

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM

KHOA: ĐIỆN – ĐIỆN TỬ

-----oOo-----



HCMUTE

ĐỒ ÁN 1

Đề tài: CẢNH BÁO KHÍ GAS

GVHD: THS. HUỖNH HOÀNG HÀ

SVTH: LÊ XUÂN ĐỈNH

20119335

TRẦN MẠNH CƯỜNG

20119326

Tp. Hồ Chí Minh, 06/2023

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên thực hiện: LÊ XUÂN ĐÌNH – 20119335

TRẦN MẠNH CƯỜNG – 20119326

Ngành: CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Tên đề tài: THIẾT BỊ CẢNH BÁO RÒ RỈ KHÍ GAS

Họ và tên giảng viên hướng dẫn: ThS. HUỖNH HOÀNG HÀ

NHẬN XÉT

1. Về nội dung đề tài và khối lượng thực hiện:

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

3. Khuyết điểm:

.....

.....

.....

4. Điểm:.....(Bằng chữ:.....)

Tp. Hồ Chí Minh, ngày tháng năm 2023

Giảng viên hướng dẫn

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Kính gửi Thầy Huỳnh Hoàng Hà,

Nhóm em xin chân thành cảm ơn thầy vì những bài giảng cùng những lời đóng góp ý kiến, chỉ bảo nhóm em trong quá trình thực hiện đồ án 1 với đề tài :” Cảnh báo khí gas” này.

Trước hết, nhóm em xin lần nữa cảm ơn thầy vì sự tận tâm và hỗ trợ hết mình với sinh viên thực hiện đồ án nói chung và nhóm em nói riêng. Nhờ vậy mà nhóm em mới được có cơ hội tìm tòi và hiểu biết thêm, mở rộng thêm về kiến thức lẫn kỹ năng về các vi điều khiển, linh kiện.v.v.v từ đó thực hiện được các mục tiêu đề ra.

Nhóm em từng gặp khó khăn rất lớn về việc truyền dữ liệu của module NRF24L01 khiến việc thực hiện đồ án bị trì trệ khá nhiều thời gian, nhờ thầy chỉ bảo và đưa ra những phương án khắc phục mà nhóm em mới có thể sửa chữa, hoàn thành đồ án kịp thời hạn.

Đồ án này không những là tiền đề đầu tiên để nhóm em thực hiện những đồ án tiếp theo mà nó còn là động lực giúp nhóm em tìm hiểu sâu hơn về cách phân bố chiến lược, thời gian thực hiện, phân bố công việc, kỹ năng làm việc nhóm và hơn thế nữa là kiến thức chuyên ngành.

Lần nữa xin chân thành cảm ơn thầy.

Nhóm sinh viên thực hiện đồ án,

Lê Xuân Đỉnh

Trần Mạnh Cường

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	9
1.1. TỔNG QUAN VỀ PHẦN MỀM	9
1.1.1. Tổng quan về Arduino IDE	9
1.2. TỔNG QUAN VỀ PHẦN CỨNG	11
1.2.1. Module ESP 8266 Node MCU	11
1.2.2. Module Arduino UNO R3.....	13
1.2.3. Module NRF24L01.....	16
1.2.4. Module OLED giao tiếp I2C	19
1.2.4. Cảm biến MQ2	20
CHƯƠNG 2: THIẾT KẾ HỆ THỐNG	21
2.1. GIỚI THIỆU VÀ TÓM TẮT	21
2.1.1. Yêu cầu chung của hệ thống.....	21
2.1.2. Phương án thiết kế	21
2.2. SƠ ĐỒ KHỐI HỆ THỐNG.....	21
2.3. LƯU ĐỒ GIẢI THUẬT.....	23
CHƯƠNG 3: THI CÔNG HỆ THỐNG	25
3.1. SƠ ĐỒ NGUYÊN LÝ	25
3.2. THIẾT KẾ PHẦN MỀM.....	25
3.3. THIẾT KẾ PHẦN CỨNG.....	34
3.3.1. Mạch PCB khối xử lý thứ cấp (Node)	34
3.3.2. Mạch PCB khối xử lý trung tâm (Gateway)	35
3.3.3. Mạch PCB.....	36
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC	37
4.1 Khối xử lý thứ cấp (Node)	37
4.2. Khối xử lý trung tâm (Gateway).....	38
4.3 Kiểm tra và đánh giá	39
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	40
5.1. KẾT LUẬN.....	40
5.2. HƯỚNG PHÁT TRIỂN.....	40

DANH MỤC HÌNH ẢNH

Hình 1.1. Arduino IDE

Hình 1.2. Giao diện Arduino IDE

Hình 1.3. Module ESP8266 Node MCU

Hình 1.4. ESP8266 Node MCU pinout

Hình 1.5. Module Arduino UNO R3

Hình 1.6. Arduino UNO R3 pinout

Hình 1.7. Module NRF24L01

Hình 1.8. Đế cấp nguồn 3.3V cho module NRF24L01

Hình 1.9. Module OLED

Hình 1.10. Cảm biến MQ2

Hình 2.1. Sơ đồ khối hệ thống

Hình 2.2. Lưu đồ giải thuật

Hình 3.1. Sơ đồ nối chân của khối xử lý thứ cấp PCB khối xử lý trung tâm

Hình 3.2. Sơ đồ nối chân của khối xử lý trung tâm

Hình 3.3. Mạch PCB khối xử lý thứ cấp

Hình 3.4. Mạch PCB khối xử lý trung tâm

Hình 4.1. Mặt trước khối xử lý thứ cấp

Hình 4.2. Mặt sau khối xử lý thứ cấp

Hình 4.3. Mặt trước của khối xử lý trung tâm

Hình 4.4. Mặt sau của khối xử lý trung tâm

DANH MỤC BẢNG BIỂU

Bảng 1. Chức năng chân của module ESP8266 Node MCU

Bảng 2. Thông số kỹ thuật module Arduino UNO R3

Bảng 3. Chức năng chân Arduino UNO R3

Bảng 4. Thông số kỹ thuật của module NRF24L01

Bảng 5. Thông số kỹ thuật module OLED

Bảng 6. Chức năng chân module OLED

Bảng 7. Chức năng chân MQ2

Bảng 8. Sơ đồ nối chân MQ2 với ESP8266 Node MCU và Arduino UNO R3

Bảng 9. Sơ đồ nối chân NRF24L01 với ESP8266 Node MCU và Arduino UNO R3

Bảng 10. Sơ đồ nối chân linh kiện khối thứ cấp

Bảng 11. Sơ đồ nối chân MQ2 với ESP8266 Node MCU

Bảng 12. Sơ đồ nối chân NRF24L01 với ESP8266 Node MCU

Bảng 13. Sơ đồ nối chân OLED với ESP8266 Node MCU

Bảng 14: Sơ đồ nối chân linh kiện khối trung tâm

MỞ ĐẦU

1. Giới thiệu và lý do chọn đề tài

Ngày nay, khi khoa học công nghệ phát triển một cách mạnh mẽ thì việc ứng dụng các thiết bị điện tử vào đời sống ngày càng phổ biến rộng rãi hơn. Với những kiến thức đã được học từ trường cùng với sự tìm tòi kiếm thêm trên các diễn đàn và khoa học công nghệ trên các trang mạng xã hội lớn khác nhau, nhóm em mong muốn góp thêm phần nào sự phát triển xã hội bằng cách học hỏi và đưa ra những sản phẩm có ích cho cuộc sống. Nhóm em xin giới thiệu một sản phẩm rất thực tế và thiết thực cho cuộc sống của chúng ta, đặc biệt là cho những khu vực nội trợ gia đình cũng như khu vực bếp của quán ăn, nhà hàng: “Thiết bị cảnh báo rò rỉ khí gas”.

Như chúng ta đã biết, hậu quả của sự cố cháy nổ và hỏa hoạn gây ra rất nặng nề và đáng báo động. Nhằm nắm bắt được những số liệu trên và có thể hạn chế được phần nào hậu quả mà hỏa hoạn và cháy nổ mang lại, nhóm em đã quyết định thiết kế và thi công hệ thống cảnh báo, phòng chống hỏa hoạn và rò rỉ khí gas. Với mục tiêu là hệ thống có thể phát hiện được khí gas rò rỉ một cách nhanh chóng và chính xác. Phương thức cảnh báo đơn giản bằng còi cảnh báo và đèn cảnh báo, đồng thời gửi thông tin đến bộ phận trung tâm nhằm phát hiện và xử lý kịp thời tránh sự cố có thể xảy ra.

Nhóm sẽ thực hiện hệ thống thông qua việc tích lũy kiến thức được học tại trường cũng như quan sát và nghiên cứu thực tiễn. Tất cả vì mục đích hoàn thành được mô hình dễ lắp đặt, dễ sử dụng, tiết kiệm điện, mang lại cho người dùng cảm giác tin cậy và an toàn.

2. Mục tiêu đề tài

Phát hiện sớm sự rò rỉ: Mục tiêu quan trọng nhất của đề tài là phát hiện sớm và cảnh báo rò rỉ khí gas. Thiết bị cảnh báo phải có khả năng nhận biết sự thay đổi nồng độ khí gas và đưa ra cảnh báo kịp thời để người dùng có thể thực hiện các biện pháp khẩn cấp.

Đánh giá mức độ nguy hiểm: Đánh giá mức độ nguy hiểm. Điều này giúp người dùng có cái nhìn rõ ràng về tình hình và quyết định hành động phù hợp.

Độ chính xác và đáng tin cậy: Thiết bị cảnh báo cần có độ chính xác và đáng tin cậy cao để tránh sự cảnh báo sai hoặc bỏ sót rò rỉ.

Tích hợp và quản lý từ xa: Mục tiêu cuối cùng là tạo ra một hệ thống cảnh báo khí gas có khả năng tích hợp với các hệ thống khác như hệ thống báo động chung,

3. Phương pháp nghiên cứu

- + Nghiên cứu tài liệu.
- + Phân tích và tổng hợp lý thuyết.
- + Tra cứu trên các diễn đàn internet.

4. Bố cục bản báo cáo

Chương 1: Cơ sở lý thuyết

- Trình bày kiến thức liên quan được sử dụng để thực hiện đề tài.

