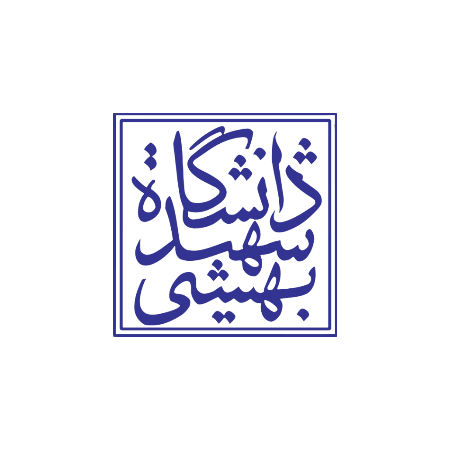
**گرازش تمرین 6**

**بسم الله الرحمن الرحيم**

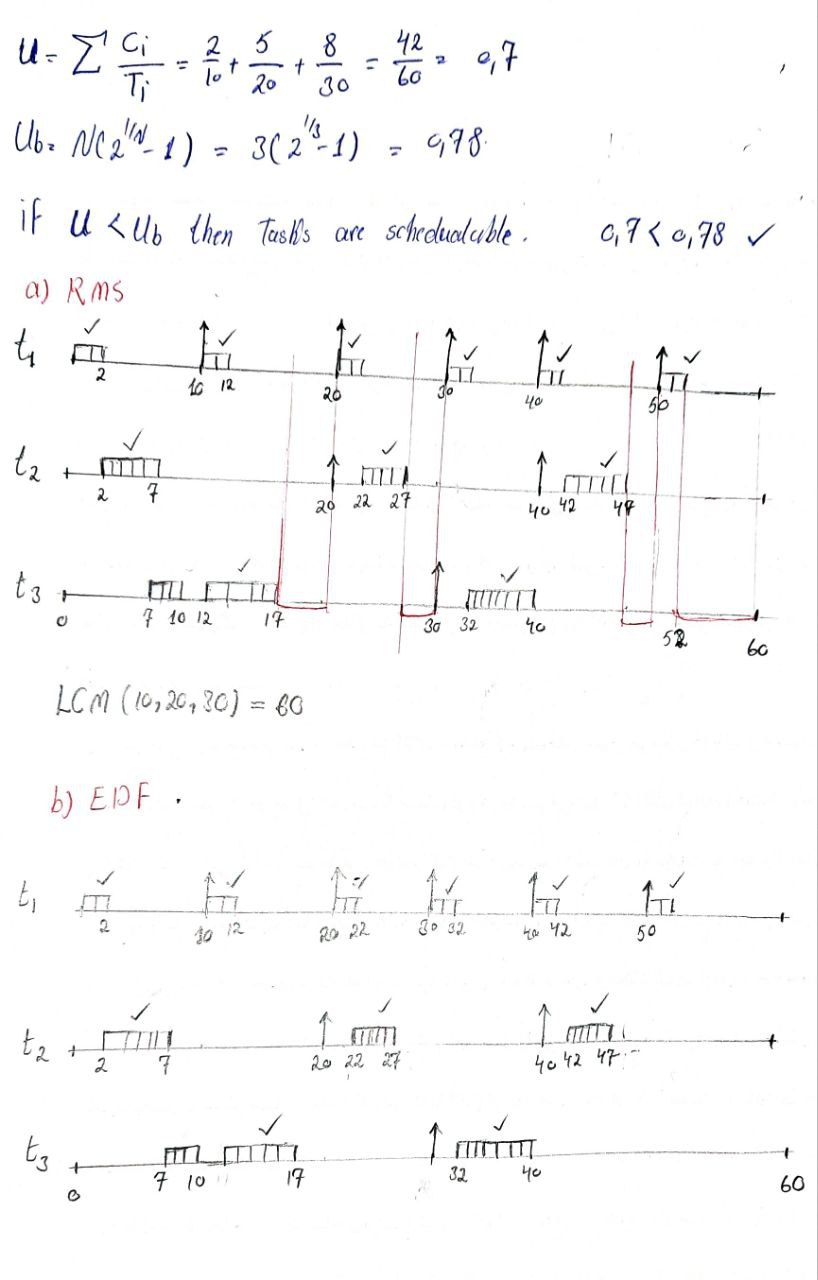
**درس مبانی سیستم نهفته و بیدرنگ**

**جواب سوال 1)**

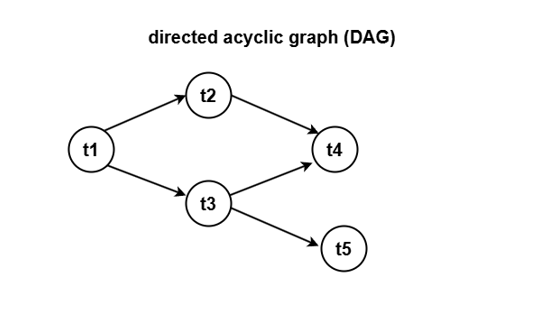
1. امکان زمانبندی مجموعه وظایف مقابل را هم بهصورت تحلیلی با به کارگیری کران های بهرهگیری مرتبط و هم با رسم زمانبندی برای یک فراتناوب بررسی کنید.

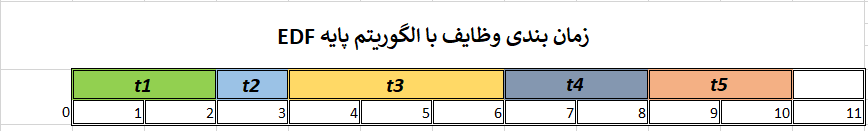
ا.RMS

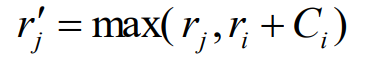
