

IDOM

Instrukcja użytkownika

Moduł termometru

Spis treści

1	Zastosowanie	2
2	Potrzebne elementy	2
3	Schemat połączeń	2
4	Programowanie modułu	2

1 Zastosowanie

Moduł termometru służy do pomiaru temperatury powietrza wewnątrz oraz na zewnątrz (tylko w przypadku wodoodpornej wersji czujnika). Maksymalny zakres pomiaru czujnik to -55°C do $+125^{\circ}\text{C}$. Dokładność $\pm 0.5^{\circ}\text{C}$ jest zadeklarowana dla zakresu od -10°C do $+85^{\circ}\text{C}$.

2 Potrzebne elementy

Nazwa	ilość
Koszyk na conajmniej 3 baterie AA z włącznikiem	1
Przetwornica step-down LM2596 1,5-35V 3A	1
ESP8266-12	1
Adapter PCB do modułów ESP	1
Czujnik temperatury ds18b20 (wodoodporna wersja, jeśli termometr będzie używany na zewnątrz)	1
Rezystor $4,7\text{k}\Omega$	1

3 Schemat połączeń

4 Programowanie modułu

Aby czujnik ds18b20 mógł działać potrzebujemy zainstalować 2 biblioteki. W tym celu w programie Arduino IDE otwieramy Narzędzia » Zarządzaj bibliotekami... Wpisujemy "onewire" i wybieramy bibliotekę stworzoną przez Jim Studt, Tom Pollard, Robin James, Glenn Trwitt, Jason Dangel, Guillermo Lovato, Paul Stoffregen, aż do Love Nystrom (nie wymieniam wszystkich autorów, jest ich za dużo). 2 biblioteka, której potrzebujemy to "DallasTemperature" stworzona przez Miles Burton, Tim Newsome, Guil Barros i Rob Tillaart. Po zainstalowaniu obu bibliotek musimy wyłączyć i ponownie włączyć program Arduino IDE.

Teraz możemy wgrać oprogramowanie na nasze ESP. Do zaprogramowania należy użyć pliku [Temperature_sensor.ino](https://www.arduino.cc/en/Reference/OneWire). Jeśli nie wiesz jak zaprogramować moduł ESP skorzystaj z poradnika "Wgrywanie oprogramowania na mikrokontrolery ESP". W poradniku tym znajdują się też 2 inne potrzebne biblioteki, jeśli go pominąłeś musisz je zainstalować (znajdą się na pod koniec instrukcji).

Na górze pliku znajduje się kilka zmiennych do ustawienia

```
/***** Variables to customize *****/
```

Najpierw ustawiamy nazwę naszej sieci (musi to być sieć 2.4GHz)

```
// Name of your network  
const char* ssid = "";
```

Następnie ustawiamy hasło naszej sieci

```
// Password of your network  
const char* password = "";
```

w tym miejscu wpisujemy adres naszego serwera razem z dopisanym portem

8001 na przykład "192.168.0.10:8001"

```
// Raspberry server address here  
const String ServerName = "";
```

Ustawiamy nazwę sensora na taką samą jak w aplikacji IDOM

```
// Name of your sensor  
String Name = "";
```

Ustawiamy numer pinu jaki użyliśmy do podłączenia termometru, domyślnie zgodnie ze schematem jest to pin 2

```
// Your GPIO pin number  
const int GPIO = 2;
```

Po ustawieniu tych wartości możemy wgrać kod a następnie cieszyć się działającym modulem termometru.