**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙢 ❖ 🙠



**ĐỒ ÁN MÔN HỌC**

**IOT VÀ ỨNG DỤNG**

**(SMART HOME)**

Sinh viên thực hiện : 14127871 Phan Bảo Trinh

14026751 Vương Quốc Việt

Giáo viên hướng dẫn : ThS. Nguyễn Thành Thái

Lớp : ĐHCNTT10B

***TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2019***

# LỜI NÓI ĐẦU

Internet of Things (IoT – kết nối vạn vật) đang trở nên phổ biến khi hàng tỷ thiết bị được kết nối chung với nhau bằng internet, dần có tác động mạnh mẽ trên nhiều lĩnh vực đời sống xã hội. Sự phát triển mạnh mẽ của hạ tầng internet, sự bùng nổ của điện thoại di động, nhiều doanh nghiệp Việt Nam đã bắt nhịp với xu hướng công nghệ mới này. Hệ thống nông nghiệp thông minh, xe tự lái, và đặc biệt nhà thông minh là một lĩnh vực tiên phong ứng dụng IoT, đã và đang chuyển mình mạnh mẽ trong giai đoạn hiện nay. Với những kiến thức đã học cùng với mong muốn thiết kế một ngôi nhà tự động hóa đáp ứng được nhu cầu sinh hoạt hằng ngày, chúng em đã chọn “Smart home” làm đề tài cho thực tập tốt nghiệp của mình .

Trong quá trình thực hiện báo cáo của mình, chúng em đã cố gắng hết sức để hoàn thiện một cách tốt nhất. Nhưng với kiến thức và sự hiểu biết có hạn nên sẽ không tránh khỏi những thiết sót mong các thầy cô bổ sung và đóng góp ý kiến cho đề tài của em có thể hoàn thiện hơn. Em xin gửi lời cảm ơn đến thầy Nguyễn Thành Thái đã nhiệt tình giúp đỡ em trong quá trình thực hiện thực tập thời gian qua.

# DANH MỤC TỪ VIẾT TẮT

CMS Content Management System

CSS Cascading Style Sheets

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

ID Identification

IP Internet Protocol

SSD Solid State Drive

SSH Secure Socket Shell

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Hệ thống chấm bài Themis. 1](#_Toc496047823)

[Hình 2.1. Use case tổng quát. 4](#_Toc496047824)

[Hình 2.2. Giao diện Mockup – Home. 14](#_Toc496047825)

[Hình 2.3. Giao diện Mockup – Customtest. 14](#_Toc496047826)

[Hình 2.4. Sơ đồ hoạt động của môi trường chấm bài. 15](#_Toc496047827)

[Hình 3.1. Cửa sổ dịch vụ của AWS. 17](#_Toc496047828)

[Hình 3.2. Thêm người dùng. 17](#_Toc496047829)

[Hình 3.3. Điền thông tin người dùng. 17](#_Toc496047830)

[Hình 3.4. Cấp quyền người dùng. 18](#_Toc496047831)

[Hình 3.5. Thông tin người dùng đã đăng ký. 18](#_Toc496047832)

[Hình 3.6. Cửa sổ dịch vụ của AWS. 19](#_Toc496047833)

[Hình 3.7. Bảng điều khiển EC2. 19](#_Toc496047834)

[Hình 3.8. Chọn hệ điều hành cho máy chủ EC2. 20](#_Toc496047835)

[Hình 3.9. Lựa chọn cấu hình máy chủ EC2. 20](#_Toc496047836)

[Hình 3.10. Lựa chọn cấu hình chi tiết. 20](#_Toc496047837)

[Hình 3.11. Thêm dung lương bộ nhớ. 21](#_Toc496047838)

[Hình 3.12. Gắn nhãn máy chủ. 21](#_Toc496047839)

[Hình 3.13. Cấu hình bảo mật. 21](#_Toc496047840)

[Hình 3.14. Xem lại thông tin máy chủ. 22](#_Toc496047841)

[Hình 3.15. Lựa chọn kerpair cho máy chủ. 22](#_Toc496047842)

[Hình 3.16. Thông báo tạo máy chủ thành công. 22](#_Toc496047843)

[Hình 3.17. Máy chủ EC2 đã tạo. 23](#_Toc496047844)

[Hình 3.18. Thông tin chi tiết của máy chủ EC2. 23](#_Toc496047845)

[Hình 3.19. Phần mềm putty và puttygen. 23](#_Toc496047846)

[Hình 3.20. Tạo khóa truy cập EC2 với Puttygen. 24](#_Toc496047847)

[Hình 3.21. Phần mềm Putty. 24](#_Toc496047848)

[Hình 3.22. Giao diện dòng lệnh làm việc với EC2 Instance. 25](#_Toc496047849)

[Hình 3.23. Bảng cấu hình AWS Command Line Interface. 27](#_Toc496047850)

[Hình 3.24. Cấu hình IP và port . 27](#_Toc496047851)

[Hình 3.25. Chương trình ở chế độ debug. 28](#_Toc496047852)

[Hình 3.26. Giao diện trang chủ. 28](#_Toc496047853)

[Hình 3.27. Giao diện đăng nhập. 29](#_Toc496047854)

[Hình 3.28. Đăng ký thành công tài khoản test01. 29](#_Toc496047855)

[Hình 3.29. Kết quả đoạn mã 1. 30](#_Toc496047856)

[Hình 3.30. Kết quả đoạn mã 2. 31](#_Toc496047857)

[Hình 3.31. Kết quả đoạn mã 3. 31](#_Toc496047858)

# DANH MỤC BẢNG BIỂU

MỤC LỤC

[LỜI NÓI ĐẦU 1](#_Toc7877452)

[DANH MỤC TỪ VIẾT TẮT 2](#_Toc7877453)

[DANH MỤC HÌNH ẢNH 3](#_Toc7877454)

[DANH MỤC BẢNG BIỂU 5](#_Toc7877455)

[MỤC LỤC 6](#_Toc7877456)

