TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRƯƠNG MINH TRÍ – 51600095**

**LÊ TIẾN THỊNH - 51600082**

**XÂY DỰNG HỆ THỐNG IOT CẢNH BÁO SỚM VÀ PHÁT HIỆN CHÁY CHO NGƯỜI QUA APP MOBILE**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KHOA HỌC MÁY TÍNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRƯƠNG MINH TRÍ – 51600095**

**LÊ TIẾN THỊNH - 51600082**

**XÂY DỰNG HỆ THỐNG IOT CẢNH BÁO SỚM VÀ PHÁT HIỆN CHÁY CHO NGƯỜI QUA APP MOBILE**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KHOA HỌC MÁY TÍNH**

Người hướng dẫn

**TS. Trương Đình Tú**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Trong thời gian làm đồ án, chúng em đã nhận được rất nhiều sự giúp đỡ và hỗ trợ nhiệt tình từ các giáo viên trong khoa, khuyến khích chúng em vượt qua tất cả những khó khăn nghiêm trọng và sau đó chúng em đã hoàn thành khoảng thời gian khó khăn nhưng đầy ý nghĩa này.

Xin chân thành cảm ơn thầy: Tiến sĩ Trương Đình Tú, người đã cho bọn em kiến thức về internet of things (IoT), phần mềm Arduino và cách tư duy trong lập trình để có thể hoàn thành đồ án một cách hoàn chỉnh và tối ưu nhất.

*TP. Hồ Chí Minh, ngày tháng năm 20*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**PHIẾU ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tên giảng viên hướng dẫn:

Ý kiến nhận xét:

Điểm tổng theo phiếu đánh giá rubrik:

*TP. Hồ Chí Minh, ngày tháng năm 20*

*Giảng viên hướng dẫn*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS Trương Đình Tú. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong luận văn còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào nhóm chúng em xin hoàn toàn chịu trách nhiệm về nội dung Báo cáo Dự án CNTT 2 của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng 02 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

**XÂY DỰNG HỆ THỐNG IOT CẢNH BÁO SỚM VÀ PHÁT HIỆN CHÁY CHO NGƯỜI QUA APP MOBILE**

TÓM TẮT

Xây dựng hệ thống, thiết bị dự báo, phát hiện cháy nổ trong nhà, căn hộ và gửi thông báo đến chủ nhà. Thiết bị gồm có 3 cảm biến chính: Nhiệt độ và độ ẩm DHT11, cảm biết lửa, cảm biến ga MQ-2, vì thế có thể dễ dàng phát hiện được trình trạng trong nhà bằng cách phân tích dữ liệu thu thập được một cách tổng quát từ đó đưa ra thông báo chính xác nhất cho người dùng. Nó sẽ giúp mọi người tránh được những sự cố không đáng có để bảo vệ bản thân và gia đình cũng như tài sản của gia đình, công ty.

**MỤC MỤC**

[LỜI CẢM ƠN i](#_Toc126932759)

[TÓM TẮT iv](#_Toc126932760)

[DANH MỤC CÁC HÌNH VẼ vii](#_Toc126932761)

[DANH MỤC CÁC BẢNG BIỂU viii](#_Toc126932762)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc126932763)

[**CHƯƠNG 1 – GIỚI THIỆU 1**](#_Toc126932764)

[1.1. Giới thiệu về mạng lưới vạn vật kết nối internet (IoT) 1](#_Toc126932765)

[*1.1.1. IoT là gì? 1*](#_Toc126932766)

[*1.1.2. Các đặt tính cơ bản của IoT. 2*](#_Toc126932767)

[*1.1.3. Cấu trúc của một hệ thống IOT. 2*](#_Toc126932768)

[*1.1.4. IoT hoạt động như thế nào? 2*](#_Toc126932769)

[*1.1.5. Ứng dụng của IoT. 3*](#_Toc126932770)

[1.2. NodeMCU ESP8266 và các cảm biến 4](#_Toc126932771)

[*1.2.1. Giới thiệu về nodeMCU. 4*](#_Toc126932772)

[*1.2.2. Làm sao để sử dụng chip NodeMCU esp8266? 5*](#_Toc126932773)

[*1.2.3. Làm thế nào để nạp code cho NODEMCU bằng ARDUINO. 5*](#_Toc126932774)

[*1.2.4. Giới thiệu về các cảm biến 5*](#_Toc126932775)

[1.3. Giới thiệu về giao diện sử dụng ARDUINO IDE 8](#_Toc126932776)

[*1.3.1. Arduino IDE là gì? 8*](#_Toc126932777)

[*1.3.2. Cổng Com và Serial Monitor. 8*](#_Toc126932778)

[*1.3.3. Cách thức giao tiếp. 9*](#_Toc126932779)

[*1.3.4. Cài đặt board ESP8266 trên Arduino IDE 9*](#_Toc126932780)

[1.4. Giới thiệu về hệ cơ sở dữ liệu FIREBASE. 9](#_Toc126932781)

[*1.4.1. Firebase là gì? 9*](#_Toc126932782)

[*1.4.2. Các tính năng của Firebase 10*](#_Toc126932783)

[*1.4.3. Cách đăng ký và sử dụng 10*](#_Toc126932784)

[**CHƯƠNG 2 – TÌNH HUỐNG VÀ GIẢI PHÁP 12**](#_Toc126932785)

[2.1. Xác định ý tưởng 12](#_Toc126932786)

[*2.1.1. Tình huống hiện tại 12*](#_Toc126932787)

[*2.1.2. Giải pháp 12*](#_Toc126932788)

[*2.1.3. Cách thức thực hiện 12*](#_Toc126932789)

[2.2 Các lắp ráp và cài đặt hệ thống báo cháy 12](#_Toc126932790)

[*2.2.1. Sơ đồ tổng quát 12*](#_Toc126932791)

[*2.2.2. Xây dựng cơ sở dữ liệu. 14*](#_Toc126932792)

[*2.2.3. Cài đặt thư viện cho sensor 15*](#_Toc126932793)

[*2.2.4. NodeMCU gửi dữ liệu 15*](#_Toc126932794)

[*2.2.5. NodeMCU nhận dữ liệu 18*](#_Toc126932795)

[2.3. Xây dựng ứng dụng WebApp hiển thị thông tin. 20](#_Toc126932796)

[*2.3.1. Đặt vấn đề 20*](#_Toc126932797)

[*2.3.2. Xây dựng giao diện đăng nhập và đăng ký. 21*](#_Toc126932798)

[*2.3.3. Xây dựng giao diện hiển thị thông số trong căn nhà 21*](#_Toc126932799)

[*2.3.4. Mô tả code 21*](#_Toc126932800)

[**CHƯƠNG 3 – KẾT LUẬN 23**](#_Toc126932801)

[3.1. Những kết quả đã đạt được 23](#_Toc126932802)

[3.2. Những vấn đề chưa làm được và hướng phát triển trong tương lai 23](#_Toc126932803)

[TÀI LIỆU THAM KHẢO 24](#_Toc126932804)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 : Internet of things 8](#_Toc126879994)

[Hình 2: Cách hoạt động của IoT 10](#_Toc126879995)

[Hình 3: Ứng dụng của IoT 10](#_Toc126879996)

[Hình 4: Thiết bị NodeMCU 12](#_Toc126879997)

[Hình 5: Cảm biến nhiệt độ và độ ẩm DHT11 13](#_Toc126879998)

[Hình 6:Cảm biến lửa 14](#_Toc126879999)

[Hình 7: Cảm biến khí MQ-2 15](#_Toc126880000)

[Hình 8: Giao diện của Arduino IDE 16](#_Toc126880001)

[Hình 9: Các tính năng của Firebase 18](#_Toc126880002)

[Hình 10: Giao diện tạo Firebase project 19](#_Toc126880003)

[Hình 11: Giao diện trang chủ Firebase project 19](#_Toc126880004)

[Hình 12: Sơ đồ tổng quát 21](#_Toc126880005)

[Hình 13: Sản phẩm thực tế 22](#_Toc126880006)

[Hình 14: Cơ sở dữ liệu firebase 23](#_Toc126880007)

[Hình 15. Include các thư viện đã cài đặt 24](#_Toc126880008)

[Hình 16. Định nghĩ cổng D1 gắn với DHT11 24](#_Toc126880009)

[Hình 17. Định nghĩa cổng A0 gắn cảm biến khí gas và D2 gắn cảm biến lửa 25](#_Toc126880010)

[Hình 18. Khai báo mạng wifi và Firebase 25](#_Toc126880011)

