

"به نام یزدان پاک"



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش آزمایش هشتم

اعضای گروه:

محمد چوپان ۹۸۳۱۱۲۵

محمد سپهر توکلی کرمانی ۹۸۳۱۱۱۱

تاریخ آزمایش : ۱۴۰۰/۰۹/۲۷

آزمایش ۸:

آزمایش 8 : اتاق تحت کنترل

هدف آزمایش:

آشنایی با پروتکل SPI

تحلیل موج خروجی آردوینوی مرکزی (master)

راه اندازی حسگر نور و دما

قطعات مورد نیاز:

- 3 عدد برد Arduino Mega
- مقاومت متغیر فتوسل (ldr)
- سنسور دمای Lm35

آنچه باید در پیش‌گزارش نوشته شود:

- به پرسش‌های درون مقدمه پاسخ داده شود.

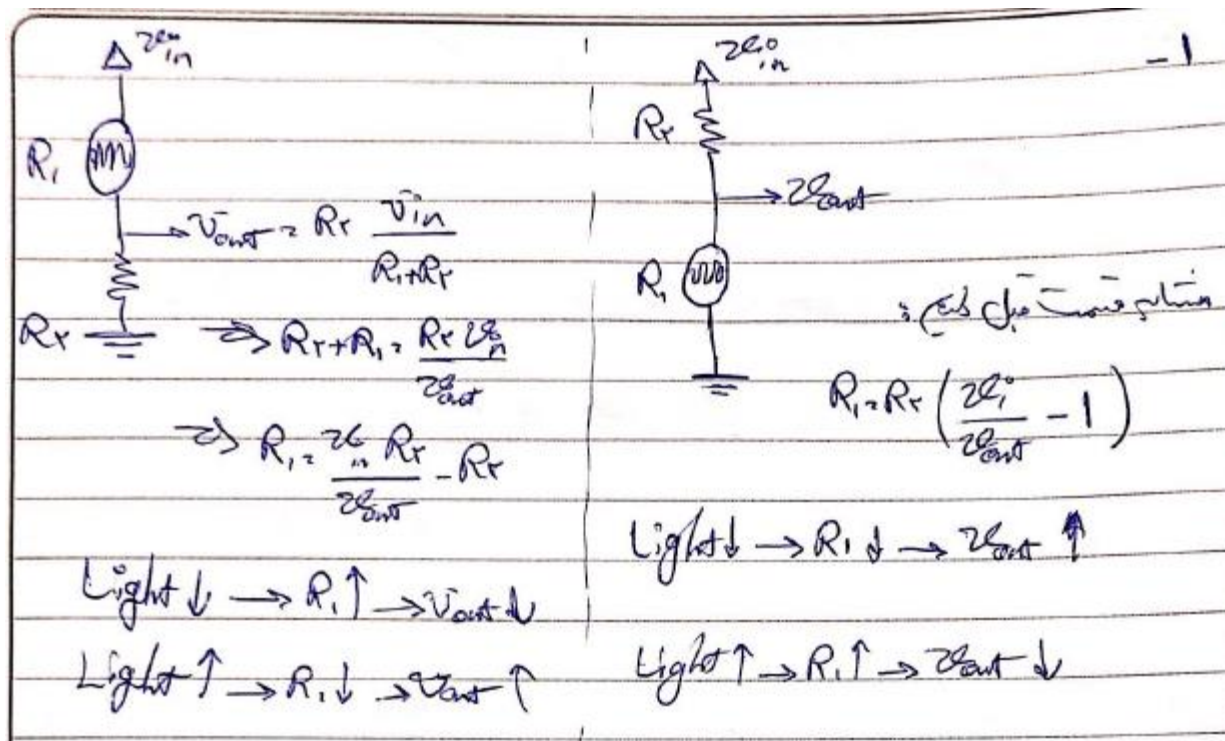
شرح آزمایش:

در این آزمایش می‌خواهیم با استفاده از پروتکل spi و دو عدد برد Arduino این پروتکل را شبیه‌سازی کنیم.

پرسش: در مورد تفاوت دو مدار فوق تحقیق کنید. میزان ولتاژ خروجی هر کدام با تغییرات نور چگونه تغییر می‌کند.

