

پیش‌گزارش کار آزمایش هشتم

۱. با فرض آن که r مقدار مقاومت ثابت، و R_1 و R_2 مقاومت‌های فتوسل باشند. در مدار اول خواهیم داشت:

$$V_o = r \frac{V_i}{r + R_1}$$

مشخص است که در این مدار V_o با مقدار R_1 نسبت عکس دارد، از طرفی می‌دانیم R_1 هم با شدت نور محیط نسبت عکس دارد. بنابراین مقدار V_o با میزان نور محیط رابطه‌ی مستقیم خواهد داشت.

همچنین در مدار دوم:

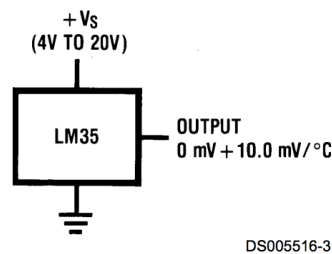
$$V_o = R_2 \frac{V_i}{r + R_2}$$

اگر شدت نور محیط را با I_v نمایش دهیم، با جایگذاری C/I_v به جای R_2 خواهیم داشت:

$$V_o = \frac{C}{I_v} \times \frac{V_i}{r + C/I_v} = \frac{C \cdot V_i}{r \cdot I_v + C}$$

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

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



**FIGURE 1. Basic Centigrade Temperature Sensor
(+2°C to +150°C)**

این قطعه از سه پایه تشکیل شده است. پایه‌ی V_s به منبع تغذیه، و پایه‌ی GND به زمین متصل می‌شود. پایه‌ی سوم، V_o ، دمای محیط را به یک محدوده‌ی ولتاژی نگاشت می‌کند.

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

$$T^{\circ C} = \frac{V_{o(mv)}}{10}$$

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

50 :MISO -

51 :MOSI -

52 :SCLK -

53 :SS -

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.

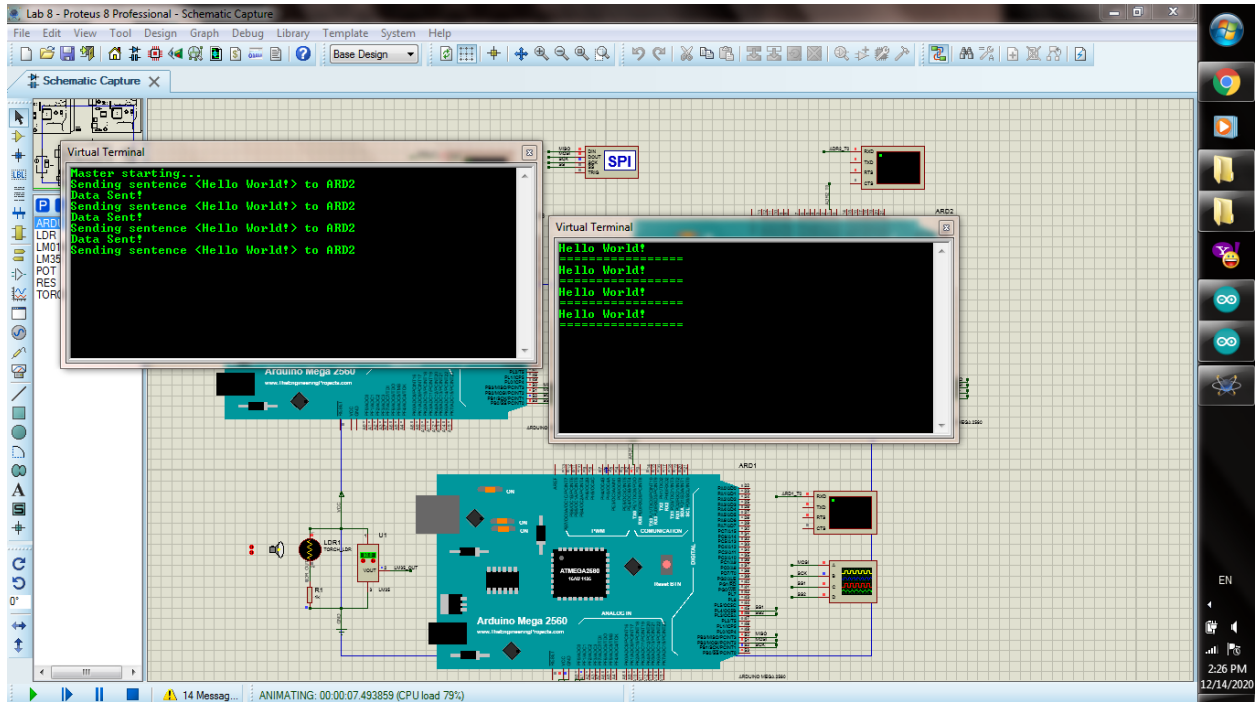
۷. برای بردن Arduino به حالت Slave از دستورالعمل زیر استفاده می‌کنیم.