[Hình 19: cài đặt cho NodeMCU gửi dữ liệu 26](#_Toc126880012)

[Hình 20: Quá trình xử lý của NodeMCU gửi dữ liệu Flame sensor 27](#_Toc126880013)

[Hình 21: Quá trình xử lý của NodeMCU gửi dữ liệu DHT11 sensor 27](#_Toc126880015)

[Hình 22: Quá trình xử lý của NodeMCU gửi dữ liệu MQ-2 sensor 27](#_Toc126880017)

[Hình 23: Quá trình xử lý của NodeMCU tính mức độ cảnh báo 27](#_Toc126880018)

[Hình 24: Cấu hình và cài đặt NodeMCU nhận dữ liệu 28](#_Toc126880019)

[Hình 25: Quá trình xử lý của NodeMCU sau khi nhận dữ liệu 29](#_Toc126880020)

[Hình 26. Giao diện hiển thị thông tin web 31](#_Toc126880021)

[Hình 26. Tải và cài đặt thư viện Firebase JavaScript 32](#_Toc126880022)

[Hình 27. Lấy dữ liệu từ firebase realtime 32](#_Toc126880023)

[Hình 28. Hiển thị notification cảnh báo khi phát hiện lửa 33](#_Toc126880025)

DANH MỤC CÁC BẢNG BIỂU

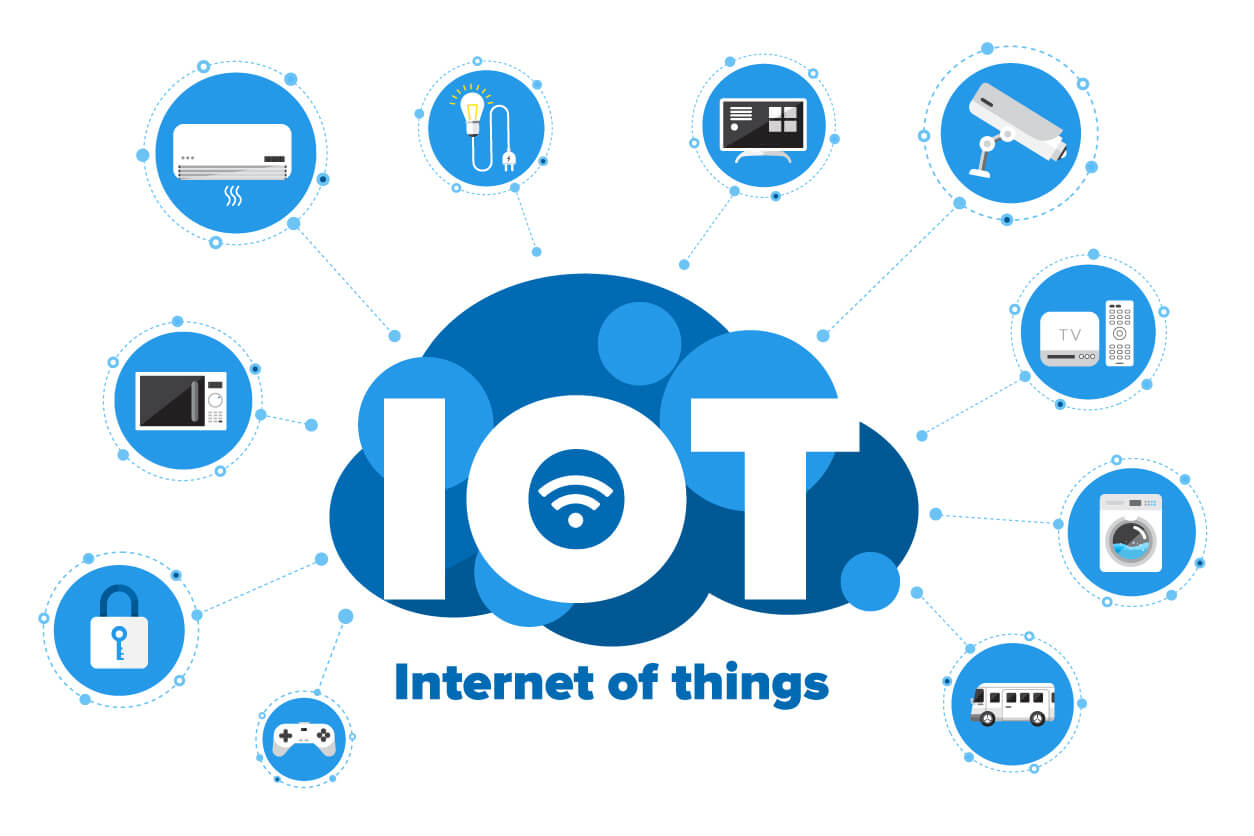
DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CSS  CDN  HTML  IoT | Cascading Style Sheets  Content Delivery Network  Hypertext Markup Language  Internet of things |
| IDE | Integated Development Environment |
| NTC  SDK | Negative Temperature Coefficient  Software Development Kit |

CHƯƠNG 1 – GIỚI THIỆU

1.1. Giới thiệu về mạng lưới vạn vật kết nối internet (IoT)

1.1.1. IoT là gì?

Mạng lưới vạn vật kết nối Internet viết tắt là IoT (tiếng Anh: Internet of Things) là một kịch bản của thế giới, khi mà mỗi đồ vật, 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. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

Hình 1 : Internet of things

IoT là tất cả các thiết bị có thể kết nối với nhau. Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại… Các thiết bị có thể là điện thoại thông minh, máy pha cafe, máy giặt, tai nghe, bóng đèn, và nhiều thiết bị khác.

1.1.2. Các đặt tính cơ bản của IoT.

Tính kết nối liên thông(interconnectivity): với IoT, bất cứ điều gì cũng có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.

Những dịch vụ liên quan đến “Things”: hệ thống IoT có khả năng cung cấp các dịch vụ liên quan đến “Things”, chẳng hạn như bảo vệ sự riêng tư và nhất quán giữa Physical Thing và Virtual Thing. Để cung cấp được dịch vụ này, cả công nghệ phần cứng và công nghệ thông tin (phần mềm) sẽ phải thay đổi.

Tính không đồng nhất: Các thiết bị trong IoT là không đồng nhất vì nó có phần cứng khác nhau, và network khác nhau. Các thiết bị giữa các network có thể tương tác với nhau nhờ vào sự liên kết của các network.

Thay đổi linh hoạt: Status của các thiết bị tự động thay đổi, ví dụ, ngủ và thức dậy, kết nối hoặc bị ngắt, vị trí thiết bị đã thay đổi, và tốc độ đã thay đổi… Hơn nữa, số lượng thiết bị có thể tự động thay đổi.

Quy mô lớn: Sẽ có một số lượng rất lớn các thiết bị được quản lý và giao tiếp với nhau. Số lượng này lớn hơn nhiều so với số lượng máy tính kết nối Internet hiện nay. Số lượng các thông tin được truyền bởi thiết bị sẽ lớn hơn nhiều so với được truyền bởi con người.

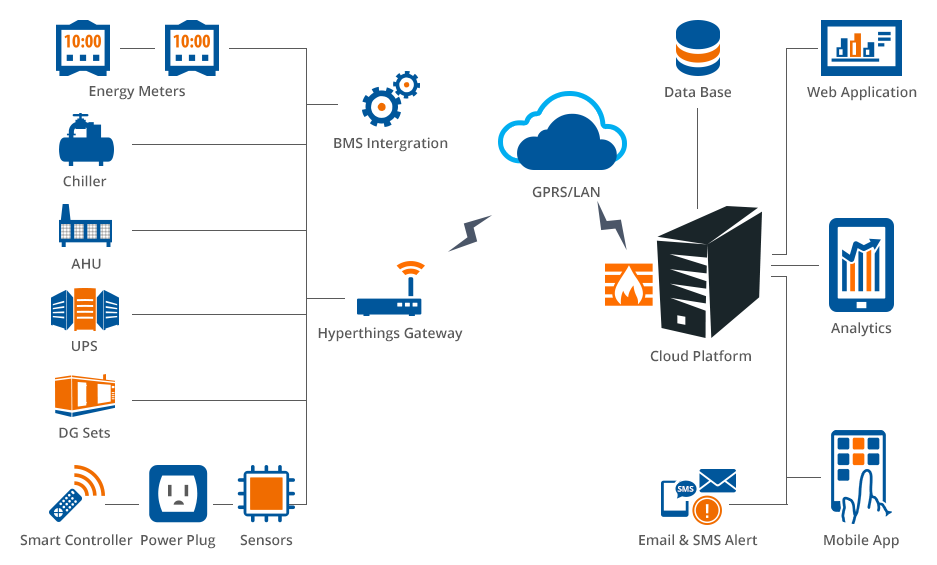
1.1.3. Cấu trúc của một hệ thống IOT.

