



## Podstawy Techniki Mikroprocesorowej Laboratorium

### Ćwiczenie 5 Magistrala SPI

#### Program ćwiczenia:

- konfiguracja transmisji danych między mikrokontrolerem a cyfrowym czujnikiem oraz sterownikiem wyświetlaczy 7-segmentowych przy użyciu magistrali SPI.

#### Zagadnienia do przygotowania:

- podstawy programowania mikrokontrolerów w języku C,
- obsługa magistrali SPI w mikrokontrolerach AVR,
- przerwania zewnętrzne w mikrokontrolerach AVR – obsługa,
- charakterystyka obsługi czujnika TC77.

#### Literatura:

- [1] Wykład
- [2] Dokumentacja mikrokontrolera ATmega8535, [www.atmel.com](http://www.atmel.com).
- [3] Dokumentacja czujników TC77, <http://www.microchip.com>
- [4] Dokumentacja układów MAX7219, <https://www.maximintegrated.com>
- [5] Mikrokontrolery AVR ATmega w praktyce, R. Baranowski, BTC 2005.
- [6] AVR i ARM7. Programowanie mikrokontrolerów dla każdego, P. Borkowski, Helion 2010.
- [7] Mikrokontrolery AVR Język C. Podstawy programowania, M. Kardaś, Atmel 2011.

## Zawartość instrukcji

1. Wstęp .....	1
2. Magistrala SPI .....	1
2.1. Informacje podstawowe.....	1
2.2. Konfiguracja magistrali .....	3
3. Czujnik temperatury TC77 .....	4
4. Układ scalony MAX7219.....	5
5. Program przykładowy .....	6
6. Przykładowe zadania.....	6

UWAGA! Do poprawnej obsługi mikrokontrolera niezbędne jest korzystanie z jego dokumentacji.

Instrukcja zawiera pewne uproszczenia w stosunku do rzeczywistych możliwości mikrokontrolera oraz magistrali SPI.

### 1. Wstęp

W trakcie ćwiczenia należy nawiązać połączenie między mikrokontrolerem a cyfrowymi czujnikami temperatury (TC77 firmy Microchip [3]) a także sterownikiem wyświetlaczy 7-segmentowych i diod LED (MAX7219 firmy Maxim Integrated [4]). Komunikacja musi się odbywać z wykorzystaniem magistrali SPI, ponieważ tylko w taki sposób wymienione układy cyfrowe przesyłają dane. Przystępując do ćwiczenia należy znać podstawy działania magistrali SPI, specyfikę tego interfejsu w mikrokontrolerach AVR oraz wymagania stosowanych czujników i sterownika co do sposobu przesyłu komend/wyników pomiarów.

