

## Podstawy systemów mikroprocesorowych

### Wykład nr 5 Interfejsy szeregowo c.d.

dr Piotr Fronczak

<http://www.if.pw.edu.pl/~agatka/psm.html>

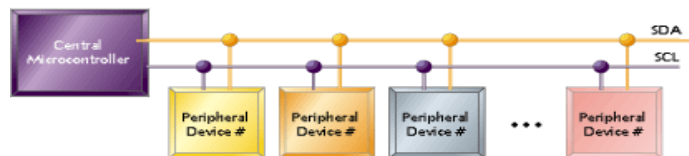
[fronczak@if.pw.edu.pl](mailto:fronczak@if.pw.edu.pl)

Pokój 6GF

## Magistrala I<sup>2</sup>C

- „Inter-integrated circuit” bus
- TWI „Two-wire Serial Interface” w proc. firmy Atmel
- EEPROMy, układy zegarkowe RTC, przetworniki ADC, czujniki termiczne, itd.
- cyfrowe przetwarzanie sygnałów w dekodernach video i procesorach audio.
- dodatkowe układy korzystające z magistrali mogą być dodawane lub wyłączane bez ingerencji w pozostały układ połączeń na magistrali
- całkowicie zintegrowany protokół eliminuje potrzebę stosowania dodatkowych układów, na przykład dekodujących adresy na magistrali
- połączenia na magistrali składają się z tylko dwóch przewodów, ograniczając w ten sposób liczbę ogólnych połączeń w urządzeniu.

## Magistrala I<sup>2</sup>C



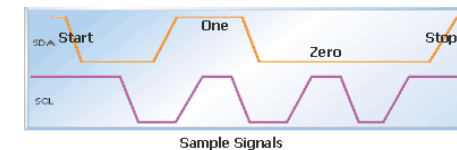
2 dwukierunkowe linie: dane (Serial **D**Ata - SDA), sygnał zegara (Serial **C**Lock - SCL)

Podczas transmisji danych, sygnał na linii SDA musi być stabilny (nie może się zmieniać), gdy linia SCL jest w stanie H. Zmiany linii SDA podczas stanu H linii SCL są interpretowane jako sygnały sterujące (control).

## Magistrala I<sup>2</sup>C

Daje to następujące warunki na magistrali:

- magistrala nie zajęta (Bus Not Busy) - linie SDA i SCL w stanie H,
- start transmisji danych - zmiana stanu linii SDA z H na L, dokonana podczas wysokiego stanu linii SCL,
- stop (zatrzymanie) transmisji danych - zmiana stanu na linii SDA z L na H dokonana podczas wysokiego stanu linii SCL,
- dane ważne (Data Valid) - po wystąpieniu warunku startu stan linii SDA pozostaje stabilny przy stanie H linii SCL. Zmiana danych na linii SDA może nastąpić podczas stanu L linii SCL.



## Magistrala I<sup>2</sup>C

- Każdą transmisję rozpoczyna znacznik startu a kończy znacznik stopu
- Ilość transmitowanych bitów pomiędzy znacznikami nie jest limitowana
- każda porcja informacji wysyłana linią SDA ma długość 8 bitów (bajt)
- dane wysyłane są od najstarszego bitu do najmłodszego
- każdy wysłany bajt musi być potwierdzony przez odbiornik
- urządzenie master zobowiązane jest do generowania dodatkowego taktu zegarowego dla uchwycenia bitu potwierdzenia
- w czasie potwierdzenia nadajnik ustawia swoje wyjście danych na H, co umożliwia odbiornikowi wysłanie potwierdzenia poprzez wymuszenie niskiego stanu na linii SDA

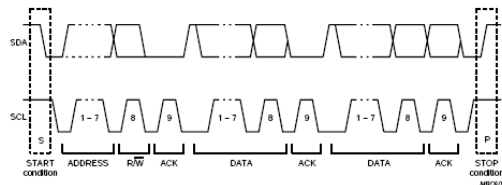
## Magistrala I<sup>2</sup>C

- jeśli odbiornik nie jest w stanie przyjąć kolejnego kompletnego bajtu, to dla poinformowania nadajnika i zwolnienia transmisji ustala stan niski na linii zegara SCL. Wznowienie transmisji może nastąpić natychmiast po tym, jak odbiornik zwolni linię SCL, usuwając z niej stan niski.
- jeśli odbiornik slave nie potwierdził wysłanego do niego adresu własnego (na przykład z powodu wykonywania jakiejś funkcji czasu rzeczywistego) lub któregośkolwiek z przesłanych później bajtów, to linię SDA powinien pozostawić w stanie wysokim, a urządzenie master powinno wysłać warunek stop.

