



MODUŁ – PROGRAMOWANIE MIKROKONTROLERÓW



Tytuł

Zastosowanie czujnika temperatury Arduino podczas pandemii COVID-19

Wymagane elementy

- 1x Arduino Uno
<https://ardushop.ro/ro/home/29-placa-de-dezvoltare-uno-r3.html>
- 1x Płyta robocza
<https://ardushop.ro/ro/electronica/33-breadboard-830.html>
- 1x Czujnik temperatury Lm35
<https://ardushop.ro/ro/electronica/192-senzor-temperatura-lm35dz.html>
- 1x Zestaw przewodów
<https://ardushop.ro/ro/electronica/28-65-x-jumper-wires.html>

Aplikacje

- Arduino IDE
lub
- Arduino Web Editor

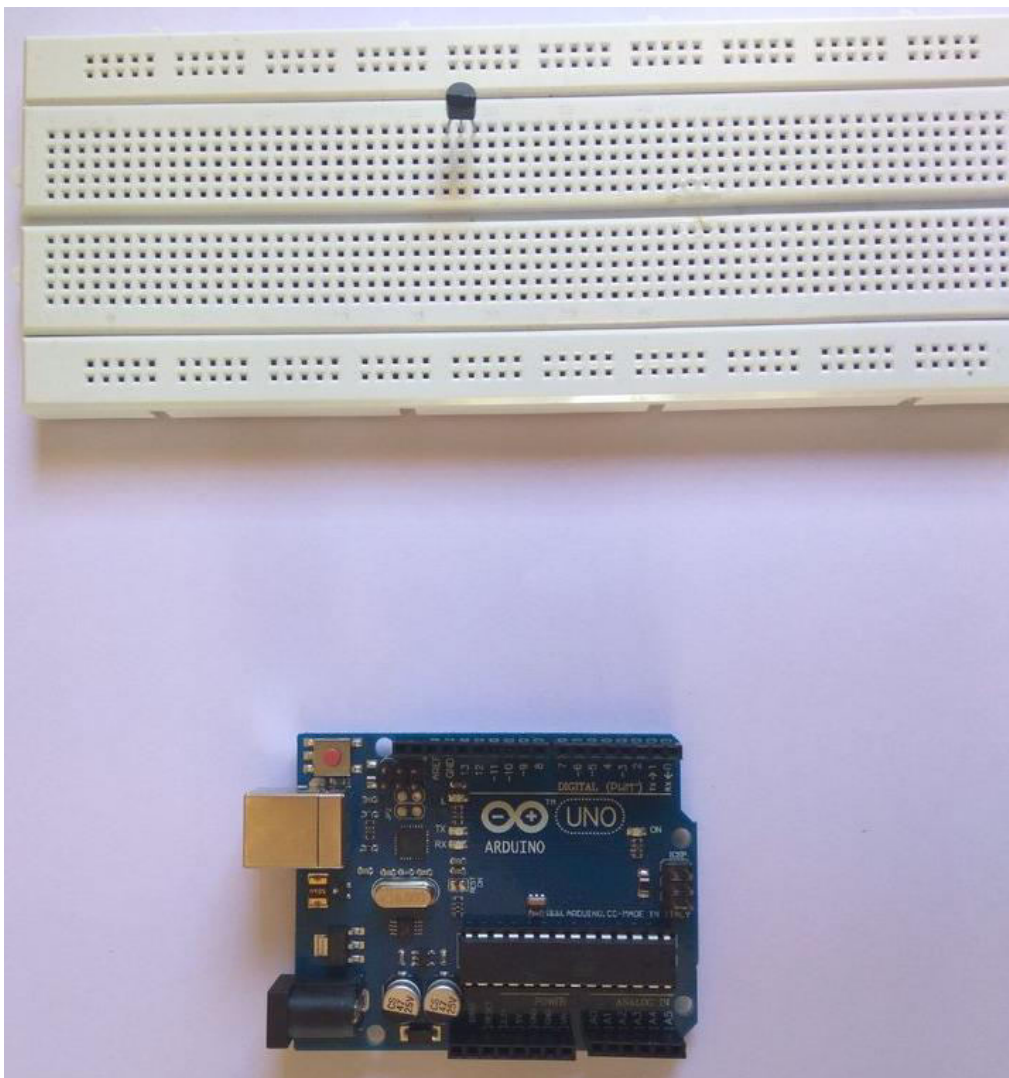
