**پیش‌گزارش کار آزمایش هشتم**

**۱.** با فرض آن که r مقدار مقاومت ثابت، و و مقاومت‌های فتوسل باشند.

در مدار اول خواهیم داشت:

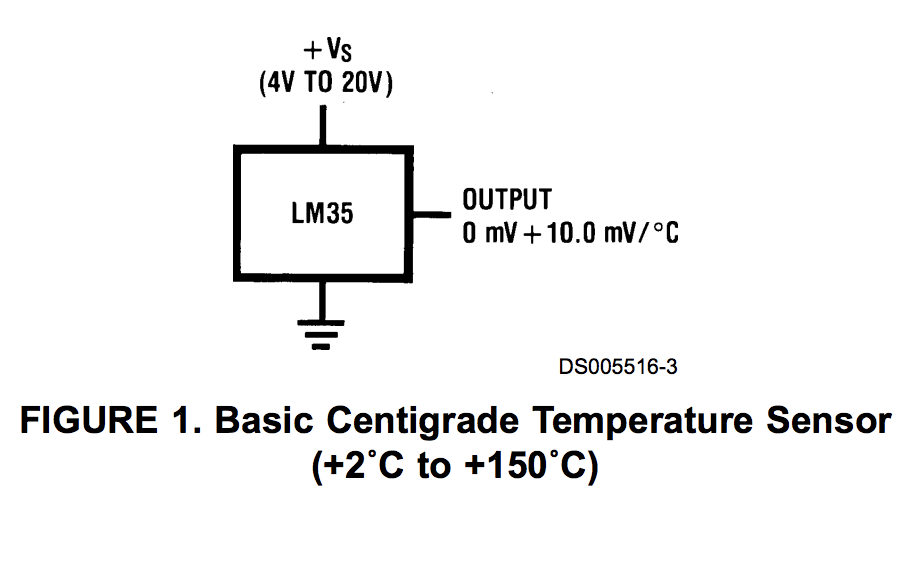
مشخص است که در این مدار با مقدار نسبت عکس دارد، از طرفی می‌دانیم هم با شدّت نور محیط نسبت عکس دارد. بنابراین مقدار با میزان نور محیط رابطه‌ی مستقیم خواهد داشت.

همچنین در مدار دوم:

اگر شدت نور محیط را با نمایش دهیم، با جایگذاری به جای خواهیم داشت:

پس مشخصا ولتاژ خروجی با شدت نور محیط رابطه‌ی عکس دارد.

**۲.** در شکل زیر نمودار شماتیک قطعه‌ی LM35 را مشاهده می‌کنید.



این قطعه از سه پایه تشکیل شده است. پایه‌ی به منبع تغذیه، و پایه‌ی به زمین متصل می‌شود.

پایه‌ی سوم، ، دمای محیط را به یک محدوده‌ی ولتاژی نگاشت می‌کند.

در حالت پایه، رابطه‌ی ولتاژ خروجی دستگاه با دمای محیط از این رابطه پیروی می‌کند.

**۳.** پایه‌های SPI در Arduino Mega به صورت زیر است.

- MISO: 50

- MOSI: 51

- SCLK: 52

- SS: 53

**۴.** پایه‌ی SS به صورت active low می‌باشد.

**۵.** مقدار کلاک همواره توسط دستگاه Master تعیین می‌شود.

**۶.** توضیح تابع‌ها به شرح زیر است.

- begin(): با صفر کردن SCK و MOSI، و یک کردن SS، ارتباط SPI را شروع می‌کند.

- setClockDivider(): کلاک سیستم را می‌گیرد، آن را تقسیم بر عدد ورودی‌اش می‌کند تا از نتیجه برای ایجاد کلاک ارتباط SPI استفاده کند. لازم به ذکر است که این تابع منسوخ شده.