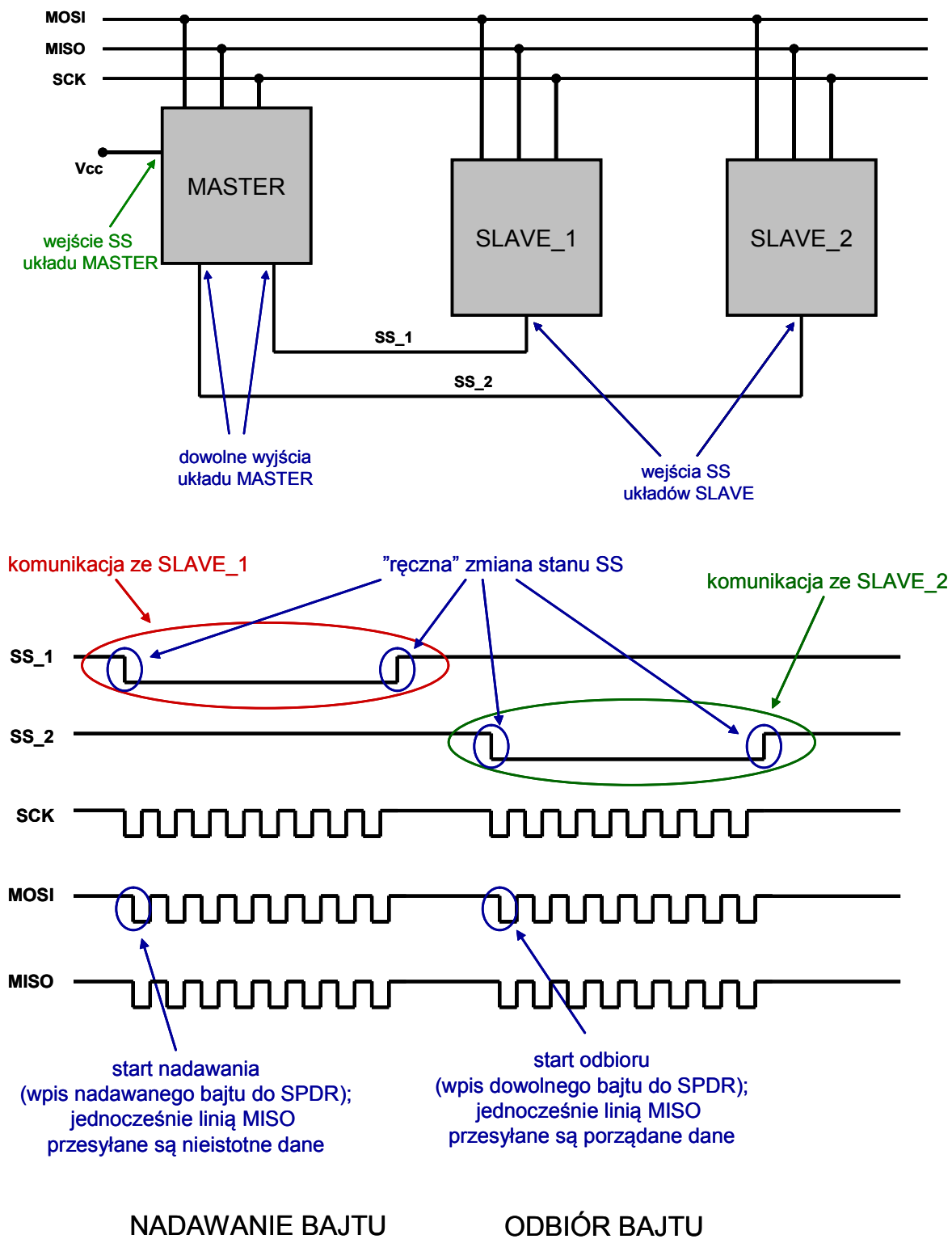
Zadania będą polegały na wysyłaniu żądania pomiaru do wybranego czujnika i na prezentacji otrzymanych odpowiedzi cyfrowych (w tym na wyświetlaczach 7-segmentowych).

### 2. Magistrala SPI

#### 2.1. Informacje podstawowe

SPI (Serial Peripheral Interface) to magistrala szeregową. Komunikacja odbywa się zawsze między dwoma układami – MASTER (nadrzędny) oraz SLAVE (podrzędny), za pośrednictwem dwóch linii danych – MOSI, MISO, linii zegarowej SCK oraz linii wyboru układu podrzędnego (SS). Linia MOSI (Master Output Slave Input) dane są wysyłane przez układ MASTER, a jednocześnie odbierane przez SLAVE. Z odwrotnym kierunkiem przepływu mamy do czynienia w linii MISO (Master Input Slave Output). Przy pomocy linii SS MASTER informuje układ SLAVE, że chce się z nim skomunikować. Dla każdego układu podrzędnego należy zapewnić indywidualną linię SS, tak więc w sieci zawierającej  $n$  układów SLAVE będzie  $n$  linii SS. Stan bezczynności na linii to logiczne '1'. Aby rozpocząć komunikację MASTER musi zmienić stan linii SS wybranego układu SLAVE na

'0'. Tym wyprowadzeniem steruje się "ręcznie" (nie jest modyfikowane automatycznie przez magistralę), poprzez wpisywanie '0' lub '1' w odpowiednich momentach. Po zakończeniu transmisji przywracane jest logiczne '1', czyli stan bezczynności. Magistrala jest znowu wolna, można rozpocząć kolejny cykl komunikacji. Na rys. 1 przedstawiono przykładową sieć SPI oraz przykładowy schemat transmisji dla niej (MASTER nadaje 1 bajt do SLAVE\_1, następnie odbiera 1 bajt od SLAVE\_2).



