TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐIỆN - ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

KHÓA LUẬN TỐT NGHIỆP

**THIẾT KẾ VÀ THI CÔNG MÔ HÌNH**

**CẢNH BÁO TRỘM GỖ, CHÁY RỪNG ỨNG DỤNG IOT**

**NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH**

Sinh viên: **TRẦN PHƯƠNG THIỆU ĐÔNG**

MSSV: 16119074

**PHẠM THANH HẢI**

MSSV: 16119084

TP. HỒ CHÍ MINH – 07/2020

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

KHÓA LUẬN TỐT NGHIỆP

**THIẾT KẾ VÀ THI CÔNG MÔ HÌNH**

**CẢNH BÁO TRỘM GỖ, CHÁY RỪNG ỨNG DỤNG IOT**

**NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH**

Sinh viên: **TRẦN PHƯƠNG THIỆU ĐÔNG**

MSSV: 16119074

**PHẠM THANH HẢI**

MSSV: 16119084

Hướng dẫn**: ThS. NGUYỄN VĂN PHÚC**

TP. HỒ CHÍ MINH – 07/2020

LỜI CẢM ƠN

Nhóm thực hiện xin được gửi lời cảm ơn chân thành đến giảng viên hướng dẫn thầy Nguyễn Văn Phúc đã hỗ trợ, hướng dẫn, góp ý nhóm trong quá trình thực hiện đồ án, người đã đưa ra hướng giải quyết khó khăn, giải đáp thắc mắc để đề tài đi đúng hướng ứng dụng với thực tế nhất có thể. Trong quá trình thực hiện nhóm đã tiếp thu được những kiến thức thực tế và cách làm việc nghiêm túc, hiệu quả từ thầy.

Nhóm em xin gửi lời tri ân thành nhất đến các quý thầy cô trong khoa Điện - Điện tử đã hỗ trợ chúng em về những kiến thức nền tảng vững vàng, tạo điều kiện tốt nhất cho sinh viên trong quá trình học tập và nghiên cứu.

Sự hỗ trợ thầm lặng và vô cùng quan trọng từ gia đình và bạn bè luôn là động lực để nhóm có thể làm việc hết khả năng và hoàn thành đồ án một cách tốt nhất.

Một lần nữa nhóm vô cùng hân hạnh khi được làm sinh viên tại trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, là học trò của những giảng viên đầy tâm huyết, lời cảm ơn này cũng là sự ghi nhận sâu sắc mà nhóm muốn gửi đến thầy cô, gia đình và bạn bè.

**Trân trọng**

TP.HCM, ngày 30 tháng 07 năm 2020

**Nhóm thực hiện đề tài**

Phạm Thanh Hải Trần Phương Thiệu Đông

TÓM TẮT

Các vấn nạn cháy rừng, trộm gỗ xảy ra mỗi năm gây thiệt hại lớn về nguồn tài nguyên thiên nhiên ngoài ra nó còn gây ra nhiều thiên tai khác như: sạt lở, lũ quét, lũ ống… Vì tính chất rộng lớn và địa hình hiểm trở, nên những vụ cháy rừng, trộm gỗ xảy ra do không phát hiện kịp thời gây nên thiệt hại to lớn về tài nguyên và kinh tế. Nhận thấy được những nguy cơ nguy hiểm đó nên chúng tôi đã chọn và thực hiện đề tài *“Cảnh báo trộm gỗ, cháy rừng”*, nhằm cảnh báo kịp thời các nguy cơ về cháy rừng, trộm gỗ một cách nhanh chóng nhất có thể.

Đề tài trình bày và mô tả về mô hình của một hệ thống cảnh báo trộm gỗ, cháy ở rừng ứng dụng công nghệ IoT kết hợp LoRa để truyền nhận dữ liệu. Và dữ liệu được thu thập từ các cảm biến tại khu rừng theo dõi, khắc phụ nhược điểm về khả năng kết nối và mở rộng của các hệ thống đã có.

Hệ thống xử lý trung tâm sẽ do NodeMcu ESP8266 đảm nhận, với khả năng kết nối đa dạng với nhiều thiết bị cùng lúc. Nhận dữ liệu từ nơi theo dõi qua sóng RF Lora, đồng thời đẩy dữ liệu lên server lưu trữ thông qua wifi. Tại server sẽ cung cấp dữ liệu để bên phía người dùng có thể theo dõi, giám sát bằng giao diện trên điện thoại thông minh hoặc thông qua website để cảnh báo trộm gỗ, cháy nhanh nhất.

Kết nối giữa các khối trong hệ thống sử dụng công nghệ truyền dẫn dữ liệu công suất thấp LoRa, thông qua module RF LoRa Ra-02 SX1278 để tận dụng ưu thế về khả năng truyền xa và tiết kiệm năng lượng của module này. Khối thu thập dữ liệu đặt tại rừng sẽ sử dụng pin như một giải pháp năng lượng cho thiết bị IoT.

**MỤC LỤC**

[DANH MỤC HÌNH xii](#_Toc47315380)

[DANH MỤC BẢNG xiv](#_Toc47315381)

[CÁC TỪ VIẾT TẮT xv](#_Toc47315382)

[Chương 1. Giới thiệu 1](#_Toc47315383)

[1.1 ĐẶT VẤN ĐỀ 1](#_Toc47315384)

[1.2 TÌNH HÌNH NGHIÊN CỨU 3](#_Toc47315385)

[1.2.1 Waspmote Plus & Sense 3](#_Toc47315386)

[1.2.2 Sử dụng vệ tinh MODIS 7](#_Toc47315387)

[1.3 mục tiêu đề tài 8](#_Toc47315388)

[1.4 nội dung thực hiện 8](#_Toc47315389)

[1.5 BỐ CỤC ĐỀ TÀI 8](#_Toc47315390)

[Chương 2. CƠ SỞ LÝ THUYẾT 9](#_Toc47315391)

[2.1 TẦM quan trọng của việc Cảnh báo cháy, trộm gỗ tại rừng 9](#_Toc47315392)

[2.2 tổng quan về iot 9](#_Toc47315393)

[2.3 TỔNG QUAN VỀ GATEWAY 9](#_Toc47315394)

[2.4 công nghệ truyền thông KHÔNG DÂY LORA 9](#_Toc47315395)

[2.4.1 Tổng quan 9](#_Toc47315396)

[2.4.2 Sơ lược về LoraWan 10](#_Toc47315397)

[2.4.3 Các thành phần trong mạng Lora 11](#_Toc47315398)

[2.5 tổng quan về ARDUINO 12](#_Toc47315399)

[2.5.1 Phần cứng 12](#_Toc47315400)

[2.5.2 Phần mềm 12](#_Toc47315401)

[2.6 phần mềm 12](#_Toc47315402)

[2.6.1 Giao Thức Websocket 12](#_Toc47315403)

[2.6.2 NodeJs 12](#_Toc47315404)

[2.6.3 React native 12](#_Toc47315405)

[2.6.4 Cơ Sở Dữ Liệu MongoDB 12](#_Toc47315406)

[2.7 LỰA CHỌN VÀ THIẾT LẬP MÔ HÌNH NĂNG LƯỢNG 13](#_Toc47315407)

[2.7.1 Năng lượng trong quá trình truyền dữ liệu 13](#_Toc47315408)

[2.7.2 Năng lượng thu thập dữ liệu 14](#_Toc47315409)

[2.7.3 Năng lượng xử lý dữ liệu 14](#_Toc47315410)

[Chương 3. THIẾT KẾ VÀ THI CÔNG 16](#_Toc47315411)

[3.1 YÊU CẦU ĐỐI VỚI HỆ THỐNG 16](#_Toc47315412)

[3.2 ĐẶC TẢ HỆ THỐNG 16](#_Toc47315413)

[3.2.1 Chức năng của hệ thống 16](#_Toc47315414)

[3.2.2 Đặc tính của hệ thống 16](#_Toc47315415)

[3.2.3 Mô hình tổng thể hệ thống 16](#_Toc47315416)

[3.3 KIẾN TRÚC CỦA HỆ THỐNG 17](#_Toc47315417)

[3.3.1 Sơ đồ khối của hệ thống 17](#_Toc47315418)

[3.3.2 Phân tích sơ đồ khối và lựa chọn linh kiện thiết kế 18](#_Toc47315419)

[3.3.3 Sơ đồ nối dây của các khối 27](#_Toc47315420)

[3.3.4 Tính toán điện áp, dòng điện và thiết kế khối nguồn 34](#_Toc47315421)

[3.3.5 Kiến trúc phần mềm thiết bị 36](#_Toc47315422)

[3.4 thi công hệ thống 39](#_Toc47315423)

[3.4.1 Thi công bo mạch 39](#_Toc47315424)

[3.4.2 Thi công, đóng gói mô hình 44](#_Toc47315425)

[3.5 Quản lý năng lượng cho hệ thống 45](#_Toc47315426)

[3.5.1 Tính toán thời gian sống của pin 47](#_Toc47315427)

[3.5.2 Giải pháp khắc phục năng lượng cho hệ thống 48](#_Toc47315428)

[3.6 Thiết kế phần mềm 49](#_Toc47315429)

[3.6.1 Thiết kế ứng dụng di động 49](#_Toc47315430)

[3.6.2 Thiết kế giao diện website 52](#_Toc47315431)

[3.6.3 Máy chủ và hệ cơ sở dữ liệu 54](#_Toc47315432)

[Chương 4. kết quả thực hiện 60](#_Toc47315433)

[4.1 KẾT QUẢ 60](#_Toc47315434)

[4.2 KẾT QUẢ THỰC NGHIỆM 61](#_Toc47315435)

[4.3 KẾT QUẢ PHẦN CỨNG 66](#_Toc47315436)

[4.4 KẾT QUẢ Phầm mềm 66](#_Toc47315437)

[4.4. NHẬN XÉT – ĐÁNH GIÁ 68](#_Toc47315438)

[4.4.1. Kiểm tra sai số 68](#_Toc47315439)

[4.4.2 Khả năng truyền tín hiệu của LoRa 71](#_Toc47315440)

[4.4.3 Khả năng nhận dạng gương mặt và quét vân tay 73](#_Toc47315441)

[4.4.4. Khả năng xử lý của website 73](#_Toc47315442)

[Chương 5. KẾT LUẬN và hướng phát triển 75](#_Toc47315443)

[5.1 kết luận 75](#_Toc47315444)

[5.2 Hướng phát triển 76](#_Toc47315445)

[**TÀI LIỆU THAM KHẢO** 78](#_Toc47315446)

[PHỤ LỤC 82](#_Toc47315447)

[Chip ESP8266 82](#_Toc47315448)

[Cảm Biến Nhiệt Độ 82](#_Toc47315449)

[Bộ chuyển đổi Adapter 82](#_Toc47315450)

[Pin li-ion 82](#_Toc47315451)

[Giới thiệu phần mềm lập trình Arduino IDE 82](#_Toc47315452)

[CÁC CHUẨN GIAO TIẾP 82](#_Toc47315453)

[Chuẩn giao tiếp one-wire 82](#_Toc47315454)

[Giao tiếp SPI 82](#_Toc47315455)

[Giao Tiếp I2C 82](#_Toc47315456)

[Mạng Không Dây Wi-Fi 82](#_Toc47315457)

[Module cảm biến khí CO MQ-7 82](#_Toc47315458)

DANH MỤC HÌNH

[Hình 2. 1. *Cấu trúc mạng LoRa dạng hình sao* 7](#_Toc535665302)

[Hình 2. 2*. Tập giao thức mạng LoRa* 7](#_Toc535665303)

[Hình 2. 3 *Các mô hình mạng zigbee* 9](#_Toc535665304)

[Hình 2. 4 *Tập giao thức của mạng Zigbee* 9](#_Toc535665305)

[Hình 2. 5 *Mô hình WiFi* 10](#_Toc535665306)

[Hình 2. 6 *Cấu tạo thẻ RFID* 12](#_Toc535665307)

[Hình 2. 7 *Nguyên lý hoạt động của RFID* 13](#_Toc535665308)

[Hình 2. 8 *Kiến trúc hệ điều hành android* 14](#_Toc535665309)

[Hình 3. 17 *Sơ đồ kết nối phần cứng của gateway tại nhà* 39](#_Toc46955426)

[Hình 3. 18 *Lưu đồ thuật toán khối Node Sensor* 40](#_Toc46955427)

[Hình 3. 19 *Lưu đồ thuật toán khối quản lý nhân công* 41](#_Toc46955428)

[Hình 3. 20 *Lưu đồ thuật toán khối thu thập* 42](#_Toc46955429)

[Hình 3. 21 *Lưu đồ chương trình khối nhận lệnh và điều khiển* 43](#_Toc46955430)

[Hình 3. 22 *Lưu đồ thuật toán khối Gateway* 44](#_Toc46955431)

[Hình 3. 26 *Lưu đồ tương tác giữa giao diện người dùng và cơ sở dữ liệu* 53](#_Toc46955432)

[Hình 4. 1 *Mô hình hệ thống nông trại sử dụng mạng IoT công suất thấp* 38](#_Toc535670717)

[Hình 4. 2 *Mô hình tại nông trại* 39](#_Toc535670718)

[Hình 4. 3 *Mô hình gateway trạm chính* 40](#_Toc535670719)

[Hình 4. 4 *Mạch gateway trạm chính* 40](#_Toc535670720)

[Hình 4. 5 *Mô hình node sensor* 41](#_Toc535670721)

[Hình 4. 6 *Mô hình tại nhà* 41](#_Toc535670722)

[Hình 4. 7 *Giao diện hiển thị các thông số môi trường tại nông trại* 42](#_Toc535670723)

[Hình 4. 8 *Giao diện thanh lựa chọn menu* 43](#_Toc535670724)

[Hình 4. 9 *Giao diện trang điều khiển tưới nước* 44](#_Toc535670725)

[Hình 4. 10 *Giao diện quản lí nhân công ban đầu* 45](#_Toc535670726)

[Hình 4. 11 *Giao diện cập nhật mức lương* 46](#_Toc535670727)

[Hình 4. 12 *Giao diện thực hiện xóa dữ liệu của nhân công* 47](#_Toc535670728)

[Hình 4. 13 *Giao diện kỹ thuật trồng cây* 48](#_Toc535670729)

[Hình 4. 14 *Giao diện kỹ thuật trồng củ cải đỏ và cải bó xôi* 48](#_Toc535670730)

DANH MỤC BẢNG

**No table of figures entries found.**

CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| IOT | Internet of Things |
| LORA | Long Range Radio |
| SF | Spreading Factor |
| GFSK | Gaussian Frequency-shift keying |
| FSK | Frequency-shift keying |
| WIFI | Wireless Fidelity |
| WAN | Wide Area Network |
| MAC | Medium access control |
| PHY | Physical |
| RF | Radio frequency |
| IEEE | Institute of Electrical and Electronics Engineers |
| DSSS | Direct sequence spread spectrum |
| RFID | Radio Frequency Identification |
| SSL | Secure Sockets Layer |
| CDN | Content delivery network |
| API | An application programming interface |
| PWM | Pulse width Modulation |
| UART | Universal Asynchronous Receiver/Transmitter |
| SPI | Serial Peripheral Interface |
| I2C | Inter-Integrated Circuit |
| WSN | Wireless sensor network |
| IDVE | DVE – Data Validation Engine |
| dB | Decibel |

# Giới thiệu

## ĐẶT VẤN ĐỀ

Thực tế hiện nay diện tích rừng bao phủ trên trái đất nói chung và nước ta nói riêng còn lại khá ít. Theo số liệu thống kê nước ta có đến ¾ là diện tích đồi núi, mà đồi núi đó thường là những núi đồi nguyên sơ ít bị sự chạm tay của con người, nhưng hiện tại chỉ sau vài chục năm thì con số đó đã không còn nguyên vẹn. Một phần là do bàn tay con người phá hủy một mặt khác là do biến đổi khí hậu, nắng nóng thất thường dẫn đến cháy rừng.

Thông tin từ cơ quan khí tượng, bên cạnh nguyên nhân trực tiếp là áp thấp nóng phía tây kết hợp với gió Lào, nhiệt độ tăng cao thì hiện tượng cháy rừng còn là hệ quả của hiện tượng biến đổi khí hậu toàn cầu, cộng thêm việc năm 2012 là năm đổi pha, sau La Nina (pha lạnh) là El Nino (pha nóng). [1]

Như vậy, hiện trạng cháy rừng ở Việt Nam là không thể xem thường và phải có phương án xử lý kịp thời.

Theo thống kê 2016 của Tổng cục Lâm nghiệp:

* Năm 2016, nước ta đã xảy ra hàng chục vụ cháy rừng. Diện tích rừng thiệt hại hơn 3000 ha.
* Năm 2017, cả nước xảy ra 182 vụ cháy rừng lớn nhỏ, diện tích rừng thiệt hại là hơn 350 ha.
* Đến đầu năm 2018, có nhiều vụ cháy rừng đã xảy ra trên nhiều địa phương. Chẳng hạn như Lạng Sơn, Lai Châu, Điện Biên, Gia Lai, Đắc Lắc, Vũng Tàu, Bình Thuận, Kiên Giang, Cà Mau… Mặc dù chưa gây thiệt hại lớn về tài nguyên và người, nhưng lại ảnh hưởng lớn đến diễn biến của thời tiết; cần phải nâng cao ý thức trong phòng chống cháy rừng. [2]

Theo Tổng cục Lâm nghiệp, chỉ hơn 5 năm (2012-2017), diện tích rừng tự nhiên bị mất do chuyển mục đích sử dụng rừng tại các dự án được duyệt chiếm 89% tổng diện tích rừng giảm; còn lại là do phá rừng trái pháp luật làm mất 11%. Từ tổng hợp của 58 tỉnh, thành phố trên cả nước, trong khoảng 5 năm qua, các cơ quan nhà nước đã phê duyệt chuyển mục đích sử dụng rừng gần 38.300 ha/1.892 dự án. Trong đó rừng tự nhiên gần 19.000 ha, rừng trồng hơn 15.800 ha, đất chưa có rừng quy hoạch cho lâm nghiệp trên 3.500 ha. [3] Đáng lưu ý, một số vụ phá rừng nghiêm trọng ở các tỉnh Tây Nguyên, Quảng Nam, Quảng Ngãi, Bình Định, Bắc Kạn, Điện Biên được phát hiện sau khi đã phá hủy. Theo Bộ NN & PTNT, chính việc xử lý thiếu kiên quyết, không nhất quán, thậm chí có biểu hiện né tránh trách nhiệm, làm ngơ, tiếp tay cho người phá rừng, gây thiệt hại lớn đối với tài nguyên rừng, gây bức xúc trong xã hội.

Thực tế cháy rừng, phá rừng gây ra nhiều hậu quả rất nghiêm trọng, bao gồm:

* Gây ra tình trạng ô nhiễm môi trường, hiệu ứng nhà kính, biến đổi khí hậu.
* Lũ lụt, xói mòn, sạt lở đất, lũ quét (nếu cháy rừng đầu nguồn, rừng phòng hộ).
* Ô nhiễm không khí. Cháy rừng, khói bụi bay vào không khí khiến cho vùng trời xung quanh bao trùm trong khối không khí bụi bẩn. Người dân sống trong khu vực đó sẽ bị ảnh hưởng nghiêm trọng đến sức khỏe, gây bệnh tật.

Những hệ lụy này hàng chục năm sau cũng không dễ dàng khắc phục được. Vì thế, việc chủ động phòng chống cháy rừng, phá rừng đã và đang được đặt lên hàng đầu cho cả xã hội, đặc biệt vào mùa khô, nắng nóng, nguy cơ cháy rừng dễ xảy ra.

Thấy được thực trạng trên, chúng tôi giới thiệu và trình bày “*Mô hình cảnh báo cháy, phá rừng ứng dụng IoT kết hợp Lora*”. Thực tế, hiện nay có rất nhiều công nghệ truyền dẫn không dây như: Hồng ngoại, Bluetooth, Wifi tuy nhiên hầu hết điều truyền trong khoản cách ngắn. Trong khi đó, LoRa cung cấp khả năng vượt trội về tầm xa. Do đó, đề tài này được thực hiện nhằm đóng góp giải pháp cho việc phát hiện cháy, phá rừng với tầm kiểm soát xa hơn, rộng hơn.

Ý tưởng cốt lõi của hệ thống này là các thông tin từ các cảm biến đã được thu thập, sau đó truyền bằng công nghệ RF LoRa đưa đến bộ xử lý trung tâm và đẩy lên cơ sở dữ liệu thông qua wifi. Website cũng như ứng dụng điện thoại tương tác với cơ sở dữ liệu, giúp người theo dõi từ xa, đưa ra các xử lý cho hệ thống hoạt động một cách phù hợp dựa vào thông tin nhận được từ các cảm biến.

## TÌNH HÌNH NGHIÊN CỨU

Hiện nay trên thế giới có nhiều hãng, công ty đã cung cấp các giải pháp quản lý và cảnh báo sớm cháy rừng.

### Waspmote Plus & Sense

Bên cạnh các sản phẩm đã từng triển khai bằng Waspmote OEM trước đây tại các quốc gia khác, giải pháp này được thiết kế cho thị trường Việt Nam dựa trên dòng sản phẩm ***Waspmote Plug & Sense Smart Environment IoT Vertical Kit* được đóng gói sắn để sử dụng ngoài trời hiệu quả hơn, an toàn hơn và lắp đặt dễ dàng, nhanh chóng.**

**Về thông số cảm biến cơ bản cho sản phẩm này sẽ là:**

Bảng 1‑1 *Thông số cảm biến của Waspmote Plus & Sense*

|  |  |  |
| --- | --- | --- |
| **STT** | **Thông số cảm biến** | **Tần suất** |
| **1** | **Nhiệt độ** | **5 – 10 phút** |
| **2** | **Độ ẩm không khí** | **5 – 10 phút** |
| **3** | **Nồng độ khí CO** | **5 – 10 phút** |
| **4** | **Nồng độ khí CO2** | **5 – 10 phút** |

Hệ thống sử dụng các công nghệ vô tuyến 802.15.4, ZigBee 3, 868, 900, WiFi, 4G, Sigfox and LoRaWAN kết hợp với module GPS để xác định vị trí chính xác nơi thiết bị được lắp đặt. Dữ liệu thu thập sẽ được gửi định kỳ (trong vòng 5 đến 10 phút) đến một thiết bị trung tâm để lưu trữ, gọi là Gateway Meshlium thông qua kết nối Wifi hoặc 3G/GPRS.

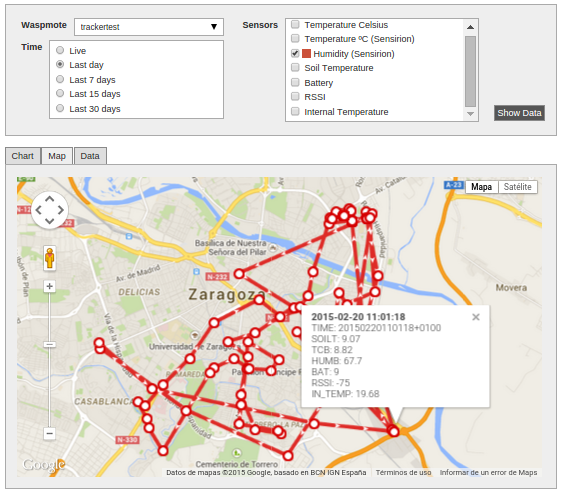
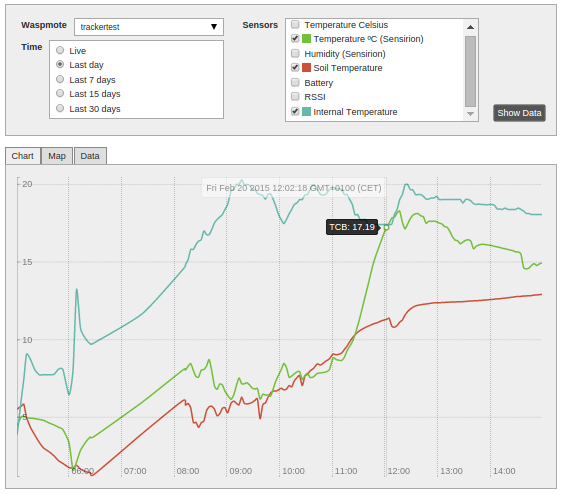


Hình 1‑1 *Thiết bị Waspmote Plus & Sense (bên trái) và Meshlium* [4], [5], [6]

Từ thiết bị trung tâm, phần mềm **PIoT-SE** sẽ cho phép truy nhập dữ liệu từ Gateway Meshlium và phân tích, cảnh báo chạm ngưỡng các dữ liệu đo được gửi về từ các Waspmote Plug & Sense. Bên cạnh các tính năng cơ bản của phần mềm, người quản trị hệ thống còn hoàn toàn có thể tuỳ biến phần tri thức kinh nghiệm trong suốt thời gian điều hành giám sát để tăng độ chính xác của hệ thống, giúp hữu ích hơn trong quá trình khai thác vận hành hệ thống sau này.

Phần mềm vẽ đồ thị và phân tích dữ liệu từ các Waspmote Plug & Sense cho phép:

* Hiển thị dữ liệu từ cảm biến vào đồ thị thời gian thực.
* Cho phép xem dữ liệu tổng hợp giữa các khoản thời gian khác nhau, như xem theo tuần, theo tháng, theo năm...
* So sánh giá trị cảm biến giữa các nút với nhau.
* Định vị được vị trí của các Waspmote Plug & Sense trên bản đồ.
* Dữ liệu cảm biến được lưu trữ sử dụng cơ sở dữ liệu Meshlium MySQL. Dữ liệu có thể được lưu lại vào điện toán đám mây (Cloud platform). [5]

Hình 1‑2  *Chức năng vẽ đồ thị của Meshlium*

Hình 1‑3 *Chức năng định vị của Meshlium*

### Sử dụng vệ tinh MODIS

MODIS (Moderate Resolution Imaging Spectroradiometers) là bộ cảm có độ phân giải trung bình đặt trên vệ tinh TERRA được NASA phóng vào quỹ đạo tháng 12/1999 và vệ tinh AQUA được phóng vào quỹ đạo tháng 5/2002 với mục đích quan trắc, theo dõi các thông tin về mặt đất, đại dương và khí quyển trên phạm vi toàn cầu. Các ứng dụng tiêu biểu có thể kể đến là: nghiên cứu khí quyển, mây, thời tiết, lớp phủ thực vật, biến động về nông nghiệp và lâm nghiệp, ***cháy rừng***, nhiệt độ mặt nước biển và màu nước biển...

Thông thường, cháy rừng thường xảy ra trên diện rộng tại những vùng có địa hình phức tạp nên việc phát hiện rất khó khăn, gây thiệt hại lớn về kinh tế cũng như hệ sinh thái, hệ thống báo cháy qua vệ tinh MODIS ra đời nhằm giải quyết vấn đề này giúp phát hiện các đám cháy lớn nhanh chóng thông qua vệ tinh.