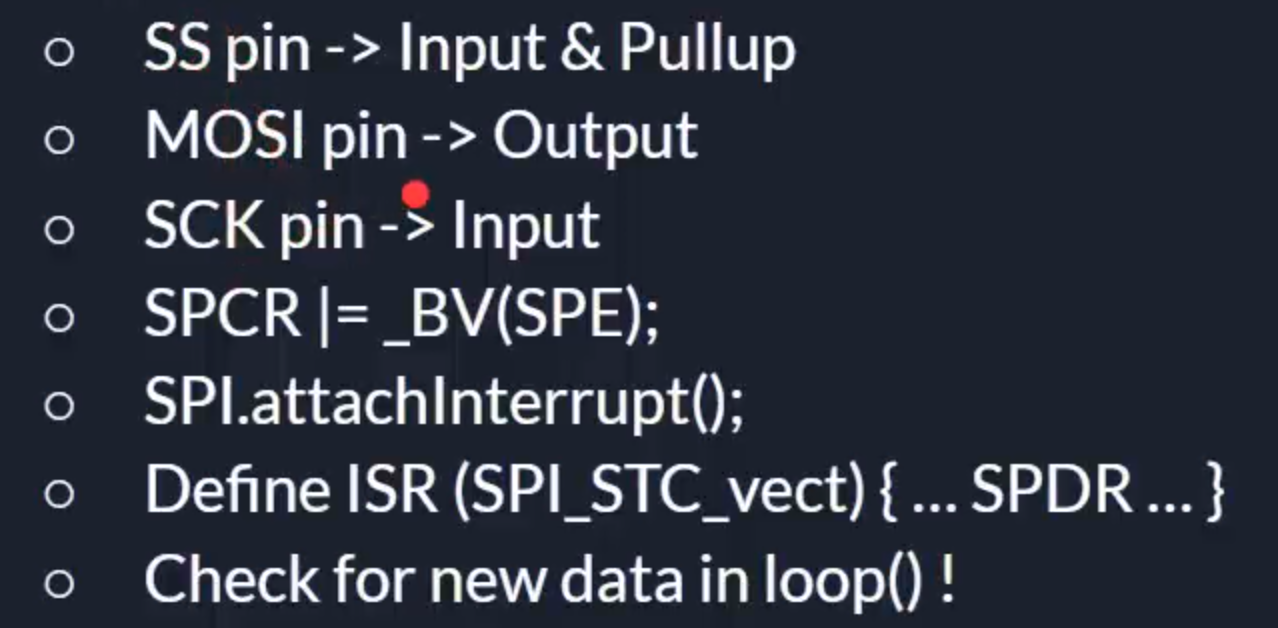
- transfer(): برای انتقال داده (به صورت هم‌زمان) می‌شود. داده‌ی گرفته شده در receivedVal ذخیره می‌شود. به عنوان ورودی هم یک، دو، یا یک آرایه از بایت‌ها می‌گیرد و ارسال می‌کند.

- attachInterrupt(): با صدا زدن این تابع، در هنگامی که یک داده از ورودی دریافت شود، تابعی با نام ISR فراخوانی می‌شود. تابع ISR باید به صورت زیر تعریف شود.

Define ISR (SPI\_STC\_vect){ …SPDR…}

در این صورت مقدار دریافت شده، در رجیستر SPDR ذخیره می‌شود.

