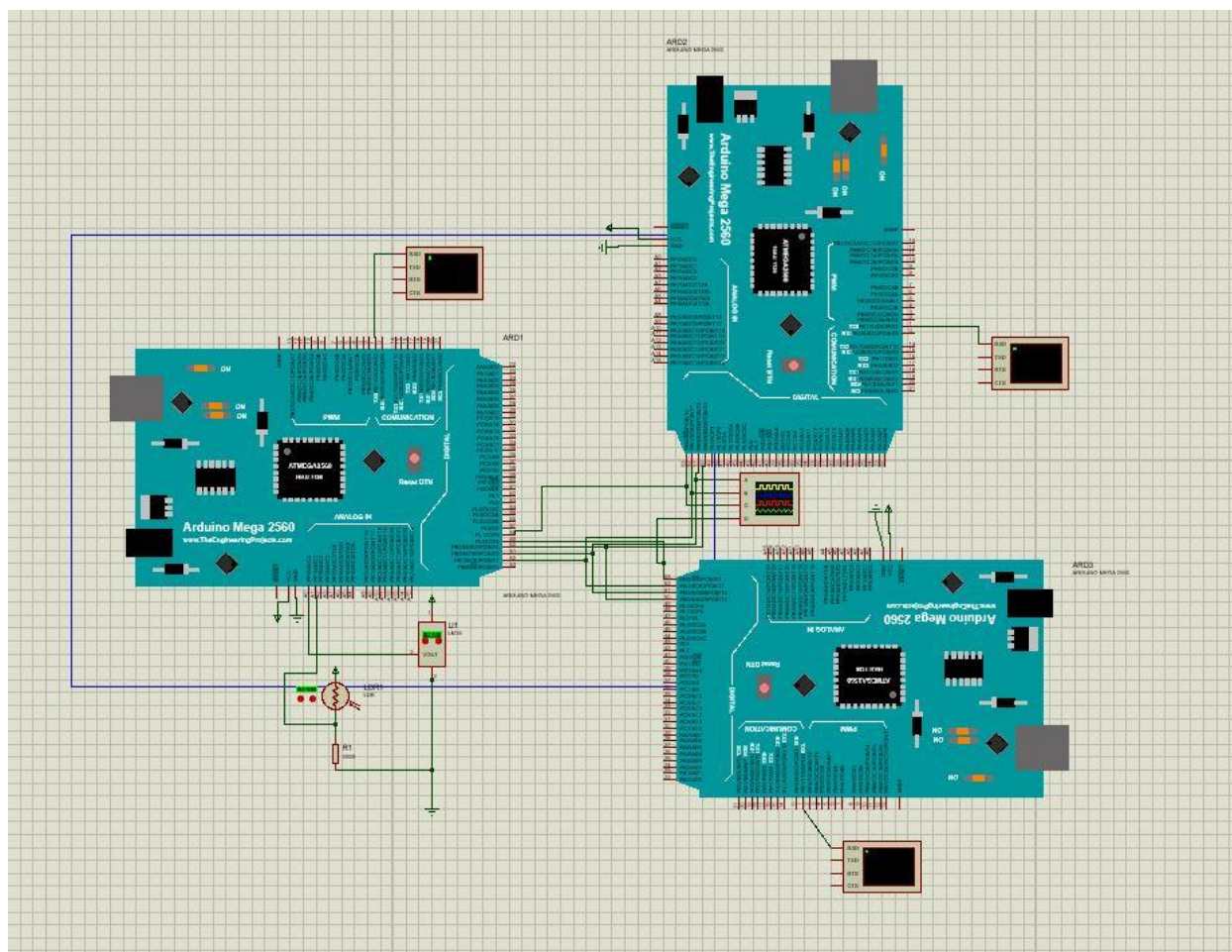
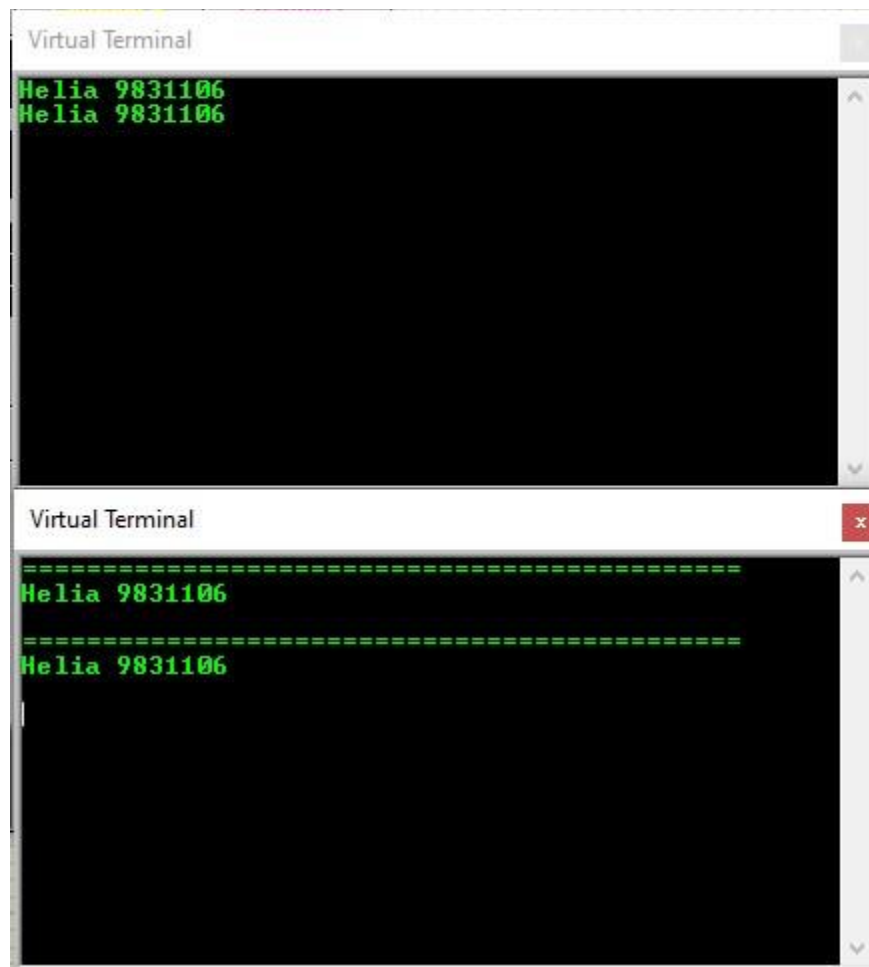