Hình 1‑4 *Hệ thống báo cháy sử dụng MODIS*

MODIS có thể thu được 2 ảnh vào ban ngày và 2 ảnh vào ban đêm, các ảnh này sẽ được xử lý và chiết xuất các điểm cháy kèm với tọa độ địa lý và đưa trực tiếp lên web chỉ sau một giờ từ khi thu nhận ảnh.

Ảnh vệ tinh MODIS là tư liệu miễn phí, có chu kỳ bay chụp ngắn (4 phiên ảnh/ngày) và trường phủ rộng có thể sử dụng để trích xuất các điểm nóng trong phát hiện sớm các vụ cháy rừng ở Việt Nam.

Kết quả kiểm chứng sự xuất hiện điểm dị thường nhiệt của các vụ cháy thực xảy ra trong quá khứ cho thấy trong 100 vụ cháy rừng thì có 71 vụ có điểm dị thường nhiệt xuất hiện cùng thời điểm diễn ra vụ cháy (đạt 71%).

Giá trị được trích xuất từ ảnh vệ tinh MODIS phù hợp cho các vụ cháy rừng ở Việt Nam đạt từ 310K và giá trị về độ lệch nhiệt độ ∆T là từ 10K trở lên cho những vụ cháy vào ban ngày ban đêm. [7], [8]

## mục tiêu đề tài

“Mô hình cảnh báo trộm gỗ, cháy rừng ứng dụng IoT” sẽ phát hiện được các dấu hiệu của sự cháy, hỏa hoạn, trộm gỗ, phá rừng tại vị trí theo dõi của nó, sau đó cảnh báo kịp thời để phòng tránh và đưa ra các phương án xử lý hiệu quả nhất. Đề tài sử dụng ESP8266 NodeMCU làm bộ xử lý trung tâm, các cảm biến nhiệt độ, cảm biến thu âm thanh, nồng độ khí CO trong không khí theo dõi trực tiếp, phát hiện ra dấu hiệu của cháy và trộm gỗ tại nhiều vị trí trong rừng gửi lên máy chủ, có thể theo dõi các thông số thông qua website và ứng dụng điện thoại. Cùng một số thiết bị đầu ra như hiển thị lên màn hình, âm thanh (chuông, còi, loa), bằng tín hiệu phát sáng (đèn)… Khi một khu vực trong hệ thống gặp nguy hiểm như nhiệt độ cao bất thường, phát hiện đám khói, có tiếng động lớn và liên tục là tiếng máy cưa... cả hệ thống báo động và nhận biết đó là khu vực nào.

Tìm hiểu, nghiên cứu một trong những phương thức truyền dữ liệu tiết kiệm năng lượng mới được phát triển và ứng dụng trong những năm gần đây – LoRa.

## nội dung thực hiện

Tìm hiểu và tham khảo các tài liệu, nghiên cứu các chủ đề, các nội dung liên quan đến đề tài. Tìm hiểu về công nghệ truyền thông không dây Lora, ứng dụng của công nghệ IoT.

Lựa chọn các thiết bị, linh kiện trong việc thiết kế mô hình, tìm hiểu và nghiên cứu phần cứng, nguyên lý hoạt động, tính năng của các module.

* Module ESP8266 NodeMCU.
* Module Arduino UNO R3.
* Module LoRa SX 1278 433MHz Ra-02, anten 433 MHz 3dBi 10W.
* Các cảm biến gồm cảm biến nhiệt độ DS18B20, nồng độ khí CO MQ-7, cường độ - tần số âm thanh môi trường Max4466.
* Màn hình LCD 20x4 với I2C.
* Các thành phần khác như: mạch nâng áp, mạch sạc pin, còi báo động, relay, FET, opto, pin li-ion.
* Các chuẩn truyền thông: I2C, SPI.
* Thiết kế và xây dựng máy chủ trên nền tảng NodeJS giao tiếp với các thiết bị người dùng qua giao thức Websocket.
* Tìm hiểu, nghiên cứu Web, hiển thị các giá trị thu được từ cảm biến.
* Thiết kế ứng dụng theo dõi và cảnh báo cho điện thoại hệ điều hành Android, iOS
* Thiết kế hệ thống cảnh báo, theo dõi, lưu đồ giải thuật và chương trình điều khiển mô hình hệ thống.
* Chạy thử nghiệm hệ thống, chỉnh sửa lỗi, hoàn chỉnh mô hình.
* Đánh giá kết quả thực hiện đề tài.
* Viết báo cáo đề tài cho đề tài.

## BỐ CỤC ĐỀ TÀI

Nội dung báo cáo được chia thành 5 chương bao gồm nội dung như sau:

**Chương 1 Giới thiệu**

Trình bày về việc đặt vấn đề, tình hình nghiên cứu, mục tiêu, nội dung thực hiện đề tài và bố cục đề tài.

**Chương 2 Cơ sở lý thuyết**

Giới thiệu về các phần cứng, phần mềm, lý thuyết liên quan đến hệ thống được sử dụng trong đề tài.

**Chương 3 Thiết kế và thi công hệ thống**

Giới thiệu tổng quan về các yêu cầu của đề tài, qua đó tiến hành tính toán, phân tích, lựa chọn và thiết kế hệ thống phù hợp với yêu cầu đặt ra. Trình bày chi tiết quá trình thi công, lắp ráp các thiết bị, kiểm tra đo đạt. Vẽ sơ đồ khối, lưu đồ thể hiện hoạt động của hệ thống.

**Chương 4 Kết quả thực hiện**

Trình bày về kết quả đạt được trong quá trình thi công và vận hành hệ thống, kết quả thực nghiệm của hệ thống từ đó đưa ra nhận xét về hệ thống so với mục tiêu của đề tài.

**Chương 5 Kết luận và hướng phát triển**

Trình bày những kết quả thông qua quá trình làm đồ án đạt được, những hạn chế, từ đó rút ra kết luận và hướng phát triển để giải quyết các vấn đề còn gặp phải, cũng như mở rộng để đề tài hoàn thiện hơn.

# CƠ SỞ LÝ THUYẾT

## TẦM quan trọng của việc Cảnh báo trộm gỗ, cháy tại rừng

Khí CO là sản phẩm của sự cháy không hoàn toàn của Carbon và các hợp chất của Carbon như gỗ, xăng, dầu… Hầu hết các đám cháy điều bắt đầu từ sự cháy không hoàn toàn. Do đó việc phát hiện sớm khí CO có vai trò quan trọng trong việc cảnh báo sớm cháy rừng.

Khi một đám cháy xảy ra, ở những vùng cháy thường có những dấu hiệu sau: lửa, khói, xung quanh chổ cháy bị phá hủy; nhiệt độ vùng cháy tăng lên cao; không khí bị oxy hóa mạnh... Dựa vào những dấu hiệu đó, đặt các cảm biến để phát hiện được đám cháy để phòng tránh và kịp thời đưa ra cảnh báo ngay giai đoạn đầu để phòng và chữa hiệu quả. Ngoài ra việc cảnh báo cháy còn giúp chúng ta liên tục theo dõi để hạn chế các vụ cháy tai hại, tăng cường độ an toàn.

Bên cạch đó các vụ phá rừng, khai thác gỗ trái phép thường chỉ phát hiện sau khi gỗ tại rừng đã hoàn toàn khai thác và vận chuyển ra khỏi rừng. Việc trộm gỗ thường thực hiện bằng máy cưa, khi cưa hạ gỗ sẽ phát ra tiếng máy rất lớn. Thế nên, khi đặt cảm biến thu thập và nhận dạng tiếng máy cưa để kiểm soát, ngăn chặn, phát hiện trộm gỗ kịp thời là cần thiết.

## tổng quan về iot

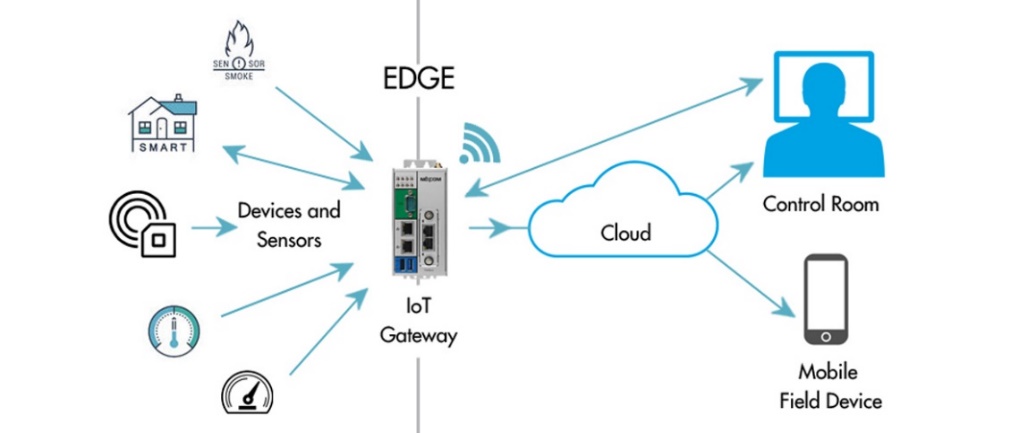
Mạng lưới vạn vật kết nối Internet hoặc mạng lưới thiết bị kết nối Internet (viết tắt là IoT) là nền tảng công nghê ̣mới của thế giới; khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không 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. 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.

IoT sẽ là mạng khổng lồ kết nối tất cả mọi thứ, bao gồm cả con người và sẽ tồn tại các mối quan hệ giữa người và người, người và thiết bị, thiết bị và thiết bị. Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng.

Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông, Bluetooth, ZigBee, hồng ngoại, RF… Các thiết bị có thể là điện thoại thông minh, tivi, máy giặt, tai nghe, bóng đèn, cảm biến và rất nhiều thiết bị khác.

## TỔNG QUAN VỀ GATEWAY

Gateway là một thiết bị đóng vai trò là điểm kết nối giữa nơi lưu trữ (đám mây) và các thiết bị điều khiển, cảm biến và các thiết bị thông minh… Tất cả dữ liệu chuyển lên đám mây hoặc ngược lại sẽ đi qua gateway. Gateway thường sẽ là một thiết bị phần cứng chuyên dụng, có thiết kế linh hoạt, chịu được môi trường khắc nghiệt, hỗ trợ các chuẩn kết nối cục bộ như: LAN, WiFi, 3G, Zigbee, Z-wave, RF.



Hình 2‑1 *Mô hình IOT kết hợp Gateway*

IoT gateway cung cấp một nơi để xử lý trước dữ liệu cục bộ trước khi nó được gửi lên đám mây. Khi dữ liệu được tổng hợp, thu gọn và phân tích, nó sẽ giảm thiểu khối lượng dữ liệu cần chuyển lên đám mây, điều này có thể tác động lớn đến thời gian hồi đáp và chi phí đường truyền mạng.

Một lợi ích khác của cổng IoT là nó có thể cung cấp cơ chế bảo mật bổ sung cho mạng IoT và dữ liệu mà nó vận chuyển. Vì gateway quản lý thông tin di chuyển theo cả hai chiều, nó có thể bảo vệ dữ liệu khi di chuyển lên đám mây khỏi các rò rỉ, hạn chế các thiết bị IoT bị xâm phạm bởi nguồn tấn công bên ngoài với các tính năng như phát hiện giả mạo, mã hóa, tạo số ngẫu nhiên bằng phần cứng và công cụ mã hóa.

Trên thị trường có rất nhiều vi xử lý phù hợp với các tiêu chí để dùng làm Gateway, nhưng với mục tiêu của đề tài là yêu cầu phải giao tiếp giữa Node và Server, với phương thức truyền không dây là LoRa và cần lưu trữ dữ liệu từ các Node gửi lên. Do đó, ESP wifi có thể đáp ứng được yêu cầu trên và cũng vì mức độ phổ biến của trên thị trường.

## công nghệ truyền thông KHÔNG DÂY LORA

### Tổng quan

LoRa dựa trên nền tảng sóng radio tầm xa công suất thấp được nghiên cứu và phát triển bởi Cycleo và sau này được mua lại bởi công ty Semtech năm 2012. LoRa hướng tới các kết nối M2M ở khoảng cách lớn mà không cần các mạch khuếch đại công suất. Nó có thể hỗ trợ liên lạc ở khoảng cách lên tới 15 – 20 km, với hàng triệu node mạng. Nó có thể hoạt động trên băng tần không phải cấp phép, với tốc độ thấp từ 0,3kbps đến khoảng 30kbps.

Một mạng LoRa có thể cung cấp vùng phủ sóng tương tự như của một mạng di động. Trong một số trường hợp, các ăng-ten Lora có thể được kết hợp với ăng-ten di động khi các tần số là gần nhau, do đó giúp tiết kiệm đáng kể chi phí. Công nghệ không dây LoRa được đánh giá là lý tưởng để sử dụng trong một loạt các ứng dụng, bao gồm: định lượng thông minh, theo dõi hàng tồn kho, giám sát dữ liệu của máy bán hàng tự động, ngành công nghiệp ô tô, các ứng dụng tiện ích và trong bất cứ lĩnh vực nào mà cần báo cáo và kiểm soát dữ liệu.

LoRa sử dụng kỹ thuật điều chế gọi là Chirp Spread Spectrum. Có thể hiểu nguyên lý này là dữ liệu sẽ được băm bằng các xung cao tần để tạo ra tín hiệu có dãy tần số cao hơn tần số của dữ liệu gốc (gọi là chipped), sau đó tín hiệu cao tần này tiếp tục được mã hoá theo các chuỗi chirp signal (là các tín hiệu hình sin có tần số thay đổi theo thời gian; có 2 loại chirp signal là up-chirp có tần số tăng theo thời gian và down-chirp có tần số giảm theo thời gian; và việc mã hoá theo nguyên tắc bit 1 sẽ sử dụng up-chirp, và bit 0 sẽ sử dụng down-chirp) trước khi truyền ra ăng-ten để gửi đi.

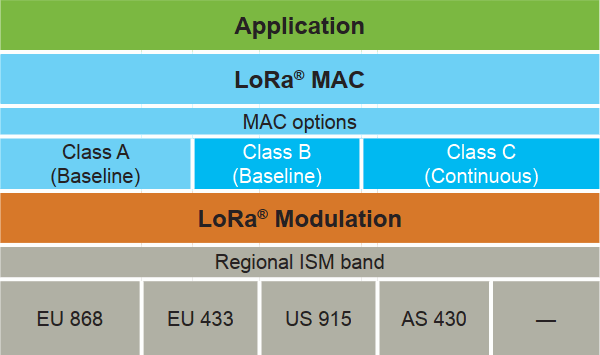
Băng tần làm việc của LoRa từ 430MHz đến 915MHz cho từng khu vực khác nhau trên thế giới: 430MHz cho châu Á, 780MHz cho Trung Quốc, 433MHz hoặc 866MHz cho châu Âu, 915MHz cho Mỹ.

Nhờ sử dụng chirp signal mà các tín hiệu LoRa với các chirp rate khác nhau có thể hoạt động trong cùng 1 khu vực mà không gây nhiễu cho nhau. Điều này cho phép nhiều thiết bị LoRa có thể trao đổi dữ liệu trên nhiều kênh đồng thời (mỗi kênh cho 1 chirprate). [9], [10]

### Sơ lược về LoraWan và cấu trúc của gói tin Lora

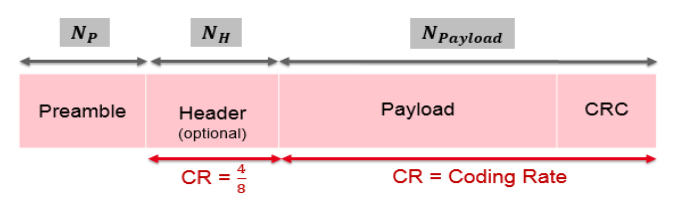
Trong khi Lora là lớp vật lý (physical layer) cho phép thiết lập các kết nối để truyền nhận dữ liệu tầm xa thì LoRaWAN là một kiến trúc hệ thống và giao thức truyền thông trong mạng dựa trên Lora.

Với Lora, có thể gửi dữ liệu với tần suất, kích thước và format tùy ý mình trong khi LoraWAN qui định chặt chẽ trong giao thức và kiến trúc mạng, là những nhân tố có ảnh hưởng nhất đến việc tăng tuổi thọ pin của một “node”, dung lượng mạng (network capacity), chất lượng dịch vụ, mức độ bảo mật, và hàng loạt các ứng dụng được phục vụ bởi network.

Cấu trúc phần mềm cơ bản của một thiết bị hỗ trợ LoRaWAN như sau: [11]

Hình 2‑2 *Cấu trúc cơ bản của LoRaWAN*

Trong cấu trúc trên thì LoRaWan gồm LoRa Mac (Class A, Class B, Class C) và hoạt động dựa trên lớp vật lý là chip LoRa. Để biết chính xác tần số cho phép hoạt động của LoRa tại các khu vực trên thế giới, có thể tìm kiếm ở trong tài liệu “LoRaWAN Regional Parameter” được cung cấp bởi tổ chức LoRa Alliance.

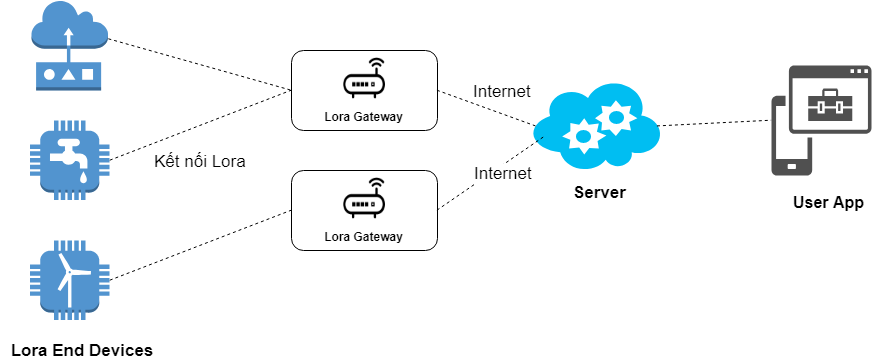


Hình 2.3 Cấu trúc của gói tin Lora

Trong đó:

* Preamble: chuỗi binary để bộ nhận phát hiện tín hiệu của gói tin Lora trong không khí
* Header: thông tin về kích cỡ gói tin truyền đi.
* Payload: là dữ liệu truyền qua Lora.

### Các thành phần trong mạng Lora

Một mạng Lora cơ bản có cấu trúc như sau:

Hình 2‑3 *Cấu trúc cơ bản của một mạng LoRa*

Lora End Devices: là các thiết bị cuối có kết nối Lora giúp thu thập dữ liệu và điều khiển hệ thống, liên lạc trực tiếp với các gateway thông qua kết nối Lora.

Lora Gateways: Do các thiết bị Lora không có kết nối Internet nên các gateway sẽ làm nhiệm vụ là trung gian tiếp nhận các dữ liệu gửi đến từ các Lora Sensors và đẩy lên server qua kết nối Internet và ngược lại sẽ nhận lệnh từ server và gửi đến các Lora Sensors thông qua kết nối Lora.

Server: là trung tâm lưu trữ và xử lý dữ liệu trong hệ thống giúp kết nối các thành phần trong mạng, có thể triển khai trên cloud hoặc máy chủ cục bộ.

User app: giúp người dùng có thể quan sát, theo dõi các thông tin gửi về và ra lệnh bật tắt các thiết bị, phổ biến nhất là một ứng dụng web hoặc mobile. [10]

## tổng quan về ARDUINO

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, 32-bit. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

### Phần cứng

Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện thêm vào giúp dễ dàng lập trình và có thể mở rộng với các mạch khác. Arduino chính thức sử dụng các dòng chip megaAVR, đặc biệt là ATmega8, ATmega168, ATmega328, ATmega1280 và ATmega2560. Hầu hết các mạch gồm một bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16 MHz. Một vi điều khiển Arduino cũng có thể được lập trình sẵn với một boot loader cho phép đơn giản là upload chương trình vào bộ nhớ flash on-chip, so với các thiết bị khác thường phải cần một bộ nạp bên ngoài. Điều này giúp cho việc sử dụng Arduino được trực tiếp hơn bằng cách cho phép sử dụng 1 máy tính gốc như là một bộ nạp chương trình.

Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài

### Phần mềm

Môi trường phát triển tích hợp (IDE) của Arduino là một ứng dụng cross-platform (đa nền tảng) được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó bao gồm một chương trình code editor với các chức năng như đánh dấu cú pháp, tự động brace matching, và tự động canh lề, cũng như compile (biên dịch) và upload chương trình lên board. Một chương trình hoặc code viết cho Arduino được gọi là một sketch.

Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được:

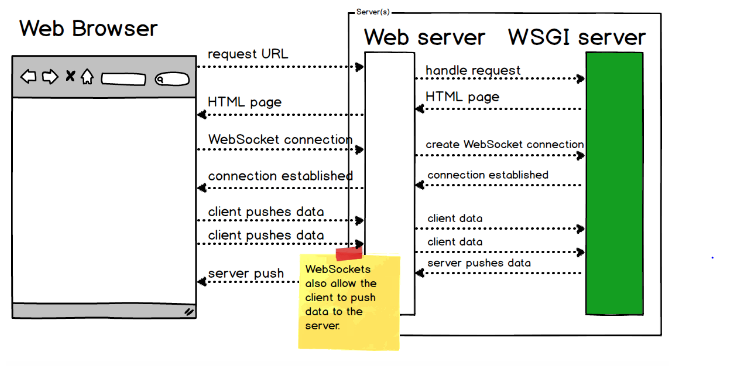
* **Setup ():** Hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt.
* **Loop ():** Hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch.
* **Chuẩn Giao tiếp:**
* **Serial:** Đây là chuẩn giao tiếp nối tiếp được dùng rất phổ biến trên các bo mạch Arduino. Mỗi board có trang bị một số cổng Serial cứng (việc giao tiếp do phần cứng trong chip thực hiện).
* **USB:** Các board Arduino tiêu chuẩn đều có trang bị một cổng USB để thực hiện kết nối với máy tính dùng cho việc tải chương trình.
* **SPI:**  Đây là một chuẩn giao tiếp nối tiếp đồng bộ và bus gồm 4 dây, có thể kết nối với các thiết bị như LCD, bộ điều khiển cảm biến các loại…
* **I2C:** Đây là một chuẩn giao tiếp đồng bộ và bus chỉ có hai dây, có thể giao tiếp với một số loại cảm biến như thermostat (bộ điều chỉnh nhiệt độ) của CPU, tốc độ quạt, một số màn hình OLED/LCD, đọc real-time clock …

## phần mềm

### Giao Thức Websocket

Websocket là giao thức chuẩn cho trao đổi dữ liệu hai chiều giữa người dùng và máy chủ. Giao thức WebSocket không chạy trên HTTP, thay vào đó nó thực hiện trên giao thức TCP.

Người ta thường dùng Websocket thay vì HTTP cho những trường hợp yêu cầu real time. Ví dụ muốn hiện thị biểu đồ, chỉ số chứng khoán, web chat.



**Hình 2‑5** *Mô hình hoạt động của Websocket*

Gói tin của WebSocket nhẹ hơn HTTP rất nhiều, giảm độ trễ của network lên đến 3 lần, không cần phải gửi request liên tiếp như HTTP.

Chuẩn giao thức của websocket là ws:// cho chuẩn thông thường và wss:// cho chuẩn secure (tương tự http:// và https://).

### NodeJs

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web. NodeJs là một mã nguồn mở có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJs cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất thông qua NPM.

NodeJS có một số ưu điểm như sau:

* ***Không đồng bộ (none-blocking)***: Tất cả các API của NodeJs đều không đồng bộ nó chủ yếu dựa trên nền của NodeJs Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của NodeJs giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Real time).
* ***Tốc độ cao:*** NodeJs được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* ***Đơn luồng nhưng có khả năng mở rộng cao:*** NodeJs sử dụng một mô hình luồng duy nhất với sự kiện lặp. Cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ có khả năng mở rộng cao, trái ngược với các máy chủ truyền thống. NodeJs sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* ***Không đệm:*** NodeJs không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.

### React native

#### Giới thiệu

React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

Chúng ta sẽ xây dựng được ứng dụng thuần di động (native), và cũng có thể xây dựng ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ xây dựng trên iOS hay Android mà chúng ta tạo ra và chạy được cả hai nền tảng.

Một điểm cộng cho React Native là giảm chi phí biên dịch lại bằng cách sử dụng Hot-Loading tức là bạn không cần phải xây dựng lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc viết và chạy ứng dụng nữa.

Và điểm lợi hại kế tiếp của React Native đó chính là chúng ta chỉ cần sử dụng Javascript để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà ứng dụng di động thuần gặp phải đã nêu ở trên. Và rồi còn cả kết hợp với những ngôn ngữ thuần cho di động như Swift, Java…

#### Ưu và nhược điểm của React Native

***Ưu điểm:***

* Hiệu quả về mặt thời gian khi muốn phát triển một ứng dụng nhanh chóng.
* Hiệu năng tương đối ổn định.
* Cộng đồng phát triển mạnh. Tiết kiệm tiền.
* Ứng dụng tin cậy và ổn định.
* Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
* Trải nghiệm người dùng (UX) tốt hơn.

***Nhược điểm:***

* Vẫn đòi hỏi native code.
* Hiệu năng sẽ thấp hơn với ứng dụng thuần native code.
* Bảo mật không cao do dựa trên Javascript.
* Quản lý bộ nhớ.
* Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

### Cơ Sở Dữ Liệu MongoDB

#### Giới thiệu

Đây là một cross-platform, mã nguồn mở, cơ sở dữ liệu hướng documents với hiệu năng cao, có thể đáp ứng cho đa dạng các hệ thống và dễ dàng để phát triển. Tư tưởng của MongoDB là thông qua khái niệm collection và document. Vì không tìm được từ tiếng Việt dịch sát nghĩa hơn nên tạm thời cho phép mình giữ nguyên các khái niệm tiếng Anh của 2 thuật ngữ này.

MongoDB thuộc NoSQL (None-Relational SQL), được phát triển dựa trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu khóa – giá trị (key – value).

#### Tại sao nên sử dụng NoSQL

***Linh hoạt:*** Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.

***Khả năng thay đổi quy mô:*** Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.

***Hiệu năng cao:*** Cơ sở dữ liệu NoSQL được tối ưu hóa theo mô hình dữ liệu (ví dụ như văn bản, khóa–giá trị và đồ thị) và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.

***Cực kỳ thiết thực:*** Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

## LỰA CHỌN VÀ THIẾT LẬP MÔ HÌNH NĂNG LƯỢNG

### Các thiết bị IoT luôn đi kèm với những hạn chế và một trong số đó là yêu cầu về tính tiết kiệm năng lượng và khi chúng được đặt xa trung tâm điều khiển luôn đòi hỏi một hệ thống cung cấp năng lượng hoạt động liên tục.

Và để các thiết bị IoT hay các module hoạt động ổn định và liên tục trong thời gian dài cần thiết phải có hệ thống năng lượng bền vững. Qua đó, chúng tôi đã tính toán và chọn lựa các giải pháp phù hợp để thiết lập năng lượng cho hệ thống.

Năng lượng cần để vận hành các thiết bị cảm biến được chia làm 3 khối chính:

* Năng lượng cho việc thu thập dữ liệu (acquisition).
* Xử lý dữ liệu (processing).
* Truyền dữ liệu (networking).