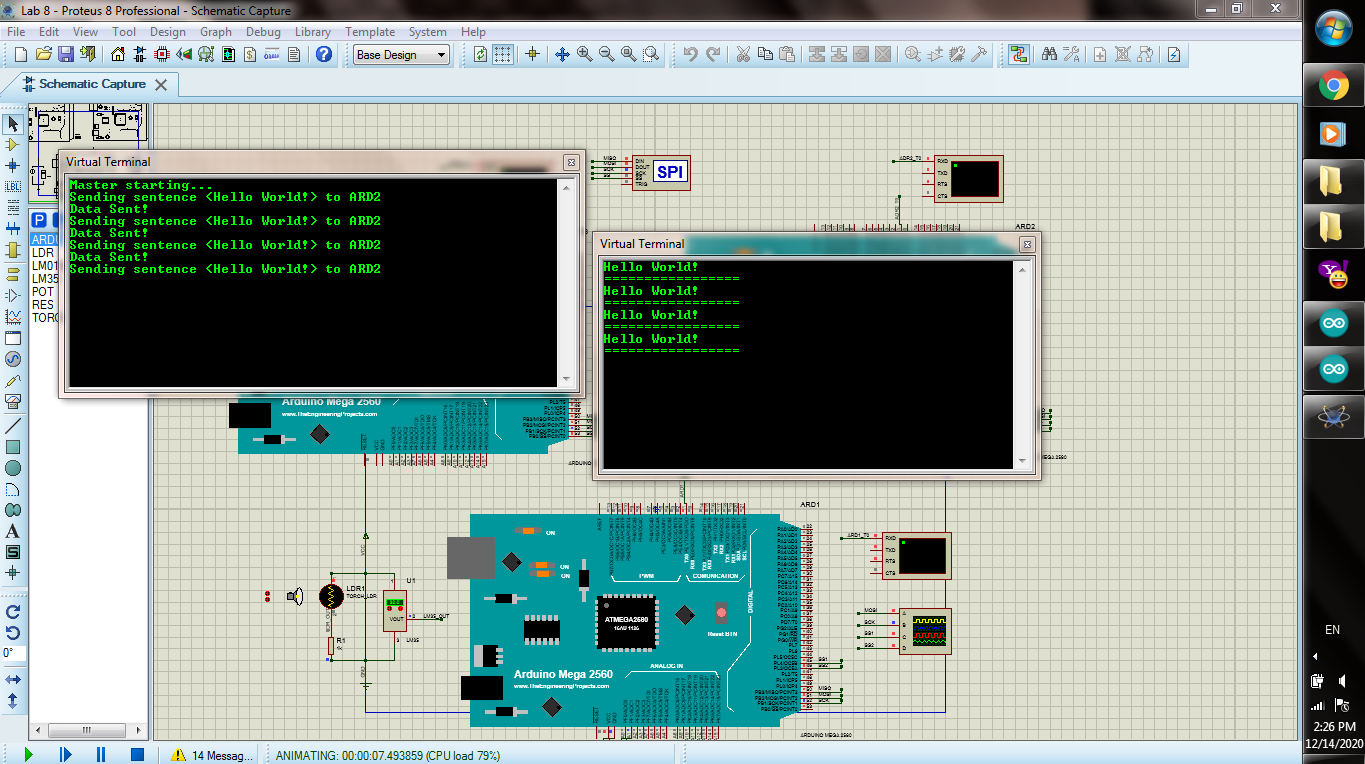
**۷.** برای بردن Arduino به حالت Slave از دستورالعمل زیر استفاده می‌کنیم.