ب.EDF



**جواب سوال 2)**

**بخش الف:**  
 ترسیم گراف وابستگی)

**بخش ب:**   
 زمان بندی با الگوریتم EDL ) **بله** همان طوریکه در شکل زیر می ببینید زمان بندی ممکن است.  


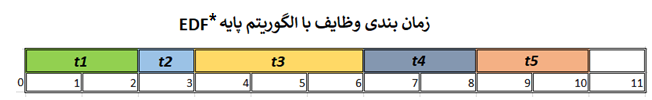
**بخش ج:**  
 برای اینکه وظایف را با استفاده از الگوریتم EDF\* زمان بندی کنیم در قدم اول باید حساب کنیم

ها را حساب کنیم پس داریم

ها را حساب کنیم پس داریم

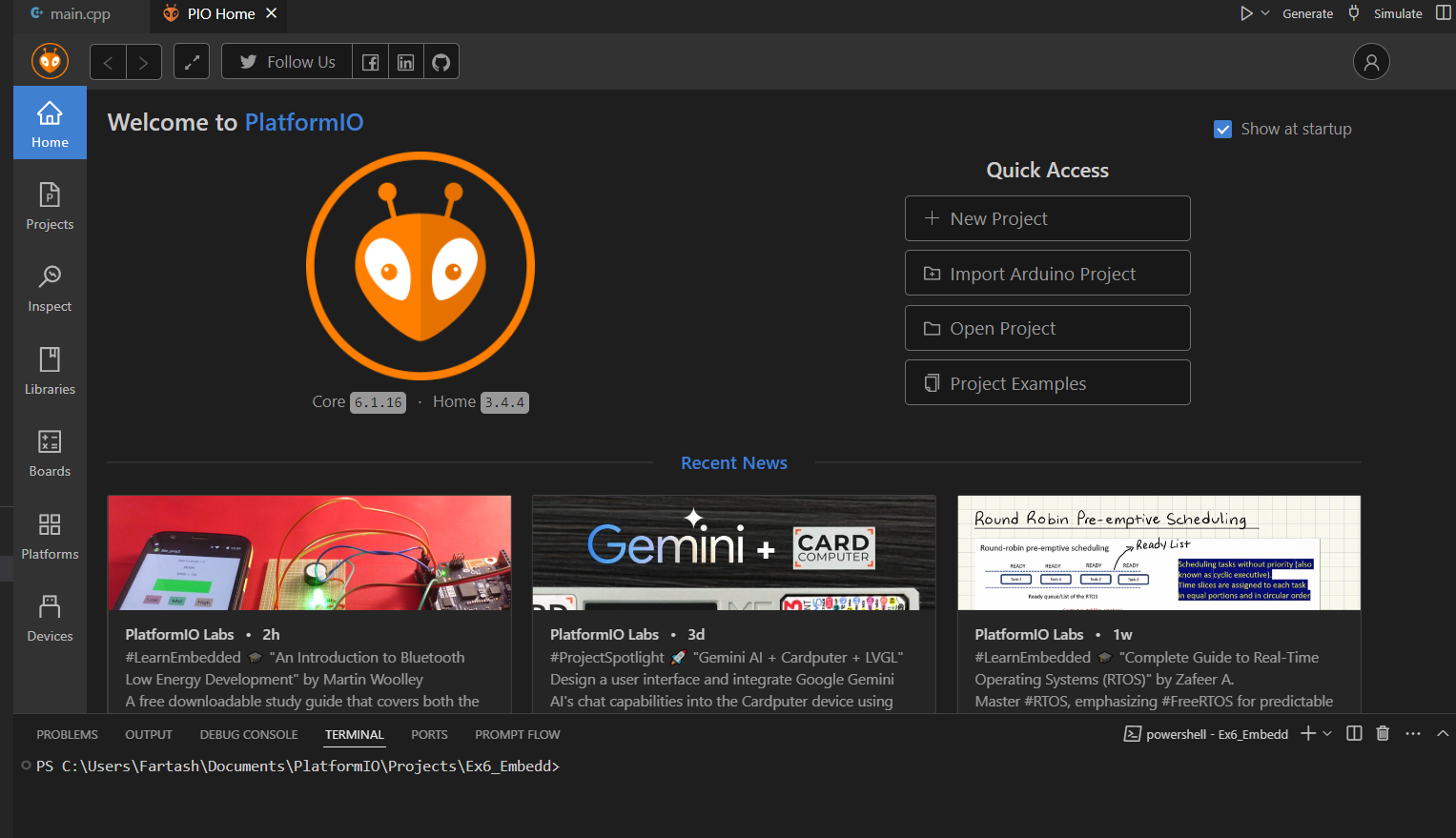
آپدیت جدول تسک ها با استفاده از الگوریتم EDF\*

