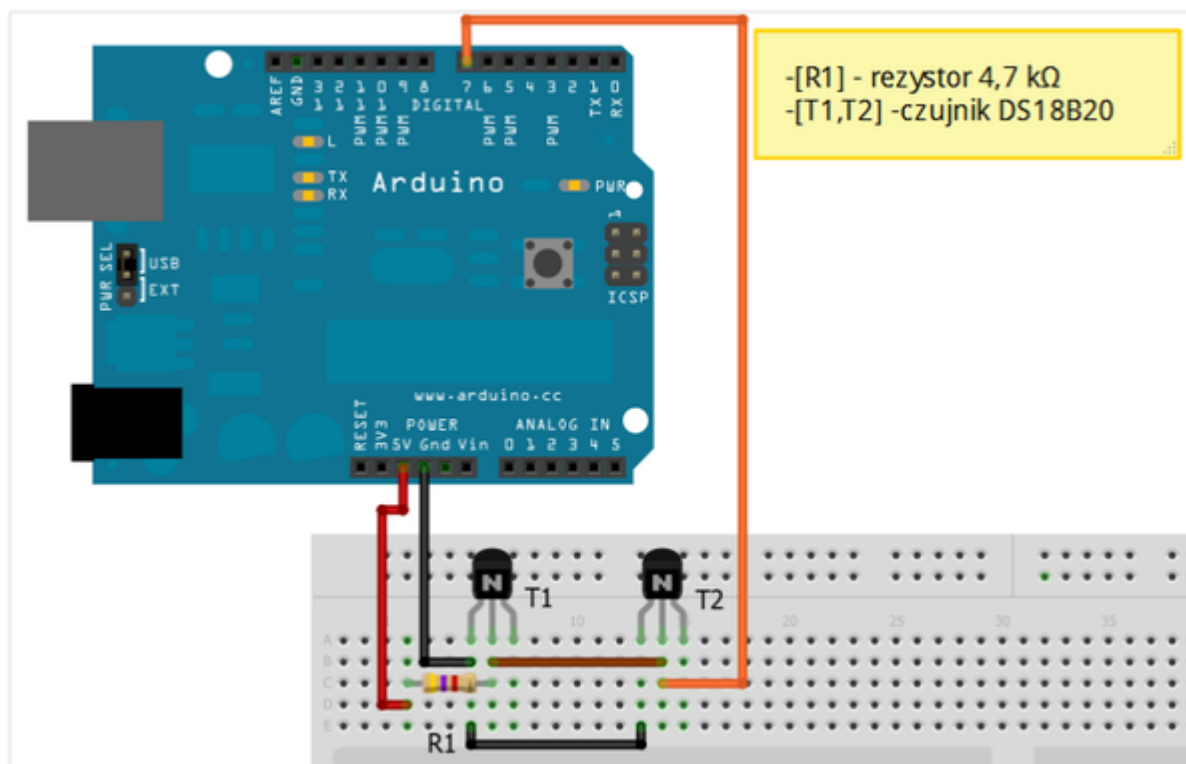


Nazwa implementacji: Protokół komunikacyjny 1-wire - rozbudowa układu o kolejne czujniki DS18B20

Autor: Krzysztof Bytow

Opis implementacji: Budowa układu i programu do odczytu danych wykorzystując interfejs 1-wire na przykładzie czujników Dallasa DS18B20. Rozszerzenie wiedzy dotyczącej adresowania czujników ich sposobów zasilania i wykorzystywania bibliotek w celu sterowania.

Schemat połączeń:

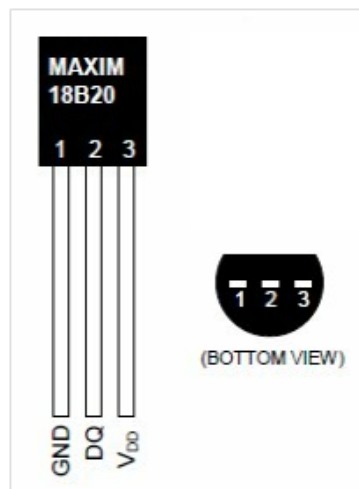


Uczeń/Uczennica po zestawieniu połączeń zgłasza nauczycielowi gotowość do sprawdzenia układu i wszystkich połączeń.

