

دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

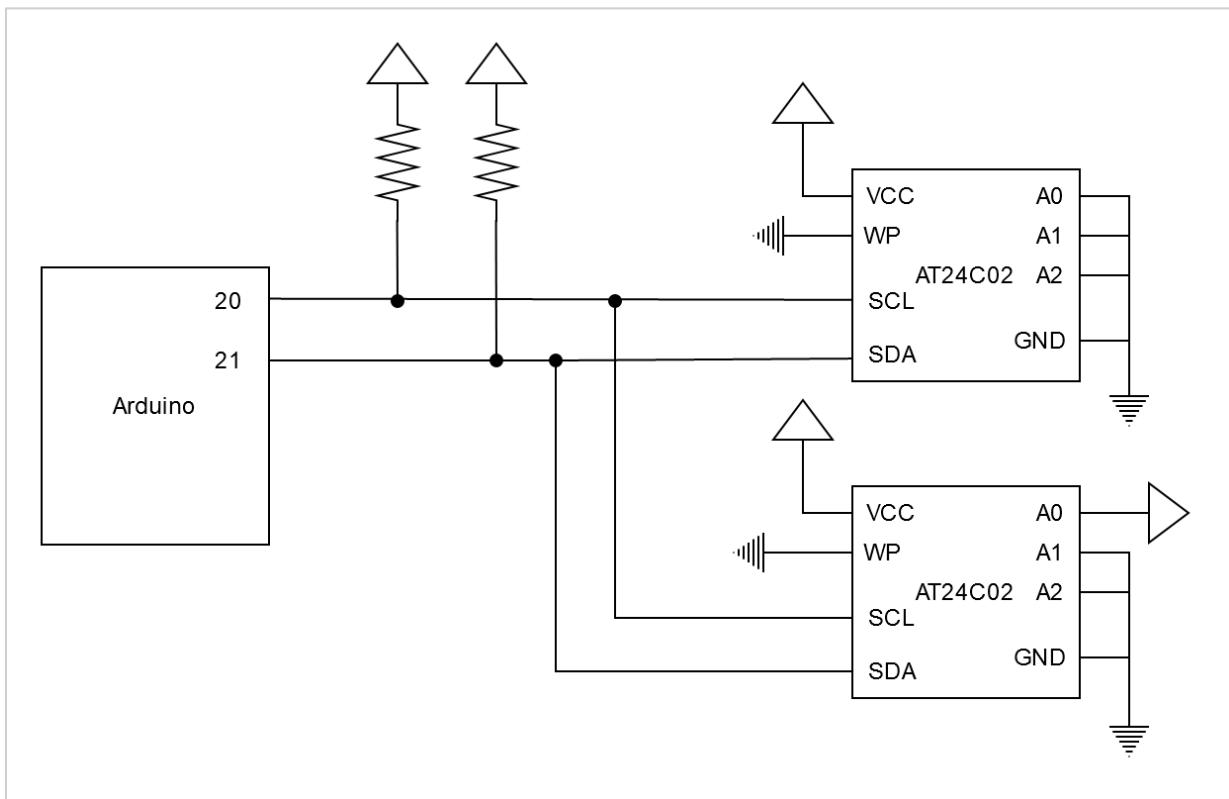
آزمایشگاه ریزپردازنده و زبان اسمنبلی

رادین شایانفر

پاییز ۱۳۹۹



- کاربردهای EEPROM و RAM دلیل استفاده به جای **FLASH**: در موقعي که نیاز است تا دادهها پس از خاموش شدن دستگاه و با قطع برق از بین نرونده لازم است که از EEPROM استفاده کنيم. دليل عدم استفاده از RAM نيز همان فرار بودن آن است که پس از هر بار قطعی برق اطلاعات آن پاک می شود. همچنان دليل استفاده نکردن از حافظه FLASH کارکرد آن به صورت **block-wise** است. در حالی که EEPROM به صورت **byte-wise** کار می کند و می توان بايت (مانند نیاز ما در اينجا) بر روی آن اطلاعات را خواند و تغيير داد. در ضمن سرعت EEPROM نيز از FLASH بيشتر است.
- نحوه نوشتمن روی FLASHها: برای اين کار از آنجا که عمل نوشتمن روی FLASHها به شكل بلاک به بلاک است، می توان هر داده را روی بلاک های مختلف نوشت. اما اين روش چندان کارا نیست. روش ديگر اين است که پيش از نوشتمن روی يك بلاک، ابتدا آن را بخوانيم، مقدارهای مورد نیاز را تغيير دهيم و سپس مجدد روی همان بلاک در FLASH بنویسيم.
- حداقل حافظه روی باس مشترك: برای اين کار به کمک دو پایه A₀ و A₁ می توان 2^2 دستگاه مختلف را آدرسدهی کرد و در نتیجه $4 \times 4\text{KB} = 16\text{KB}$ خواهيم داشت.
- شماتيک اتصال دو AT24C02 روی باس مشترك:



شکل (۱) – شماتيک اتصال دو AT24C02 روی باس مشترك



• هم خوانی فریم‌های AT24C02 و TWI:

فریم :write

Memory: start (1 bit) → device address (7 bits) → W mode (1 bit) → ACK

TWI: start (1 bit) → device (7 bits) → W mode (1 bit) → ACK

→ word address (8 bits) → ACK → data (8 bits) → ACK → ... → stop (1 bit)

→ data (8 bits) → ACK → data (8 bits) → ACK → ... → stop (1 bit)

فریم :read

Memory: start (1 bit) → device address (7 bits) → W mode (1 bit) → ACK

