• در مورد تفاوت دو مدار تحقیق کنید.

— طبق قانون تقسیم ولتاژ داریم: $V_{out} = R_2 \times \left(\frac{V_{in}}{R_1+R_2}\right)$
که در شکل ۱ زیر V_{in} برابر با 5 ولت هست V_0 ولتاژی است که روی پایه آردوینو
می افتد. R_1 مقاومت فوتوسل و R_2 مقاومتی است که به زمین وصل است

— با افزایش مقاومت فوتوسل ولتاژ خروجی کم میشود و با کاهش مقاومت فوتوسل ولتاژ خروجی
افزایش می یابد. (شکل ۱ دستور کار)

در شکل ۲ زیر همان رابطه تقسیم ولتاژ بالا برقرار است با این تفاوت که در این جا R_1 مقاومتی
است که به زمین وصل است و R_2 مقاومت فوتوسل هست. در این حالت می توان
دقیق پیش بینی کرد که تغییرات R_2 چه تاثیری بر ولتاژ خروجی می گذارد. (شکل ۱ دستور کار)

• در مورد پایه های سنسور دما lm35 تحقیق کنید:

— دارای سه پایه هست.

- VCC — برای روشن شدن سنسور Power Supply (ولتاژی بین 4 تا 20 ولت)
- Vout — ولتاژی آنالوگ که به ازای هر یک درجه سلسیوس 10 میلی ولت افزایش می یابد.
- GND : اتصال به زمین

این سنسور به ازای هر یک درجه سلسیوس افزایش دما ولتاژ خروجی اش را به 10 میلی ولت
می دهد. ما باید ولتاژ خروجی اش را باید به یکی از پایه های آردوینو که ADC دارد
وصل کنیم تا ولتاژ بین 0 تا 5 ولت را به 1024 قسمت تقسیم کند.
رابطه زیر را داریم:

```
Vout = analogRead(A0);
Temp = (Vout x 500) / 1023 ;
```

• در مورد پایه‌های MOSI و MISO و SCLK تحقیق کنید.

MOSI: برای انتقال داده هست و داده خارج شده از master را به slave وارد می‌کند و پایه پیش فرض آن در برد ما 51 هست.

MISO: برای انتقال داده هست و داده خارج شده از slave را به master وارد می‌کند. و پایه پیش فرض آن در برد ما 50 است.

SCLK: برای انتقال کلاک مشترک از سمت master است و پایه پیش فرض آن در برد ما 52 است. و پایه پیش فرض SS 53 است.

• در مورد نحوه انتخاب برد slave توسط SS تحقیق کنید.

پایه‌های SS همواره در حالت pull up قرار دارند و برای انتخاب کردن یک slave و شروع برقراری ارتباط منطق 0 از سمت master روی سیم SS قرار می‌گیرد تا slave مورد نظر فعال شود سپس کمی delay لازم است تا تنظیمات لازم در slave انجام شوند سپس انتقال داده را می‌توان شروع کرد. برای اینکه برد مرکزی در هر ثانیه بتواند به یک برد slave داده ارسال کند در هر ثانیه باید یک برد را high و دیگری را low کنیم با استفاده از digitalWrite.

نحوه پیاده‌سازی در کد هست.

• مقدار کلاک توسط master تعیین میشود یا Slave ؟ مقید

• هر یک از تابع‌های نوشته شده را بررسی کنید.

- Begin() : قطعه سخت افزاری SPI داخل پردازنده را initiate می‌کند رجیسترها را آماده می‌کند پین‌های مورد نیاز برای ارتباط را آماده می‌کند و clk و MISO و MOSI و SS را output می‌کند و SS را high می‌کند و بقیه پایه‌ها را low می‌کند.

- SetclockDivider : این تابع دیپریکیت شده است کلاک SPI را متناسب با systemclock ست می‌کند در حالت دیفالت ۱/۴ کلاک سیستم اصلی است.

- transfer : که در ورودی بایت می‌گیرد و به slave ارسال می‌کند همچنین می‌توان چند بایت داد و سایز را مشخص کرد و خودش با فور می‌کند و می‌فرستد اگر دیتایی دریافت کند return می‌کند.

- attachInterrupt : که در حالت کلی برای Pinها interrupt را اضافه می‌کند و می‌توان این ISR نیز داد در آزمایش ما به SPI وقفه را اضافه می‌کند تا با ورود دیتا وقفه داده شود.

• دستور مورد نیاز تا آردوینو در حالت Slave باشد.

- پین SS را به عنوان ورودی و Pullup تعریف می‌کنیم چون Slave نیازی به کلاک ندارد پین MISO را به عنوان خروجی تعریف می‌کنیم پین مربوط به کلاک را ورودی تعریف می‌کنیم SPCR یکی از رجیسترهای مربوط به SPI در آردوینو با این دستور مود SPI را به Slave تغییر میدهد تابع attachInterrupt وقتی یک وقفه بیاید تابع ISR را فراخوانی کند.

```
void Setup(){
  Pinmode(SS, Input_PULLUP);
  Pinmode(MISO,OUTPUT);
  Pinmode(SCK,INPUT);
  SPCR |= _BV(SPE);
SPI.attachInterrupt();}
ISR(SPI_STC_vect){
  myArray[i]=SPDR
  i++;}
```

• تابع ISR در کد slave به چه منظور است؟

← برای گرفتن داده از سمت master از تابع ISR استفاده می‌شود. وقتی یک وقفه پیش بیاید تابع ISR را فراخوانی می‌شود. دیتایی که در سمت برای ISR آورده شد بایت به بایت دریافت و خوانده شده از بافر ورودی در رجیستر SPDR ریخته می‌شود.

Scanned with CamScanner