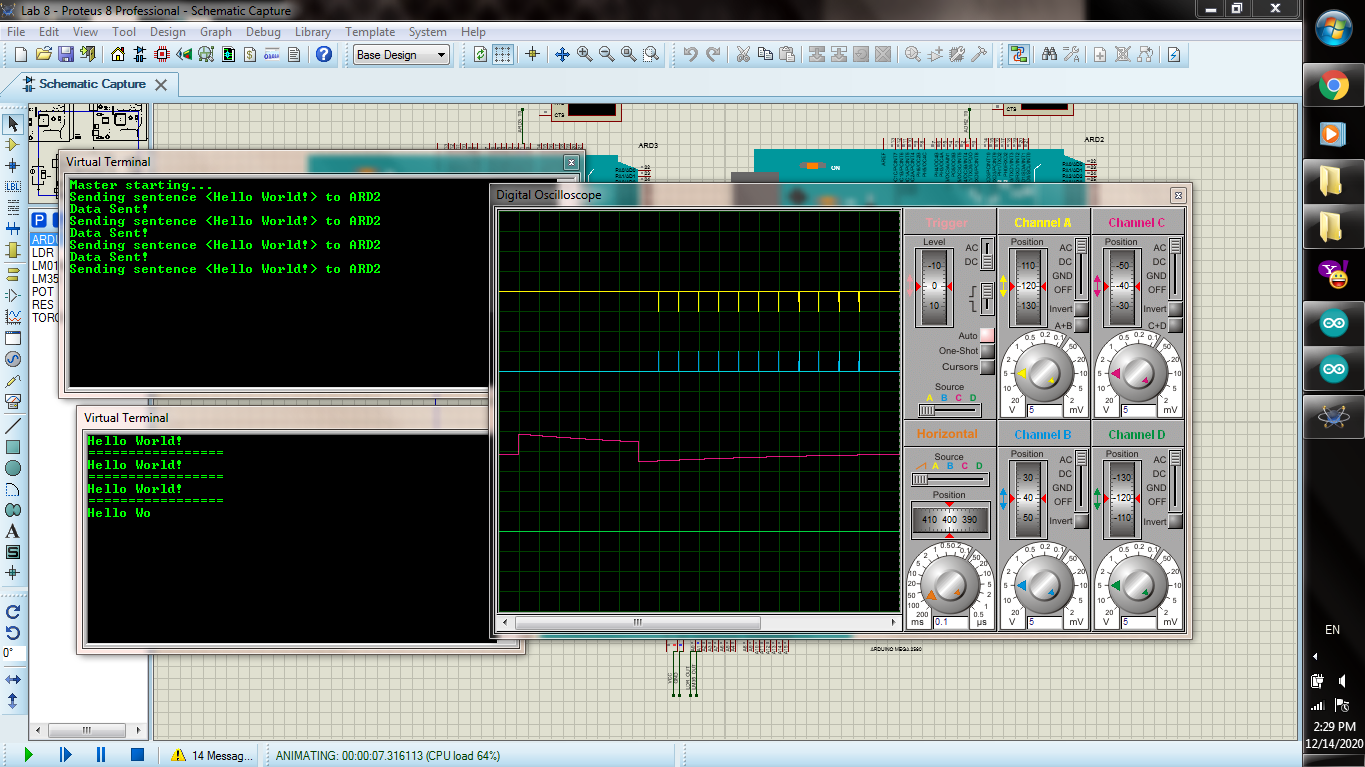
**۸.** در سوال ۶ توضیح داده شد.

