hành router.

Trong trường hợp không tìm thấy IOS hoặc IOS trong flash lỗi, router sẽ tiến hành tìm kiếm hệ điều hành từ các TFTP server xung quanh nó để load về và vận hành

Nếu cả hai tình huống trên đều không tìm được IOS thì router sẽ load hệ điều hành phụ trong ROM để chạy

Bước 4: router load tập tin cấu hình cho router để chạy cho nó.

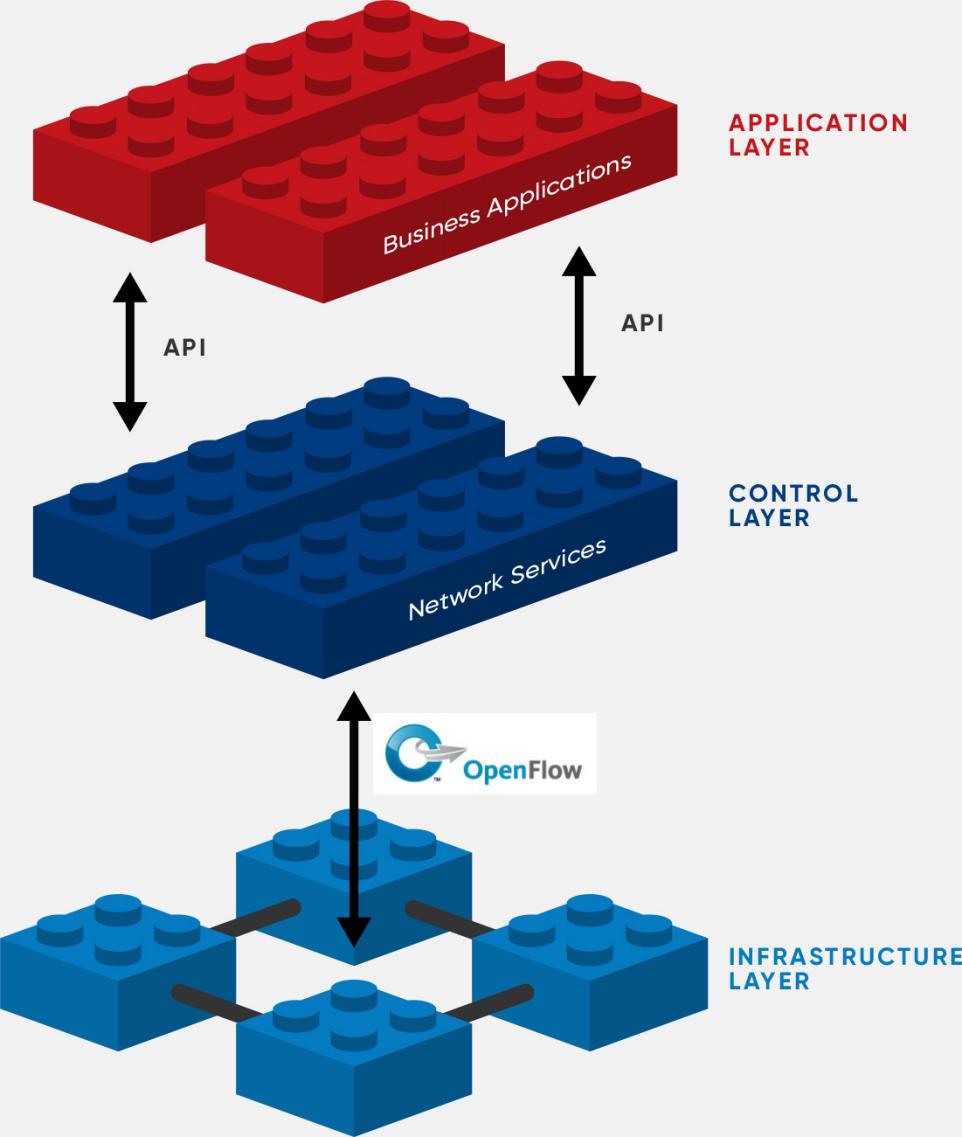
## GIỚI THIỆU CÔNG NGHỆ SDN

### SDN là gì?

Mạng điều khiển bằng phần mềm (SDN) là một kiến trúc mới nổi có tính năng động, dễ quản lý, hiệu quả về chi phí và có thể thích nghi, làm cho nó trở nên lý tưởng cho tính chất động, băng thông cao của các ứng dụng ngày nay. Kiến trúc này tách các chức năng điều khiển và chuyển tiếp mạng cho phép điều khiển mạng trở thành lập trình trực tiếp và cơ sở hạ tầng cơ bản được trừu tượng hóa cho các ứng dụng và dịch vụ mạng. Giao thức OpenFlow là một yếu tố cơ bản để xây dựng các giải pháp SDN.

Kiến trúc SDN gồm 3 lớp riêng biệt:

* Lớp ứng dụng (Application layer): là các ứng dụng được triển khai dựa trên việc kết nối với lớp điều khiển thông qua các API.
* Lớp điều khiển (Control layer): là nơi tập trung các bộ điều khiển thực hiện việc điều khiển và cấu hình mạng theo các yêu cầu từ lớp ứng dụng.
* Lớp cơ sở hạ tầng (Infrastructure layer): Là các thiết bị mạng thực tế (vật lý hay ảo hóa) thực hiện việc chuyển tiếp gói tin theo sự điều khiển của lớp điểu khiển. Một thiết bị mạng có thể hoạt động theo sự điều khiển của nhiều bộ điều khiển khác nhau, điều này giúp tăng cường khả năng ảo hóa của mạng.



### Giao thức OpenFlow

OpenFlow là tiêu chuẩn cung cấp khả năng giao tiếp giữa các giao diện của lớp điều khiển và lớp cơ sở hạ tầng trong kiến trúc SDN. OpenFlow cho phép truy cập trực tiếp và điều khiển lớp cơ sở hạ tầng như switch, router cả thiết bị thật lẫn ảo. Nhờ có OpenFlow mà tách được lớp điều khiển với các thiết bị chuyển mạch thực tế.

Giao thức OpenFlow phải được triển khai trên cả hai thiết bị ở lớp điều khiển và lớp cơ sở hạ tầng. Việc sử dụng OpenFlow có thể thông qua các tập lệnh để điều khiển mặt phẳng cơ sở hạ tầng.

