

Laboratorium z przedmiotu Systemy Wbudowane (SW)				
Zadanie nr 3				
Prowadzący:	Temat zajęć:	Autor:	Grupa dziekańska:	Ocena:
mgr inż. Ariel Antonowicz	Arduino UNO - elementy pomiarowe	145383	i2.2	

Zadanie 1.

OneWire jest rodzajem interfejsu elektronicznego jak również i protokołu komunikacyjnego pomiędzy dwoma (lub więcej) urządzeniami. Jego nazwa wywodzi się stąd, że do całkowitej komunikacji używana jest tylko jedna linia danych. Dodatkowo, odbiornik może być zasilany bezpośrednio z linii danych, wykorzystując zasilanie pasożytnicze, co jest ogromną zaletą tego interfejsu. Odbiornik wyposażony jest bowiem w kondensator o pojemności 800 pF, króty jest ładowany bezpośrednio z linii danych - następnie energia w nim zgromadzona używana jest do zasilania odbiornika.

Kod źródłowy programu:

```
#include <OneWire.h>
#include <DS18B20.h>

// Numer pinu do którego podłączasz czujnik
#define ONEWIRE_PIN 2

// Adres czujnika
byte address[8] = {0x28, 0x12, 0xB, 0x79, 0x97, 0x11, 0x3, 0xF8};

OneWire onewire(ONEWIRE_PIN);
DS18B20 sensors(&onewire);

int i = 0;
float temperatures[18];

int ma(float arr[]);
int mi(float arr[]);
float srednia(float arr[]);

void setup() {
  while(!Serial);
  Serial.begin(9600);

  sensors.begin();
  sensors.request(address);
}

void loop() {
```

```

if (sensors.available() && i<18)
{
    temperatures[i] = sensors.readTemperature(address);

    Serial.print(temperatures[i]);
    Serial.println(F(" 'C"));

    sensors.request(address);
    i++;
} else if(i==18){
    Serial.print("Srednia: ");
    Serial.println(srednia(temperatures));
    i++;
}

}

```

```

int ma(float arr[]){
    float maxi = arr[0];
    int iter = 0;
    for (int i = 1; i<18;i++){
        if (arr[i] > maxi){
            maxi = arr[i];
            iter = i;
        }
    }
    return iter;
}

