TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG ĐIỆN – ĐIỆN TỬ**

----- □ & □ -----



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**MẠNG CẢM BIẾN KHÔNG DÂY**

**Đề tài: Mạng cảm biến nhiệt độ trong môi trường**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Trần Quang Minh | 20181659 |
| 2 | Nguyễn Văn Dũng | 20191784 |
| 3 | Nguyễn Trọng Phong | 20192017 |

**Hà Nội, tháng 12/2022**

# MỤC LỤC

[MỤC LỤC 2](#_Toc123077318)

[DANH MỤC HÌNH VẼ 4](#_Toc123077319)

[DANH MỤC BẢNG 5](#_Toc123077320)

[CHƯƠNG 1. PHÂN TÍCH YÊU CẦU CỦA DỰ ÁN 6](#_Toc123077321)

[CHƯƠNG 2. KẾ HOẠCH THỰC HIỆN CHUNG 14](#_Toc123077322)

[CHƯƠNG 3. KẾ HOẠCH VÀ NỘI DUNG THỰC HIỆN CỦA TỪNG THÀNH VIÊN 15](#_Toc123077323)

[3.1 Trần Quang Minh 15](#_Toc123077324)

[3.2 Nguyễn Văn Dũng 16](#_Toc123077325)

[3.3 Nguyễn Trọng Phong 17](#_Toc123077326)

[CHƯƠNG 4. TÌM HIỂU CÔNG NGHỆ TRUYỀN THÔNG KHÔNG DÂY 19](#_Toc123077327)

[4.1 Tổng quan về công nghệ truyền thông không dây 19](#_Toc123077328)

[4.1.1 Mô hình truyền thông không dây 19](#_Toc123077329)

[4.1.2 Phân loại mạng không dây 19](#_Toc123077330)

[4.2 WiFi 21](#_Toc123077331)

[4.3 Zigbee 21](#_Toc123077332)

[4.4 Bluetooth 21](#_Toc123077333)

[4.5 Lora 24](#_Toc123077334)

[4.5.1 Giới thiệu về LoRa 24](#_Toc123077335)

[4.5.2 Tỷ lệ mã hóa (Coding Rate) 25](#_Toc123077336)

[4.5.3 Hệ số trải phổ (Spread Factor) 26](#_Toc123077337)

[4.5.4 Băng thông 26](#_Toc123077338)

[CHƯƠNG 5. TÌM HIỂU CÁC DỰ ÁN, NGHIÊN CỨU LIÊN QUAN 29](#_Toc123077339)

[5.1 Hệ thống giám sát nhiệt độ không dây của Therm – X 29](#_Toc123077340)

[5.2 Chức năng hệ thống 29](#_Toc123077341)

[5.3 Các thiết bị trong phần cứng của hệ thống giám sát nhiệt độ không dây 29](#_Toc123077342)

[5.4 Phần mềm giám sát nhiệt độ cho hệ thống 30](#_Toc123077343)

[CHƯƠNG 6. LỰA CHỌN GIẢI PHÁP VÀ LÊN PHƯƠNG ÁN THIẾT KẾ 32](#_Toc123077344)

[6.1 Kiến trúc chung của nút cảm biến không dây 32](#_Toc123077345)

[6.2 Nguồn và vấn đề công suất tiêu thụ của nút cảm biến 32](#_Toc123077346)

[6.2.1 Các chế độ hoạt động của nút cảm biến 33](#_Toc123077347)

[6.2.2 Công suất tiêu thụ của nút cảm biến không dây 34](#_Toc123077348)

[6.3 Sơ đồ kiến trúc hệ thống và chức năng 35](#_Toc123077349)

[6.4 Sơ đồ triển khai chi tiết từng khối 36](#_Toc123077350)

[6.4.1 Sơ đồ triển khai chi tiết Sensor Node 36](#_Toc123077351)

[6.4.1 Sơ đồ triển khai chi tiết Gateway 42](#_Toc123077352)

[CHƯƠNG 7. TÀI LIỆU THAM KHẢO 46](#_Toc123077353)

# DANH MỤC HÌNH VẼ

[*Hình 4‑1 Mô hình truyền thông không dây 19*](#_Toc123077290)

[*Hình 4‑2 Các chuẩn tương ứng với các mạng truyền thông tương ứng 20*](#_Toc123077291)

[*Hình 4‑3 So sánh tốc độ truyền và khoảng cách truyền của Bluetooth, Zigbee và WiFi 22*](#_Toc123077292)

[*Hình 4‑4 Hai loại tín hiệu Up-chirp và Down-chirp trong kĩ thuật CSS 25*](#_Toc123077293)

[*Hình 4‑5 Các giá trị CR và dữ liệu tăng thêm tương ứng (Datasheet SX1278) 26*](#_Toc123077294)

[*Hình 4‑6 Mã hóa một Symbol bằng 2SF chip 26*](#_Toc123077295)

[*Hình 4‑7 Ba mức băng thông thường dùng trong mạng LoRa 27*](#_Toc123077296)

[*Hình 5‑1 Hình ảnh thiết bị giám sát nhiệt độ không dây 29*](#_Toc123077297)

[*Hình 5‑2 Hình ảnh ứng dụng 30*](#_Toc123077298)

[*Hình 6‑1 Kiến trúc chung của nút cảm biến không dây 32*](#_Toc123077299)

[*Hình 6‑2 Công suất tiêu thụ trong các chế độ hoạt động 34*](#_Toc123077300)

[*Hình 6‑3 Sơ đồ khối mô tả hệ thống 36*](#_Toc123077301)

[*Hình 6‑4 Sơ đồ khối sạc pin 37*](#_Toc123077302)

[*Hình 6‑5 Sơ đồ nguyên lý khối nguồn LDO RT9013 38*](#_Toc123077303)

[*Hình 6‑6 Sơ đồ nguyên lý tụ bypass cho STM32 38*](#_Toc123077304)