**گزارش کار آزمایش هشتم**

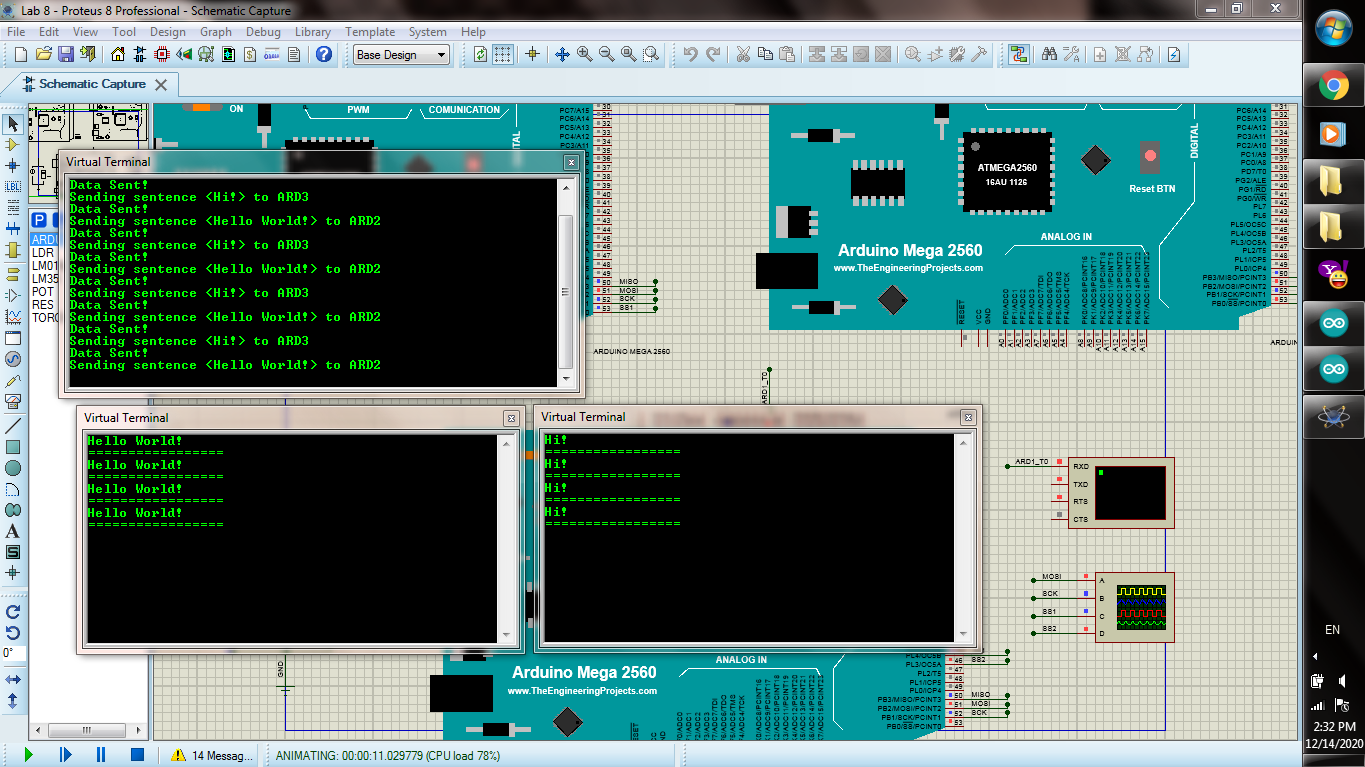
**۱.**

****

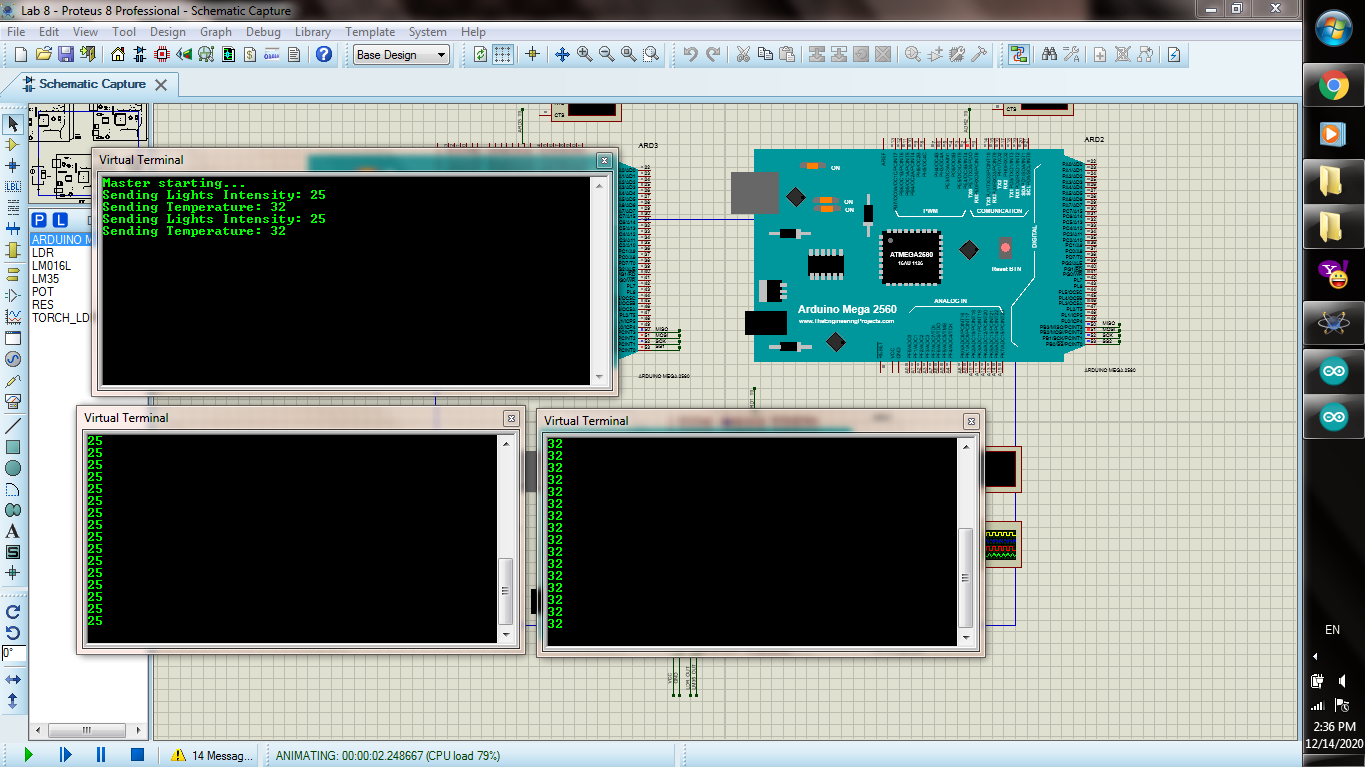