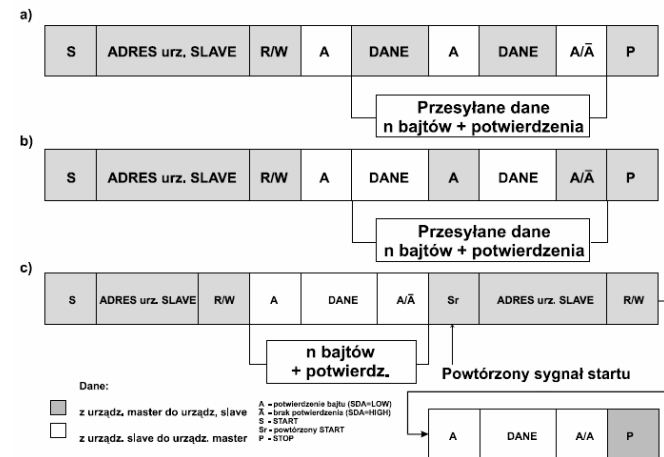
## Format ramki z adresem

Składa się z 9 bitów:

- 7-bitowy sprzętowy adres urządzenia slave,
- bit READ/WRITE:
- 1 oznacza odczyt z urządzenia slave,
- 0 oznacza zapis do urządzenia slave.
- bit potwierdzenia — slave potwierdza odbiór własnego adresu, generując sygnał ACK w dziewiątym cyklu zegara (SCL).



## Tryby transmisji



## Magistrala TWI w mikroprocesorach AVR

Mikrokontrolery ATmega16 i ATmega32 mają wbudowany moduł Two-Wire Serial Interface (TWI), który:

- realizuje komunikację I<sup>2</sup>C,
- zwalnia programistę z konieczności implementacji tej komunikacji na poziomie sygnałów na liniach SCL i SDA.
- Gdy moduł TWI jest włączony, sygnał SCL jest wyprowadzony na pin PC0, a sygnał SDA jest wyprowadzony na pin PC1.
- Można aktywować (rejestr PORTC) wewnętrzne rezystory podciągające na pinach PC0 i PC1 i w pewnych warunkach używać ich zamiast rezystorów zewnętrznych.

## Magistrala TWI w mikroprocesorach AVR

Jeśli mikrokontroler pracuje w trybie slave:

- nie generuje sygnału na SCL,
- częstotliwość zegara systemowego musi być co najmniej 16 razy większa niż częstotliwość sygnałów na linii SCL.

Jeśli mikrokontroler pracuje w trybie master:

- częstotliwość na linii SCL wynosi

$$\frac{\text{clk}}{16 + 2 \cdot \text{TWBR} \cdot 4^{\text{TWPS}}}$$

- clk – częstotliwość zegara systemowego,
- TWBR – wartość rejestru TWBR,
- TWPS – wartości bitów preskalera.

## Rejestr TWCR

7	6	5	4	3	2	1	0	
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR

**TWINT:**

- jest ustawiany sprzętowo po zakończeniu operacji na magistrali I<sup>2</sup>C (ale nie po sygnale STOP!),
- nigdy nie jest automatycznie zerowany,
- jego wyzerowanie (przez wpisanie **jedyńki**) inicjuje kolejną operację na magistrali.

**TWEA:**

ustawienie powoduje automatyczne generowanie sygnału ACK, gdy:

- urządzenie odczyta własny adres w trybie slave,
- urządzenie otrzyma bajt danych w trybie odbioru.

## Rejestr TWCR

7	6	5	4	3	2	1	0	
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR

**TWSTA:**

- ustawienie na jeden powoduje przełączenie w tryb master:
- dopóki szyna jest zajęta, mikrokontroler czeka na sygnał STOP
- następnie generuje sygnał START,
- bit musi zostać wyzerowany programowo (w zwykły sposób).

**TWSTO:**

- w trybie master ustawienie na jeden powoduje wygenerowanie sygnału STOP i automatyczne wyzerowanie bitu,
- w trybie slave przełącza SCL i SDA do stanu wysokiej rezystancji.