[*Hình 6‑7 Sơ đồ nguyên lý vi điều khiển STM32 39*](#_Toc123077305)

[*Hình 6‑8 Sơ đồ nguyên lý khối thạch anh ngoài cho vi điều khiển STM32 39*](#_Toc123077306)

[*Hình 6‑9 Khối giám sát điện áp pin LiPo 40*](#_Toc123077307)

[*Hình 6‑10 Sơ đồ nguyên lý module Ra-02 LoRa 40*](#_Toc123077308)

[*Hình 6‑11 Sơ đồ nguyên lý khối debug 41*](#_Toc123077309)

[*Hình 6‑12 Sơ đồ nguyên lý các nút nhấn và LED 41*](#_Toc123077310)

[*Hình 6‑13 Sơ đồ nguyên lý khối hiển thị LED 7 thanh 42*](#_Toc123077311)

[*Hình 6‑14 Sơ đồ nguyên lý khối nguồn LDO 43*](#_Toc123077312)

[*Hình 6‑15 Sơ đồ nguyên lý vi điều khiển ESP32 43*](#_Toc123077313)

[*Hình 6‑16 Sơ đồ nguyên lý các nút nhấn và LED 44*](#_Toc123077314)

[*Hình 6‑17 Sơ đồ nguyên lý khối debug và nạp code 44*](#_Toc123077315)

[*Hình 6‑18 Sơ đồ nguyên lý khối hỗ trợ nạp 45*](#_Toc123077316)

[*Hình 6‑19 Sơ đồ nguyên lý module Ra-02 LoRa 45*](#_Toc123077317)

# DANH MỤC BẢNG

[*Bảng 1‑1 Yêu cầu đặt ra 7*](#_Toc123077275)

[*Bảng 1‑2 Thời gian chuyển đổi ứng với độ phân giải của cảm biến DS18B20 8*](#_Toc123077276)

[*Bảng 1‑3 Thông số của module LoRa sử dụng trong Gateway và Node 9*](#_Toc123077277)

[*Bảng 1‑4 Bảng giá trị SNR 10*](#_Toc123077278)

[*Bảng 1‑5 Phân chia vùng nhớ OTA theo dữ liệu lưu trữ và địa chỉ 13*](#_Toc123077279)

[*Bảng 2‑1 Phân chia công việc chung 14*](#_Toc123077280)

[*Bảng 3‑1 Phân chia công việc của Trần Quang Minh 16*](#_Toc123077281)

[*Bảng 3‑2 Phân chia công việc của Nguyễn Văn Dũng 17*](#_Toc123077282)

[*Bảng 3‑3 Phân chia công việc của Nguyễn Trọng Phong 18*](#_Toc123077283)

[*Bảng 4‑1 So sánh các công nghệ truyền thông không dây 24*](#_Toc123077284)

[*Bảng 6‑1 Các chế độ của thiết bị 33*](#_Toc123077285)

[*Bảng 6‑2 Công suất tiêu thụ của từng phần tử 35*](#_Toc123077286)

# TÌM HIỂU LÝ THUYẾT

## ADC

<https://tapit.vn/chuc-nang-adc-su-dung-vi-dieu-khien-stm32f103c8t6/>

Trong các ứng dụng vi điều khiển – hệ thống nhúng, bộ chuyển đổi tương tự-số (ADC) là 1 thành phần rất quan trọng để có thể chuyển đổi các dữ liệu dạng analog từ môi trường (nhiệt độ, độ ẩm, độ sáng, …) sang dạng digital để vi điều khiển có thể xử lý được.

STM32F103C8 có tích hợp sẵn các bộ chuyển đổi ADC với độ phân giải 12bit. Có 12 kênh cho phép đo tín hiệu từ 10 nguồn bên ngoài và 2 nguồn nội bên trong.

Trong bộ chuyển đổi ADC, có 2 thuật ngữ mà chúng ta cần chú ý đến, đó là độ phân giải (resolution) và thời gian lấy mẫu (sampling time).

* Độ phân giải (resolution): dùng để chỉ số bit cần thiết để chứa hết các mức giá trị số (digital) sau quá trình chuyển đổi ở ngõ ra. Dễ dàng nhận thấy với một bộ chuyển đổi có độ phân giải càng thấp, quá trình chuyển đổi sẽ cho ra kết quả là một điện áp càng biến dạng ở ngõ ra so với ngõ vào và ngược lại. Bộ chuyển đổi ADC của STM32F103 có độ phân giải mặc định là 12 bit, tức là có thể chuyển đổi ra 212 = 4096 giá trị ở ngõ ra số.
* Thời gian lấy mẫu (sampling time): khái niệm được dùng để chỉ thời gian giữa 2 lần số hóa của bộ chuyển đổi.

A diagram of a computer network

Description automatically generated

Vi điều khiển STM32F103 được tích hợp bộ chuyển đổi ADC 12bit loại SAR. Bộ chuyển đổi này cho phép lựa chọn 1 trong 19 kênh đầu vào, trong đó có 16 kênh đầu vào ngoại (là các chân vi điều khiển) và 3 kênh đầu vào nội là điện áp nguồn cung cấp cho vi điều khiển, cảm biến nhiệt độ được tích hợp sẵn của vi điều khiển và một kênh đo điện áp chân V\_BAT. Các kênh đầu vào này có thể chia làm 2 nhóm chuyển đổi: regular (thông thường) hoặc injected (chèn vào khi 1 ADC khác đang chuyển đổi).

**Cấu hình ở CubeMX:**

* Data Alignments: Bộ ADC của STM32F103 có độ phân giải là 12bit mà ta sẽ phải cần lưu trữ vào một thanh ghi 32 bit, do đó sẽ còn thừa 20 bit. Chúng ta sẽ cấu hình việc căn lề cho 12 bits này nằm bên phải hay bên trái trong thanh ghi 32 bits đó.
* Scan Conversion Mode: được sử dụng để “quét” qua lần lượt các kênh ADC trong quá trình đọc dữ liệu.