Opis

Czujnik lm35 to analogowy czujnik temperatury. Oznacza to, że napięcie wyjściowe jest proporcjonalne do temperatury. Napięcie wyjściowe wzrasta o 10mv na każdy wzrost temperatury o 1 stopień Celsjusza. Arduino może odczytywać dane wejściowe z zakresu 0-5v. Arduino przechowuje to jako 10-bitową liczbę (0-1023). Metoda, którą teraz zastosujemy może być użyta do pomiaru temperatury od 2 stopni Celsjusza do maksymalnej temperatury jaką może zmierzyć twój lm35.



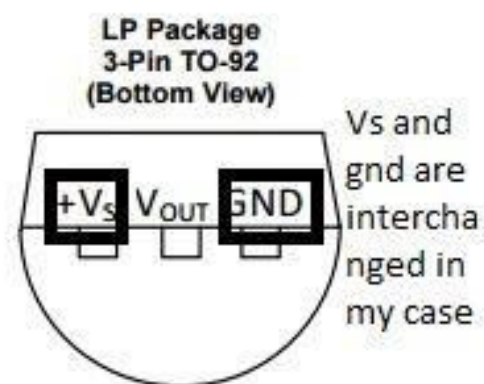
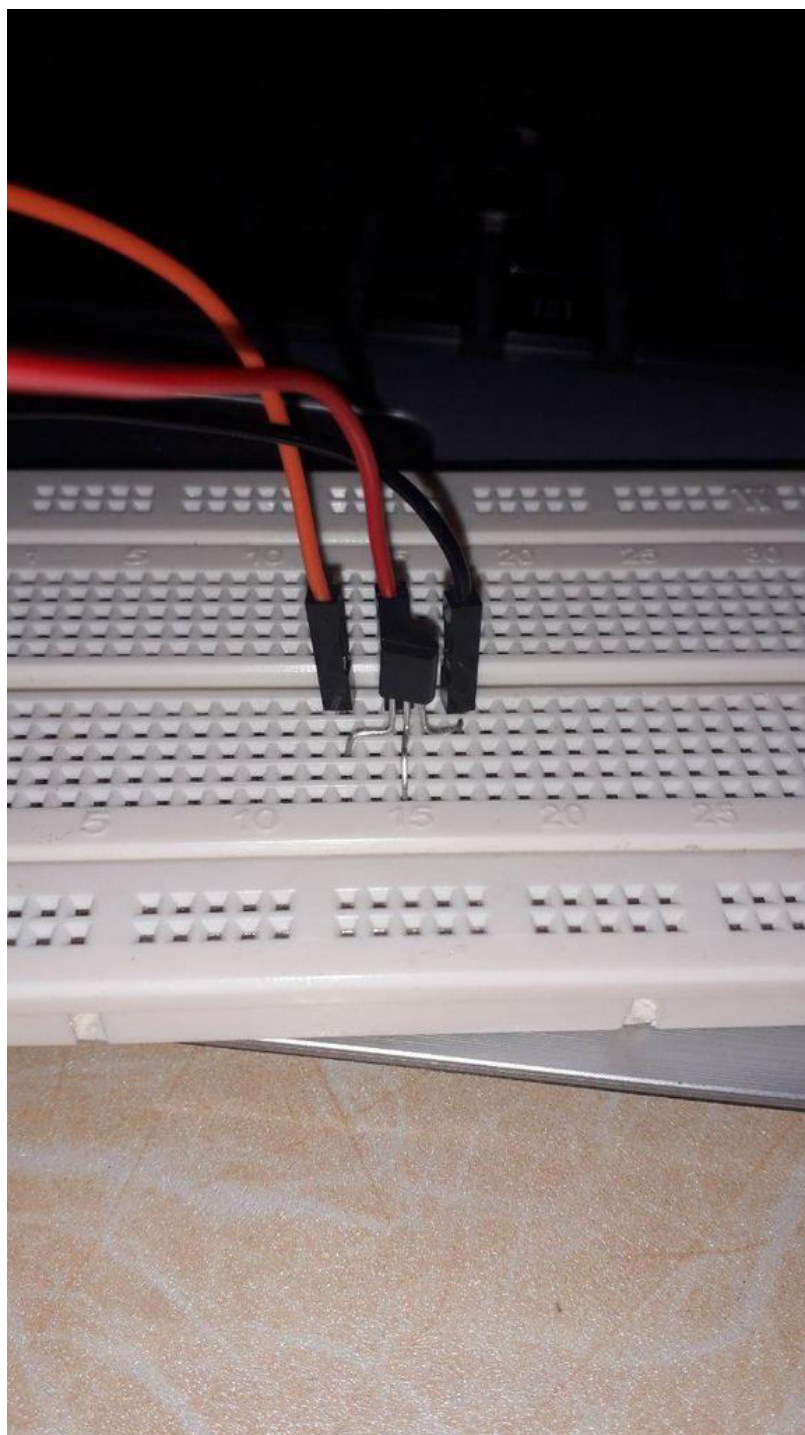
Realizacja projektu