Rys. 1. Przykładowa sieć SPI oraz przykładowy schemat transmisji dla niej (MASTER nadaje 1 bajt do SLAVE\_1, następnie odbiera 1 bajt od SLAVE\_2).

Inne cechy magistrali:

- sygnał zegarowy generuje MASTER, układy SLAVE z niego korzystają (linia SCK);
- sygnał zegarowy jest generowany tylko podczas transmisji;
- *start* transmisji to wpisanie danych do rejestru SPDR;
- transmisję zawsze inicjuje MASTER, niezależnie czy nadaje, czy odczytuje dane;
- SLAVE nie może zainicjować transmisji, w każdym przypadku musi czekać aż zrobi to MASTER;
- magistrala SPI pracuje w trybie *full duplex*, tzn. dane są zawsze jednocześnie nadawane i odbierane;
- jeśli MASTER chce odebrać dane od SLAVE musi coś do niego wysłać (tak działa *full duplex*). Np. jeśli chcemy odczytać 1-bajtową odpowiedź czujnika, musimy wysłać do niego 1-bajt dowolnych danych.

Z punktu widzenia mikrokontrolerów AVR nawiązanie komunikacji wymaga:

- ustawienia logicznego '1' na linii SS;
- konfiguracji magistrali;
- zmiany stanu SS na logiczne '0' (informacja dla SLAVE, że będzie transmisja);
- wpisania danych do SPDR (*start* transmisji);
- obserwacji stanu flagi przerwania SPIF;
- zmiany stanu SS na logiczne '1' (informacja dla SLAVE, o zakończeniu transmisji).

## 2.2. Konfiguracja magistrali

Konfiguracja magistrali polega na odpowiednim ustawieniu rejestrów specjalnych SPCR, SPSR, SPDR.

### Rejestr kontrolny SPCR (tab. 1):

służy do konfiguracji magistrali.

Tab. 1. Bity rejestru SPCR

7	6	5	4	3	2	1	0	nr bitu
SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	<b>SPCR</b>
0	0	0	0	0	0	0	0	stan pocz.

SPIE – aktywacja przerwania magistrali;

SPE – aktywacja magistrali;

DORD – kolejność przesyłu danych – od najstarszego (MSB) do najmłodszego (LSB) gdy '0';

MSTR – tryb pracy – mikrokontroler jako MASTER gdy '1';

CPOL, CPHA – opcje, w trakcie laboratorium wpisać "00";

SPR1, SPR0 – prescaler definiujący częstotliwość transmisji, można wpisać "00".

### Rejestr stanu SPSR (tab. 2):

zawiera dwie flagi informacyjne oraz trzeci bit prescalera.

Tab. 2. Bity rejestru SPCR

7	6	5	4	3	2	1	0	nr bitu
SPIF	WCOL	-	-	-	-	-	SPI2X	<b>SPSR</b>

0 0 0 0 0 0 0 0 stan pocz.

SPIF – flaga przerwanie, informuje o zakończeniu transmisji, bardzo istotna;

WCOL – flaga informująca o kolizji danych (wpis nowych danych do SPDR przed wysłaniem poprzednich);

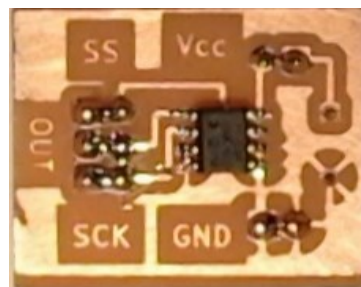
SPI2X – trzeci bit prescalera (zob. SPR1, SPR0).