### Rejestr TWCR

7	6	5	4	3	2	1	0	
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	—	TWIE	TWCR

#### TWWC:

- ustawiany przy próbie zapisu do TWDR, gdy TWINT jest w stanie niskim, zerowany po zapisie do TWDR, gdy TWINT jest w stanie wysokim

#### TWEN:

- ustawienie powoduje uaktywnienie interfejsu I<sup>2</sup>C i odłączenie wyprowadzeń PC0, PC1 od portu C

#### TWIE:

- włącza przerwanie. Przerwanie jest aktywne tak długo, jak długo TWINT jest ustawiony

### Rejestr TWSR

7	6	5	4	3	2	1	0	
TWS7	TWS6	TWS5	TWS4	TWS3	—	TWPS1	TWPS0	TWSR

- Bit 7 do 3 — kod błędu, zależny od trybu pracy, szczegóły w dokumentacji
- Bit 1, 0 — bity preskalera.

### Rejestr TWAR

7	6	5	4	3	2	1	0	
TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR

- Jeśli mikrokontroler pracuje w trybie slave, zawiera adres mikrokontrolera.
- Adresy postaci 0 i 1111xxx są zarezerwowane i nie należy ich używać.
- Bit 0 jest ustawiany, gdy mikrokontroler ma reagować na tzw. General Call, czyli adres 0.

### Rejestr TWDR

- Two-Wire Data Register
- W trybie nadawania zawiera następne dane (lub adres) do wysłania.
- W trybie odbioru zawiera ostatnio odebrane dane (lub adres).
- Można do niego pisać, gdy TWINT zostanie sprzętowo ustawiony na jedynkę.

#### Przykładowe procedury

```
// procedura transmisji sygnału START
void twistart(void)
{
    TWCR = (1<<TWINT)|(1<<TWSTA)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT)));
}

// procedura transmisji sygnału STOP
void twistop(void)
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);
    while ((TWCR & (1<<TWSTO)));
}
```

### Przykładowe procedury

```
// procedura transmisji bajtu danych
void twiwrite(uint8_t bajt)
{
    TWDR = bajt;
    TWCR = (1<<TWINT) | (1<<TWEN);
    while (!(TWCR & (1<<TWINT)));
}

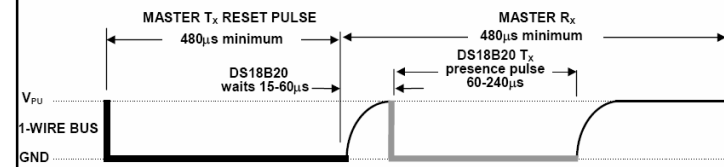
//procedura odczytu bajtu danych
uint8_t twiread(uint8_t ack)
{
    TWCR = (1 << TWINT) | (1 << TWEN) | (ack << TWEA));
    while (!(TWCR & (1<<TWINT)));
    return TWDR;
}
```

## Magistrala 1Wire

- Do całkowitej komunikacji używana jest tylko jedna linia danych. Dodatkowo, odbiornik może być zasilany bezpośrednio z linii danych (zasilanie pasytywne). Kondensator ładowany bezpośrednio z linii danych używany jest do zasilania odbiornika.
- Stosunkowo niewielka przepustowość transmisji danych - standardowo 16 kbps.
- 1-Wire jest podobne do interfejsu I<sup>2</sup>C, ale z uwagi na pojedynczą linię komunikacyjną jest zarówno wolniejsze, jak i tańsze.
- Termometry cyfrowe, instrumenty metrologiczne, sterowniki ładowania akumulatorów, zamki elektroniczne, itd.
- jeden układ nadrzędny (master) i wiele urządzeń podrzędnych (slave)

## Inicjalizacja

- Każda transmisja rozpoczyna się sekwencją inicjalizacyjną zawierającą impuls resetu RESET PULSE wysyłany z Mastera, po którym następuje impuls „przedstawienia się” PRESENCE PULSE wysyłany przez Slave.
- Master wysyła reset przez ustawienie linii danych w stan niski przez minimum 480µs. Po tym zabiegu Master zwalnia magistralę i przechodzi w tryb odbiornika.
- Kiedy magistrala jest zwolniona, rezystor podciąga linię 1-Wire do stanu wysokiego.
- Kiedy Slave wykryje narastające zbocze, odczeka 15 - 60µs, a następnie wyśle impuls PRESENCE PULSE, poprzez ustawienie magistrali w stanie niskim przez 60 - 240µs.



## Komendy

Każde urządzenie 1-Wire posiada unikalny 64 bitowy numer identyfikacyjny zapisany w ROMie:

- 8 bitowy kod typu układu
- 48 bitowy numer seryjny
- 8 bitowa suma kontrolna obliczona na podstawie 56 pierwszych bitów.