Na początku do dyspozycji mamy płytę roboczą z czujnikiem oraz Arduino, które podłączymy do czujnika.



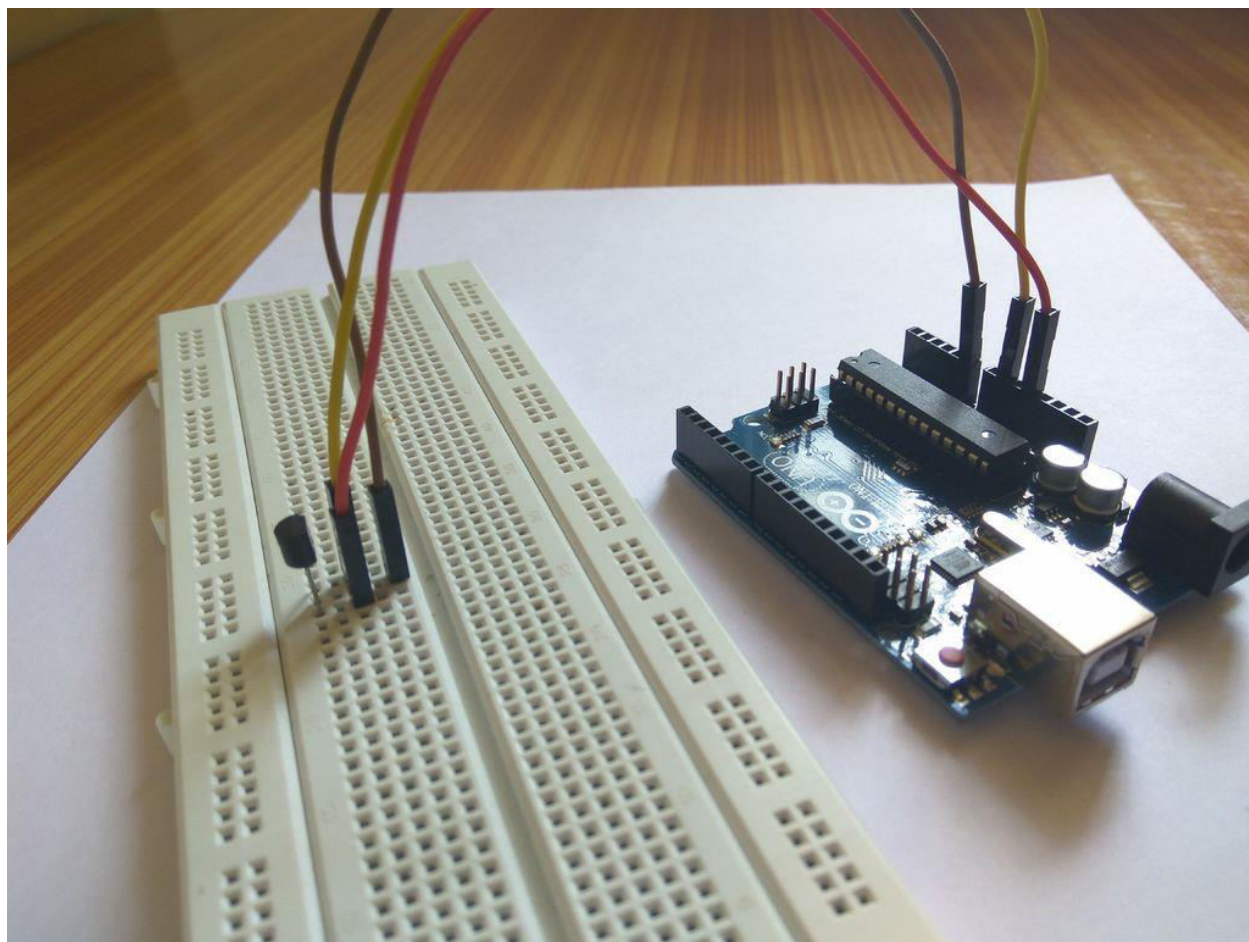


Krok 1: Tworzenie połączeń





Podłącz pin Vs do pinu 5v na Arduino i uziemnij do jednego z 2 pinów na szynie zasilania. Podłącz pin Vout do jednego z pinów analogowych, w naszym przypadku A0.



Lm35, którego użyliśmy, ma zamieniony pin uziemienia i pin Vs w porównaniu z czujnikiem Texas Instruments, którego karta katalogowa jest powszechnie dostępna. Jeśli zamienisz piny podczas podłączania czujnika, będzie się on nagrzewał, więc będziesz wiedział, czy jest nieprawidłowy.



Wskazówka:

Temperatura, którą uzyskasz nie będzie wiarygodna, jeśli połączysz wiele kabli ze sobą, aby przewód był długi.

Krok 2: Kod

Najpierw potrzebujemy ustawić.

```
int temppin = 0;

float temp;

void setup()
{
    Serial.begin(9600);
}
```

Tutaj inicjalizujemy temperaturę pinu, z którym będziemy pracować oraz zmienną, która będzie przechowywać temperaturę osoby.

Następnie piszemy funkcję o nazwie loop.

```
void loop()
{
    temp = analogRead(temppin); // Reading data from the sensor. This voltage
    is stored as a 10bit number

    temp = (5.0 * temp * 100.0) / (1024 * 10)
}
```

Ta funkcja pozwala zmiennej Temp odczytać dane, które są jej podawane z czujnika. Napięcie zostanie zapisane jako liczba 10-bitowa.

Po tym używamy prostego wzoru, 5.0 (.0 tak, że bierzemy wszystkie liczby rzeczywiste i ich dziesiętne przy użyciu operatorów) * temp * 100.0 (ta sama historia, którą przedstawiliśmy wcześniej) / 1024 * 10 -> **(5.0 * temp * 100.0) / (1024. 10)**

Część 5 * temp / 1024 służy do konwersji liczby 10-bitowej na odczyt napięcia. Zostanie to pomnożone przez 1000, aby przekonwertować go na miliwolt, a następnie podzielone przez 10, ponieważ każdy wzrost stopnia powoduje wzrost o 10 miliwoltów.



Następnie zapisujemy szkic w Arduino IDE.