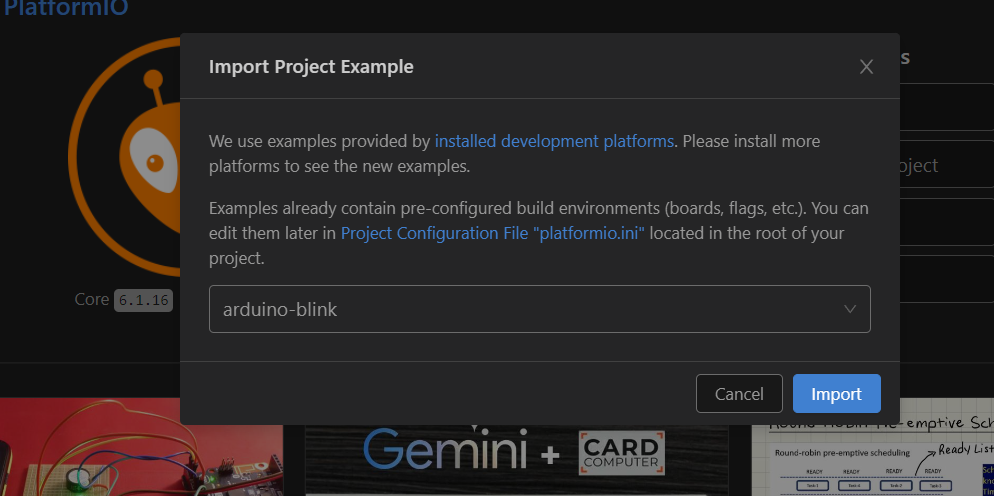
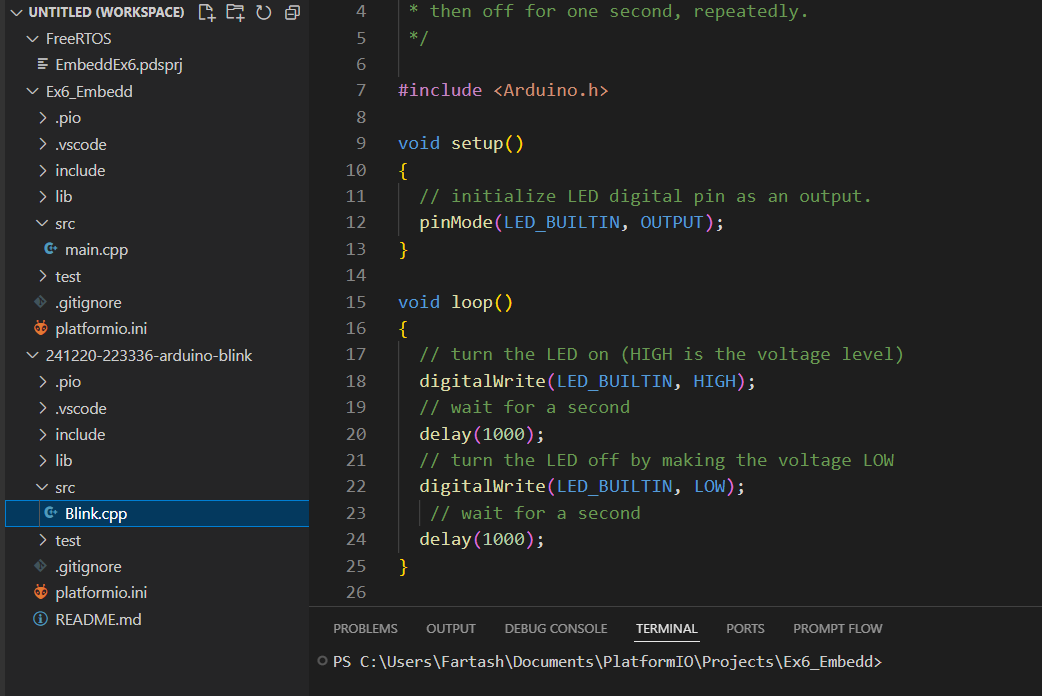
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **5** | **0** | **6** | **0** | **2** |  |
| **7** | **2** | **7** | **1** | **1** |  |
| **8** | **3** | **8** | **3** | **3** |  |
| **10** | **6** | **10** | **4** | **2** |  |
| **10** | **6** | **11** | **5** | **2** |  |