## CÔNG NGHỆ WEB

- World Wide Web, gọi tắt là Web hoặc WWW, mạng lưới toàn cầu là một không gian thông tin toàn cầu mà mọi người có thể truy cập (đọc và viết) qua các thiết bị kết nối với mạng Internet; một hệ thống thông tin trên Internet cho phép các tài liệu được kết nối với các tài liệu khác bằng các liên kết siêu văn bản, cho phép người dùng tìm kiếm thông tin bằng cách di chuyển từ tài liệu này sang tài liệu khác.

- Website còn gọi là trang web, là một tập hợp các trang web con, bao gồm văn bản, hình ảnh, video, flash v.v.. website chỉ nằm trong một tên miền hoặc tên miền phụ lưu trữ trên các máy chủ.

- Một website cần có những thành phần chính như tên miền là tên riêng trỏ đến máy chủ chứa các tập tin nguồn, hosting là máy chủ chứa các tập tin nguồn, mã nguồn là các tập tin html,xhml… Website được tương tác và hiển thị với người dùng qua trình duyệt web.

Các nền tảng để xây dựng website:

**HTML**: là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thi thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

**CSS**: CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thay đổi “diện mạo” của các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,…rất nhiều.

**PHP**: PHP được xây dựng ban đầu như là một dự án mã nguồn mở nhỏ, nhưng càng về sau người ta càng cảm thấy được sự hữu ích của nó nên đã phát triển php thành ngôn ngữ phổ biến như bây giờ. Tên ban đầu của ngôn ngữ php là php/fi, php/fi do Rasmus Lerdorf tạo ra năm 1994, ban đầu php/fi được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của Rasmus Lerdorf trên mạng. Sau một thời gian Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó, đồng thời cải tiến mã nguồn.

PHP là viết tắt của "PHP: Hypertext Preprocessor".

PHP là ngôn ngữ lập trình kịch bản viết cho máy chủ mà được nhúng trong HTML. Nó được sử dụng để quản lý nội dụng động, cơ sở dữ liệu, kiểm tra phiên làm việc...

PHP được tích hợp với một số Database thông dụng như MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.

PHP hỗ trợ một số lượng rộng rãi các giao thức lớn như POP3, IMAP, và LDAP. PHP4 bổ sung sự hỗ trợ cho Java và các cấu trúc đối tượng phân phối (COM và CORBA).

PHP có khả năng cực lớn trong việc lập trình web:

* PHP thực hiện được các hàm hệ thống, ví dụ: từ các file trên một hệ thống, nó có thể tạo, mở, đọc, ghi và đóng chúng.
* PHP có thể xử lý các form, ví dụ: thu thập dữ liệu từ file, lưu dữ liệu vào một file, thông qua email ta có thể gửi dữ liệu, trả về dữ liệu tới người dùng.
* Bạn có thể thêm, xóa, sửa đổi các phần tử bên trong Database của bạn thông qua PHP.
* Truy cập các biến Cookie và thiết lập Cookie.
* Sử dụng PHP có thể mã hóa dữ liệu theo các chuẩn khác nhau.

Một trong những tính năng ưu việt của php là hỗ trợ gửi và nhận các Restful request-công nghệ cốt lõi trong đề tài luận văn này.

**MySQL**: là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

**JavaScript**: Là một trong những ngôn ngữ lập trình chính cho lập trình web. Javascritp giúp điều chỉnh lại cách hoạt động của một trang web. Javascritp có khả năng liên kết trực tiếp với các thẻ html thay đổi nội dung cũng như cách hiển thị của chúng. Javascript có thể lắng nghe các sự kiện trên trang web thậm chí đén từng cú click chuột của người dùng với mọi thành phần của trang web. Nói tóm lại javascript mang một tiềm năng rất lớn giúp website tương tác tốt hơn đối với các khách hàng sử dụng. Mang nhiều ưu điểm tuy nhiên javascipt là một ngôn ngữ lập trình hoạt động ở phía client tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập. Một số trình duyệt sẽ cho phép người dùng tắt bỏ đi javascript nên hãy lưu ý cách mà website sẽ hoạt động nếu nó không có javascritp.

**jQuery**: Là một thư viện kiểu mới của JavaScript, được tạo bởi John Resig vào năm 2006 với một phương châm tuyệt vời: Write less, do more - Viết ít hơn, làm nhiều hơn. jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax. Với jQuery, khái niệm RAPId Web Development đã không còn quá xa lạ. jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn.

**Bootstrap**: Là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive dễ dàng hơn và nhanh chóng hơn.

**Công nghệ Restful**: Rest Là một kiều kiến trúc lập trình, nó định nghĩa các quy tắc để thiết kết các web service chú trọng vào tài nguyên hệ thống. Trong kiến trúc REST mọi thứ đều được coi là tài nguyên, chúng có thể là: tệp văn bản, ảnh, trang html, video, hoặc dữ liệu động… REST server cung cấp quyền truy cập vào các tài nguyên, REST client truy cập và thay đổi các tài nguyên đó. Ở đây các tài nguyên được định danh dựa vào URI, REST sử dụng một vài đại diện để biểu diễn các tài nguyên như văn bản, JSON, XML.

Bốn quy tắc cơ bản để cài một Restful webservice:

* Sử dụng một cách rõ ràng các phương thức http

Để tạo một tài nguyên trên server ta dùng phương thức POST.

Để lấy(đọc) tài nguyên trên server ta dùng phương thức GET.

Để update tài nguyên trên server ta dùng phương thức PUT.

Để xóa tài nguyển trên server ta dùng phương thức DELETE.

* Phi trạng thái: Theo REST, trạng thái hoặc được giữ trên client hoặc được chuyển thành trạng thái của tài nguyên. Nói một cách khác một server sẽ không bao giờ giữ trạng thái thông tin trao đổi với bất kỳ client nào nó giao tiếp, mỗi request lên server thì client phải đóng gói thông tin đầy đủ để thằng server hiểu được. Điều này giúp hệ thống của bạn dễ phát triển,bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client. Ngoài ra còn một nguyên nhân quan trọng hơn đó là nó tách biệt client khỏi sự thay đổi của server.

