Giới thiệu.

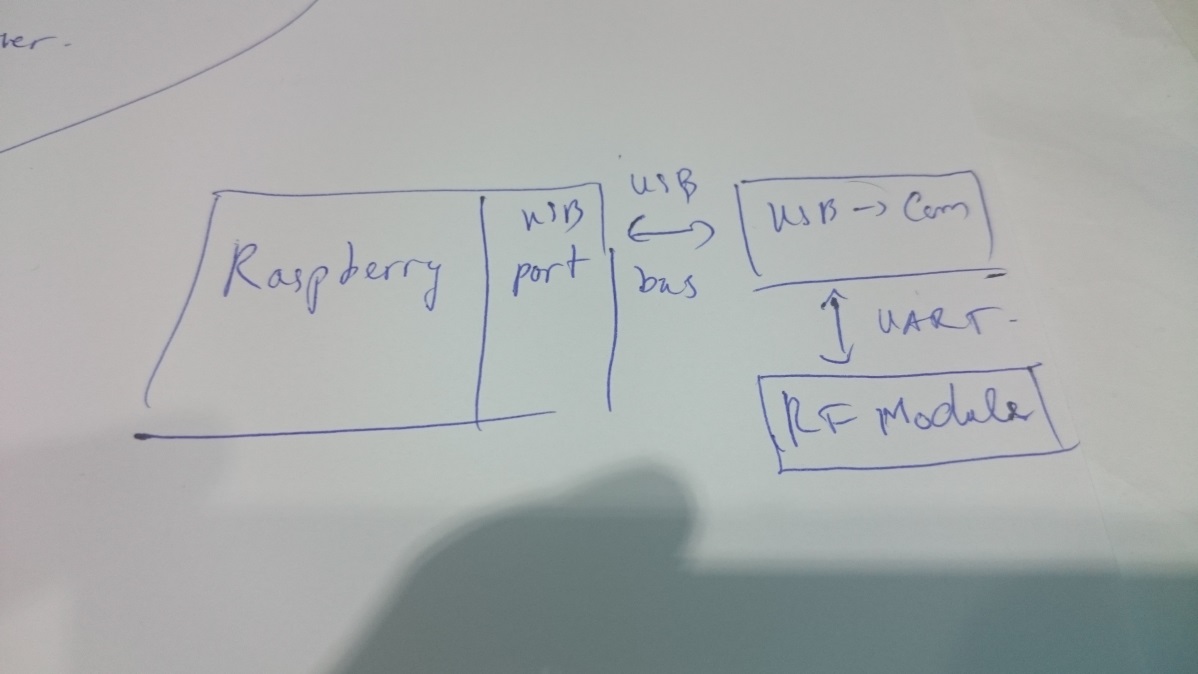
Lời cảm ơn.

1. Mục lục.
2. Giới thiệu đề tài.
3. Cơ sở lý thuyết.
4. Vi điều khiển STM32F103C8T6.
5. Cảm biến nhiệt độ DS18B20.
6. Cảm biến ánh sáng TEPT57000.
7. Cảm biến ga – khói MQ-02.
8. Raspberry Pi 2.
9. SQLite database.
10. Web server.
11. PHP language.
12. Mục tiêu và giới hạn.
13. Hiện thực.
14. Phần cứng.
    1. Board mở rộng cho Raspberry Pi 2.

Vì board Raspberry Pi không hỗ trợ 2 giao tiếp RF và RS485 nên chúng ta cần hiện thực trên board mở rộng.

1. Hiện thực mở rộng giao tiếp RF.

RF hiện thực theo sơ đồ sau:

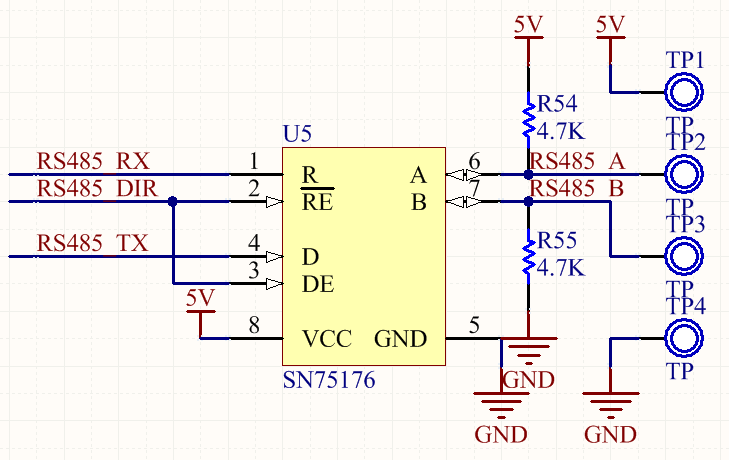


Module RF NRF24L01 được chuyển đổi giao tiếp từ SPI sang UART thông qua MCU STM32F103C8T6. Giao tiếp UART kết nối với Raspberry Pi 2 thông qua module chuyển đổi USB to COM.