**بله** همان طوریکه در شکل زیر می ببینید با استفاده از دلاین ها و زمان منتشر جدید که بدست آوردیم امکان زمان بندی ممکن است.

**جواب سوال 3)**

قسمت الف)





**قسمت ب)   
بخش الف:**

**وظایف تولید خروجی‌ها:**

1. **وظیفه vFanSpeedTask:**
   * مسئول تنظیم سرعت فن است.
   * سرعت فن از طریق صفی به نام xSpeedQueue دریافت می‌شود.
   * با استفاده از تابع analogWrite مقدار سرعت (Duty Cycle) به پایه فن اعمال می‌شود.
   * در صورت دریافت سرعت جدید، مقدار به‌روز می‌شود و پیام مناسبی برای نمایش سرعت فن ثبت می‌شود.
2. **وظیفه vFanDirectionTask:**
   * مسئول تنظیم جهت چرخش فن است.
   * مقدار جهت از طریق صف xDirectionQueue دریافت می‌شود.
   * از یک سروو موتور برای تغییر جهت استفاده می‌کند و موقعیت جدید سروو را بر اساس مقدار دریافت‌شده تنظیم می‌کند.
   * تغییر جهت با حرکت نرم (Smooth) انجام می‌شود و موقعیت نهایی ثبت می‌گردد.

**وظایف پردازش ورودی‌ها:**

1. **وظیفه vInputProcessingTask:**
   * ورودی‌های مربوط به دما و وضعیت کلید را پردازش می‌کند.
   * دمای سنسور از طریق پایه TEMP\_SENSOR\_PIN خوانده می‌شود و به مقدار سلسیوس تبدیل می‌شود.
   * وضعیت کلید از پایه SWITCH\_PIN خوانده می‌شود.
   * بر اساس دما، مقدار سرعت و بر اساس وضعیت کلید، مقدار جهت چرخش تعیین شده و در صف‌های مربوطه قرار می‌گیرد.
2. **تابع Callback تایمر vTimerCallback:**
   * به صورت دوره‌ای (هر ۱۰۰۰ میلی‌ثانیه) یک مقدار سرعت پیش‌فرض را به صف xSpeedQueue ارسال می‌کند.
   * این تابع برای تولید رویدادهای دوره‌ای استفاده می‌شود و به‌نوعی مکمل وظایف پردازش ورودی‌ها است.

**نکات اضافی:**

* **استفاده از وقفه:** در برنامه از یک وقفه شبیه‌سازی‌شده (vISRHandler) استفاده شده است که وظیفه فعال‌سازی یک Semaphore را برای هماهنگی وظیفه‌ها برعهده دارد.
* **ارتباط وظیفه‌ها:** وظایف از طریق صف‌ها و Semaphore با یکدیگر ارتباط دارند و هرکدام بخشی از کار را مستقل و همزمان انجام می‌دهند.

این ساختار به خوبی اصول طراحی برنامه‌های چندوظیفه‌ای را نشان می‌دهد و ارتباط مؤثر بین وظیفه‌ها و استفاده از منابع بهینه را تضمین می‌کند.