Nếu sử dụng chế độ đơn kênh, chế độ này sẽ không có tác dụng.

* Continous Conversion Mode: quyết định cho bộ ADC của chúng ta có sử dụng chế độ chuyển đổi liên tục hay không.

Nếu chúng ta không *enable* mode này, sau mỗi lần chuyển đổi, ta sẽ phải gọi lại lệnh đọc giá trị ADC để bắt đầu quá trình chuyển đổi mới.

* Sampling Time: chọn thời gian lấy mẫu trong quá trình số hóa.

Tùy vào ứng dụng mà chúng ta có thể chọn thời gian lấy mẫu cho phù hợp. Thời gian lấy mẫu càng ngắn, việc tái thiết tín hiệu càng chính xác nhưng năng lượng tiêu tốn sẽ càng cao.

* Rank: là thứ tự lấy mẫu của các kênh ADC (đối với chuyển đổi ADC đa kênh), cao nhất là 1 lớn nhất là 10.

Bộ ADC sẽ scan từ kênh có rank là 1 đến n với n là số kênh các bạn cần chuyển đổi.

* NVIC Settings:

**Các chế độ chuyển đổi:**

* Chế độ đơn kênh, sử dụng interrupt để báo hiệu quá trình chuyển đổi hoàn tất.

<https://tapit.vn/chuc-nang-adc-su-dung-vi-dieu-khien-stm32f103c8t6/>

<https://tapit.vn/hieu-va-doc-gia-tri-adc-stm32f411-bo-chuyen-doi-tin-hieu-tuong-tu/> (xem ở phần single channel, single conversion mode)

* Chế độ nhiều kênh, chuyển đổi 1 lần - Multi-channel (scan), single conversion: Ở chế độ này, bộ ADC sẽ lần lượt chuyển đổi một chuỗi các kênh đầu vào đã được cấu hình từ trước, tối đa có thể chuyển đổi được 16 kênh. kết quả chuyển đổi của kênh cuối cùng sẽ được lưu vào thanh ghi ADC\_DR, cờ báo kết thúc quá trình chuyển được được bật lên, yêu cầu ngắt sẽ được sinh ra nếu người dùng cấu hình cho phép ngắt.

Chế độ này cần được kết hợp với tính năng DMA để có thể nhận đầy đủ kết quả chuyển đổi của các kênh.

Thứ tự chuyển đổi của các kênh có thể được người dùng sắp xếp và thời gian lấy mẫu của các kênh có thể được cấu hình độc lập.

* Chế độ đơn kênh, chuyển đổi liên tục – Single-channel, continous conversion mode: Ở chế độ này kênh được chọn sẽ được chuyển đổi liên tục mà không cần sự can thiệp của CPU trong mỗi lần chuyển đổi. Chức năng DMA với mode circular nên được sử dụng trong chế độ này để giảm tải cho CPU.
* Chế độ đa kênh, chuyển đổi liên tục – Multi-channel (scan), continous conversion mode: Ở chế độ này các kênh được chọn sẽ được chuyển đổi liên tục mà không cần sự can thiệp của CPU trong mỗi lần chuyển đổi. Chức năng DMA với mode circular kết hợp với việc tăng bộ nhớ tự động nên được sử dụng trong chế độ này để giảm tải cho CPU.

<https://khuenguyencreator.com/lap-trinh-stm32-adc-nhieu-kenh-su-dung-dma/>

A diagram of a program

Description automatically generated

**Tính năng DMA (Direction Memory Access):** hỗ trợ ngoại vi có thể chuyển dữ liệu trực tiếp đến bộ nhớ mà không cần sự can thiệp của CPU.

<https://www.youtube.com/watch?v=MYWiCRR2oHc&ab_channel=HuyB%C3%B9iV%C4%83n>

<https://www.youtube.com/watch?v=MMqaF6qgrP0&ab_channel=WonyeobChrisPark>

**Cấu hình DMA:**

* DMA Settings:
* Mục DMA Request thêm kênh ADC1 các mục còn lại sẽ tự động được cài đặt theo.
* Mode: Circular với mode này dữ liệu sẽ liên tục được gửi qua lại giữa ADC và Vùng nhớ sử dụng DMA
* Data Width: Half Word 16 bit

**Ứng dụng trong project:**

<http://www.efton.sk/STM32/STM32_VREF.pdf>

<https://stackoverflow.com/questions/58328342/calibrating-stm32-adc-vrefint>

<https://stackoverflow.com/questions/57752689/how-to-use-the-vrefint-in-stm32f103-bluepill>

Từ những khó khăn được nêu ra trong tài liệu STM32\_VREF, trong dự án này, để đo được điện áp của pin, em sử dụng chế độ đa kênh, chuyển đổi liên tục kết hợp tính năng DMA đối với bộ chuyển đổi ADC. Cụ thể:

* Chế độ đa kênh: kênh ADC1\_IN1 và kênh Vrefint.
* Cấu hình ADC: chế độ đa kênh, chuyển đổi liên tục.

A screenshot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
| ADC1 Clock | 12MHz | 12MHz |
| Channel | ADC1\_IN1 | ADC1\_Vrefint |
| Thời gian lấy mẫu | 239.5 cycles ≈ 20 µs | 239.5 cycles ≈ 20 µs |

* Cấu hình DMA: giống như cấu hình trên, tuy nhiên ở mục “Data Width” chọn thông số “Word” tương ứng 32 bits.