Chương 2: Thiết kế hệ thống

- Trình bày sơ đồ khối hệ thống và giải thích cách thức hoạt động của nó.
- Đưa ra các phương pháp lựa chọn phần cứng và xác định lựa chọn phù hợp với yêu cầu của đề tài.
- Xây dựng giải thuật và thuật toán cho hệ thống.

Chương 3: Thi công hệ thống

- Tiến hành thi công hệ thống.
- Dẫn chứng code.
- Hình ảnh thực tế của hệ thống sau khi thi công.

Chương 4: Kết luận và hướng phát triển

- Đưa ra kết luận và hướng phát triển cho hệ thống sau này.

NỘI DUNG

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Để tiến hành thiết kế phần cứng cũng như phần mềm, nhóm đã tìm hiểu một số kiến thức sau:

1.1. TỔNG QUAN VỀ PHẦN MỀM

1.1.1. Tổng quan về Arduino IDE

Arduino IDE (Integrated Development Environment) là một phần mềm mã nguồn mở được phát triển bởi Arduino.cc, cung cấp một môi trường lập trình cho vi điều khiển Arduino và các board phụ trợ. Nó là một công cụ phổ biến trong cộng đồng Arduino và rất dễ sử dụng cho người mới học lập trình.



Hình 1.1. Arduino IDE

Dưới đây là một số thông tin tổng quan về phần mềm Arduino IDE:

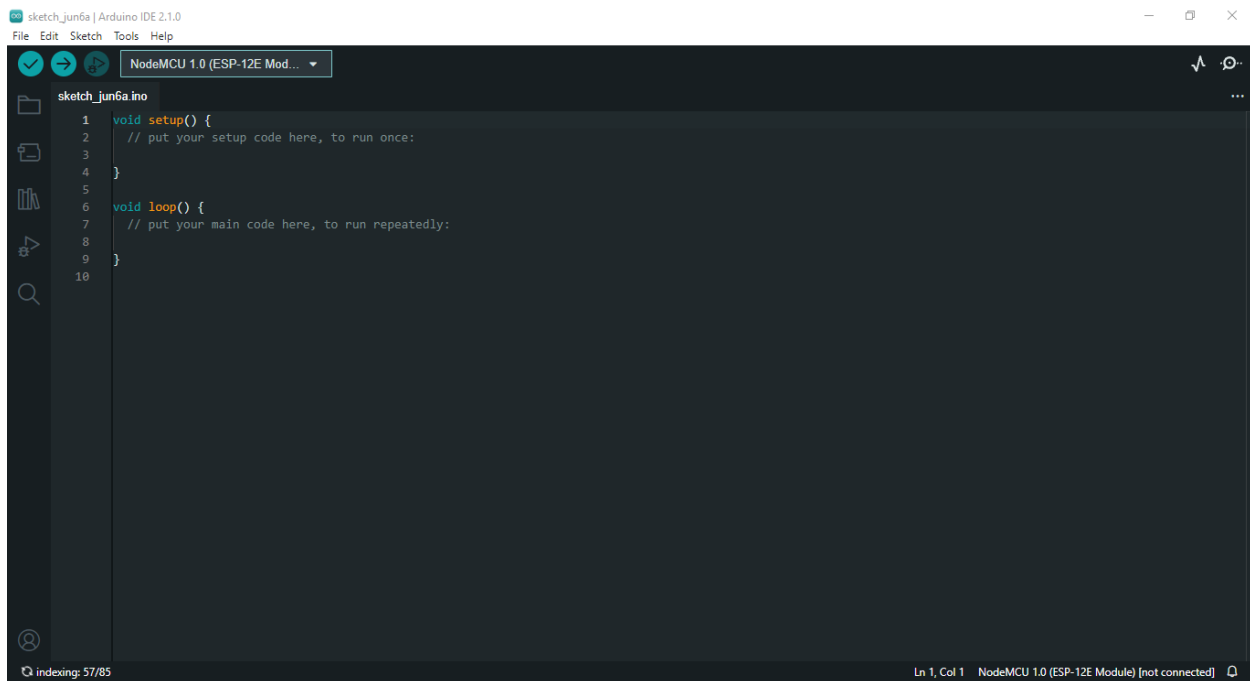
1. Ngôn ngữ lập trình: Arduino IDE sử dụng ngôn ngữ lập trình dựa trên Wiring, một ngôn ngữ lập trình đơn giản dựa trên C++.
2. Ngôn ngữ lập trình: Arduino IDE sử dụng ngôn ngữ lập trình dựa trên Wiring, một ngôn ngữ lập trình đơn giản dựa trên C++.
3. Thư viện: Arduino IDE đi kèm với một số thư viện tiêu chuẩn, cung cấp các chức năng và chương trình mẫu đã được viết sẵn giúp cho việc lập trình

Arduino trở nên dễ dàng hơn. Ngoài ra, người dùng cũng có thể thêm thư viện bổ sung từ cộng đồng Arduino.

4. Biên dịch và nạp chương trình: Arduino IDE cung cấp các công cụ để biên dịch mã nguồn và nạp chương trình lên board Arduino thông qua cổng COM. Người dùng có thể chọn board và cổng COM phù hợp trên giao diện để thực hiện việc này.

5. Hỗ trợ nhiều loại board: Arduino IDE không chỉ hỗ trợ các board Arduino chính thống như Arduino Uno, Mega, Nano, mà còn hỗ trợ các board phụ trợ khác như ESP8266, ESP32, Adafruit, etc. Điều này cho phép người dùng phát triển ứng dụng trên nhiều loại board khác nhau.

6. Cộng đồng và tài liệu: Arduino IDE có một cộng đồng mạnh mẽ với nhiều tài liệu, ví dụ như hướng dẫn, ví dụ mã nguồn và diễn đàn, giúp người dùng tìm hiểu và giải quyết các vấn đề phát sinh trong quá trình lập trình.



Hình 1.2. Giao diện Arduino IDE

Arduino IDE là một công cụ mạnh mẽ và phổ biến để lập trình vi điều khiển Arduino. Nó được sử dụng rộng rãi trong các dự án điện tử, robocon, IoT và nhiều ứng dụng khác.

1.2. TỔNG QUAN VỀ PHẦN CỨNG

1.2.1. Module ESP 8266 Node MCU

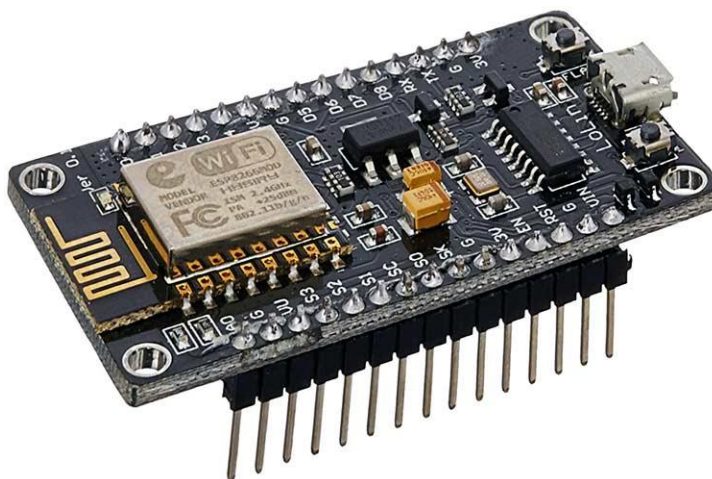
Module ESP8266 NodeMCU là một phiên bản phát triển của vi mạch ESP8266, được tích hợp trên một board phát triển có tên là NodeMCU. ESP8266 là một con chip Wi-Fi mạnh mẽ, có khả năng kết nối mạng và xử lý dữ liệu. NodeMCU là một board phát triển mã nguồn mở dựa trên ESP8266, được thiết kế để đơn giản hóa việc phát triển ứng dụng Internet of Things (IoT) và các dự án nhúng.

Module ESP8266 NodeMCU cung cấp một môi trường lập trình đơn giản và linh hoạt thông qua ngôn ngữ Lua, cho phép người dùng nhanh chóng phát triển các ứng dụng IoT. Ngoài ra, NodeMCU cũng hỗ trợ Arduino IDE, cho phép người dùng sử dụng ngôn ngữ lập trình Arduino thông qua một board có tích hợp sẵn ESP8266.

Module này có đầy đủ tính năng của ESP8266 như kết nối Wi-Fi để truyền và nhận dữ liệu, giao tiếp UART, GPIO (General Purpose Input/Output) cho việc kết nối với các linh kiện và cảm biến ngoại vi, và SPI (Serial Peripheral Interface) cho việc giao tiếp với các thiết bị ngoại vi khác.

Một ưu điểm của Module ESP8266 NodeMCU là kích thước nhỏ gọn, điều này giúp nó phù hợp với các ứng dụng có không gian giới hạn. Nó cũng được trang bị bộ nhớ flash tích hợp để lưu trữ mã nguồn và dữ liệu.

Với sức mạnh và tính linh hoạt của nó, Module ESP8266 NodeMCU thường được sử dụng trong các dự án IoT như kiểm soát thiết bị thông qua mạng Wi-Fi, thu thập dữ liệu từ các cảm biến và truyền tải nó đến máy chủ, hoặc thực hiện các tác vụ tự động thông qua kết nối mạng. Nó cũng có thể được sử dụng làm một điểm truy cập Wi-Fi hoặc bộ điều khiển cho các ứng dụng IoT khác.