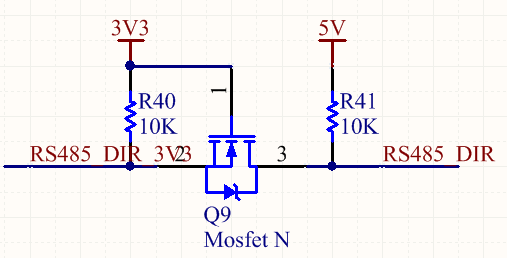


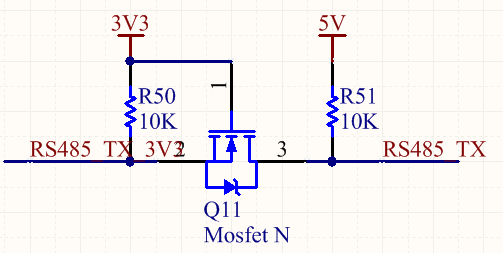
1. Hiện thực mở rộng giao tiếp RS485.

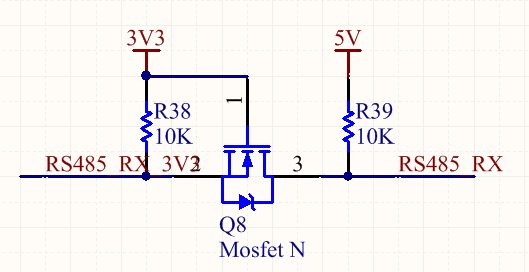
UART trên board Raspberry Pi 2 kết nối với IC chuyển đổi UART 🡪 RS485 SN75176



Tuy nhiên mức logic hoạt động trên Raspberry PI 2 và RS485 khác nhau, do đó ta cần sử dụng một mạch chuyển đổi mức điện áp để cả hai có thể hoạt động ổn định.

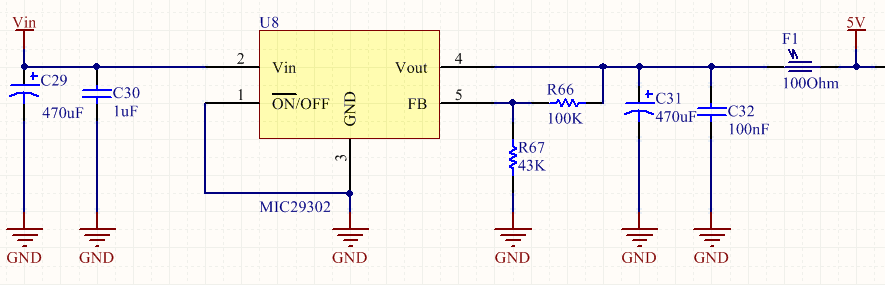






1. Hiện thực hệ thống nguồn nuôi.

Board Raspberry sử dụng 2 nguồn 5V và 3V3, interface RS485 sử dụng nguồn 5V. Nguồn 3V3 được hệ thống nguồn trên board Raspberry tạo ra sử được tận dụng trong mạch chuyển logic. Do đó chúng ta chỉ cần cung cấp nguồn 5V cho cả Raspberry Pi 2 và module mở rộng.



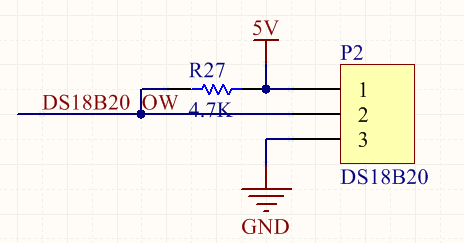
Sử dụng IC nguồn MIC29302 kết hợp với bộ lọc nhiễu ta có mạch nguồn với đầu vào từ 9V đến 15V, đầu ra 5V và dòng cao nhất 3A.

* 1. Board Sensors Host.

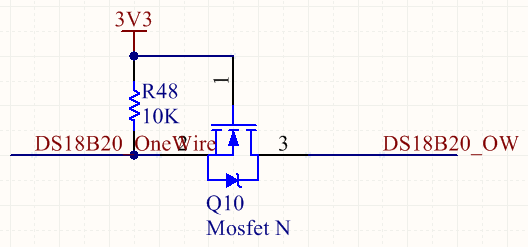
Sensor Host board là board chuyên trách đọc và gửi đi dữ liệu cảm biến, đưa ra cảnh báo trực tiếp thông qua hệ thống loa và rờ-le.