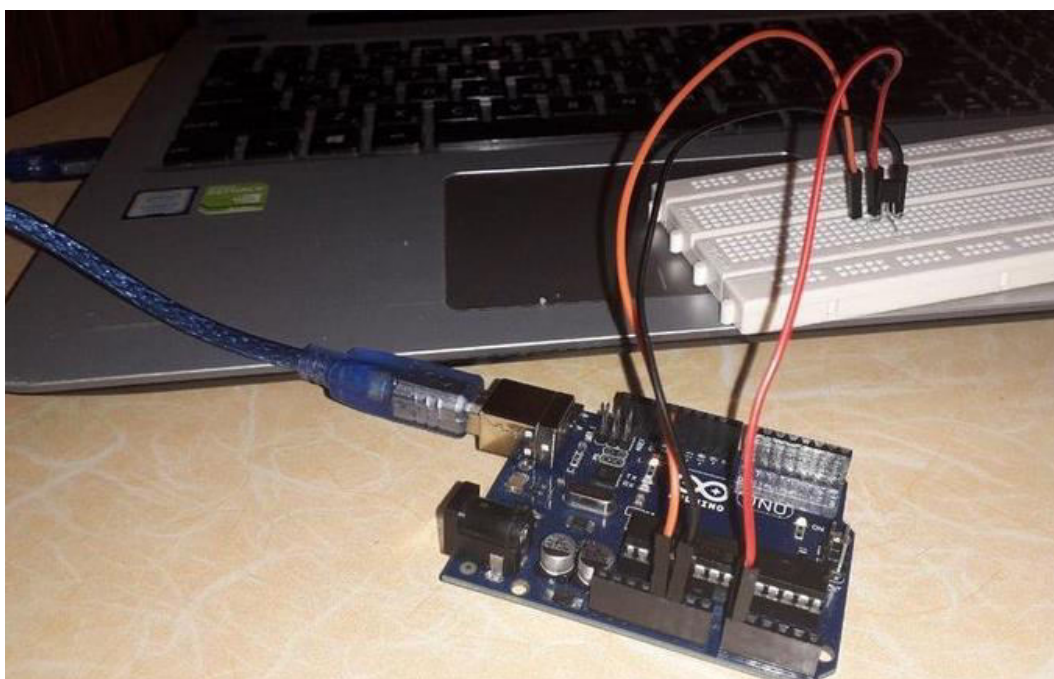
```
temperature_arduino | Arduino 1.8.13 (Windows Store 1.8.39.0)
File Edit Sketch Tools Help

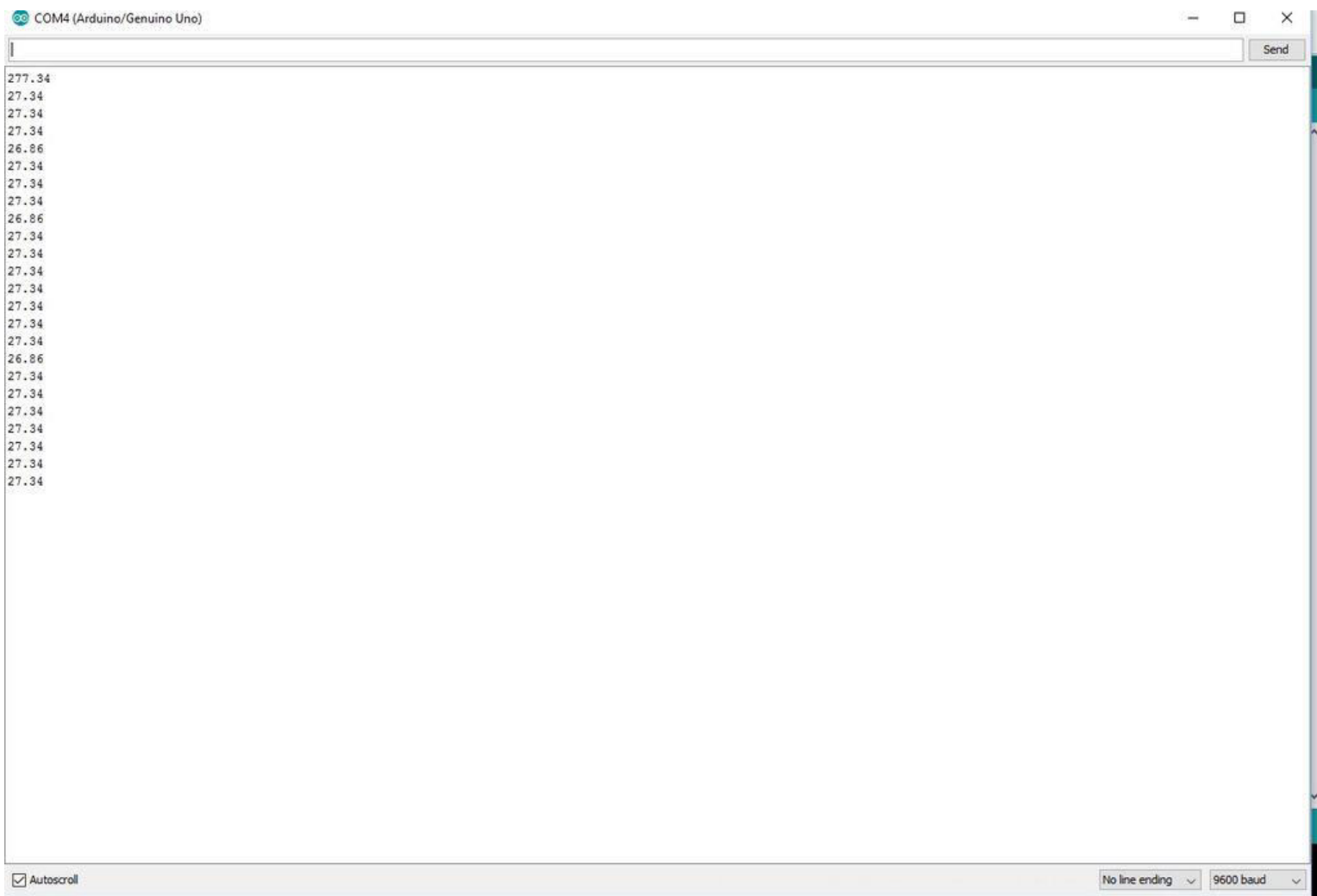
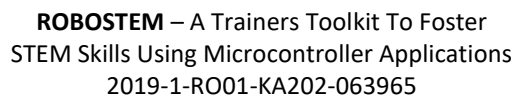
temperature_arduino $
int tempin = 0;

float temp;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  temp = analogRead(tempin); // Reading data from the sensor. This voltage is stored as a 10bit number
  temp = (5.0 * temp * 100.0)/(1024 *10)
  /* 5 * temp /1024 is to convert the 10bit number to a voltage reading.
  This is multiplied by 1000 to convert it to millivolt.
  We then divide it by 10 because each degree rise results in a 10 millivolt increase*/
}
Serial.println(temp);
delay(800); // This is because we don't want a continuous stream of data
```