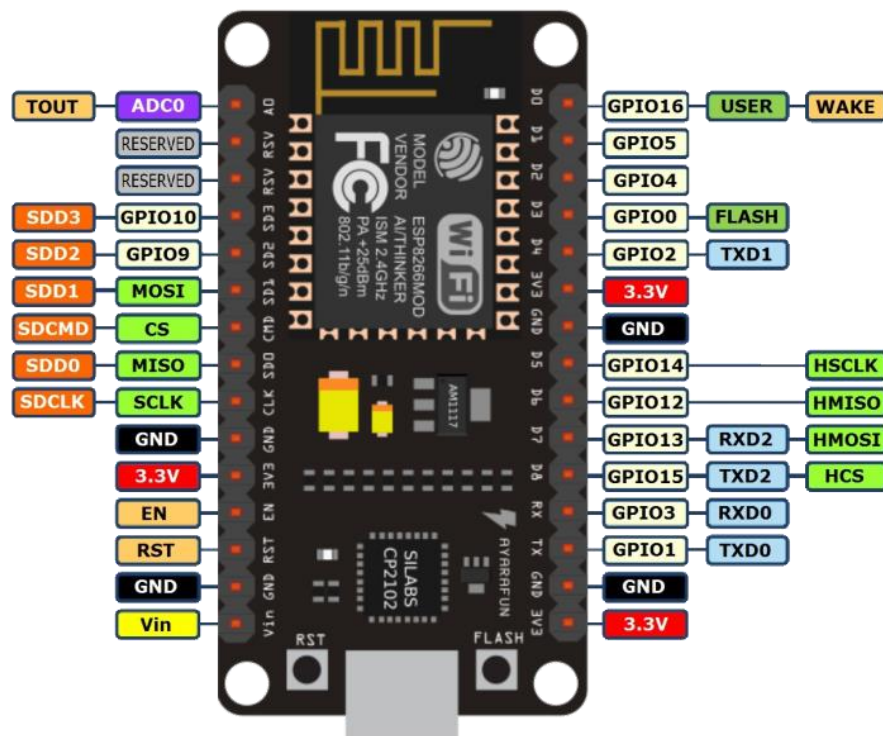


Hình 1.3. Module ESP8266 Node MCU

- Thông số kỹ thuật của module ESP8266 Node MCU:

- Chip: ESP8266 Wifi SoC (ESP-12)
- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
- Điện áp hoạt động: 3.3V
- Điện áp vào: 5V thông qua cổng USB
- 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 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4MB
- Giao tiếp: Cable Micro USB
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp: TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU – Lua

- Sơ đồ chân và chức năng chân:



Hình 1.4. ESP8266 Node MCU pinout

Bảng 1. Chức năng chân của module ESP8266 Node MCU

Tên	Chức năng
3.3V	Nguồn 3.3 V.
GND	Chân nối mass.
Vin	Nguồn điện cấp từ ngoại vi
EN – RST	Được sử dụng để đo điện áp tương tự trong khoảng 0-3,3V
A0	Được sử dụng để đo điện áp tương tự trong khoảng 0-3,3V
GPIO1 – GPIO16	NodeMCU có 16 chân đầu vào-đầu ra mục đích chung trên bo mạch của nó
SD1 – CMD – SD0 – CLK	NodeMCU có sẵn bốn chân để giao tiếp SPI.
TXD0 RXD0 TXD2 RXD2	NodeMCU có hai giao diện UART, UART0 (RXD0 & TXD0) và UART1 (RXD1 & TXD1). UART1 được sử dụng để tải lên phần sụn / chương trình.

1.2.2. Module Arduino UNO R3

Arduino UNO là một board phát triển phổ biến và phổ biến nhất trong gia đình Arduino. Nó là một board vi điều khiển nhỏ gọn, dễ sử dụng và có thể được sử dụng cho nhiều dự án khác nhau.

Arduino UNO có trọng lượng 25g với chiều dài và chiều rộng lần lượt là 68.6 mm và 53.4 mm.



Hình 1.5. Module Arduino UNO R3

- Thông số kỹ thuật:

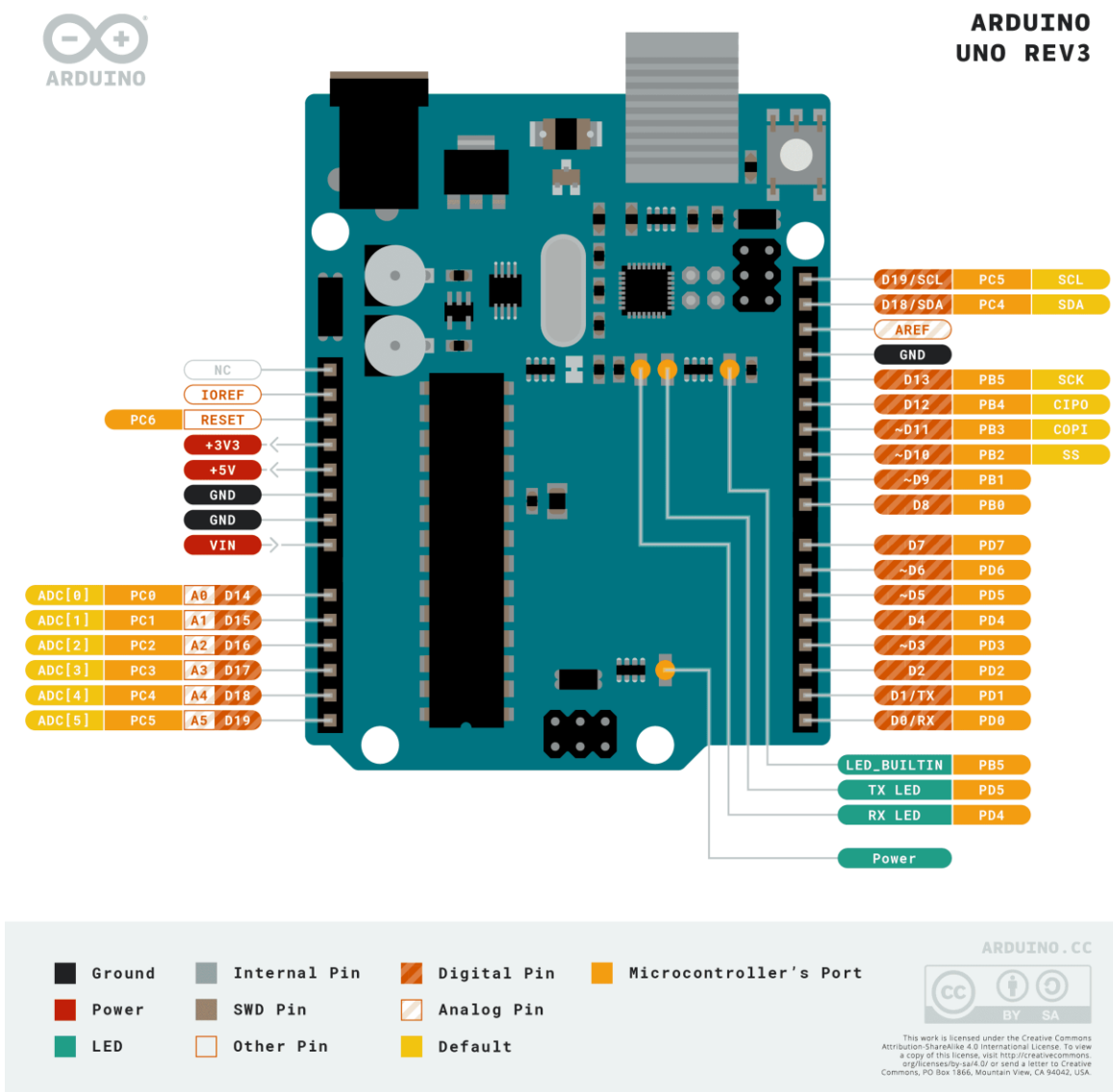
Bảng 2. Thông số kỹ thuật module Arduino UNO R3

Arduino UNO	Thông số kỹ thuật
Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16MHz
Dòng tiêu thụ	Khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 Ma
Bộ nhớ flash	32 kB (ATmega328) với 0.5kB dùng bởi bootloader
SRAM	2 kB (ATmega328)
EEPROM	1 kB (ATmega328)

Với vi điều khiển này, tổng cộng có 14 pin (ngõ) ra / vào được đánh số từ 0 tới 13 (trong đó có 6 pin PWM, được đánh dấu ~ trước mã số của pin). Song song đó, có thêm 6 pin nhận tín hiệu analog được đánh kí hiệu từ A0 - A5, 6 pin này cũng có thể sử dụng được như các pin ra / vào bình thường (như pin 0 - 13). Ở các pin được đề cập, pin 13 là pin đặc biệt vì nối trực tiếp với LED trạng thái trên board.

Trên board còn có 1 nút reset, 1 ngõ kết nối với máy tính qua cổng USB và 1 ngõ cấp nguồn sử dụng jack 2.1mm lấy năng lượng trực tiếp từ AC-DC adapter hay thông qua ắc-quy nguồn.

- Sơ đồ chân và chức năng chân:



Hình 1.6. Arduino UNO R3 pinout

Bảng 3. Chức năng chân Arduino UNO R3

Tên Pin	Chức năng
5V	Cung cấp điện áp 5V từ nguồn điện ngoài hoặc từ USB.
3.3V	Cung cấp điện áp 3.3V từ nguồn điện ngoài.
GND	Chân nối mass.
D0 – D13	Ngõ vào / ra số.
A0 – A5	Cổng đầu vào Analog.
Vin	Đầu vào điện áp DC từ nguồn ngoại vi

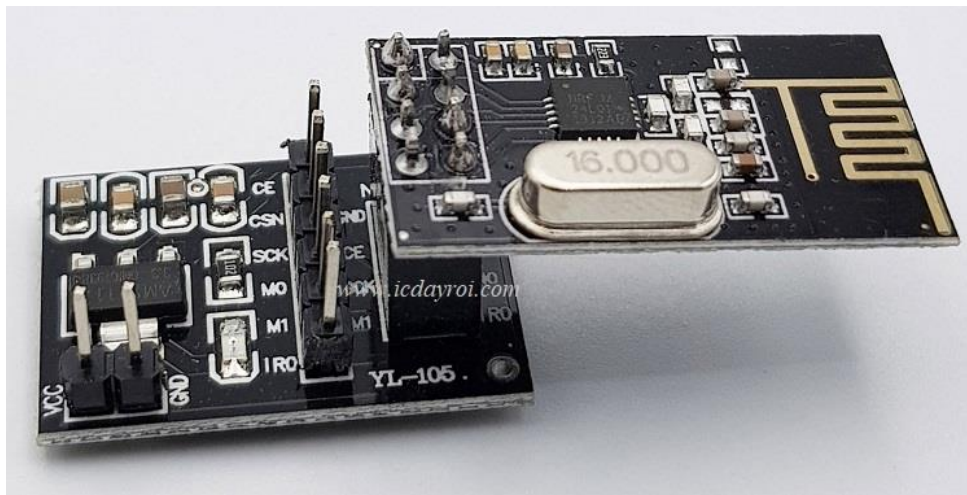
RESET	Chân reset cho vi vi điều khiển.
AREF	Điện áp tham chiếu analog.