Một hệ thống IoT sẽ bao gồm 4 thành phần chính đó là:

* Things: Thiết bị
* Gateway: Tạm kết nối
* Network and cloud: Hạ tầng mạng
* Service-creation and Solution Layers: Bộ phân tích và xử lý dữ liệu

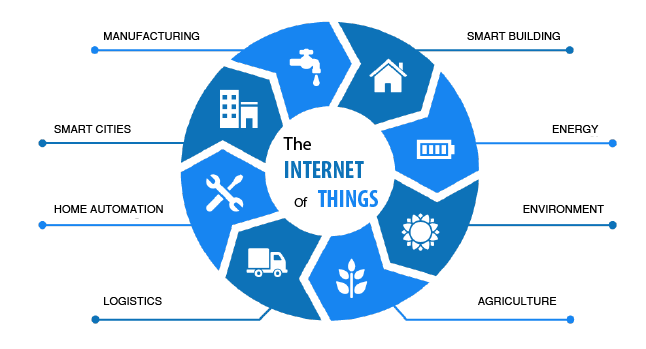
1.1.4. IoT hoạt động như thế nào?

Điều kì diệu ở đây đó chính là cảm biến. Các thiết bị cần kết nối phải được tích hợp một chip cảm biến để có thể chuyển đổi, phát hiện các hiện tượng trong môi trường tự nhiên và biến nó thành dữ liệu trong môi trường Internet để xử lý dữ liệu và tiến hành thực thi các điều hướng trong mạng Internet đó theo cách mà người dùng mong muốn.

 Hình 2: Cách hoạt động của IoT

Lấy ví dụ hệ thống tưới nước cây tự động như ở trên thì hệ thống sinh thái của chúng ta phải được gắn 1 bộ cảm biến dùng để nhận biết các yếu tố như: nhiệt độ, lượng nước, độ ẩm, thời tiết,… Sau đó được chuyển thành dữ liệu và các dữ liệu này được sử dụng và được thiết lập các thiết lập các chế độ theo mục đích sử dụng.  Và qui trình này sẽ kết nối và hoạt động trong môi trường Internet để thông báo và tạo giao diện đến người dùng.

1.1.5. Ứng dụng của IoT.



Hình 3: Ứng dụng của IoT

Thực ra khái niệm Internet of Things được đưa ra vào năm 1999. Khi mà công nghệ Internet đang từng bước phát triển. Và cho đến ngày nay, Internet Of Things không còn là 1 vấn đề quá viễn vông, cụ thể là rất nhiều các phát mình, ứng dụng được trình làng. Khái niệm thiết bị thông minh, tủ lạnh thông minh, tivi thông minh, … được sử dụng rỗng rãi trong thể giới công nghệ ngày nay. Các bạn có thể điều khiển 1 chiếc tivi bằng điều hướng bàn tay, giọng nói,…bằng công nghệ smart tivi của hãng Samsung, máy lạnh tự động điều chỉnh nhiệt độ theo thời tiết,…Hay như xe ô tô tích hợp chức năng chống sốc tự dộng, tự động báo cho người sử dụng khi lốp xe bị xẹp hay gặp trúng vật cản phía trước khoảng bao nhiêu mét chẳng hạn. Thậm chí là ngôi nhà – nơi chúng ta đang sống cũng có định nghĩa là ngôi nhà thông minh với rất nhiều ứng dụng công nghệ hiện đại. Rất và rất nhiều những ứng dụng trong Internet Of Things đã được các công ty công nghệ khai thác vấn đề này.

1.2. NodeMCU ESP8266 và các cảm biến

1.2.1. Giới thiệu về nodeMCU.

Bảng phát triển NodeMCU ESP8266 đi kèm với mô-đun ESP-12E chứa chip ESP8266 có bộ vi xử lý Tensilica Xtensa 32-bit LX106 RISC. Bộ vi xử lý này hỗ trợ RTOS và hoạt động ở tần số xung nhịp có thể điều chỉnh từ 80MHz đến 160 MHz.

NodeMCU có 128 KB RAM và 4MB bộ nhớ Flash để lưu trữ dữ liệu và chương trình. Sức mạnh xử lý cao của nó với Wi-Fi / Bluetooth và các tính năng Điều hành Ngủ sâu tích hợp khiến nó trở nên lý tưởng cho các dự án IoT

NodeMCU có thể được cấp nguồn bằng giắc cắm Micro USB và chân VIN (Chân nguồn cung cấp bên ngoài). Nó hỗ trợ giao diện UART, SPI và I2C

Bảng phát triển NodeMCU có thể được lập trình dễ dàng với Arduino IDE

Hình 4: Thiết bị NodeMCU

1.2.2. Làm sao để sử dụng chip NodeMCU esp8266?

Bảng phát triển NodeMCU nổi bật với khả năng wifi, chân analog, chân digital và các giao thức giao tiếp nối tiếp.

Để bắt đầu sử dụng NodeMCU cho các ứng dụng IoT trước tiên, chúng ta cần biết về cách viết / tải xuống phần mềm (firmware) NodeMCU trong Bảng phát triển NodeMCU. Và trước đó, phần mềm NodeMCU này sẽ nhận được theo yêu cầu.

Có các bản dựng tùy chỉnh NodeMCU trực tuyến có sẵn bằng cách sử dụng phần mềm NodeMCU tùy chỉnh theo yêu cầu vào mục đích sử dụng.

1.2.3. Làm thế nào để nạp code cho NODEMCU bằng ARDUINO.

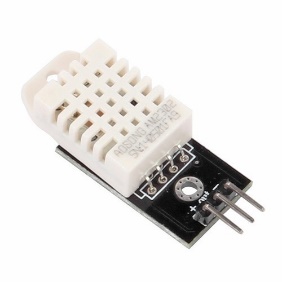
Sau khi thiết lập ESP8266 với chương trình cơ sở Node-MCU, hãy cho phép xem IDE (Integrated Development Environment) cần thiết để phát triển NodeMCU.

Arduino IDE là một cách khác để phát triển NodeMCU với IDE nổi tiếng, tức là Arduino IDE. Chúng tôi cũng có thể phát triển các ứng dụng trên NodeMCU bằng môi trường phát triển Arduino. Điều này giúp các nhà phát triển Arduino dễ dàng hơn việc học ngôn ngữ và IDE mới cho NodeMCU.

1.2.4. Giới thiệu về các cảm biến

Cảm dùng để chuyển đổi tín hiệu của đối tượng môi trường thành tín hiệu dòng 4-10mA 0-10V; chuyển thiết bị điều khiển để cung cấp cho việc sử dụng mong muốn. Có các loại cảm biến như cảm biến nhiệt độ, cảm biến siêu âm, cảm biến mực nước…

1.2.4.1. Cảm biến nhiệt độ và độ ẩm DHT11

Cảm biến DHT11 được sử dụng để đo nhiệt độ và độ ẩm. Nó còn được gọi là AM2302. Cảm biến này rẻ và cũng có độ chính xác tốt hơn. Trong ứng dụng này, chúng tôi sẽ sử dụng cảm biến nhiệt độ và độ ẩm DHT11 với Arduino

Hình 5: Cảm biến nhiệt độ và độ ẩm DHT11

Các thông số kỹ thuật của cảm biến nhiệt độ và độ ẩm DHT11 như sau.

* Phạm vi nhiệt độ từ -40 đến 125 độ C với độ chính xác là ± 0,5 ͦ C.
* Phạm vi độ ẩm là từ 0 đến 100% với độ chính xác là ± 2-5%.
* Tốc độ lấy mẫu là 0,5 Hz.
* Điện áp hoạt động là 3-5V.
* Dòng tối đa trong khi đo là 2,5mA.