در مدار اول با افزایش نور مقاومت R1 کمتر شده در نتیجه ولتاژ خروجی افزایش می‌یابد.

اما در مدار دوم با افزایش مقدار نور مقاومت بیشتر شده و در نتیجه ولتاژ کمتر می‌شود.



راه اندازی سنسور دما Im35 :

سنسور دما، میزان دمای محیط را بر حسب درجه سانتی گراد به ولتاژ آنالوگ تبدیل می کند.

پرسش: در مورد پایه های آن و همینطور نحوه تبدیل ولتاژ خروجی به میزان دما تحقیق کنید.

در این سنسور ۲ پایه مربوط به VCC GND است و پایه سوم ولتاژ خروجی است. این دما سنج دمای بین ۵۵ - تا ۱۵۰ درجه را می تواند نشان دهد. و در این بازه کار میکند و ولتاژ خروجی آن نیز برابر با ۰/۵- تا ۵ ولت می باشد.

پرسش: آیا در پروتکل SPI امکان حضور چند master وجود دارد؟ چه پروتکلی این امکان را به ما می دهد؟

خیر نمی توانند و اما در پروتکل i2c این قابلیت را داریم.

پرسش: در رابطه با ارتباط full-duplex تحقیق کنید. آیا پروتکل SPI از این امکان بهره مند است؟

Full-duplex به این معنا است که هم master و هم slave بتوانند با هم به صورت موازی داده انتقال دهند

و بله ما در پروتکل spi این ویژگی را داریم.

پرسش: در مورد پایه های SCLK، MISO، MOSI در آردوینو Mega تحقیق کنید. پایه ی پیش فرض برای SS کدام پایه است؟ برای مشاهده آن می توانید به محل نصب آردوینو رفته، مسیر زیر را دنبال نمایید و در انتها فایل داخل پوشه را باز نمایید:

پایه ها به ترتیب : SS:53 SCK:52 MOSI:51 MISO:50

پرسش : در مورد نحوه‌ی انتخاب برد Slave توسط SS تحقیق نموده و نحوه پیاده‌سازی برنامه را برای این‌که برد مرکزی بتواند به ترتیب و در هر ثانیه برای یکی از بردهای Slave داده ارسال کند، شرح دهید. (برای این‌کار بهتر است نمونه کدهایی که برای ارتباط بین دو آردوینو از طریق پروتکل SPI در اینترنت موجود است را بررسی نمایید.)

برای ارسال برای SLAVE باید خط SS مرتبط با آن را به صورت LOW در بیاوریم.

برای ارسال در هر ثانیه نیز می‌توانیم از DELAY استفاده بکنیم.

پرسش : مقدار کلاک توسط Master تعیین می‌شود یا Slave ؟

توسط MASTER تعیین می‌شود.

پرسش: هر یک از تابع‌های نوشته شده را از راه لینک کتابخانه Wire، در مستندات آردوینو بررسی کنید.

BEGIN() : ارتباط SPI را با تنظیم SS, MOSI, SCK به حالت خروجی و در ابتدا با LOW کردن مقدار MOSI, SCK و

HIGH کردن مقدار SS شروع میکند.

● `setClockDivider()`

: این تابع برای تقسیم فرکانس اصلی برد به توان ۲ استفاده می‌شود.

● `transfer()`

: داده‌ها را ارسال و به صورت همزمان داده‌های دریافت شده را خروجی می‌دهد.

● `attachInterrupt()`

: با استفاده از این تابع Interrupt service routine (ISR) مرتبط با آن وقفه روی یک پایه قرار

داده می‌شود.

پرسش: دستور مورد نیاز تا آردوینو در حالت Slave قرار گیرد را نوشته و در مورد کارایی آن تحقیق نمایید.

دستور `SPCR |= _BV(SPE)` برد آردوینو ما را با تنظیم کنترل رجیستر در حالت SLAVE قرار می‌دهد.

