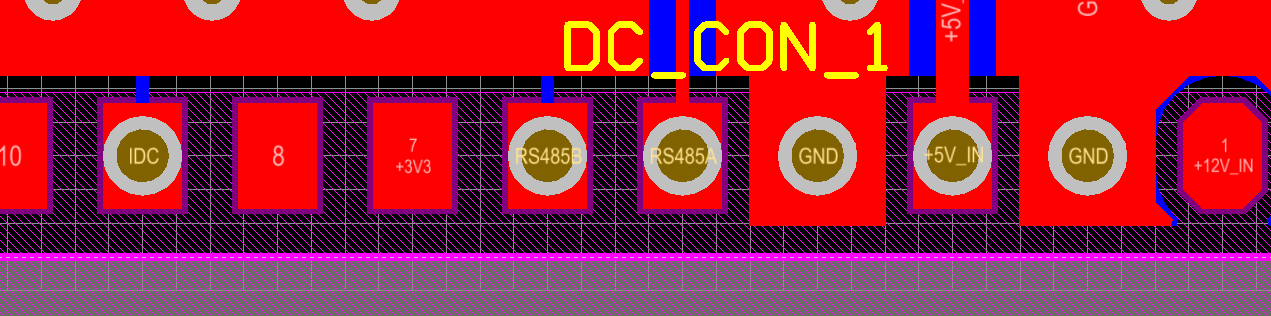
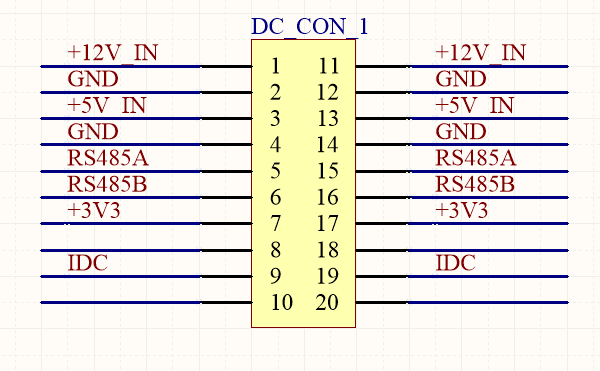
**Yêu cầu đối với firmware :**

* Viết code rõ ràng, có comment chú thích đầy đủ, mô tả các hàm, biến. Ko đặt các tên biến vô nghĩa kiểu int a => Biến, hàm cần khai báo int temp, doam…
* Cấu hình USART với các module là **115200-8-1** ( đối với AVR để tốc độ ở chế độ x2)

**Mô tả hệ thống :**

Hệ thống gồm nhiều bo mạch kết nối với nhau qua bus dữ liệu, power chính



Khi 1 trong các bo mạch truyền bản tin, tất cả cá bo mạch khác đều nhận được bản tin.

Bo mạch trung tâm là bo mạch CPU: Thu thập số liệu, phân luồng dữ liệu, điều khiển. CPU lần lượt hỏi các module và chờ 1 khoảng thời gian để respon, nếu ko respon thì module đó có vấn đề. OK

Khi máy tính nối với thiết bị để cấu hình, máy tính phải đóng vai trò là slave. Khi CPU hỏi, máy tính mới trả lời. Nếu máy tính trả lời, CPU tách dữ liệu ra ( dữ liệu cấu hình từ PC). Nếu PC ko trả lời thì kệ. Timeout. PC thu thập các bản ti để hiển thị lên phần mềm trên máy tính.

Mạch CPU điều khiển mạch FAN : Bật tắt 1 trong 4 quạt. Nếu có quạt nào hỏng, mạch FAN báo lại cho mạch CPU biết.

Mạch CPU điều khiển mạch Relay: Mạch CPU truyền bản tin bật tắt các relay trong mạch relay theo lệnh, mạch relay thực hiện. Mạch relay là mạch ko truyền gì, chỉ thực thi lệnh từ CPU.

Mạch CPU thu thập dữ liệu từ mạch đo ADC, khi mạch CPU gửi request lấy giá trị ADC, mạch ADC phải phản hồi dữ liệu ADC mà bo mạch ADC thu thập được từ 8 đầu vào. Khi có sự kiện ( giá trị ADC nằm ngoài ngưỡng MAX, MIN) mạch ADC chủ động gửi bản tin đi. Trong bản tin request của CPU có thể chứa bản tin cấu hình ngưỡng (MAX, MIN) hoặc bản tin cấu hình ngưỡng riêng.