TWI: start (1 bit) → device (7 bits) → W mode (1 bit) → ACK

→ word address (8 bits) → ACK → stop (1 bit) → start (1 bit)

→ data (8 bits) → ACK → stop (1 bit) → start (1 bit)

→ device address (7 bits) → R mode (1 bit) → ACK → data (8 bits) → ACK → ...

→ device (7 bits) → R mode (1 bit) → ACK → data (8 bits) → ACK → ...

→ stop (1 bit)

→ stop (1 bit)

• فرکانس کلک: فرکانس کلک در برد آردوینو (master) پیکربندی می‌شود و همچنین master

مسئولیت تولید این کلک را به عهده دارد. از آنجا که برای هر بار عمل نوشتن تک بایت نیاز به ۲۹ بیت

داده داریم، سرعت نوشتن با کلک 10 KHz برابر است با:

$$\frac{10 \times 10^3}{29} \sim 344 \text{ byte/s}$$

• توابع Wire

○ ارتباط slave و master را با C² آغاز می‌کند. اگر پارامتری نداشته باشد به عنوان

master عمل می‌کند و در غیر این صورت پارامتر وارد شده شماره slave را تعیین می‌کند.

○ برای تغییر فرکانس ارتباط استفاده می‌شود.

○ ارتباط را برای شروع ارسال داده به آدرس داده شده را آغاز می‌کند.

○ داده را روی slave می‌نویسد (این تابع پس از beginTransmission() معمولاً

صدای زده می‌شود).



آزمایش هفتم

- endTransmission(): ارتباط را پایان می‌دهد.
 - requestFrom(): برای درخواست خواندن داده از slave توسط master صدا زده می‌شود.
 - available(): تعداد بایت‌هایی که آماده دریافت توسعه read() هستند را می‌دهد.
 - read(): بایت ارسال شده توسط slave را می‌خواند.
- کد تولید فریم خواندن و نوشتن:

Read:

```
wire.beginTransmission(DEVICE_ADDRESS);  
wire.write(WORD_ADDRESS);  
wire.endTransmission();  
wire.requestFrom(DEVICE_ADDRESS, SIZE);  
wire.read();
```

Write:

```
wire.beginTransmission(DEVICE_ADDRESS);  
wire.write(WORD_ADDRESS);  
wire.write(DATA);  
wire.endTransmission();
```