پرسش: تابع ISR در کد Slave به چه منظور استفاده می‌شود؟ رجیستر مربوط به بایت دریافتی چیست ؟

در صورت رخ دادن وقفه روی آن پین ISR مرتبط را فراخوانی میکند. برای خواندن بایت دریافتی SPDR را باید خواند.

صورت آزمایش:

1. ارتباط میان دو دستگاه آردوینو از طریق SPI برقرار نمایید. بدین منظور دو برنامه یکی برای دریافت اطلاعات توسط برد

Slave و دیگری برای ارسال اطلاعات از طریق برد master بنویسید. لازم به توضیح است master هر ثانیه کلمه اسم و

شماره دانشجویی شما را برای برد Slave ارسال می‌کند. حتما پایه SS در آردوینو master را پایه‌ای به جز پایه پیش‌فرض

آردوینو قرار دهید.

2. موج خروجی سه پایه SCLK, MOSI, SS را برای سه مقدار Clock توسط اسیلوسکوپ مشاهده کنید.

کد ها :

کد ماستر :

```
#include <SPI.h>

#define MESSAGE    "mohamad 9831125!\r"

const int SS_PIN = 45;

void setup() {
    Serial.begin(9600);
    Serial.println("Starting master");

    pinMode(SS_PIN, OUTPUT);
    digitalWrite(SS_PIN, HIGH);

    SPI.begin();
}

void loop() {
    digitalWrite(SS_PIN, LOW);
    delay(10);

    for (const char *p = MESSAGE ; char c = *p; p++) {
        SPI.transfer(c);
        Serial.print(c);
        delay(5);
    }
    Serial.println();

    digitalWrite(SS_PIN, HIGH);

    delay(1000);
}
```