## LoRa

### Cơ bản về giao thức SPI

<https://khuenguyencreator.com/lap-trinh-stm32-voi-giao-thuc-spi/#Che_do_hoat_dong_cua_giao_thuc_SPI>

<https://tapit.vn/chuan-giao-tiep-spi-tren-stm32f4/>

<http://www.hocavr.com/2018/06/bai-7-giao-tiep-spi.html>

A screenshot of a computer

Description automatically generated

### Triển khai hardware

Module SX1278 sẽ được cấp nguồn 3.3V được lọc qua tụ . Vi điều khiển sử dụng truyền thông SPI để giao tiếp với module với Baud Rate là 8 Mbit/s, 2 chân LORA\_RST và LORA\_IO0 được sử dụng để reset và gửi tín hiệu ngắt đến vi điều khiển.

Diagram

Description automatically generated with medium confidence

LORA\_RST: chân này dùng cho manual reset.

A screenshot of a computer

Description automatically generated

### Cấu hình ở CubeMX

SPI2 Mode and Configuration:

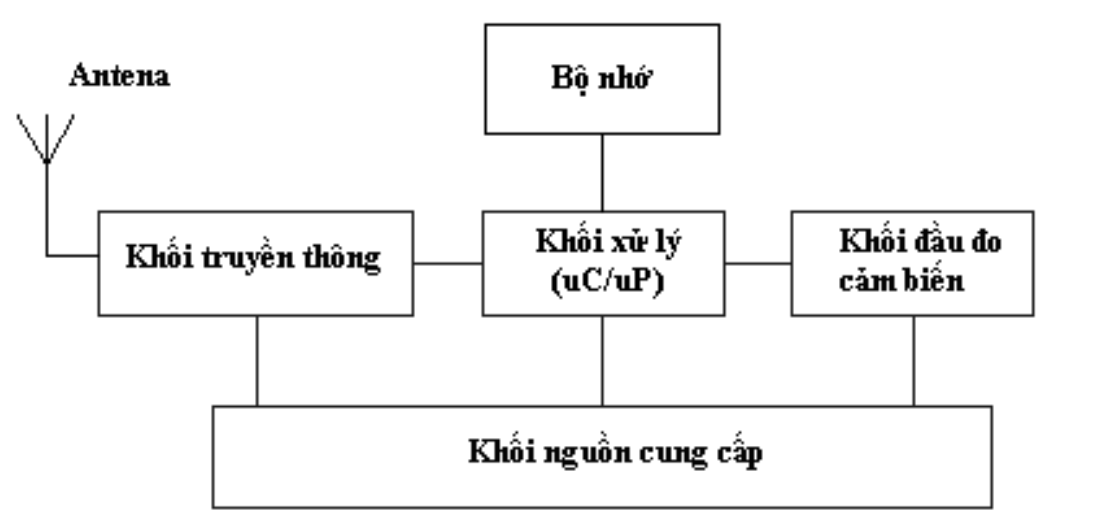
* Mode: Full-Duplex Master
* Hardware NSS Signal: Disable (chân CS em sẽ sử dụng chân GPIO khác để dễ dàng điều khiển nếu nhiều slave)
* Chân PA8: cấu hình cho chân CS của SPI2

PA15: nối với chân LORA\_RST

PB12: nối với chân LORA\_DI0

# LỰA CHỌN GIẢI PHÁP VÀ LÊN PHƯƠNG ÁN THIẾT KẾ

## Kiến trúc chung của nút cảm biến không dây



Hình 6‑1 Kiến trúc chung của nút cảm biến không dây

Một nút cảm biến không dây: bản chất là một thiết bị đo mang tính nhỏ gọn, tiêu thụ năng lượng thấp, dễ di chuyển và lắp đặt, sử dụng công nghệ truyền tin không dây.

Một nút cảm biến không dây bao gồm các khối:

1. Khối nguồn cung cấp
2. Khối đầu đo cảm biến/cơ cấu chấp hành
3. Khối điều khiển
4. Bộ nhớ
5. Khối truyền thông không dây

## Nguồn và vấn đề công suất tiêu thụ của nút cảm biến

Để đảm bảo công suất tiêu thụ thấp khi thiết kế, nhóm sẽ phân tích về các vấn đê tiêu thụ năng lượng của từng phần tử trong cảm biến và mạng.

Năng lượng tiêu thụ của mạng được tính bằng thời gian cảm biến truyền hay nhận hay ngủ.

Năng lượng tiêu thụ của cảm biến gồm hai phần:

1. Năng lượng của mỗi linh kiện cấu tạo nên nút cảm biến
2. Năng lượng truyền/nhận sóng khi các nút cảm biến trong mạng trao đổi thông tin với nhau.

### Các chế độ hoạt động của nút cảm biến

Một nút cảm biến có 3 chế độ hoạt động:

1. Chế độ hoạt động tích cực (Active Mode)
2. Chế độ ngủ (Sleep mode)
3. Chế độ nghỉ (Idle)

|  |  |
| --- | --- |
| Chế độ hoạt động tích cực (Active Mode) | Nút cảm biến thực hiện các nhiệm vụ như đo nhiệt độ, truyền thông tin đi hay nhận thông tin về với khối Gateway |
| Ở chế độ này, khối xử lý trung tâm (MCU) của nút cảm biến luôn luôn hoạt động và gần như tất cả các khối còn lại cũng đều hoạt động: bộ truyền nhận sóng LoRa, bộ nhớ. Các khối đồng thời cùng hoạt động (do thiết kế firmware sử dụng hệ điều hành thời gian thực RTOS) |
| Tối ưu công suất tiêu thụ thực hiện ở giai đoạn thiết kế firmware |
| Công suất tiêu thụ khi nút cảm biến hoạt động ở chế độ tích cực là |
| Chế độ ngủ  (Sleep Mode) | Nút cảm biến chưa phải thực hiện một nhiệm vụ cảm biến hay truyền thông nào, tuy nhiên nút cảm biến vẫn tham gia hoạt động trong mạng.  Sau *một khoảng thời gian nghỉ t*, nút sẽ hoạt động trở lại |
| Ở chế độ này, khối xử lý trung tâm (MCU) của nút cảm biến luôn ở trạng thái ngủ. Nhóm chúng em sẽ tính toán thời gian và năng lượng sử dụng để thực hiện việc chuyển đổi giữa các trạng thái với nhau ở phía dưới. |
| Công suất tiêu thụ khi nút cảm biến hoạt động ở chế độ ngủ là |
|  |
| Chế độ nghỉ  (Idle Mode) | Nút cảm biến chưa phải thực hiện một nhiệm vụ cảm biến hay truyền thông nào và không tham gia hoạt động trong mạng. |
| Ở chế độ này, khối xử lý trung tâm (MCU) của nút cảm biến luôn ở trạng thái ngủ. Các khối khác cũng ở chế độ ngủ. |