Nhiệt độ làm việc và độ ẩm DHT11

* Cảm biến DHT11 bao gồm hai thành phần để đo; Thành phần cảm biến độ ẩm và cảm biến nhiệt độ NTC (hoặc nhiệt điện trở). Có một IC ở phía sau làm cho các bài đọc có thể đọc được bằng Arduino.
* Trong khi đo độ ẩm, cảm biến độ ẩm sẽ hoạt động. Cảm biến độ ẩm bao gồm hai điện cực với chất giữ ẩm giữa chúng. Vì vậy, khi độ ẩm thay đổi, độ dẫn của chất nền thay đổi hoặc bạn có thể nói rằng điện trở giữa các điện cực thay đổi. Sự thay đổi về điện trở này sau đó được trao cho IC khiến cho nó phải đọc bởi Arduino.
* Trong khi đo nhiệt độ, cảm biến nhiệt độ NTC hoặc nhiệt điện trở hoạt động. Điện trở của nhiệt điện trở thay đổi theo sự thay đổi nhiệt độ. Đây là các hệ số nhiệt độ âm (NTC) có nghĩa là điện trở của chúng giảm khi tăng nhiệt độ.

1.2.4.2. Cảm biến lửa (Flame sensor)

Hình 6:Cảm biến lửa

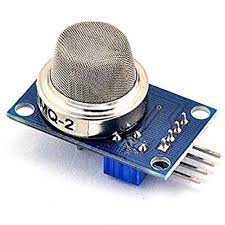
Cảm biến này được sử dụng để phát hiện cháy tầm ngắn và có thể được sử dụng để giám sát các dự án hoặc để phòng ngừa an toàn để cắt / bật thiết bị. Nó đã tìm thấy đơn vị có độ chính xác lên đến khoảng 3 feet.

Cách thức hoạt động của cảm biến lửa?

* Cảm biến ngọn lửa rất nhạy với bước sóng IR ở ánh sáng 760nm ~ 1100nm.
* Ngõ ra tương tự (A0): Tín hiệu điện áp đầu ra thời gian thực trên điện trở nhiệt.
* Đầu ra kỹ thuật số (D0): Khi nhiệt độ đạt đến một ngưỡng nhất định, ngưỡng tín hiệu cao và thấp đầu ra có thể điều chỉnh thông qua chiết áp.

1.2.4.3. The MQ-2 Gas sensor

**Mô-đun cảm biến khí bao gồm một bộ khung bằng thép, trong đó phần tử cảm biến được đặt trong đó. Yếu tố cảm biến này chịu sự tac động của dòng điện thông qua các đầu nối. Dòng điện này được gọi là dòng nhiệt qua nó, các khí đến gần phần tử cảm biến bị ion hóa và được hấp thụ bởi phần tử cảm biến. Điều này thay đổi điện trở của phần tử cảm biến làm thay đổi giá trị của dòng điện đi ra khỏi nó.**

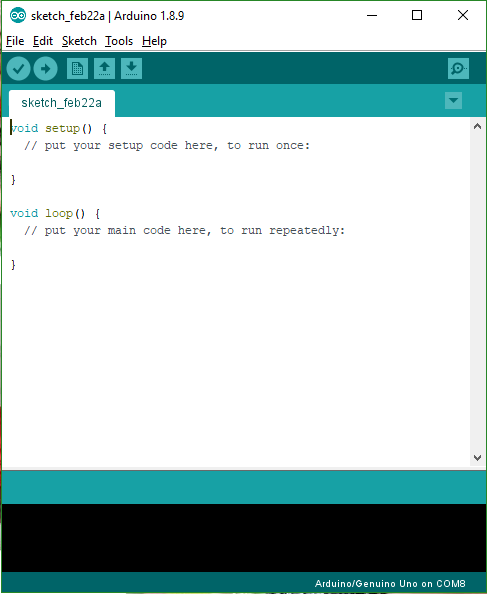
* Phạm vi phát hiện rộng
* Phản ứng nhanh và độ nhạy cao
* Ổn định và lâu dài Mạch ổ đĩa đơn giản
* Được sử dụng trong thiết bị kiểm soát chất lượng không khí cho các tòa nhà / văn phòng, phù hợp để phát hiện
* NH3, NOx, rượu, Benzen, khói, CO2, v.v.
* Kích thước: 35mm x 22mm x 23mm (dài x rộng x cao)
* Điện áp làm việc: DC 5 V
* Hướng dẫn đầu ra tín hiệu.
* Ngõ ra tín hiệu kép (đầu ra analog và đầu ra kỹ thuật số cao / thấp) Điện áp đầu ra analog 0 ~ 4.2V, nồng độ càng cao thì điện áp càng cao.

Hình 7: Cảm biến khí MQ-2

1.3. Giới thiệu về giao diện sử dụng ARDUINO IDE

1.3.1. Arduino IDE là gì?

Arduino IDE là gì? Arduino IDE được viết tắt (Arduino Integrated Development Environment) là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch Arduino.



Hình 8: Giao diện của Arduino IDE

1.3.2. Cổng Com và Serial Monitor.

Cổng nối tiếp (Serial port) là một cổng thông dụng trong các máy tính trong các máy tính truyền thống dùng kết nối các thiết bị ngoại vi với máy tính như: bàn phím, chuột điều khiển, modem, máy quét...Cổng nối tiếp còn có tên gọi khác như: Cổng COM, communication. Ngày nay, do tốc độ truyền dữ liệu chậm hơn so với các cổng mới ra đời nên các cổng nối tiếp đang dần bị loại bỏ trong các chuẩn máy tính hiện nay, chúng được thay thế bằng các cổng có tốc độ nhanh hơn như: USB, FireWire

Serial Monitor là thành phần của Arduino IDE, giúp bo mạch và máy tính có thể gửi và nhận dữ liệu với nhau qua giao tiếp USB.

1.3.3. Cách thức giao tiếp.

Để giao tiếp với máy tính, chúng ta cần phải thiết lập tốc độ kết nối qua cổng usb bằng hàm Serial.begin(). Sau đó dùng hàm Serial.print() để hiện thị nội dung ra màn hình Serial Monitor.

1.3.4. ****Cài đặt board ESP8266 trên Arduino IDE****

Sau khi đã thiết lập phiên bản mới nhất của Arduino IDE, tiếp tục thực thi những bước sau đây để thực thi thiết lập thư viện và công dụng nạp code cho Arduino IDE :

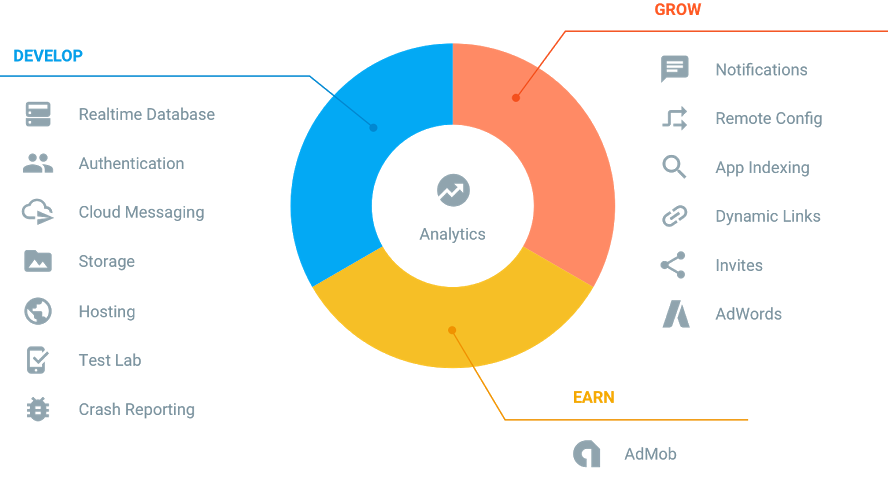
1. Vào **File**  > **Preferences**
2. Nhập **http://arduino.esp8266.com/stable/package\_esp8266com\_index.json** vào ô “**Additional Board Manager URLs**” như được hiển thị trong hình bên dưới. Sau đó, nhấp vào nút “OK”
3. Tiếp theo vào**Tools** > **Board** > **Boards Manager**
4. vào **ESP8266 by ESP8266 Community**, click vào **Install**

**1.4. Giới thiệu về hệ cơ sở dữ liệu FIREBASE**.

1.4.1. Firebase là gì?

Firebase là một nền tảng ứng dụng di động và web với các công cụ và hạ tầng được thiết kế để giúp các lập trình viên xây dựng các ứng dụng chất lượng cao. Thay vì trực tiếp cung cấp các ứng dụng, họ cung cấp các dịch vụ nền tảng cho các lập trình viên, sử dụng để xây dựng ứng dụng cũng như hỗ trợ cho việc tối ưu hóa, tối đa hóa ứng dụng của mình. Với nhiều dịch vụ chất lượng cao đi kèm mức giá phải chăng, Firebase đã và đang, không chỉ là sự lựa chọn hàng đầu cho các lập trình viên đơn thân (single dev) hay các công ty khởi nghiệp (start ups), mà các công ty, tổ chức lớn có tên tuổi cũng sử dụng “Ngọn lửa” để xây dựng các tính năng, các chương trình mới, cũng như chuyển đổi các dịch vụ trước đây sang hệ thống của Firebase. Chẳng hạn như Shazam, Fabulous và cả chính Google nữa, khi nền tảng nhắn tin Allo được xây dựng trên nền tảng Firebase Realtime Database.