Cấu trúc thư mục như URI: URI trong RESTful web service phải tự mô tả, hoặc tham chiếu được cái mà nó trỏ tới và các tài nguyên liên quan. Ngoài ra URI cũng phải đơn giản, có thể đoán biết được, và dễ hiểu. Để tạo ra URI với yêu cầu trên thì ta nên định nghĩa URI có câu trúc giống thư mục. Loại URI này có phân cấp, có gốc là một đường dẫn đơn, các nhánh từ gốc là các đường dẫn phụ dẫn đến các các vùng service chính.

* Chuyển đổi XML, JSON hoặc cả hai: Điều cuối cùng trong tập các ràng buộc khi thiết kế RESTful web service phải làm là định dạng dữ liệu mà ứng dụng và service trao đổi trong phụ tải request/response hoặc trong HTTP body. Cung cấp nhiều đại diện biểu diễn cho tài nguyên cho các request khác nhau. Cụ thể ở đây ta có thể sử dụng các một vài kiểu MIME thông dụng sau:
* JSON
* XML
* Điều này cho phép các service sử dụng bởi các client viết bởi các ngôn ngữ khác nhau, chạy trên nhiều nền tảng và thiết bị khác nhau. Sử dụng các kiểu MIME cho phép client chọn dạng dữ liệu phù hợp với nó.

# CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

### MÔ TẢ CHẾ ĐỘ HOẠT ĐỘNG

Mô hình được thiết kế có thể chạy ở 1 trong 2 chế độ:

* Chế độ ban ngày: khi ở chế độ ban ngày, các LED 7 đoạn sẽ đếm ngược trở về 0 để chuyển màu đèn tín hiệu. Các đèn tín hiệu sẽ chuyển lần lượt từ màu xanh sang vàng, vàng sang đỏ, đỏ trở về xanh và lặp lại liên tục. Ở hai hướng kề nhau thì màu đèn tín hiệu không được trùng nhau trong cùng thời điểm.
* Chế độ ban đêm: khi ở chế độ ban đêm, các đèn LED 7 đoạn sẽ được tắt để tiết kiệm năng lượng, đồng thời các đèn tín hiệu màu vàng sẽ sáng và tắt theo chu kỳ 1 giây.

### YÊU CẦU KỸ THUẬT

* Các LED 7 đoạn phải đếm ngược đúng như dữ liệu nhập vào.
* Đèn tín hiệu chuyển đổi màu đúng như mô tả hệ thống.
* Mô hình phải chạy vận hành được trong thời gian dài.

### THIẾT KẾ TƯƠNG TÁC

Cấp nguồn

Xuất dữ liệu

Điều khiển

Điều khiển

Nhập dữ liệu

Xuất dữ liệu

Điều khiển

Người dùng

Arduino

I2C

LCD

Người dùng

Hiển thị

Bàn phím

Arduino

74HC595

LED

Hình 15. Sơ đồ tương tác của người dùng và mô hình

Sơ đồ trên mô tả cách thức tương tác giữa người dùg và hệ thống. Khởi đầu, người dùng sẽ cấp nguồn điện cho Arduino, lúc này Arduino sẽ xuất dữ liệu sang I2C để I2C điều khiển hiển thị lên LCD. Tiếp sau đó, dựa vào nội dung trên màn hình LCD, người dùng sẽ điều khiển bàn phím để nhập dữ liệu vào Arduino. Sau khi có được dữ liệu từ người dùng, Arduino sẽ xuất dữ liệu đã qua xử lý sang IC 74HC595 để điều khiển các LED và LED 7 đoạn.

Chương trình sẽ được vận hành liên tục cho đến khi người dùng ngắt điện. Khi người dùng muốn thay đổi dữ liệu nhập vào thì có thể ngắt điện trực tiếp và cấp lại hoặc nhấn nút Reset trên Arduino.

### THIẾT KẾ GIẢI THUẬT

Bắt đầu

Kết thúc

Sai

Đúng

Chọn chế độ

Đã chọn chế độ

Hình 16. Giải thuật chọn chế độ.

Hình 17. Giải thuật vận hành chế độ ban ngày

Đúng

Bắt đầu

Nhập giá trị đèn xanh lighttime[0]

Nhập giá trị đèn vàng lighttime[1]