**بخش دوم:**

**وظایف متناوب (Time-Triggered):**این وظایف به صورت دوره‌ای اجرا می‌شوند و زمان‌بندی آن‌ها مشخص است.

1. **وظیفه vInputProcessingTask:**
   * **نوع تحریک:** متناوب.
   * **روش اجرا:** این وظیفه هر 500 میلی‌ثانیه با استفاده از تأخیر (vTaskDelay) اجرا می‌شود.
   * **وظیفه:** خواندن ورودی‌های دما و کلید، پردازش آن‌ها و ارسال مقادیر به صف‌های xSpeedQueue و xDirectionQueue.
2. **تابع vTimerCallback:**
   * **نوع تحریک:** متناوب.
   * **روش اجرا:** این تابع توسط یک تایمر نرم‌افزاری FreeRTOS به نام xTimer هر 1000 میلی‌ثانیه اجرا می‌شود.
   * **وظیفه:** ارسال مقدار سرعت پیش‌فرض به صف xSpeedQueue برای به‌روزرسانی متناوب سرعت فن.

**وظایف نامتناوب (Event-Triggered):**این وظایف فقط در پاسخ به یک رویداد) مانند داده در صف یا فعال شدن ( Semaphore اجرا می‌شوند.

1. **وظیفه vFanSpeedTask:**
   * **نوع تحریک:** نامتناوب.
   * **روش اجرا:** زمانی که داده‌ای در صف xSpeedQueue قرار گیرد، وظیفه با استفاده از xQueueReceive فعال می‌شود.
   * **وظیفه:** تنظیم سرعت فن با مقدار دریافتی.
2. **وظیفه vFanDirectionTask:**
   * **نوع تحریک:** نامتناوب.
   * **روش اجرا:** زمانی که داده‌ای در صف xDirectionQueue قرار گیرد، وظیفه با استفاده از xQueueReceive فعال می‌شود.
   * **وظیفه:** تغییر جهت چرخش فن با مقدار دریافتی.
3. **Semaphore مربوط به وقفه (ISR):**
   * **نوع تحریک:** نامتناوب.
   * **روش اجرا:** وقفه شبیه‌سازی‌شده (vISRHandler) یک Semaphore باینری به نام xISRTrigger را آزاد می‌کند. وظیفه vInputProcessingTask می‌تواند در صورت دریافت این Semaphore سریعاً به ورودی‌های جدید پاسخ دهد.
   * **وظیفه:** همگام‌سازی یک رویداد خاص (مانند تغییر حالت کلید یا وضعیت خاص) با وظایف.

### ****نحوه اتصال وظایف به ISR:****

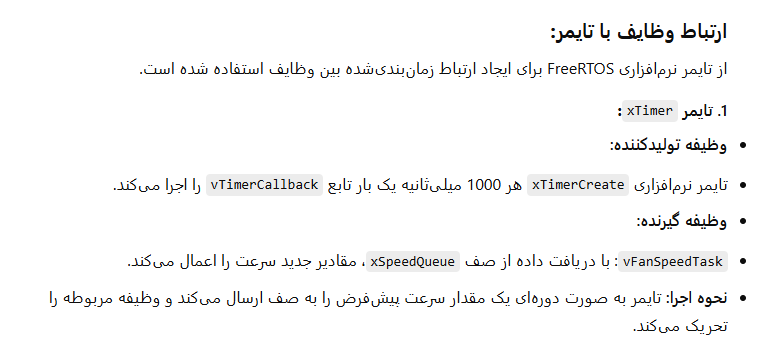
1. **وقتی که ورودی جدیدی در صف‌ها قرار می‌گیرد:**
   * وظایف vFanSpeedTask و vFanDirectionTask از این مکانیزم استفاده می‌کنند. به محض اینکه داده‌ای وارد صف شود، وظیفه از حالت بلوک خارج شده و اجرا می‌شود.
2. **وقتی که یک وقفه خارجی شبیه‌سازی شده اجرا می‌شود:**
   * در ISR (vISRHandler) از xSemaphoreGiveFromISR برای فعال کردن وظایف مرتبط (مانند vInputProcessingTask) استفاده شده است. این روش وظیفه را فوراً در پاسخ به رویداد فعال می‌کند.
3. **با استفاده از تایمر:**
   * وظیفه متناوب vTimerCallback به تایمر نرم‌افزاری FreeRTOS متصل است. این تایمر هر 1000 میلی‌ثانیه یک بار رویداد ایجاد می‌کند.