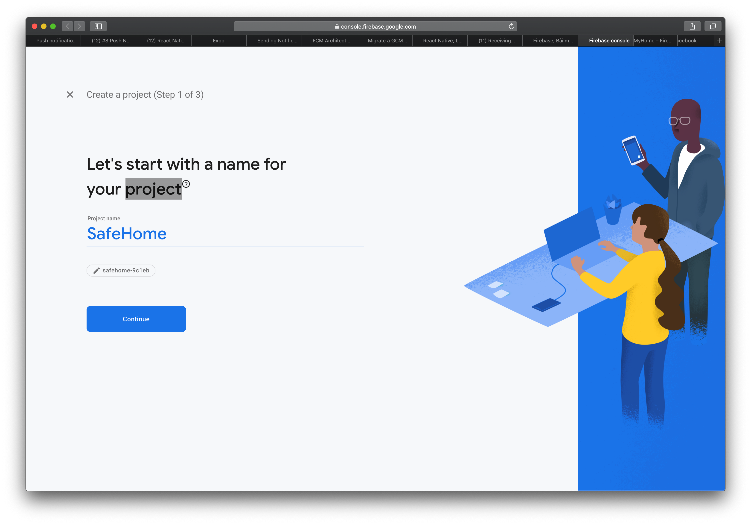
1.4.2. Các tính năng của Firebase

Firebase cung cấp rất nhiều dịch vụ hỗ trợ giúp các lập trình viên xây dựng ứng dụng của mình. Việc tích hợp Firebase vào ứng dụng có thể hạ nỗi lo lắng của các dev xuống ở mức coding, đặc biệt là cho các dev viết các ứng dụng có liên quan tới web

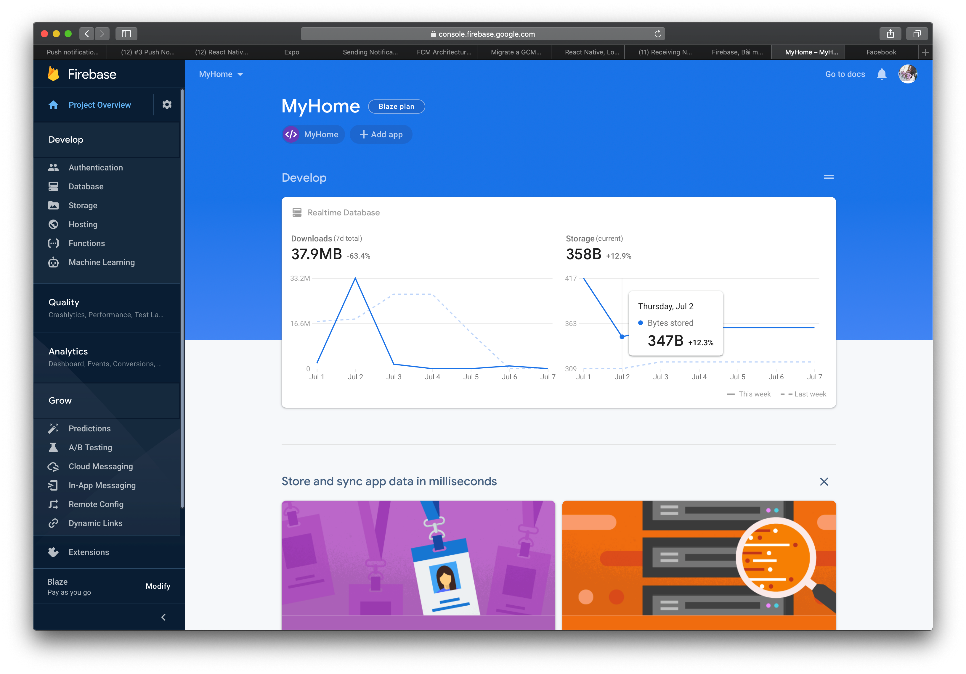
Hình 9: Các tính năng của Firebase

1.4.3. Cách đăng ký và sử dụng

Firebase vừa là một dịch vụ miễn phí và có phí được cung cấp bởi google. Với những dự án nhỏ thì mình hoàn toàn có thể triển khai được với firebase một cách hoàn toàn miễn phí nhưng khi một dự án lớn hơn đòi hỏi đến khả năng xử lý cũng như thuận tiện hơn thì google cung cấp nhiều hơn các tính năng chuyên sâu nhằm hỗ trợ những dự án ấy một cách tốt nhất. Nhưng trong khuôn khổ bài làm của chúng em nên chúng em sẽ hướng dẫn đăng ký và sử dụng các tính năng cơ bản của google firebase.

 Để tạo một Firebase Project mới, chúng ta cần truy cập vào trang chủ firebase console sau đó đăng nhập với tài khoản google của mình hoặc mình cũng có thể đăng ký nếu chưa có và nhấn *Create new project* và tạo project

Hình 10: Giao diện tạo Firebase project

Sau khi tạo xong chúng ta sẽ thấy được giao diện trang chủ tương tự như sau:

Hình 11: Giao diện trang chủ Firebase project

Ở giao diện trang chủ firebase khi nhìn sang bên phải chúng ta có thể thấy được các chức năng chủ yếu như là Authenticaton, Database, Storage… Tuỳ vào từng dự án thì sẽ có các đặc thù công nghệ riêng và ở đây firebase đã thiết lập sẵn nhằm mục đích giúp các lập trình viên khởi tạo một dự án nhanh và thuận tiện nhất có thể mà không mất nhiều thời gian vào lập trình.

CHƯƠNG 2 – TÌNH HUỐNG VÀ GIẢI PHÁP

2.1. Xác định ý tưởng

2.1.1. Tình huống hiện tại

Năm 2022, trên cả nước đã xảy ra hơn 1.700 vụ cháy, nổ

* Tử vong hơn 100 người
* Hàng ngàn người bị thương.
* Số lượng người tử vong trong một dám cháy tăng, cháy karaoke An Phú ở  Bình Dương khiến 32 người chết và 17 người bị thương.

Cháy nổ của nhà và căn hộ đang ở mức cực kỳ nguy hiểm, ảnh hưởng đến tính cách và tài sản của con người. Do đó, cần xây dựng một hệ thống để lường trước nguy cơ hỏa hoạn để cảnh báo người dân.

2.1.2. Giải pháp

Để khắc phục và hạn chế thiệt hai do cháy nổ gây ra nên chúng em đã xây dựng một hệ thống cảnh báo sớm đến người dùng nhầm ngăn chặn kịp thời khi hỏa hoan xảy ra. Hệ thống gồm có:

* Cảm biến nhiệt dộ bên trong nhà hoặc chung cư.
* Cảm biến rò rỉ khí ga trong nhà bếp.
* Cảm biến độ ẩm trong không khí để phát hiện hiện kịp thời các đám cháy nhỏ.
* Cảm biến lửa giúp cảm báo sớm nhất có thể cho người dung.