- SS pin -> Input & Pullup
- MOSI pin -> Output
- SCK pin -> Input
- SPCR |= _BV(SPE);
- SPI.attachInterrupt();
- Define ISR (SPI_STC_vect) { ... SPDR ... }
- Check for new data in loop() !

۸. در سوال ۶ توضیح داده شد.

گزارش کار آزمایش هشتم

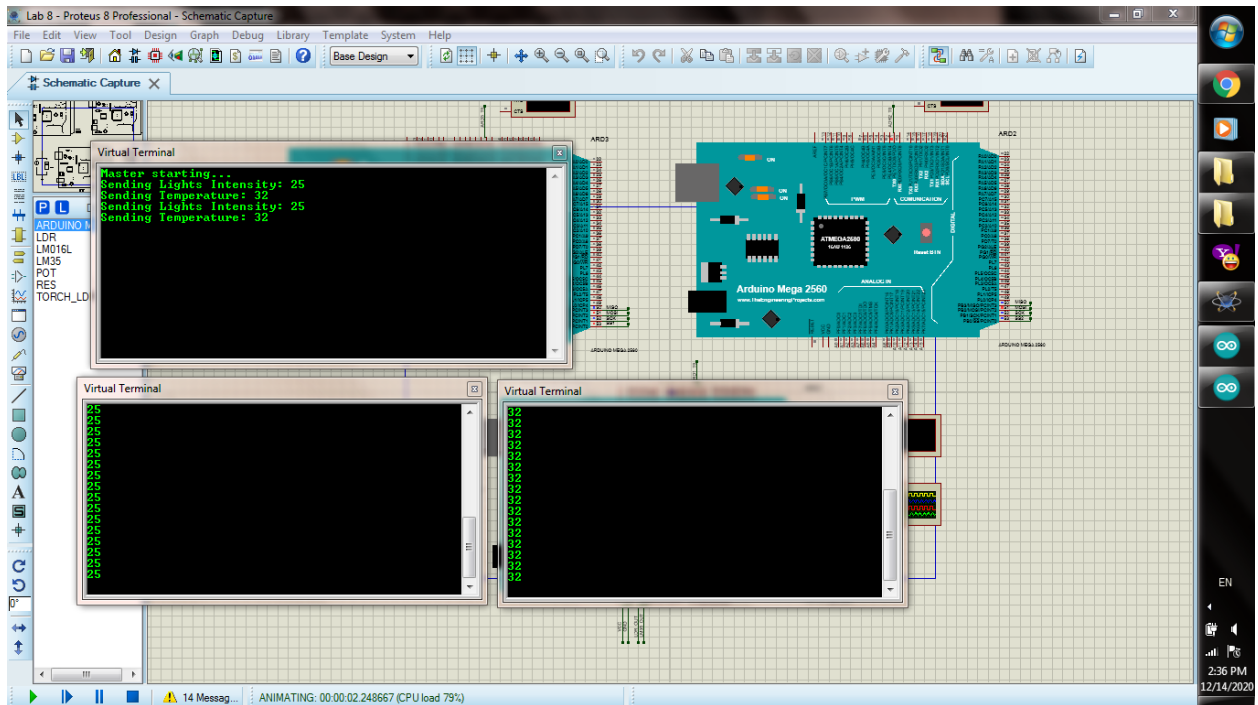
۱.



۲.



.۴



.۵