Bảng 6‑1 Các chế độ của thiết bị

Ở mục này, sau khi thiết kế phần cứng xong nhóm chúng em sẽ thực hiện thử nghiệm với hai trường hợp:

Khi nút cảm biến đang hoạt động tích cực (active), tại thời điểm t1 là thời điểm đưa ra quyết định (event): nút cảm biến vẫn hoạt động tích cực hay chuyển sang trạng thái ngủ (sleep) để giảm công suất tiêu thụ từ Pactive sang Psleep

* Trường hợp 1: nếu nút cảm biến đưa ra quyết định duy trì trạng thái hoạt động tích cực và trạng thái ấy kéo dài đến thời điểm tevent thì toàn bộ năng lượng tiêu thụ trong trường hợp không sử dụng trạng thái nghỉ được tính bằng:
* Trường hợp 2: nếu nút cảm biến đưa ra quyết định rơi vào trạng thái ngủ, và phải mất một khoảng thời gian tdown thì cảm biến mới chuyển hoàn toàn từ trạng thái Pactive xuống Psleep

Diagram

Description automatically generated

Hình 6‑2 Công suất tiêu thụ trong các chế độ hoạt động

### Công suất tiêu thụ của nút cảm biến không dây

Nút cảm biến không dây:

* MCU: STM32F103C8T6
* Vcc: 3.3V
* Tần số hoạt động tối đa: 72MHz
* Cảm biến: DS18B20
* Khối truyền thông: LoRa SX1278
* Khối hiển thị: 4 LED 7 thanh.

Từ các thông số thiết kế nút cảm biến ở trên, nhóm chúng em đề xuất bảng công suất tiêu thụ của nút cảm biến không dây:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linh kiện | Chế độ | Điện áp hoạt động  (V) | Dòng điện tiêu thụ  (mA) | Công suất tiêu thụ  (mW) |
| DS18B20 | Active | 3.3 | 1.5 | 0.495 |
| MCU (STM32) | Active | 3.3 | 50.0 | 165.0 |
| Sleep | 3.3 | 8.0 | 26.4 |
| SX1278 | Standby | 3.3 | 1.6 | 5.28 |
| Tx | 3.3 | 120.0 | 396.0 |
| Rx | 3.3 | 12.0 | 39.6 |
| Sleep | 3.3 | 0.0002 | 0.00066 |
| LED 7 SEG |  | 3.3 | 80.0 | 264.0 |

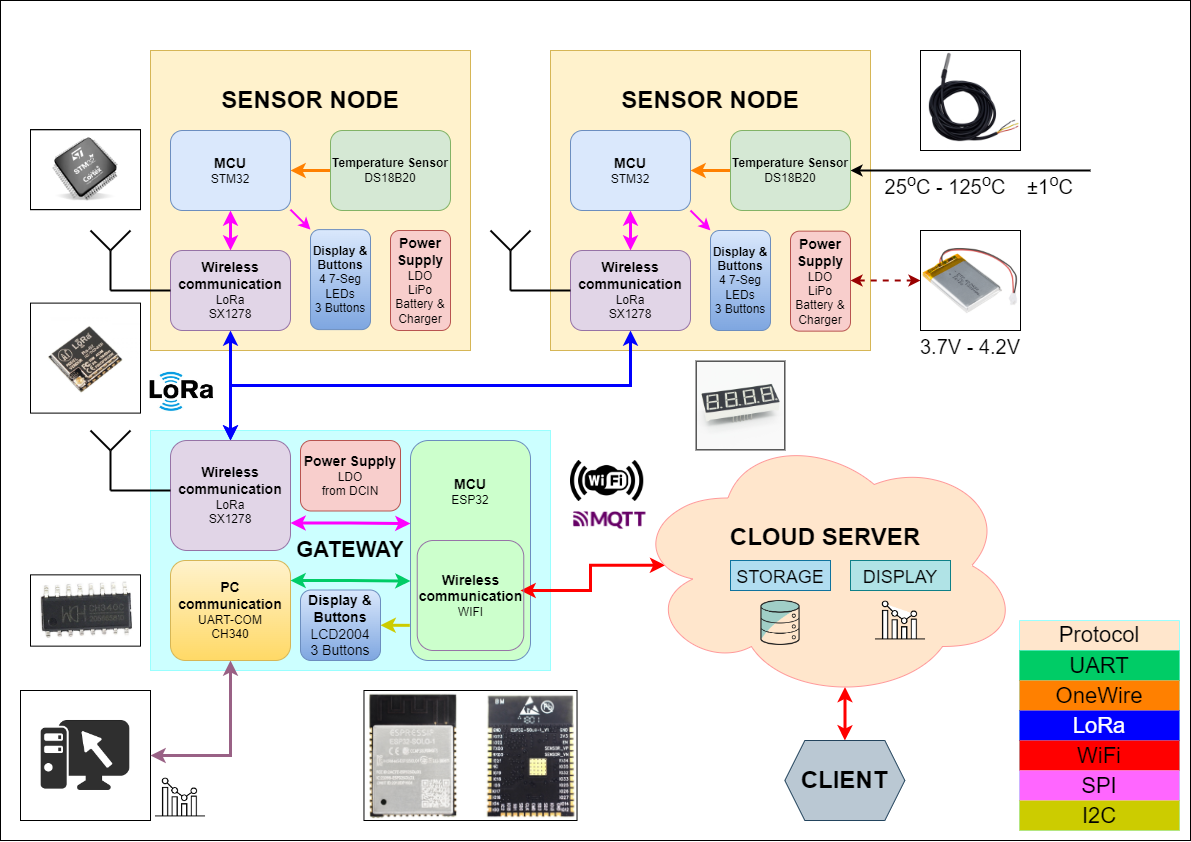
Bảng 6‑2 Công suất tiêu thụ của từng phần tử

Từ bảng trên, công suất tiêu thụ bởi nút cảm biến không dây ở chế độ Sleep là 26.9 mW, ở chế độ Active với module truyền thông ở mode Tx là 1617.5 mW, ở chế độ Active với module truyền thông ở mode Rx là 1261.1 mW.