1.2.3. Module NRF24L01

Module NRF24L01 là một module truyền thông không dây sử dụng công nghệ truyền thông radio tần số 2.4GHz. Nó được phát triển bởi Nordic Semiconductor và được sử dụng phổ biến trong các ứng dụng IoT, điều khiển từ xa, mạng cảm biến và truyền thông không dây.



Hình 1.7. Module NRF24L01



Hình 1.8. Đế cấp nguồn 3.3V cho module NRF24L01

Bảng 4. Thông số kỹ thuật của module NRF24L01

NRF24L01	Thông số kỹ thuật
Tần số hoạt động	2.4 GHz
Giao thức truyền	Không dây
Tốc độ truyền	Tốc độ tối đa 2Mbps

Khoảng cách truyền	Từ vài chục m đến trăm m
Kênh truyền	126 kênh
Điện áp hoạt động	3.3V
Giao diện	Sử dụng giao diện SPI

Module NRF24L01 sử dụng 3 thư viện “SPI.h”, “nRF24L01.h” và “RF24.h”. Chúng ta có thể tham khảo đoạn code bên nhận và bên gửi thông qua module NRF24L01 như sau:

Bên gửi:

```
#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8); // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
    radio.begin();

    //set the address
    radio.openWritingPipe(address);

    //Set module as transmitter
    radio.stopListening();
}

void loop()
{
    //Send message to receiver
    const char text[] = "Hello World";

    radio.write(&text, sizeof(text));

    delay(1000);
}
```

```
}
```

Bên nhận:

```
#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8); // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
    while (!Serial);
    Serial.begin(9600);
    radio.begin();
    //set the address
    radio.openReadingPipe(0, address);
    //Set module as receiver
    radio.startListening();
}

void loop()
{
    //Read the data if available in buffer
    if (radio.available())
    {
        char text[32] = {0};
        radio.read(&text, sizeof(text));
        Serial.println(text);
    }
}
```

```

}
}

```

1.2.4. Module OLED giao tiếp I2C

Màn hình Oled 0.96 inch giao tiếp I2C cho khả năng hiển thị đẹp, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO.



Hình 1.9. Module OLED

Bảng 5. Thông số kỹ thuật module OLED

OLED	Thông số kỹ thuật
Điện áp sử dụng	2.2 – 5.5 VDC
Công suất tiêu thụ	0.04w
Góc hiển thị	Lớn hơn 160 độ
Số điểm hiển thị	128x64 điểm
Độ rộng màn hình	0.96 inch
Màu hiển thị	Trắng/Xanh Dương
Giao tiếp	I2C
Driver	SSD1306

Bảng 6. Chức năng chân module OLED

OLED	Chức năng
VCC	Nguồn 3.3V
GND	Chân nối mass.
SCL	Xung Clock
SDA	Dữ liệu đầu vào

1.2.4. Cảm biến MQ2

Cảm biến khí ga MQ2 sử dụng phân tử SnO_2 có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO_2 sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện. MQ2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH_4) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.



Hình 1.10. Cảm biến MQ2

- Thông số kỹ thuật cảm biến MQ2:

- + Nguồn hoạt động: 5V
- + Loại dữ liệu: Analog
- + Phạm vi phát hiện rộng
- + Tốc độ phản hồi nhanh và độ nhạy cao
- + Mạch đơn giản
- + Ổn định khi sử dụng trong thời gian dài

- Chức năng chân:

Bảng 7. Chức năng chân MQ2

Tên	Chức năng
A0	Đo và cung cấp nồng độ khí
D0	Phát hiện sự có mặt của khí
GND	Chân nối mass

VCC	Chân nối nguồn (thường là 3.3V)
-----	-----------------------------------

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG

Sau khi nhóm đã tìm hiểu một số khái niệm liên quan cần thiết, nhóm tiến hành thiết kế và xây dựng phần cứng cũng như phần mềm.

2.1. GIỚI THIỆU VÀ TÓM TẮT

2.1.1. Yêu cầu chung của hệ thống

Với tiêu chí thiết kế, xây dựng một hệ thống cảnh báo cháy dễ sử dụng. Hệ thống có các tính năng như sau:

- Hệ thống sử dụng cảm biến đo lường nồng độ khí gas ở môi trường xung quanh.
- Từ các vị trí khác nhau có thể gửi dữ liệu đến khu vực trung tâm để kịp thời xử lý, tránh xảy ra hậu quả lớn.
- Các thiết bị đèn và còi cảnh báo hoạt động khi nồng độ khí gas vượt mức an toàn.

2.1.2. Phương án thiết kế

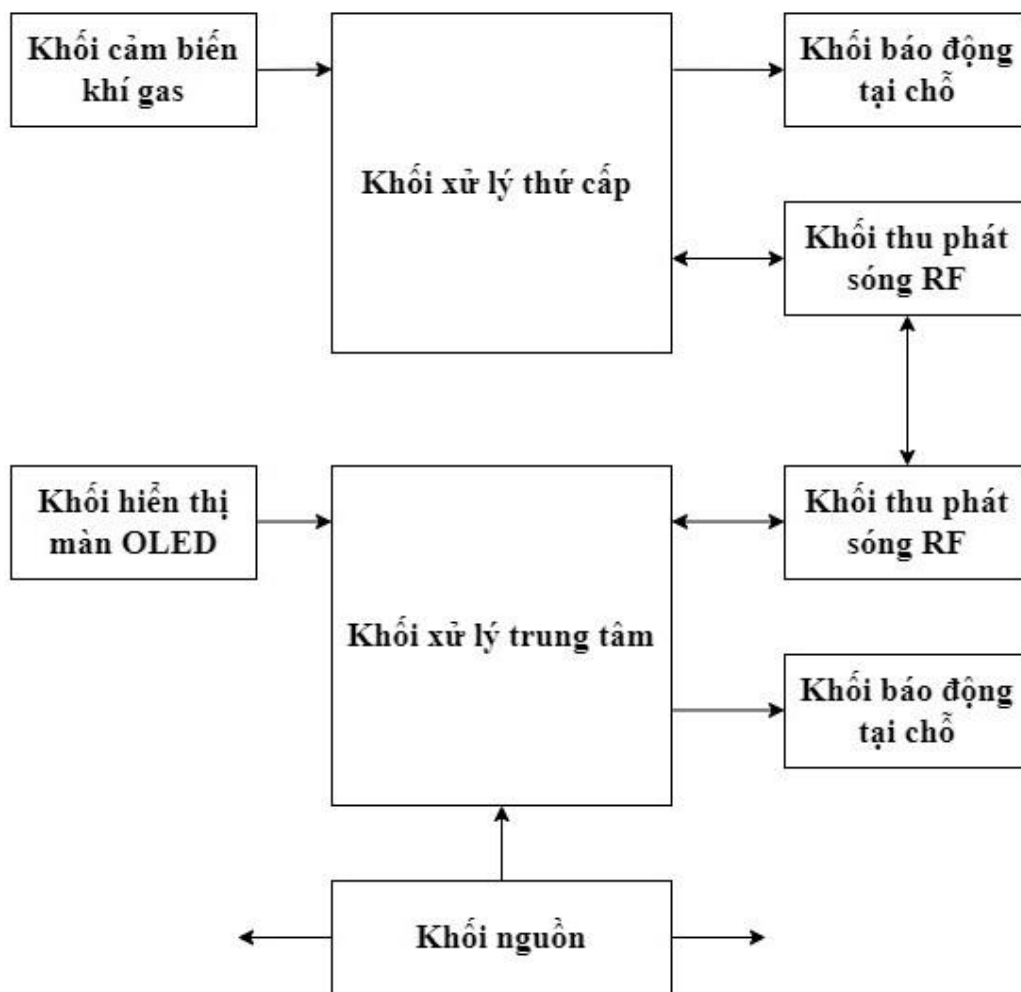
Với yêu cầu của hệ thống trên nhóm đã tiến hành đề ra các phương án thiết kế cho phần cứng như sau:

- Khối xử lý sẽ tận dụng các module có sẵn ngoài thị trường như module ESP8266 Node MCU, module Arduino UNO.
- Khối nguồn sẽ sử dụng nguồn được cấp thông qua USB (sau này sẽ phát triển về nguồn rời cho hệ thống).
- Khối hiển thị sẽ sử dụng module OLED 0.96 inch giao tiếp I2C.
- Thiết kế vẽ board mạch kết nối các module lại.

Nhóm sử dụng phần mềm Arduino IDE để thực hiện thiết kế hệ thống.

2.2. SƠ ĐỒ KHỐI HỆ THỐNG

Hệ thống bao gồm các khối xử lý thứ cấp cùng với một khối xử lý trung tâm, bên cạnh đó hệ thống sử dụng các khối cảm biến, khối báo động tại chỗ, khối RF, khối hiển thị nhằm giúp người dùng đánh giá tính hình tốt hơn và đưa ra những hành động kịp thời.



Hình 2.1. Sơ đồ khối hệ thống

- Chức năng các khối:

- **Khối nguồn:** Cung cấp nguồn hoạt động cho hệ thống.
- **Khối cảm biến khí:** Đo nồng độ khí gas và gửi về cho ESP8266 NodeMCU xử lý.
- **Khối báo động:** Phát ra tiếng báo động / đèn cảnh báo tại chỗ khi cảm biến ở bất cứ nơi đâu trong hệ thống phát hiện có nguy hiểm.
- **Khối thu phát sóng RF:** Thu phát sóng RF để giao tiếp giữa khối xử lý thứ cấp và khối xử lý trung tâm.
- **Khối hiển thị:** Hiển thị nồng độ khí gas đo được và trạng thái an toàn hay nguy hiểm.