[Chương 1: Giới thiệu 9](#_Toc7877457)

[**1.1.** **Giới thiệu về IOT** 9](#_Toc7877458)

[**1.2.** **Giới thiệu đề tài** 10](#_Toc7877459)

[**1.3.** **Mô tả đề tài và ứng dụng, công nghệ sử dụng** 11](#_Toc7877460)

[Chương 2: Thiết bị 12](#_Toc7877461)

[**2.1.** **Giới thiệu các thiết bị IOT cơ bản.** 12](#_Toc7877462)

[**2.2.** **Danh sách các thiết bị sử dụng.** 12](#_Toc7877463)

[**2.3.** **Bảng kinh phí** 12](#_Toc7877464)

[**2.4.** **Cảm biến** 13](#_Toc7877465)

[**2.5.** **Arduino (Esp8266).** 15](#_Toc7877466)

[**2.6.** **Raspberry Pi B+** 17](#_Toc7877467)

[CHƯƠNG 3: Giao thức 21](#_Toc7877468)

[**3.1.** **Raspberry Pi B+** 21](#_Toc7877469)

[CHƯƠNG 4: Ngôn ngữ lập trình 25](#_Toc7877470)

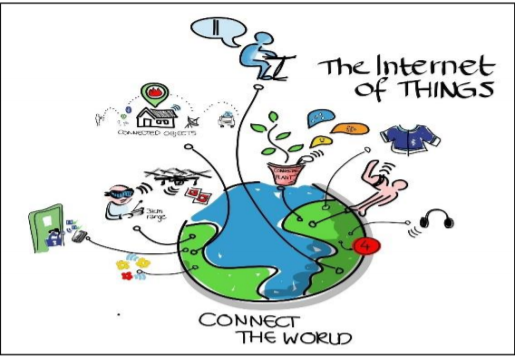
[**4.1.** **Giới thiệu các ngôn ngữ, framework** 25](#_Toc7877471)

[KẾT LUẬN 26](#_Toc7877472)

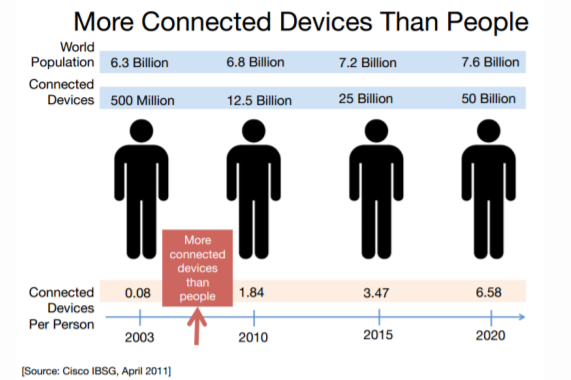
[TÀI LIỆU THAM KHẢO 27](#_Toc7877473)

# Giới thiệu

* 1. **Giới thiệu về IOT**
* Internet of Things (IoT) là thuật ngữ dùng để chỉ các đối tượng có thể được nhận biết cũng như sự tồn tại của chúng trong một kiến trúc mang tính kết nối. Đây là một viễn cảnh trong đó mọi vật, mọi con vật hoặc con người được cung cấp các định danh và khả năng tự động truyền tải dữ liệu qua một mạng lưới mà không cần sự tương tác giữa con người-với-con người hoặc con người-với-máy tính. IoT tiến hoá từ sự hội tụ của các công nghệ không dây, hệ thống vi cơ điện tử (MEMS) và Internet. Cụm từ này được đưa ra bởi Kevin Ashton vào năm 1999. Ông là một nhà khoa học đã sáng lập ra Trung tâm Auto-ID ở đại học MIT.



- "Thing" - sự vật - trong Internet of Things, có thể là một trang trại động vật với bộ tiếp sóng chip sinh học, một chiếc xe ô tô tích hợp các cảm biến để cảnh báo lái xe khi lốp quá non, hoặc bất kỳ đồ vật nào do tự nhiên sinh ra hoặc do con người sản xuất ra mà có thể được gán với một địa chỉ IP và được cung cấp khả năng truyền tải dữ liệu qua mạng lưới. IoT phải có 2 thuộc tính: một là đó phải là một ứng dụng internet. Hai là, nó phải lấy được thông tin của vật chủ.