### Rejestr danych SPDR:

wpisujemy tu dane do wysłania oraz odczytujemy otrzymane.

## 3. Czujnik temperatury TC77

Czujnik TC77 (rys.2) mierzy temperaturę. Wyniki podaje w postaci 13-bitowej liczby w kodzie U2. Wykorzystuje protokół SPI.



Rys. 2. Czujnik TC77 oraz moduł laboratoryjny z czujnikiem.

Czujnik posiada 4 wyprowadzenia, które należy podłączyć do odpowiadających im nóżek mikrokontrolera:

- V<sub>CC</sub> – zasilanie, 5 V (na module oznaczone kolorem zielonym!);
- GND – uziemienie;
- SCK – linia zegarowa magistrali SPI;
- SI/O – linia danych magistrali SPI (MISO).
- SS – wejście *Slave Select* czujnika.

Czujnik przetwarza temperaturę w sposób ciągły, aktualizując się co około 300 ms. Odczytać ją można w dowolnym momencie. Komunikacja polega na wysłaniu 2 dowolnych bajtów danych do czujnika, jednocześnie (*full duplex*) otrzymamy odczyt. Temperatura zakodowana jest na 13 najstarszych bitach odpowiedzi, zgodnie z tab. 3. Rozdzielczość czujnika to 0,0625°C.

Tab. 3. Odpowiedzi czujnika TC77 [3]

Temperature	Binary MSB / LSB	Hex
+125°C	0011 1110 1000 0111	3E 87h
+25°C	0000 1100 1000 0111	0B 87h
+0.0625°C	0000 0000 0000 1111	00 0Fh
0°C	0000 0000 0000 0111	00 07h
-0.0625°C	1111 1111 1111 1111	FF FFh
-25°C	1111 0011 1000 0111	F3 87h
-55°C	1110 0100 1000 0111	E4 87h

## 4. Układ scalony MAX7219

Układ scalony MAX7219 jest sterownikiem wyświetlaczy 7-segmentowych lub matrycy diod LED. Umożliwia jednoczesną obsługę ośmiu wyświetlaczy lub 64 LED. Do sterowania jego pracą wykorzystuje się magistralę SPI. Układ znajduje się na makiecie laboratoryjnej i jest szczegółowo omówiony w [4].

MAX7219 jest wyposażony w:

- sterownik segmentów i poszczególnych wyświetlaczy jednocyfrowych;
- rejestry sterujące pracą wyświetlacza (5 szt.);
- dekodery BCD (kod B);
- układy multipleksowania wyświetlaczy;
- dwuportową pamięć SRAM 8x8 przechowującą dane do wyświetlenia.

Układ pracuje zawsze w trybie SLAVE i posiada 4 wyprowadzenia dla magistrali SPI:

- CLK – odpowiada SCK;
- DIN – odpowiada MOSI;
- DOUT – odpowiada MISO;
- LOAD – nóżka SS dla MAX7219.

W tab. 4 przedstawiono organizację 16-bitowego rejestru przesuwającego, do którego przesyłane są dane z wyprowadzenia DIN układu. Bity D15-D12 nie są wykorzystywane i ich wartość nie ma znaczenia. Bity D11-D8 określają adresy rejestrów (zob. dalsza część instrukcji), zaś D7-D0 dane, które zostaną wpisane do tych rejestrów.

Tab. 4. Organizacja danych wejściowych dla układu MAX7219

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADRES				MSB	DANE				LSB		

MAX7219 zawiera 5 rejestrów sterujących oraz 8 rejestrów danych (po jednym dla każdego z 8-u wyświetlaczy 7-segmentowych). Korzystając z dokumentacji technicznej MAX7219 należy odpowiednio ustawić bity dla wszystkich rejestrów, począwszy od sterujących. W tab. 5 zebrano podstawowe informacje, szczegóły znajdują się w [4].

