به نام خدا

# پیش گزارش آزمایش هفتم

# آزمایشگاه میکرو پردازنده و زبان اسمبلی

سید سعید شفیعی

آبان ۱۴۰۰

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

در مواقعی که به یک حافظه ی ثابت نیاز داریم که با قطع برق اطلاعاتش از بین نرود از EEPROM استفاده می کنیم. حافظه ی RAM یک حافظه ی *volatile* است که یعنی با قطع برق اطلاعاتش از بین می رود و در اینجا استفاده نمیشود . حافظه ی Flash فرار نیست ولی برای نوشتن در آن محدودیت داریم. هم چنین برای نوشتن درون حافظه Flash نیاز به پاک کردن تمامی داده ها و درج داده از ابتدا داریم. ذخیره اطلاعات در Flash به صورت بلاک بندی شده است در صورتی که در EEPROM به صورت بایت به بایت است. به همین دلیل روش دوم سرعت بالاتری هم دارد.

از تفاوت RAM و EEPROM میتوان به فرار بودن RAM و دائمی بودن *EEPROM* اشاره کرد علاوه بر این، RAM از EEPROM سریع تر است.

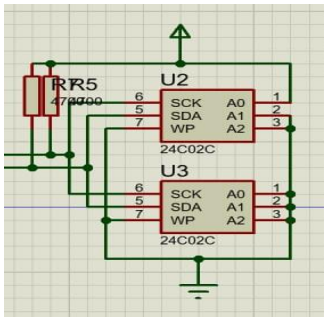
پرسش: اگر بخواهیم برای نگهداری حالت های کاری حافظه Flash را به کار ببریم، فرآیند نوشتن باید چگونه انجام شود که داده های دیگری که درهمان بلاک هستند، از دست نروند؟

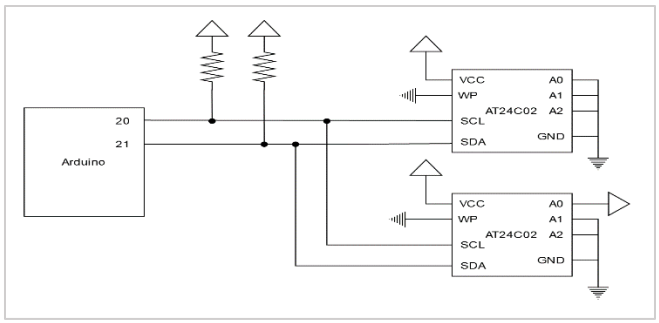
همانطور که در سوال قبل گفته شد FLASH ها با بلاکهای دیتا کار دارند پس برای تغییر داده ها بر روی آن باید ابتدا کل بلاک داده را خواند و آن جاهایی از بلاک که نیاز به تغییر دارند را تغییر داد و سپس کل بلاک را دوباره بر روی حافظه نوشت بدین صورت دیتاهای دیگر روی حافظه و بلاکها از دست نمیروند.

پرسش: اگر یک حافظه ی EEPROM بیرونی دارای KB۴ حافظه و ۲ پایه آدرس باشد، در این صورت میتوان حداکثر چند KB حافظه EEPROM بیرونی بر روی یک باس مشترک داشت؟

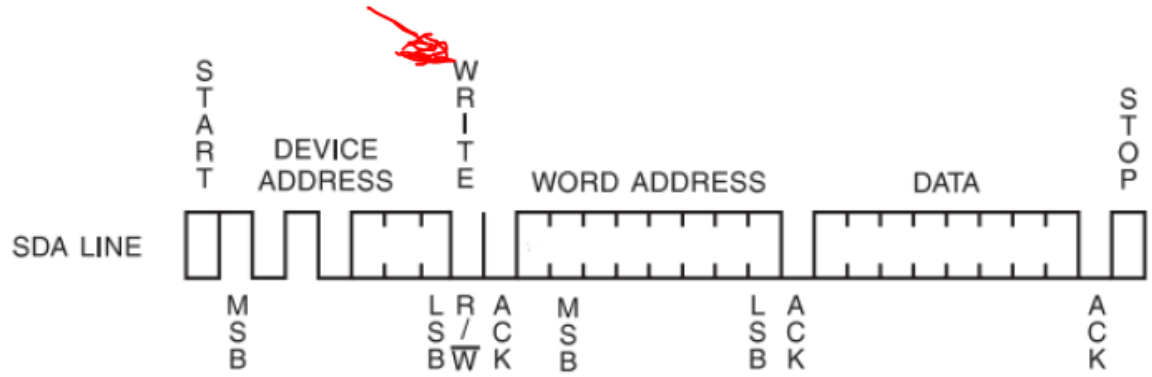
با توجه به داشتن ۲ پایه برای آدرس، می توان ۴ حالت معادل سازی انجام شود. پس ۴ نوع حافظه EEPROM می توان داشت. با فرض اینکه هر کدام از این حافظه ها 4kb ظرفیت داشته باشند، پس به طور کل 16kbحداکثر مقدار حافظه ماست.