Trong dự án này, nhóm em chọn thời gian lấy mẫu là 5s, số lượng Sensor Node quản lý là 10. Khi bắt đầu đo, tất cả module SX1278 của các Sensor Node sẽ ở chế độ Rx. Sensor Node nào được Gateway yêu cầu thì sau khi phản hổi lại dữ liệu sẽ vào chế độ Sleep trong 35s. Sau 35s thì module sẽ được chuyển sang chế độ Rx và đợi yêu cầu từ Gateway. Trong khoảng thời gian 15s tiếp theo sẽ có khoảng 0.3s module chuyển sang chế độ Tx để gửi dữ liệu. Tổng kết lại trong 50s thì module SX1278 ở chế độ Sleep 35s, chế độ Rx 14.7s, chế độ Tx 0.3s.

Coi các thành phần còn lại ở chế độ Active, 7 thanh sáng toàn bộ tính được năng lượng tiêu thụ của nút cảm biến trong một giờ là:

## Sơ đồ kiến trúc hệ thống và chức năng



Hình 6‑3 Sơ đồ khối mô tả hệ thống

Sơ đồ khối mô tả một cách khái quát về mối liên kết giữa các thành phần của hệ thống, chúng được tách ra làm 3 phần riêng biệt:

* Khối Gateway: nhận dữ liệu nhiệt độ từ các Sensor Node và đẩy dữ liệu đó lên Cloud Server thông qua giao thức MQTT.
* Khối Sensor Node: giám sát và thu thập dữ liệu nhiệt độ và gửi dữ liệu đó về Gateway để đánh giá thông qua công nghệ truyền thông không dây LoRa.
* Khối Cloud Server: quản lý và hiển thị dữ liệu theo giao diện thống nhất.

Nhóm chúng em sẽ đi vào demo sơ đồ triển khai chi tiết từng khối phía dưới và phân tích rõ chức năng từng khối trong hệ thống này.

## Sơ đồ triển khai chi tiết từng khối

### Sơ đồ triển khai chi tiết Sensor Node

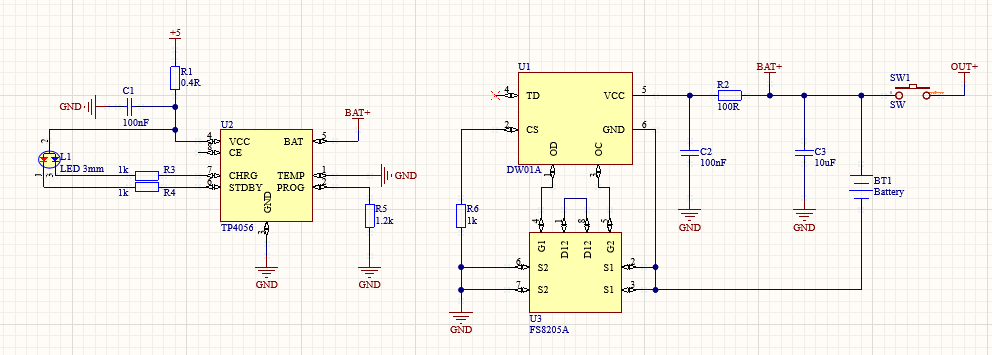
#### Khối nguồn

Với yêu cầu về khối nguồn của dự án, nhóm em sử dụng nguồn từ Pin Lithium và đồng thời thiết kế mạch sạc để có thể sạc trực tiếp, điện áp đầu vào để sạc cho Pin Lithium là 5V. Nguồn sạc cho Pin được cung cấp bởi IC TP4056, đây là loại IC chuyên dụng để sạc cho 1 cell Pin Lithium, dòng sạc tối đa là 1A. Đồng thời IC TP4056 có thể tự động ngừng sạc khi dòng sạc nhỏ hơn 100mA. Để bảo vệ Pin khi điện áp Pin bị sụt quá mức cho phép làm giảm tuổi thọ của Pin, nhóm em sử dụng IC DW01A và IC FS8205A. Bản thân IC FS8205A là 2 mosfet kênh P có cổng Drain nối với nhau, khi điện áp Pin sụt xuống còn 3V, chân OD của IC FS8205A nối với cổng Gate mosfet sẽ xuất ra logic 0 để ngắt Pin khỏi tải, nhờ đó Pin sẽ được bảo vệ.

Trong sơ đồ hình 6 – 1, = 0.4và = 100nF được dùng để lọc nhiễu điện áp đầu vào. Theo datasheet thì chân PROG của TP4056 dùng để chọn dòng sạc cho pin theo công thức:

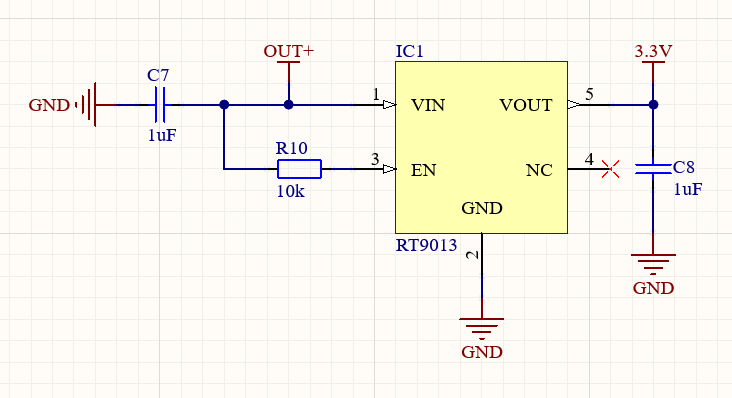
trong đó V nên với dòng sạc 1A thì = 1.2k.