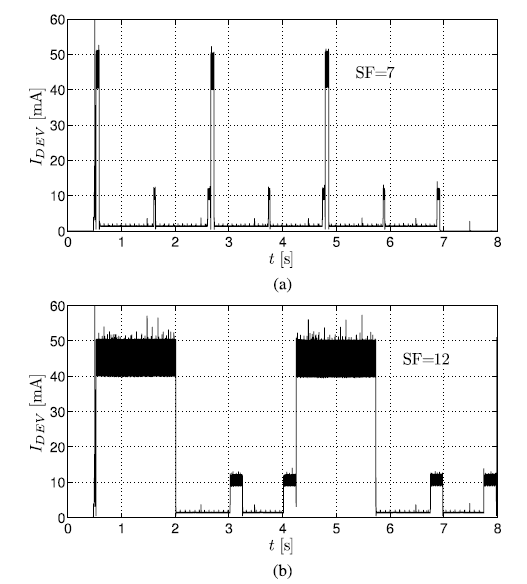


**Hình 2‑6** *Biểu đồ sử dụng năng lương theo thời gian của thiết bị*

Với mô hình như trên, chúng tôi sẽ phân tích chi tiết từng khối. Tính toán năng lượng tiêu thụ trung bình trong quá trình thực hiện nhiệm vụ tương ứng. [6]

### Năng lượng trong quá trình truyền dữ liệu

Công nghệ LoRa sử dụng các spreading factor (SF) khác nhau để điều chỉnh phạm vi cũng như năng lương tiêu thụ của việc truyền dẫn. Để tăng phạm vi truyền, LoRa cho phép cấu hình SF từ 7 đến 12 (128-4096 bits/symbol). Việc thay đổi SF này gián tiếp làm giảm tốc độ bit, ảnh hưởng đến thời gian truyền tải một gói tin. Do đó, làm thay đổi công suất tiêu thụ.



**Hình 2‑7** *Mức tiêu thụ năng lượng của LoRa với 2 mức spreading factor*

1. SF = 7 b) SF = 12

Về cơ bản, mỗi mức thay đổi SF làm tăng 2 lần thời gian radio ở trong trạng thái hoạt động. Do đó công suất tiêu thụ được tính theo hàm mũ:

|  |  |  |
| --- | --- | --- |
|  | *H(N)* = *O(h(N))* = 2*N* | (3) |

Ta được công thức thể hiện mối quan hệ giữa và số spreading factor:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Trong đó:

: Điện tích trung bình của khối network

: Điện tích trung bình để truyền một gói tin

NSF: Số Spreading factor

: Điện tích trung bình để khối hoạt động ở chế độ chờ.

Từ (3) ta dễ dàng có được:

|  |  |  |
| --- | --- | --- |
|  | NET = + | (5) |

TMSG: là thời gian truyền gói tin

: Dòng của khối network khi hoạt động ở chết độ standby.

### Năng lượng thu thập dữ liệu

Dòng điện cần cung cấp cho quá trình nhận giá trị từ cảm biến được tính theo công thức:

|  |  |  |
| --- | --- | --- |
|  | + | (6) |

Trong đó:

: Dòng điện cho khối cảm biến

: Điện tích cần thiết cho một mẫu, để thực hiện quá trình nhận và chuyển vào bộ nhớ chuyển bị cho quá trình xử lý.

TRCD: Thời gian giữa hai phiên thu nhận dữ liệu liên tiếp.

NS: Số mẫu lấy được trong thời gian TRCD.

B: Dòng điện ở chế độ stand-by của cảm biến

Trong đề tài, chúng tôi sử dụng 3 cảm biến: DS18B20, MQ-7 và Max4466. Dòng điện cho Arduino Uno thực hiện quá trình nhận và lưu dữ liệu xem như không đổi. Do đó công thức (5) được chuyển đổi như sau:

|  |  |  |
| --- | --- | --- |
|  | + (MQ-7 + DS18B20+ Max4466) | (7) |

### Năng lượng xử lý dữ liệu

Để xác định năng lượng cho quá trình xử lý dữ liệu cần thiết phải biết các tham số tham gia vào hành vi thuật toán. Đối với các thuật toán đã biết, độ phức tạp của thuật toán được xác định bởi hàm *T(n)*. Mặt khác, nếu thuật toán được thiết kế tùy chỉnh, độ phức tạp không rõ ràng độ phức tạp của thuật toán nên được suy ra từ dữ liệu thực nghiệm hay giả lặp [24].

Độ phức tạp của thuật toán được tính dựa vào thuật toán Fast Fourier Transform (FFT). Theo FFT độ phức tạp T(n) = O(f(n)) = N.log(N). Theo đó, thời gian để xử lý sẽ tỷ lệ thuận với hệ số này. Như vậy ta có công thức tính năng lượng xử lý với độ chính xác hợp lý:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

Trong đó:

k: Hằng số đại diện cho số lượng phép toán thực hiện trên mỗi điểm FFT.

: Điện tích cho 1 phép toán số học.

: Dòng tiêu hao cho tất cả các chức năng của bộ xử lý.

# THIẾT KẾ VÀ THI CÔNG

## YÊU CẦU ĐỐI VỚI HỆ THỐNG

Hệ thống được thiết kế với giả thiết dành cho các ứng dụng theo dõi, cảnh báo và quản lý các khu rừng trên diện tích lớn dựa trên nền tảng IoT kết hợp Lora. Kết hợp theo dõi năng lượng tiêu thụ tại nơi thu thập theo thời gian. Từ đó, người dùng có thể kiểm soát và bảo trì hệ thống, giúp hệ thống hoạt động ổn định, dữ liệu được gửi về máy chủ đều đặn.

Hệ thống được thiết kế với các yêu cầu cần đạt như sau:

* Hệ thống thiết kế gồm một Gateway và hai Node có thể giao tiếp với nhau, và Gateway có thể đẩy dữ liệu lên server thông qua giao thức websocket.
* Hệ thống phải thu thập, cung cấp các thông tin về điều kiện môi trường tại điểm theo dõi bao gồm nhiệt độ không khí, phát hiện khói của đám cháy và phát hiện âm thanh của máy cưa trộm gỗ;
* Hệ thống hoạt động bằng nguồn pin, thay đổi chu kỳ thu thập và gửi dữ liệu để tiết kiệm năng lượng. Khả năng tự bổ sung năng lượng đã mất bằng tấm pin năng lượng mặt trời. Đồng thời đọc giá trị năng lượng Pin.
* Đáp ứng được truyền dữ liệu từ Node đến Gateway là khoảng cách xa cỡ 2 – 5Km.
* Phải cho phép người dùng có thể truy cập thông tin về nhiệt độ, nồng độ CO, trên ứng dụng điện thoại thông minh, website.
* Hệ thống phải hoạt động và cảnh báo, hiển thị thông tin thu thập tại gateway khi xảy ra tình trạng mất điện tạm thời.
* Quan trọng hết, hệ thống phải quản lý được năng lượng tại các node cảm biến.

## ĐẶC TẢ HỆ THỐNG

### Chức năng của hệ thống

Dựa trên các yêu cầu về tính năng của hệ thống đã được trình bày, nhóm đã góp nhặt, đánh giá và lên ý tưởng xây dựng một hệ thống phát hiện và cảnh báo cháy, trộm gỗ gồm các chức năng:

Các trạm cảm biến (tạm gọi là node 1, node 2) được đặt tại hai khu vực trong rừng, thu thập các thông số của môi trường bao gồm nhiệt độ không khí, nồng độ khí CO trong không khí và âm thanh của máy cắt trộm gỗ và đọc giá trị năng lượng Pin. Dữ liệu tại các Node được gửi lên trạm trung tâm gateway thông qua LoRa.

Tại gateway, dữ liệu sẽ được truyền lên website và ứng dụng điện thoại thông qua giao thức websocket và hiển thị thông tin cho người dùng. Tại đây, người dùng có thể quan sát được dữ liệu tại các Node cũng như theo dõi tình hình tại khu rừng, hơn nữa, người dùng có thể thống kê các thông tin được thu thập theo tuần, tháng hoặc năm.

Hệ thống có chức năng kết nối với internet và truyền nhận dữ liệu giữa các node thông qua các mạng truyền thông không dây Lora.

Hệ thống hiển thị các thông tin cần thiết để cho phép người dùng theo dõi về điều kiện môi trường và tình hình hỏa hoạn của khu rừng qua điện thoại thông minh hoặc website.

Hệ thống cần xác thực thông tin người dùng nhằm đảm bảo tính an toàn cho dữ liệu, bảo mật cho hệ thống.

Hệ thống cho phép người dùng có thể can thiệp, điều khiển các Node từ xa qua ứng dụng điện thoại thông minh để thay đổi chu kỳ nhận dữ liệu từ các Node.

Trạm Gateway nhận dữ liệu được từ các Node, xử lý dữ liệu để hiển thị ra LCD theo dõi, giám sát tại chỗ và thực hiện cảnh báo bằng còi. Đồng thời đẩy dữ liệu lên server. Khi có sự cố xảy ra (cháy rừng, trộm gỗ), hệ thống sẽ gửi thông báo về phía người dùng qua các thông báo trên điện thoại hoặc website.

Hệ thống có chức năng sạc Pin tại Gateway khi nguồn năng lượng yếu dần qua việc cấp sạc bằng công tắc và năng lượng Pin này để duy trì hoạt động khi có sự cố mất điện tạm thời.

Tại Node thu thập âm thanh của máy cưa, khi phát hiện có trộm gỗ sẽ đưa ra cảnh báo mở còi.

Trang web hiển thị dữ liệu cảm biến thu thập, năng lượng Pin, cập nhật dòng thông báo, theo dõi tình trạng năng lượng Pin tiêu thụ và cảnh báo thông số năng lượng thấp hơn ngưỡng.

Ứng dụng trên di động đọc giá trị dữ liệu trên server và hiển thị, có chức năng chọn thời gian để nhận dữ liệu từ nơi thu thập.

Báo động trên điện thoại giúp ta đáp ứng nhanh các thông tin về sự cố để được xử lí kịp thời.

### Đặc tính của hệ thống

Với trường hợp hệ thống được phát triển dành cho các việc kiểm soát, theo dõi tình hình hỏa hoạn, cháy ở rừng cũng như trộm gỗ phải đáp ứng các yêu cầu về đặc tính như sau:

Hệ thống thu thập thông tin chính xác, nhanh chóng, đồng thời mạng đầu cuối phải có công suất tiêu thụ thấp để có thể triển khai trên diện rộng và phải có tính di động.

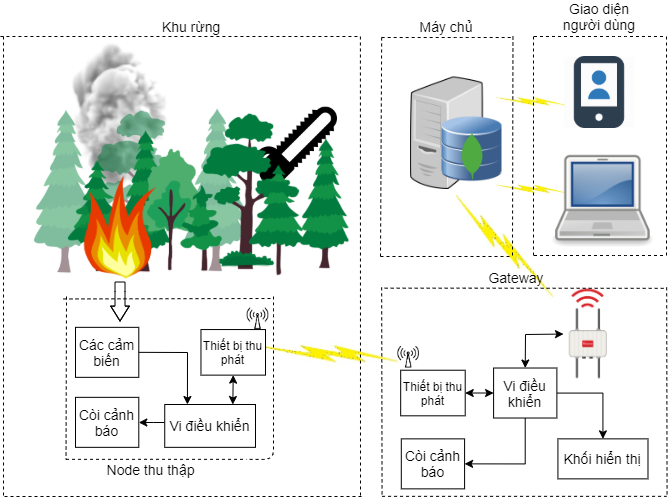
Các trang thiết bị được lựa chọn sử dụng trong hệ thống phải hợp lý, độ sai số thấp; đáp ứng cho các mô hình ứng dụng vừa và nhỏ.

Hệ thống phải được thiết kế nhỏ gọn, linh động, giá thành hợp lý, lựa chọn các thiết bị, linh kiện phải đảm bảo thuận tiện cho việc bảo trì hệ thống trong suốt vòng đời hoạt động của nó.

Giao diện được thiết kế trên điện thoại di động và website đầy đủ thông tin, đẹp mắt, thân thiện với người dùng, đơn giản và dễ sử dụng.

Mặt khác, hệ thống được đưa ra vận hành ngoài trời nên các thiết bị và linh kiện cần được bảo vệ để tránh khỏi các tác nhân không mong muốn từ môi tường tự nhiên một cách tốt nhất.

### Mô hình tổng thể hệ thống

Dựa trên các yêu cầu ban đầu đặt ra, cũng như chức năng của hệ thống *Thiết kế và thi công mô hình cảnh báo trộm gỗ cháy rừng ứng dụng IoT*như đã nêu ở phần trước, mô hình tổng thể của hệ thống được đề xuất như hình 3.1.

Hình 3‑1 *Mô hình tổng thể của hệ thống*

Mô tả mô hình hệ thống:

**Node thu thập:** Gồm các cảm biến thu thập thông tin về nhiệt độ môi trường, khói từ đám cháy (nồng độ CO trong không khí), phát hiện tần số âm thanh đặt tại các điểm được chỉ định trong khu rừng và truyền dữ liệu về gateway. Khoảng cách truyền nhận dữ liệu giữa các nút cảm biến trong mạng cảm biến với gateway khoảng từ 2 km đến 3 km.

**Khối Gateway:** Gateway đóng vai trò là thiết bị trung tâm chuyển đổi giữa hai giao thức truyền không dây LoRa và WiFi. Đồng thời nhận các thông số dữ liệu từ Node cảm biến và sau đó cập nhật lên cơ sở dữ liệu.

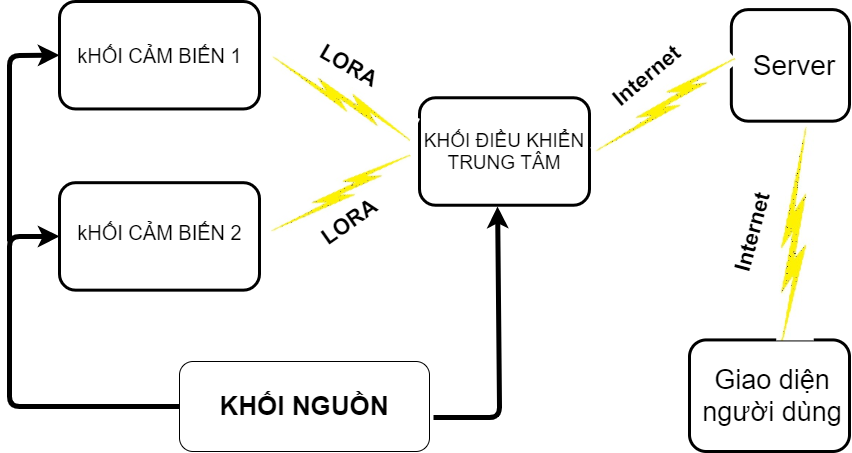
Mặt khác là nhận tín hiệu điều khiển từ web hoặc ứng dụng điện thoại để gửi xuống lại cho khối Node thực hiện. Đồng thời hiện thị các dữ liệu ra LCD,.

**Khối cơ sở dữ liệu:** sử dụngMonggo DB là một hệ cơ sở dữ liệu thời gian thực, tự xây dựng và phát triển.

**Khối giao diện người dùng:** là ứng trên điện thoại thông minh và website lấy dữ liệu trên cơ sở dữ liệu để hiển thị cho người dùng theo dõi, đồng thời nhận lệnh điều khiển từ người dùng.

## KIẾN TRÚC CỦA HỆ THỐNG

### Sơ đồ khối của hệ thống

Dựa vào các yêu cầu trên cùng với mô hình tổng quan hệ thống nhóm thực hiện đề tài đề xuất sơ đồ khối như bên dưới:

Hình 3‑2 *Sơ đồ khối của hệ thống*

**Chức năng của các khối:**

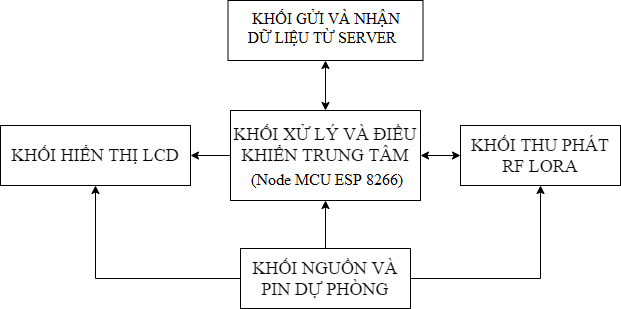
* **Khối cảm biến 1, 2** (Node 1, 2) đều có chức năng như nhau:
* Đọc và gửi giá trị cảm biến về khối điều khiển trung tâm.
* Được cung cấp năng lượng bằng Pin để hoạt động, đọc giá trị Pin gửi về khối trung tâm.
* Có nhiều chế độ hoạt động để tiết kiệm năng lượng.
* Cảnh báo trộm gỗ tại chổ bằng còi báo động.
* **Khối điều khiển trung tâm:**
* Nhận dữ liệu cảm biến và xử lý, kiểm tra nồng độ CO và nhiệt độ tăng bất thường để cảnh báo cháy. Xử lý dữ liệu của âm thanh máy cưa trộm gỗ và đưa ra cảnh báo. Thực hiện cảnh báo qua còi báo động.
* Hiển thị dữ liệu bằng màn hình LCD.
* Được hoạt động bằng Pin khi xảy ra mất điện tạm thời, đọc giá trị Pin.
* Đẩy dữ liệu lên cơ sở dữ liệu.
* **Khối giao tiếp người dùng:**
* Đọc giá trị cảm biến từ Server và hiển thị lại màn hình điện thoại và giao diện website.
* Chọn chế độ để thay đổi thời gian nhận dữ liệu từ khối cảm biến.
* Khi xảy ra cháy hoặc trộm gỗ, hệ thống gửi tín hiệu cảnh báo cho người dùng trên ứng dụng của điện thoại và website.
* **Server:**
* Nhận và lưu dữ liệu từ khối điều khiển trung tâm, phân tích và tổng hợp dữ liệu.
* Tính toán giá trị trung bình theo ngày, tháng năm.
* Xác thực người dùng
* **Khối nguồn:** Cung cấp nguồn hoạt động thích hợp cho các khối hoạt động.

### Phân tích sơ đồ khối và lựa chọn linh kiện thiết kế

#### Tại khối điều khiển trung tâm - Gateway

* **Gateway** là cổng liên lạc giúp chuyển đổi các giao thức giữa thiết bị và internet. Trong các hệ thống IoT, Arduino, Raspberry Pi và NodeMCU thường được dùng làm gateway vì đều đáp ứng được các chức năng cơ bản thêm vào đó là đều có kích thước nhỏ gọn, độ phổ biến cao và cộng đồng người dùng lớn.

Với mục tiêu của đề tài là yêu cầu phải giao tiếp giữa Node và server với hai phương thức truyền không dây là LoRa và WiFi thì các module này đều có thể đáp ứng. Với lại tại gateway chỉ cần số ít các chân I/O và là trung gian nhận dữ liệu từ các Node và truyền lên server, do đó, NodeMCU có thể đáp ứng được yêu cầu trên và giá thành tương đối rẻ, nên chúng tôi quyết định lựa chọn module này làm gateway cho đề tài.

****Dựa trên sơ đồ khối hệ thống, chúng tôi thiết kế sơ đồ khối chi tiết tại Gateway như sau:

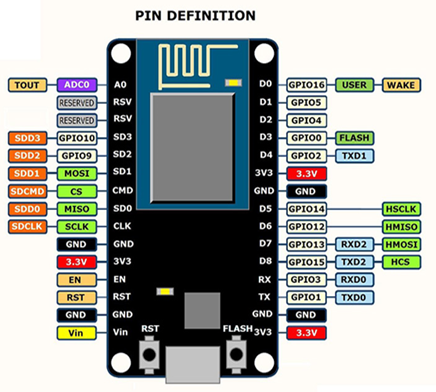
Hình 3‑3  *Sơ đồ khối tại gateway*

* **Khối xử lý và điều khiển trung tâm**

Kit thu phát Wifi ESP8266 NodeMCU là kit phát triển dựa trên nền chip Wifi SoC ESP8266 (ESP-12) với thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code. Nó sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver.

*Thông số kỹ thuật Kit Wi-fi ESP8266 NodeMCU*

* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Điện áp hoạt động: 3.3V
* Điện áp vào: cấp 5V thông qua cổng USB hoặc Vin.
* Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0). Số chân Analog Input: 1
* Bộ nhớ Flash: 4MB
* Hỗ trợ bảo mật: WPA/WPA2. Tích hợp: TCP/IP
* Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU – Lua

**

Hình 3‑4 *Các chân vào ra và chuẩn giao tiếp của NodeMCU Esp8266.*

Có tới 17 chân GPIO, tất cả các chân IO được bảo vệ khỏi điện áp quá mức với mạch đảo được kết nối giữa mạch dẫn và GND. Điện áp đảo thường là khoảng 6V và điện áp giữ là 5.8V.

* **Khối thu phát LoRa**

Là trung tâm xử lý và giao tiếp chính của hệ thống, đảm bảo sự liên lạc giữa gateway với các node. Hiện nay, trên thị trường phổ biến gồm các mô-đun giaotiếp với vi điều khiển thông qua giao tiếp SPI hoặc tích hợp chuyển đổi UART-USB.

****Qua tìm hiểu, chúng tôi thấy rằng mô-đun LoRa SX1278 433MHz RA-02 với giao tiếp SPI có ưu điểm là phổ biến trên thị trường, không cần bộ chuyển đổi UART do đó có thể hoạt động với các kit không hỗ trợ đầu USB cùng với đó là giá thành rẻ, các thông số có thể trực tiếp can thiệp bằng lập trình, khoảng cách truyền lý thuyết lên đến 10km, phù hợp với những yêu cầu đặt ra của đề tài nên chúng tôi sẽ sử dụng mô-đun này. Bên cạnh đó nhằm đảm bảo tính đồng bộ nên chúng tôi cũng sẽ sử dụng mô-đun này ở các node.

Hình 3‑5 *Mô-đun LoRa SX1278 433MHz Ra-02 và anten 3dBi*

Dưới đây là thông số kỹ thuật Module LoRa và sơ đồ kết nối chân giữa ESP8266 NodeMCU và mô-đun tại gateway:

Bảng 3‑1 *Thông số kỹ thuật Module LoRa*

|  |  |
| --- | --- |
| Model | LoRa RA-02 SX1278 433 Mhz |
| IC chính | SX1278 đến từ SEMTECH |
| Điện áp hoạt động | 1.8 – 3.7 VDC |
| Tần số hoạt động | 420 - 450 Mhz |
| Công suất | 20 dBm (10 mW) |
| Dòng điện ở chế độ Sleep | 1,7 µA |
| Dòng điện khi gửi bản tin | 100mA |
| Dòng điện khi nhận | 40mA |
| Chuẩn giao tiếp | SPI half-duplex/GPIO |
| Độ dài chuỗi khi nhận | Gói dữ liệu 256 bytes |
| Khoảng cách truyền tối đa trong điều kiện lý tưởng | 10 km |
| Tốc độ bit | Lập trình có thể đạt đến 300 kbps |
| Hỗ trợ chế độ FSK, GFSK, MSK, GMSK, LoRaTM và OOK Modulation Mode | |
| Nhiệt độ làm việc | -40 oC đến 85 oC |
| Kích cỡ | 17x16 mm |

Bảng 3‑2  *Sơ đồ kết nối chân ESP8266 NodeMCU và mô-đun LoRa tại gateway.*

|  |  |
| --- | --- |
| **Chân ESP8266 Node Mcu** | **Chân LoRa** |
| Chân 3.3V | Chân nguồn 3,3 V |
| GND | Chân GND |
| D0 | DIO0 |
| D4 | RST |
| D7 | MOSI |
| D6 | MISO |
| D5 | SCK |
| D8 | NSS |

Phải sử dụng anten giúp tăng cường khoảng cách thu phát cho module. Anten 433Mhz 3dbi Ipex được sử dụng với các mạch thu phát sóng 433Mhz chuẩn kết nối ipex giúp khuếch đại khoảng cách và công suất thu phát với thông số gain 3dbi, anten có chất lượng tốt, độ bền cao, có vỏ bọc bảo vệ và kháng nhiễu, sử dụng hợp kim đồng chất lượng cao cho lõi anten, giúp tăng cường độ bền và ổn định.

* **Khối hiển thị**

Để giao tiếp LCD thì vi điều khiển cần tối thiểu 6 chân là RS, EN, D7 – D4. Điều này sẽ gây khó khăn cho ứng dụng vì nó tốn chân của vi điều khiển. Trong khi đó, nếu sử dụng module chuyển đổi I2C thì sẽ tiết kiệm được rất nhiều chân. Module cần 2 chân cấp nguồn và 2 chân giao tiếp truyền nhận dữ liệu SCL và SDA. Ngoài ra có thể điều chỉnh được độ tương phản bởi biến trở gắn trên module thay vì gắn thêm biến trở ngoài.

Chúng tôi chọn LCD 20×04 để hiện thị, LCD có có 4 hàng, mỗi hàng có 20 kí tự. Để điều khiển LCD có các IC chuyên dùng được tích hợp bên dưới LCD có mã số 447801 đến các IC 447809. Trong IC có bộ nhớ RAM dùng để lưu trữ dữ liệu cần hiển thị và thực hiện việc điều khiển LCD hiển thị.

Từ những phân tích trên, việc sử dụng module I2C cho LCD là điều cần thiết vì nó tiết kiệm chân điều khiển trên vi điều khiển, mở rộng khả năng điều khiển hoặc giao tiếp với các thiết bị khác.

*Thông số kĩ thuật module I2C:*

* Điện áp hoạt động: 5VDC
* Hỗ trợ màn hình: LCD1602, 1604, 2004 (Driver HD44780)
* Địa chỉ mặc định: 0X27 (có thể thay đổi địa chỉ bằng ba chân A0/A1/A2)
* Kích thước: 98mm x 60mm x 13.5mm
* **Khối gửi và nhận dữ liệu từ server**

Là khâu trung gian đóng vai trò là điểm kết nối giữa nơi lưu trữ và khối điều khiển trung tâm. Tất cả dữ liệu chuyển lên server hoặc ngược lại sẽ đi qua khâu này.

Nơi đây quản lý thông tin di chuyển theo cả hai chiều, nó có thể bảo vệ dữ liệu khi di chuyển lên đám mây khỏi các rò rỉ, hạn chế bị xâm phạm bởi nguồn tấn công bên ngoài. Có rất nhiều vi xử lý phù hợp để dùng làm khâu trung gian và nó được tính hợp chung vào vi điều khiển như đã chọn ở trên là ESP8266 NodeMCU.