Lighttime[0]+lighttime[1]<100 &&

Lighttime[0] >= 10&&

Lighttime[1] >= 3&& lighttime[1] <= 5

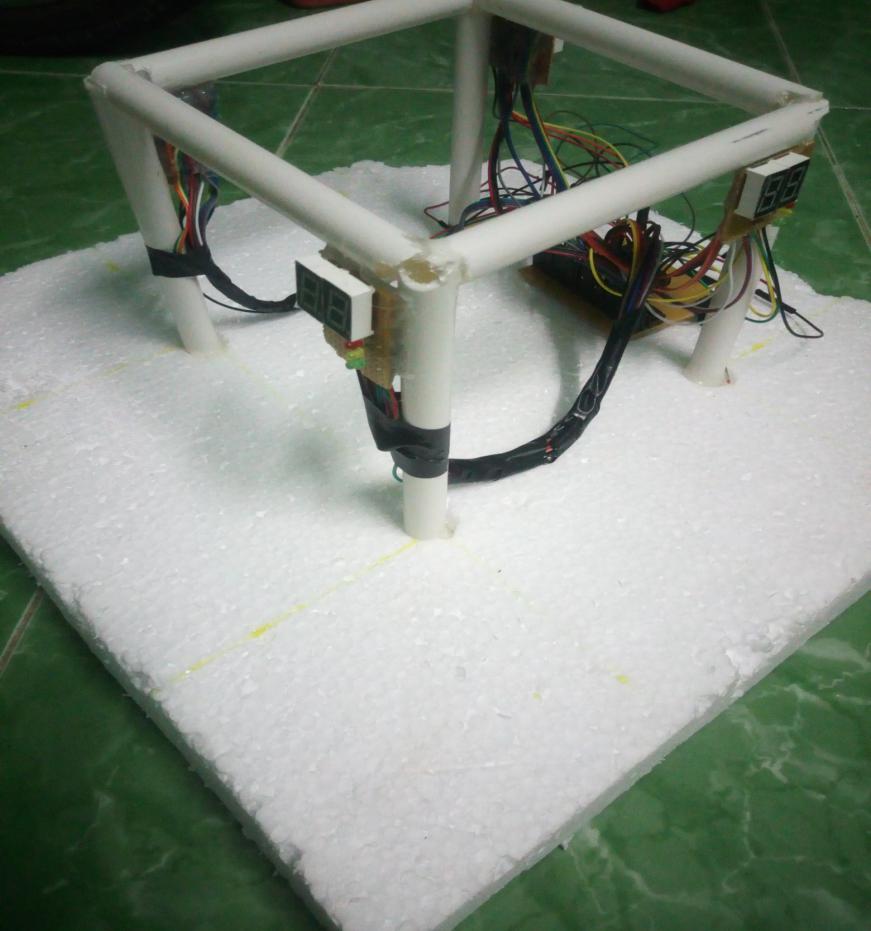
Kết thúc

Sai

### LẮP MÔ HÌNH

* Kết nối keypad với Arduino: trên keypad có 8 dây được sắp xếp lần lượt từ trái sang phải tương ứng với dây 1 đến 8. Ta sử dụng dây dẫn để nối từ keypad sang Arduino theo thứ tự dây 1 đến 8 trên keypad nối lần lượt từ chân số 12 đến chân số 5 của Arduino
* Kết nối I2C với Arduino: để I2C hoạt động được cần phải cấp nguồn cho nó nên ta nối dây GND, VCC trên I2C lần lượt các tới chân GND, 5V trên Arduino. Sau khi đã cấp nguồn cho I2C, ta nối dây từ chân SCL trên I2C tới chân A5 của Arduino và nối dây từ chân SDA trên I2C đến chân A4 trên Arduino.
* Kết nối I2C với LCD : để I2C có thể điều khiển được LCD ta phải ghép I2C vào phía sau LCD và kết nối 16 chân của I2C tương ứng với 16 chân của LCD.
* Nối 74HC595 với Arduino: để IC 74HC595 có thể nhận được nguồn từ Arduino ta phải cung cấp nguồn âm và các chân 8 và 13 trên 74HC595 và cấp nguồn 5v và các chân 10 và 16 của 74HC595. Sau đó, kết nối các chân 11, 12, 14 của IC lần lượt tới các chân 13, 2, 3 của Arduino.
* Nối tiếp các 74HC595: khi điều khiển số lượng lớn các LED và LED 7 đoạn ta không thể nối nhiều con IC 74HC595 và Arduino vì sẽ gây lãng phí chân nên ta chọn giải pháp nối tiếp các IC 74HC595 bằng cách nối mỗi IC từ con thứ 2 trở về sau sẽ nối lần lượt với con IC trước nó bằng cách nối chân 11, 12, 14 của con IC sau lần lượt tới chân 11, 12, 9 của IC đứng trước nó.
* Nối LED 7 đoạn với 74HC595: các chân LED 7 đoạn từ a đến g tương ứng nối tới chân 15 và các chân từ 1 đến 6 của IC và cấp nguồn dương hay âm vào chân nguồn tùy thuộc vào loại LED 7 đoạn đang sử dụng
* Cấp nguồn cho Arduino: cấp nguồn một chiều từ 6 - 20V nhưng khuyên dùng điện áp từ 7 - 12V để đảm bảo hoạt động tốt và sử dụng lâu bền.

Sau khi đã liên kết các linh kiện với nhau, ta sẽ gắn các linh kiện lên mô hình.



Hình 18. Mô hình đèn giao thông.

### KIỂM THỬ

Sau khi lắp ráp mô hình và kiểm thử sản phẩm, nhóm nhận thấy các đèn trên mô hình hoạt động đúng như mục tiêu đề ra, ánh sáng phát ra rất tốt, có thể thấy được từ khoảng cách xa. Mô hình hoạt động được trong thời gian rất dài mà không xảy ra lỗi.

Tuy nhiên nhóm lại gặp vấn đề là có một LED 7 đoạn bị mất 1 nét, chúng tôi đã kịp nhận ra điều này và kiểm tra lại thì nhận thấy trong quá trình hàn mạch thì rào cắm điện bị hở khiến cho chân IC không tiếp xúc được với dây dẫn đến chân LED 7 đoạn. Ngoài ra không có vấn đề nào khác đối với mô hình.

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT QUẢ ĐẠT ĐƯỢC

Sau khoảng thời gian nghiên cứu, tìm tòi và học hỏi, nhóm đã hoàn thành nghiên cứu và đã giải quyết được 4 vấn đề sau:

- Tìm hiểu cơ bản về lập trình nhúng.

- Tìm hiểu về Arduino và kết nối một số IC, đèn LED.

- Tìm hiểu về cách thức hoạt động, kết nối cũng như là lập trình để kết nối và điều khiển được các đèn Led, bo mạch, IC, LCD, Led 7 đoạn và Arduino.

- Đèn tín hiệu chạy tốt và không có lỗi gặp phải.

Điểm hạn chế của đề tài cần được khắc phục:

- Dây quá nhiều và rối, không thiết kế dây thẩm mỹ.

- Không in mạch và mô hình còn khá sơ sài.

- Bước đầu lập trình nhúng căn bản nên còn khá sơ sài so với thực tiễn đã áp dụng bên ngoài.

## HƯỚNG PHÁT TRIỂN

Đề tài bước đầu chỉ là tìm hiểu và thực hành cơ bản về lập trình nhúng, hướng đến mục tiêu sẽ áp dụng internet, có máy chủ các hệ thống đèn là các thiết bị được điều khiển từ xa.

Thêm các module để xử lý lưu lượng xe trên đường và tự động thay các số giây trên đèn nhằm đảm bảo tính nhanh và dể dàng cho các phương tiện tham gia lưu thông.

Có đèn cảnh báo khi nếu có xe nào muốn và vượt đèn đỏ. Đồng thời thông tin xe sẽ được gửi về máy chủ để xử lý.