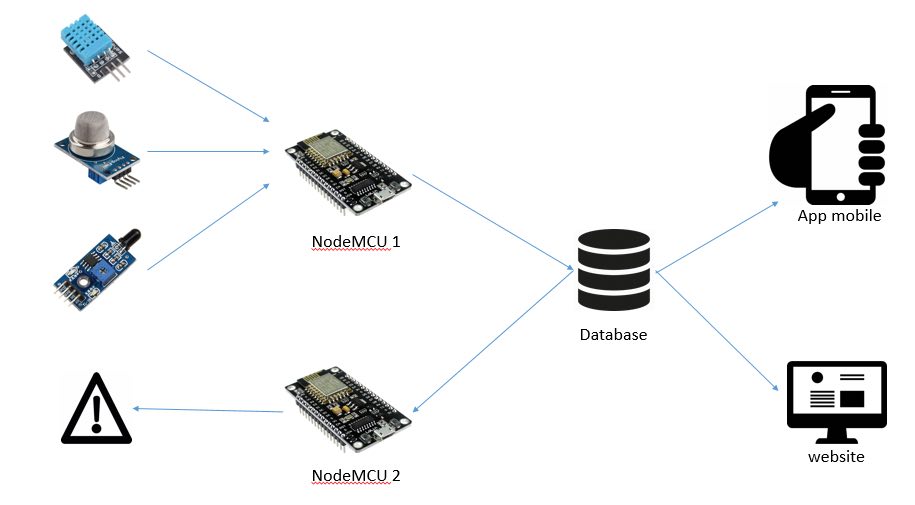
2.1.3. Cách thức thực hiện

Liên kết NodeMCU với 3 cảm biến: cảm biến lửa, cảm biến nhiệt độ và độ ẩm DHT11, cảm biến khí MQ -2 như đã giới thiệu tại mục 1.2. Các cảm biến này có nhiệm vụ liên tục ghi lại các thông tin như nhiệt độ trong nhà, mức độ khí gas và phát hiện thấy lửa trong nhà hay không sau đó gửi thông tin đến NodeMCU ( hình 4, mục 1.2.1), khi đó NodeMCU sẽ phân tích các dữ liệu thu nhận được và gửi các thông tin đó lên cơ sở dữ liệu thông qua wifi đồng thời bật đèn tín hiệu từ màu xanh (an toàn) cho đến mức cao nhất là màu đỏ (nguy hiểm). Dữ liệu sẽ được cập nhật lên ứng dụng điện thoại để người dùng dễ dàng theo dõi.

2.2 Các lắp ráp và cài đặt hệ thống báo cháy

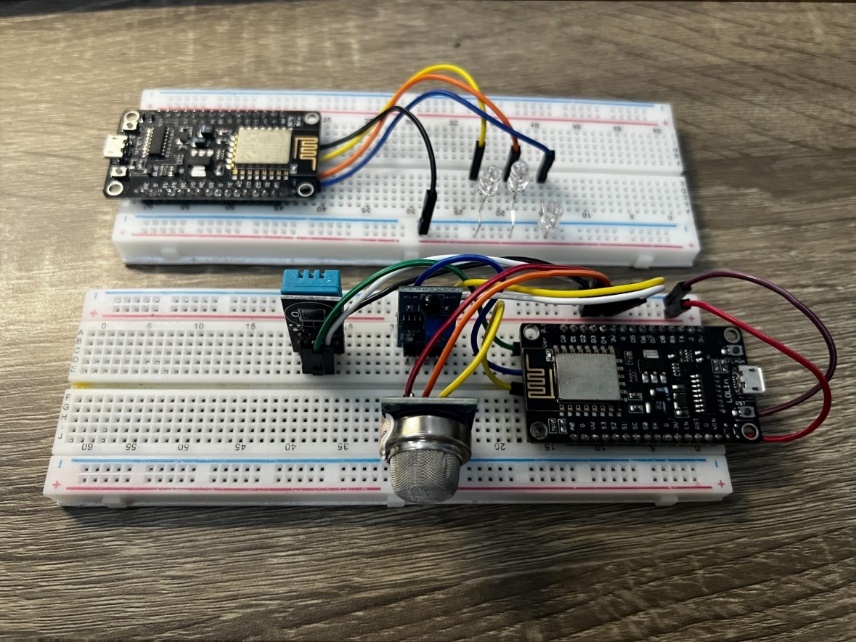
2.2.1. Sơ đồ tổng quát

Chúng em sử dụng 2 nodeMCU: 1 nodeMCU dùng để lấy thông tin dữ liệu và cập nhật dữ liệu thường xuyên lên database và một nodeMCU dùng để nhận giữ liệu về để đưa ra các cảnh báo. Đồng thời tạo ra hai nền tảng là ứng dụng di động và website để hiển thị thông tin và cũng như thông báo những cảnh báo đến người dùng (Hình 12).



Hình 12: Sơ đồ tổng quát

* Tại NodeMCU 1 sẽ là nơi nhận giá trị từ các cảm biến nhiệt độ, độ ẩm, cảm biến khí gas và cảm biến phát hiện lửa (mục 1.2.4) sau đó gửi dữ liệu lên database.
* Database được sử dụng thông qua google firebase (mục 1.4) dùng để lưu trữ dữ liệu được gửi lên từ NodeMCU 1 và chuyển dữ liệu đến các nền tảng khác.
* NodeMCU 2 sẽ được lắp đặt ở một vị trí thuận tiện trong căn nhà nhằm có mục đích cảnh báo đến cho người cư trú thông qua các tín hiệu đèn LED, chuông báo và hệ thống dập lửa.
* App mobile và Website là 2 nền tảng chính để tương tác trực tiếp với người ở căn và là luôn được cập nhật thông tin theo thời gian thực để người sở hữu có thể tiện theo dõi ngay cả khi không có ở nhà.

Để đáp ứng được các yêu cầu đặt ra nên trong đồ án này chúng em đã sử dụng các thư viện Firebase ESP8266, ESP8266 WiFi, DHT22 và MQ-2. Chúng được sử dụng cho mục đích như sau:

Hình 13: Sản phẩm thực tế

* Firebase ESP8266: Thư viện này cung cấp các hàm dùng để gửi và nhận dữ liệu với cơ sở dữ liệu firebase
* ESP8266Wifi: Nhằm mục đích kết nối wifi cho NodeMCU.
* DHT: Dùng để đọc giữ liệu từ cảm biến DHT11(mục 1.2.4.1) để lấy giá trị nhiệt độ và độ ẩm.
* MQ-2: Đọc giữ liệu từ cảm biến MQ-2(mục 1.2.4.3) khi phát hiện được ra khí gas.

2.2.2. Xây dựng cơ sở dữ liệu.

Sau khi các NodeMCU ghi nhận được các thông tin cần thiết từ các cảm biến, chúng em tiếp tục chuyển qua xây dựng một cơ sở dữ liệu để NodeMCU gửi dữ liệu lên nhằm cho mục đích lưu trữ. Bên cạnh đó thì tạo cơ sở dữ liệu sẽ giúp chúng em triển khai được việc bố trí thiết bị thuận tiện hơn. Ví dụ như ta có thể lắp đặt cảm biến khí gas ở nhà bếp cùng với nodeMCU gửi dữ liệu mà vẫn có thể thông báo được đến chuông báo cháy tại phòng khách thông qua một NodeMCU khác lấy dữ liệu về từ database. Ngoài ra xây dựng cơ sở dữ liệu còn giúp xây dựng ứng dụng điện thoại mà chúng em sẽ giới thiệu ở phần tiếp theo.

Đầu tiên chúng em khởi tạo một Firebase Project (mục 1.4.3) sau đó thiết lập ra 1 đối tượng chính . Mỗi đối tượng sẽ có những thuộc tính như sau:

* [https://fire-warning-9e33c-default-rtdb.firebaseio.com/](https://console.firebase.google.com/u/0/project/fire-warning-9e33c/database/fire-warning-9e33c-default-rtdb/data/~2F)
* Flame\_status: Thể hiện giá trị phát hiện lửa.
* Gas: Lưu trữ chỉ số gas trong không khí.
* Warning: Mức độ nguy hiểm của căn nhà.
* Humidity: Độ ẩm trong căn nhà.
* Temperature: Nhiệt độ trong căn nhà.



Hình 14: Cơ sở dữ liệu firebase

2.2.3. Cài đặt thư viện cho sensor

Tại màn hình chính Arduino IDE, chúng ta có thể cài thư viện cho các cảm biến một cách dễ dàng. Để cài đặt thư viện, chúng ta cần thực hiện theo các bước sau.

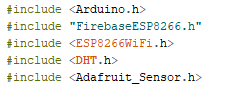
1. Vào Sktech > Chọn Include library > Manage Libraris…
2. Tìm thư viện muốn cài đặt > Chọn Install
3. Sau khi thư viện được cài đặt xong > Chọn Close

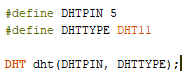
Các thư viện được dùng trong đồ án này gồm:

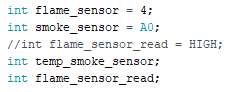
* DHT sensor Library
* Firebase ESP8266 Client
* Adafruit Unified Sensor

2.2.4. NodeMCU gửi dữ liệu

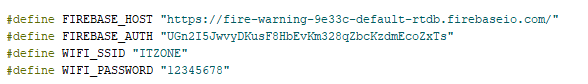
Để cài đặt cho NodeMCU gửi dữ liệu ( NodeMCU 1 tại hình 12) sẽ cần thực hiện 3 bước chính là: Thêm vào các thư viện cần thiết như đã đề cập tại sơ đồ tổng quát (mục 2.2.1), cấu hình cài đặt ban đầu và cuối cuối cùng là cài đặt xuyên suốt thời gian hoạt động.