کد slave :

```
#include <SPI.h>

const int MISO_PIN = 50, MOSI_PIN = 51, SCK_PIN = 52, SS_PIN = 53;
volatile int ind = 0;
volatile boolean finished = false;
char message[20];

void setup() {
  Serial.begin(9600);
  Serial.println("Starting slave");

  pinMode(MOSI_PIN, INPUT);
  pinMode(MISO_PIN, OUTPUT);
  pinMode(SCK_PIN, INPUT);
  pinMode(SS_PIN, INPUT_PULLUP);

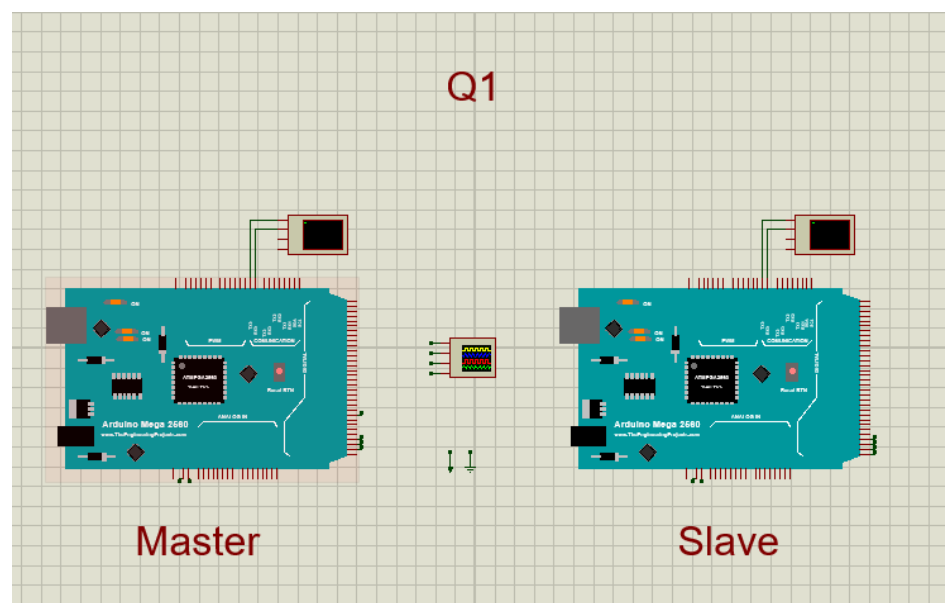
  SPCR |= _BV(SPE);
  SPI.attachInterrupt();
}

void loop() {
  if (finished) {
    Serial.println(message);
    ind = 0;
    finished = false;
  }
}

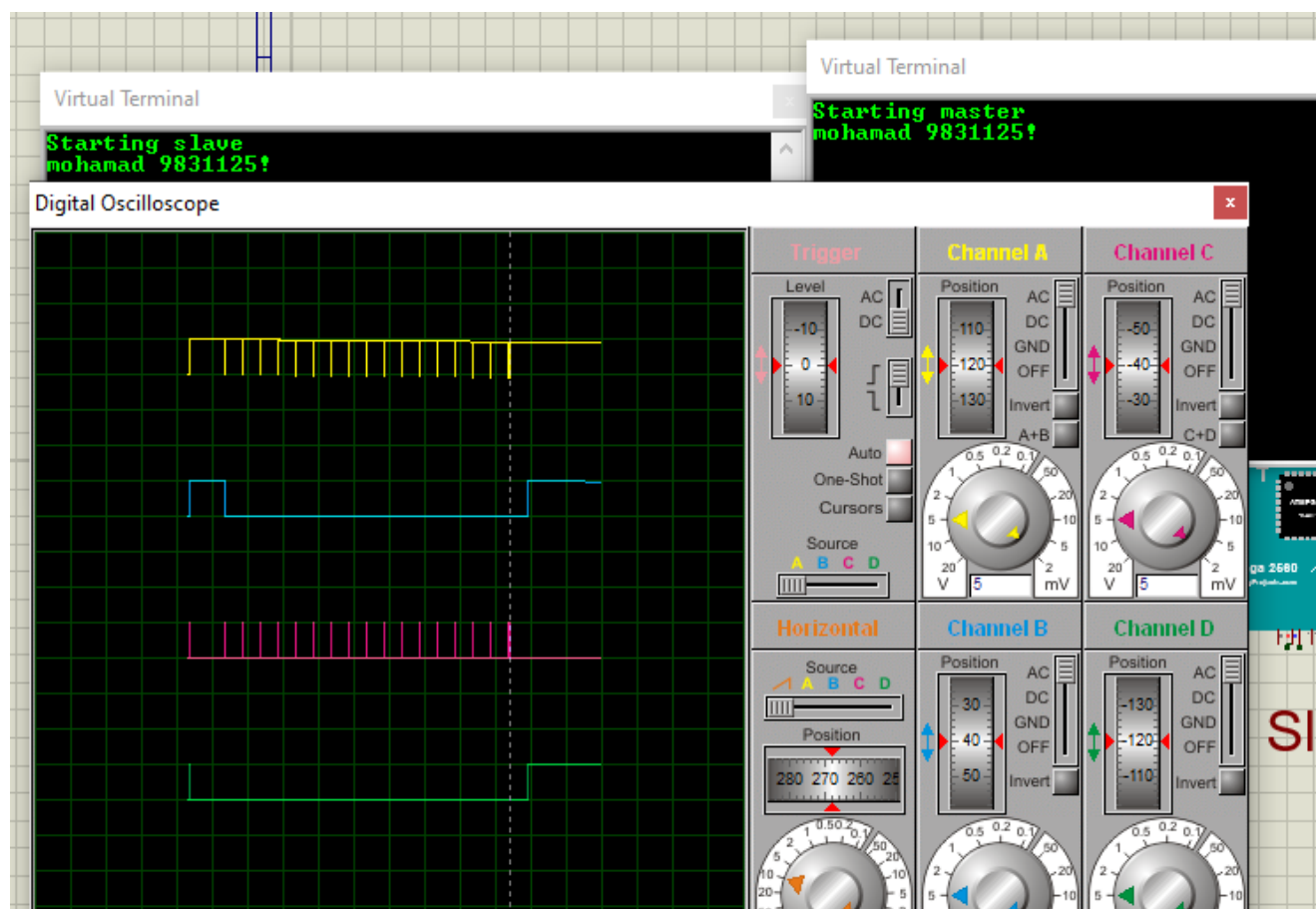
ISR (SPI_STC_vect)
{
  byte c = SPDR;
  if (ind < sizeof message) {
    message[ind++] = c;

    if (c == '\r') {
      finished = true;
    }
  }
}
```

شماتیک مدار :



خروجی حاصل :



3. کد قسمت Master و اتصالات آن را به گونه ای تغییر دهید که بعد از اضافه کردن یک Slave و قرار دادن کد مربوط به برد Slave، به طور متناوب و هر ثانیه به آردوینو دوم اسم شما ارسال شود و در ثانیه بعدی آردوینو اول کلمه "your Hello name" را دریافت کند.