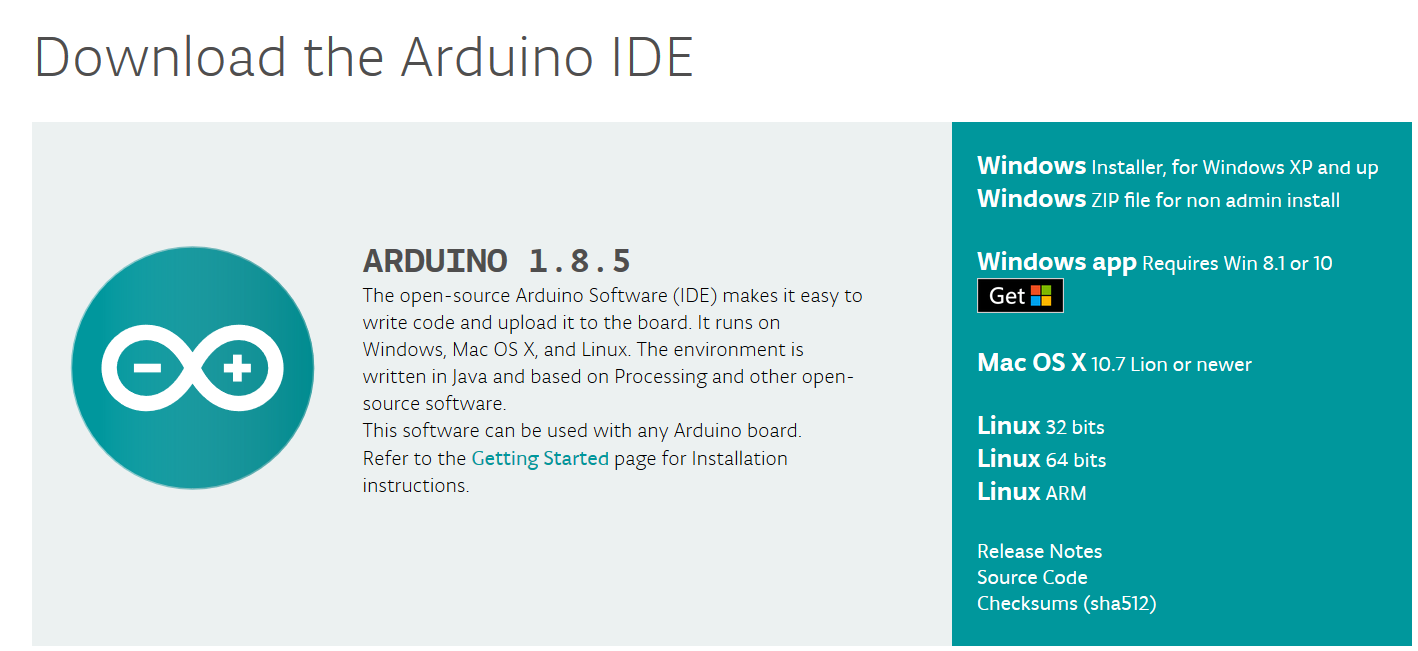
# PHỤ LỤC 1

**HƯỚNG DẪN CÀI ĐẶT ARDUINO IDE**

* Để cài đặt IDE dành cho phần lập trình Arduino cần vào trang chủ đê donwload

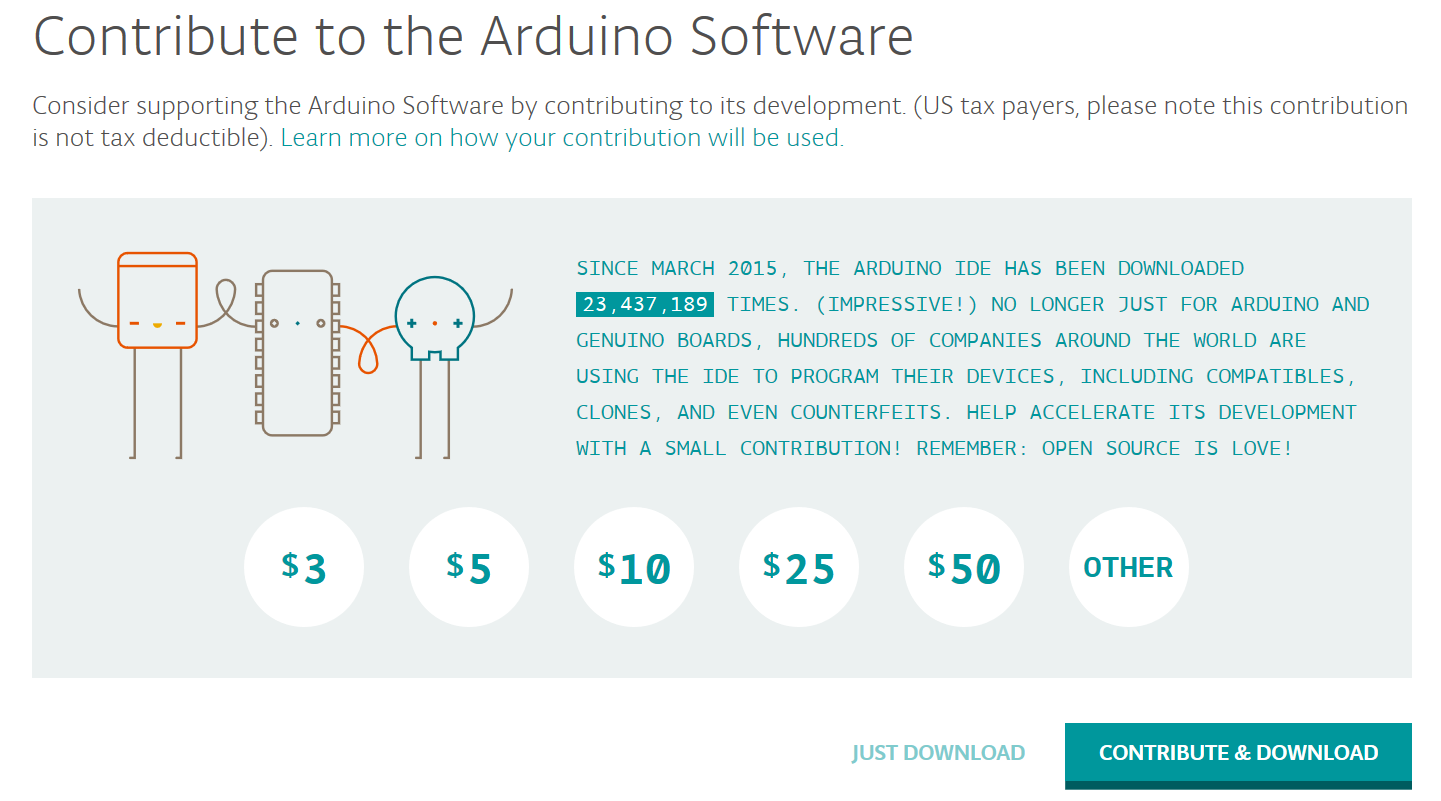
**Bước 1:** Vào địa chỉ <https://www.arduino.cc/en/Main/Software>.