1. Cảm biến nhiệt độ DS18B20.

Cảm biến nhiệt độ DS18B20 sử dụng giao tiếp OneWire đơn giản. Mạch kết nối chỉ sử dụng một trở kéo lên 4K7 Ohm.

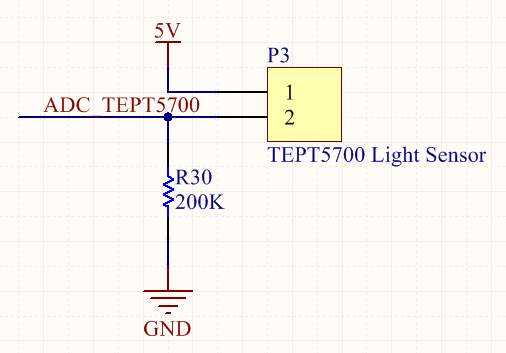


Tuy nhiên, do khác biệt mức logic trên vi điều khiển STM32F103C8T6 nên ta cần thêm mạch chuyển đổi logic sau.



1. Cảm biến ánh sáng TEPT57000.

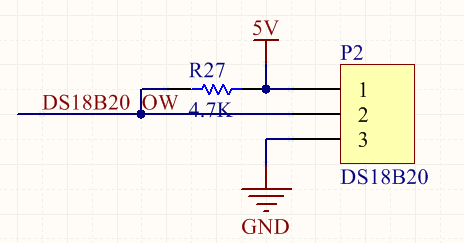
Cảm biến ánh sáng TEPT57000 có cấu trúc giống một photo transistor nên mạch đọc khá đơn giản. Sử dụng một trở tạo áp để đo dòng chạy qua cảm biến khi có ánh sáng.



Điện áp ra từ 0V đến 5V vượt ngưỡng đo của bộ ADC trong STM32F103C8T6, do đó ta sử dụng thêm Op-amp với mục đích chia áp và ngăn dòng từ cảm biến. Mạch Op-amp sẽ được giới thiệu ở mục sau.

1. Cảm biến ga – khói MQ-02.

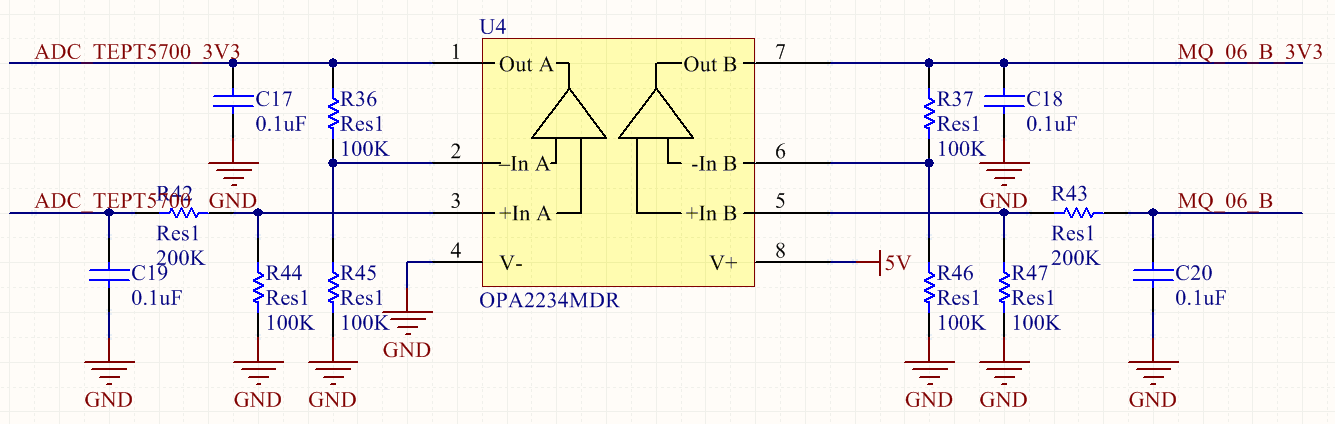
Cảm biến ga khói hoạt động gần giống như một biến trở, sơ đồ mạch đơn giản như sau.



Do điện áp hoạt động là 5V và điện áp ngõ ra có thể đạt ngưỡng 5V nên cần sử dụng thêm Op-amp để đảm bảo bộ ADC vi điều khiển hoạt động tốt.