کد ها :

کد slave :


```

#include <SPI.h>

const int MISO_PIN = 50, MOSI_PIN = 51, SCK_PIN = 52, SS_PIN = 53;
volatile int ind = 0;
volatile boolean finished = false;
char message[20];

void setup() {
    Serial.begin(9600);
    Serial.println("Starting slave");

    pinMode(MOSI_PIN, INPUT);
    pinMode(MISO_PIN, OUTPUT);
    pinMode(SCK_PIN, INPUT);
    pinMode(SS_PIN, INPUT_PULLUP);

    SPCR |= _BV(SPE);
    SPI.attachInterrupt();
}

void loop() {
    if (finished) {
        Serial.println(message);
        ind = 0;
        finished = false;
    }
}

ISR (SPI_STC_vect)
{
    byte c = SPDR;
    if (ind < sizeof message) {
        message[ind++] = c;

        if (c == '\r') {
            finished = true;
        }
    }
}

```

کد مستر :

```

#include <SPI.h>

#define MESSAGE0    "mohamad 9831125\r"
#define MESSAGE1    "Hello sepehr\r"

const int SS0_PIN = 44, SS1_PIN = 45;

void setup() {
    Serial.begin(9600);
    Serial.println("Starting master");

    pinMode(SS0_PIN, OUTPUT);
    digitalWrite(SS0_PIN, HIGH);

    pinMode(SS1_PIN, OUTPUT);
    digitalWrite(SS1_PIN, HIGH);

    SPI.begin();
}

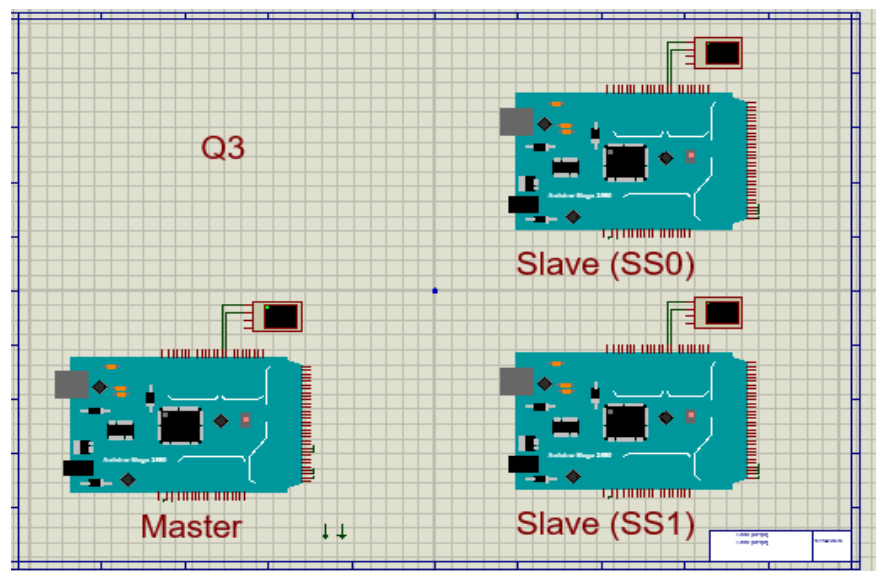
void sendMessage(const char *message) {
    for (const char *p = message; char c = *p; p++) {
        SPI.transfer(c);
        Serial.print(c);
        delay(5);
    }
    Serial.println();
}

void loop() {
    digitalWrite(SS0_PIN, LOW);
    sendMessage(MESSAGE0);
    digitalWrite(SS0_PIN, HIGH);
    delay(100);

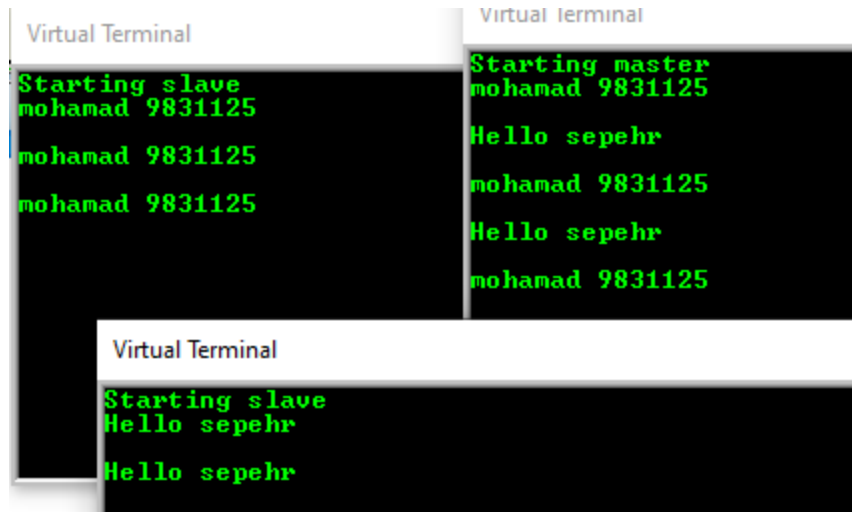
    digitalWrite(SS1_PIN, LOW);
    sendMessage(MESSAGE1);
    digitalWrite(SS1_PIN, HIGH);
    delay(100);
}