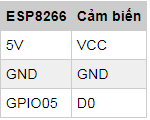
* 1. **Giới thiệu đề tài**
* Trong các ngôi nhà hiện đại ngày nay, số lượng trang thiết bị điện, điện tử đang không ngừng gia tăng. Tuy nhiên, do khác nhau về kiến trúc, việc điều khiển các thiết bị đôi khi bất cập. Thêm vào đó, việc điều khiển các thiết bị một cách thủ công với khoảng các địa lý lớn không dễ. Vì vậy, việc áp dụng các công nghệ điều khiển tự động nhằm giải quyết tương tác giữa môi trường và các thiết bị trong nhà một cách linh hoạt, dễ dàng là điều tất yếu, khái niệm nhà thông minh ra đời.
* Nhiều công nghệ đã được áp dụng khi xây dựng nhà thông minh .Tuy nhiên, sự phức tạp nằm ở chỗ các hệ thống điều khiển phải cân bằng giữa sự phức tạp của hệ thống và tính tiện dụng cho người dùng, đặc biệt là có thể được điều khiển ở bất cứ đâu, từ trong chính ngôi nhà đó hay bất kỳ nơi nào trên thế giới thông qua điện thoại hoặc internet. Công nghệ Internet of Things (IoT) đang trở lên phổ biến, hàng tỷ thiết bị được kết nối chung với nhau bằng internet.
* Với IoT mỗi đồ vật, thiết bị, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính . Bên cạnh đó, IoT có thể triển khai một mạng lưới các thực thể thông minh, có khả năng tự tổ chức và hoạt động tùy theo tình huống, môi trường, đồng thời chúng cũng có thể liên lạc với nhau để trao đổi thông tin, dữ liệu. Với khả năng định danh cao, số lượng các thực thể trong hệ thống được định danh chính xác, duy nhất, đảm bảo tốt khả năng quản lý, điều khiển của hệ thống.
  1. **Mô tả đề tài và ứng dụng, công nghệ sử dụng**
* IoT có ứng dụng rộng vô cùng, có thể kể ra một số thư như sau:
  + Quản lí chất thải
  + Quản lí và lập kế hoạch quản lí đô thị
  + Quản lí môi trường
  + Phản hồi trong các tinh huống khẩn cấp
  + Mua sắm thông minh
  + Quản lí các thiết bị cá nhân
  + Đồng hồ đo thông minh
  + Tự động hóa ngôi nhà
* Nhà thông minh là một ứng dụng trong số các ứng dụng của IoT. Nhà thông minh (tiếng Anh: home automation, smart home hoặc Intellihome) là kiểu nhà được lắp đặt các thiết bị điện, điện tử có tác dụng tự động hoá hoàn toàn hoặc bán tự động, thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử này giao tiếp với chủ nhân nhà thông qua bảng điện tử đặt trong nhà, phần mềm điện thoại di động, máy tính bảng hoặc một giao diện web.
* Trong căn nhà thông minh, đồ dùng trong nhà từ phòng ngủ, phòng khách đến toilet đều gắn các bộ điều khiển điện tử có thể kết nối với Internet và điện thoại di động, cho phép chủ nhân điều khiển vật dụng từ xa hoặc lập trình cho thiết bị ở nhà hoạt động theo lịch. Nhà thông minh ngoài ra còn có một số ứng dụng sáng tạo hơn, gồm hệ thống điều khiển giải trí tại gia – loa công suất khác nhau, hệ thống điện thoại, liên lạc nội bộ, hệ thống tưới nước... Các thành phần của hệ thống nhà thông minh bao gồm các cảm biến (như cảm biến nhiệt độ, cảm biến ánh sáng và cảm biến khí gas), các bộ điều khiển hoặc máy chủ và các thiết bị chấp hành khác. Nhờ hệ thống cảm biến, các bộ điều khiển và máy chủ có thể theo dõi các trạng thái bên trong ngồi nhà để đưa ra các quyết định điều khiển các thiết bị chấp hành một cách phù hợp nhằm đảm bảo môi trường sống tốt nhất cho con người.

# Chương 2: Thiết bị

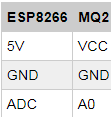
* 1. **Giới thiệu các thiết bị IOT cơ bản.**
* Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số...
* Raspberry Pi là từ để chỉ các máy tính chỉ có một board mạch (hay còn gọi là máy tính nhúng) kích thước chỉ bẳng một thẻ tín dụng, được phát triển tại Anh bởi Raspberry Pi Foundation với mục đích ban đầu là thúc đẩy việc giảng dạy về khoa học máy tính cơ bản trong các trường học và các nước đang phát triển.
  1. **Danh sách các thiết bị sử dụng.**
* Arduino (Esp8266)
* Raspberry Pi B+
* Cảm biến:
* Cảm biến ánh sáng quang trở
* Cảm biến khí gas MQ2
* Cảm biến nhiệt độ, độ ẩm DHT11
  1. **Bảng kinh phí**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên sản phẩm** | **Số lượng** | **Đơn giá vnđ** |
| 1 | Raspberry Pi 3 Model | 1 | 1.050.000 |
| 2 | ESP8266 – IoT Wifi Uno | 1 | 250.000 |
| 3 | Nguồn USB 5V 2.5A Cáp Micro | 1 | 125.000 |
| 4 | Vỏ Case cho Raspberry | 1 | 110.000 |
| 5 | Cảm biến khí ga MQ2 | 1 | 50.000 |
| 6 | Breadboard 830 tie – points MB – 102 | 1 | 45.000 |
| 7 | Cảm biến số nhiệt độ, độ ẩm DHT11 | 1 | 35.000 |
| 8 | Cảm biến ánh sáng quang trở | 1 | 18.000 |
| 9 | Cable Micro USB – B HTC | 1 | 15.000 |
| 10 | Bộ 20 dây cắm | 1 | 12.000 |
| 11 | Bộ 20 dây cắm testboard Female – Male 20cm | 1 | 12.000 |
| 12 | Bộ 20 dây cắm testboard Male – Male 20cm | 1 | 12.000 |
| 13 | LED chân cắm 5mm | 1 | 4.000 |
| **TỔNG CỘNG** | | | 1.738.000 |

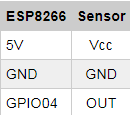
* 1. **Cảm biến**
     1. **Cảm biến ánh sáng quang trở**
* Cảm biến ánh sáng quang trở phát hiện cường độ ánh sáng, sử dụng bộ cảm biến photoresistor loại nhạy cảm, cho tín hiệu ổn định, rõ ràng và chính xác hơn so với quang trở.
* Ngõ ra D0 trên cảm biến được dùng để xác định cường độ sáng của môi trường, khi ở ngoài sáng, ngõ ra D0 là giá trị 0, khi ở trong tối, ngõ ra D0 là 1. Trên cảm biến có 1 biến trở để điều chỉnh cường độ sáng phát hiện, khi văn cùng chiều kim đồng hồ thì sẽ làm giảm cường độ sáng nhận biết của cảm biến, tức là môi trường phải ít sáng hơn nữa thì cảm biến mới đọc gía trị digital là 1.
* Thông số kỹ thuật:
* Điện áp làm việc: 3.3 ~ 5VDC
* Output: Digital
* Có thể điều chỉnh cường độ ánh sáng phát hiện bằng biến trở gắn trên cảm biến
* Kích thước: 3.2cm x 1.4cm
* Kết nối