شرح آزمایش:

در ابتدا مدار مورد نیاز را به صورت زیر بستیم :



1. در این بخش مستر هر یک دقیقه پیامی را برای اسلیو میفرستد و اسلیو آن را چاپ می‌کند. بالایی مستر و پایینی اسلیو است.



```
Virtual Terminal
Helia 9831106
Helia 9831106

Virtual Terminal
=====
Helia 9831106
=====
Helia 9831106
```

برای کد مستر این قسمت هم چنین عمل کردیم:
ابتدا کتابخانه‌ی مورد نظر و پین‌ها و سپس اطلاعات را به صورت رشته‌ای از بایت‌ها تعریف کردیم.
در بخش `setup` مود پین‌ها را مشخص و سپس در `loop` تابع مورد نظر را که `selector` برده را تعیین و اطلاعات را بایت به بایت میفرستد صدا کردیم. پس از اتمام ارسال `slave select` را `high` میکند.

کد مستر:

```
#include <SPI.h>
#define SS0 48
#define SS1 49
#define MISO 50
#define MOSI 51
#define SCK 52
byte myInfo[] = "Elia 9831106";
volatile byte process_it=0;

void setup() {
  pinMode(SS0, OUTPUT);
  pinMode(SS1, OUTPUT);
  digitalWrite(SS0, HIGH);
  digitalWrite(SS1, HIGH);
  pinMode(MISO, INPUT);
  pinMode(MOSI, OUTPUT);
  pinMode(SCK, OUTPUT);
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV128);

  Serial.begin(9600);
}

void loop() {
  printInfo();
  delay(1000);
}

void printInfo()
{
  digitalWrite(SS0, LOW);
  delay(100);
  for (int i=0; i<sizeof myInfo; i++)
  {
    SPI.transfer(myInfo[i]);
    Serial.print((char)myInfo[i]);
  }
  Serial.println();
  delay(100);
  digitalWrite(SS0, HIGH);
}
```