**Nguyên tắc truyền bản tin:**

Chân IDC là chân dùng để báo hiệu và kiểm tra xem bus dữ liệu có đang bận hay không. Cách tryển bản tin:

* Check bus dữ liệu rảnh
* Đợi bus dữ liệu rảnh
* Truyền bản tin
* Ví dụ :

While(IDC==1);// Đợi bus dữ liệu rảnh

IDC=1;// lấy quyền truyền dữ liệu

Data\_transmit(data);// Truyền dữ liệu

IDC=0;//Trả lại bus dữ liệu rảnh

Cú pháp bản tin có dạng như sau:

[ký tự bắt đầu][Nội dung dưới dạng HEX][check\_sum][ký tự kết thúc]

Trong dấu [] là 1 byte.

Ví dụ

“>OMC 0x01 0x02 0xAA 0x34<”

Như vậy ‘>’ là ký tự bắt đầu, ‘<’ là ký tự kết thúc, [0x34] dữ liệu check\_sum, “OCM” là ký tự nhận dạng, [0x01][0x02] là dữ liệu.

**Nguyên tắc nhận dữ liệu:**

* Có ngắt USART để báo có dữ liệu đến
* Thuật toán nhận dữ liệu :

+ Có dữ liệu đến, lưu dữ liệu

+ biến thời gian đếm

+ Time out => Đủ dữ liệu => Cờ xử lý dữ liệu

+ Main task xử lý bóc tách dữ liệu

* Lưu dữ liệu và xử lý dữ liệu nhân được với khung truyền như được mổ tả ở trên
* Bóc tách dữ liệu, phục hồi dữ liệu
* Thực hiện lệnh nếu có

Ví dụ nhận được bản tin “>OMC 0x01 0x02 0xAA 0x34<”

Kiểm tra bản in đúng cú pháp ? ký tự bắt đầu, ký tự nhận dạng, ký tự kết thúc, check\_sum => nếu tất cả hợp lệ => data1 = 0x01; data2=0x02 hoặc data= (0x01<<8) + 0x02;

**Yêu cầu đối với mạch đo giá trị ADC: MCU Atmega88PA**

* Mạch có sơ đồ nguyên lý theo file pdf đính kèm
* Mạch tự động và luôn luôn đo giá trị ADC 10bit theo các kênh 0-7 lưu vào 1 mảng int adc\_data[8]
* Nếu có lệnh **request** từ bo mạch CPU có dạng :

“>CPU[mã bo DC]r[checksum]<”

Trong đó ‘r’ nghĩa là câu lệnh request

[mã bo DC] 1 mã tự định nghĩa để biết là CPU đang hỏi mạch nào [0x0A-19]

Nếu check đúng là lệnh từ CPU, đúng mã số của mạch DC >> gửi dữ liệu lại cho mạch CPU

Ví dụ :

‘>DCM[adc0\_byte\_cao][ adc0\_byte\_thap]…[ adc7\_byte\_cao][ adc7\_byte\_thap][check\_sum]<’

* ~~Có các ngưỡng ADC MAX, MIN lưu trong eeprom~~
* Nếu ADC[x] vượt qua ngưỡng => gửi cảnh báo qua mạng truyền thông bằng cách gửi bản tin

‘>DCM[adc0\_byte\_cao][ adc0\_byte\_thap]…[ adc7\_byte\_cao][ adc7\_byte\_thap][check\_sum]<’

**Yêu cầu đối với mạch điều khiển quạt: MCU Atmega88PA**

* Mạch có sơ đồ nguyên lý theo file pdf đính kèm
* Nếu có lệnh **control** từ bo mạch CPU có dạng :

“>CPU[control\_byte][mã bo FAN][checksum]<”

Trong đó [control\_byte] chứa lệnh điều khiển

[mã bo FAN] 1 byte mã tự định nghĩa để biết là CPU đang hỏi mạch nào

Nếu check đúng là lệnh từ CPU, đúng mã số của mạch FAN >> bật quạt tương ứng

4 bit thấp của [control\_byte] tương ứng với 4 quạt

Khi quạt chạy đo dòng tương ứng, nếu có bất thường, gửi bản tin cảnh báo quạt hỏng

Ví dụ :

‘>FAN[cảnh báo quạt hỏng byte][check\_sum]<’

**Yêu cầu đối với mạch điều khiển relay: MCU Atmega88PA**

* Mạch có sơ đồ nguyên lý theo file pdf đính kèm
* Nếu có lệnh **control** từ bo mạch CPU có dạng :

“>CPU[control\_byte\_1] [control\_byte\_2] [mã bo Relay][checksum]<”

Trong đó [control\_byte\_1] [control\_byte\_2] chứa lệnh điều khiển

[mã bo Relay]là 1 byte mã tự định nghĩa để biết là CPU đang hỏi mạch nào

Nếu check đúng là lệnh từ CPU, đúng mã số của mạch relay >> bật relay tương ứng

10 bit thấp của [(control\_byte\_1<<8)+ control\_byte\_2] tương ứng với 10 relay

Module này không có phản hồi gì về CPU