1. Mạch Op-amp.

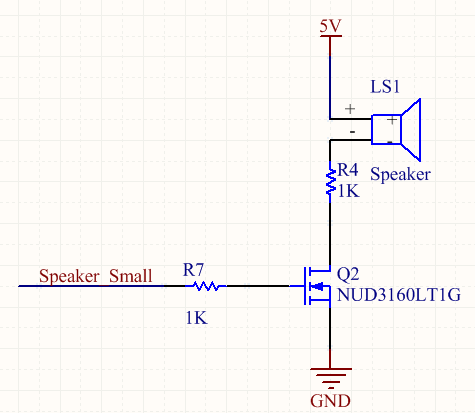
Do sự khác biệt của điện áp ra của cảm biến và ngưỡng đo của ADC vi điều khiển cũng như ngăn dòng từ cảm biến vào vi điều khiển để đảm bảm an toàn mạch, ta sử dụng thêm mạch Op-amp.

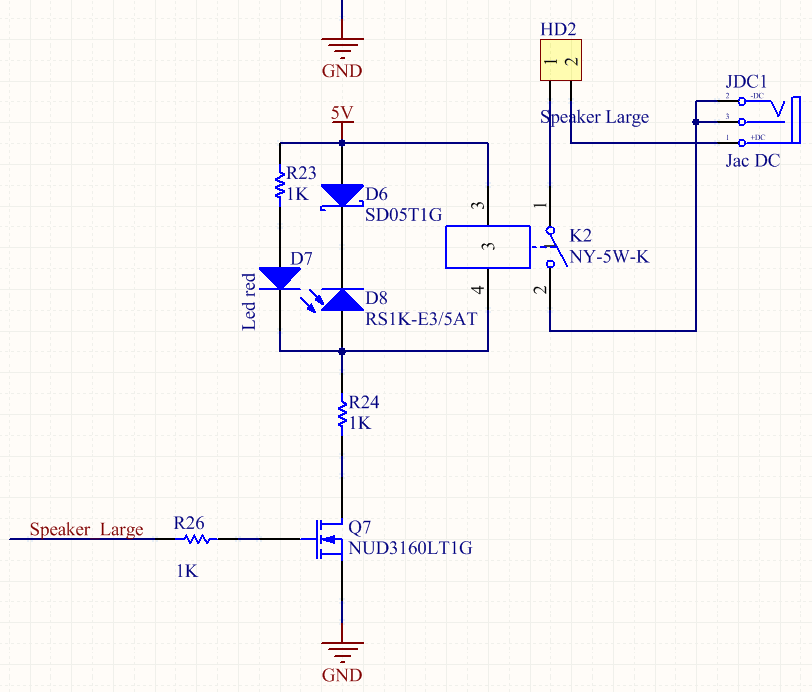


Điện áp ngõ ra được tính theo công thức đã được nêu ở mục III.

1. Khối loa, rờ-le.

Ngoài chức năng đọc cảm biến, board Sensor Host cần phải đưa ra cảnh báo, cảnh báo được đưa ra khi giá trị cảm biến cho biết môi trường có những tính chất vượt ngưỡng an toàn. Cảnh báo được thực hiện bằng âm thanh và tin nhắn. Board Sensor Host tích hợp sẵn một loa nhỏ, tuy nhiên có hỗ trợ 2 rờ-le hoạt động tương tự để kích hoạt hệ thống loa lớn hơn.





Ngoài ra, board Sensor Host còn tích hợp sẵn module Sim900 hỗ trợ việc cảnh báo bằng tin nhắn. Tuy nhiên, để giảm chi phí, module này có thể không được tích hợp.

1. Phần mềm.
   1. Protocol giao tiếp.
2. Định dạng các gói truyền nhận.
3. Quy trình trao đổi dữ liệu.
4. Quy trình cập nhật Sensor Host mới.
   1. Raspido.

Raspido là tên phần mềm đảm nhận các chức năng sau:

* Thu thập dữ liệu cảm biến thông qua hệ thống bus RS485 và RF.
* Lưu trữ dữ liệu vào database.
* Xử lí dữ liệu, hiệu chỉnh các ngưỡng cảnh báo.

Do hệ thống có nhiều board Sensor Host kết nối và sử chung đường truyền nên Raspido cần xử lý các vấn đề sau:

* Đảm bảo tất cả các Sensor Host được kết nối và truyền nhận dữ liệu.
* Giải quyết tranh chấp trên đường truyền.

Để giải quyết các vấn đề trên, Raspido được thiết kế theo mô hình nhiều luồng (thread) hoạt động theo nguyên tắt sau:

* Mỗi thread xử lý giao tiếp với một Sensor Host duy nhất.
* Dữ liệu được lưu trữ tập trung trên một database.
* Các tranh chấp được giải quyết bằng mutex.
* Thread hoạt động độc lập về thời gian, không theo một cấu trúc tuần tự nào.
  1. Web server.

1. Đánh giá kết luận.
2. Tài liệu tham khảo.
3. Phụ lục.