کد اسلیو: مانند مستر کتابخانه را تعریف کردیم، در این بخش داده‌ها بایت به بایت دریافت و در ریخته میشوند. تا ۱۳ بایت مورد نظر کامل دریافت و چاپ شده و دوباره داده‌ی ارسالی بعدی دریافت شود.

```

#include<SPI.h>
volatile int i = 0;
byte myArray[13];

void setup()
{
    Serial.begin(9600);
    pinMode(SS, INPUT_PULLUP);
    pinMode(MOSI, OUTPUT);
    pinMode(SCK, INPUT);
    SPCR |= _BV(SPE);
    SPCR |= _BV(SPIE);
    SPI.attachInterrupt(); //allows SPI interrupt
}

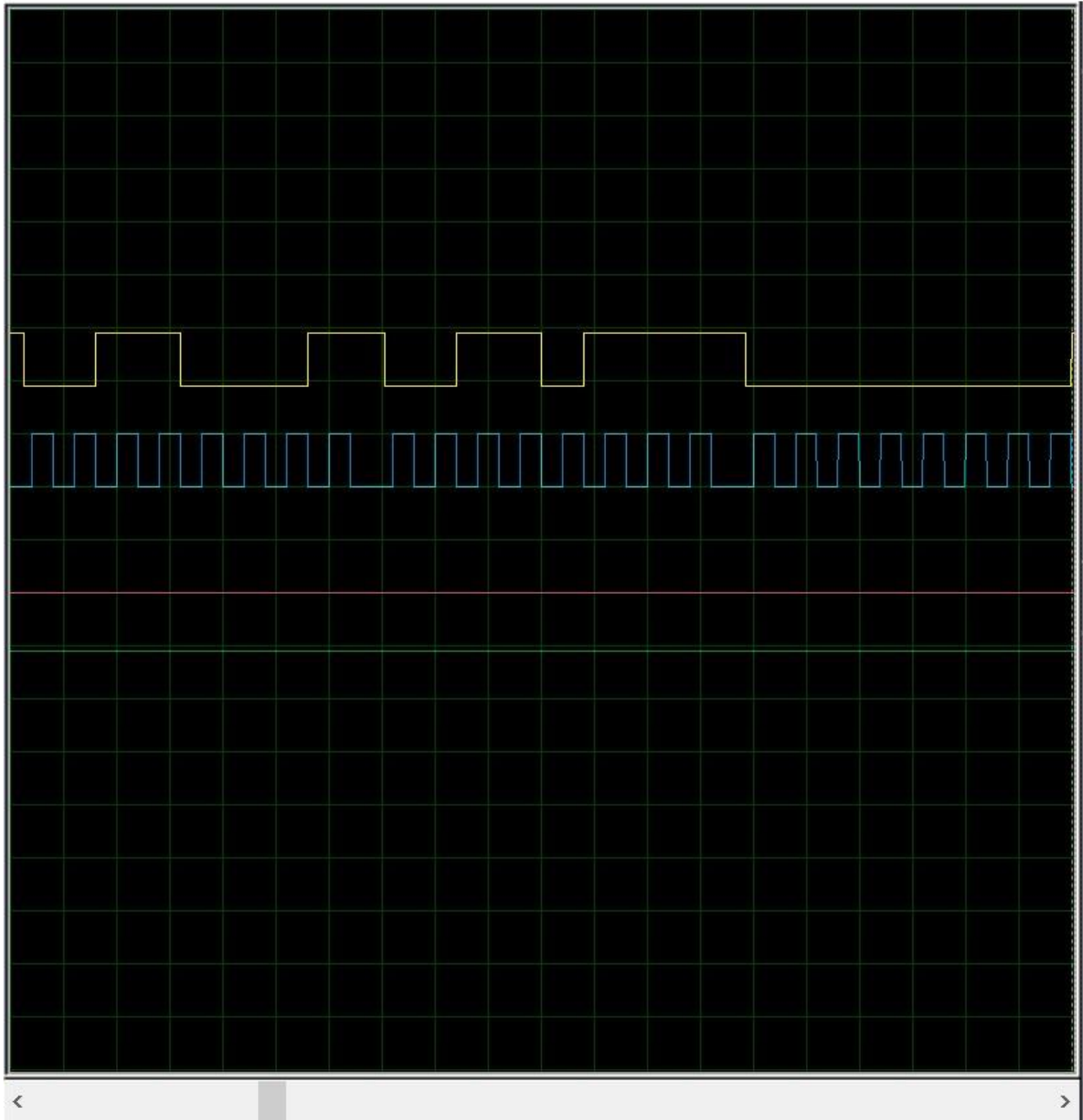
void loop(void)
{
    if (i == 13)
    {
        int x = (int)myArray[0]<<8|(int)myArray[1];
        String myString = String((char*)myArray);
        Serial.println("=====");
        Serial.println(myString);
        i=0;
    }
}

ISR (SPI_STC_vect) //Interrupt routine function
{
    myArray[i] = SPDR;
    i++;
}