```

شماتیک مدار :



خروجی حاصل :



4. داده‌های Hello world و Hi را با اطلاعات مربوط به دو سنسور دما و نور جای‌گزین نمایید. برای خواندن ولتاژ خروجی هر یک از سنسورها کافیست از دستور analogRead استفاده نمایید. سپس عدد به دست آمده را به بازه مناسب map نمایید.
5. موج خروجی را برای چهار پایه SCLK، MOSI، SS2، SS1 توسط اسیلوسکوپ مشاهده نمایید.

کد ها :

کد مستر :

```
#include <SPI.h>

const int SS0_PIN = 44, SS1_PIN = 45, TempPin = A8, LightPin = A9;

int analogValue;
uint8_t mapped;

void setup() {
    Serial.begin(9600);
    Serial.println("Starting master");

    pinMode(SS0_PIN, OUTPUT);
    digitalWrite(SS0_PIN, HIGH);

    pinMode(SS1_PIN, OUTPUT);
    digitalWrite(SS1_PIN, HIGH);

    pinMode(TempPin, INPUT);
    pinMode(LightPin, INPUT);

    SPI.begin();
}
```

```

void loop() {
    delay(100);

    analogValue = analogRead(LightPin);
    mapped = map(analogValue, 0, 1023, 0, 100);
    digitalWrite(SS0_PIN, LOW);
    SPI.transfer(mapped);
    Serial.print("Light: ");
    Serial.print(mapped);
    Serial.println("%");
    delay(5);
    digitalWrite(SS0_PIN, HIGH);
    delay(100);

    analogValue = analogRead(TempPin);
    mapped = map(analogValue, 0, 306, 0, 150);
    digitalWrite(SS1_PIN, LOW);
    SPI.transfer(mapped);
    Serial.print("Temp: ");
    Serial.print(mapped);
    Serial.println(" C");
    delay(5);
    digitalWrite(SS1_PIN, HIGH);
    Serial.println();
}

```

کد slave دما :

```

#include <SPI.h>

const int MISO_PIN = 50, MOSI_PIN = 51, SCK_PIN = 52, SS_PIN = 53;
int value;
volatile boolean finished = false;

void setup() {
    Serial.begin(9600);
    Serial.println("Starting slave (Temperature)");

    pinMode(MOSI_PIN, INPUT);
    pinMode(MISO_PIN, OUTPUT);
    pinMode(SCK_PIN, INPUT);
    pinMode(SS_PIN, INPUT_PULLUP);

    SPDR |= _BV(SPE);
    SPI.attachInterrupt();
}

void loop() {
    if (finished) {
        Serial.print("Temprature : ");
        Serial.print(value);
        Serial.println(" C");
        finished = false;
    }
}

ISR (SPI_STC_vect)
{
    byte number = SPDR;
    value = (uint8_t)number;
    finished = true;
}