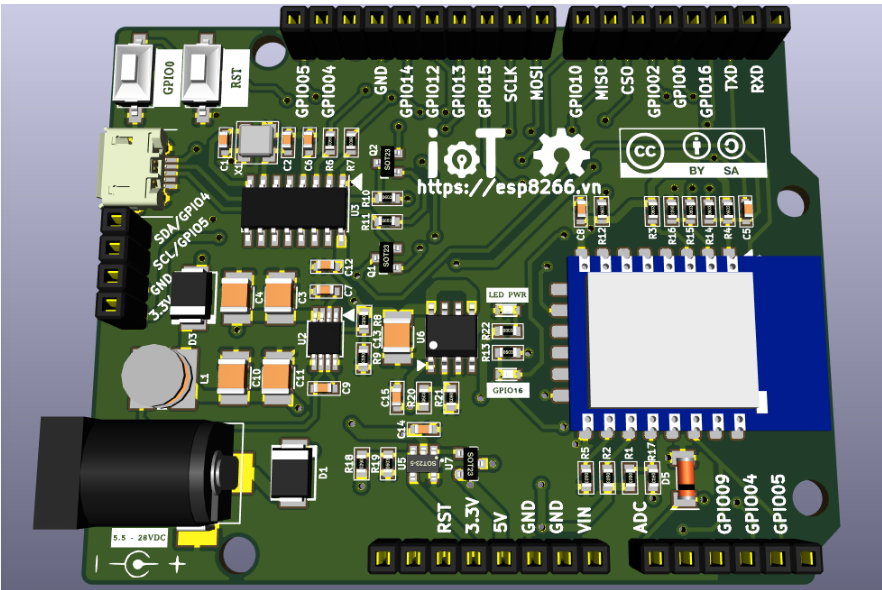
* + 1. **Cảm biến khí gas MQ2**
* Cảm biến khí ga MQ2 là một trong những loại cảm biến được sử dụng để nhận biết: LPG, i-butan, Propane, Methane , Alcohol, Hydrogen, Smoke và khí ga. Được thiết kế với độ nhạy cao, thời gian đáp ứng nhanh. Gía trị đọc được từ cảm biến sẽ được đọc về từ chân Analog của vi điều khiển.
* Thông số kỹ thuật:
* Nguồn hoạt động: 5VDC
* Dòng: 150mA
* Tính hiệu tương tự (analog)
* Hoạt động trong thời gian dài, ổn định
* Kết nối:



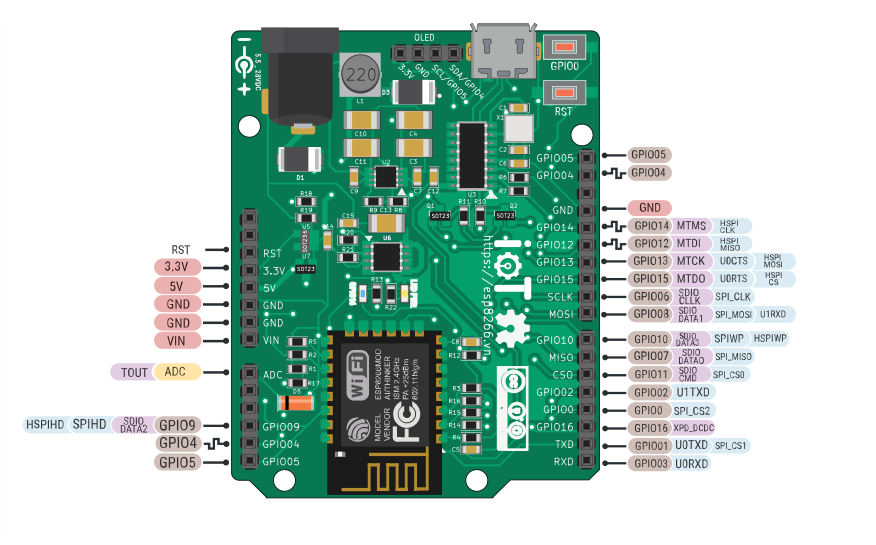
* + 1. **Cảm biến số nhiệt độ, độ ẩm DHT11**
* DHT11 Là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.
* Thông số kỹ thuật:
* Điện áp hoạt động: 3V - 5V (DC)
* Dãi độ ẩm hoạt động: 20% - 90% RH, sai số ±5%RH
* Dãi nhiệt độ hoạt động: 0°C ~ 50°C, sai số ±2°C
* Khoảng cách truyển tối đa: 20m
* Kết nối:



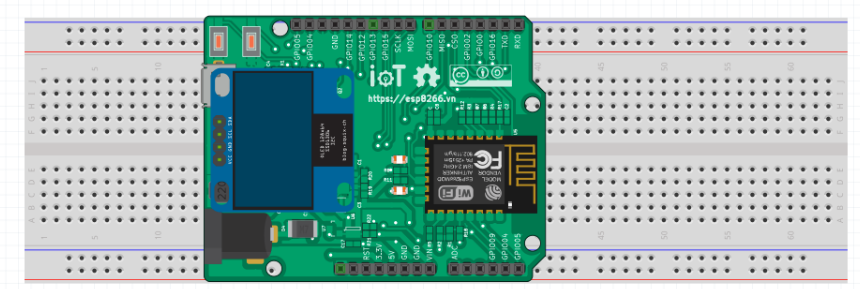
* 1. **Arduino (Esp8266).**
     1. **Thông số kỹ thuật**
* ESP8266 - IoT WiFi Uno với thiết kế tương thích với Arduino Uno, có MCU là module ESP-12E hổ trợ firmware chương trình Arduino giúp dễ dàng lập trình. Đặc biệt mạch có thể được lập trình để chạy các ứng dụng thu thập và điều khiển dữ liệu qua wifi.
* Đặc điểm:
* CPU ESP8266
* Hỗ trợ kết nối WiFi
* Tương thích Arduino UNO
* Có thể lập trình được bằng C/C++, Arduino IDE, Micropython, NodeMCU - Lua
* Nguồn 9-24V hay 5V từ USB
* 11 IO, 1 Analog in
* 4 Mbytes Flash
* Module ESP-12F chỉ 3.3VDC ( tối đa 3.6 VDC )
* Ảnh 3D