```

2. نتایج اسیلوسکوپ:

Digital Oscilloscope




```

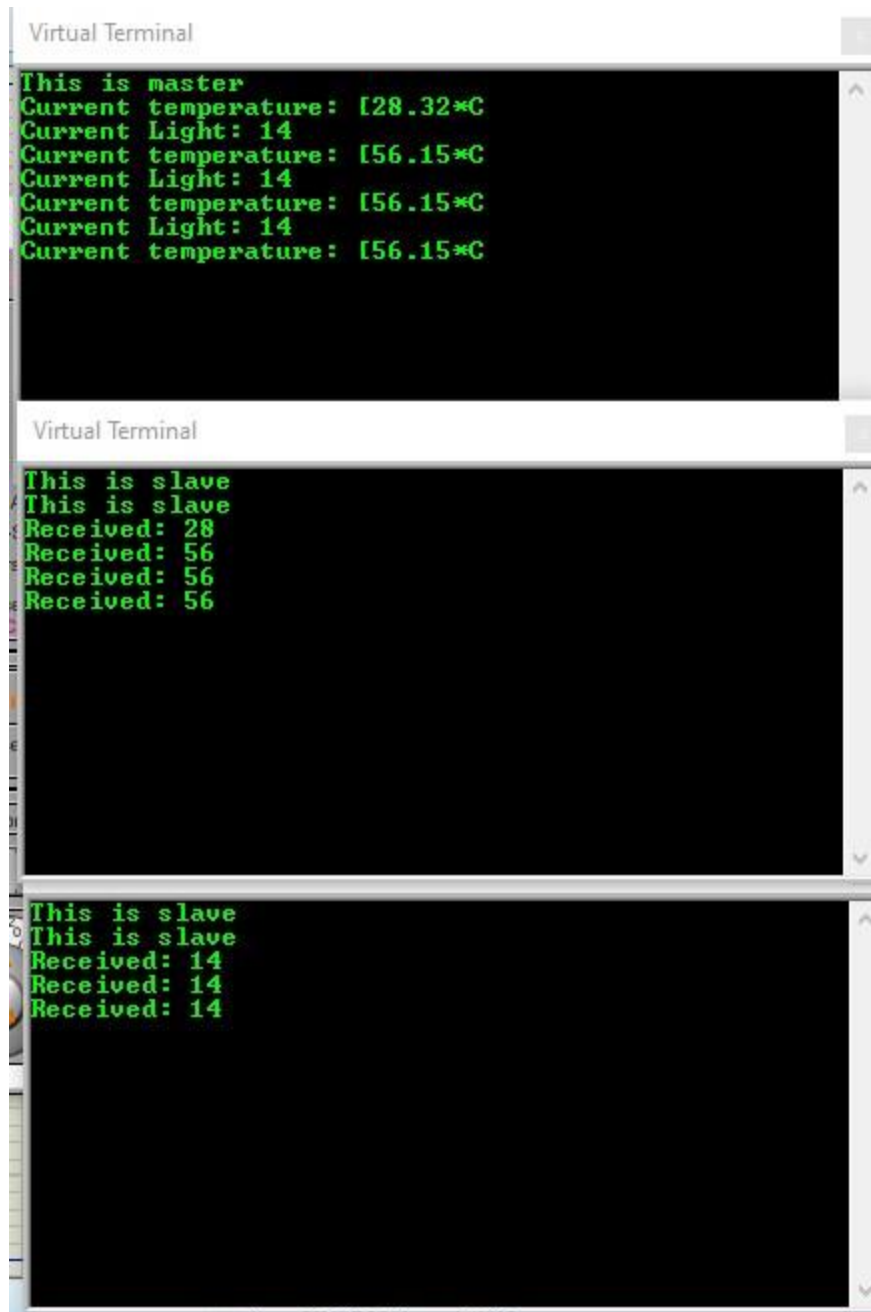
void printHiHello(){
  Serial.println("Sending>>HELLOWHELIA");
  digitalWrite(SS, LOW);//Enable Slave0
  delay(50);
  for(int i=0; i<11; i++){
    SPI.transfer(HELLO_WORLD[i]);
    Serial.print((char)HELLO_WORLD[i]);
    delay(10);
  }
  delay(50);
  digitalWrite(SS, HIGH);//Enable Slave0
  delay(500);

  Serial.println("Sending>>HELIA");
  digitalWrite(SS1, LOW);//Enable Slave0
  delay(50);
  for(int i=0; i<5; i++){
    SPI.transfer(HI[i]);
    Serial.print((char)HI[i]);
    delay(10);
  }
  delay(50);
  digitalWrite(SS1, HIGH);//Enable Slave0
  delay(500);
}

