

پرسش: در چه کاربرد هایی EEPROM به کار برده می شود؟ چرا در اینجا حافظه RAM یا Flash را به کار نمی بریم؟ تفاوت حافظه EEPROM با RAM در چیست؟

به منظور این که بعد از قطع شدن و خاموشی دستگاه داده ها حفظ شوند و از بین نروند.

در RAM اگر قطعی داشته باشیم، اطلاعات از بین می روند. Flash نیز به صورت block-wise کار می کند در حالی که ما در اینجا نیاز به خواندن و نوشتن داده به صورت byte-wise مانند EEPROM را داریم که سرعت بیشتری هم دارد.

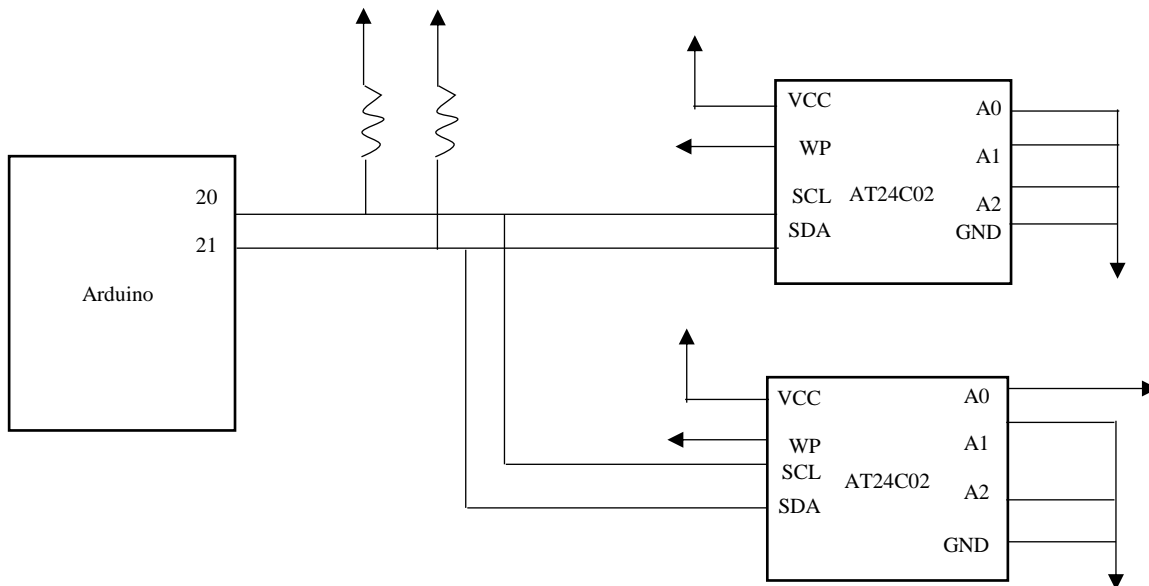
پرسش: اگر بخواهیم برای نگهداری مدهای کاری حافظه flash را به کار ببریم، فرآیند نوشتن باید چگونه انجام شود که داده های دیگری که بر روی همان بلاک هستند از دست نروند؟

Flash به صورت block-wise کار می کند که نتیجه برای این که کارآمد باشد، قبل از نوشتن روی هر بلوک آن را خوانده و تغییرات لازمه را اعمال کرده و در آخر دوباره روی همان بلوک فلش می نویسیم.

پرسش: گر یک حافظه ی EEPROM بیرونی دارای 4KB حافظه و 2 پایه آدرس باشد، در این صورت می توان حداکثر چند KB حافظه EEPROM بیرونی بر روی یک باس مشترک داشت؟

با ۲ پایه A0 و A1 میتوان ۲۴ دستگاه مختلف را آدرس دهی کرد که در نتیجه ۴*۴ (۱۶KB) خواهیم داشت.

پرسش: نمودار شماتیک برای این که دو AT24C02 را به یک باس مشترک وصل کنیم و حفاظت نوشتن غیر فعال باشد را رسم کنید. (آدرس دهی سخت افزاری دل خواه - باس را هم به پایه های میکروکنترلر متصل کنید)



پرسش: هم خوانی این دنباله فریم ها را با پروتکل TWI بررسی کنید. (فریم های آدرس و داده را مشخص کنید، دستور خواندن یا نوشتن چگونه مشخص می شوند؟)

Write frame:

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 (8bits) → ACK → ... → stop (1 bit)

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

Read frame:

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)

پرسش: فرکانس کلاک در کدام دستگاه پیکربندی می شود؟ کلاک را کدام دستگاه فراهم می کند؟ با توجه به زمان مورد نیاز برای انجام عملیات نوشتن، با فرض این که کلاک را 10KHz تنظیم کرده باشیم، در این صورت حداکثر با چه نرخ می توان عملیات نوشتن را انجام داد؟

در برد آردینو (master)

Master کلاک را فراهم می کند.

هر بار برای نوشتن هر بایت به ۲۹ بیت داده نیاز داریم. در نتیجه سزعت نوشتن با کلاک 10KHz:

$$(10 \cdot 10^3) / 29 = 344 \text{ byte/3}$$

پرسش: هر یک از تابع های نوشته شده را از راه لینک کتابخانه wire، در مستندات آردوینو بررسی کنید و کد لازم را برای تولید دنباله ی فریم ها برای عملیات نوشتن و خواندن گفته شده (با این تابع ها) بنویسید.

Begin() با I²C ارتباط بین master و slave را آغاز می کند. اگر پارامتری نداشته باشد، master است و در غیر این صورت پارامتر وارد شده، شماره slave است.

setClock(): تغییر دادن فرکانس ارتباط

beginTransaction(): آغاز ارتباط برای شروع ارسال داده به آدرس داده شده

Write(): نوشتن دیتا روی slave (بعد beginTransmission() به طور معمول صدا زده می شود)

endTransmission(): برای پایان دادن ارتباط

requestFrom(): برای خواندن داده از slave توسط master استفاده می شود.

Available(): دادن تعداد بایت های آماده دریافت توسط read()

Read(): خواندن بایت ارسال شده توسط slave

کد لازم برای تولید دنباله ی فریم ها برای عملیات نوشتن و خواندن:

نوشتن:

```
wire.beginTransaction(DRVICE_ADDRESS);  
wire.write(WORD_ADDRESS);  
wire.write(DATA);  
wire.endTransmission();
```

خواندن:

```
wire.beginTransaction(DRVICE_ADDRESS);  
wire.write(WORD_ADDRESS);  
wire.endTransmission();  
wire.requestFrom(DRVICE_ADDRESS, SIZE);  
wire.read();
```