```

```

int mi(float arr[]){
    float mini = arr[0];
    int iter = 0;
    for (int i = 1; i<18;i++){
        if (arr[i] < mini){
            mini = arr[i];
            iter = i;
        }
    }
    return iter;
}

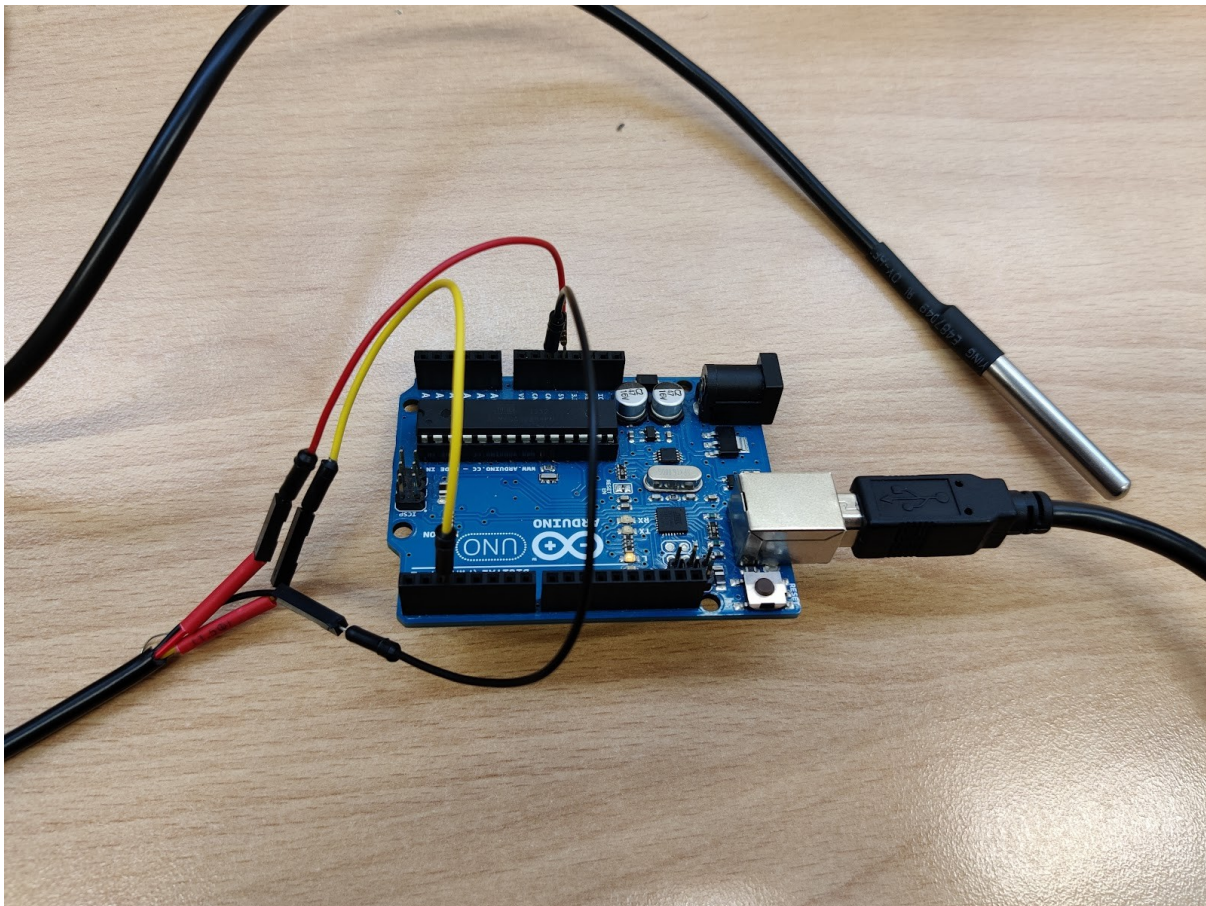
```

```

float srednia(float arr[]){
    int min_iter = mi(arr);
    int max_iter = ma(arr);
    float sum = 0.0;
    for(int i=0; i<18; i++){
        if(i != min_iter && i != max_iter){
            sum += arr[i];
        }
    }
}

```

```
}  
return sum/16;  
}
```



Zadanie 2.

SPI - jest to magistrala szeregową stosowaną w systemach opartych głównie o mikrokontrolery. Pozwala on na stosowanie synchronicznej, dwukierunkowej transmisji danych (full-duplex). Prędkość transmisji dostosowywana jest do najwolniejszego ze stosowanych układów. Transmisja poprzez standardowy interfejs SPI odbywa się po 8 bitach czyli po jednym bajcie. Jako pierwszy transmitowany jest najbardziej znaczący bit.

Kod źródłowy programu:

```
#include <OneWire.h>  
#include <DS18B20.h>  
#include <SPI.h>  
#include <MFRC522.h>  
  
// Numer pinu do którego podłączasz czujnik  
#define ONEWIRE_PIN 2  
#define SS_PIN 10
```

```

#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

// Adres czujnika
byte address[8] = {0x28, 0x12, 0xB, 0x79, 0x97, 0x11, 0x3, 0xF8};

OneWire onewire(ONEWIRE_PIN);
DS18B20 sensors(&onewire);
float temperature;

void setup() {
  while(!Serial);
  Serial.begin(9600);

  sensors.begin();
  sensors.request(address);

  SPI.begin();    // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card
}

void loop() {

  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;
  }

  if (sensors.available()){
    if(mfrc522.uid.uidByte[0] == 0xCF &&
       mfrc522.uid.uidByte[1] == 0xB9 &&
       mfrc522.uid.uidByte[2] == 0xFC &&
       mfrc522.uid.uidByte[3] == 0x7D)
    {
      temperature = sensors.readTemperature(address);

      Serial.print(temperature);
      Serial.println(F(" 'C"));

      sensors.request(address);
    }
    else{
      Serial.println("Access denied");
    }
  }
}

```

}

}