```

کد اسلیو برای این دو بخش دقیقا مانند بخش ۱ است و تنها ۱ متناسب با طول آن‌ها تنظیم می‌شود.

4. مستر دما و حرارت را دریافت کرده و اسلیوها چاپ میکنند.



The image shows three stacked screenshots of a 'Virtual Terminal' window. Each window has a title bar that says 'Virtual Terminal'. The first window shows the master's output: 'This is master' followed by three pairs of 'Current temperature' and 'Current Light' readings. The second window shows the first slave's output: 'This is slave' (twice), followed by three 'Received' values. The third window shows the second slave's output: 'This is slave' (twice), followed by three 'Received' values.

```
This is master
Current temperature: [28.32*C
Current Light: 14
Current temperature: [56.15*C
Current Light: 14
Current temperature: [56.15*C
Current Light: 14
Current temperature: [56.15*C

This is slave
This is slave
Received: 28
Received: 56
Received: 56
Received: 56

This is slave
This is slave
Received: 14
Received: 14
Received: 14
```

کد مستر و اسلیو این بخش خیلی شبیه به بخش‌های قبل است و کد هر دو برده یکی است. زیرا طول داده‌ی دریافتی آن‌ها یکی است.

کد مستر:

```
void sendTempLight(){
    digitalWrite(SS, LOW); //Enable Slave0
    delay(50);
    float temp = analogRead(A0);
    temp *= 0.48828125;
    Serial.println("Current temperature: " + String(temp)+"°C");
    byte tempTemperature[2];
    tempTemperature[0] = (int)temp;
    tempTemperature[1] = (int)temp >> 8;

    for(int i=0; i<2; i++){
        SPI.transfer(tempTemperature[i]);
        delay(10);
    }
    delay(50);
    digitalWrite(SS, HIGH); //Enable Slave0
    delay(500);

    digitalWrite(SS1, LOW); //Enable Slave0
    delay(50);

    int analogLight = analogRead(A1)-3;
    byte tempLight[2];
    tempLight[0] = analogLight;
    tempLight[1] = analogLight >> 8;
    Serial.println("Current Light: " + String(analogLight));
    for(int i=0; i<2; i++){
        SPI.transfer(tempLight[i]);
        delay(10);
    }
    delay(50);
    digitalWrite(SS1, HIGH); //Enable Slave0
    delay(500);
}
```

کد اسلیوها:

```
volatile int i = 0;
byte myArray[2];

void setup() {
  Serial.begin(9600);
  Serial.println("This is slave");
  pinMode(SS, INPUT_PULLUP);
  pinMode(MOSI, OUTPUT);
  pinMode(SCK, INPUT);
  SPCR |= _BV(SPE);
  // SPCR |= _BV(SPE);
  SPCR |= _BV(SPIE); //interrupts
  SPI.attachInterrupt(); //allows SPI interrupt
  Serial.println("This is slave");
}

void loop() {
  if (i == 2)
  {
    // int x = (int)myArray[0]<<8|(int)myArray[1];
    Serial.print("Received: ");
    int x = myArray[0] + ( myArray[1] << 8);
    // Serial.print(myArray[0]);
    // if(myArray[1] != 0)
    Serial.print(x);

    Serial.println("");
    i=0;
  }
}

ISR (SPI_STC_vect) //Interrupt routine function
{
  myArray[i] = SPDR;
  i++;
}
```