Hai chân CHRG và STDBY có vai trò cho biết trạng thái sạc của pin. Khi ngừng sạc thì chân STDBY có mức logic 0, chân CHRG float còn trong quá trình sạc thì ngược lại. Vậy nên nhóm em sử dụng LED L1 có 2 màu để hiển thị trạng thái sạc, điện trở và dùng để hạn dòng cho LED. Đầu ra của IC TP4056 sẽ được lọc qua tụ = 10uF và đến cực dương của Pin.



Hình 6‑4 Sơ đồ khối sạc pin

Với điện áp Pin từ 3.7 – 4.2V, nhóm em sử dụng IC LDO RT9013 để hạ áp xuống 3.3V để cấp nguồn cho các khối khác. IC RT9013 cho dòng cung cấp tối đa là 0.5A khi tối thiểu 250mV. Tụ và dùng để lọc nhiễu, chân EN của IC RT9013 sẽ được kéo trở cao để luôn cho phép hoạt động. Ở Sensor Node sẽ có 2 IC RT9013 để có được dòng 1A đảm bảo cung cấp nguồn ổn định.



Hình 6‑5 Sơ đồ nguyên lý khối nguồn LDO RT9013

#### Khối xử lý trung tâm

Điện áp IC RT9013 sẽ được lọc qua các tụ bypass theo yêu cầu trong datasheet của STM32 trước khi cấp nguồn.

Diagram, schematic

Description automatically generated

Hình 6‑6 Sơ đồ nguyên lý tụ bypass cho STM32

Như đã nói trong phần 6.3, ở Sensor Node nhóm em sử dụng vi điều khiển là STM32F103C8T6 có tần số thạch anh 72MHz, bộ nhớ Flash 64KB, SRAM 20KB. Để vi điều khiển có thể nạp bằng mạch nạp ST – Link, chân BOOT0 sẽ được kéo xuống GND qua trở 100k để chọn chế độ Main flash memory. Chân SENSOR\_DATA là chân để giao tiếp với cảm biến DS18B20. Các chân DSIN, CS, CLK để giao tiếp với IC MAX7219 để điều khiển 4 LED 7 thanh.

Diagram, schematic

Description automatically generated

Hình 6‑7 Sơ đồ nguyên lý vi điều khiển STM32

Mặc dù bản thân MCU STM32F103C8T6 đã được hãng trang bị bộ tạo dao động nội HSI lên đến 8MHz tuy nhiên bộ dao động này có độ chính xác không cao, HSI tốc độ chỉ có 8MHz không đáp ứng được yêu cầu hệ thống. Thạch anh HSE 8MHz được mắc vào 2 chân OSC\_IN và OSC\_OUT qua bộ PLL có thể nhân tần số hoạt động của chip lên tối đa 72Mhz. Các tụ C18 và C20 được thêm vào để giúp thạch anh hoạt động ổn định và chính xác.

Diagram

Description automatically generated with medium confidence

Hình 6‑8 Sơ đồ nguyên lý khối thạch anh ngoài cho vi điều khiển STM32

Do điện áp của Pin có thể lên đến 4.2V vượt quá dải điện áp cho phép từ 0 – 3.3V của STM32 nên cần được phân áp trước khi đến chân ADC của STM32. Ở đây em chọn 2 điện trở là k và 22k, khi 4.2V thì V nằm trong dải điện áp cho phép.

Diagram, schematic

Description automatically generated

Hình 6‑9 Khối giám sát điện áp pin LiPo

Module SX1278 sẽ được cấp nguồn 3.3V được lọc qua tụ . Vi điều khiển sử dụng truyền thông SPI để giao tiếp với module với Baud Rate là 8Mbit/s, 2 chân LORA\_NSS và LORA\_IO0 được sử dụng để Reset và gửi tín hiệu ngắt đến vi điều khiển.

Diagram

Description automatically generated with medium confidence

Hình 6‑10 Sơ đồ nguyên lý module Ra-02 LoRa

Ngoài ra nhóm em thiết kế thêm khối Debug để in ra dữ liệu cần thiết, thuận tiện khi lập trình và kiểm tra lỗi. USB Micro dùng để chuyển tiếp dữ liệu tới máy tính đồng thời cấp nguồn sạc cho Pin. Hai chân D\_N và D\_P sẽ được nối với chân UD+ và UD- ở IC CH340 , IC này sẽ chuyển đổi dữ liệu sang UART để giao tiếp với vi điều khiển.