```

کد slave نور :

```
#include <SPI.h>

const int MISO_PIN = 50, MOSI_PIN = 51, SCK_PIN = 52, SS_PIN = 53;
int value;
volatile boolean finished = false;

void setup() {
  Serial.begin(9600);
  Serial.println("Starting slave (Light)");

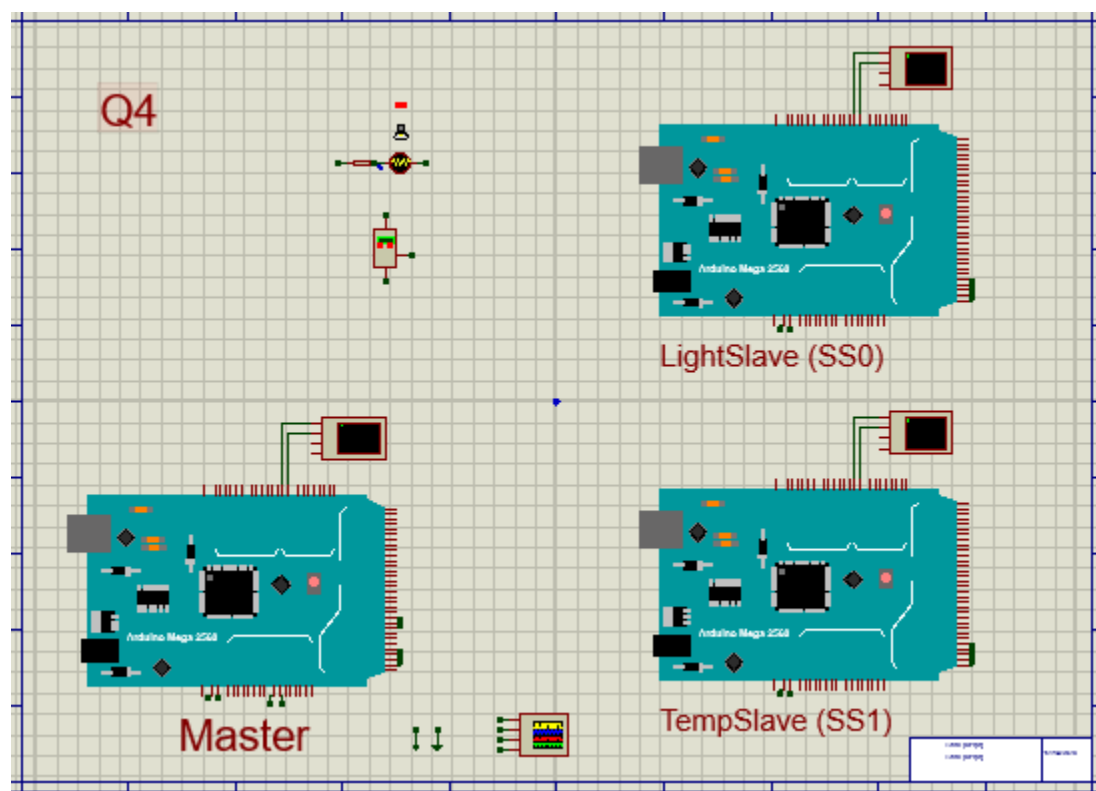
  pinMode(MOSI_PIN, INPUT);
  pinMode(MISO_PIN, OUTPUT);
  pinMode(SCK_PIN, INPUT);
  pinMode(SS_PIN, INPUT_PULLUP);

  SPCR |= _BV(SPE);
  SPI.attachInterrupt();
}

void loop() {
  if (finished) {
    Serial.print("Light: ");
    Serial.print(value);
    Serial.println("");
    finished = false;
  }
}

ISR (SPI_STC_vect)
{
  byte number = SPDR;
  value = (uint8_t)number;
  finished = true;
}
```

شماتیک مدار :



خروجی حاصل :