Następnie wgrywamy kod do Arduino Uno, który powinien wyglądać na tym etapie mniej więcej tak:





Tutaj możemy zobaczyć temperaturę obiektu, na którym testowaliśmy nasz czynn timer temperatury sterowany przez Arduino.



Kod źródłowy

```
int temppin = 0;

float temp;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    temp = analogRead(temppin); // Odczyt danych z czujnika. Napięcie to jest
    zapisywane jako liczba 10-bitowa

    temp = (5.0 * temp * 100.0)/(1024 *10)

    /* 5 * temp /1024 służy do konwersji 10-bitowej liczby na odczyt napięcia.
    Jest ono mnożone przez 1000 aby przeliczyć je na miliwolt.
    Następnie dzielimy przez 10, ponieważ każdy wzrost stopnia powoduje wzrost
    napięcia o 10 miliwoltów*/
}

Serial.println(temp);

delay(800); // Dzieje się tak dlatego, że nie chcemy ciągłego strumienia d
anych
```

Tutaj mamy kod źródłowy z kilkoma komentarzami, które powinny pomóc każdemu lepiej zrozumieć, co zrobiliśmy.



Cennik

- 1x Arduino Uno – 18,48 zł
- 1x Płyta robocza – 9,28 zł
- 1x Czujnik temperatury Lm35 - 12,47 zł
- 1x Zestaw przewodów - 12,37 zł

Łączna cena to około 52,60 zł

Przydatne linki

- <https://arduinomodules.info/>
- <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>
- <https://www.youtube.com/watch?v=nL34zDTPkcs>
- https://www.youtube.com/watch?v=QO_Jlz1qpDw
- <https://randomnerdtutorials.com/9-arduino-compatible-temperature-sensors-for-your-electronics-projects/>