* **Khối báo động tại chổ**

Thiết bị báo động thường có hai loại là báo động tại chổ hoặc báo động và theo dõi qua website, điện thoại. Báo động tại chổ thường sử dụng các chuông điện, mạch còi báo động, phát ra âm thanh cảnh báo, đèn báo cháy…

Nhóm thực hiện chọn còi báo động 12V rất thông dụng trên thị trường và phù hợp với mô hình đề tài

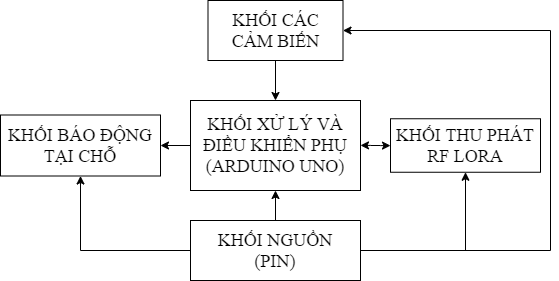
Hình 3‑6  *Hình ảnh còi báo động*

*Thông số kỹ thuật*

* Điện áp định mức: 12V. Điện áp làm việc 3-24V
* Dòng điện định mức: 12mA tại 12VDC
* Âm thanh kêu liên tục vớit ần số cộng hưởng: 3,0 ± 0,5 kHz.
* Nhiệt độ hoạt động: -20 ~ + 45 ° C.

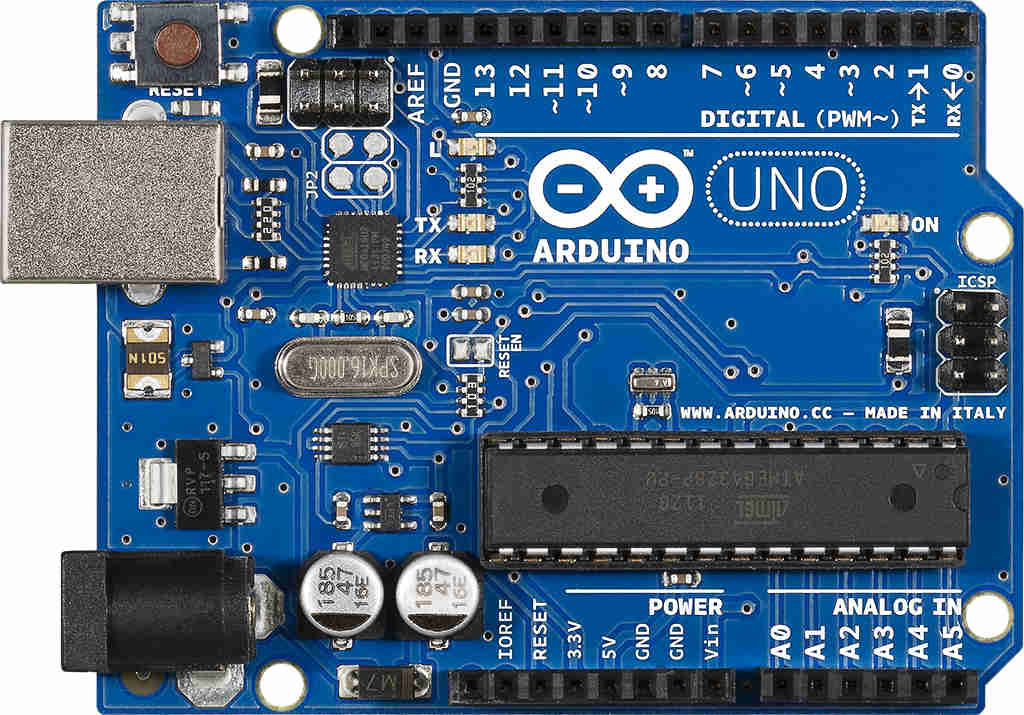
#### Tại khối cảm biến 1, 2 – Node 1, 2

Khối cảm biến 1, 2 có chức năng hoạt động giống nhau nhưng các khu vực giám sát được đặt ở những nơi khác nhau.



Hình 3‑7  *Sơ đồ khối tại các Node*

Theo yêu cầu đặt ra ban đầu của hệ thống, các vị trí này đảm nhận các chức năng giao tiếp có dây với các cảm biến, không dây với gateway bằng công nghệ LoRa, các yêu cầu trên khá cơ bản, không đòi hỏi bộ xử lý mạnh nên chúng tôi sử dụng mô-đun Arduino UNO R3, với ưu điểm là giá thành rẻ, phổ biển.



Hình 3‑8  *Mô-đun Arduino UNO R3*

**Thông số kỹ thuật**

* Vi điều khiển: ATmega328p (họ 8bit)
* Điện áp cấp: 7-12 VDC, điện áp hoạt động: 5V DC
* Tần số hoạt động: 16 MHz
* Dòng tiêu thụ: 45 – 80 mA. Dòng tối đa trên mỗi chân I/O: 20 mA
* Số chân Digital I/O: 14 (6 chân PWM)
* Số chân Analog: 6 (độ phân giải 10bit)
* Dòng ra tối đa (5V): 500 mA. Dòng ra tối đa (3.3V): 50 mA
* Bộ nhớ flash: 32 KB với 0.5KB dùng bởi bootloader
* SRAM: 2 KB, EEPROM: 1 KB
* **Khối thu phát LoRa**

Nhằm đảm bảo tính đồng bộ nên chúng tôi sử dụng chung module Lora Ra-02 ở các node. Việc giao tiếp các chân Lora và Arduino được thể hiện khác.

Bảng 3‑3 *Sơ đồ kết nối chân ESP8266 NodeMCU và mô-đun LoRa tại gateway.*

|  |  |
| --- | --- |
| **Chân Arduino Uno R3** | **Chân LoRa** |
| Cấp nguồn 3.3V cho Lora được thể hiện chi tiết ở phần sơ đồ nguyên lý | |
| GND | Chân GND |
| 8 | DIO0 |
| 9 | RST |
| 11 | MOSI |
| 12 | MISO |
| 13 | SCK |
| 10 | NSS |

#### Khối cảm biến bao gồm nhiều cảm biến thu thập các thông số, dữ liệu tại khu vực theo dõi.

**Cảm biến thu thập giá trị nhiệt độ - cảm biến DS18B20**

Đối với cảm biến đo nhiệt độ không khí, trên thị trường có rất nhiều loại, phổ biến nhất là DHT-11, DHT-22, DS18B20, LM35… Ở đề tài này chúng tôi quyết định chọn cảm biến DS18B20, tuy không vượt trội như DHT về mặt thông số (dải nhiệt độ hoạt động) và sai số nhưng đổi lại là chỉ dùng một dây giao tiếp, số lượng cảm biến trên một dây có thể mở rộng để tăng phạm vi theo dõi, đáp ứng được mục tiêu của đề tài nên không yêu cầu quá cao về độ chính xác.

**Cảm biến thu thập giá trị khí CO, khói – cảm biến MQ7**

Để phát hiện khói sinh ra trong đám cháy, chúng tôi sử dụng loại cảm biến khói, khí CO phổ biến trên thị trường cùng với IC chuyển đổi so sánh LM393 với giá thành rẻ và dễ tìm kiếm, độ chính xác tương đối, phù hợp với mục đích nghiên cứu của đề tài.

Chúng tôi chọn module cảm biến MQ-7 để phát hiện khói, khí CO từ những đám cháy. Là cảm biến bán dẫn giá rẻ có khả năng phát hiện khí CO có nồng độ từ 20 đến 2000 ppm (parts per million). Cảm biến được làm từ SnO2, có độ dẫn điện thấp trong không khí sạch.

Cảm biến có thể hoạt động được ở nhiệt độ từ: -10 ᴼC đến 50 ᴼC và tiêu thụ dòng khoảng 150 mA tại 5 V. Tuổi thọ cao, chi phí thấp.

Hình 3‑9 *Cảm biến MQ-7 ngoài thực tế.*

*Thông số kỹ thuật*

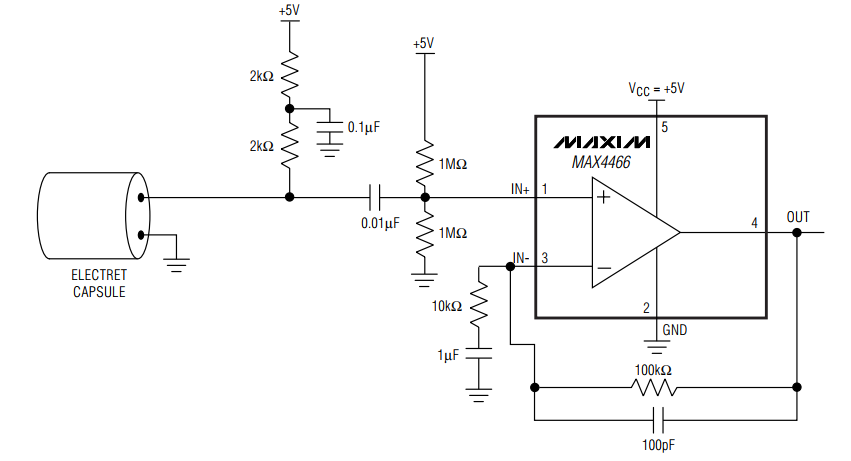
* Điện áp hoạt động: 3.3V-5V.
* IC so sánh: LM393.
* AO: Đầu ra Analog.

Bảng 3‑4 *Điểm làm việc của MQ-7*

|  |  |
| --- | --- |
| **Voltage(V)** | **Ampere(mA)** |
| 5 | 120 |
| 4 | 95 |
| 3 | 75 |
| 2.5 | 65 |

**Phát hiện âm thanh máy cưa gỗ**

Để phát hiện âm thanh máy cưa gỗ, chúng tôi chọn cảm biến âm thanh Microphone Amplifier Max4466. Do là đề tài đồ án nên chúng tôi chọn cảm biến Max446 với giá thành tương đối rẻ và chỉ nghe âm thanh trong khoảng 20cm. Còn đối với việc ứng dụng mô hình thực tế cần chọn những cảm biến khác thu thập âm thanh xa hơn.

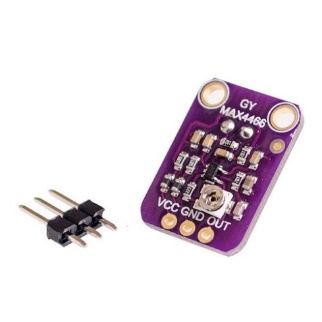


Hình 3‑10  *Sơ đồ nguyên lý của cảm biến âm thanh Max4466*

Các âm thanh lớn khi qua mạch sẽ tự động giảm tới mức ấn định, tương tự các âm thanh nhỏ hoặc xa sẽ được tự động tăng lên tới mức ấn định mà không cần căn chỉnh. Đối với việc khuếch đại, sử dụng Maxim Max4466, một op-amp được thiết kế đặc biệt cho việc này.

Để có hiệu suất tốt nhất, nên sử dụng nguồn cung cấp ổn định là nguồn cung cấp 3.3V. Dạng sóng âm thanh sẽ xuất hiện từ chân OUT, các tín hiệu ra sẽ có một điện áp khác nhau.

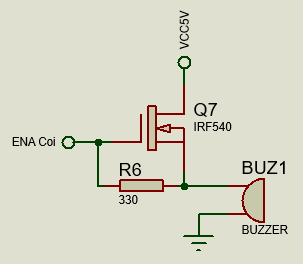
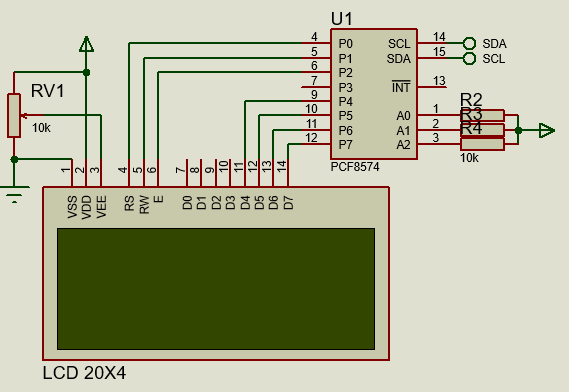
**Thông số kỹ thuật:**

* Điện áp hoạt động: 2.4V-5.5V
* Dãi tần số phát hiện: 20Hz – 20 KHz
* Tự động điều chỉnh để đạt đủ cường độ, hoặc lựa chọn max từ 40dB, 50dB, 60dB qua việc chỉnh biến trở.

Hình 3‑11 *Ảnh thực tế Max4466*

### Sơ đồ nối dây của các khối

#### Tại khối điều khiển trung tâm - Gateway:

Hình 3‑12 *Sơ đồ nguyên lý của khối Gateway kết nối Lora*

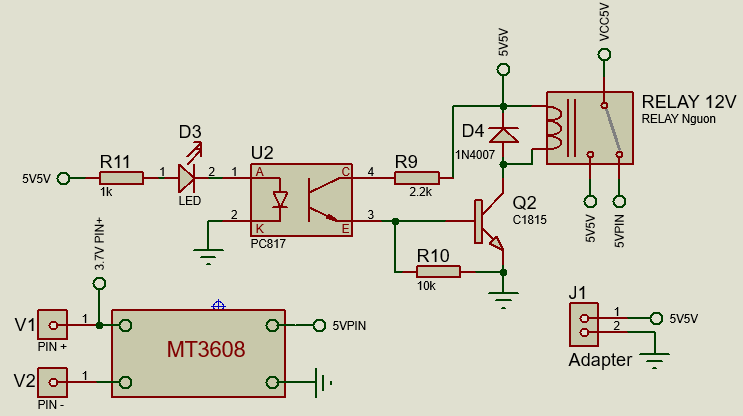
Hình 3‑13 ***Sơ đồ nguyên lý của khối Gateway kết nối LCD, Còi báo***

Vấn đề của chúng tôi khi thiết kế tại gateway là số chân của NodeMcu bị giới hạn, nên chúng tôi phải cân nhắc kĩ và sử dụng các chân phù hợp. Kết nối với Lora tốn 6 chân, LCD 2 chân và còn 1 chân của NodeMcu được dùng để điều khiển mở còi báo động.

Việc điều khiển mở còi báo, do sử dụng còi báo 12V nên chúng tôi sử dụng Fet IRF540 để cho phép mở điện áp từ cực D xuống cực S. Mắc trở 330Ω để 1 phần bù điện áp suy hao từ cực D xuống cực S.

**Khối mạch chọn nguồn**

Chúng tôi sử dụng hai nguồn cấp cho mạch hoạt động, thứ nhất sẽ cấp nguồn thông qua Adapter, thứ hai khi xảy ra tình trạng mất điện tạm thời sẽ cấp nguồn Pin. Việc thực hiện chọn nguồn cấp được thực hiện tự động thông qua khối mạch relay.



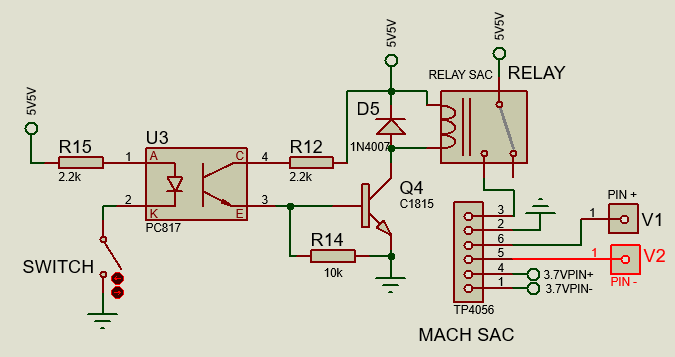
Hình 3‑14 *Khối mạch Relay chọn nguồn*

**Mạch tăng áp MT3608**

Do mạch hoạt động với điện áp cấp 5V DC, nên phải cần thêm mạch tăng áp khi sử dụng nguồn Pin 3.7V

Module tăng áp với điện áp đầu vào từ 2VDC đến 24VDC có thể điều chỉnh điện áp đầu ra từ 5VDC đến 28VDC và dòng tải tối đa là 2A. Mạch hoạt động với hiệu suất chuyển đổi lên đến 93% giúp cho điện áp đầu ra được ổn định.

**Khối mạch sạc:**



Hình 3‑15 *Khối mạch Relay sạc Pin*

Về cơ bản hoạt động khối mạch relay để sạc Pin tương tự như khối mạch chọn nguồn, nhưng việc thực hiện kích này tác động bằng tay qua công tắc. Khi bật công tắc sẽ điều khiển kích, đóng relay và cấp sạc Pin.

**Module sạc Pin TP4056**

Quá dòng, quá áp là nguyên nhân phổ biến khiến thiết bị hư hỏng và không thể khắc phục. Nên chúng tôi sử dụng mạch sạc Pin TP4056 sẽ giúp hạn chế hiện tượng quá dòng, quá áp, bảo vệ, tăng cao tuổi thọ của pin.

Kết nối Pin vào 2 cực Bat của module sạc Pin TP4056 1A – có IC bảo vệ quá dòng - quá áp.



Hình 3‑16 *Mạch sạc Pin TP4056*

* Sạc Pin 3.7V: Pin Ultrafire, Pin Lion 3.7V, Pin Lithium.
* Chức năng bảo vệ xả tránh cạn pin, có chế độ tự ngắt khi sạc đầy.
* Được tích hợp đèn LED báo đang sạc và đã sạc đầy.

Thông số:

* Dòng sạc tối đa: 1000mA.
* Điện áp vào: 5 VDC. Điện áp sạc đầy: 4.2VDC +/-1%.
* Chống quá áp của pin: 2.5V. Bảo vệ quá dòng pin: 3A.
* Giao tiếp đầu vào: Mini USB hoặc chân Vin +/-.
* Kích thước: 2.6x1.7 mm.

Để đọc điện áp Pin và hiển thị về thời gian sống của Pin lên LCD, thực hiện kết nối chân Pin+ 3.7V với chân A0 của NodeMcu.

#### Hoạt động của khối chọn nguồn tại gateway

Khối nguồn dự phòng sẽ cấp nguồn cho vi điều khiển hoạt động trong một khoảng thời gian, tránh trường hợp không thể báo động cháy, trộm gỗ khi có sự cố mất điện. Ta cần thiết kế một cơ chế tự thay đổi nguồn điện chính và nguồn dự phòng bằng cách sử dụng một relay để chọn nguồn. Hai đầu cuộn dây relay sẽ nối với 2 cực của nguồn điện chính, tiếp điểm thường đóng của relay sẽ nối với cực dương của nguồn dự phòng, tiếp điểm thường hở nối với cực dương của nguồn điện chính và đầu còn lại của relay sẽ nối với ngõ vào điện áp của mạch cần sử dụng nguồn dự phòng khi mất điện. Cực âm của 2 nguồn và mạch đều được nối chung.

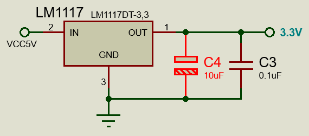
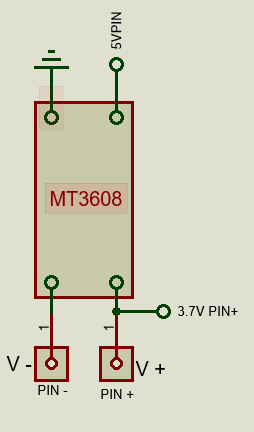
Khi có nguồn cấp là adapter, dòng điện qua trở, qua led sẽ giảm dòng xuống thấp. Dòng nhỏ này đi qua 2 đầu của led có trong opto làm cho led phát sáng, làm thông 2 cực của photo diot, mở cho dòng điện chạy qua. Đồng thời, điện áp bên cuộn dây của relay qua trở 2.2k đến cực B của transistor C1815 và điều khiển mở nó. Diode 1N4007 phân cực tạo từ trường hút rờ-le chọn chân và chân này sẽ được cấp là nguồn adapter. Khi mất điện, mạch relay không được kích, nên rờ-le sẽ ở vị trí ban đầu và chân này sẽ cấp nguồn là Pin.

#### Tại khối cảm biến 1, 2 – Node 1, 2

Hình 3‑17 *Sơ đồ nguyên lý tại Node cảm biến kết nối Lora*

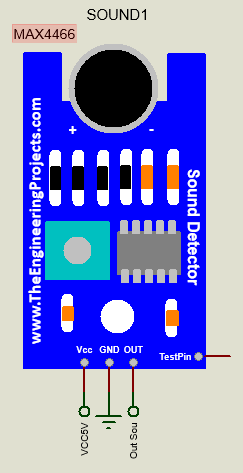
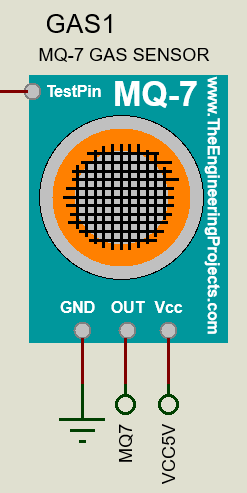
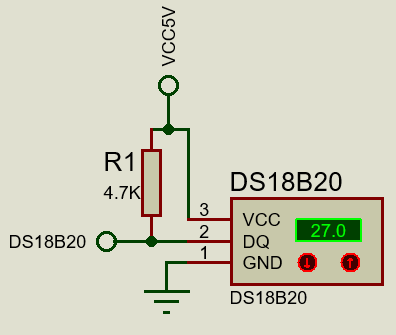
Sơ đồ nguyên lý về kết nối của Lora, còi báo tương tự như tại Gateway, nhưng tại Node chỉ sử dụng năng lượng cấp là Pin nên việc thực hiện kết nối cấp nguồn là khác.

Do dòng ra ở chân nguồn 3.3V trên Arduino (tối đa 50mA) không đủ dòng cấp cho Lora (100mA) khi gửi dữ liệu, nên cần cấp từ mạch nguồn khác. Và chúng tôi sử dụng IC ổn áp LM1117 3.3V, kết hợp với tụ điện để tạo điện áp chuẩn cấp cho Lora.



Hình 3-18 Sơ đồ nguyên lý tại node kết nối nguồn cấp

Vì điện áp tối đa của pin li-ion chỉ 4,2V nên ta phải sử dụng bộ nâng áp lên 5V để cung cấp cho mạch. Chúng tôi dùng mạch tăng áp MT3608 để nâng đủ điện áp cấp. Đọc giá trị điện của Pin sử dụng FET IRF540 để mở áp từ Pin vào chân analog của Arduino, thực hiện kích để mở FET với chân Digital của Arduino.



Hình 3.19 Sơ đồ nguyên lý tại Node kết nối các cmar biến

Và các cảm biến DS18B20, MQ-7, Max4466 thực hiện kết nối chi tiết như sơ đồ nguyên lý như trên. Còn còi báo, chỉ sử dụng cho cảnh báo trộm gỗ.

### Tính toán điện áp, dòng điện và thiết kế khối nguồn

Công việc chính của trạm cảm biến thường là gửi và nhận dữ liệu từ xa. Trong quá trình hoạt động, việc gửi và nhận dữ liệu là một trong những nhiệm vụ tốn kém năng lượng nhất. Việc tính toán và chọn vi điều khiển, các cảm biến, các linh kiện để tiết kiệm năng lượng là điều cần thiết.

Hệ thống cần tối ưu năng lượng, nên chúng tôi tính toán chọn điện áp và dòng điện để cấp nguồn cho hệ thống. Dựa vào dòng tiêu thụ của từng module cảm biến, khối cảm biến và khối xử lý trung tâm việc tính toán lựa chọn khối nguồn như sau:

Bảng 3‑5 *Bảng thống kê số lượng và dòng tiêu thụ của linh kiện tại Gateway*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Linh kiện** | **Cấp điện áp** | **Dòng tiêu thụ** |
| 1 | Kit ESP8266 NodeMcu | 3.3V | 70 - 400 mA |
| 2 | Module LoRa SX1278 433MHz Ra-02 | 3.3V | 10 – 100 mA |
| 3 | Module LCD 20x4 và I2C | 5V | 80 mA |
| 4 | Còi báo động | 5V | 12 mA |
| 5 | Khối mạch chọn nguồn | 5V | 80 mA |
| 6 | Khối mạch sạc Pin | 5V | 80 mA |
| 7 | Mạch sạc pin TP4056 |  | 80mA |
|  | Tổng (tính dòng tiêu thụ cao nhất) |  | I = 832 mA |

Bảng 3‑6 *Bảng thống kê số lượng và dòng tiêu thụ của linh kiện tại Gateway*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Linh kiện** | **Cấp điện áp** | **Dòng tiêu thụ** |
| 1 | Kit Arduino Uno R3 | 5V | 40 – 80 mA |
| 2 | Module LoRa SX1278 433MHz Ra-02 | 3.3V | 40 - 100mA |
| 3 | Module cảm biến MQ-7 | 3.3V | 80 mA |
| 4 | Cảm biến DS18B20 | 5V | 1.5 mA |
| 5 | Cảm biến Max4466 | 5V | < 20mA |
| 6 | Còi báo động | 5V | < 12mA |
|  | Tổng (tính dòng tiêu thụ cao nhất) |  | I = 284 mA |

**Khối nguồn tại Gateway:** Tại đây cần cấp nguồn liên tục để hoạt động, do khi thực hiện truyền và nhận dữ liệu, kết nối wifi NodeMCU tiêu thụ dòng là cao. Và tổng dòng tiêu thụ dựa trên lý thuyết là 832mA, nên chúng tôi sẽ chọn nguồn cấp là adapter chuyển đổi áp từ 220VAC sang 5VDC dòng 2A để cung cấp mạch.

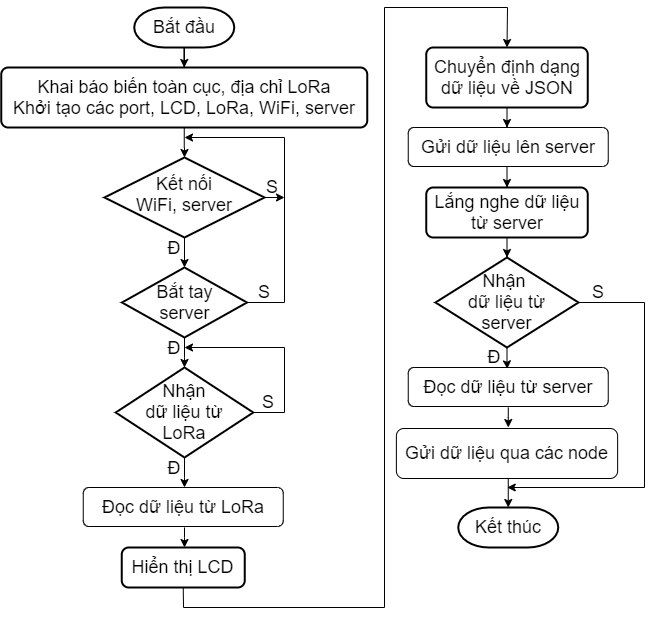
Khi mất điện chúng tôi chỉ cảnh báo qua còi và hiển thị dữ liệu lên LCD. Thực hiện Sleep Lora và NodeMcu cho việc sử dụng năng lượng thấp phù hợp với Pin. Dòng tiêu thụ khi dùng là khoảng 200mA bỏ qua tiêu thụ của khối mạch chọn nguồn và khối mạch sạc. Sử dụng hai pin li-ion 3.7V 3000mA ghép song song cung cấp năng lượng cho mạch.

