



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه دوم

Mod-bus

نام اعضای گروه:

مینا حیدری فراهانی - ۹۷۲۳۱۱۷

علی مطلبی - ۹۷۲۳۰۸۵

مازیار دادور - ۹۷۲۳۰۲۹

میلاد مرادی بید هندی - ۹۷۲۳۰۸۴

زمستان ۱۴۰۱

مقدمه

هدف از انجام این پروژه برقراری ارتباط یک مستر با دو عدد slave به وسیله ی mod-bus است. با استفاده از مد باس می‌خواهیم که وضعیت یک رله (slave 1) را تغییر دهیم و دما را از سنسور (slave 2) بخوانیم. ۳ عدد کد با آردوینو برای master و slave ها نوشته شده است که هر کدام جداگانه توضیح داده میشوند.

:Master

هر slave دارای یک آدرس از ۱ تا ۲۴۷ می‌باشد. زمانی که master درخواست اطلاعات می‌کند، اولین بایت پیغام به عنوان آدرس slave فرستاده می‌شود. بدین صورت slave متوجه می‌شود که پیام ارسال شده برای آن است یا خیر. برای slave ها آدرس 0x07 و 0x05 را قبلاً انتخاب کردیم.

دستوراتی که مستر ارسال میکند به ترتیب و به صورت زیر ارسال میشود.

در تابع زیر ابتدا آدرس slave مورد نظر، سپس function code مربوط به read coil و سپس شماره coil مورد نظر روی پورت سریال (Serial2) ارسال میشود.

```
void read_coil(uint16_t address, uint16_t data)
{
    Serial2.write(address);

    Serial2.write(0x01);
    |
    Serial2.write(data);
}
```

در اینجا هم مانند بالا ابتدا آدرس slave مورد نظر، سپس function code مربوط به write coil و سپس شماره coil مورد نظر (p) به همراه دستور روشن یا خاموش کردن (pm) روی پورت سریال ارسال میشود.

```
void write_coil(uint16_t address, uint16_t p, uint16_t pm)
{
    Serial2.write(address);

    Serial2.write(0x05);

    Serial2.write(p);

    Serial2.write(pm);
}
```

در لوپ اصلی master دائما از طريق پورت سريال (serial1) چك ميكند كه دستوري داده شده است يا خير و مطابق با آن كار هاي زير انجام ميشود:

خواندن پایه ۲۲ که مربوط به رله ۱ است --> 'C'

خواندن پایه ۲۳ که مربوط به رله ۲ است --> 'V'

خواندن درجه حرارت از 'M' --> Slave 2

۱ کردن پایه ۲۲ --> 'W'

• کردن پایه ۲۲ --> 'Q'

• کردن پایه ۲۳ --> 'E'

۱ کردن پایه ۲۳ --> 'R'

در ادامه Response های slave روی صفحه کامپیوتر دیده میشود.

```
if (Serial2.available())
{
    Serial.println(Serial2.read());
    Serial.println(Serial2.read());
    Serial.println(Serial2.read());
    Serial.println(Serial2.read());
    Serial.println("jjjjjjjjjjjjjjjjjjjjjjjj");
}
```

Slave 1 (relays)

آدرس این 0x05 slave است. برای دسترسی به وضعیت ۴۰ پایه از آرایه n[40] استفاده کردیم.

```
int n[40];
#define adr 0x05

void read_coil(int r)
{
    Serial.println("CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC");
    delay(5);
    Serial12.write(adr);
    Serial12.write(0x01);
    Serial12.write(n[r]);
}
```

در تابع بالا مقدار ۲ که پایه رله است گرفته شده است و وضعیت آن خوانده میشود.

```
void write_coil(int p , int out)
{
    Serial.println("XXXXXXXXXXXXXXXXXXXXXXXXXXXX");

    digitalWrite(p,out);
    n[p]=out;
    Serial2.write(adr);
    Serial2.write(0x05);
    if(p==22)
        Serial2.write(0x16);
    else
        Serial2.write(0x17);
    if(out==0)
        Serial2.write(0x00);
    else
        Serial2.write(0x01);
}
```

در تابع write_coil پایه مورد نظر (p) و وضعیت آن (out) به عنوان ورودی گرفته میشوند. در digitalWrite پایه مورد نظر روشن یا خاموش میشود و در n[p] وضعیت پایه که ۱ یا ۰ است نوشته میشود. در ادامه به عنوان response آدرس slave همراه با function code آن و شماره پایه و وضعیت پایه ها فرستاده میشود.

```

void loop() {
  if (Serial2.available()) {
    if (Serial2.read() == adr)
    {
      Serial.println("address");
      if (Serial2.read() == 0x01)
      {
        Serial.println("coil");
        if (Serial2.read() == 0x16)
          read_coil(22);
        else
          read_coil(23);
      }
    }
    else
    {
      if (Serial2.read() == 0x16)
      {
        if (Serial2.read() == 0x00)
          write_coil(22, 0);
        else
          write_coil(22, 1);
      }
      else
      {
        if (Serial2.read() == 0x00)
          write_coil(23, 0);
        else
          write_coil(23, 1);
      }
    }
  }
}

```

در لوپ اصلی slave درخواست ها و دستوراتی که از master آمده است بررسی میشود. Slave دائماً پورت سریال را چک میکند. در ابتدا آدرس slave چک میشود. اگر آدرس فرستاده شده برای slave 1 بود، function code بررسی میشود. اگر 0x01 باشد مربوط به read coil است و پایه ۲۲ یا ۲۳ طبق دستور مستر خوانده میشود. اگر function code مربوط به read coil نباشد دستورات write coil اجرا میشود و با توجه به اینکه دستور در مورد پایه ۲۲ باشد یا ۲۳، این دو پایه ۰ یا ۱ میشوند.

:Slave 2 (temperature sensor)

در اینجا به علت نبود سنسور دما را ۲۴ درجه (0x18) فرض کردیم. در لوپ اصلی مانند قبل آدرس slave چک میشود. اگر آدرس فرستاده شده برای slave 2 بود، function code بررسی میشود. اگر 0x04 باشد مربوط به read register است که مقدار آن خوانده میشود که ۲۴ درجه است.

```
void loop() {  
  if (Serial2.available()) {  
    if (Serial2.read() == adr)  
    {  
      Serial.println("adress");  
      if (Serial2.read() == 0x04)  
      {  
        Serial.println("readreg");  
        if (Serial2.read() == 0x01)  
          read_reg();  
      }  
    }  
  }  
}
```

** لازم به ذکر است که سرعت انتقال بیت‌ها در مد باس بسیار سریع بوده و ما از Baud rate معمول در تجهیزات یعنی ۹۶۰۰ برای ارتباط استفاده کردیم.