**۲.**

****

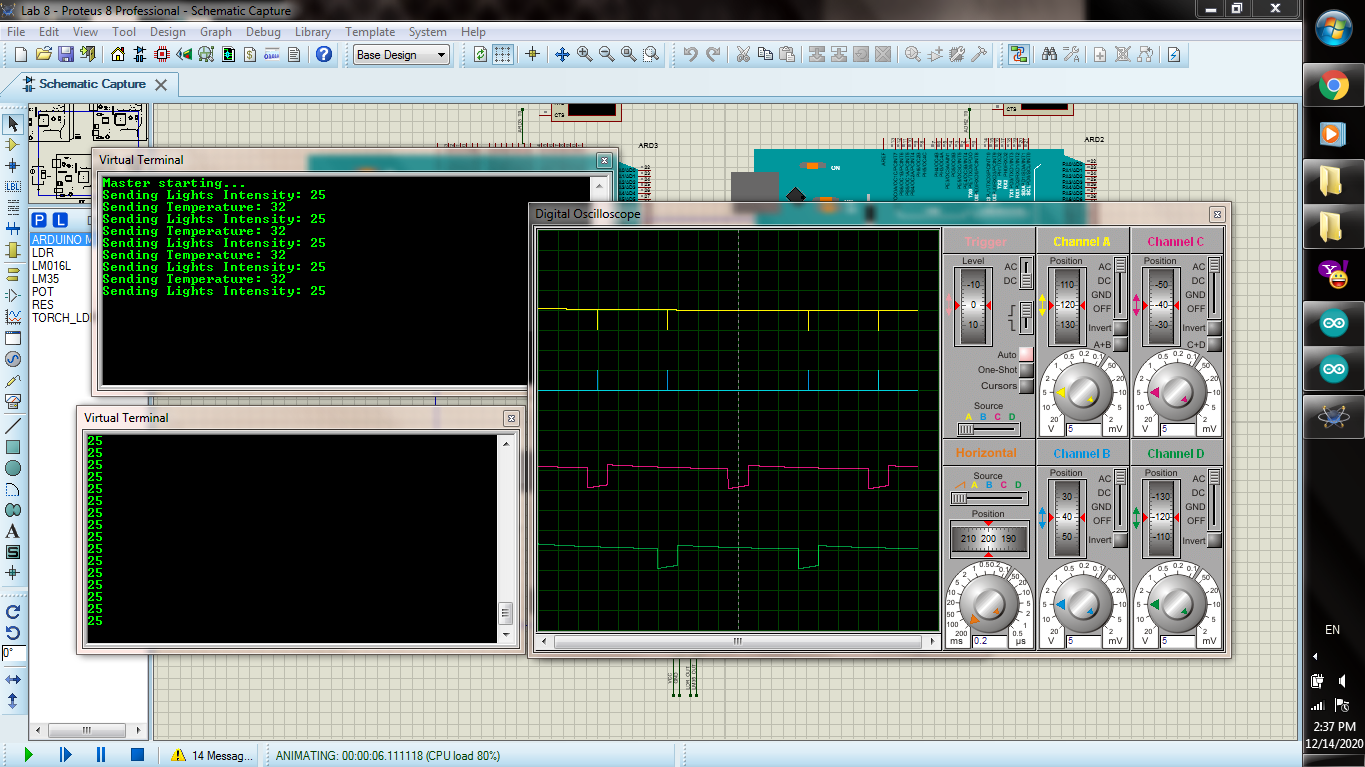
**۳.**

****

**۴.**

****

**۵.**

****