**Bước 2:** Chọn vào đúng hệ điều hành phù hợp để tải về.



Hình 19: Giao diện tải IDE dành cho Arduino

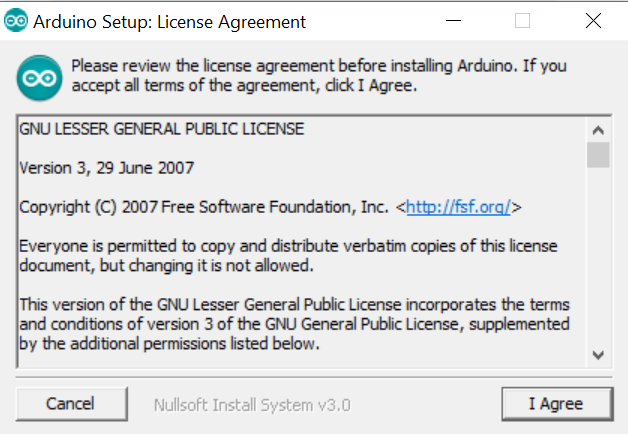
**Bước 3**: Chọn bắt đầu download, chỉ tải về hoặc là đóng góp kinh phí cho chương trình



Hình 20: Giao diện chỉ tải Arduino IDE về hoặc đóng góp kinh phí

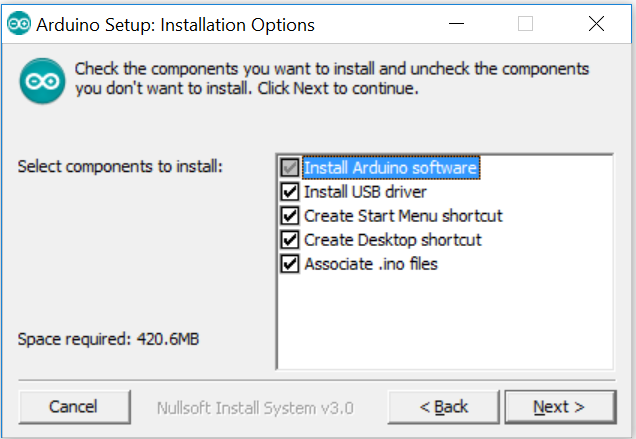
* Tiến hành cài đặt Arduino IDE cho hệ điều hành

**Bước 1:** Chọn vào I Agree để đồng ý với điều khoảng của phần mềm



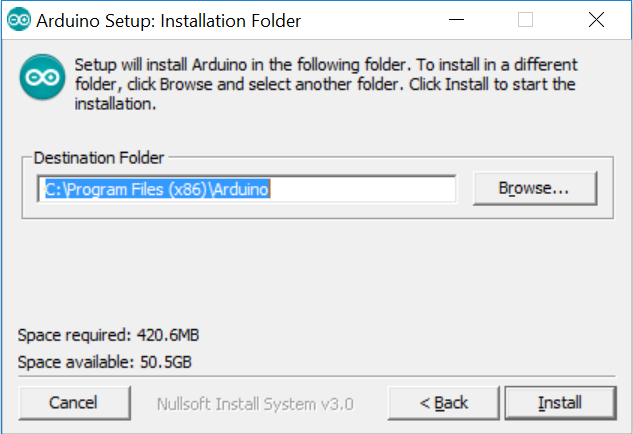
Hình 21: Chấp nhận điều khảng cài đặt IDE

**Bước 2**: Chọn các tùy chọn cài đặt



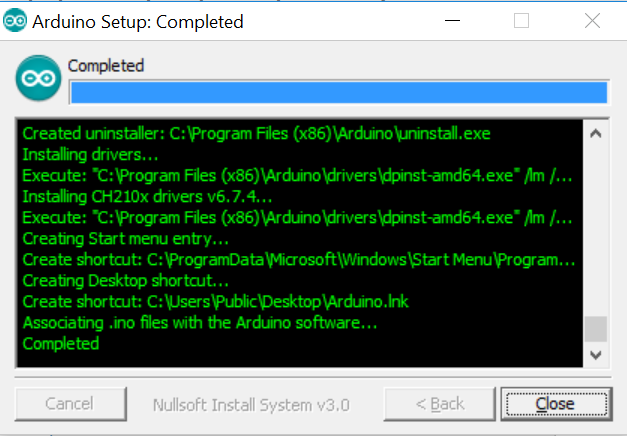
Hình 22: Chọn phần cài đặt của gói phần mềm