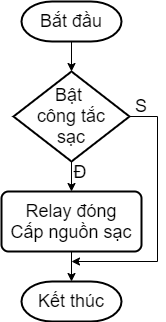
**Về khối nguồn tại Node:** chúng tôi sẽ sử dụng Pin để cung cấp nguồn điện cho các Node cảm biến, và tại đây công suất tiêu thụ năng lượng thấp, đặt trong rừng, cách xa nơi cấp điện. Và được thể hiện chi tiết ở phần *Quản lý năng lượng cho hệ thống*.

*Thông số kỹ thuật của Pin 3.7V 3000mA*

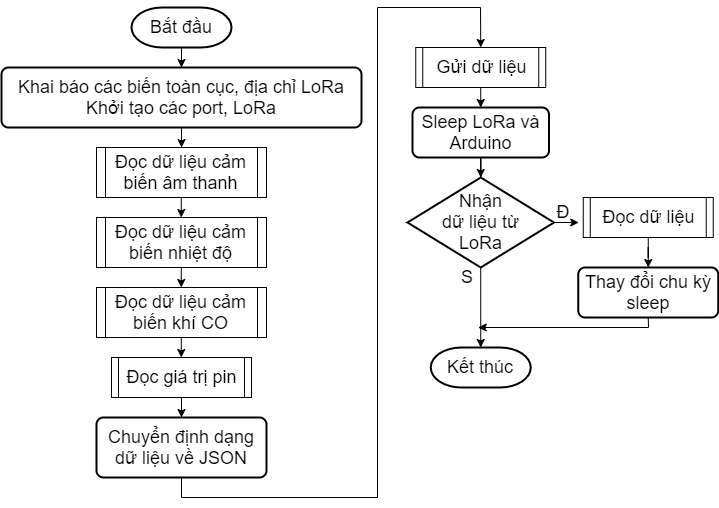
* Điện áp: 3.7V, tự ngắt nếu điện áp pin dưới 2.75V.
* Dung lượng: 3000mA và pin có thể sạc 500 lần.

### Kiến trúc phần mềm thiết bị

* **Lưu đồ chương trình chính tại Gateway**

Hình 3‑20  *Lưu đồ thuật toán khối Gateway*

Hình 3‑21 *Lưu đồ thuật toán khối chọn nguồn*

* **Lưu đồ chương trình tại Node**

Hình 3‑19  *Lưu đồ thuật toán khối Node*

* **Nguyên lí hoạt động toàn hệ thống**

Tại gateway, vi điều khiển có nhiệm vụ quan trọng là nhận các thông số dữ liệu từ node, sau đó cập nhật lên cơ sở dữ liệu Monggo DB, hiển thị ra giao diện người dùng. Ngoài ra, nó còn có nhiệm vụ khác là nhận tín hiệu điều khiển từ web hoặc ứng dụng di động để gửi xuống lại cho khối Node thực hiện.

Khi cấp nguồn vào hệ thống, tại gateway khởi động NodeMcu, Module Lora, LCD…; tại node khởi động Arduino, cảm biến DS18B20, MQ-7, Max4466... Mặc định ban đầu hệ thống sẽ nhận dữ liệu với chu kỳ gửi từ node là 5 phút và được thay đổi chu kỳ thông qua nhận lệnh từ người dùng.

Cảm biến khí MQ-7 đặt ở các khu vực 1, 2 liên tục thu thập giá trị nồng độ khí CO; cảm biến nhiệt độ DS18B20 thu thập thông số về nhiệt độ từ môi trường; MAX4466 lắng nghe âm thanh thường xuyên. Âm thanh nghe được so sánh với ngưỡng cho phép do người dùng đặt, nếu quá ngưỡng sẽ cảnh báo qua còi báo. Bên cạnh đó, những số liệu thu được đó sẽ được gửi lên server qua module ESP8266 MCU và vẽ thành biểu đồ dễ dàng theo dõi.

Khi một trong các giá trị cảm biến quá ngưỡng đã thiết đặt ban đầu, hệ thống sẽ phát cảnh báo tại gateway, đồng thời server gửi tín hiệu đến các thiết bị đang giữ kết nối để thông báo về hiện tượng bất thường đang xảy ra tại khu vực theo dõi nào.

Khi cấp nguồn vào hệ thống, khởi động Arduino, mô-đun LoRa, các mô-đun cảm biến… Hệ thống hoạt động bao gồm các chức năng hiển thị các thông số của môi trường như nhiệt độ, nồng độ CO, âm thanh, tiếng động môi trường trên LCD, website và ứng dụng điện thoại.

Tại website và điện thoại, thực hiện điều khiển với chức năng thay đổi chu kỳ đọc các giá trị cảm biến, pin tại các trạm thu thập. Điều này phù hợp cho các khu vực có khả năng thường xuyên xảy ra cháy hay khu vực có thể xảy ra tình trạng trộm gỗ... Tính năng này giúp theo dõi tình hình tại các khu vực chính xác hơn và xử lý nhanh hơn khi có sự cố xảy ra.

* **Giải thích lưu đồ khối node**

Chương trình bắt đầu sẽ vào khởi tạo các chân chuẩn SPI, khởi tạo LoRa, khởi tạo các biến, khởi tạo cảm biến và kết nối LoRa với các thông số băng tầng đã thiết lập. Ban đầu các node sẽ ở trạng thái gửi gói tin, sau khi đọc các giá trị các cảm biến, pin, node gửi gói tin qua Lora. Tại gateway sau khi khởi tạo sẽ lắng nghe các gói tin từ node, nếu đúng gói tin chương trình sẽ đọc và mã hóa gói tin gửi lên server.

* Nhận gói tin Lora: nếu có gói tin gửi đến sẽ tiến hành đọc địa chỉ của gói tin. Nếu đúng địa chỉ của thiết bị gửi đến thì sẽ tiến hành đọc hết gói tin ngược lại sẽ bỏ qua gói tin đó.
* Truyền gói tin Lora: Chương trình sẽ bắt đầu mã hóa thành chuỗi Json, ghi gói tin, ghi dữ liệu và đóng gói. Kết thúc gói tin sẽ được gửi đi.

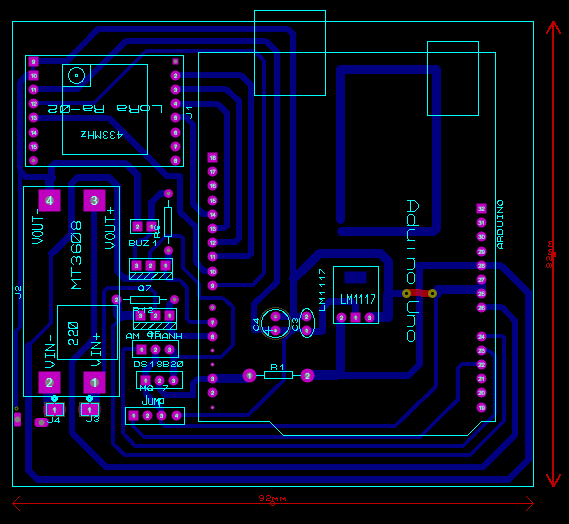
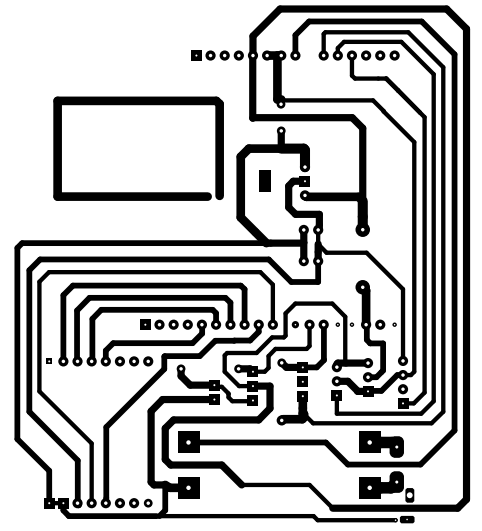
## thi công hệ thống

### Thi công bo mạch

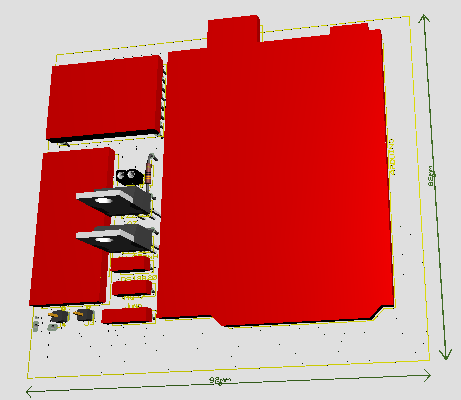
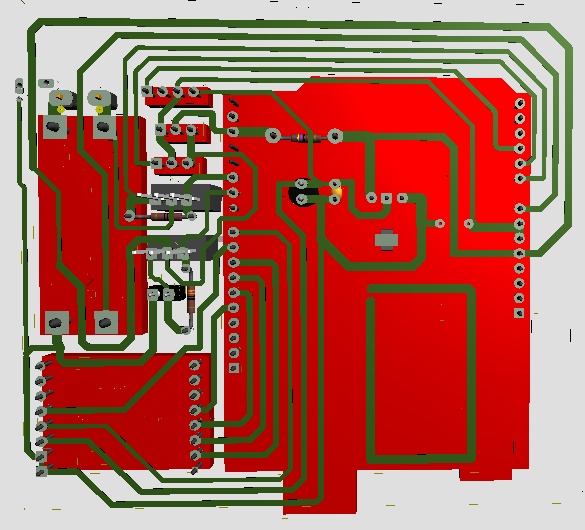
Dựa trên cơ sở lý thuyết từ chương 2 và tính toán, thiết kế, sơ đồ nguyên lý của hệ thống ở chương 3, nhóm đã tiến hành thiết kế và vẽ mạch PCB một lớp thủ công của các thành phần trên. Thực hiện thi công bo mạch tại node 1, 2 và tại gateway được thể hiện bên dưới.

#### Node cảm biến 1, 2

Thực hiện vẽ mạch in của mạch trên phần mềm proteus.

****

**Hình 3‑20 S***ơ đồ đi dây và mạch in của node*

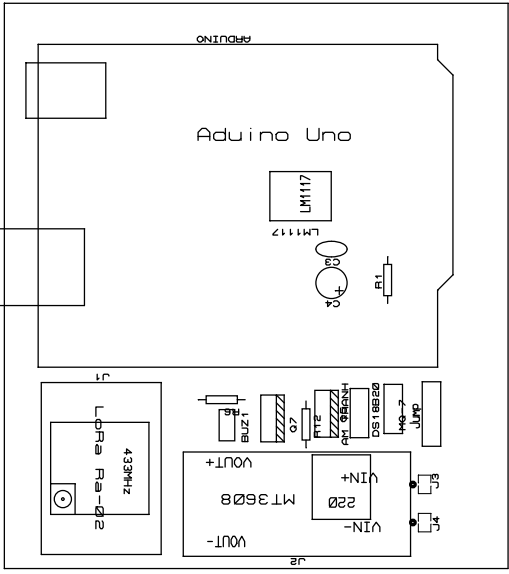
****Bên dưới là sơ đồ mạch in ở chế độ 3D. Ở chế độ này có thể quan sát được mặt trước và sau của mạch in một cách gần giống thực tế. Từ đó có thể điều chỉnh cách bố trí các linh kiện cho hợp lý.

**Hình 3‑21** *Sơ đồ 3D mặt trước và sau của mạch tại node*

Thực tế tại hai node được thiết kế là giống nhau, nên danh sách các linh kiện sử dụng của mạch tại hai node cũng giống nhau. Dưới đây là danh sách linh kiện cho một node liệt kê ở bảng 3-7 và cách bố trí linh kiện trên bo mạchở hình 3.22.

Bảng 3‑7 *Linh kiện được sử dụng tại một node*

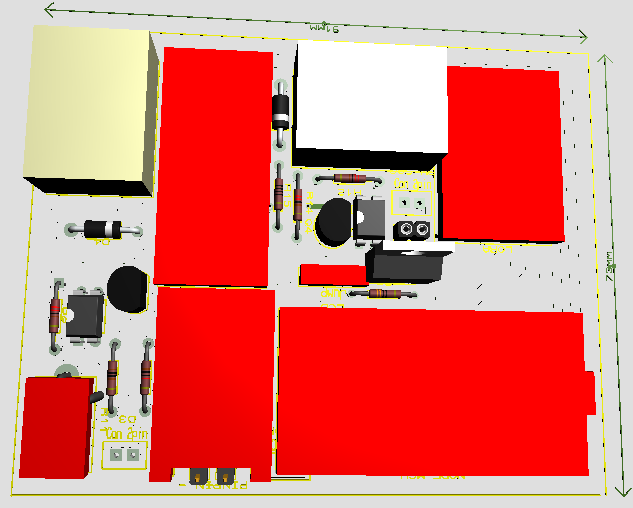
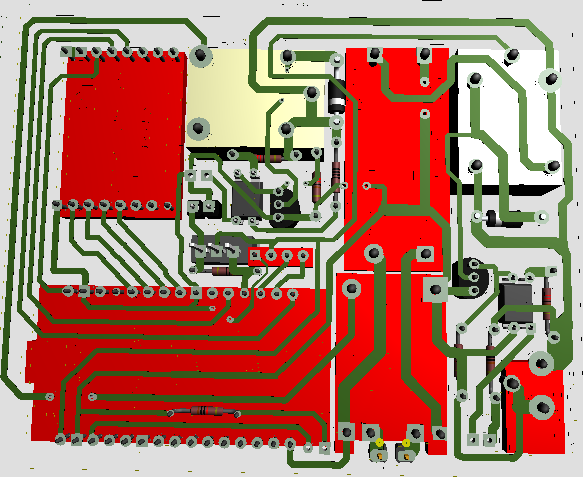
|  |  |  |  |
| --- | --- | --- | --- |
| STT | TÊN LINH KIỆN | CHIP CHÍNH | SỐ LƯỢNG |
| 1 | Kit Arduino Uno R3 | ATmega-328 | 1 |
| 2 | Module LoRa Ra-02 | SX1278 | 1 |
| 3 | Module cảm biến khí CO MQ-7 | LM393 | 1 |
| 4 | Cảm biến nhiệt độ | DS18B20 | 1 |
| 5 | Mạch cảm biến âm thanh | MAX4466 | 1 |
| 6 | Mạch tăng áp mini 2A | MT3608 | 1 |
| 7 | Còi báo động | - | 1 |
| 8 | FET IRF530 | - | 2 |
| 9 | AMS1117 - 3.3V | - | 1 |
| 10 | Ăng ten 433Mhz 3dBi | - | 1 |
| 11 | Pin Sạc 18650 3000mAh 3.7V | - | 2 |
| 12 | Điện trở, tụ điện | - | - |
| 13 | Ăng ten 433Mhz 3dBi | - | 1 |

******

***Hình 3.22 Sơ đồ bố trí linh kiện trên mạch ở node***

#### Gateway

**Hình 3‑20 S***ơ đồ đi dây và mạch in của gateway*

******Giống như tại node, tại gateway sơ đồ mạch in ở chế độ 3D được thể hiện như hình 3.24.

**Hình 3‑23** *Sơ đồ 3D mặt trước và sau của mạch tại gateway*

Danh sách các linh kiện sử dụng của mạch gateway được liệt kê ở Bảng 3-8.

Bảng 3‑8 *Danh sách linh kiện sử dụng cho gateway*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TÊN LINH KIỆN** | **Chíp chính** | **SỐ LƯỢNG** |
| 1 | Kit ESP8266 NodeMCU | ESP8266 12E | 1 |
| 2 | Module LoRa Ra-02 | SX1278 | 1 |
| 3 | Ăng ten 433Mhz 3dBi | - | 1 |
| 4 | Mạch tăng áp mini | MT3608 | 1 |
| 5 | Mạch sạc Pin Lihtium | TP4056 | 1 |
| 6 | FET IRF530 | - | 1 |
| 7 | Relay 5VDC | - | 2 |
| 8 | Opto PC817 | - | 2 |
| 9 | Adapter | - | 1 |
| 10 | Transistor C1815, điện trở, diode, Led | - | - |

#### Lắp ráp và kiểm tra bo mạch

Phần này chúng tôi sẽ thực hiện lắp ráp và kiểm tra độ ổn định của các bo mạch bao gồm mạch của gateway, node 1, node 2 và lắp ráp Kit NodeMCU và Arduino với mô-đun LoRa. Quy trình lắp ráp – kiểm tra bo mạch gateway, node 1 và node 2 cơ bản gồm các bước sau:

**Bước 1:** Sau khi in, ủi và rửa thành bo mạch hoàn chỉnh, dùng khoan kĩ thuậttiến hành khoan các lỗ chân linh kiện trên bo mạch.

**Bước 2:** Kiểm tra sơ bộ mạch:

Kiểm tra sơ bộ mạch nhằm kiểm tra lỗi có thể xảy ra trong quá trình thi công mạch in: đứt dây tín hiệu, ngắn mạch... Việc này là bước quan trọng nhằm đánh giá mạch có thể sử dụng được hay không để tiến hành thay đổi sớm trước khi tiến hành thi công hàn linh kiện.

**Bước 3:** Lắp ráp mạch nguồn.

Sau bước kiểm tra sơ bộ, mạch không còn các lỗi thì tiến hành chuyển sang giai đoạn bắt đầu hàn linh kiện lên mạch.

Mạch nguồn tại gateway gồm có Jack DC và Adapter, còn tại các node dùng nguồn Pin cắm thông qua header. Dùng VOM đo xem nguồn cấp vào mạch có phải 5V chuẩn và nguồn ra của module có phải đúng 5V và 3.3V không.

Sau khi hàn xong, tiếp tục dùng VOM kiểm tra các chân linh kiện có bị chập nhau không trước khi cấp nguồn. Sau đó cấp nguồn vào khối, dùng VOM để đo điện áp đã đạt chưa. Nếu chưa đạt tiến hành kiểm tra lại, còn nếu đạt thì tiếp tục tiến hành lắp các khối tiếp theo.

**Bước 4:** Hàn các module vào mạch.

Các module cần hàn đã được liệt kê ở bảng bên trên.

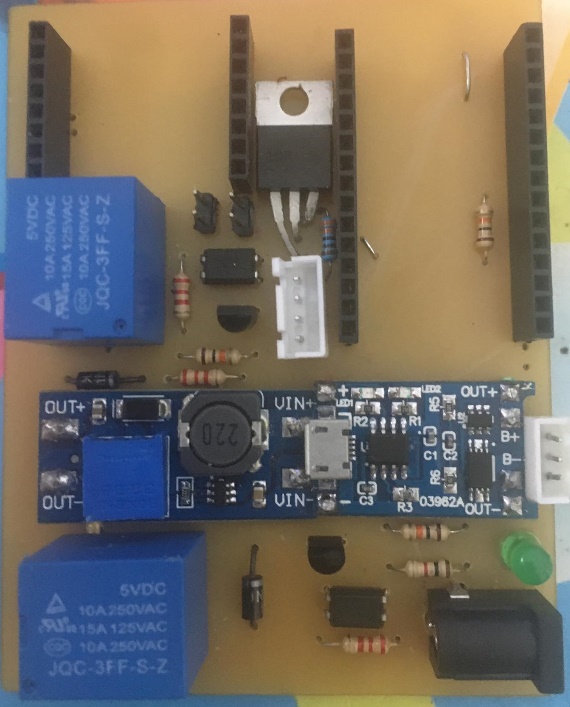
Lần lượt kiểm tra các linh kiện, sau đó tiến hành hàn lên mạch và kiểm tra như bước 2. Việc hàn cần kết hợp song song với việc kiểm tra lỗi để mạch của mô hình sau khi hoàn thành là một thiết bị hoàn chỉnh. Sau khi đặt hết linh kiện lên mạch tiến hành kiểm tra thêm 1 lần sau cùng cho chính xác.

Thực hiện theo các bước trên, được bo mạch hoàn chỉnh của node 1 và node 2 như Hình 3-25



**Hình 3.25**:Hình ảnh thực tế mạchnodesau khi hàn linh kiện

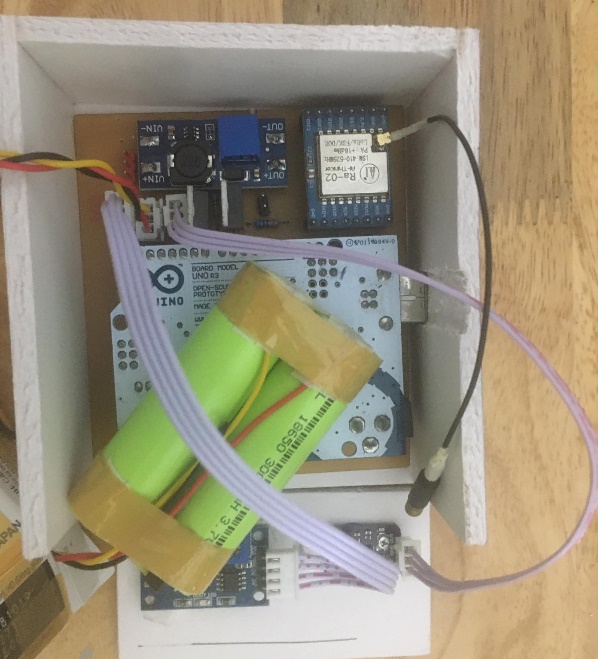
Giống như node ở gateway cũng thực hiện các bước trên được bo mạch như hình 3.26



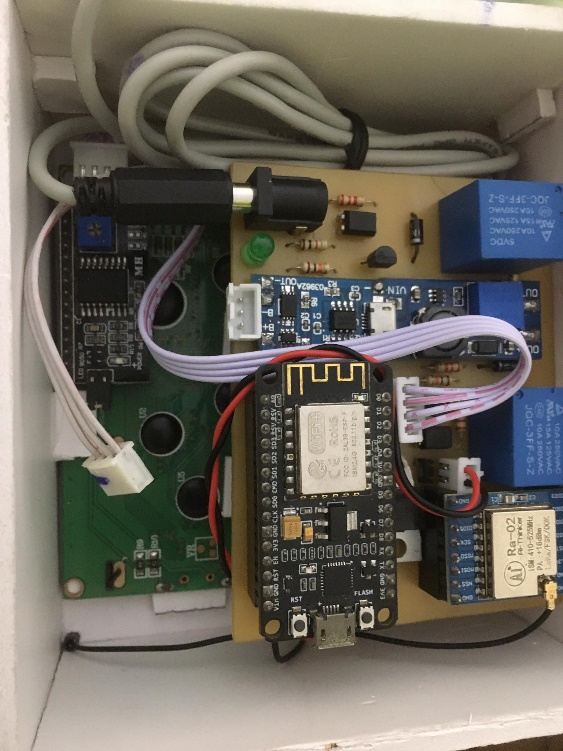
**Hình 3.26**:Hình ảnh thực tế mạch gateway sau khi hàn linh kiện

### Thi công, đóng gói mô hình

Sau khi hoàn tất việc lắp ráp và kiểm tra bo mạch, chúng tối tiến hành thi công mô hình hệ thống, các thành phần của mô hình được ghép từ các tấm bìa cứng độ dày 5mm. Cuối cùng, tiến hành đặt bo mạch, lắp các cảm biến, còi báo, nguồn của hệ thống vào mô hình đóng gói thành sản phẩm hoàn chỉnh như Hình 3.27 và Hình 3.28.



**Hình 4.8** Hình ảnh đóng gói mô hình của node



**Hình 4.9** Hình ảnh đóng gói mô hình của gateway

## Quản lý năng lượng cho hệ thống

Quản lí năng lượng trong các thiết bị IoT bắt đầu từ việc thiết kế phần cứng và điều này là quan trọng nhất. Do đó, chúng tôi phải tính toán phần cứng khi hoạt động sẽ tiêu tốn bao nhiêu năng lượng và các module, cảm biến, trở, led... sẽ ảnh hưởng tới mức tiêu thụ năng lượng ra sao. Dung lượng Pin cung cấp cho hệ thống và so sánh với mức tiêu thụ thực tế để tính toán thời gian hoạt động. Ngoài ra trạng thái hoạt động của hệ thống (tại node cảm biến) cũng ảnh hưởng rất lớn tới mức tiêu thụ năng lượng và thời gian sống của Pin.

Theo mô hình hệ thống thiết kế (tại node cảm biến), các mức tiêu thụ năng lượng tính theo DataSheet và theo phần *3.3.4 Tính toán điện áp, dòng điện* được liệt kê ở bảng 3.9: [15]

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Dòng tiêu thụ khi hoạt động** | **Dòng tiêu thụ khi Sleep** |
| Arduino Uno R3 | 40 – 80 mA | 35 mA |
| Lora Ra-02 | Khi gửi gói tin – 100 mA | 10 µA |
| Khi nhận gói tin – 40 mA |
| Khi chờ - 10 mA |
| DS18B20 | 1.5 mA | 5 µA |
| MQ-7 | 80 mA | - |
| Max4466 | 20 mA | - |
| Còi báo | 12 mA | - |
|  |  |  |
| Tổng |  |  |

Các cảm biến cần thu thập dữ liệu cách xa module xử lí (Arduino) và cần từ một đến nhiều cảm biến để thu thập thông tin môi trường. Mỗi cảm biến sẽ có mức tiêu thụ năng lượng cụ thể, vì vậy việc lựa chọn cảm biến ngoài độ tin cậy, còn cần phải chú ý đến mức tiêu thụ của cảm biến. Dữ liệu thu thập được sẽ được xử lí tại Node và gửi về gateway theo khoảng thời gian nhất định.

Do năng lượng cung cấp tại các node là liên tục và các node đặt ở nơi cách nơi cung cấp điện áp (điện 220V). Vì thế cần phải tính toán chi tiết và sát với tiêu thụ năng lượng thực tế tại các node.

Dựa trên sự tiêu thụ của các cảm biến, module mà nhà sản xuất cung cấp khi chúng hoạt động và quá trình đo thực tế. Chúng tôi tiến hành tính toán tiêu thụ của node trong từng giai đoạn. Quá trình tính toán và thu thập được thực hiện bằng cách chia code thành nhiều đoạn nhỏ và đo giá trị giảm áp trên điện trở 1 ohm. Được thực nghiệm với hai trường hợp: và đạt kết quả như sau:

Bảng ‑10: Kết quả thực nghiệm năng lượng tại node

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Quá trình | Chu kỳ - 30 giây | | Chu kỳ - 5 phút | | Chu kỳ - 10 phút | |
| Tiêu thụ (mA) | Thời gian | Tiêu thụ (mA) | Thời gian | Tiêu thụ (mA) | Thời gian |
| Sleep | 42 | 30 giây | 42 | 5 phút | 42 | 20 phút |
| Chờ | 126 | 0.5 giây | 126 | 0.5 giây | 126 | 0.5 giây |
| Lấy mẫu xử lý | 157 | 1.5 giây | 157 | 1.5 giây | 157 | 1.5 giây |
| Truyền | 210 | 10 ms | 210 | 10 ms | 210 | 10 ms |

Dựa theo công thức tính năng lượng:

|  |  |  |
| --- | --- | --- |
|  | Q = I.t (A.s) | (9) |

Thực hiện tính toán được bảng giá trị sau:

Bảng ‑: Năng lượng tiêu hao tại node trong từng giai đoạn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ký hiệu | Chu kỳ - 30 giây | Chu kỳ - 5 phút | Chu kỳ - 20 phút | Đơn vị |
| Sleep | a | 1.26 | 12.6 | 50.4 | C |
| Chờ | b | 0.063 | 0.063 | 0.063 | C |
| Lấy mẫu, xử lý | c | 0.24 | 0.24 | 0.24 | C |
| Truyền | d | 0,021 | 0.021 | 0.021 | C |

### Tính toán thời gian sống của pin

Chúng tôi chỉ tính thời gian sống của pin tại node.

