1. Sensor

Các yêu cầu

* Với dải đo yêu cầu 25oC – 125oC
* Độ chính xác 1oC
* Độ phân giải hiển thị : 0.1oC

Hiện nay thị trường có nhiều loại cảm biến đo nhiệt độ, có các loại loại cơ bản : Cảm biến nhiệt độ can nhiệt ( Thermocouple), Cảm biến nhiệt độ điện trở ( Phổ biến như Pt100,…), Thermistor, cảm biến nhiệt độ Bán dẫn, Cảm biến nhiệt độ Bức xạ, …

Dựa trên yêu cầu của bài toán và sự phổ biến của loại cảm biến trong đo nhiệt độ của môi trường trong nhà, đảm bảo đáp ứng được thông số đề bài, chúng em dự định sử dụng cảm biến nhiệt độ bán dẫn và cụ thể là cảm biến LM35 của hãng Texas Instruments

Là cảm biến có dải đo phù hợp cho nhiệt độ môi trường ( -55 oC – 150 oC), độ chính xác là 0.5 oC nhưng sử dụng dòng thấp ( ít hơn 60 uA) phù hợp cho các thiết không dây, và có dải nhiệt độ tuyến tính, cứ tăng 1 oC thì tương ứng dòng đầu ra tăng 10.0mV

2. Battery Life

Trong dự án, các mạch phần cứng sẽ được lập trình chế độ Sleep Mode-chế độ tiết kiệm năng lượng cả node sensor và Gateway.

Chúng em sử dụng mode Deep Sleep. Lúc này CPU, RAM và các ngoại vi đều bị tắt, còn bộ RTC, ngoại vi RTC (bao gồm bộ ULP) và bộ nhớ RTC vẫn hoạt động.

CPU core bị tắt còn bộ ULP thực hiện các phép đo cảm biến và được đánh thức thông qua:

* Timer: định thời gian wake up theo chu kì
* Touch Pad: sử dụng các chân cảm biến
* External wake up: sử dụng 2 ngắt ngoài EXT0 và EXT1
* ULP Processor

Trong dự án chúng em sử dụng 2 cách để wake up là Timer và ngắt ngoài. Với Timer thì hệ thống sẽ ngắt theo chu kì được định sẵn, còn với ngắt thì khi trong quá trình người dùng muốn xem thông số cảm biến khi hệ thống đang sleep thì có thể wake up.

Trong mode này ESP32 tiêu thụ từ 10uA đến 0.15mA. Còn mode active khi hoạt động bình thường sử dụng các ngoại vi và WiFi, dòng tiêu thụ lên tới 500mA.

Nếu setup 80% thời gian trong 1 chu kì hệ thống sleep và dựa theo công thức tính dòng trung bình là:

Graphical user interface, text, email

Description automatically generated

Tính được consumption ~ 100mA, từ đó tính được thời gian sử dụng Pin với loại Pin lithium 2200mAh là:

Graphical user interface, text

Description automatically generated

Battery Life = 13.2h.

* Đáp ứng được yêu cầu.
* Sạc Pin được thực hiện trên thiết bị.

3 - Kích thước – Trọng lượng sản phẩm

Các yêu cầu

* 70x50x100 mm
* Trọng lượng <150g

Với thiết bị có kích thước hình trụ, dễ dàng di chuyển và lắp đặt, đảm bảo độ thẩm mỹ cho thiết bị, nhóm dự tính sử dụng các linh kiện để phù hợp với kích thước đó

Dự tính một số thiết bị chiếm diện tích như :

* MCU sử dụng ESP32 DEVKIT-V1 kích thước 55 x 28 x 10 mm
* Pin Lithium 3.7V 2000mA kích thước 18 x 65 mm
* Cảm biến LM35 TO-92 Plastic Package kích thước với chiều cao khoảng 20mm, kích thước đầu bán dẫn khoảng 5 x 4 mm
* LM1117 - Ổn áp 3.3V cấp cho MCU, điện trở, tụ điện, …

=>Với những linh kiện đó, nhóm dự kiến kích thước sẽ đáp ứng được bài toán kích thước

Về cân nặng, nhóm sử dụng 01 Lithium cấp nguồn cho mạch là yếu tố chiếm phần lớn cân nặng của mạch ( ~45g) , ngoài ra một số yếu tố chứa cân nặng như vỏ hộp, kit esp32,… tuy nhiên với cân nặng tiêu chuẩn <150g

=>Nhóm đáp ứng được yêu cầu cân nặng

4 - Thời gian đo một chu kỳ đo

Yêu cầu : <20s

Project sử dụng ESP32 có xung nhịp lên tới 240 MHz, xung nhịp max tương đối cao, và chúng em dự định sử dụng ESP IDF để lập trình, có hỗ trợ RTOS, thông qua việc đọc giá trị điện áp từ LM35 và gửi bản tin cho thiết bị GateWay để push giá trị của Node lên web, vì vậy nhóm dự định thời gian lấy mẫu của mỗi con Node và gửi bản tin là không lớn, và thời gian gửi từ Gateway lên web.

=>Sẽ khống chế trong thời gian 20s với tương ứng ít nhất 10 Node đo và trao đổi bản tin với Gateway.

5. Kết nối máy tính

RF sử dụng wifi với phạm vi tỏa sóng ở trong bán kính 100m và hoạt động tốt nhất ở khoảng cách nhỏ hơn 20m với điều kiện môi trường ít vật cản và ít chịu ảnh hưởng của các điều kiện môi trường xung quanh

=> Khoảng cách truyền trong phạm vi 20m từ hệ thống đo đến trạm thu RF có nối nguồn và mạng có thể đáp ứng được

6. Quản lý thiết bị

Tối thiểu 10 thiết bị đo với phương án mở rộng số thiết bị với khoảng cách từ trạm tiếp nhận có thể lên tới 100m

7. Phần mềm máy tính

+ Thu thập giá trị đo từ thiết bị đo, quản lý dữ liệu

+ Hiển thị giá trị nhiệt độ hiện tại đo được

+ Hiển thị version của firmware, nếu có version mới thì có hiển thị nút download

+ Có thể xuất data ra file excel hoặc file text hoặc đồ thị

+ Có hiển thị nút nhấn bắt đầu đo trên phần mềm

+ Có thể cập nhật lại các ngưỡng nhiệt độ

8. Nút nhấn và led cảnh báo

+ Có nút bấm bắt đầu đo

+ Có nút bấm Reset

+ 3 đèn LED cảnh báo nhiệt độ cho các ngưỡng nhiệt độ đo được, các ngưỡng có thể được cập nhật từ máy tính

9. OTA (Over The Air)

Để có thể đưa sản phẩm ra thương mại thì sẽ gặp rất nhiều vẫn đề, bởi vì các sản phẩm khi release chưa chắc đã hoàn hảo, cần phải update thêm tính năng hoặc sửa lỗi. Vậy nên chúng em sẽ ứng dụng OTA vào Project để có thể lưu trữ, quản lý và cập nhật version của sản phẩm.

Trong Project chúng em sẽ sử dụng Server của OTA Drive, một website quản lý firmware miễn phí.

Khi cập nhật hệ thống sẽ tuân theo các bước:

* Bước 1: Chúng em sẽ upload firmware lên OTA Server.
* Bước 2: Người dùng sẽ thấy version mới và nhấn download lúc này MCU sẽ gọi 1 API lên Server đó, truy vấn đến bản cập nhật mới.
* Bước 3: Nếu nhận được thông tin thay đổi version, MCU sẽ download firmware về thông qua internet.
* Bước 4: MCU sẽ khởi động lại với firmware mới