Diagram

Description automatically generated

Hình 6‑11 Sơ đồ nguyên lý khối debug

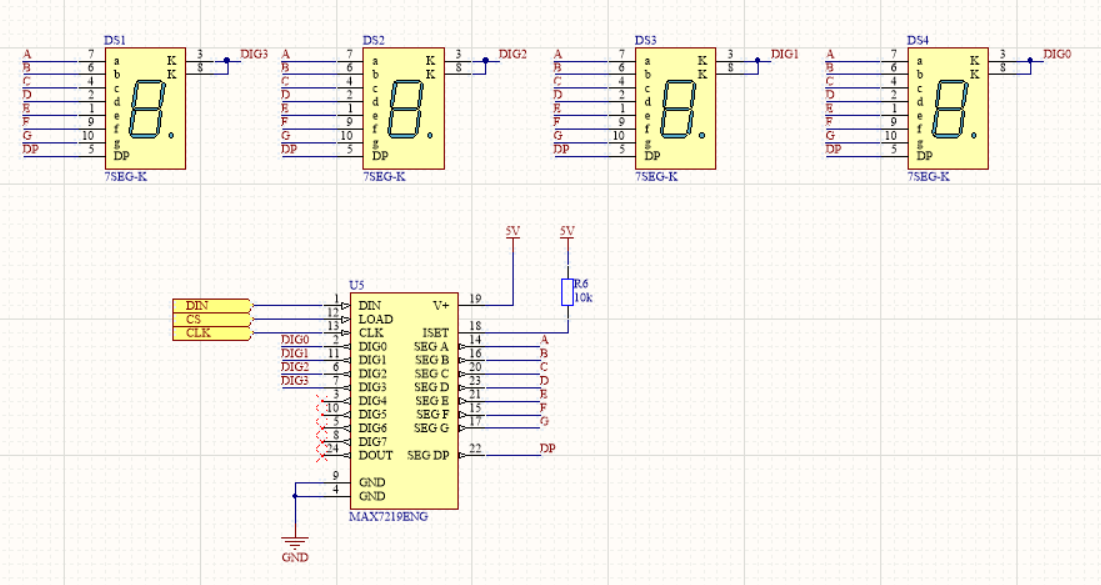
Theo như yêu cầu của dự án, ở Sensor Node sẽ có 3 LED cảnh báo ngưỡng và một nút nhấn chức năng. Vi điều khiển sẽ dùng ngắt ngoài để bắt được xung cạnh xuống khi nhấn nút nhấn để tiến hành hoạt động theo từng chức năng của nút nhấn.

A picture containing chart

Description automatically generated

Hình 6‑12 Sơ đồ nguyên lý các nút nhấn và LED

#### Khối hiển thị



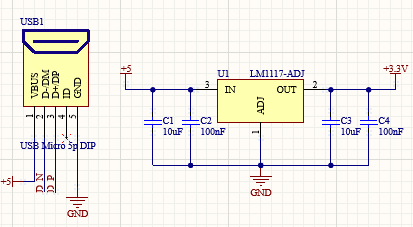
Hình 6‑13 Sơ đồ nguyên lý khối hiển thị LED 7 thanh

Các LED 7 thanh được sử dụng là loại Katot chung, được điều khiển qua LED Driver MAX7219. IC MAX7219 giao tiếp với VĐK qua giao thức SPI. MAX7219 là Driver LED 7 thanh được thiết kế cho tối đa 8 số mỗi IC và có khả năng mở rộng bằng cách nối nối tiếp nhiều IC qua nhau. Trong ứng dụng này chúng em chỉ sử dụng 4 LED 7 thanh nên các cực Katot chung của các LED được nối vào các chân DIG0-3. Các tín hiệu của từng thanh trong các LED được nối song song với nhau và nối vào các chân SEGA-G và SEFDP. Ngoài ra, độ sáng của các thanh còn có thể được tùy chỉnh thông qua trở R6 quy định dòng qua các thanh. Nguồn điện áp pin được cấp trực tiếp cho các LED.

### Sơ đồ triển khai chi tiết Gateway

#### Khối nguồn

Gateway được cung cấp nguồn đầu vào 5V được lọc nhiễu qua tụ và , để hạ áp xuống 3.3V cấp nguồn cho khối xử lý trung tâm thì nhóm em sẽ sử dụng IC LDO LM1117 có khả năng cung cấp dòng điện lên đến 1A đủ đáp ứng cho các khối còn lại.



Hình 6‑14 Sơ đồ nguyên lý khối nguồn LDO

#### Khối xử lý trung tâm

Ở Gateway, nhóm em sử dụng vi điều khiển ESP32. Đây là dòng vi điều khiển hiệu năng rất tốt, tần số hoạt động lên tới 240MHz. Đồng thời hãng đã trang bị cho nó bộ nhớ Flash 4M, SRAM 520KB thích hợp để sử dụng hệ điều hành FreeRTOS. Vi điều khiển được cấp nguồn 3.3V được lọc nhiễu qua 2 tụ và theo yêu cầu trong datasheet. Hai chân LCD\_SDA và LCD\_SCK được dùng để giao tiếp với màn hình LCD 1602. Để Gateway đảm bảo kết nối với WiFi nhất thì có thể sử dụng antenna rời, không sử dụng antenna on – chip.

Diagram, schematic

Description automatically generated

Hình 6‑15 Sơ đồ nguyên lý vi điều khiển ESP32

Như đã đề cập ở yêu cầu 8, Gateway sẽ có 3 nút nhấn chức năng để thuận tiện thao tác vật lý, đồng thời có 2 LED để chỉ thị trạng thái hoạt động.

A picture containing table

Description automatically generated

Hình 6‑16 Sơ đồ nguyên lý các nút nhấn và LED

Để phục vụ quá trình nạp firmware và debug, một khối chuyển đổi UART sang USB và một cổng MicroUSB được sử dụng giống như ở Sensor Node.

Diagram

Description automatically generated

Hình 6‑17 Sơ đồ nguyên lý khối debug và nạp code

Khối phụ trợ cho khối nạp firmware, có nhiệm vụ chính là đưa module ESP32 vào chế độ boot bằng cách lần lượt kéo chân EN và GPIO0 xuống GND. Hai nút bấm để reset module và để kéo GPIO0 xuống GND trong trường hợp khối nạp firmware hoạt động không đúng mong đợi.

Diagram, schematic

Description automatically generated

Hình 6‑18 Sơ đồ nguyên lý khối hỗ trợ nạp

Module SX1278 vẫn sẽ được sử dụng để truyền thông với Sensor Node, các chân được sử dụng giống với thiết kế ở Sensor Node.

Diagram

Description automatically generated with medium confidence

Hình 6‑19 Sơ đồ nguyên lý module Ra-02 LoRa

# TÀI LIỆU THAM KHẢO