**Dòng trung bình**

Tại node, có hai giai đoạn làm việc. Giai đoạn I, mạch tiến hành lấy mẫu và xử lý, số lượng mẫu tùy thuộc vào chu kỳ làm việc được thiết đặt, sau đó truyền dữ liệu về gateway. Giai đoạn II, mạch chuyển sang chế độ ngủ, nhằm tiết kiệm năng lượng tiêu thụ.

Áp dụng công thức (9) vào mô hình năng lượng để tìm :

|  |  |  |
| --- | --- | --- |
|  | = | (10) |

T: là thời gian cho một chu kỳ hoạt động

**Dung lượng nguồn Pin:** Trên thực tế, mạch cần sử dụng nguồn pin gồm 4 pin li-ion ghép song song. Mỗi pin có dung lượng 3000mA, điện áp 3.7V. Do đó:

Pnguồn = 4x3000x3.7 = 31080 mWh

**Năng lượng trung bình của hệ thống:** Với điện áp hoạt động 5V ta có:

|  |  |  |
| --- | --- | --- |
|  | Pavg = ( x 5) | (11) |

**Thời gian pin có thể cấp:**

|  |  |  |
| --- | --- | --- |
|  | TLife  = (giờ) | (12) |

Chúng tôi thu được kết quả như sau:

Bảng ‑: Dòng trung bình và thời gian sống của pin trong từng chu kỳ

|  |  |  |  |
| --- | --- | --- | --- |
|  | Chu kỳ - 30 giây | Chu kỳ - 5 phút | Chu kỳ - 20 phút |
|  | 49,5 mA | 42,9 mA | 50.7 mA |
| Pavg | 0.2475 Wh | 0.2145 Wh | 0.285 Wh |
| Thời gian pin có thể cấp | < 125.6 giờ  5 ngày | < 144.9 giờ  6 ngày | < 109.6 giờ  4.54 ngày |

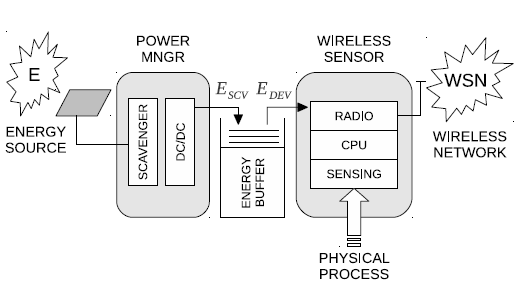
Do khi vào chế độ ngủ, mô đun Arduino Uno tiêu thụ dòng lớn, nên chu kỳ ngủ càng lớn thì thời gian sống của pin càng giảm xuống.

### Giải pháp khắc phục năng lượng cho hệ thống

**Đối với nguồn cấp**

Đối với những dự án IoT dùng Pin là cần thiết nhưng Pin cấp cho hệ thống sau một khoảng thời gian sẽ hết, và trong những hệ thống như thế cần tính toán và xử lí vấn đề tiết kiệm năng lượng. Và việc quản lí năng lượng này là một việc đầy thách thức vì các thiết bị phải luôn được cấp nguồn và có thể đặt được ở khắp mọi nơi.

Và những tấm pin năng lượng mặt trời, thu năng lượng từ môi trường, lưu trữ năng lượng trong energy buffer và cung cấp năng lượng thực hiện nhiệm vụ. [6]



Hình ‑: Giải pháp khắc phục năng lượng hệ thống **[23]**

Trong mô hình trên, năng lượng trung bình thu được từ môi trường phải lớn hơn năng lượng mà thiết bị yêu cầu trong suốt thời gian hoạt động. Tức là:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Trong đó:

: công suất tức thời thu được từ pin năng lượng mặt trời.

: công suất tiêu thụ năng lượng tức thời của thiết bị và được tính theo công thức:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

**Cải thiện Vi điều khiển**

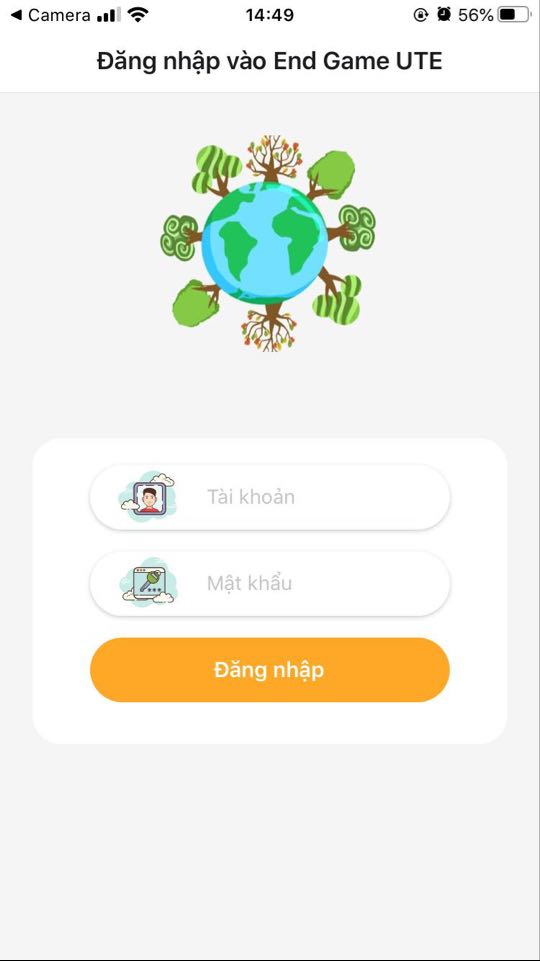
Quan trọng hơn, chúng tôi phải tính toán để khối Node cảm biến tiêu thụ năng lượng thấp nhất có thể. Và vấn đề của chúng tôi là ở khối Node, do sử dụng vi điều khiển chính là mô đun Arduino Uno, khi vào chế độ ngủ nó vẫn tiêu thụ dòng lớn (khoảng 35mA).

Trên mô đun Arduino Uno R3 tích hợp Led, USB UART để chuyển đổi Serial và nó vẫn hoạt động khi vào chế độ ngủ Để năng lượng tiêu thụ là thấp nhất khi vào chế độ ngủ, chúng tôi sẽ bỏ những thành phần không cần như USB UART, Led dòng sẽ giảm xuống rất lớn từ 40mA chỉ còn 14mA. [20]

Việc bỏ đi những thành phần như trên sẽ cắt đi những thành phần khác không tiêu thụ năng lượng. Như thế khi vào chế độ ngủ cho Arduino chỉ còn tiêu thụ dòng khoảng 5 – 10 mA.

## Thiết kế phần mềm

### Thiết kế ứng dụng di động

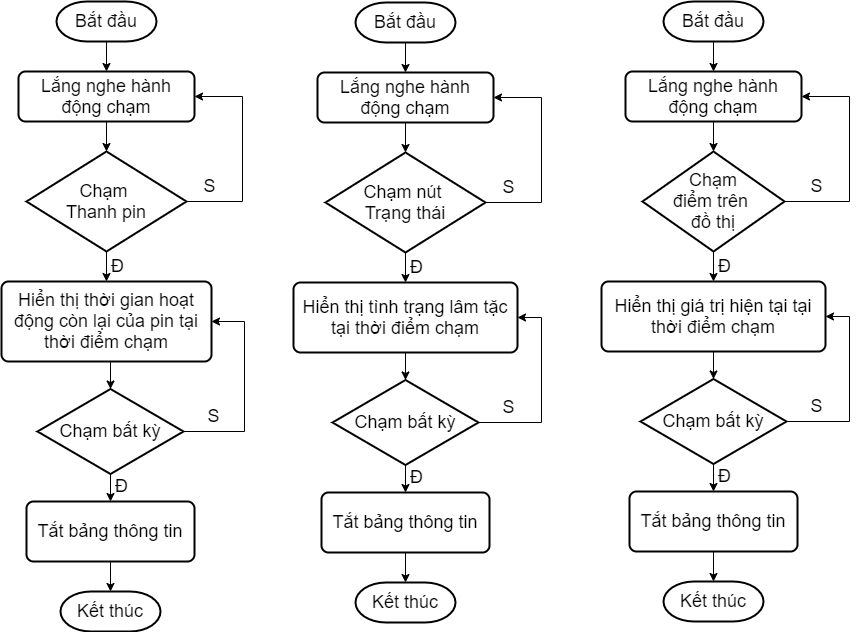
Như mục tiêu đã đề ra, một ứng dụng quản lí trên điện thoại thông minh đóng vai trò cho chức năng hiển thị của hệ thống. Chúng tôi đã tìm hiểu và lập trình một ứng dụng có thể chạy được trên cả Android và iOS.

Hình 3‑29 *Giao diện đăng nhập của ứng dụng*

Lưu đồ đăng nhập của ứng dụng

Hình 3‑30 *Lưu đồ đăng nhập*

Ứng dụng lắng nghe dữ liệu từ máy chủ từ máy chủ, xử lý và hiển thị dạng biểu đồ cho thuận tiện quan sát và phân tích. Trong ứng dụng, người dùng có thể theo dõi được tình trạng pin còn lại của các node đặt ngoài rừng cũng như đưa ra quyết định sạc pin hay không và theo dõi được tình hình lâm tặc tại thời điểm.

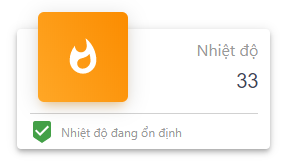
Hình 3‑31 *Lưu đồ lắng nghe sự kiện của ứng dụng*

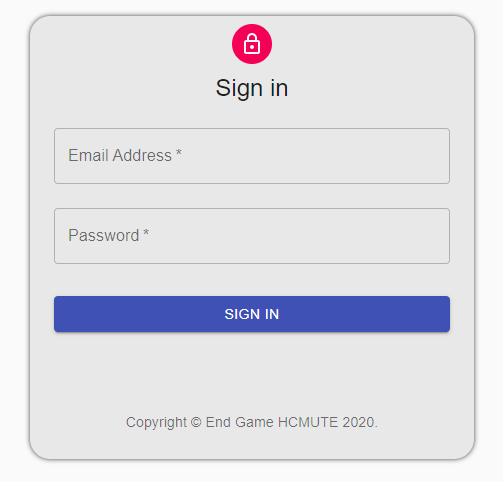
### Thiết kế giao diện website

Trong ReactJs, các components hoạt động giống như các hàm trả về các phần tử HTML. Được tạo ra trong phương thức render() với Class Component. Có 2 loại components là Class Component và Functional Component.

Với giao diện web này chúng tôi thực hiện cả hai Class Component và Functional Component để tạo ra một trang hoàn chỉnh.

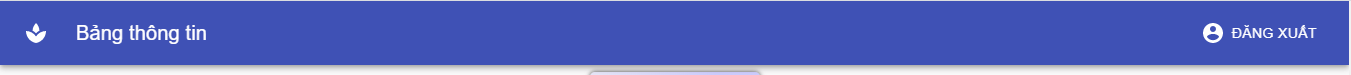
Giao diện web tổng thể gồm các components chính như sau:

 <Card />

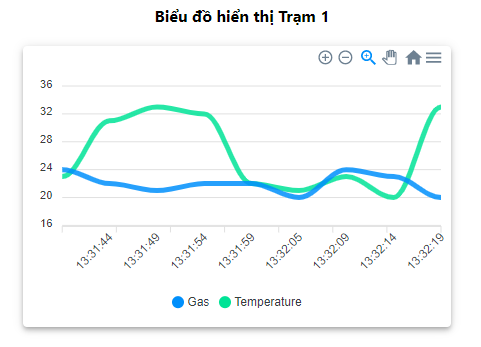
Hình 3‑32 *Component <Card/> khi được render*

<SignIn />

Hình 3‑33 *Component <SignIn/> khi được render*

<AppBar />

Hình 3‑34 *Component <AppBar/> khi được render*

<Chart />

Hình 3‑35 *Component <Chart/> khi được render*

<ExpandButton />

Hình 3‑36 *Component <ExpandButton/> khi được render*

lưu đồ tổng thể

### Máy chủ và hệ cơ sở dữ liệu

Như đã nếu các phần trước, cơ sở dữ liệu là một phần không thể thiếu trong một hệ thống IoT. Hệ thống sử dụng hệ cơ sở dữ liệu với khả năng quản lý dữ liệu lâu dài, khả năng truy nhập các khối dữ liệu lớn một cách hiệu quả. Dữ liệu được cung cấp cho chúng ta dưới dạng JSON và luôn đồng bộ thời gian thực đến mọi kết nối thiết bị người dùng.

Hiện tại có 2 phương pháp thiết kế đồ hình chuỗi thời gian liên tục (time-series schema) cho hệ thống.

***Cách 1:*** Cứ có dữ liệu đến là lưu lại thành một document (One document per data point)

  {

    "\_id" : ObjectId("5f0b36297050be4e04c9f831"),

    "type" : "temperature",

    "valueNode1" : "26",

    "valueNode2" : "26",

    "date" : "Jul 12 2020",

    "time" : "23:11:21",

    "\_\_v" : 0

  },

  {

    "\_id" : ObjectId("5f0c6fba729b4f15e0d9be0d"),

    "type" : "temperature",

    "valueNode1" : "28",

    "valueNode2" : "28",

    "date" : "Jul 13 2020",

    "time" : "21:29:14",

    "\_\_v" : 0

  }

***Cách 2:*** Gộp dữ liệu lại và lưu mỗi document trong một phút (Time-based bucketing of one document per minute)

{

    "\_id" : ObjectId("5f0b36297050be4e04c9f831"),

    "type" : "temperature",

    "valueNode1" : {

      "1" : "26",

      "2" : "26",

      "3" : "26",

      ...

      "58" : "26",

      "59" : "26",

    },

    "valueNode2" : {

      "1" : "26",

      "2" : "26",

      "3" : "26",

      ...

      "58" : "26",

      "59" : "26",

    },

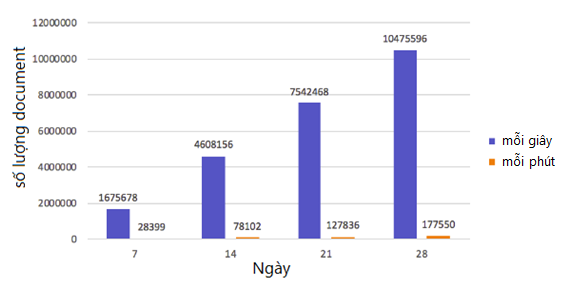
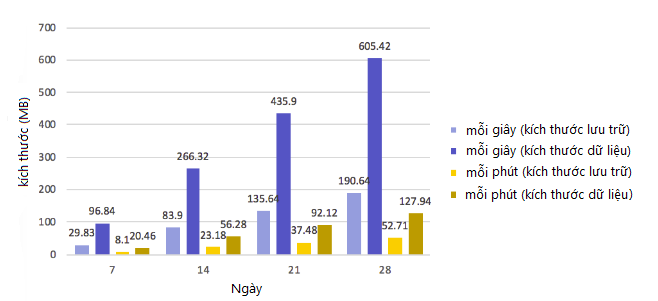
    "date" : "Jul 12 2020",

    "time" : "23:11:00",

    "\_\_v" : 0

  }

Về cơ bản, sự khác nhau giữa 2 cách trên là về sự lưu trữ dữ liệu. Chúng ta cùng phân tích sơ bộ về kích thước lưu trữ và tác động đến bộ nhớ dựa trên dữ liệu thu thập trong 4 tuần với thông tin từ 3 cảm biến.

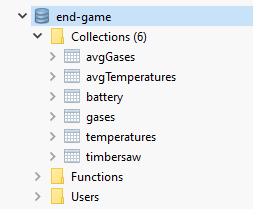
**Hình 3‑37 *Số lượng document theo tuần*

Hình 3‑38 *Kích thước dữ liệu theo tuần*

Qua các hình ảnh số liệu cho thấy dữ liệu được thu thập từ cảm biến và gửi đến gateway xử lý và được gửi lên máy chủ xử lý một lần nữa là rất lớn. Vì vây, ta cần tìm cách để giải quyết bài toán lưu trữ và xử lý khối dữ liệu to lớn này.

Do MongoDB được phát triển dựa trên Javascript framework, nên việc truy vấn dữ liệu trên hệ cơ sở dữ liệu này áp dụng những công cụ từ Javascript để duyệt dữ liệu. Theo thống kê [10], để duyệt một mảng với 10000 phần tử thì Javascript có thể mất đến ~113µs. Trong khi đó, dữ liệu ngày càng phình to nên ta cần một giải pháp về kiến trúc lưu trữ dữ liệu cho mô hình này.

Nhóm sẽ gửi dữ liệu lên máy chủ với tần số 5 giây cho mỗi document. Khi đó, dữ liệu của một collection sẽ là ~17280 documents từ đây sẽ giúp việc truy vấn dữ liệu sẽ nhanh hơn, giúp cải thiện hiệu năng và giảm áp lực thực thi trên máy chủ.

******Qua mỗi ngày, dữ liệu sẽ được xóa trên mỗi collection và tạo một document mới trên các collection dành cho dữ liệu trung bình hằng ngày.

Hình 3‑39 *Các documents của mô hình*

Các cặp khóa – giá trị của mỗi document:

{

    "\_id" : ObjectId("") – id được tạo tự động,

    "type" : "loại cảm biến",

    "valueNode1" : "giá trị của cảm biến tại node 1",

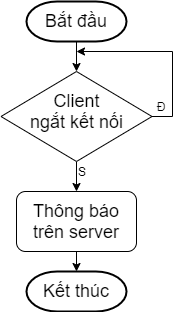
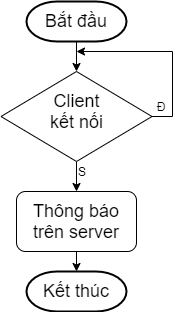
    "valueNode2" : "giá trị của cảm biến tại node 2",

    "date" : "ngày thu thập",

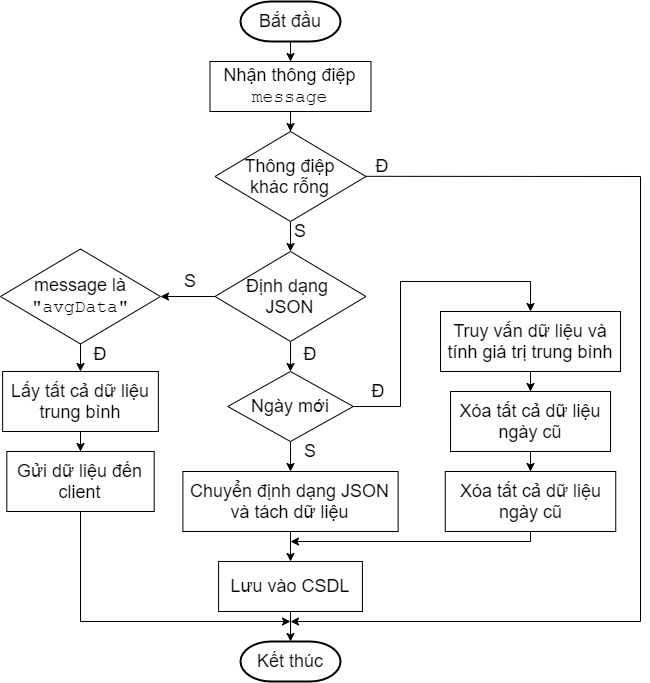
    "time" : "thời gian thu thập ",

    "\_\_v" : 0

}

 Lưu đồ hoạt động của server

Lưu đồ lắng nghe open, close

lưu đồ lắng nghe message

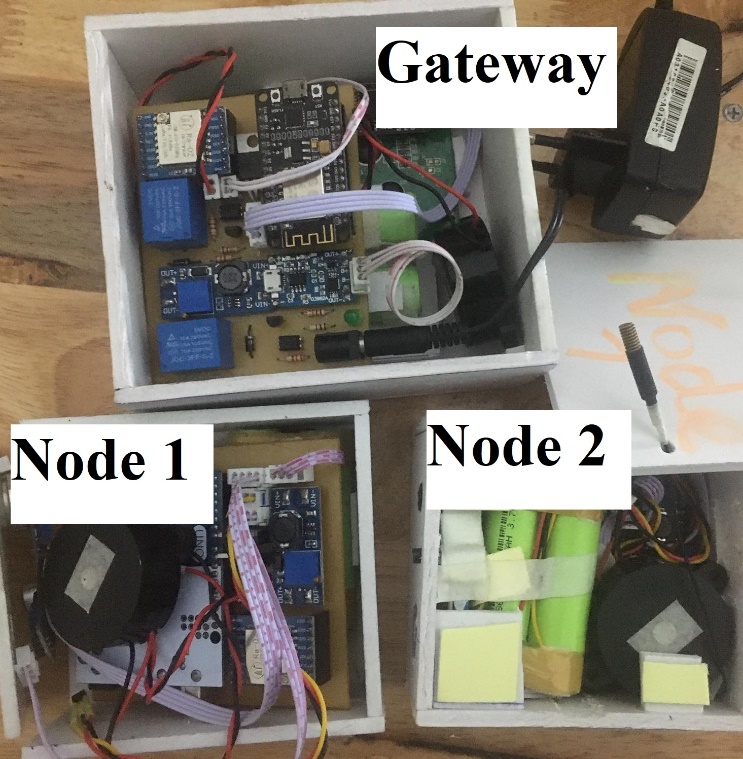
# kết quả thực hiện

## KẾT QUẢ

Sau quá trình tìm hiểu, nghiên cứu cùng sự hướng dẫn tận tình của giảng viên hướng dẫn Thầy Th.S Nguyễn Văn Phúc chúng tôi đã hoàn thành đồ án tốt nghiệp với đề tài “THIẾT KẾ VÀ THI CÔNG MÔ HÌNH CẢNH BÁO TRỘM GỖ, CHÁY RỪNG ỨNG IOT”.

Với những kết quả nghiên cứu thực hiện ban đầu chúng tôi đã phát triển một mô hình cảnh báo trộm gỗ, cháy rừng với hai node cảm biến và một gateway tại rừng và tại nơi theo dõi. Mô hình đáp ứng được những điều kiện cần thiết để thu thập nhiệt độ, nồng độ khí CO, âm thanh của máy.

Do học kỳ này, với diễn biến phức tạp của dịch COVID-19, chúng tôi hạn chế gặp nhau, khó khăn địa điểm, cũng như kinh phí để thực hiện một hệ thống IoT tại khu rừng thật. Chúng tôi chỉ thực hiện trên mô hình nhỏ với đầy đủ những yêu cầu và chức năng đã đặt ra.



Hình 4. 1 *Mô hình hệ thống sử dụng mạng IoT*

Trong quá trình thực hiện, chúng tôi đã lên ý tưởng thiết kế một mô hình gồm hai phần tách biệt tại rừng và tại nơi theo dõi.

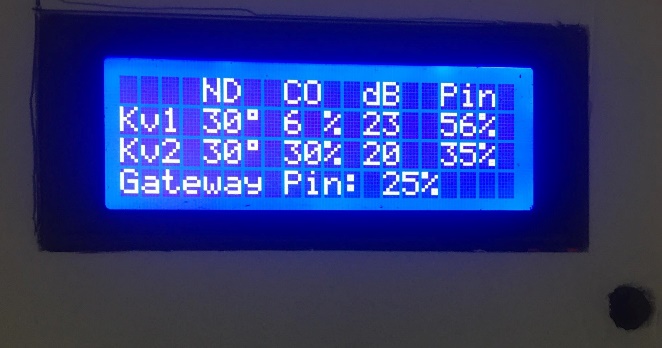
Gateway được lắp đặt tại nơi thuận tiện để có thể theo dõi, quan sát thường xuyên. Tại nơi này sẽ thực hiện nhận dữ liệu từ node 1 và node 2 là các thông số nhiệt độ, nồng độ khí CO, âm thanh máy cưa gỗ, năng lượng pin còn sống. Dữ liệu nhận được sẽ gửi lên server, hiển thị ra màn hình LCD, đồng thời cảnh báo qua còi khi có sự cố bất thường.

Node 1, node 2 được đặt ở những khu vực trong rừng, khu vực này có thể thường xuyên xảy ra tình trạng cháy, trộm gỗ hoặc nhiệt độ cao hoặc kiểm lâm khó có thể đến để kiểm tra thường xuyên.

## KẾT QUẢ THỰC NGHIỆM

Các kết quả thực nghiệm ban đầu chọn chu kỳ thu thập dữ liệu là 30 giây.

Cấp nguồn cho các khối mạch trong hệ thống, cấp nguồn adapter và pin dự phòng cho khối gateway, nguồn pin cho khối node 1, 2. Sau khi cấp nguồn, chờ hệ thống khởi tạo ban đầu, hệ thống hoạt động và kết quả thể hiện như các hình bên dưới:



Hình 4.2 Hiển thị thông tin ra LCD

****Ở hình 4.2, thông tin hiển thị ra LCD gồm có giá trị nhiệt độ, nồng độ khí CO, âm thanh thu thập được chuyển về dB, khả năng pin có thể cung cấp, được hiển thị cho cả hai khu vực node 1 và 2. Đồng thời, cũng hiển thị khả năng pin dự phòng có thể cấp và tình trạng có đang sạc pin không.