Hình 15. Include các thư viện đã cài đặt

Định nghĩa các cổng nối với các thiết bị sensor để định nghĩ đúng kiểu dữ liệu và lấy dữ liệu từ sensor một cách chính xác nhất.

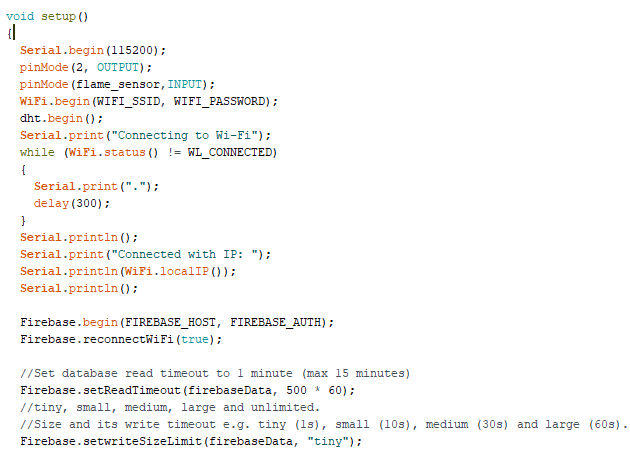
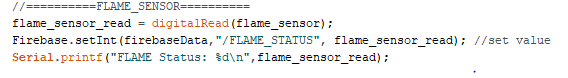
Hình 16. Định nghĩ cổng D1 gắn với DHT11

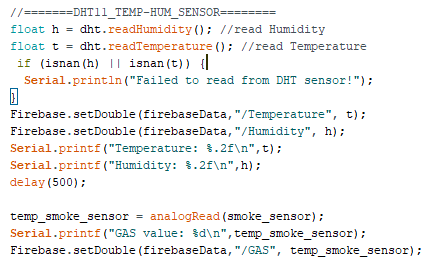
Hình 17. Định nghĩa cổng A0 gắn cảm biến khí gas và D2 gắn cảm biến lửa

Để có thể kết nối và đẩy dữ liệu từ chip ESP8266 lên Firebase chúng ta cần kết nối internet không dây (Wifi) bao gồm tên Wifi và password. Tiếp theo là định nghĩa đầy đủ các thông tin về Firebase thì NodeMCU mới có thể kết nối được với Firebase chúng ta đã tạo.

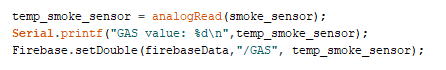
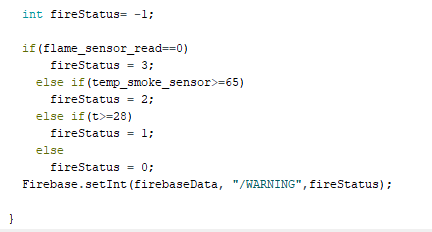
Hình 18. Khai báo mạng wifi và Firebase

Sử dụng hàm setup() này sẽ được chạy duy nhất vào lần đầu tiên khi NodeMCU được cấp nguồn và vì thế nên ở đây mình sẽ sử dụng để cài đặt các cầu hình cơ bản trước khi để NodeMCU này lấy và gửi dữ liệu đi. Đầu tiên chúng em cài đặt cổng vào cho cảm biến phát hiện lửa. Tại vì cảm biến nhiệt độ, độ ẩm và cảm biến gas đã có thư viện hỗ trợ và đã được thêm vào trước đó nên sẽ không phải cài đặt thêm mà chỉ cần sự dụng qua các hàm được thư viện cung cấp. Sau đó là cài đặt để kết nối wifi cho con chip. Cuối cùng sẽ là cài đặt kết nối đến cơ sở dữ liệu

Hình 19: cài đặt cho NodeMCU gửi dữ liệu

Hình 20: Quá trình xử lý của NodeMCU gửi dữ liệu Flame sensor

Hình 21: Quá trình xử lý của NodeMCU gửi dữ liệu DHT11 sensor

Hình 22: Quá trình xử lý của NodeMCU gửi dữ liệu MQ-2 sensor

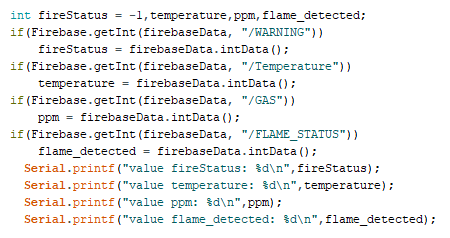
Hình 23: Quá trình xử lý của NodeMCU tính mức độ cảnh báo

Tại hàm loop() sẽ là hàm để chạy xuyên suốt thời gian còn lại sau khi đã được cấu hình xong (hình 16) và sẽ thực hiện trình tự các bước từ đọc dữ liệu đến xử lý và gửi thông tin đi:

* Đọc và gửi giá trị lên từ cảm biến phát hiện lửa (hình 20)
* Đọc và gửi giá trị nhiệt độ, độ ẩm từ cảm biến DHT11 (hình 21).
* Đọc và gửi giá trị lên từ cảm biến khí gas (hình 22).
* Phân tích tình trạng căn nhà dựa trên những giá trị lấy được từ các cảm biến và sau đó tính toán đến mức độ an toàn trong căn nhà và gửi kết quả đánh giá lên cơ sở dữ liệu (hình 23).

2.2.5. NodeMCU nhận dữ liệu

Các cấu hình của NodeMCU nhận dữ liệu cũng tương tự như NodeMCU gửi dữ liệu nhưng sẽ được tùy biến để cài đặt các đầu ra output để có thể xuất được trình trạng mức độ an toàn trong căn nhà.



Hình 24: Cấu hình và cài đặt NodeMCU nhận dữ liệu

Trong NodeMCU này sẽ cài đặt những kết nối cần thiết cho mỗi lần khởi động:

-       Cài đặt các đầu vào đầu ra của NodeMCU và khởi tạo hàm tìm kiếm wifi cũng như bắt đầu tạo đường kết nối đến cơ sở dữ liệu thông qua 2 hàm đó là wifi.begin() và Firebase.begin() giống như chip gửi dữ liệu lên database

-       Định nghĩa biến temperature lấy giá trị /Temperature từ firebase

- Định nghĩa biến ppm lấy giá trị /GAS từ firebase

- Định nghĩa biến flame lấy giá trị /FLAME\_STATUS từ firebase

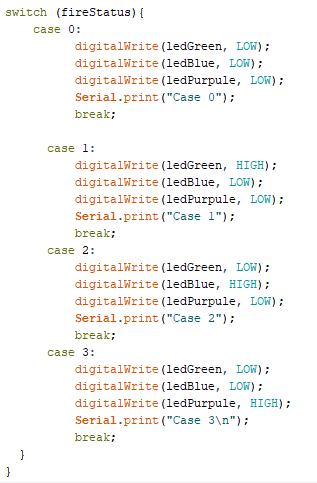
- Định nghĩa biến fireStaus lấy giá trị /Warning từ firebase

- Show ra các giá trị đã lấy về qua màn hình Serial Monitor của Arduino IDE

Ở NodeMCU này (hình 25) có nhiệm vụ liên tục cập nhật tình trạng căn nhà thông qua việc lấy thông tin qua database đã kết nối trước đó bằng hàm Firebase.getInt() rồi sau đó dựa vào kết quả thu được thì sẽ hiển thị các thông báo khác nhau qua việc bật tắt đèn LED.