Czujnik DS18B20, aby działał należy posiadać wgraną bibliotekę DallasTemperature oraz OneWire do pobrania np.:

http://www.milesburton.com/?title=Dallas_Temperature_Control_Library#Introduction <http://www.arduino.cc/playground/Learning/OneWire>

Wyprowadzenia czujnika DS18B20:



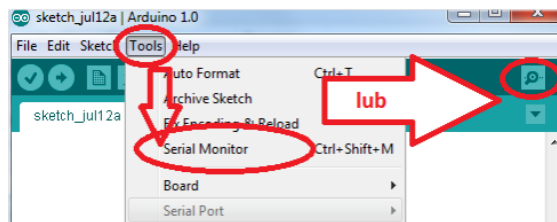
1 - GND - masa ; 2 - DQ - (dane) ; 3 - Vdd +5V

Kod implementacji:

```
#include <OneWire.h>           // podłączamy bibliotekę
#include <DallasTemperature.h> // podłączamy bibliotekę
OneWire linia_danych(7);      // tworzymy obiekt OneWire, z użyciem określonego pinu
DallasTemperature czujnik(&linia_danych); // wskazujemy OneWire

void setup()
{
  Serial.begin(9600);          // ustawienie prędkości komunikacji
  czujnik.begin();            // uruchamiamy bibliotekę
}
void loop()                   // główna pętla
{
  czujnik.requestTemperatures(); // wysłanie polecenia pobrania temperatury
  Serial.print("Czujnik numer 1: "); // wyświetlenie tekstu
  Serial.println(czujnik.getTempCByIndex(0)); // wyświetlenie tekstu - temperatury
  Serial.print("Czujnik numer 2: "); // wyświetlenie tekstu
  Serial.println(czujnik.getTempCByIndex(1)); // wyświetlenie tekstu - temperatury
  Serial.print("\r\n"); // wyświetlenie tekstu
  delay(2000); // czekaj wyznaczony czas
}
```

Po wgraniu kodu należy w programie Arduino IDE uruchomić Serial Monitor (lub terminal), aby obserwować wyniki pomiaru.



Kod implementacji nr 2 – poniższy kod sprawdza liczbę czujników podpiętych do OneWire i wyświetla ich adresy, następnie dokonuje prezentacji pomiaru temperatur.

```
#include <OneWire.h>                // podłączamy bibliotekę
#include <DallasTemperature.h>      // podłączamy bibliotekę
#define liczba_czujnikow 2
byte adresy [liczba_czujnikow][8];
byte urzadzen;

OneWire linia_danych(7);           // tworzymy obiekt OneWire, z użyciem
DallasTemperature czujniki(&linia_danych); //określonego pinu wskazujemy OneWire

void setup()
{
  Serial.begin(9600);              // ustawienie prędkości komunikacji
  czujniki.begin();               // uruchamiamy bibliotekę
  urzadzen = wyszukaj_adres();
  for (byte i=0; i < urzadzen; i++)
    czujniki.setResolution(adresy[i], 10);
}

byte wyszukaj_adres()
{
  byte j=0;
  while ((j < liczba_czujnikow) && (linia_danych.search(adresy[j])))
  {
    j++;
  }
  for (byte i=0; i < j; i++)
  {
    Serial.print("Adres czujnika numer "); // wyświetlenie tekstu
    Serial.print(i);                       // wyświetlenie tekstu
    Serial.print(": ");                     // wyświetlenie tekstu
    adres(adresy[i]);
  }
  Serial.print("\r\n");                    // wyświetlenie tekstu
  return j ;
}

void adres(DeviceAddress addr)
{
  byte i;
  for( i=0; i < 8; i++)
  {
    3
  }
}
```

```

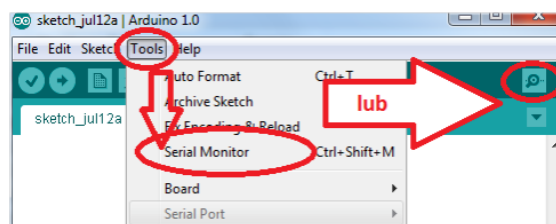
Serial.print("0x");
if (addr[i] < 16)
{
  Serial.print('0');
}
Serial.print(addr[i], HEX);
if (i < 7)
{
  Serial.print(" ");          // wyświetlenie tekstu
}
}
Serial.println("");          // wyświetlenie tekstu
}

void loop()                  // główna pętla
{
  delay(2000);                // czekaj wyznaczony czas
  czujniki.requestTemperatures();
  for (byte i=0; i < urzadzen; i++)
  {
    Serial.print("Czujnik ");  // wyświetlenie tekstu
    Serial.print(i);          // wyświetlenie tekstu
    Serial.print(" ");        // wyświetlenie tekstu
    wyswietl(adresy[i]);
    Serial.print("\n\r");     // wyświetlenie tekstu
  }
  Serial.print("\n\r");       // wyświetlenie tekstu
}

void wyswietl(DeviceAddress addr)
{
  float tempC = czujniki.getTempC(addr);
  if (tempC == -127.00)
  {
    Serial.print("Error !!!"); // wyświetlenie tekstu
  }
  else
  {
    Serial.print(tempC);       // wyświetlenie tekstu
    Serial.print(" C");       // wyświetlenie tekstu
  }
}

```

Po wgraniu kodu należy w programie Arduino IDE uruchomić Serial Monitor (lub terminal), aby obserwować wyniki pomiaru.



4



SWO1

**STRATEGIA WOLNYCH I OTWARTYCH
IMPLEMENTACJI – www.e-swoi.pl**

Filmy instruktażowe:

<http://youtu.be/615DgFv7vrQ>

