

پیش‌گزارش آزمایش هشتم: اتاق تحت کنترل

گروه ۳: نگار موقتیان - مریم موسوی

به پرسش‌های درون مقدمه پاسخ دهید.

۱. در مورد تفاوت دو مدار فوق (فتوسل) تحقیق کنید. میزان ولتاژ خروجی هر کدام با

تغییرات نور چگونه تغییر می‌کند.

تفاوت دو مدار (فتوسل‌ها) برمی‌گردد به ارتباط V_o با نور اگر فتوسل در قبل V_o قرار بگیرد؛ ولتاژ خروجی با نور ارتباط مستقیم خواهد داشت، یعنی با زیاد شدن نور، ولتاژ نیز زیاد می‌شود. در صورتی که مقادیر قبل V_o و فتوسل بعد از V_o قرار بگیرد، رابطه ولتاژ خروجی با نور عکس خواهد بود.

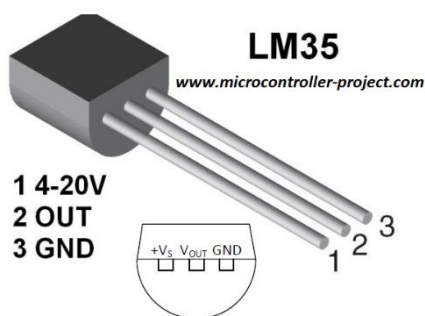
۲. در مورد پایه‌های آن (سنسور دما $lm35$) و همین‌طور نحوه تبدیل ولتاژ خروجی به میزان دما تحقیق کنید.

سنسور $lm35$ سه پایه دارد و یکی پایه ولتاژ ورودی است که مقداری بین ۴ تا ۲۰ ولت می‌تواند داشته باشد. پایه دوم پایه ولتاژ خروجی آن است که دما را در واحد ولت و به صورت آنالوگ برمی‌گرداند و پایه سوم آن پایه GND است.

تبدیل ولتاژ ورودی به خروجی آن بر اساس مقیاس خطی است. (linear scale factor) به این معنا که اگر مقیاس خطی آن ۱۰ میلی‌ولت بر درجه سانتی‌گراد باشد، یعنی به ازای هر ۱۰ میلی‌ولت ولتاژ ورودی که افزایش می‌یابد، دما ۱ درجه بالا می‌رود. بنابراین به دست آوردن دما خواهد بود:

$$\text{Centigrade temperature} = \text{voltage read by ADC} / 10\text{mV}$$

LM35 Pin Out



۳. در مورد پایه‌های $MOSI$ ، $MISO$ ، $SCLK$ در آردوینو $Mega$ تحقیق کنید. پایه‌ی

پیش‌فرض برای SS کدام پایه است؟

$SCLK$: source clock/ slave clock : کلاک مرجع را منتقل می‌کند.

master in slave out:MISO: برای انتقال داده است. داده از slave خارج شده و به master وارد می‌شود. وجود این پایه و پایه MOSI باعث می‌شود که این پروتکل full duplex باشد و master و slave همزمان برای یک دیگر داده بفرستند.

MOSI: master out slave in: برای انتقال داده استفاده می‌شود. داده را از master خارج کرده و به slave وارد می‌کند.

پایه SS به صورت active low است. یعنی در حالت عادی یک است و در زمان فعال شدن صفر می‌شود.

۴. در مورد نحوه‌ی انتخاب برد Slave توسط SS تحقیق نموده و نحوه پیاده‌سازی برنامه را برای این که برد مرکزی بتواند به ترتیب و در هر ثانیه برای یکی از بردهای Slave داده ارسال کند؛ شرح دهید.

زمانی که master می‌خواهد با یک slave ارتباط برقرار کند؛ (از او داده بگیرد یا به او داده بدهد.) باید مشخص کند که می‌خواهد با کدام slave ارتباط برقرار کند. در نتیجه، لازم است که ارتباط با تمام slaveها در ابتدا غیرفعال باشد و یک slave به خصوص انتخاب شود. این کار توسط SS انجام می‌شود. به گونه‌ای که پایه SS برای slave مورد نظر فعال می‌شود و سپس داده انتقال می‌یابد. بنابراین master باید به تعداد slaveها پایه ورودی داشته باشد تا SS مورد نظر را فعال کند. راه کار دیگر این است که یک Decoder در مسیر SS با master قرار داده شود. تا با تعداد پایه‌های کمتر بتواند slave مورد نظر را فعال کند. برای ارتباط به ترتیب master با slaveها می‌توان از یک counter استفاده کرد. متناسب با خروجی counter می‌توان خروجی SS مورد نظرا را فعال کرد و یک کلاک به آن داده فرستاد و در کلاک بعدی counter یک مقدار افزایش یافته و SS به slave بعدی می‌رود.

۵. مقدار کلاک توسط Master تعیین می‌شود یا Slave؟

توسط master تعیین می‌شود.

۶. هر یک از تابع‌های نوشته شده را از راه لینک کتابخانه wire، در مستندات آردوینو بررسی کنید.

`SPI.beginTransaction()`: گذرگاه SPI را بر اساس تنظیمات `SPISettings` آغاز می‌کند.

`SPI.end()`: گذرگاه SPI را می‌بندد.

`SPI.setBitOrder()`: تعداد بیت‌هایی که به بیرون یا داخل bus شیفت داده شده‌اند را مشخص

می‌کند. به عنوان پارامتر، مشخص می‌کند که از سمت پرارزش شیفت انجام شود یا کم ارزش.

`SPI.setClockDivider()`: تنظیمات لازم برای تطبیق تقسیم کلاک و کلاک سیستم را انجام

می‌دهد. به صورت پیش فرض بر روی ۴ مگاهرتز است. پارامتر ورودی باید توانی از ۲ باشد.

`SPI.setDATAMode()`: مود داده را تنشیم می‌کند که می‌تواند `polarity` یا `phase` باشد.
`SPI.transfer()`: بر اساس فرستادن و گرفتن داده به صورت همزمان کار می‌کند. پارامتر ورودی را می‌فرستد و داده‌ی گرفته شده را برمیگرداند.
`SPI.usingInterrupt()`: اگر انتقال داده در یک وقفه رخ می‌دهد این تابع به آن نیاز دارد. عدد وقفه را میگیرد و در یک ثبات ذخیره می‌کند و تابع `ISR` را مطابق آن فراخوانی می‌کند.
۷. دستور مورد نیاز تا آردوینو در حالت **slave** قرار گیرد را نوشته و در مورد کارایی آن تحقیق نمایید.

۱) `MISO` را در حالت `OUTPUT` قرار داده تا داده دریافت کند.
۲) `slave mode` را از طریق تنظیم ثبات `SPCR` فعال کرده.
۳) امکان وقفه رافراهم کرده در صورت دریافت داده.
۸. تابع `ISR` در کد **slave** به چه منظور استفاده می‌شود؟ رجیستر مربوط به بایت دریافتی چیست؟

برای بررسی کردن (`handling`) وقفه‌ها است. با تعریف تابع `SPI.attachInterrupt()`، با آمدن یک وقفه به سیستم، تابع `ISR` فراخوانی می‌شود.
ثبات `SPDR` بایت خوانده شده را در خود نگه می‌دارد که همان بایت دریافتی است.