پرسش: نمودار شماتیک برای اینکه دو AT24C02 را به یک باس مشترک وصل کرد و حفاظت نوشتن غیر فعال باشد را رسم کنید.(آدرس دهی سخت افزاری دلخواه - همچنین باس را به پایه های میکروکنترلر نیز متصل کنید.

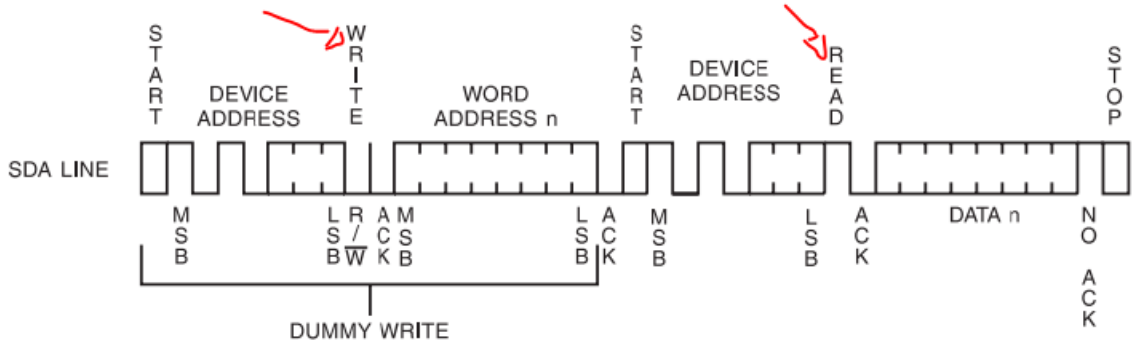


پرسش: هم خوانی این دنباله فریم ها را با پروتکل TWI بررسی کنید. ( فریم های آدرس و داده را مشخص کنید. دستور خواندن یا نوشتن چگونه مشخص می شوند؟)




برای نوشتن، تک بیت ابتدایی برای start داریم. پس از آن ۷ بیت آدرس دستگاه مدنظر ارسال می شود. سپس نوع عملیات که در اینجا خواندن است و همانطور که مشخص است برای خواندن بیت صفر ارسال می شود. سپس یک بیت ack ارسال می شود از سمت دریافت کننده. سپس آدرس داده درون حافظه ارسال می شود. باز هم یک بیت ack ارسال می شود. در نهایت داده ارسال شده و با یک بیت ack و بیت پایانی stop به عملیات خاتمه می دهیم.




در عملیات خواندن تفاوتی وجود دارد و آن این است که ابتدا آدرسی که می خواهیم بخوانیم بر روی باس می نویسیم و سپس به صورت مشابه فرآیند خواندن اطلاعات را آغاز می کنیم.

پرسش: فرکانس کلاک در کدام دستگاه پیکر بندی می شود؟ کلاک را کدام دستگاه فراهم می کند؟ با توجه به زمان مورد نیاز برای انجام عملیات نوشتن، با فرض اینکه کلاک را 10khz تنظیم کرده باشیم، در این صورت حداکثر با چه نرخی می توان عملیات نوشتن را انجام داد؟

همانطور که مشخص است پیکر بندی و تولید کلاک به عهده Master در اینجا بورد آردوینو می باشد.

با توجه به شماتیک های بالا برای نوشتن یک بایت داده ، نیاز به ۲۹ کلاک یا بیت مختلف برای ارتباط کامل داریم. با توجه به سرعت 10khz کلاک :

$$\frac{10 * 10^3}{29} bps$$

پرسش: بررسی توابع کتابخانه Wire و نوشتن سینتکس کد لازم برای تولید فریم ها

| تابع | توضیحات |
|---|---|
| Begin() | شروع ارتباط I2C، اگر آرگومان ورودی دهیم، آدرس به عنوان slave است در غیر این صورت یعنی بدون آرگومان master است. |
| SetClock() | برای تنظیم فرکانس کلاک در ارتباط I2C. |
| BeginTransmission() | انتقال اطلاعات با آدرس slave ای که در ورودی به آن داده ایم. |
| Write() | دیتا را ازslave در پاسخ به درخواست master میفرستد. یا اینکه بایت ها را برای انتقال از master به slave در صف می گذارد. |
| EndTransmission() | انتقال را پایان می دهد، اگر بایتی در صف ارسال باشند، ابتدا ارسال آنها را انجام می دهد و سپس پایان ارتباط. |
| RequestFrom() | Master برای درخواست دیتا از slave این دستور را صدا می زند. در ورودی دستور، آدرس device و سایز و تعداد بایت های را می دهیم. |
| Available() | Master بعد از صدا زدن دستور requestFrom با استفاده از این دستور می تواند تعداد بایت های آماده برای خواندن را بگیرد. |
| Read() | یک بایت که بر روی باس نوشته شده می خواند و بر می گرداند. |

**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();
```