SỬA LẠI GIAO DIỆN< ĐỔI HÌNH

****

**Màn hình chính của Trạm 1 và Trạm 2**

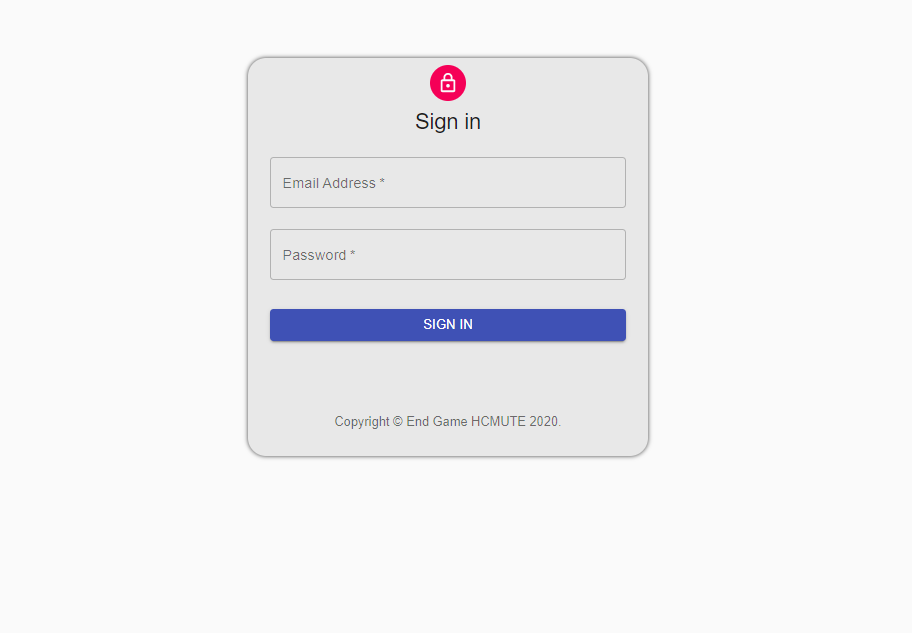
**Chọn xem thời lượng pin và tình trạng trộm gỗ**

****

**Chọn chu kỳ thu thập thông tin**

**Giao diện web**

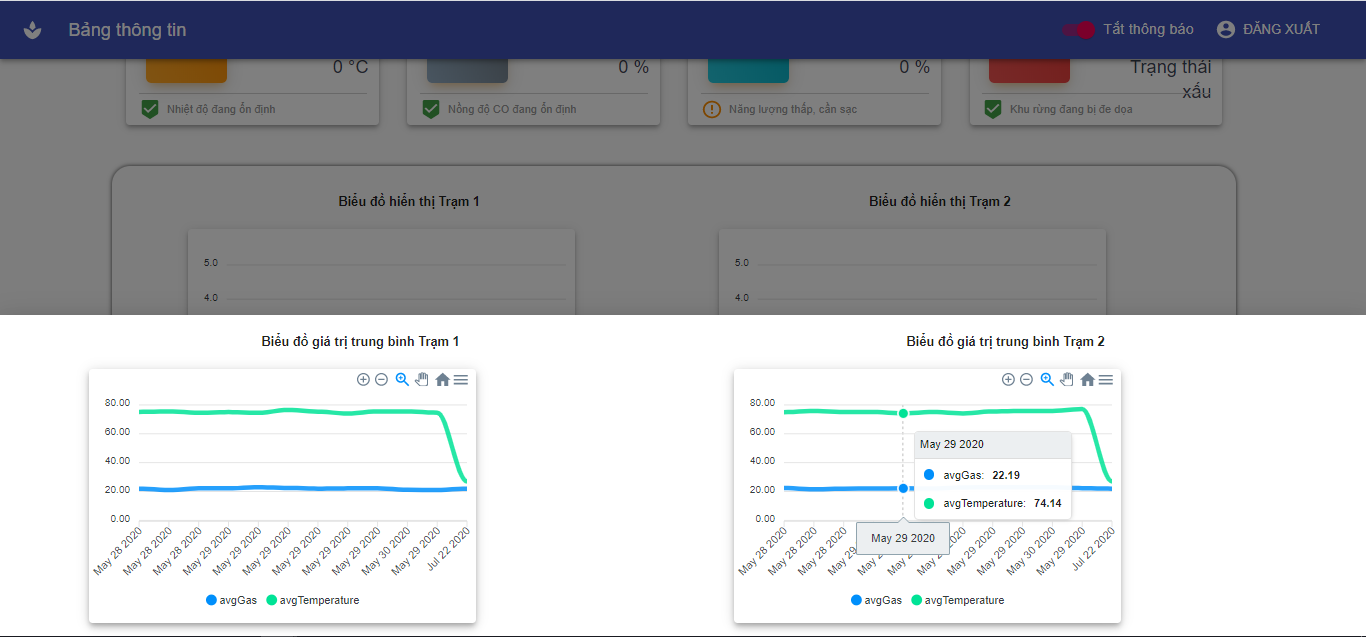
Màn hình đăng nhập





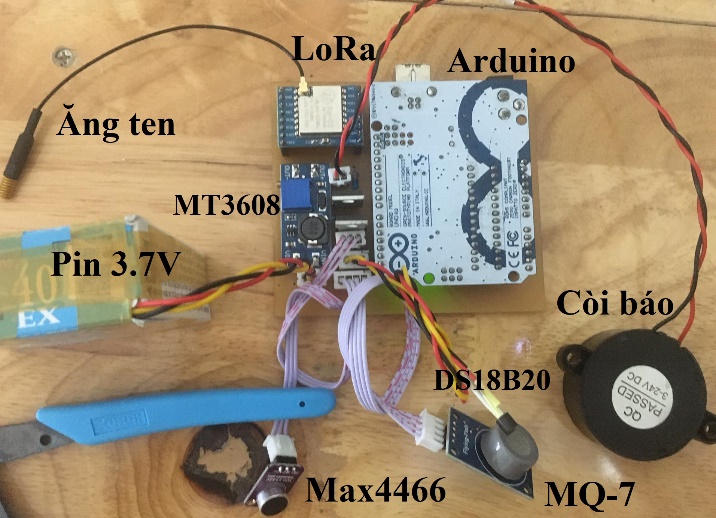
Hình .3 Biểu đồ thể hiện khả năng đáp ứng của cảm biến với sự thay đổi nhiệt độ, khí CO.

Hình 4.3 là biểu đồ đánh giá độ nhạy của cảm biến DS18B20 với sự thay đổi của môi trường (hơi nước đang sôi) trong 30 giây đầu kể từ khi bắt đầu đo.



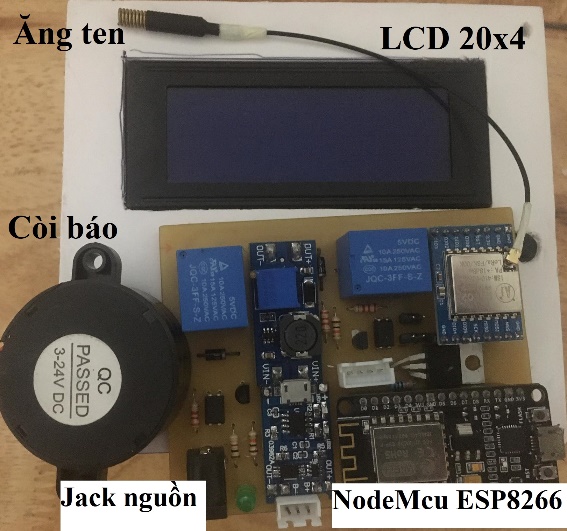
**Chế độ xem các giá trị trung bình theo ngày**

## KẾT QUẢ PHẦN CỨNG



Hình 4.2 Ảnh thực tế khối Node

Khối node cảm biến gồm có cảm biến DS18B20, MQ-7, Max4466, còi báo, pin, MT3608, Lora + ăng ten, kit Arduino uno và được kết nối với bo mạch thể hiện ở hình 4.2.



Hình 4.2 Ảnh thực tế khối Gateway

Khối gateway gồm có NodeMcu ESP8266, Lora + ăng ten, LCD 20x4, còi báo, relay chọn nguồn, relay sạc, MT3806, TP4056 và được kết nối với bo mạch thể hiện ở hình 4.3.

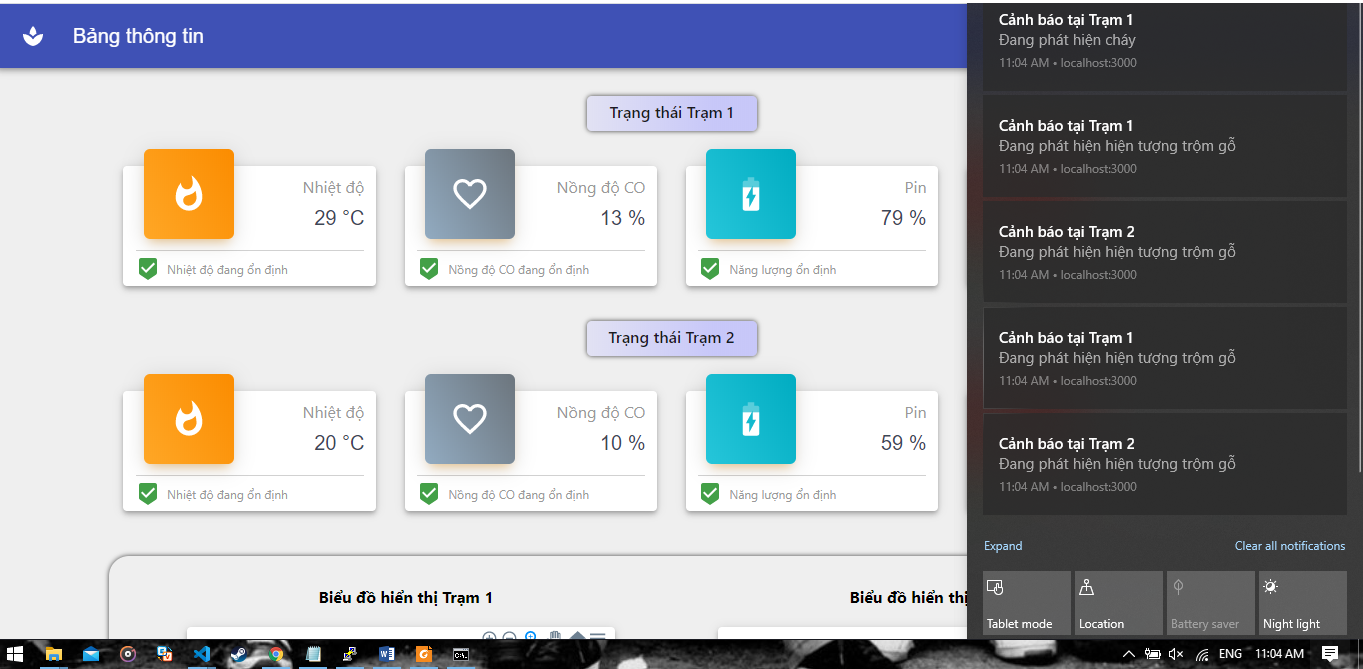
## KẾT QUẢ Phầm mềm

Màn hình chính của web gồm các khối nhỏ hiển thị giá trị của cảm biến tại thời điểm hiện tại. Đồng thời, web sẽ lưu lại các trạng thái đó để hiển thị thành dạng biểu đồ nhằm mục đích dễ dàng theo dõi tình hình.

Màn hình chính của ứng dụng di động cũng gồm 1 khối hiển thị các thông số cơ bản về tình trạng pin của node và tình hình âm thanh của khu rừng; và 1 khối biểu đồ tương tự như web.

Khi người dùng chọn vào nút *Xem Giá Trị Trung Bình* trên ứng dụng web và nút *Xem Theo Ngày* trên ứng dụng di động, web và ứng dụng di động sẽ hiển thị 2 biểu đồ biểu diễn các giá trị trung bình của các node đã thu thập được qua các ngày. Điều này giúp người dùng có thể phân tích, đánh giá được tình hình của khu vực nhằm đưa ra các quyết định để quản lý khu rừng tốt hơn.

Sự khác biệt giữa web và ứng dụng di động là web sẽ hiển thị cả 2 biểu đồ cho 2 node, mỗi biểu đồ hiển thị giá trị của 2 cảm biến trong khi trên ứng dụng di động thì mỗi biểu đồ chỉ có khả năng hiển thị một giá trị cảm biến.

Khi phát hiện các giả trị bất thường nhận được từ server, web sẽ gửi thông báo đến người dùng qua hệ điều hành họ đang dùng.

Kết quả hiển thị thông báo khi có sự cố trên Windows

Với ứng dụng di động, hệ thống sẽ gửi tín hiệu qua ứng dụng. Từ đó, ứng dụng sẽ đẩy thông báo đến người dùng thông qua ứng dụng.

Kết quả hiển thị thông báo khi có sự cố trên iOS

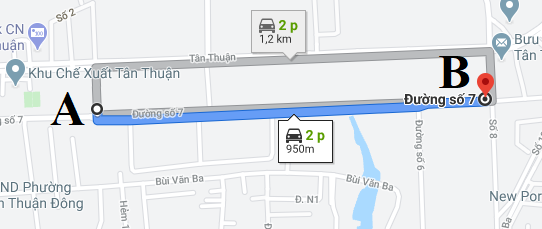
Ngoài ra, trên di động, người dùng có thể thay đổi tần suất thu thập thông tin và gửi lên server thông qua việc chọn chu kỳ gửi.

**Nhận xét:**

Hệ thống các cảm biến vẫn hoạt động chưa thật sự chính xác nhưng vẫn đáp ứng mục đích thu thập giá trị nhiệt độ, độ ẩm không khí, độ ẩm đất và với mục đích sử dụng chỉ là theo dõi các yếu tố môi trường cơ bản của cây trồng, thì sai số như vậy là trong mức cho phép được. Tuy nhiên do mặc hạn chế không có thiết bị đo giá trị chuẩn nên sử dụng dữ liệu trên website để làm chuẩn.

## Kiểm tra, đánh giá truyền gói tin của LoRa

Kiểm tra khoảng cách truyền, nhận tín hiệu thực tế của LoRa. Theo thông số lý thuyết do nhà sản xuất cung cấp, Lora có thể truyền, nhận lên tới khoảng cách 10km. Nhưng trên thực tế, chúng tôi đã kiểm tra và thu được kết quả như sau:



Hình 4.8 Khoảng cách truyền, nhận trong thực tế của Lora

Kết quả đo được là 950m trong điều kiện rộng, thoáng đường trống và xem như đây là môi trường không có rào cản. Chúng tôi thực nghiệm với một node truyền tín hiệu cho gateway. Đặt gateway ở điểm A như trên hình, sau đó di chuyển node đến từ từ đến vị trí cuối cùng là B và có thể nhận dữ liệu.

**Nhận xét**

Tuy khoảng cách như trên không đạt như lý thuyết là 10Km, nhưng với khả năng truyền nhận gần 1km là khá thấp để áp dụng trong thực tế của mô hình.

Với khoảng cách chúng tôi thực nghiệm thấp là do:

Chúng tôi chỉ đặt gateway và node cách mặt đất tầm 3m, với thông số cài đặt SF = 7 và BW = 125kHz.

Và trong Lora thì SF và BW ảnh hưởng lớn đến thời gian truyền gói tin cũng như khoảng cách truyền.

SF có giá trị từ 6 đến 12, SF càng lớn thì thời gian truyền lâu nhưng khoảng cách truyền xa hơn. Còn BW dùng để xác định biên độ, tần số. BW càng cao thì thời gian mã hóa Chipped Signal càng ngắn, thời gian truyền dữ liệu giảm, nhưng khoảng cách truyền ngắn.

# KẾT LUẬN - hướng phát triển

## kết luận

Đặc biệt hơn, trong học kỳ thực hiện đồ án tốt nghiệp này, do ảnh hưởng dịch bệnh COVID-19, nên chúng tôi gặp rất nhiều khó khăn khi thực hiện đồ án. Với những mục tiêu đặt ra ban đầu, qua quá trình tìm hiểu, phân tích, thiết kế, thi công. Và cơ bản khóa luận tốt nghiệp “Thiết kế và thi công mô hình cảnh báo trộm gỗ, cháy rừng ứng dụng IoT đã hoàn thành và đạt được kết quả đề ra. Đề tài đã giải quyết và hoàn thành được những yêu cầu ban đầu như: Xây dựng một hệ thống IoT công suất thấp, quản lý năng lượng, theo dõi các thông số môi trường, có hệ thống báo động tại chỗ và báo động từ xa thông qua Internet đến các ứng dụng web và di động.

Phần cứng được lựa chọn để đảm bảo về sự tối giản về thiết kế, tiết kiệm chi phí và hữu dụng. Có LCD tiện cho việc giám sát tại chỗ hoặc giám sát từ xa thông qua các ứng dụng web và di động đã thiết kế, hiển thị thông tin đầy đủ, trực quan, sinh động. Phần cứng được thiết kế nhỏ gọn, mô hình được sử dụng qua pin nên có thể đặt ở bất cứ nơi đâu. Các thông số được đo đạc chính xác với thực tế. Chi phí của hệ thống hợp lí và đáp ứng được đầy đủ các tính năng.

Về giao diện ứng dụng web và ứng dụng di động thì đơn giản, đủ thông tin cần thiết cho người dùng, dễ sử dụng. Ứng dụng di động được thiết kế sạch sẽ, code gọn gàng dễ bảo trì và hiệu năng cao, tiết kiệm năng lượng cho di động.

Bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại một vài hạn chế:

* Sự truyền nhận giữa các node và gateway chưa hoàn toàn ổn định, vẫn còn hiện tượng mất gói tin. Khoảng cách truyền nhận tương đối gần so với mục tiêu ban đầu.
* Các cảm biến chỉ thu thập ở phạm vi gần.
* Hệ thống mang tính mô hình chưa thể ứng dụng trong thực tế.
* Giao diện trên web và điện thoại chưa phản hồi hết những yêu cầu khi áp dụng trong thực tế yêu cầu.

## Hướng phát triển

Sau khi hoàn thành hệ thống và những kết quả thực hiện trong Chương 4, chúng tôi nhận thấy khả năng phát triển trong tương lai khả thi nên đã đề xuất những nghiên cứu mở rộng chức năng và tính ứng dụng của đề tài như sau:

* Thêm tính năng theo dõi các thông số thu thập của các cảm biến trung bình theo ngày, tuần, tháng, quý, năm... trực quan và dễ sử dụng như các biểu đồ hiển thị chứng khoán.
* Chọn lựa các cảm biến tốt hơn, phù hợp hơn để tăng phạm vi thu thập thông tin. Tăng thêm số lượng khối cảm biến và số lượng cảm biến trong khối.
* Ứng dụng Machine Learning vào việc phát hiện âm thanh. Phân tích, lọc nhiễu để nhận dạng được tiếng máy cưa trong tình huống tạp âm cao hoặc âm thanh có cường độ nhỏ.
* Thiết kế, tối ưu lại giao diện website và ứng dụng điện thoại mang tính thẩm mỹ hơn, giúp người dùng dễ dàng thao tác.
* Dùng tấm pin năng lượng mặt trời cho các node cảm biến để giảm thời gian, chi phí cho việc bảo trì, được thể hiện chi tiết ở phần *3.5.2 Giải pháp khắc phục năng lượng cho hệ thống*.

# TÀI LIỆU THAM KHẢO

[1] Thông tin ngành Lâm nghiệp, “*Hiện trạng, nguyên nhân và cách phòng chống cháy rừng ở Việt Nam”,* <http://tuaf.edu.vn/khoalamnghiep/bai-viet/hien-trang-nguyen-nhan-va-cach-phong-chong-chay-rung-o-viet-nam-20033.html>

[2] “*Thực trạng cháy rừng ở Việt Nam hiện nay*”,

<https://socialforestry.org.vn/chay-rung-o-viet-nam/>

[3] Wikipedia, *“Nạn phá rừng ở Việt Nam”,*

[https://vi.wikipedia.org/wiki/](https://vi.wikipedia.org/wiki/N%E1%BA%A1n_ph%C3%A1_r%E1%BB%ABng_%E1%BB%9F_Vi%E1%BB%87t_Nam)Nạn\_phá\_rừng\_ở\_Việt\_Nam

[4] Công ty PHA Việt Nam, *“Giải pháp quản lý và cảnh báo sớm cháy rừng”*,

[*https://www.phadistribution.com/index.php/giai-phap/iot/qun-ly-va-cnh-bao-sm-chay-rng*](https://www.phadistribution.com/index.php/giai-phap/iot/qun-ly-va-cnh-bao-sm-chay-rng)

[5] Thông tin sản phẩm “*Libelium Smart Environment IoT Vertical Kit”*

[*https://www.the-iot-marketplace.com/libelium-smart-environment-iot-vertical-kit*](https://www.the-iot-marketplace.com/libelium-smart-environment-iot-vertical-kit)

[6] Ngô Tấn Lợi – Trịnh Đình Hiệp, *“Hệ thống cảnh báo cháy rừng ứng dụng công nghệ LoRa”, 2015.*

[7] N. H. Quảng, N. H. Minh, N. T. Hà, N. T. Hải & N. T. Hùng, *“Thu nhận và sử dụng dữ liệu Modis phục vụ quản lý lữa rừng tại Việt Nam”*, Cục kiểm lâm, Bộ Nông nghiệp và Phát triển Nông thôn, 2007.

[8] Tran Quang Bao; Le Ngoc Hoan, *“Kiểm chứng khả năng phát hiện cháy rừng từ ảnh vệ tinh MODIS ở Việt Nam”.*

[9] ĐH. Bách Khoa Đà Nẵng, *“Mạng không dây LoRa cho ứng dụng IoT tầm xa*”, <https://congnghiepcongnghecao.com.vn/tin-tuc/t22516/mang-khong-day-lora-cho-ung-dung-iot-tam-xa.html>

[10] “Ứng dụng Lora”, *https://www.getblocky.com/*

[11] Taoufik Bouguera, Jean-François Diouris, Jean-Jacques Chaillout, Randa, “*Jaouadi, Guillaume Andrieux Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN”,* <https://hal.archives-ouvertes.fr/hal-01828769>

[12] Yotam Kadishay, *“3 JavaScript Performance Mistakes You Should Stop Doing”*, [*https://hackernoon.com/3-javascript-performance-mistakes-you-should-stop-doing-ebf84b9de951*](https://hackernoon.com/3-javascript-performance-mistakes-you-should-stop-doing-ebf84b9de951)

[13] Vấn đề về quản lí năng lượng trên thiết bị IoT, <https://www.makerlab.vn/van-de-ve-quan-li-nang-luong-tren-thiet-bi-iot/>

[14] B. Martinez, X. Vilajosana, F. Chraim, I. Vilajosana, and K. Pister, “When scavengers meet industrial wireless,” Industrial Electronics, IEEE Transactions on Industrial Electronics, vol. 62, pp. 2994–3003, May 2015.

[15] Arduino Uno Tutorial [Pinout], <https://diyi0t.com/arduino-uno-tutorial/>

[16] Ngô Thành Đạt, Lê Khải Nguyên, “Thiết kê và thi công hệ thống IOT chăm sóc vườn cây ăn quả sử dụng pin năng lượng mặt trời” Đồ Án Tốt Nghiệp, Trường ĐH Sư Phạm Kỹ Thuật Tp.HCM, 2018.

[17] B. Martinez, M Montón, F. Chraim, I. Vilajosana, and J. D. Prades, “The Power of Models: Modeling Power Consumption for IoT Devices,” Industrial Electronics, IEEE Transactions on, vol. 15, pp. 5777–5789, Oct 2015.

[18] Mạng không dây LoRa cho ứng dụng IoT tầm xa, Công nghiệp Công nghệ cao, <https://congnghiepcongnghecao.com.vn/>

Các trang web tham khảo

1. <https://iotmaker.vn/module-lora-sx1278-433mhz.html>
2. Migrating from version 5 to 6, <https://arduinojson.org/v6/doc/upgrade/>
3. TUTORIAL:A GUIDE TO PUTTING YOUR ARDUINO TO SLEEP,

https://thekurks.net/blog/2018/1/24/guide-to-arduino-sleep-mode/

[19] SparkFun Electronics, *“Reducing Arduino Power Consumption”*, <https://learn.sparkfun.com/>

[20] *“How to Reduce Arduino Uno Power Usage by 95%”*, <https://www.defproc.co.uk/>

[1] Vấn đề về quản lí năng lượng trên thiết bị IoT

<https://www.makerlab.vn/van-de-ve-quan-li-nang-luong-tren-thiet-bi-iot/>

1. [2] Nguyễn Đình Phú, Giáo trình *Vi điều khiển*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.

Huỳnh Xuân Dũng – Trần Nhật Minh, Đồ án tốt nghiệp: *Hệ thống IoT điều khiển*

*và giám sát ngôi nhà*, 2018

[15] Arduino Uno Tutorial [Pinout]

<https://diyi0t.com/arduino-uno-tutorial/>

[11] Lê Công Vĩnh Khải, “*Một số so sánh giữa công nghệ Zigbee và LoRa*”. [tapit.vn].

**. Trang web tham khảo**

1. www.alldatasheet.com
2. www.hshop.vn
3. www.arduino.vn
4. www.instructables.com

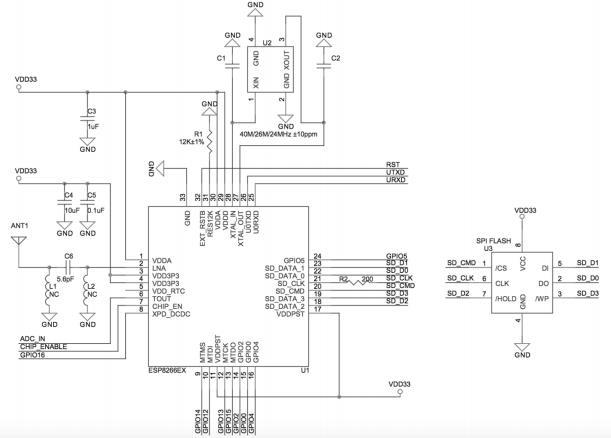
PHỤ LỤC

## Chip ESP8266

Chip ESP8266 được phát triển bởi Espressif để cung cấp giải pháp giao tiếp Wi-Fi cho các thiết bị IoT. Điểm đặc biệt của dòng ESP8266 là nó được tích hợp các mạch RF, TX power amplifier và RX filter, với kích thước chỉ 5x5 mm nên các board sử dụng ESP8266 không cần kích thước board lớn cũng như không cần nhiều linh kiện xung quanh.

Cấu trúc phần cứng của dòng chip sử dụng ESP8266 có thể tóm tắt như sau:

* Sử dụng 32-bit MCU core có tên là Tensilica.
* Tốc độ xung hệ thống có thể thiết lập ở 80MHz hoặc 160 MHz.
* Không tích hợp bộ nhớ FLASH để lưu chương trình.
* Tích hợp 50 KB RAM để lưu dữ liệu ứng dụng khi chạy.
* Có đầy đủ các ngoại vi chuẩn để giao tiếp.
* Tích hợp các mạch RF để truyền nhận dữ liệu ở tần số 2.4 GHz.

Do không hỗ trợ bộ nhớ Flash nên các board sử dụng ESP8266 phải gắn thêm chip FLASH bên ngoài và thường là FLASH SPI để ESP8266 có thể đọc chương trình ứng dụng với chuẩn SDIO hoặc SPI.

Hình 1 *Mạch nguyên lý đầy đủ ESP8266*

Có thể thấy board ESP8266 chỉ cần thạch anh, SPI FLASH chip và vài linh kiện điện trở. Do đó việc tích hợp giao tiếp Wi-Fi vào board ứng dụng với ESP8266 rất dễ dàng và nhanh chóng.

Công ty Espressif cũng cung cấp khá đầy đủ tài liệu và công cụ phần mềm để các nhà phát triển sản phẩm có thể nhanh chóng làm quen và phát triển ứng dụng với ESP8266.

Về mô hình lập trình ứng dụng với ESP8266, có thể chia làm 2 loại:

* Sử dụng firmware được cung cấp bởi Espressif và giao tiếp thông qua tập lệnh AT.
* Lập trình firmware trực tiếp vào ESP8266 sử dụng bộ thư viện SDK cung cấp bởi Espressif.