**Bước 3**: Chọn thư mục để cài đặt



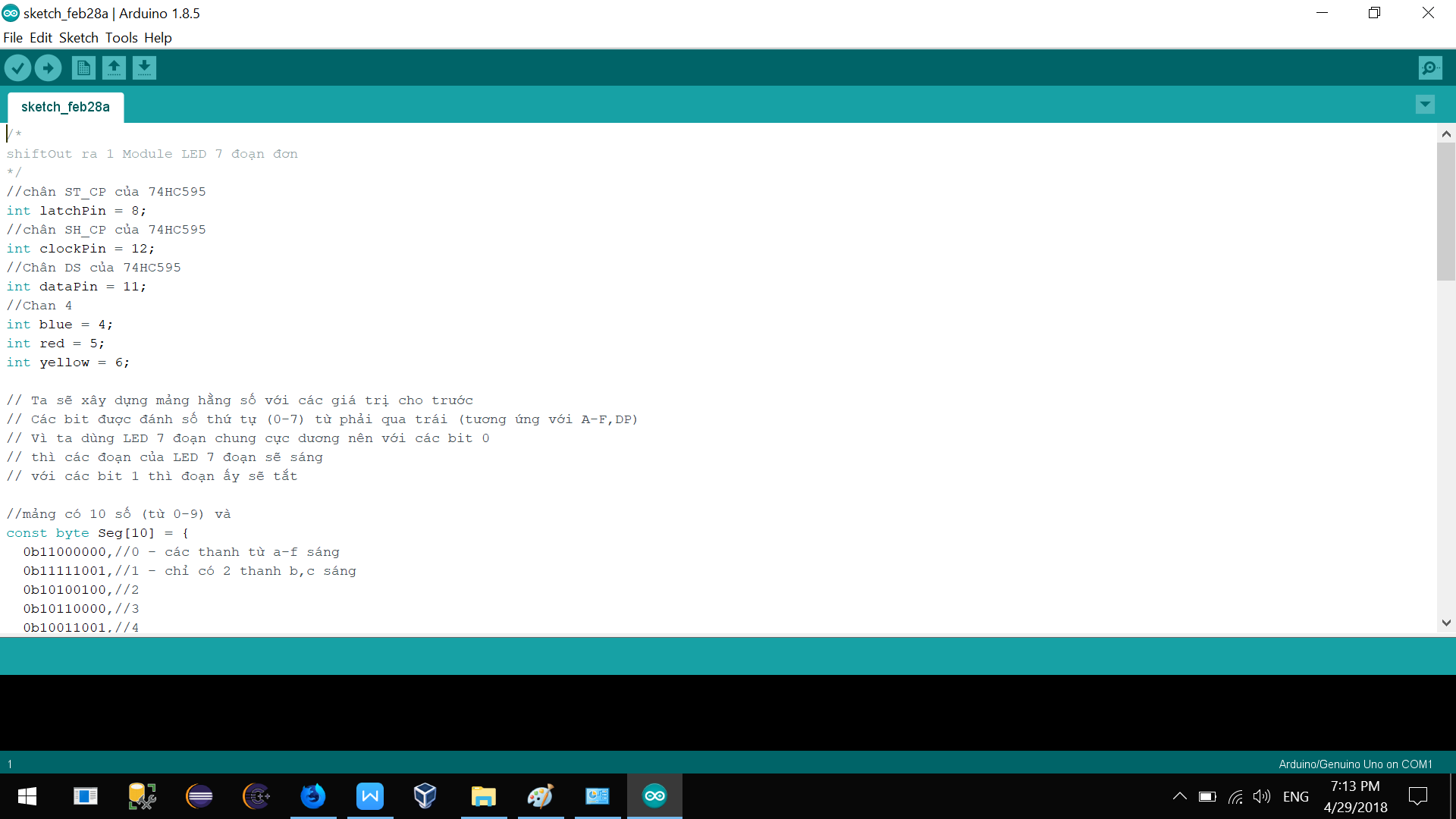
Hình 23: Chọn thư mục cài đặt IDE

**Bước 4**: Nhấn Close để kết thúc khi cài đặt xong



Hình 24: Hoàn thành cài đặt IDE

**Bước 5**: khởi động giao diện viết code và nạp trương trình vào Arduino



Hình 25: Giao diện Arduino IDE

# 

# PHỤ LỤC 2

**HƯỚNG DẪN NẠP CHƯƠNG TRÌNH VÀO ARDUINO**

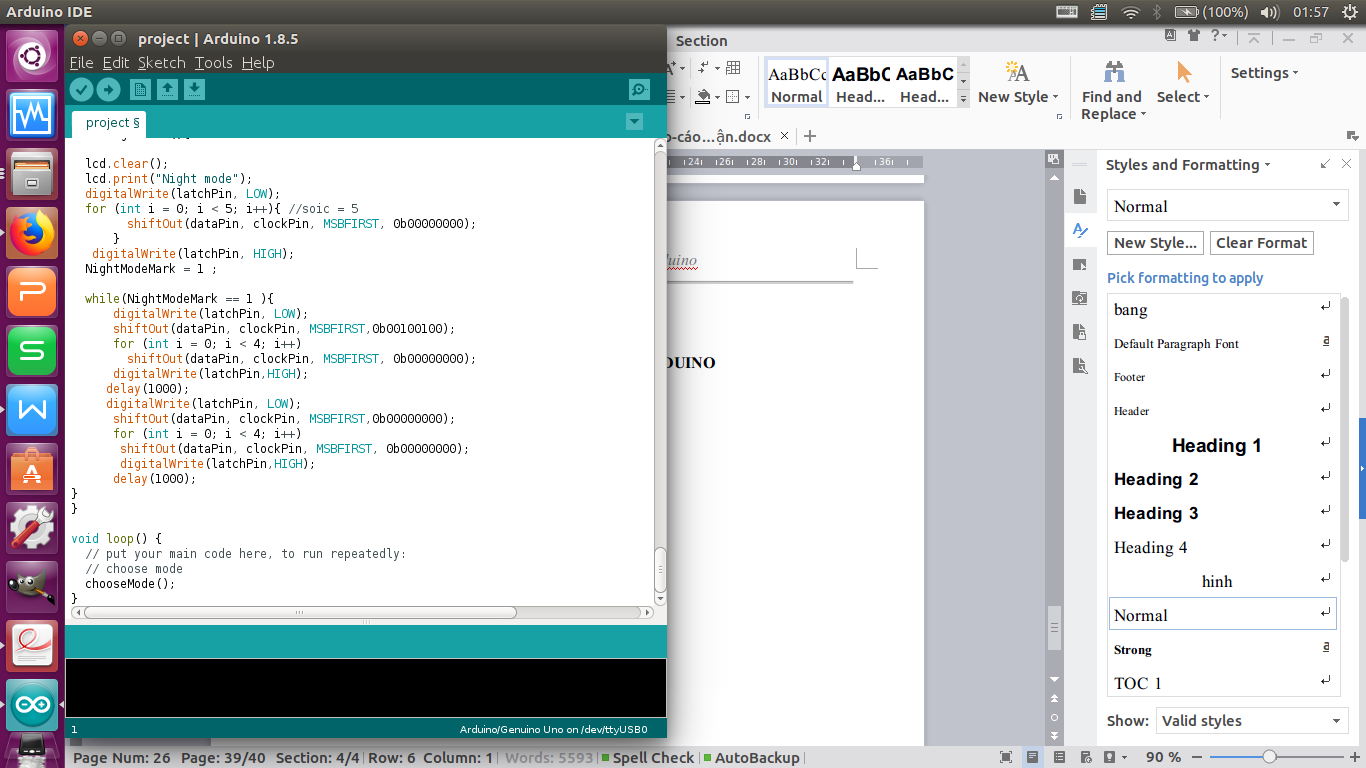
**Bước 1:** Khởi động Arduino IDE

**Bước 2:** Kết nối Arduino với máy tính



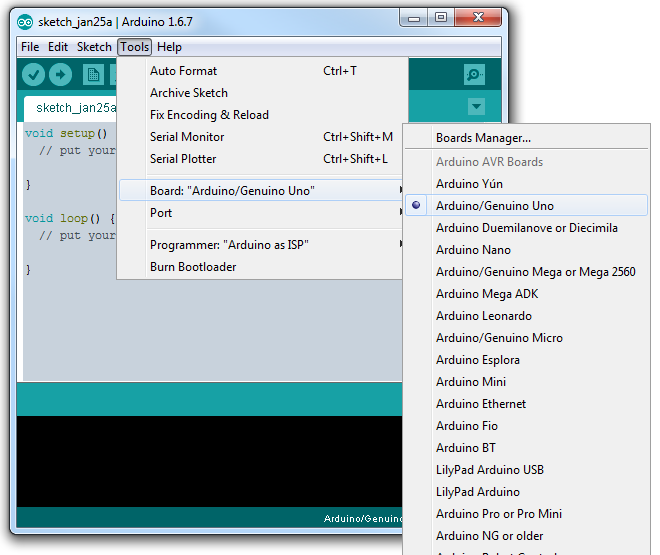
Hình 26. Kết nối Arduino với máy tính

