

## TERMOMETR CYFROWY DS18B20

### SPOSÓB PODŁĄCZENIA I PROGRAMOWANIA TERMOMETRU DS18B20

---

#### WPROWADZENIE

Termometr cyfrowy DS18B20 jest czujnikiem temperatury wyprodukowanym przez firmę Dallas. Czujnik wykorzystuje protokół One-Wire, który umożliwia podłączenie do jednego pinu cyfrowego wiele czujników. Domyślnie czujnik temperatury zwraca informację w stopnia Celsjusza. Czujnik ma rozdzielczość między 9-12 bitów. Oznacza to, że w zależności od wybranej rozdzielczości można otrzymać dokładniejszy wynik (im wyższa rozdzielczość tym dłuższy czas oczekiwania na wynik). Dokładność pomiaru czujnika temperatury wynosi  $\pm 0.5^{\circ}\text{C}$  w zakresie od  $-10^{\circ}\text{C}$  do  $85^{\circ}\text{C}$ . Oznacza to, że w przypadku gdy w analizowanym pomieszczeniu jest  $22.5^{\circ}\text{C}$  to informacja odebrana z czujnika oscylować będzie między  $22$  a  $23^{\circ}\text{C}$ . Należy jednak pamiętać, że wszystko zależy od indywidualnej charakterystyki czujnika. Kolejnym ważnym aspektem w sytuacji zbierania pomiarów czujnika jest tzw. dryf pomiaru, polegający na tym, że przy tej samej mierzonej temperaturze, termometr w jednym czasie może wskazać inną temperaturę, a w innym czasie inną. Dryf opisywanego czujnika to  $\pm 0.2^{\circ}\text{C}$ . Oznacza to, że w przypadku temperatury  $24^{\circ}\text{C}$  czujnik co pewien czas może pokazywać wyniki z zakresu  $23.8$  do  $24.2^{\circ}\text{C}$ . Podsumowując, gdy czujnik temperatury wskazuje temperaturę  $24^{\circ}\text{C}$ , to założyć można, że rzeczywista temperatura w pomieszczeniu mieści się w zakresie  $23.0$ - $25.0^{\circ}\text{C}$  (uwzględniając wartości graniczne możliwych błędów).



Fotografia 1. DS18B20 w wodoodpornej obudowie.

#### ONE-WIRE

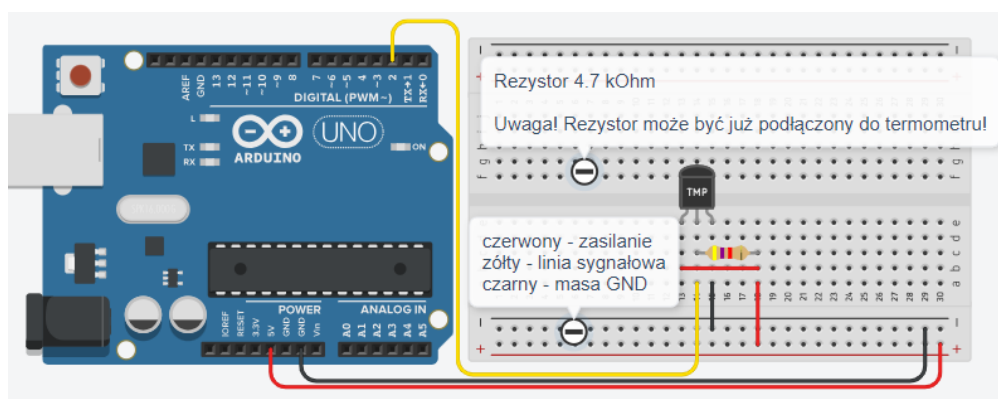
1-Wire – interfejs elektroniczny, jak również i protokół komunikacyjny pomiędzy urządzeniami. Jego nazwa wywodzi się z faktu, że do komunikacji używana jest tylko jedna linia danych (oraz linia zerowa). Odbiornik może być zasilany bezpośrednio z linii danych, wykorzystując zasilanie pasożytnicze, co jest zaletą tego interfejsu. Odbiornik wyposażony jest w kondensator o pojemności  $800\text{ pF}$ , który jest ładowany z linii danych – następnie energia w nim zgromadzona używana jest do zasilania odbiornika. Protokół 1-Wire został opracowany przez firmę Dallas Semiconductor. Zapewnia on stosunkowo niewielką przepustowość transmisji danych – standardowo  $16\text{ kbps}$

w trybie regular do 115,2 kbps w trybie overdrive. 1-Wire jest podobne do interfejsu I<sup>2</sup>C, lecz z uwagi na pojedynczą linię komunikacyjną jest zarówno wolniejsze, jak i tańsze. Interfejs 1-Wire jest zazwyczaj używany do komunikacji pomiędzy niewielkimi urządzeniami, takimi jak: termometry cyfrowe, instrumenty metrologiczne, sterowniki ładowania akumulatorów, zamki elektroniczne typu iButton itd. Urządzenia takie są zazwyczaj zamknięte w solidnej obudowie, często znacznie większej niż samo urządzenie elektroniczne.