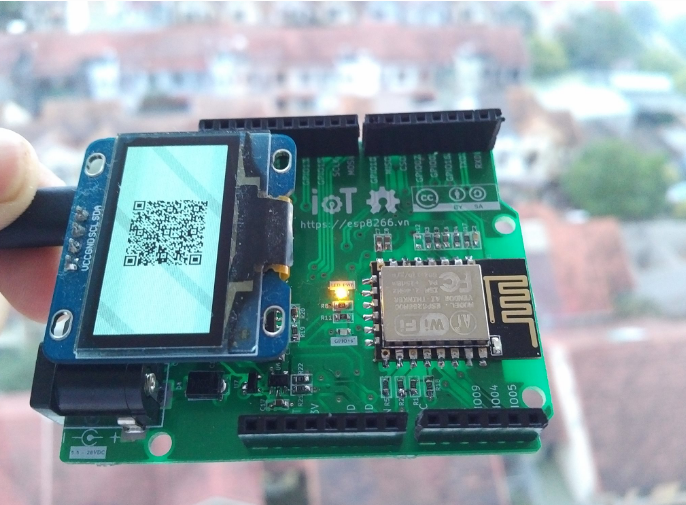
* Sơ đồ pin Digital



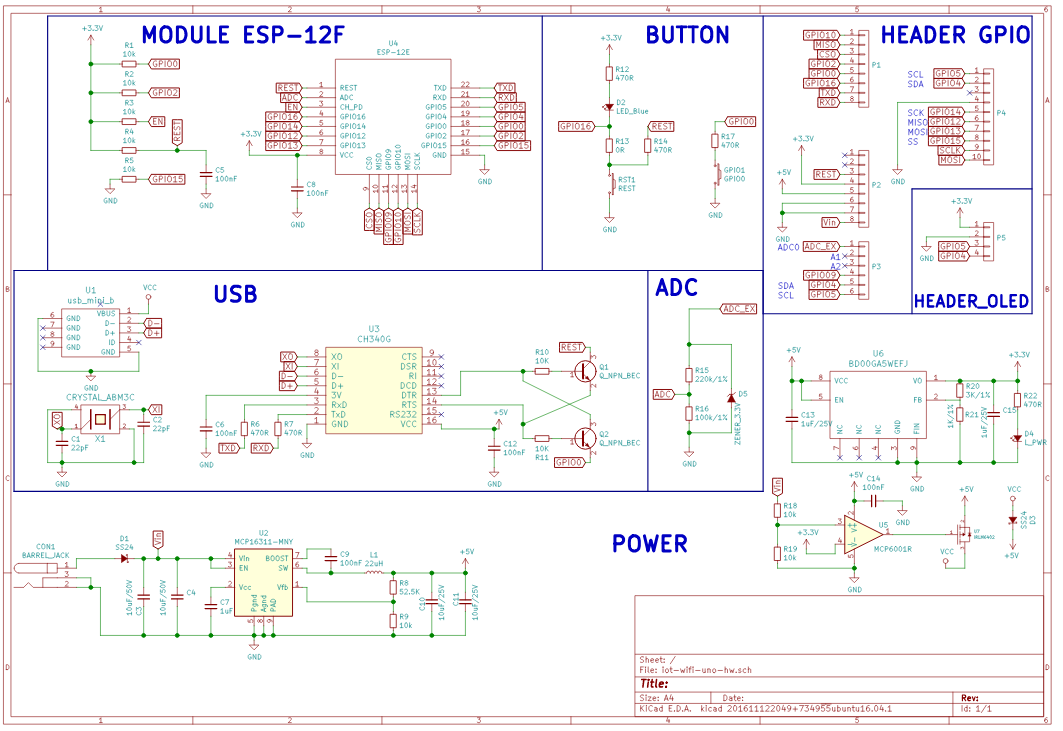
* Fritzing part



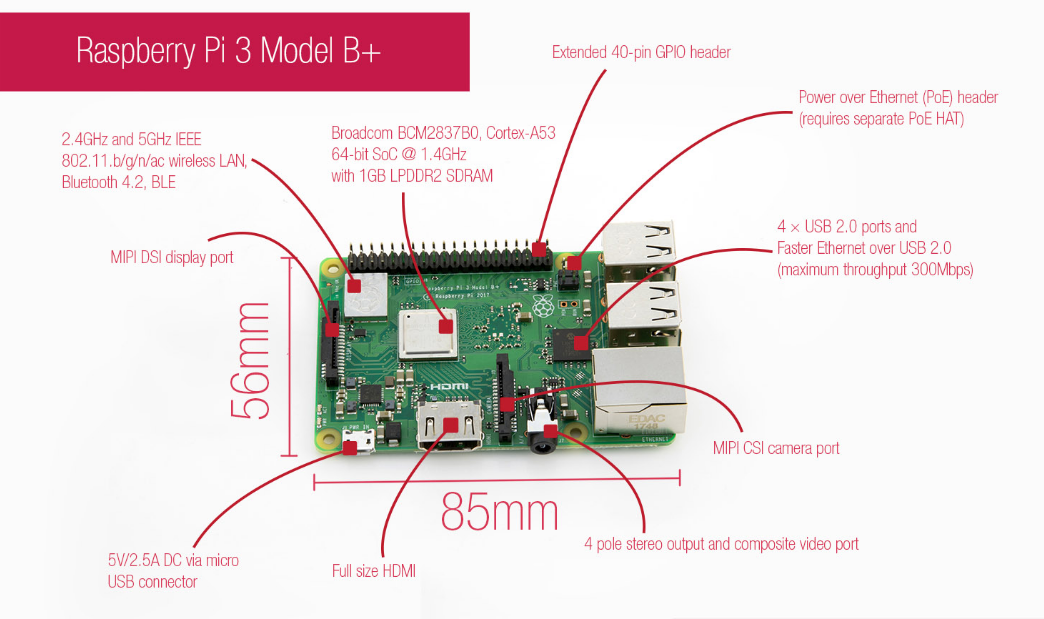
* Thực tế

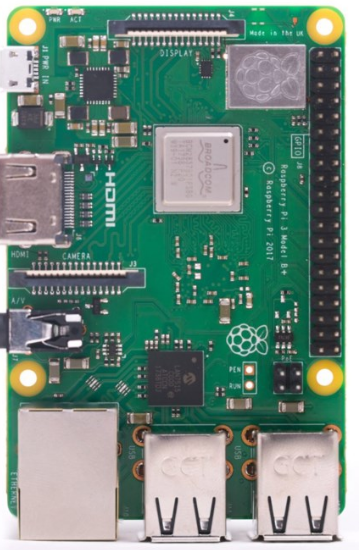


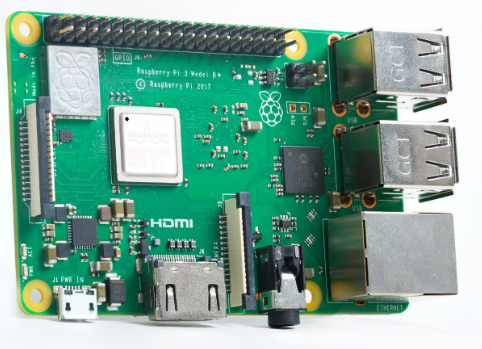
* Mạch nguyên lý

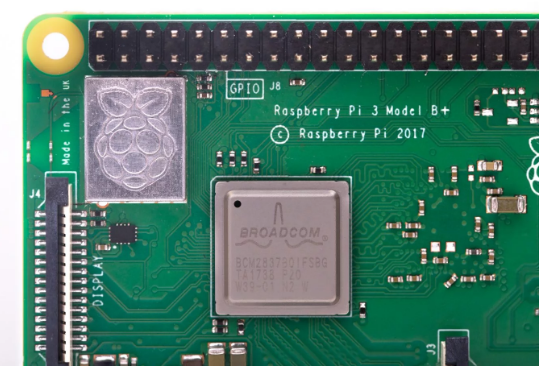


* 1. **Raspberry Pi B+**
     1. **Thông số kỹ thuật**
* Raspberry Pi 3 Model B+ là sản phẩm mới nhất trong gia đình Raspberry Pi, nổi bật với chip 4 nhân 64-bit có tốc độ 1.4GHz – nhanh nhất từ trước đến nay! Phiên bản này hỗ trợ Wifi Dual-band 2.4GHz và 5GHz, Bluetooth 4.2/Bluetooth Low Energy, cổng Ethernet tốc độ cao (300Mbps) và Power over Ethernet (PoE) thông qua PoE HAT.
* Bên cạnh những thay đổi đó, kích thước bề ngoài cũng như phần mềm của Raspberry Pi 3 Model B+ vẫn được giữ nguyên, vì vậy bạn hoàn toàn có thể sử dụng lại các phụ kiện cũ như vỏ, thẻ nhớ, adapter (nguồn điện) hoặc các phần mềm đang chạy trên Raspbian/Ubuntu hiện tại. Tuy nhiên, về hệ điều hành thì các bạn cần tải bản mới nhất để có thể hỗ trợ được bo mạch mới nhất này.
* Thông số kỹ thuật:
* Vi xử lý: Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz
* RAM: 1GB LPDDR2 SDRAM
* Kết nối: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps).
* Hỗ trợ: 40-pin GPIO, 4 cổng USB2.0
* Video và âm thanh: 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân.
* Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
* Lưu trữ: MicroSD
* Điện áp hoạt động: 5V/2.5A DC cổng microUSB, 5V DC trên chân GPIO, Power over Ethernet (PoE) (yêu cầu thêm PoE HAT).
* Một số hình ảnh:











# CHƯƠNG 3: Giao thức

* 1. **Raspberry Pi B+**
     1. **Giới thiệu các giao thức trong IOT**

#### Bluetooth

* Một công nghệ giao tiếp truyền thông trong khoảng cách ngắn vô cùng quan trọng, đó là Bluetooth. Hiện nay, bluetooth xuất hiện hầu hết ở các thiết bị như máy tính, điện thoại/ smartphone,….và nó được dự kiến là chìa khóa cho các sản phẩm IoT đặc biệt, cho phép giao tiếp thiết bị với các smartphone – một “thế lực hùng hậu” hiện nay.
* Hiện nay, BLE – Bluetooth Low Energy – hoặc Bluethooth Smart là một giao thức được sử dụng đáng kể cho các ứng dụng IoT. Quan trọng hơn, cùng với một khoảng cách truyền tương tự như Bluetooth, BLE được thiết kế để tiêu thụ công suất ít hơn rất nhiều.

#### Wifi

* Wifi (là viết tắt từ Wireless Fidelity hay mạng 802.11) là hệ thống mạng không dây sử dụng sóng vô tuyến, cũng giống như điện thoại di đông, truyền hình và radio. Kết nôi Wifi thường là sự lựa chọn hàng đầu của rất nhiều kỹ sư giải pháp bởi tính thông dụng và kinh tế của hệ thống wifi và mạng LAN với mô hình kết nối trong một phạm vi địa lý có giới hạn.
* Các sóng vô tuyến sử dụng cho WiFi gần giống với các sóng vô tuyến sử dụng cho thiết bị cầm tay, điện thoại di động và các thiết bị khác. Nó có thể chuyển và nhận sóng vô tuyến, chuyển đổi các mã nhị phân 1 và 0 sang sóng vô tuyến và ngược lại. Tuy nhiên, sóng WiFi có một số khác biệt so với các sóng vô tuyến khác ở chỗ: Chúng truyền và phát tín hiệu ở tần số 2.4 GHz hoặc 5 GHz. 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, đa số các thiết bị wifi đều tuân theo chuẩn 802.11n, được phát ở tần số 2.4Ghz và đạt tốc độ xử lý tối đa 300Megabit/giây

