

# **Podstawy Systemów Mikroprocesorowych**

Lab. 10. Magistrala 1-Wire. v.0.1

Dariusz Tefelski

2020-11-15

## Spis treści

<b>Laboratorium 10.</b>	<b>3</b>
Magistrala 1-Wire. . . . .	3
Część 1: . . . . .	3
Część 2: . . . . .	4
Część 3: . . . . .	4
Część 4 (nieobowiązkowa): . . . . .	4

## Laboratorium 10.

### Magistrala 1-Wire.

Ćwiczenie ma na celu zapoznanie użytkownika z interfejsem 1-Wire oraz jego obsługą w mikrokontrolerze ATmega32. Za pomocą interfejsu 1-Wire można m.in. sterować skalonym termometrem **DALLAS DS18B20** pozwalającym na pomiar temperatury w zakresie od -55°C do 125°C.

#### Część 1:

- Zapoznać się z notą katalogową układu termometru cyfrowego **DS18B20**.
- Podłączyć układ termometru **DS18B20** (pod układem **EEPROM** pierwszy pin od lewej w złączu goldpin, podpisany pionowo DS18B20) do linii **PB3** mikrokontrolera **ATmega32**.
- Przygotować bibliotekę obsługi interfejsu **1-Wire** (np. pliki i1wire.h i i1wire.c). Powinna ona zawierać definicje wykorzystywanego portu i pinu:

```
#define OW_PIN PB3
#define OW_DIR DDRB
#define OW_OUT PORTB
#define OW_IN PINB
```

oraz funkcje do resetu, wysyłania i odbioru na magistrali 1-Wire:

```
uint8_t OW_reset(void);
void    OW_send(uint8_t byte);
uint8_t OW_recv(void);
```

Funkcja OW\_reset() powinna zwracać stan obecności urządzenia 1-Wire na linii (0 – obecny, 1 – nie-obecny, błąd).

Przydatne będą makra realizujące zmianę stanu linii kontrolującej 1-Wire:

```
#define OW_low  OW_DIR |= 1 << OW_PIN
#define OW_high OW_DIR &= ~( 1 << OW_PIN )
```

Oraz makro sprawdzające stan linii:

```
#define OW_check ( OW_IN & ( 1 << OW_PIN ) )
```

Magistrala 1-Wire wymaga zastosowania rezystora podciągającego napięcie do +5 V. Jest on już umieszczony na płycie uruchomieniowej EvB 5.1. Stan wysoki na linii wygodnie jest więc uzyskać wprowadzając linię portu w stan wejściowy, co objawia się stanem wysokiej impedancji ze strony mikrokontrolera

i dzięki rezystorowi stanem wysokim na linii. W tym momencie mikrokontroler przygotowany jest także do odbioru odpowiedzi od układu SLAVE.

Funkcje obsługi magistrali 1-Wire przygotować korzystając z noty katalogowej opisującej protokół 1-Wire. Istotne są tutaj zależności czasowe impulsów. Zwróć uwagę, że dane po 1-Wire wysyłane są w kolejności bitów: najpierw najmniej znaczące (najmłodsze bity). Oznacza to, że przy transmisji i odbiorze przydatne będzie wykorzystanie przesunięcia bitowego w prawo. Np. w trakcie odbioru wygodnie przesuwać ją za każdym odbieranym bitem w prawo i ustawiać zawsze jej najstarszy bit (operacja: | **0x80** ).

### Część 2:

- Przygotować bibliotekę obsługi termometru **DS18B20** (np. pliki ds18b20.h i ds18b20.c), korzystając z jego noty katalogowej i wykorzystując przygotowaną wcześniej bibliotekę do obsługi interfejsu 1-Wire. W bibliotece obsługi termometru powinny znaleźć się funkcje:

```
uint8_t DS18B20_init(void);  
uint8_t DS18B20_start(void);  
int16_t DS18B20_read(void);
```

- Przygotować program główny, który na wyświetlaczu LCD prezentować będzie aktualną temperaturę otoczenia. Wyświetlana wartość temperatury powinna zawierać wartości ułamkowe. Można to prosto uzyskać bez korzystania z liczb zmiennoprzecinkowych (double).

### Część 3:

Rozwinąć program o obsługę interfejsu **RS232**. Mikrokontroler powinien na przychodzącą komendę odczytu temperatury zwracać jej wartość do komputera.

### Część 4 (nieobowiązkowa):

Z wykorzystaniem NI LabVIEW przygotować aplikację mierzącą temperaturę i prezentującą ją za pomocą wskaźnika temperatury oraz wykresu temperatury w zależności od czasu. Wykorzystać przygotowaną w części 3 obsługę interfejsu **RS232** i skorzystać z narzędzi VISA Serial pod NI LabVIEW.