**Bước 3:** Viết mã lệnh cho chương trình



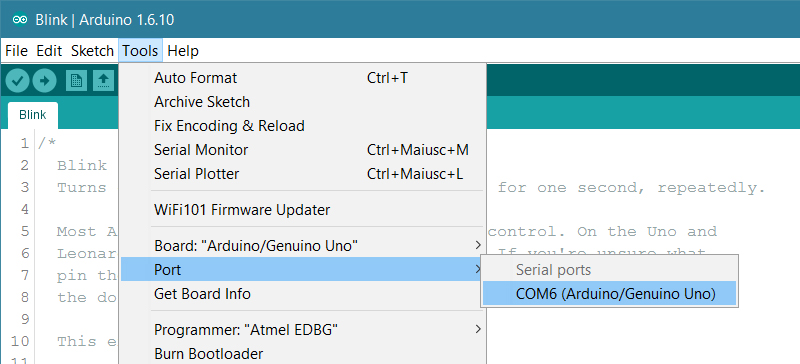
Hình 27. Viết mã lệnh cho chương trình

**Bước 4:** Chọn bo mạch đúng với tên bo mạch đang sử dụng



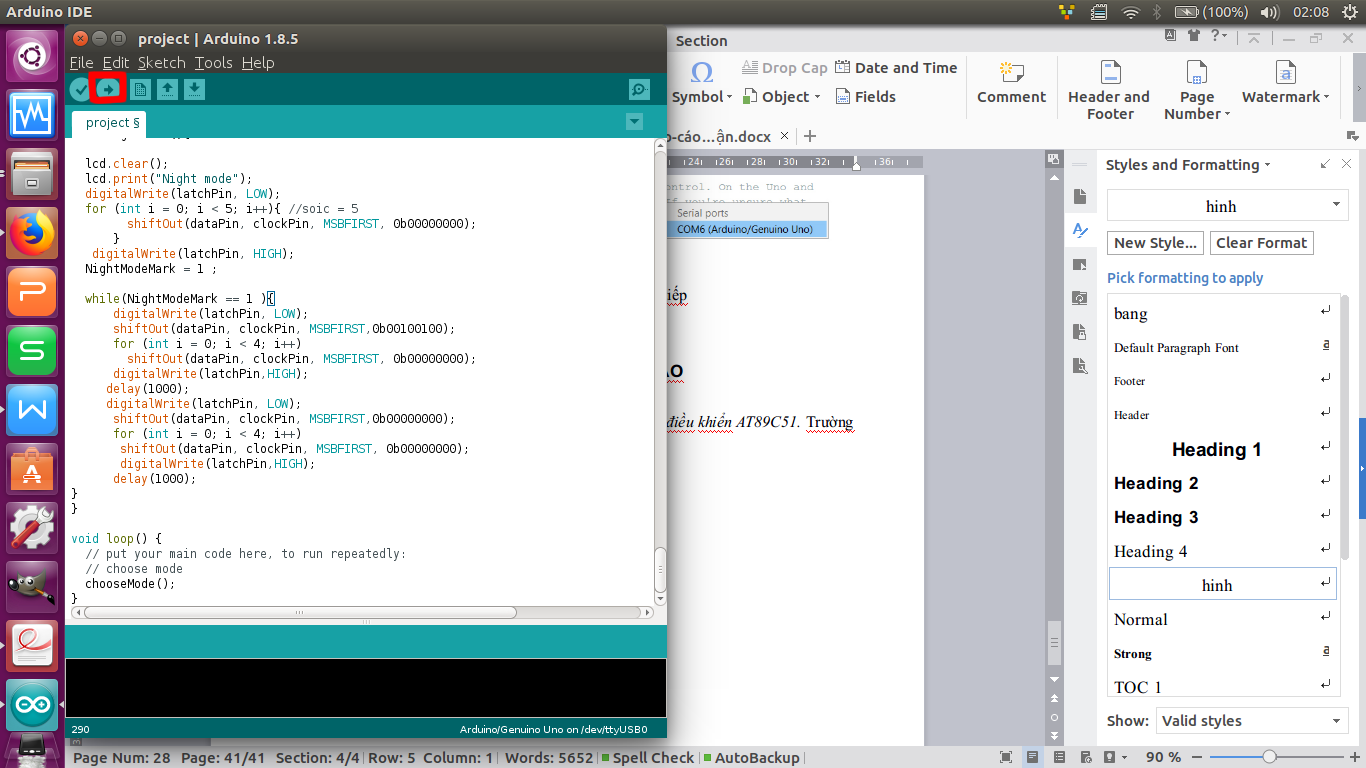
Hình 28. Chọn bo mạch

**Bước 5:** Chọn cổng giao tiếp



Hình 29. Chọn cổng giao tiếp

**Bước 6:** Nạp chương trình bằng cách ấn vào nút Verify



Hình 30. Nạp chương trình cho Arduino

# TÀI LIỆU THAM KHẢO

[1] **Lê Trần Minh.** *Mô hình đèn giao thông dùng vi điều khiển AT89C51.* Trường Đại Học Kỹ Thuật Thành Phố Hồ Chí Minh. 2011.