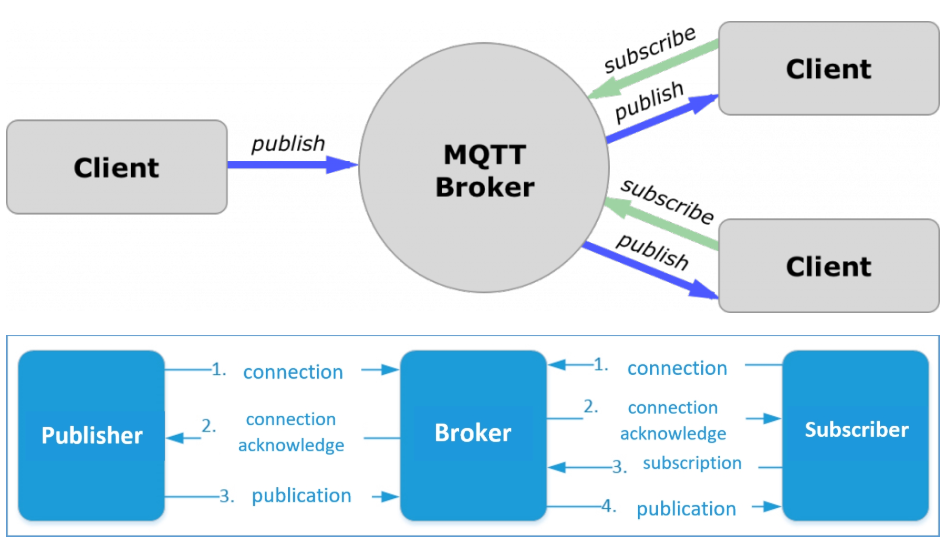
#### NFC

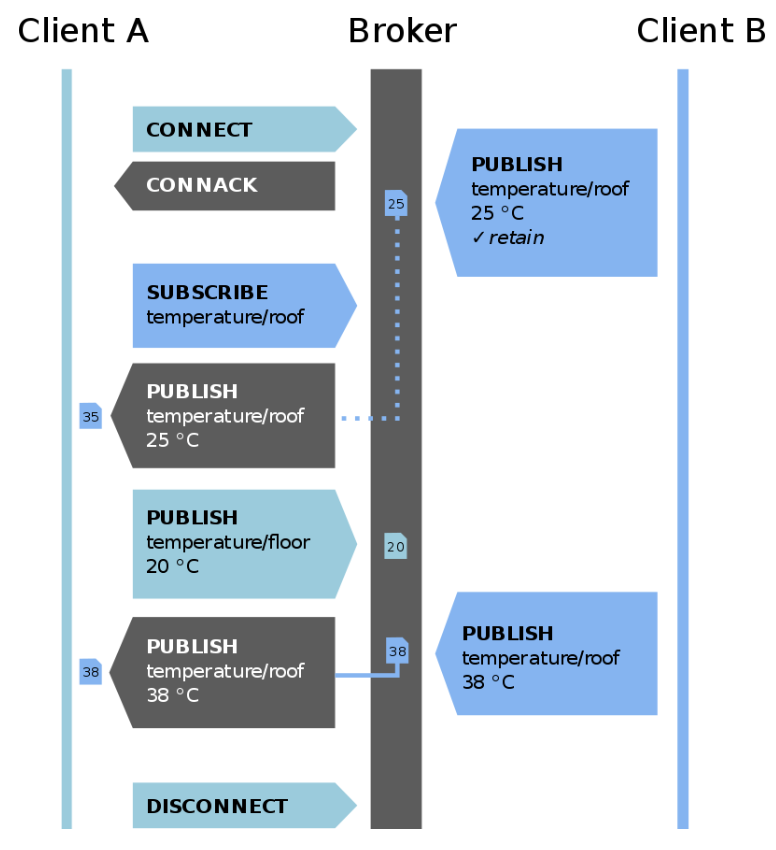
* NFC (Near-Field Communications) là công nghệ kết nối không dây trong phạm vi tầm ngắn trong khoảng cách 4 cm. Công nghệ này sử dụng cảm ứng từ trường để thực hiện kết nối giữa các thiết bị (smartphone, tablet, loa, tai nghe …) khi có sự tiếp xúc trực tiếp (chạm)
* Khi hai thiết bị đều có kết nối NFC, bạn có thể chạm chúng vào nhau để kích hoạt tính năng này và nhanh chóng truyền tập tin gồm danh bạ, nhạc, hình ảnh, video, ứng dụng hoặc địa chỉ website… Ở các nước phát triển, NFC còn được xem là chiếc ví điện tử khi có thể thanh toán trực tuyến, tiện lợi và nhanh chóng.
* Ngoài việc giúp truyền tải dữ liệu như trên thì NFC còn mở rộng với những công dụng ví dụ như bạn đến quán café có một thẻ NFC để trên bàn, trong thẻ này đã cài đặt sẵn wifi, thông tin của quán…lúc này bạn lấy chiếc điện thoại chạm vào NFC này thì máy sẽ bật tất cả tính năng được cài sẵn trong thẻ đó mà không cần phải nhờ gọi nhân viên. Hoặc tiên tiến hơn thì sau này có thể khi mua đồ trong siêu thị lớn thì quẹt NFC của điện thoại để thanh toán tiền luôn.