Hiện tại có khá nhiều module khác nhau cho ESP8266 được sản xuất bởi công ty AI-Thinker. Đặc điểm khác nhau giữa các module này gồm:

* Loại anten sử dụng (PCB anten, chip anten hoặc gắn anten ngoài)
* Dung lượng của chip FLASH SPI trên board
* Kích thước board của module.
* Có gắn khung nhôm chống nhiễu hay không.
* Số lượng pin GPIO đưa ra chân kết nối.

Hiện tại AI-Thinker sản xuất 14 loại module cho ESP từ module ESP-01 đến ESP-14. Ở thị trường Việt Nam thì các module là ESP-01, ESP-07 và ESP-12, ESP NodeMCU khá phổ biến.

## Cảm Biến Nhiệt Độ

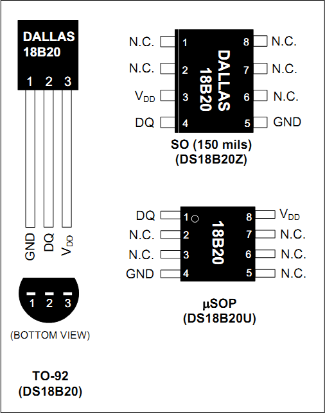
Cảm biến nhiệt độ là thiết bị dùng để cảm nhận sự biến đổi về nhiệt độ của đại lượng cần đo. Đối với các loại cảm biến nhiệt thì có 2 yếu tố ảnh hưởng trực tiếp đến độ chính xác đó là “Nhiệt độ môi trường cần đo” và “Nhiệt độ cảm nhận của cảm biến”.

#### Giới thiệu

Cảm biến nhiệt DS18B20 đã có bộ chuyển đổi ADC bên trong và chỉ có 1 tín hiệu giao tiếp vào ra còn gọi là 1 dây hay one wire.

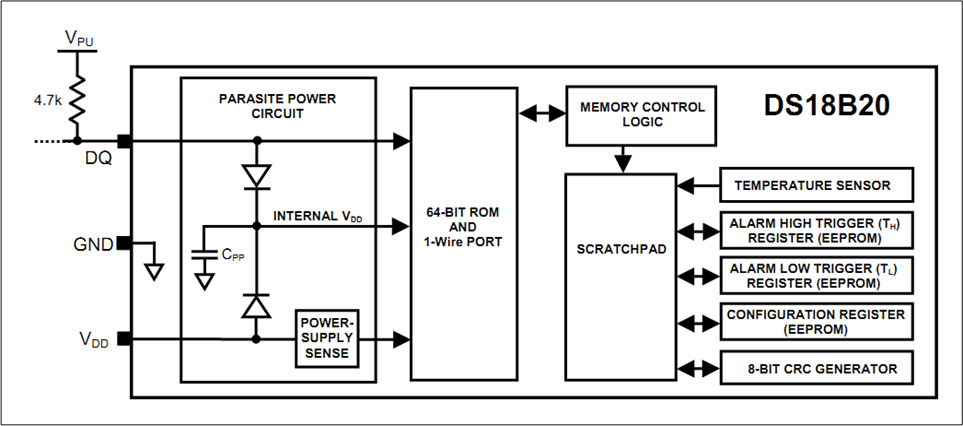
Các thông tin của cảm biến như sau:

* Chỉ sử dụng một dây giao tiếp.
* Mỗi cảm biến đều có mã 64 bit lưu trong bộ nhớ ROM.
* Là cảm biến đa năng thay cho hệ thống cảm biến nhiệt phức tạp.
* Phần cứng đơn giản thay vì chỉ dùng 1 dây không cần dùng ADC nên rất thích hợp cho vi điều khiển không tích hợp ADC.
* Có thể cấp nguồn từ đường tín hiệu, điện áp từ 3 V-5.5 V.
* Tầm đo từ -55 ᴼC đến +125 ᴼC (-67 ᴼF đến +257 ᴼF).
* Sai số 0.5 ᴼC cho tầm đo từ -10 ᴼC đến +85 ᴼC.
* Có thể chọn độ phân giải bằng phần mềm từ 9 đến 12 bit.
* Thời gian chuyển đổi lớn nhất cho 12 bit là 750 ms.
* Có thể cài đặt nhiệt độ cảnh báo lưu vào bộ nhớ ROM.
* Có nhiều dạng chân: loại 3 chân, 8 chân, hay loại dán.
* Thiết bị được ứng dụng rộng rãi trong kiểm soát nhiệt độ: hệ thống công nghiệp, sản phẩm hàng tiêu dùng, nhiệt kế hay bất kỳ một hệ thống cảm biến nhiệt nào.

Các hình dạng khác nhau và tên các chân cảm biến như hình sau:

Hình *Các loại cảm biến DS18B20*

#### Sơ đồ khối

Hình 3 *Sơ đồ khối của cảm biến DS18B20*

Các khối bao gồm:

* Khối mạch nguồn cấp cho các khối bên trong hoạt động lấy từ nguồn bên ngoài.
* Khối bộ nhớ ROM 64 bit và port 1 dây.
* Khối điều khiển bộ nhớ.
* Khối scratchpad liên kết với các khối cảm biến nhiệt độ, khối lưu nhiệt độ byte cao TH và byte thấp TL để báo động quá nhiệt, khối thanh ghi định cấu hình và bộ kiểm tra mã dùng chuẩn CRC.

#### Nguyên lý hoạt động

DS18B20 là cảm biến nhiệt độ chuyển đổi trực tiếp sang dữ liệu số. Độ phân giải của cảm biến có thể định cấu hình là 9, 10, 11 hoặc 12 bit tương ứng với hệ số chuyển đổi theo thứ tự là 0.5 ºC, 0.25 ºC, 0.125 ºC và 0.0625 ºC. Độ phân giải khi cấp điện là 12 bit.

Khi cấp điện thì DS18B20 ở trạng thái nghỉ để giảm công suất tiêu thụ. Để bắt đầu quá trình chuyển đổi tương tự sang số của nhiệt độ đo thì thiết bị chủ phải cấp phát lệnh chuyển đổi (có mã là 44H). Sau khi chuyển đổi xong thì dữ liệu nhiệt độ lưu vào 2 thanh ghi nhiệt độ trong vùng nhớ scratchpad và cảm biến trở lại trạng thái nghỉ.

Nếu cảm biến cấp nguồn bên ngoài thì thiết bị chủ có thể phát các khe thời gian đọc sau khi cấp phát lệnh chuyển đổi nhiệt độ để kiểm tra cảm biến chuyển đổi xong chưa: nếu cảm biến đáp ứng bằng cách phát bit có mức logic 0 thì quá trình chuyển đổi đang xảy ra, nếu phát bit có mức logic 1 thì quá trình chuyển đổi đã kết thúc.

Tất cả các giao tiếp với bus 1 dây đều phải bắt đầu với trình tự động. Trình tự khởi động bao gồm 1 xung reset được phát bởi thiết bị chủ và tiếp theo là xung của thiết bị tớ hay của cảm biến DS18B20. Chức năng của xung reset có chức năng kiểm tra thiết bị tớ là cảm biến có nối với bus hay không, nếu có thì thiết bị tớ là cảm biến sẽ trả lời bằng cách tạo ra 1 xung tương ứng – quá trình này còn gọi là kiểm tra sự hiện diện của cảm biến.

## Module cảm biến khí CO MQ-7

Nguyên lý hoạt động của cảm biến:

* Lớp cảm biến của MQ-7 được cấu tạo từ chất bán dẫn SnO2. Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất gây cháy, độ dẫn điện của SnO2 thay đổi.
* Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quang MQ-7 càng cao.
* Đơn vị đo của cảm biến là ppm (10 – 4%) – mật độ tương đối của 1 chất trong hỗn hợp cụ thể, trường hợp này là không khí. Trong hướng dẫn của nhà sản xuất không đề cập đến việc quy đổi giá trị nồng độ này ra điện áp ngõ ra.

Biến trở trên Module có chức năng điều chỉnh điện áp tham chiếu (ngưỡng), khi MQ-7 phát hiện CO đại đến ngưỡng thì chân CO đảo trạng thái.

## KHối RElay

### Giới thiệu về relay

Relay là một loại thiết bị điện tự động mà tín hiệu đầu ra thay đổi nhảy cấp khi tín hiệu đầu vào đạt những giá trị xác định. Relay là thiết bị điện dùng để đóng cắt mạch điện điều khiển, bảo vệ và điều khiển sự làm việc của mạch điện.

Các bộ phận chính của khối mạch relay

* Cơ cấu tiếp thu (khối tiếp thu): Tiếp nhận những tín hiệu đầu vào và biến đổi nó thành đại lượng cần thiết cung cấp tín hiệu cho khối trung gian.
* Cơ cấu trung gian (khối trung gian):Tiếp nhận những tín hiệu đưa đến từ khối tiếp thu và biến đổi nó thành đại lượng cần thiết cho rơle tác động.
* Cơ cấu chấp hành (khối chấp hành): Làm nhiệm vụ phát tín hiệu cho mạch điều khiển.

*Thông số kĩ thuật:*

* Điện áp điều khiển: 5V. Điện áp AC cực đại: 250 VAC.
* Dòng tiêu thụ: 80mA. Dòng điện cực đại: 10A
* Thời gian tác động: 10ms. Thời gian nhã hãm: 5ms.
* Nhiệt độ hoạt động: - 45 oC ~ 75oC

**Opto PC817** là một linh kiện bán dẫn cấu tạo gồm 1 bộ phát quang và một cảm biến quang tích hợp trong 1 khối bán dẫn. Bộ phát quang là 1 doide phát quang dùng để phát ra ánh sáng kích cho các cảm biến quang dẫn.

## Bộ chuyển đổi Adapter

* Khái niệm

Adapter là bộ chuyển đổi điện áp giữa các thiết bị điện tử sử dụng nguồn điện khác 220V xuống một điện áp thấp hơn. Thiết bị chuyển đổi này được gọi chung là Adapter.

* Chức năng chính của Adapter

Cung cấp nguồn điện phù hợp cho các thiết bị điện tử hoạt động ổn định.

Ngoài ra Adapter còn có thể sạc pin cho thiết bị.

* Các thông số kỹ thuật của Adapter
* Điện áp

Để cung cấp cho thiết bị cần sử dụng ổn định. Từ nguồn điện cao áp chuẩn (AC) sẽ được biến đổi thành điện áp thấp hơn (DC) để nạp vào pin hoặc cung cấp trực tiếp cho thiết bị hoạt động.

* Cường độ dòng điện

Được đo bằng đơn vị Ampe(A) để thiết bị hoạt động. Một thiết bị có ghi 3A nghĩa là dòng điện đầu vào phải có cường độ đủ 3A thì mới đáp ứng được. Cường độ dòng điện chuẩn của nguồn 220V thường là 12A các thiết bị sử dụng adapter phải có cường độ thấp hơn con số này.



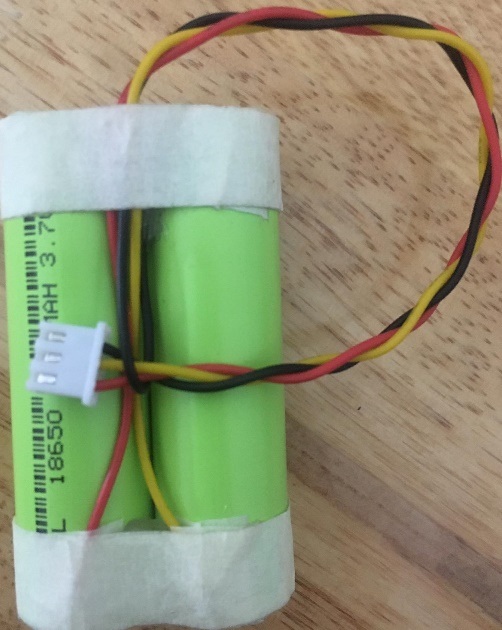
Hình 4 Hình ảnh adapter

## Pin li-ion

Pin li-ion 3.7V có kích thước 18x65mm không quá lớn kết hợp với dung lượng cao và dòng xả tải ổn định nên được sử dụng nhiều trong các dự án cần tính di động và trong các mô hình IOT.

Pin có 2 thông số quan trọng, đầu tiên là dung lượng pin. Dung lượng pin có đơn vị mà mAh. VD: 1 viên pin ghi là 3000mAh, nghĩa là theo dung lượng danh nghĩa thì pin này có thể cấp 1 dòng điện liên tục 1000mA liên tục trong 3h thì hết pin.

Thông số quan trọng thứ 2 trên pin là hiệu điện thế 3.7V, nghĩa là từ lúc đầy pin đến lúc hết pin, hiệu điện thế trung bình của pin là 3.7V.



Hình 5 Hình ảnh hai pin li-ion ghép song song

Những lưu ý khi sử dụng Pin:

* Pin lithium ion (Li-Ion) có thể gây nguy hiểm nếu sử dụng không đúng cách.
* Cần sạc đầy trước khi sử dụng cho lần đầu.
* Tất cả các pin lithium đều có chu kỳ sống, tuy nhiên pin phải được thay 2 năm một lần và tùy theo điều kiện.
* Không bao giờ xả hết pin
* Không tiếp xúc với nhiệt độ cực cao

## Giới thiệu phần mềm lập trình Arduino IDE

Môi trường phát triển tích hợp Arduino IDE là một ứng dụng đa nền tảng được viết bằng Java. Với ưu điểm là một môi trường lập trình đơn giản, dễ sử dụng, ngôn ngữ lập trình dễ hiểu và dựa trên nền tảng C/C++ quen thuộc đối với người học kỹ thuật cũng như rất dễ làm quen cho những người mới bắt đầu và quan trọng nhất là Arduino là mã nguồn mở, được hỗ trợ kho thư viện khổng lồ và có một cộng đồng người sử dụng rộng lớn.

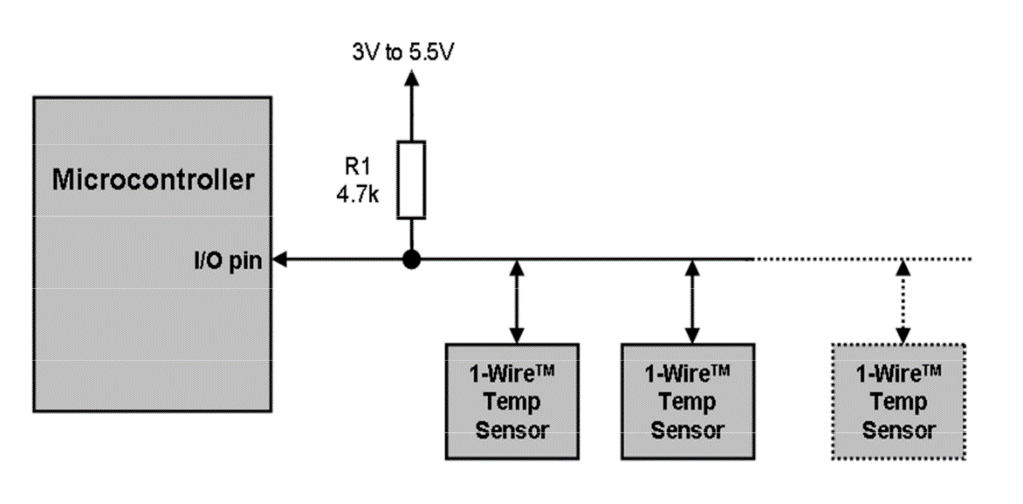
Để lập trình cho Arduino và ESP8266, nhóm đã sử dụng phần mềm Arduino IDE (**I**ntergrated **D**evelopment **E**nvironment). Arduino IDE rất dễ sử dụng, sử dụng ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

## CÁC CHUẨN GIAO TIẾP

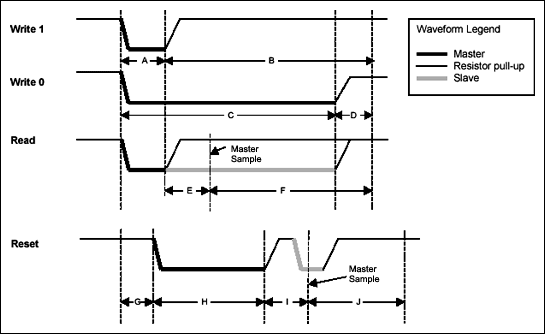
### Chuẩn giao tiếp one-wire

Chuẩn giao tiếp 1 dây (one-wire) được thiết kế bởi Dallas Semiconductor và được Maxim mua lại năm 2001. One-Wire dùng một dây để truyền nhận nên có tốc độ thấp. Chủ yếu sử dụng cho việc thu thập dữ liệu, truyền nhận dữ liệu thời tiết, nhiệt độ, công việc không yêu cầu tốc độ cao. Là chuẩn giao tiếp không đồng bộ và bán song công (half-duplex). Giao tiếp tuân theo mối quan hệ chủ tớ một cách chặc chẽ. Trên cùng một bus thì chúng ta có thể gắn 1 hoặc nhiều thiết bị slave nhưng chỉ có một master có thể kết nối được với bus này. Khi không có dữ liệu trên đường truyền thì bus dữ liệu được xem là ở trạng thái rãnh.



Hình 5 Sơ đồ giao tiếp One – Wire

Để giao tiếp được với vi điều khiển, tín hiệu trên bus one-wire chia thành các khe thời gian 60 µs. Một bit dữ liệu được truyền trên bus dựa trên khe thời gian (time slots). Các thiết bị slave khác nhau cho phép có thời gian quy định khác nhau. Nhưng quan trọng nhất trong chuẩn giao tiếp này là cần chính xác về thởi gian. Vì vậy để tối ưu đường truyền thì cần một bộ định thời để delay chính xác nhất.



Hình 6 Dạng sóng hoạt động của giao tiếp One - Wire

Bốn thao tác hoạt động cơ bản của bus 1 wire là Reset/Presence, gửi bit 1, gửi bit 0, và đọc bit cụ thể là:

* Write 1 (gửi bit 1): Master kéo xuống 0 một khoảng A (us) rồi về mức 1 khoảng B (us).
* Write 0 (gửi bit 0): Master kéo xuống 0 khoảng C (us) rồi trả về 1 khoảng D.
* Read (Đọc một Bit): Master kéo xuống 0 khoảng A rồi trả về 1 delay khoảng E rồi đọc giá trị slave gửi về delay F (us).
* Restart: Master kéo xuống 0 một khoảng H rồi nhả lên mức 1 sau đó cấu hình Master là chân In delay I (us) rồi đọc giá trị slave trả về. Nếu bằng 0 thì cho phép giao tiếp, nếu bằng 1 đường truyền lỗi hoặc slave đang bận. Thiết bị master kéo bus xuống thấp ít nhất 8 khe thời gian (tức là 480 µs) và sau đó nhả bus. Khoảng thời gian bus ở mức thấp đó gọi là tín hiệu reset. Nếu có thiết bị slave gắn trên bus nó sẽ trả lời bằng tín hiệu Presence tức là thiết bị tớ sẻ kéo bus xuống mức thấp trong khoảng thời gian 60µs.

### Giao tiếp SPI

SPI - Serial Peripheral Bus là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất. Đây là kiểu truyền thông Master-Slave, trong đó có một chip Master điều phối quá trình tuyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave. SPI là một cách truyền song công (full duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn đó là SCK (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Ouput Slave Input) và SS (Slave Select).



Hình 7 Kết nối 1 thiết bị với Arduino theo chuẩn giao tiếp SPI

SCK: Xung giữ nhịp cho giao tiếp SPI vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi. Sự tồn tại của chân SCK giúp quá trình truyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.

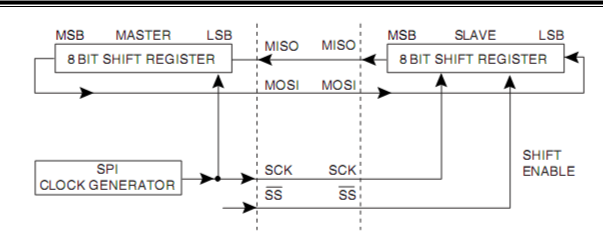
MISO: Nếu chip Master thì là đường Input còn nếu là chip Slave thì MISO là Output. MISO của Master và các Slaves được nối trực tiếp với nhau.

MOSI: Nếu là chip Master thì là đường Output còn nếu là chip Slave thì MOSI là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.

SS: Đường chọn Slave cần giao tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc. Nếu chip Master kéo đường SS của một Slave nào xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó. Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master.

* Nguyên lí hoạt động:

Mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bit. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, 1bit trong thanh ghi của Master được truyền qua Slave trên đường MOSI, đồng thời 1bit trong thanh ghi của chip Slave cũng được truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip được gởi qua lại đồng thời nên quá trình truyền dữ liệu này được gọi là “song công”. Hình dưới mô tả quá trình truyền 1 gói dữ liệu thực hiện bởi module SPI trong AVR, bên trái là chip Master và bên phải là Slave.

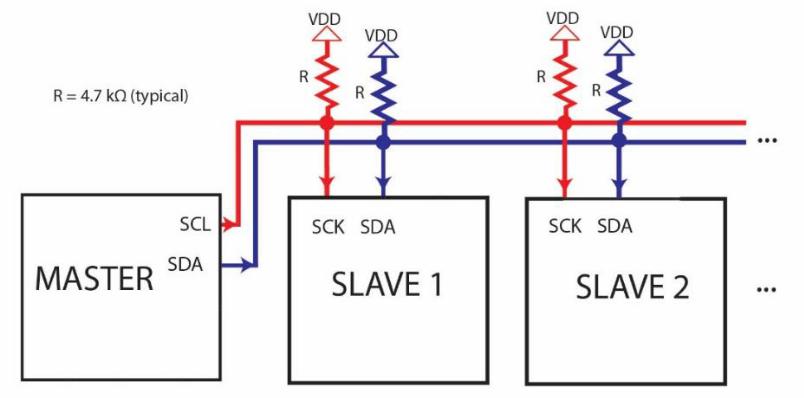


Hình 8 Hoạt động truyền nhận dữ liệu của chuẩn giao tiếp SPI

### Giao Tiếp I2C

Giới thiệu

I2C – Inter Intergrated Circuit, được phát triển bởi Philips Semiconductors để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một board mạch chỉ sử dụng hai đường truyền tín hiệu. Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM... Chip nhớ như: RAM tĩnh (Static Ram), EEPROM, bộ chuyển đổi tín hiệu tương tự sang tín hiệu số (ADC), bộ chuyển đổi tín hiệu số sang tín hiệu tương tự (DAC), IC điểu khiển LCD, LED...



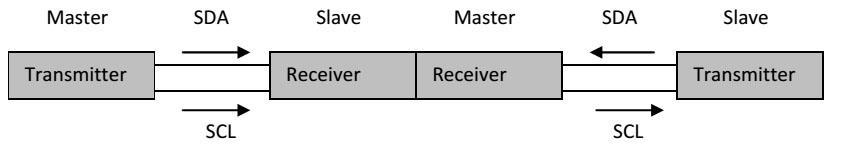
Hình 9 *Bus I2C và các thiết bị ngoại vi*

Đặc điểm

Một giao tiếp I2C gồm có 2 dây: Serial Data (SDA) và Serial Clock (SCL).

SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung để đồng bộ và chỉ theo một hướng.

Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên (pullup resistor). Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực máng hở (open-drain hay open-collector). Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng 1 KΩ đến 4.7 KΩ.

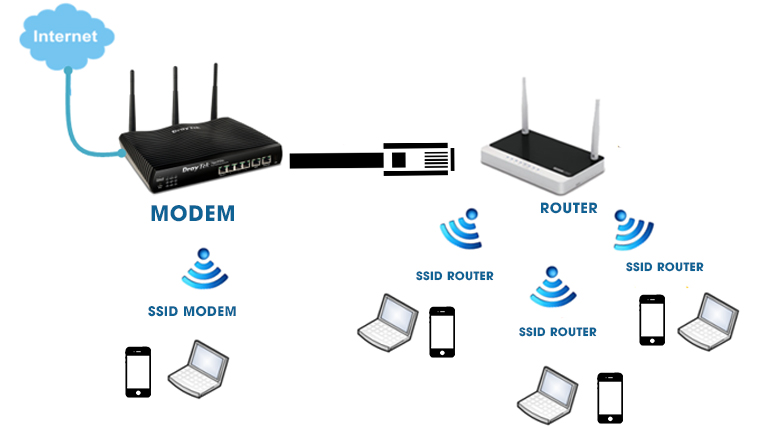
Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ để phân biệt, nó còn được cấu hình là thiết bị đầu hay thiết bị cuối. Thiết bị đầu nắm vai trò tạo xung cho toàn hệ thống, khi giữa hai thiết bị đầu – cuối giao tiếp thì thiết bị đầu có nhiệm vụ tạo xung và quản lý địa chỉ của thiết bị cuối trong suốt quá trình giao tiếp. Thiết bị đầu giữ vai trò chủ động, còn thiết bị cuối giữ vai trò bị động trong việc giao tiếp.

Hình 10 *Chiều của xung Clock và đường dữ liệu*

## Mạng Không Dây Wi-Fi

Wi-Fi (viết tắt từ Wireless Fidelity) hay mạng IEEE 802.11 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống các thiết bị như điện thoại di động, tuyền hình hay radio.

Để tạo được kết nối Wi-Fi nhất thiết phải có bộ định tuyến (Router), thiết bị này lấy thông tin từ mạng Internet qua kết nối hữu tuyến (wired) rồi chuyển nó sang tín hiệu vô tuyến (wireless) và gửi đi, bộ chuyển tín hiệu không dây (Adapter) trên các thiết bị di động thu nhận tín hiệu này rồi giải mã nó sang những dữ liệu cần thiết. Quá trình này có thể thực hiện ngược lại, bộ định tuyến nhận tín hiệu vô tuyến từ bộ chuyển tín hiệu không dây và giải mã chúng rồi gửi qua Internet.

Hình 12 Mô hình mạng Wi-Fi kết hợp IoT ****

Sóng Wi-Fi truyền nhận dữ liệu ở tần số 2.4 GHz đến 5 GHz, cao hơn rất nhiều so với các tần số của điện thoại di động, truyền hình, radio. Tần số cao này cho phép nó mang nhiều dữ liệu hơn nhưng phạm vi truyền của nó bị giới hạn, còn các loại sóng khác tuy tần số thấp nhưng nó có thể truyền được đi rất xa.

Kết nối Wi-Fi sử dụng chuẩn kết nối 802.11 trong thư viện IEEE, chuẩn này bao gồm 4 chuẩn nhỏ là a/b/g/n.

* Chuẩn *802.11b* là phiên bản rẻ nhất và yếu nhất, nó hoạt động ở 2.4 GHz và có thể xử lý đến 11 megabit/giây.
* Chuẩn *802.11g* nhỉnh hơn đôi chút so với chuẩn b, tuy nó cũng hoạt động ở tần số 2.4 GHz nhưng nó có thể xử lý 54 megabit/giây.
* Chuẩn 802.11a phát ở tần số cao hơn là 5 GHz và tốc độ xử lý đạt 54 megabit/giây.
* Chuẩn 802.11n hoạt động ở tần số 2.4 GHz nhưng tốc độ xử lý lên đến 300 megabit/giây.
* Chuẩn 802.11ac sử dụng tần số 5 GHz và tốc độ xử lý tối đa đạt 1730 megabit/giây.