Całość układu jest zazwyczaj zasilana napięciem stałym o wartości 5 V. Linia danych jest podłączona do zasilania przez rezystor (tzw. pull-up), co powoduje jednocześnie zasilanie odbiorników. Przesłanie każdego bitu informacji jest inicjowane przez urządzenie typu master. Master rozpoczyna transmisję sekwencji bitów poprzez wystawienie impulsu reset, czyli zwarcia linii danych na 480  $\mu$ s do masy. Powoduje to zresetowanie wszystkich podłączonych odbiorników (urządzeń typu slave). Następnie każde urządzenie slave potwierdza swoją obecność wystawiając na linię danych impuls obecności – zwierając linię danych do masy na 60  $\mu$ s. Przesłanie logicznej jedynki na magistralę oznacza wystawienie przez mastera krótkiego (od 1 do 15  $\mu$ s) impulsu niskiego (zwarcie linii) oraz następnie wysokiego o długości 60  $\mu$ s. Logiczne zero odpowiada niskiemu impulsowi o długości 60  $\mu$ s. Opadające zbocze impulsu aktywuje przerzutnik astabilny w urządzeniu slave. Przerzutnik ten taktuje wewnętrzny mikroprocesor, co powoduje odczyt danych z linii po ok. 30  $\mu$ s od momentu pojawienia się zbocza rosnącego. Z uwagi na wewnętrzne opóźnienia urządzenia slave czas trwania pojedynczego impulsu musi wynosić właśnie 60  $\mu$ s (a impuls startujący nie może być dłuższy niż 15  $\mu$ s) – zapewnia to poprawny odczyt danych w każdej sytuacji.

Przed odbiorem każdego bitu danych master wysyła niski impuls startu (od 1  $\mu$ s do 15  $\mu$ s), po czym wraca do stanu wysokiego na linii danych. Jeśli slave wysyła logiczną jedynkę – nie robi nic, pozostawiając linię w stanie wysokim. Jeśli slave wysyła zero, wówczas zwierza linię danych do masy na 60  $\mu$ s. Po przesłaniu 8 bitów następuje wysłanie komendy (rozkażu) (również ośmiobitowej). Ewentualne błędy w transmisji mogą być wykryte za pomocą wbudowanego algorytmu CRC-8.

## SPOSÓB PODŁĄCZENIA CZUJNIKA DO PŁYTKI ARDUINO UNO



## ODCZYTANIE ADRESU CZUJNIKA TEMEPRATURY DS18B20

```
#include <OneWire.h>

const byte ONEWIRE_PIN = 2; // Numer pinu cyfrowego
OneWire onewire(ONEWIRE_PIN);

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

void loop()
{
    byte address[8];

    onewire.reset_search();
    while(onewire.search(address))
    {
        if (address[0] != 0x28)
            continue;

        if (OneWire::crc8(address, 7) != address[7])
        {
            Serial.println(F("Błędny adres, sprawdź połączenia"));
            break;
        }

        for (byte i=0; i<8; i++)
        {
            Serial.print(F("0x"));
            Serial.print(address[i], HEX);

            if (i < 7)
                Serial.print(F(", "));
        }
        Serial.println();
    }

    while(1);
}
```

Wynikiem działania powyższego programu jest wyświetlenie adresu czujnika temperatury w monitorze portu szeregowego (np. 0x28, 0x87, 0x6A, 0xA1, 0x3, 0x0, 0x0, 0x1F).

## ODCZYTYWANIE TEMPERATURY

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

#define ONEWIRE_PIN 2 // Numer pinu cyfrowego

byte address[8] = {0x28, 0xB1, 0x6D, 0xA1, 0x3, 0x0, 0x0, 0x11}; // Adres

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

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

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

void loop() {
    if (sensors.available())
    {
        float temperature = sensors.readTemperature(address);

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

        sensors.request(address);
    }

    // tu umieść resztę twojego programu
    // Będzie działał bez blokowania
}
```

## ŹRÓDŁA

- [1] <http://akademia.nettigo.pl/ds18b20/>
- [2] <http://www.easy-soft.net.pl/artykuly/techniki-technologie-dla-elektronikow/1-wire-podstawy-protokolu-i-budowy-sieci>
- [3] <https://pl.wikipedia.org/wiki/1-Wire>