- **Khối xử lý thứ cấp:** Thu nhận dữ liệu từ cảm biến và xử lý, nếu vượt quá giới hạn cho phép thì gửi tín hiệu báo nguy hiểm cho khối xử lý trung tâm xử lý.
- **Khối xử lý trung tâm:** Hiển thị nơi rò rỉ khí lên màn OLED nhằm giúp người sử dụng giám sát được nơi nào có tín hiệu báo nguy hiểm.

Nguyên lý hoạt động chung: Khi nồng độ khí gas ở mỗi khu vực vượt quá an toàn, các thiết bị cảnh báo ở khu vực đó sẽ hoạt động, đồng thời gửi tín hiệu đến khu trung tâm và hiển thị khu vực nguy hiểm để nhằm xử lý kịp thời.

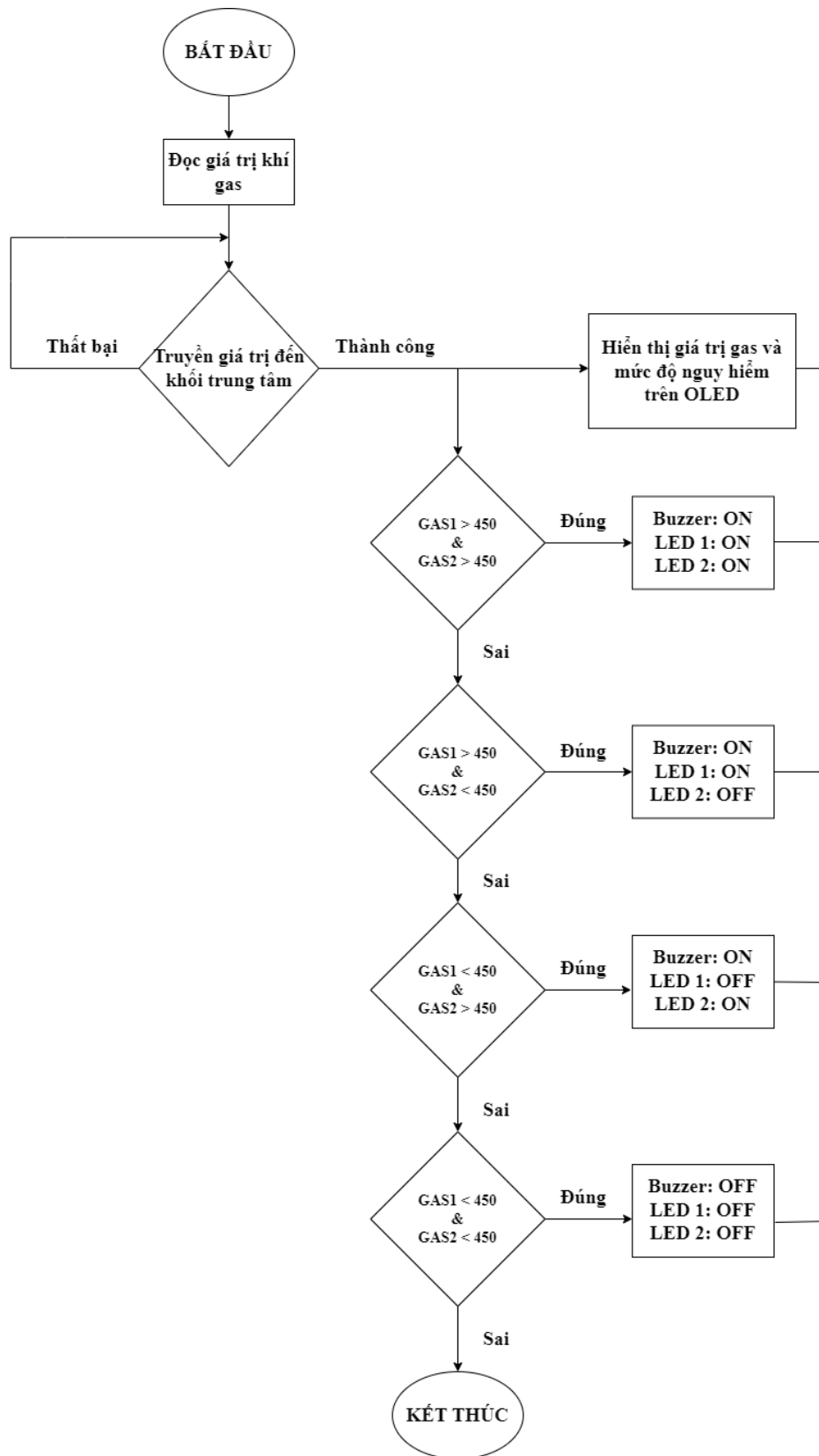
2.3. LƯU ĐỒ GIẢI THUẬT

Dưới đây là lưu đồ giải thuật cho hệ thống:

Quy trình thực hiện của hệ thống sẽ bắt đầu vào việc đọc giá trị analog của khí gas. Sau đó, giá trị thu được sẽ chuyển vào khối xử lý thứ cấp và được phân tích dựa theo giá trị cho trước nhằm phân bố khu vực đó thuộc tình trạng nguy hiểm hay an toàn.

Nếu giá trị gas vượt ngưỡng an toàn, còi cảnh báo sẽ vang lên đồng thời led cảnh báo cũng sáng nhấp nháy để gây chú ý đến người xung quanh.

Đồng thời, giá trị khí gas thu được sẽ được gửi về khối trung tâm nhằm hiển thị giá trị và mức độ nguy hiểm giúp người quản lý dễ dàng nắm bắt tình hình khu vực.



Hình 2.2. Lưu đồ giải thuật

CHƯƠNG 3: THI CÔNG HỆ THỐNG

3.1. SƠ ĐỒ NGUYÊN LÝ

3.1.1. Khối xử lý thứ cấp

3.1.2. Khối xử lý trung tâm

3.2. THIẾT KẾ PHẦN MỀM

3.2.1. Code khối xử lý thứ cấp

- Thư viện: <RF24> của TMRh20

```
#include <RF24.h>
```

- Sơ đồ nối chân:

Bảng 8. Sơ đồ nối chân MQ2 với ESP8266 Node MCU và Arduino UNO R3

	MQ2	ESP8266 Node MCU	Arduino UNO R3
1	A0	A0	A0
2	D0		
3	VCC	3.3V	3.3V
4	GND	GND	GND

Bảng 9. Sơ đồ nối chân NRF24L01 với ESP8266 Node MCU và Arduino UNO R3

	NRF24L01	ESP8266 Node MCU	Arduino UNO R3
1	VCC	Vin	Vin
2	GND	GND	GND
3	CE	D4	9
4	CSN	D8	10
5	SCK	D5	13
6	MOSI	D7	11
7	MISO	D6	12
8	IRQ		

Bảng 10. Sơ đồ nối chân linh kiện khối thứ cấp

		ESP8266 Node MCU	Arduino UNO R3
1	Led kết nối	D0	2
2	Led cảnh báo	D1	3
3	Buzzer	D2	4

- Code khối xử lý thứ cấp:

- ESP8266 Node MCU:

```

#include <RF24.h>

#define CE_PIN D4
#define CS_PIN D8
RF24 radio(CE_PIN, CS_PIN);

float gasValue1;
const uint64_t pipe1 = 0x00000000000E; // Địa chỉ ống truyền và nhận

#define Led_Connect D0
#define Led_Warning D1
#define Buzzer D2
void setup()
{
    pinMode(gasValue1, INPUT);
    pinMode(Led_Connect, OUTPUT);
    pinMode(Led_Warning, OUTPUT);
    pinMode(Buzzer, OUTPUT);
    //Cài đặt thông số kỹ thuật cho RF
    Serial.begin(9600);
    radio.begin();
    radio.openWritingPipe(pipe1);
    radio.setPALevel(RF24_PA_MIN);
    radio.stopListening();
}
void loop()
{
    gasValue1 = analogRead(A0);
    radio.write(&gasValue1, sizeof(gasValue1));
    Serial.print("Gas Value Node 1:");
    Serial.println(gasValue1);

    if (gasValue1 > 550)
    {
        digitalWrite(Led_Connect, HIGH);
        digitalWrite(Buzzer, HIGH);

        digitalWrite(Led_Warning, HIGH);
        delay(100);
        digitalWrite(Led_Warning, LOW);
        delay(100);
    }
    else
    {
        digitalWrite(Led_Connect, HIGH);
    }
}

```

```
    digitalWrite(Buzzer, LOW);  
  }  
}
```

- Arduino UNO R3:

```
#include <RF24.h>  
#include <SPI.h>  
#include <nRF24L01.h>  
  
#define CE_PIN 9  
#define CS_PIN 10  
RF24 radio(CE_PIN, CS_PIN);  
  
float gasValue1;  
const uint64_t pipe1 = 0xE8E8F0F0E1LL; //RF1  
  
#define Led_Connect 2  
#define Led_Warning 3  
#define Buzzer 4  
void setup()  
{  
  pinMode(gasValue1, INPUT);  
  pinMode(Led_Connect, OUTPUT);  
  pinMode(Led_Warning, OUTPUT);  
  pinMode(Buzzer, OUTPUT);  
  //Cai dat thong so ky thuat cho RF  
  Serial.begin(9600);  
  radio.begin();  
  radio.openWritingPipe(pipe1);  
  radio.setPALevel(RF24_PA_MIN);  
  radio.stopListening();  
}  
void loop()  
{  
  gasValue1 = analogRead(A0);  
  radio.write(&gasValue1, sizeof(gasValue1));  
  Serial.print("Gas Value Node 1:");  
  Serial.println(gasValue1);  
  
  if (gasValue1 > 450)  
  {  
    digitalWrite(Led_Connect, HIGH);  
    digitalWrite(Buzzer, HIGH);  
  
    digitalWrite(Led_Warning, HIGH);  
  }  
}
```

```

    delay(100);
    digitalWrite(Led_Warning, LOW);
    delay(100);
}
else
{
    digitalWrite(Led_Connect, HIGH);
    digitalWrite(Buzzer, LOW);
}
}

```

3.2.3. Code khối xử lý trung tâm

- Thư viện:

- <RF24.h> của TMRh20.
- <SPI.h>
- <Wire.h>
- <nRF24L01.h>
- <Adafruit_GFX.h> của Adafruit
- <Adafruit_SSD1306.h> của Adafruit

```

• #include <RF24.h>
• #include <nRF24L01.h>
• #include <Wire.h>
• #include <SPI.h>
• #include <Adafruit_GFX.h>
• #include <Adafruit_SSD1306.h>

```

- Sơ đồ nối chân:

Bảng 11. Sơ đồ nối chân MQ2 với ESP8266 Node MCU

	MQ2	ESP8266 Node MCU
1	A0	A0
2	D0	
3	VCC	3.3V
4	GND	GND

Bảng 12. Sơ đồ nối chân NRF24L01 với ESP8266 Node MCU

	NRF24L01	ESP8266 Node MCU
1	VCC	Vin
2	GND	GND
3	CE	D4
4	CSN	D8

5	SCK	D5
6	MOSI	D7
7	MISO	D6
8	IRQ	

Bảng 13. Sơ đồ nối chân OLED với ESP8266 Node MCU

	OLED	ESP8266 Node MCU
1	VCC	3.3V
2	GND	GND
3	SCL	D1
4	SDA	D2

Bảng 14: Sơ đồ nối chân linh kiện khối trung tâm

		ESP8266 Node MCU
1	Led cảnh báo 1	D0
2	Led cảnh báo 2	D3

- Code khối xử lý trung tâm:

```
#include <RF24.h>
#include <nRF24L01.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET -1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define Led_Warning_N1 D3
#define Led_Warning_N2 D0
#define CE_PIN D4
#define CS_PIN D8
RF24 radio(CE_PIN, CS_PIN);
float gasValue1;
float gasValue2;
const uint64_t pipe1 = 0xE8E8F0F0E1LL; //RF1
const uint64_t pipe2 = 0x00000000000F; //RF2

void setup(){
  Serial.begin(9600);
  radio.begin();
```

```

radio.openReadingPipe(1,pipe1);
radio.openReadingPipe(1,pipe2);
radio.setPALevel(RF24_PA_MIN);
radio.startListening();

display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDRESS);
display.display();
delay(1000);
display.clearDisplay();

pinMode(Led_Warning_N1, OUTPUT);
pinMode(Led_Warning_N2, OUTPUT);
}
void loop(){
  RF1();
  RF2();
  OLED();
}
void OLED(){
  if ((gasValue1 > 450)&(gasValue2 > 450)){
    //Node 1
    digitalWrite(Led_Warning_N1, HIGH);
    delay(100);
    digitalWrite(Led_Warning_N1, LOW);
    delay(100);
    //Node 2
    digitalWrite(Led_Warning_N2, HIGH);
    delay(100);
    digitalWrite(Led_Warning_N2, LOW);
    delay(100);

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    //Node 1
    display.setCursor(0,0);
    display.println("Gas Value KV1:");
    display.setCursor(85,0);
    display.print(gasValue1);
    display.setCursor(15,30);
    display.println("Khu vuc 1 nguy hiem");
    //Node 2
    display.setCursor(0,15);
    display.println("Gas Value KV2:");
    display.setCursor(85,15);
  }
}

```

```
    display.print(gasValue2);
    display.setCursor(15,50);
    display.println("Khu vuc 2 nguy hiem");

    display.display();
}
if((gasValue1 > 450)&(gasValue2 < 450)){
    //Node 1
    digitalWrite(Led_Warning_N1, HIGH);
    delay(100);
    digitalWrite(Led_Warning_N1, LOW);
    delay(100);
    //Node 2
    digitalWrite(Led_Warning_N2, LOW);

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    //Node 1
    display.setCursor(0,0);
    display.println("Gas Value KV1:");
    display.setCursor(85,0);
    display.print(gasValue1);
    display.setCursor(15,30);
    display.println("Khu vuc 1 nguy hiem");
    //Node 2
    display.setCursor(0,15);
    display.println("Gas Value KV2:");
    display.setCursor(85,15);
    display.print(gasValue2);
    display.setCursor(15,50);
    display.println("Khu vuc 2 an toan");

    display.display();
}
if((gasValue1 < 450)&(gasValue2 > 450)){
    //Node 1
    digitalWrite(Led_Warning_N1, LOW);
    //Node 2
    digitalWrite(Led_Warning_N2, HIGH);
    delay(100);
    digitalWrite(Led_Warning_N2, LOW);
    delay(100);

    display.clearDisplay();
```

```

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
//Node 1
display.setCursor(0,0);
display.println("Gas Value KV1:");
display.setCursor(85,0);
display.print(gasValue1);
display.setCursor(15,30);
display.println("Khu vuc 1 an toan");
//Node 2
display.setCursor(0,15);
display.println("Gas Value KV2:");
display.setCursor(85,15);
display.print(gasValue2);
display.setCursor(15,50);
display.println("Khu vuc 2 nguy hiem");

display.display();
}
if((gasValue1 < 450)&(gasValue2 < 450)){
//Node 1
digitalWrite(Led_Warning_N1, LOW);
//Node 2
digitalWrite(Led_Warning_N2, LOW);

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
//Node 1
display.setCursor(0,0);
display.println("Gas Value KV1:");
display.setCursor(85,0);
display.print(gasValue1);
display.setCursor(15,30);
display.println("Khu vuc 1 an toan");
//Node 2
display.setCursor(0,15);
display.println("Gas Value KV2:");
display.setCursor(85,15);
display.print(gasValue2);
display.setCursor(15,50);
display.println("Khu vuc 2 an toan");

display.display();
}

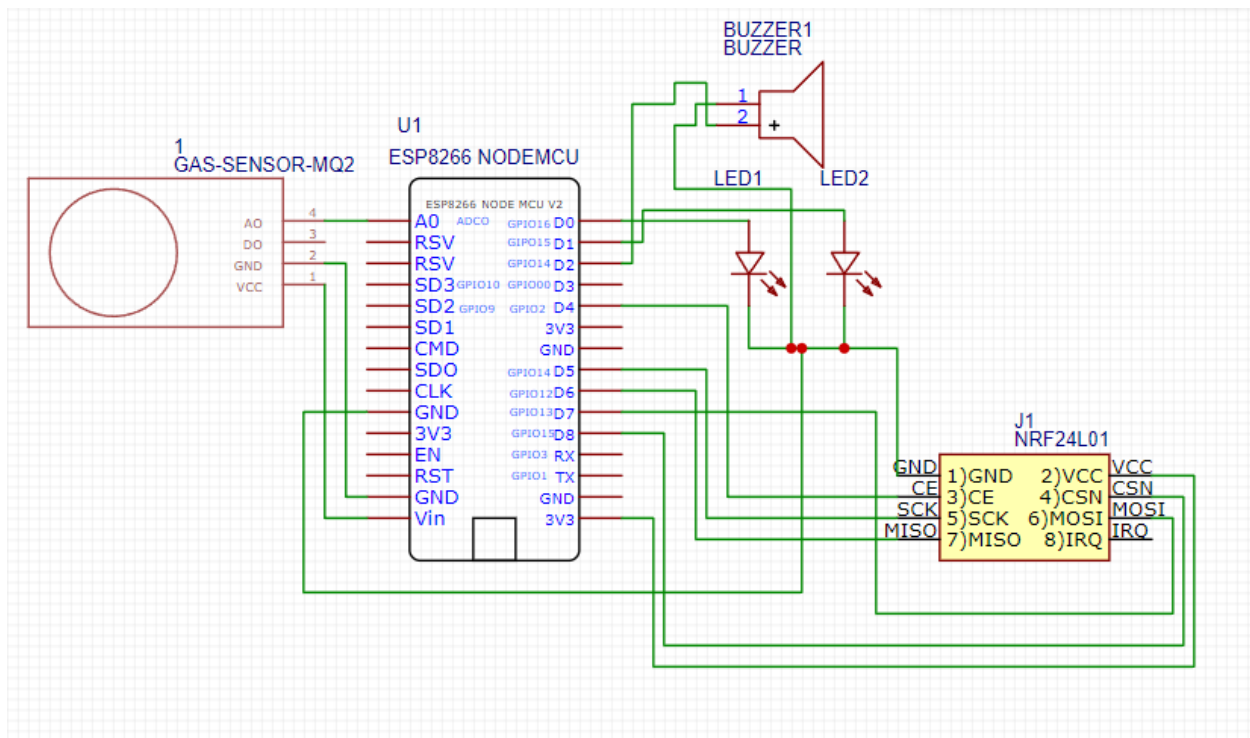
```



```
}  
void RF1(){  
    radio.begin();  
    radio.openReadingPipe(1,pipe1);  
    radio.setPALevel(RF24_PA_MIN);  
    radio.startListening();  
    delay(10);  
    if (radio.available()){  
        radio.read(&gasValue1, sizeof(gasValue1));  
        Serial.print("Gas Value KV1:: ");  
        Serial.println(gasValue1);  
    }  
    else{  
        Serial.println("khong co tin hieu 1");  
    }  
}  
void RF2(){  
    radio.begin();  
    radio.openReadingPipe(1,pipe2);  
    radio.setPALevel(RF24_PA_MIN);  
    radio.startListening();  
    delay(10);  
    if (radio.available()){  
        radio.read(&gasValue2, sizeof(gasValue2));  
        Serial.print("Gas Value KV2: ");  
        Serial.println(gasValue2);  
    }  
    else{  
        Serial.println("khong co tin hieu 2");  
    }  
}
```