Các trạng thái đèn led sẽ thay đổi như sau:

* Khi Warning = 0: Sẽ không đèn nào được bật sáng
* Khi Warning =1: Đèn led nối với cổng D4 sẽ bật sáng
* Khi Warning =2: Đèn led nối với cổng D2 sẽ bật sáng
* Khi Warning =3: Đèn led nối với cổng D1 sẽ bật sáng

 Hình 25: Quá trình xử lý của NodeMCU sau khi nhận dữ liệu

2.3. Xây dựng ứng dụng WebApp hiển thị thông tin.

2.3.1. Đặt vấn đề

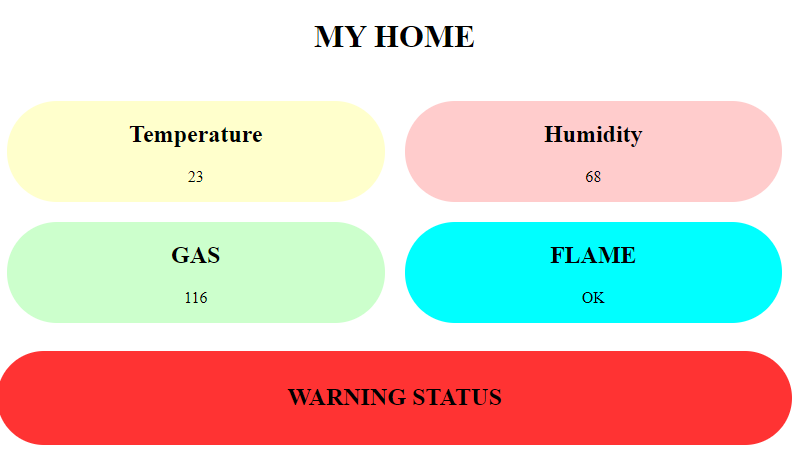
Trong thời đại công nghệ hiện nay thì các thiết bị thông minh đã rất phổ biến và gần như mỗi người đều trang bị cho mình 1 chiếc điện thoại riêng hoặc 1 chiếc laptop cá nhân. Chính vì thế chúng em đã quyết định tạo nên một ứng dụng WebApp dùng để cập nhật tất cả các thông tin, tình trạng căn nhà và báo động cho chủ căn nhà mọi trường hợp khẩn cấp. Ứng dụng này như là một cách nhanh và hiệu quả nhất để tương tác trực tiếp với người sử dụng.

Để xây dựng nên một ứng dụng diện thoại đáp ứng nhu cầu trên thì chúng em đã chọn ngôn ngữ HTML, CSS, Javascripst để xây dựng ứng dụng này bởi vì sự tối ưu, dễ dàng xây dựng cũng như khả năng có thể tùy biến giao diện khi mở trên các kích thước màn hình thiết bị khác nhau

2.3.2. Xây dựng giao diện đăng nhập và đăng ký.

Chúng em xây dựng giao diện cơ bản gồm có tên tài khoản và mật khẩu và giao diện đăng ký cho những người dùng mới. Giao diện đầu tiên khi vào ứng dụng sẽ là màn hình đăng nhập cùng với đó là chức năng đăng ký tài khoản mới hoặc người dùng có thể dễ dàng đăng nhập hơn nếu đã có sẵn tài khoản google thì sẽ hoàn toàn có thể đăng nhập được vào ứng dụng với mọi thông tin của tài khoản thông qua API được cung cấp của nhà phát hành Google

2.3.3. Xây dựng giao diện hiển thị thông số trong căn nhà

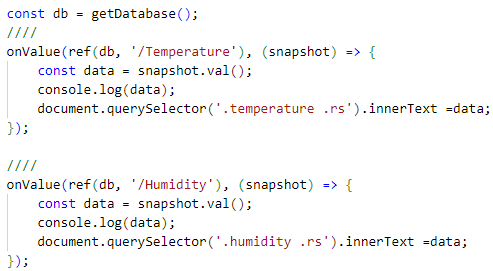
Màn hình *Dashboard* sẽ là màn hình chính dùng để tương tác trực tiếp với người dùng được hiển thị sau khi người dùng đăng nhập thành công. Khi người dùng đăng nhập thành công thì những lần mở web tiếp theo sẽ luôn hiển thị màn hình này cho đến khi đăng xuất để đăng nhập vào một tài khoản khác. Màn hình thể hiện đầy đủ thông tin từ các cảm biến từ đó giúp người dùng dễ dàng nắm bắt, theo dõi thông tin.

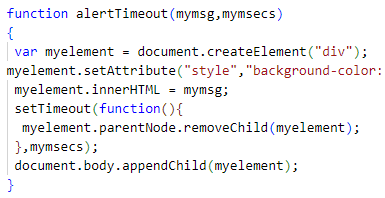
Hình 26. Giao diện hiển thị thông tin web

Đây là màn hình giao diện chính của ứng dụng nhằm thể hiện tất cả các thông tin cần thiết về căn nhà cũng như hiển thị mức độ an toàn trong căn nhà.

2.3.4. Mô tả code

Để có thể hiển thị thông tin liên tục từ database, chúng ta cần kết nối đến firebase. Tải và cài đặt thư viện Firebase JavaScript SDK từ CDN.

Hình 26. Tải và cài đặt thư viện Firebase JavaScript

Hình 27. Lấy dữ liệu từ firebase realtime

Hình 28. Hiển thị notification cảnh báo khi phát hiện lửa

Các thông tin được hiển thị lên như chỉ số nhiệt độ, độ ẩm, hệ thống phát hiện lửa và chỉ số rò gas. Kèm với đó là hiển thị mức độ nguy hiểm trong căn nhà với các mức được lấy từ database firebase được xử lý sẵn với NodeMCU trước đó.

CHƯƠNG 3 – KẾT LUẬN

3.1. Những kết quả đã đạt được

Sau khi hoàn thành dự án này chúng em đã đạt được:

* Nắm được kiến thức chủ yếu về IoT và tương lai của lập trình IoT trong thời đại ngày ngay.
* Xây dựng ra được sản phẩm thiết thực từ chip NodeMCU, cảm biến với các mạnh điện.
* Biết được thêm nhiều công nghệ mới như Firebase và tạo ứng dụng di động với React Native.

3.2. Những vấn đề chưa làm được và hướng phát triển trong tương lai

Từ khi thực hiện cho đến khi hoàn thành dự án lần này thì chúng em vẫn còn nhiều vấn đề chưa giải quyết được như sau:

* Cách lắp ráp các thiết bị chưa được gọn gàng và thuận tiện nhất.
* Hệ cơ sở dữ liệu còn có lỗ hổng và chưa triển khai được trên nhiều người cùng sử dụng.
* Ứng dụng điện thoại chưa có khả năng tạo thông báo khẩn cấp đến người dùng.

Chúng em đã quyết định trong tương lai sẽ tiếp tục hoàn thiện dự án này bằng cách bổ sung tiếp những điều hiện tại chưa làm được và thêm nhiều chức năng khác. Chúng em thực hiện dự án này với mong muốn sẽ giải quyết được các trường hợp mất mát không đáng có vì tình trạng cháy nổ trong hộ gia đình hiện nay.

TÀI LIỆU THAM KHẢO

[http://arduino.cc](https://l.facebook.com/l.php?u=http%3A%2F%2Farduino.cc%2F%3Ffbclid%3DIwAR0rkcSr4Lh6ZxeRs7cK-KhAYJpjoAz6wwrS2inw3o9US1eZUZHMut7fpoM&h=AT38m8i9Ab-CRmq0NepVNb1dx1BaBFmc4i5IsHWmBPgQhgRN7HnR_q1UjENnK_skBV_EWpmxYjOXeU-lyynaOFoOV-XH8QJNvfkIp0y95RB6jmmtersvJSFZj358ZuQ-3Qhz)

[https://github.com](https://l.facebook.com/l.php?u=https%3A%2F%2Fgithub.com%2F%3Ffbclid%3DIwAR2it_TQ7irWY4zH37Y74Hq0opxQFheIDPXouQie0bA154oDWc23dDjRqwg&h=AT38m8i9Ab-CRmq0NepVNb1dx1BaBFmc4i5IsHWmBPgQhgRN7HnR_q1UjENnK_skBV_EWpmxYjOXeU-lyynaOFoOV-XH8QJNvfkIp0y95RB6jmmtersvJSFZj358ZuQ-3Qhz)

[https://firebaseopensource.com/](https://l.facebook.com/l.php?u=https%3A%2F%2Ffirebaseopensource.com%2F%3Ffbclid%3DIwAR1NVatIuSpPJNTVPDwKmmUR-oah9ohX1naruwtD1tnHPIezLdvwGeVzUo4&h=AT38m8i9Ab-CRmq0NepVNb1dx1BaBFmc4i5IsHWmBPgQhgRN7HnR_q1UjENnK_skBV_EWpmxYjOXeU-lyynaOFoOV-XH8QJNvfkIp0y95RB6jmmtersvJSFZj358ZuQ-3Qhz)

<https://eitguide.net/firebase-bai-2-1-gioi-thieu-ve-realtime-database/>

<https://viblo.asia/p/lam-viec-voi-firebase-realtime-database-ZjlvalVXkqJ>

<http://arduino.vn/bai-viet/1496-esp8266-ket-noi-internet-phan-1-cai-dat-esp8266-lam-mot-socket-client-ket-noi-toi>

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css/>

<https://www.w3schools.com/javascripts/>