Tab. 5. Podstawowe informacje nt. rejestrów wewnętrznych MAX7219

Rejestr	Adres	Możliwe wartości	Funkcja
Display Test	0x0F	0 lub 1	Test diod LED
Shutdown	0x0C	0 lub 1	Włączenie układu
Scan Limit	0x0B	0÷7	Wybór liczby wyświetlaczy (1÷8)
Intensity	0x0A	0÷15	Jasność świecenia LED
Decode Mode	0x09	0, 1, 0x0F lub 0xFF	Sposób definiowania znaków (kod B lub manualnie)
wyświetlacz 0	0x08	0÷15 gdy wybrano kod B lub 0÷255 gdy wybrano ustawianie manualne	Odpowiadają za wyświetlanie konkretnych znaków. Sposób interpretacji wpisywanych danych zależy wyboru dokonanego w Decode Mode
wyświetlacz 1	0x07		
wyświetlacz 2	0x06		
wyświetlacz 3	0x05		
wyświetlacz 4	0x04		
wyświetlacz 5	0x03		
wyświetlacz 6	0x02		
wyświetlacz 7	0x01		

## 5. Program przykładowy

ZADANIE:

„przesłanie 4 bajtów: dwa pierwsze od MASTER do SLAVE dwa kolejne od SLAVE do MASTER”  
(jest to sformułowanie przewrotne – ponieważ FULL DUPLEX...)

```

;PROGRAM PRZYKŁADOWY:
;przesłanie 4 bajtów: dwa pierwsze od MASTER do SLAVE
;dwaj kolejne od SLAVE do MASTER

.include "m8535def.inc"
:[inicjalizacja stosu]

;konfiguracja wyprowadzeń
LDI R16,0b10110000
OUT DDRB,R16      ;konfiguracja wyprowadzeń MASTERa (SCK MOSI,MISO,SS)
SBI PORTB,4       ;zapewnienie '1' na nóżce SS MASTERa

SBI DDRA,0        ;do sterowania nóżką SS SLAVEa
SBI PORTA,0       ;stan bezczynności SPI

;konfiguracja rejestru sterującego
LDI R16,0b01010000
OUT SPCR,R16      ;praca w trybie MASTER, transmisja od MSB

;start transmisji
LDI R17,$0C       ;bajt nr 1 (od MASTER do SLAVE)
LDI R18,$01       ;bajt nr 2 (od MASTER do SLAVE)
LDI R19,0         ;R19 do odebrania bajtu nr 3 (od SLAVE)
LDI R20,0         ;R20 do odebrania bajtu nr 4 (od SLAVE)
RCALL wyslij

nic:
RJMP nic

wyslij:
CBI PORTA,0       ;nóżka SS SLAVEa na '0' - start sesji

OUT SPDR,R17      ;bajt nr 1 (wysyłanie) - automatyczny start transmisji
RCALL czekaj;...aż wysłaje cały bajt z SPDR

OUT SPDR,R18      ;bajt nr 2 (wysyłanie)
RCALL czekaj

OUT SPDR,R19      ;odbiór bajtu nr 3: wysyłamy cokolwiek...
RCALL czekaj
IN R19,SPDR       ;...a SLAVE "odpowiada" bajtem nr 3 (kopiujemy go z SPDR)

OUT SPDR,R20      ;odbiór bajtu nr 4: SLAVE nie może nadawać samodzielnie...
RCALL czekaj
IN R20,SPDR       ;...wysyła dane wtedy i tylko wtedy, gdy nadaje MASTER
;(full duplex)

SBI PORTA,0       ;nóżka SS SLAVEa na '1' - koniec sesji
RET

czekaj:
SBIS SPSR,SPIF    ;testowanie flagi SPIF
RJMP czekaj
RET

```

## 6. Przykładowe zadania

- odczytaj temperaturę zmierzoną przez czujnik;
- przekształć wartość podaną w kodzie U2 na kod dziesiętny;
- wyświetl napis „HI WORLD” na wyświetlaczach 7-segmentowych.