W standardzie zestaw komend ROM:

Command	Code	Usage
READ ROM	33H	Identification
SKIP ROM	CCH	Skip addressing
MATCH ROM	55H	Address specific device
SEARCH ROM	F0H	Obtain IDs of all devices on the bus

## Komendy

**READ ROM** – przydatne, gdy jeden układ Slave – odczytuje jego 64 bitowy identyfikator.

**SKIP ROM** – gdy jeden układ Slave, wystarczy by go zaadresować, gdy wiele układów Slave adresuje wszystkie układy (np. by rozpocząć pomiar na kilku sensorach jednocześnie).

**MATCH ROM** – adresuje konkretny układ Slave. Po komendzie wysyłany jest na magistralę 64 bitowy identyfikator.

**SEARCH ROM** – wyszukuje identyfikatory wszystkich układów na magistrali – skomplikowany proces (dokumentacja).

Oprócz uniwersalnych komend ROM, komendy specyficzne dla danego układu.

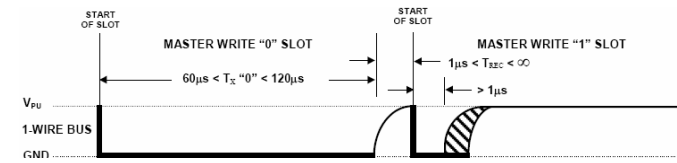
Przykład: komendy sterujące termometrem cyfrowym DS18B20

Table 3. DS18B20 Function Command Set

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED
<b>TEMPERATURE CONVERSION COMMANDS</b>			
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).
<b>MEMORY COMMANDS</b>			
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T <sub>H</sub> , T <sub>L</sub> , and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.
Copy Scratchpad	Copies T <sub>H</sub> , T <sub>L</sub> , and configuration register data from the scratchpad to EEPROM.	48h	None
Recall E <sup>2</sup>	Recalls T <sub>H</sub> , T <sub>L</sub> , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.

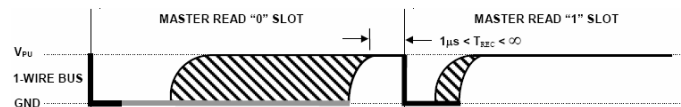
## Zapis bitu

- Aby zapisać 1 do układu slave, po ustawieniu magistrali w stan niski, Master powinien zwolnić magistralę w przeciągu 15µs. Kiedy magistrala jest zwolniona, rezystor podciąga linię danych do stanu wysokiego.
- Aby zapisać 0, po ustawieniu magistrali w stan niski, Master powinien kontynuować trzymanie magistrali w stanie niskim, przez okres najmniej 60µs.

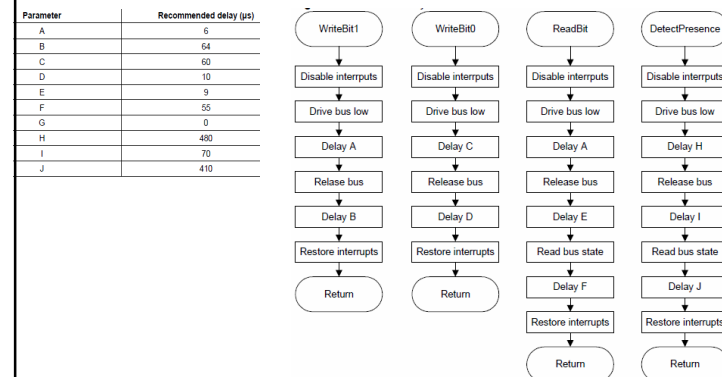


## Odczyt bitu

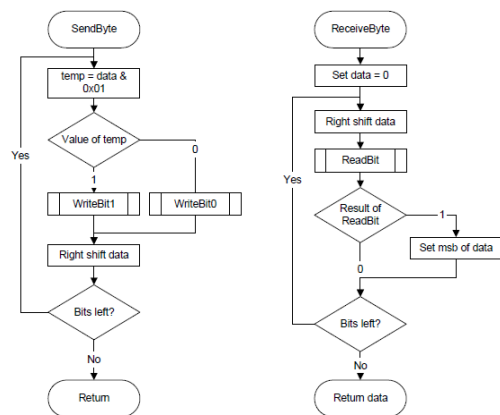
- Inicjowane przez układ Master, przez ustawienie magistrali 1-Wire w stan niski przez minimum 1µs a następnie zwolnienie magistrali.
- Po inicjacji układ Slave zaczyna generować dane na magistralę (zera lub jedynki). Zera transmitowane są przez ustawienie magistrali w stanie niskim, zaś jedynki poprzez podciągnięcie jej do stanu wysokiego.



## Przykładowe procedury



### Przykładowe procedury



### Przykładowe procedury