3.3. THIẾT KẾ PHẦN CỨNG

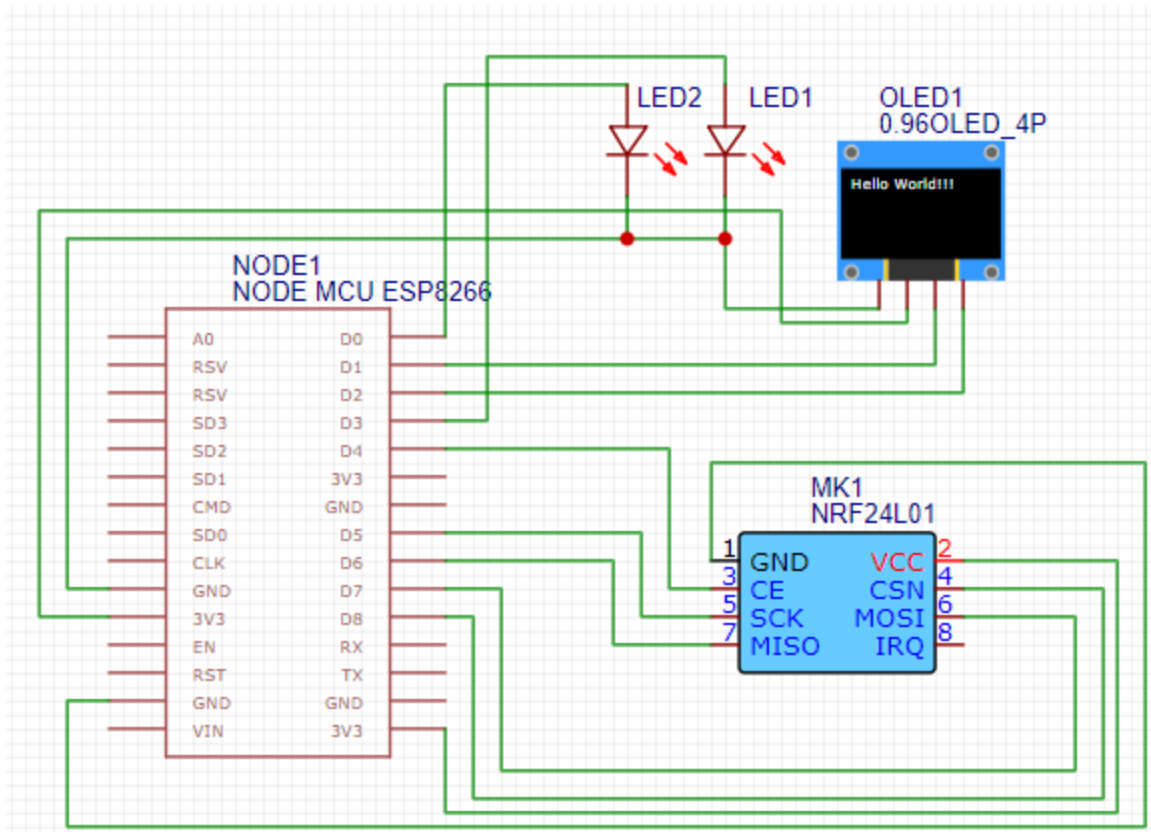
3.3.1. Mạch PCB khối xử lý thứ cấp (Node)



Hình 3.1. Sơ đồ nối chân của khối xử lý thứ cấp

- Khối xử lý thứ cấp thu thập dữ liệu khí gas qua cảm biến MQ2 và gửi dữ liệu đi dùng module NRF24L01 đến khối xử lý trung tâm (GATEWAY). Ngoài ra còn có 1 đèn cho biết trạng thái kết nối và 1 đèn nháy khi ngưỡng khí gas vượt mức độ cho phép thì lúc này BUZZER cũng sẽ hoạt động.
- Hệ thống gồm 2 khối xử lý thứ cấp gửi dữ liệu về cho khối xử lý trung tâm.

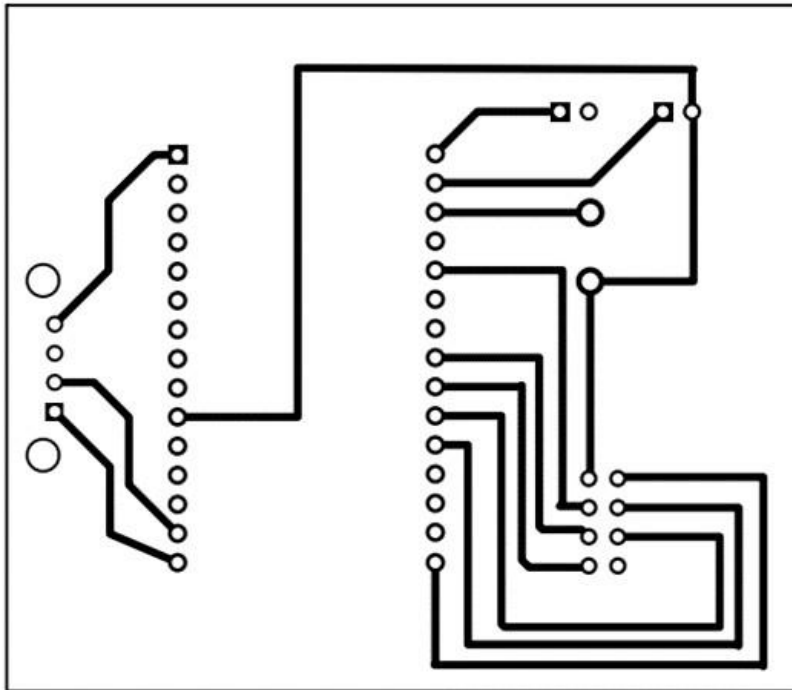
3.3.2. Mạch PCB khối xử lý trung tâm (Gateway)



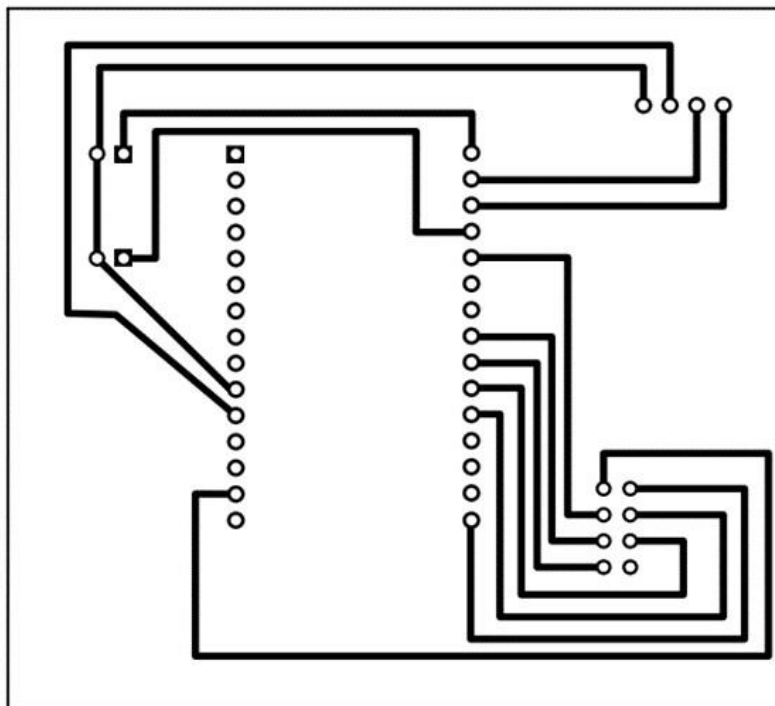
Hình 3.2. Sơ đồ nối chân của khối xử lý trung tâm

- Khối xử lý trung tâm dùng module NRF24L01 nhận dữ liệu khí gas từ 2 khối xử lý thứ cấp (NODE) và hiện thị các giá trị đó ra màn hình OLED.
- Màn hình OLED sẽ hiện thị 2 giá trị ở các NODE và cho biết là khu vực đó có an toàn không.
- Nếu ở NODE nào vượt ngưỡng gas cho phép thì đèn nháy cảnh báo ở GATEWAY cũng sẽ thông báo tương ứng.

3.3.3. Mạch PCB



Hình 3.3. Mạch PCB khối xử lý thứ cấp



Hình 3.4. Mạch PCB khối xử lý trung tâm

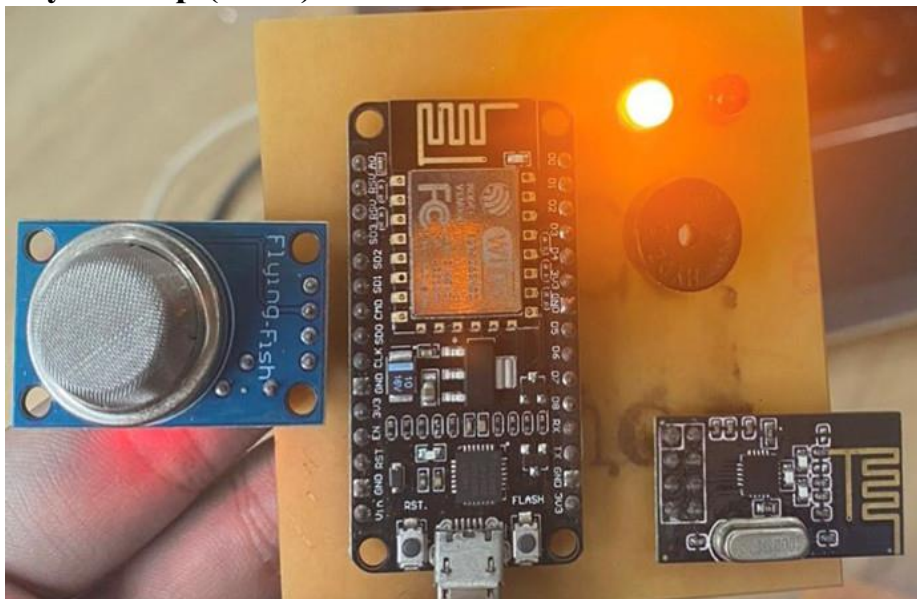
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

Sau khi thiết kế mạch PCB cho hệ thống, nhóm em đã tiến hành thực hiện mạch in cho hệ thống, với các công đoạn sau:

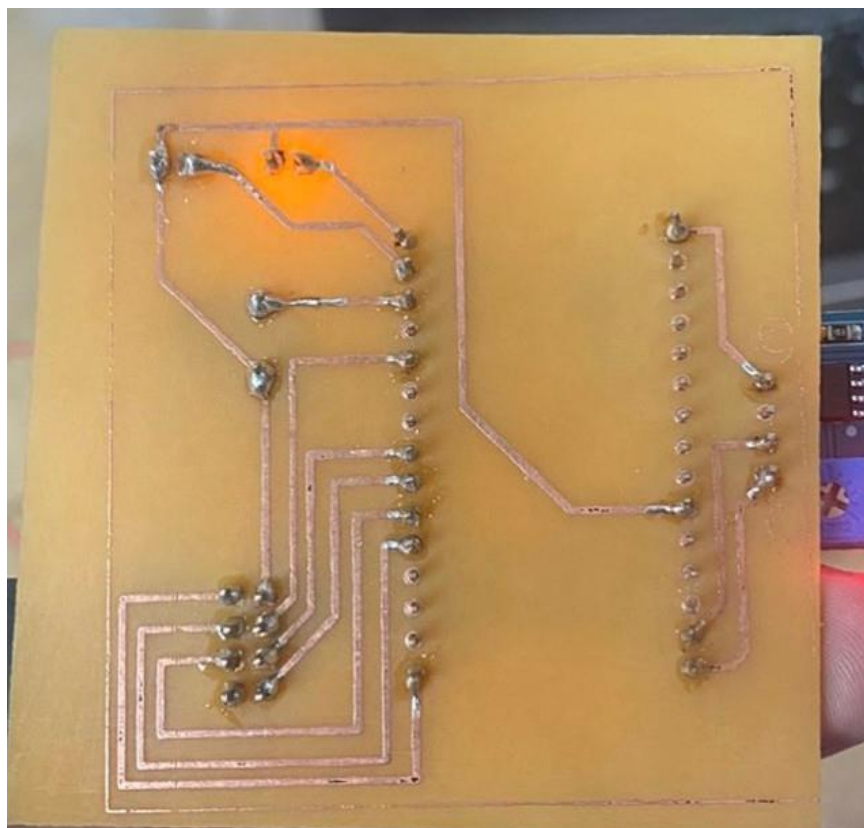
- In mạch.
- Mua bảng đồng và dung dịch axit rửa mạch.
- Tiến hành ủ hình mạch in lên bảng đồng.
- Ngâm axit để rửa mạch.
- Dùng khoan mini và các mũi khoan thích hợp để đục lỗ mạch.
- Tiến hành hàn mạch.

Và nhóm em đã thu được một số kết quả dưới đây.

4.1 Khối xử lý thứ cấp (Node)

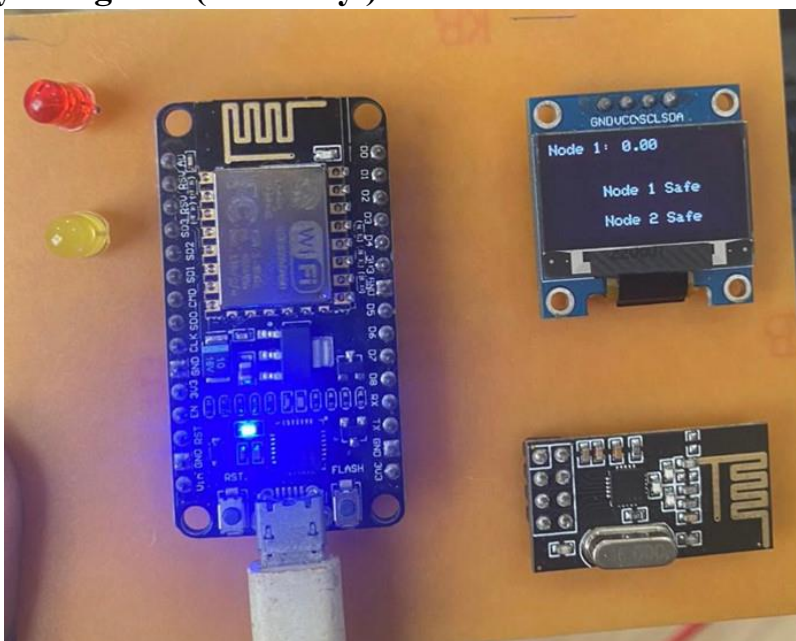


Hình 4.1. Mặt trước khối xử lý thứ cấp

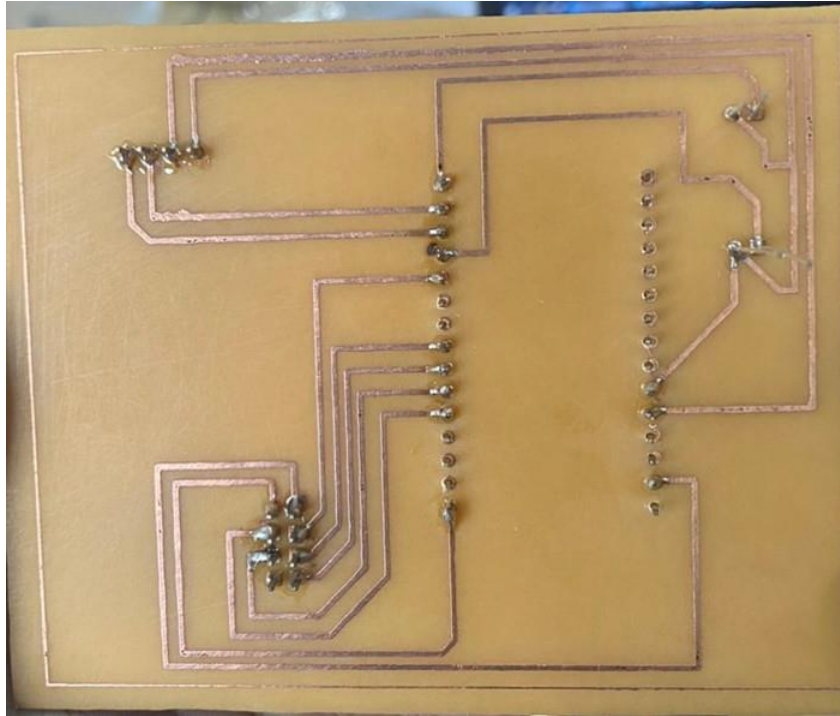


Hình 4.2. Mặt sau khối xử lý thứ cấp

4.2. Khối xử lý trung tâm (Gateway)



Hình 4.3. Mặt trước của khối xử lý trung tâm



Hình 4.4. Mặt sau của khối xử lý trung tâm

4.3 Kiểm tra và đánh giá

Sau khi hoàn thành các bước trên tiến hành đến việc cho hệ thống hoạt động.

- Cảm biến MQ2 phản hồi chính xác và nhanh chóng khi có khí gas đưa vào, đáp ứng được khi xảy ra hiện tượng rò rỉ khí gas trong thực tế. Sau khi nồng độ khí gas trong không khí vượt ngưỡng an toàn thì NODE sẽ có đèn, còi báo động và gửi tín hiệu và giá trị khí gas cho bộ phận trung tâm.
 - Tại bộ phận trung tâm (GATEWAY) sẽ hiển thị các thông số trên màn hình OLED về giá trị gas ở từng NODE và cho biết tình trạng về nồng độ khí gas ở đó. Khi nhận được tín hiệu cảnh báo khí gas vượt ngưỡng an toàn từ NODE, sẽ có đèn báo động ở vị trí NODE nào xảy ra sự rò rỉ khí gas và hiển thị NGUY HIỂM ở NODE đó trên màn hình OLED.
- Tổng kết hệ thống cảnh báo rò rỉ khí gas hoạt động đáp ứng được yêu cầu đề ra cho hệ thống.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN

Với những mục tiêu được đề ra, nhóm em đã hoàn thành thiết kế và thi công sản phẩm hệ thống cảnh báo khí gas khu vực thứ cấp và khu vực trung tâm đơn giản, dễ sử dụng và an toàn giúp mọi người dễ nắm bắt được tình hình và đưa ra những quyết định hành động chính xác và kịp thời giúp tránh né được những rủi ro có thể xảy ra.

Tuy nhiên, hệ thống nhóm em vẫn còn nhiều thiếu sót như việc người dùng không có ở nhà khó có thể theo dõi được tình hình, hệ thống không có tính năng chữa cháy tự động và không có sự điều khiển từ xa....

5.2. HƯỚNG PHÁT TRIỂN

Từ những hạn chế của đề tài, nhóm em xin đưa ra một số ý kiến về hướng phát triển của đề tài trong tương lai nhằm giúp đề tài được hoàn chỉnh và hoạt động hiệu quả hơn:

- Phát triển Web hiển thị và điều khiển báo động.
- Gửi giá trị gas lên database và liên kết qua app giúp người dùng dễ dàng theo dõi và quản lý hơn.
- Sử dụng Module Sim900A Mini để gửi thông tin thông qua nhắn tin và gọi điện đến người dùng khi có sự cố nguy hiểm.
- Thiết kế một hệ thống tự động chữa cháy khi ngưỡng khí gas vượt mức an toàn.

TÀI LIỆU THAM KHẢO

[1] Điện tử Hello, “ARDUINO VỚI HỆ THỐNG GIÁM SÁT VÀ BẢO ĐỘNG KHÍ GAS”, dientuhello.com, 2020.

<https://www.dientuhello.com/arduino-voi-he-thong-giam-sat-va-bao-dong-khi-gas/>

[2] Nshopvn,” Cảm Biến Khí Gas (LPG/CO/CH4) MQ-2”, nshopvn.com, 2019.

<https://nshopvn.com/product/cam-bien-khi-gas-mq-2/>

[3] Nguyen Manh Hung,” Sử dụng Module NRF24L01 - Thu phát sóng vô tuyến 2.4GHz với Arduino”, arduino.cc, 2015.

<http://arduino.vn/bai-viet/562-su-dung-module-nrf24l01>

[4] loc4atnt,”Giới thiệu màn hình OLED 0.96 inch I2C”, arduino.cc, 2017.

<http://arduino.vn/bai-viet/1503-gioi-thieu-man-hinh-oled-096-inch-i2c>

[5] Huỳnh Nhật Tùng,” NRF24L01 GIAO TIẾP ARDUINO, THU PHÁT RF NRF24L01 2.4GHZ + LCD1602”, huynhnhattung.com, 2022.

<https://huynhnhattung.com/nrf24l01-giao-tiep-arduino-thu-phat-rf-nrf24l01-2-4ghz-lcd1602/>