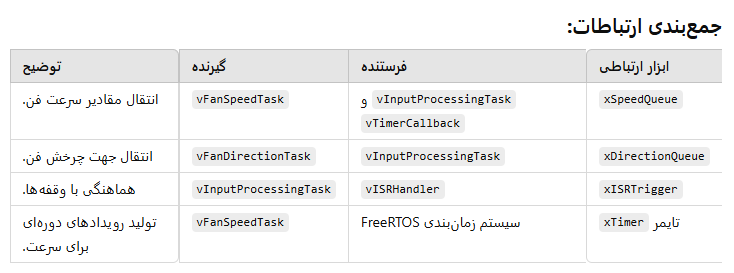


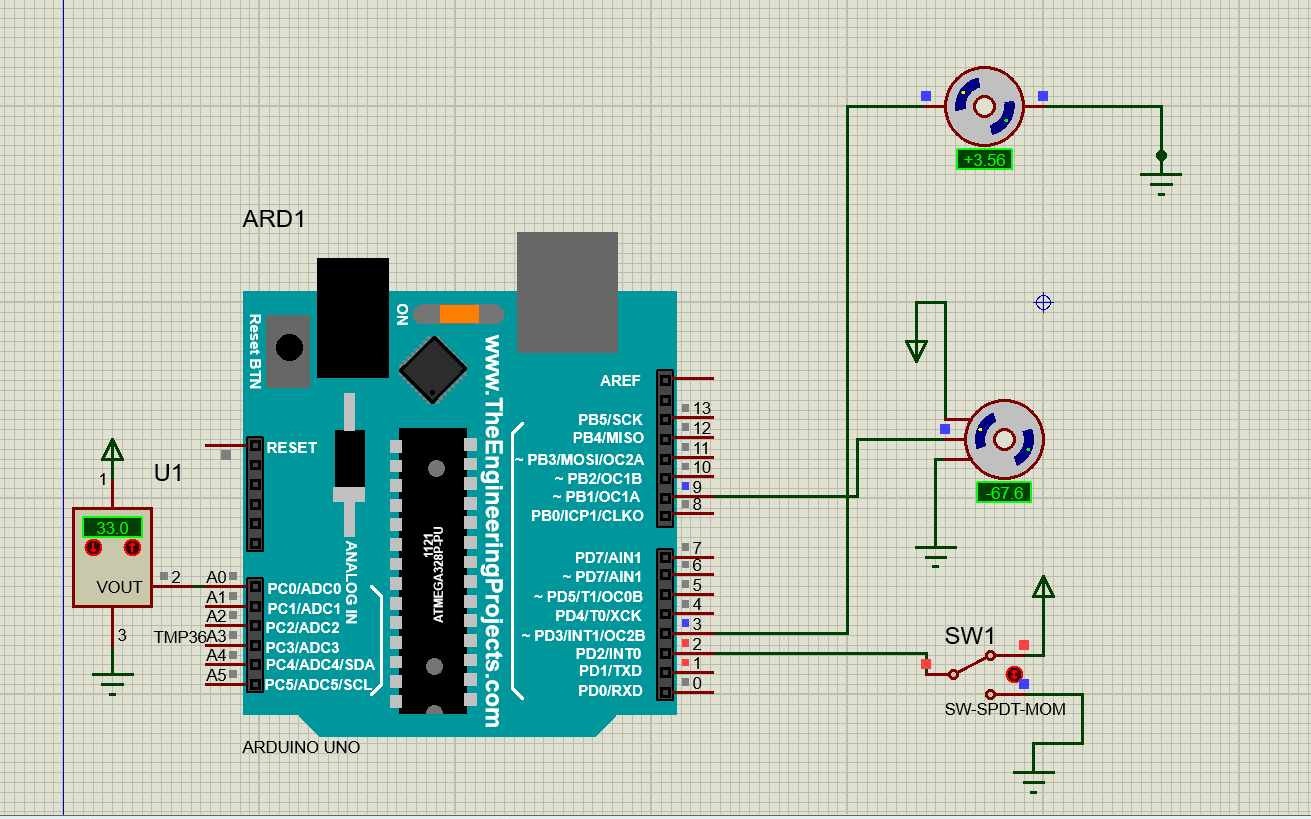
**بخش سوم:**

****

****

****

****

**قسمت ج)** بخش 1)

**بخش 3و2) تست شبه سازی حسگرها و فعالگرهای**  