#### Serial

#### MQTT

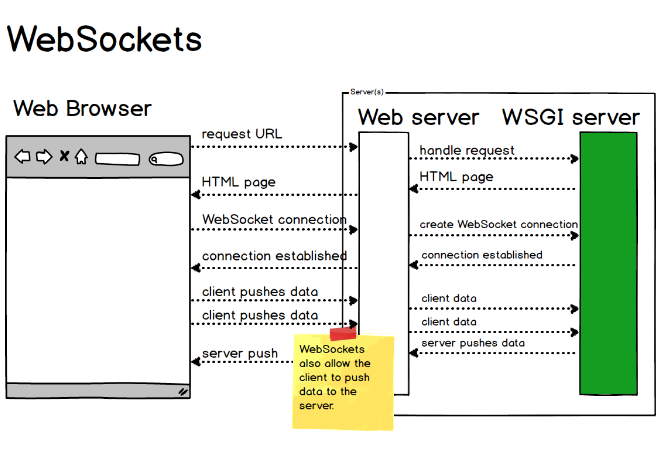
* Giao thức MQTT hoạt động theo mô hình publish/subcribe . Trong mô hình hoạt động của giao thức MQTT, bên gửi dữ liệu được gọi là thiết bị publisher và bên nhận được gọi là thiết bị subcriber. Các thiết bị này còn được gọi chung là MQTT client. Bên cạnh các thành phần publisher và subcriber thì giao thức MQTT sử dụng một thiết bị trung gian đóng vai trò trung chuyển các gói tin gọi là MQTT Broker. Để nhận được thông tin thì các thiết bị subscriber sẽ gửi thông báo yêu cầu cung cấp thông tin (subcribe) theo một chủ đề (topic) nhất định đến MQTT broker. Chủ đề là một cái tên để phân biệt các kênh truyền từ các thiết bị publisher khác nhau và có dạng cây A/B/C/.../X/Y.
* Khi một thiết bị publisher gửi thông báo cung cấp thông tin với chủ đề nhất định lên MQTT Broker thì MQTT Broker sẽ gửi thông báo đó đến các thiết bị subcriber đã yêu cầu chủ đề đó. MQTT Broker không chỉ đóng vai trò chuyển tiếp các thông báo từ publisher tới các subcriber dựa trên chủ đề (topic) của thông báo mà còn đóng vai trò các vai trò khác như duy trì các kết nối giữa các publisher và subcriber tới MQTT Broker, phát 8 muốn kết nối với một hệ thống khác nằm ngoài miền sẽ phải thông qua một Gateway Interface





#### WebSocket

* Websocket là giao thức chuẩn cho trao đổi dữ liệu hai chiều giữa client và server. Giao thức WebSocket không chạy trên HTTP, thay vào đó nó thực hiện trên giao thức TCP.
* WebSocket là một kiểu Reverse Ajax
* Người ta thường dùng Websocket thay vì HTTP cho những trường hợp yêu cầu real time (thời gian thực). Ví dụ bạn muốn hiện thị biểu đồ, chỉ số chứng khoán, web chat… thì không thể gửi lệnh AJAX liên tiếp tới server để lấy dữ liệu mới rồi cập nhật lên màn hình, như thế sẽ tốn nhiều tài nguyên, traffic.
* Gói tin của WebSockets nhẹ hơn HTTP rất nhiều, giảm độ trễ của network lên đến 3 lần, không cần phải gửi request liên tiếp như HTTP.

[](https://stackjava.com/wp-content/uploads/2018/04/Screenshot_1-3.png)

# CHƯƠNG 4: Ngôn ngữ lập trình

* 1. **Giới thiệu các ngôn ngữ, framework**
     1. **Python**
* Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa 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.
* Python có cho mọi hệ điều hành: Windows, Linux/Unix, OS/2, Mac, Amiga, và những hệ điều hành khác. Thậm chí có cả những phiên bản chạy trên .NET, máy ảo Java, và điện thoại di động (Nokia Series 60). Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.
* Cài đặt Python dùng giấy phép nguồn mở nên được sử dụng và phân tối tự do, ngay cả trong việc thương mại. Giấy phép Python được quản lý bởi Python Software Foundation.
  + 1. **Angular**
* Angular là một Framwork mạnh mẽ được phát triển bởi Google. Nó được sử dụng để xây dựng các ứng dụng tương tác với người dùng thông qua giao diện Web. Angular cho cung cấp cho lập trình viên nền tảng công nghệ hiện đại dựa trên kiến trúc Hướng dịch vụ SOA và cho phép tích hợp với hàng loạt các nền tảng công nghệ lập trình hiện đại như .Net, Java, Php …
* Với đặc trưng mạnh mẽ của các phiên bản trước kia như Angular 1.x, từ phiên bản Angular 2 (Hiện nay là phiên bản Angular 5) trở đi , nền tảng công nghệ có sự thay đổi lớn, ở đây ngôn ngữ lập trình được thực hiện đó là TypeScrip (Ngôn ngữ lập trình hiện đại và hướng đối tượng), và framework này cũng phân chia việc phát triển ứng dụng thành nhiều các module riêng biệt, điều này giúp cho lập trình viên có thể tiếp cận theo cách hướng mô hình một cách dễ dàng hơn.
  + 1. **Database: Postgresql**
* PostgreSQL sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi fault-tolerant giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, PostgreSQL cũng có khả năng mở rộng tuyệt vời. Ví dụ, bạn có thể định nghĩa các kiểu dữ liệu riêng của bạn, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu!
* PostgreSQL tuân theo tiêu chuẩn SQL nhưng không mâu thuẫn với các tính năng truyền thống hay có thể dẫn đến các quyết định kiến trúc gây hại. Nhiều tính năng theo tiêu chuẩn SQL được hỗ trợ, tuy nhiên đôi khi có thể có cú pháp hoặc hàm hơi khác một chút.
  + 1. **Arduino: C/C++**
* Ngôn ngữ C là ngôn ngữ cộng sinh vì nó sở hữu tính di động và năng suất. C cho phép người dùng truyền đạt được ý tưởng của chương trình một cách tốt nhất. Đó là lý do tại sao nó được gọi là cha đẻ của ngôn ngữ lập trình. Các nhà phát triển chính sử dụng C là Microsoft Windows, Linux, Mac.
* Ngôn ngữ lập trình này hơi khó nhưng được giới chuyên gia đánh giá cao. Nhiều công ty lớn sử dụng ngôn ngữ này vì nó giúp kỹ sư kiểm soát nhiều hơn. C++ là một hình thức nâng cấp của ngôn ngữ lập trình C. Điều khác biệt giữa nó và C là nó cho phép nhiều đối tượng được tạo trong mã.

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO