

گزارش کار آزمایش هفتم MICRO

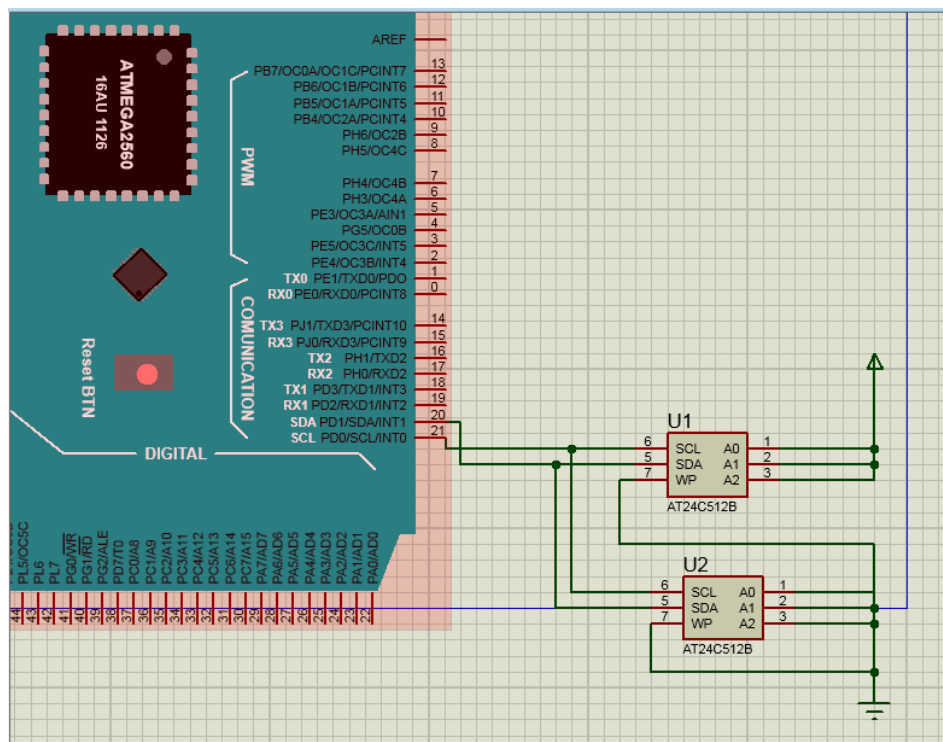
۱. EEPROMها حافظه‌هایی عموماً با گنجایش پایین هستند، که با قطع جریان برق داده‌ی ذخیره شده را حفظ می‌کنند. از کاربردهای آن‌ها می‌توان به ذخیره‌ی تنظیمات (Configurations) دستگاه‌ها اشاره کرد. کاربر رایج دیگر EEPROMها ذخیره‌ی فایل BIOS کامپیوترهای شخصی می‌باشد. Flashها هم نوعی از EEPROM هستند. مهم‌ترین تفاوت Flash و EEPROM این است که در EEPROMها داده‌ها به صورت بایت به بایت نوشته یا پاک می‌شوند، در حالی که Flashها با بلاک‌های داده کار می‌کنند. پس از Flashها هم می‌توان در این جا استفاده کرد. در نهایت علت عدم استفاده از RAM در چنین کاربردهایی این است که این قطعات به صورت فرّار (Volatile) هستند؛ در صورتی که در کاربرد فعلی، ما نیاز داریم که با خاموش شدن دستگاه ماشین لباس‌شویی، تنظیمات آن همچنان ذخیره بمانند.

۲. روند کلی نوشتن روی یک بلاک از Flash Memory به این صورت است که ابتدا طی فرآیند erase، تمام بیت‌های آن بلاک، مقدار 1 می‌گیرند، به عبارت دیگر تمام بیت‌های بلاک را شارژ می‌کنیم. سپس بیت‌هایی که نیاز است 0 باشند را دشارژ می‌کنیم. پس قاعده‌ی کلی Flash Memoryها این است که شارژ کردن را باید تنها به طور یک‌جا روی کل بلاک، و دشارژ کردن را باید بیت به بیت انجام داد. برای نوشتن یک بایت، در صورتی که بلاک مورد نظر دارای بایت خالی (8 بیت با مقدار 1) داشته باشد، به راحتی کافیست که بیت‌های مورد نیاز آن بایت را صفر کنیم. اما در صورتی که یک بلاک خالی نباشد، و بخواهیم داده‌ی یک بایتی جدید را جایگزین یکی از بایت‌های از پیش نوشته شده‌ی بلاک کنیم، باید از روال copy-modify-erase-write استفاده کنیم؛ یعنی ابتدا داده‌های آن بلاک را در یک جای دیگر کپی کنیم، سپس بلاک را erase کنیم. پس از آن بایت جدیدی که می‌خواهیم بنویسیم را با صفر کردن مقادیر مورد نیاز، ایجاد می‌کنیم. در نهایت برای بازبایی بایت‌های دیگر آن بلاک، دوباره بیت‌های مورد نیاز را (طبق مقادیر کپی شده) به صفر برمی‌گردانیم.

۳. با دو پایه‌ی آدرس حداکثر می‌توان چهار قطعه را آدرس‌دهی کرد. چون هر قطعه هم 4_{KB} حافظه دارد، پس کل حافظه‌ی بیرونی‌ای که خواهیم داشت برابر خواهد بود با:

$$4 * 4_{KB} = 16_{KB}$$

۴. مدار به صورت مقابل خواهد بود:



۵. دنباله فریم‌های داده شده، دقیقاً طبق پروتکل I2C (start, address, r/w, ack, data, ack...stop) هستند.

۶. فرکانس کلاک در دستگاه master (در این جا میکروکنترلر) تعیین و تولید می‌شود. با فرض آن که کلاک را روی 10_{KH} تنظیم کرده باشیم، حداکثر نرخ نوشتن برابر خواهد بود با:

$$\frac{10_{KH}}{29} \times 8$$

طبق شکل صفحه‌ی ۳۳، برای ارسال هر بایت داده، ۲۹ کلاک نیاز است.

۷. کدهای آزمایش به پیوست ضمیمه شده‌اند. بررسی توابع داده شده:

- `begin()`: برای اضافه شدن دستگاه به شبکه‌ی I2C استفاده می‌شود. در صورت داشتن پارامتر، آن پارامتر آدرسش به عنوان slave خواهد بود. در غیر این صورت به عنوان master اضافه می‌شود.
- `setClock()`: در صورت master بودن دستگاه در شبکه، فرکانس ارتباط را تنظیم می‌کند.
- `beginTransaction()`: یک آدرس به عنوان ورودی می‌گیرد، ارتباط با آن آدرس را شروع می‌کند. یعنی ۱ بیت شروع و ۷ بیت آدرس می‌فرستد.
- `write()`: داده‌ی گرفته شده به عنوان ورودی را ارسال می‌کند. ورودی‌اش می‌تواند یک مقدار عددی، یک string، یا یک آرایه باشد. در صورت فرستادن آرایه، باید سایز آرایه هم به عنوان ورودی مشخص شود.
- `endTransmission()`: این تابع ارتباط شروع شده را پایان می‌دهد. (بیت Stop)

- `requestFrom()` master با این تابع از slave درخواست دیتا می‌کند. ورودی Boolean می‌گیرد که نشان می‌دهد آیا پس از ارسال درخواست master باید باس را رها کند یا خیر.
- `available()`: تعداد بایت‌های آماده برای دریافت را برمی‌گرداند.
- `read()`: داده‌های ارسالی را دریافت می‌کند.