

# Laboratorium Techniki Mikroprocesorowej **Tematy ćwiczeń laboratoryjnych**

Maciej Dzieaniakowski  
Grzegorz Wrona

wersja 1.01

<b>Wprowadzenie</b>	<b>3</b>
<b>Zadania</b>	<b>4</b>
<i><b>BIN <math>\Leftrightarrow</math> BCD (BIN2BCD)</b></i>	<b>4</b>
<i><b>Bit Banging I<sup>2</sup>C</b></i>	<b>4</b>
<i><b>Generator przebiegu SIN</b></i>	<b>5</b>
<i><b>Generator przebiegów liniowych 1</b></i>	<b>6</b>
<i><b>Generator przebiegów liniowych 2</b></i>	<b>7</b>
<i><b>Opóźnianie sygnału analogowego</b></i>	<b>8</b>
<i><b>Generator przebiegu prostokątnego (enkoder)</b></i>	<b>10</b>
<i><b>Linijka świetlna (enkoder)</b></i>	<b>11</b>
<i><b>Filtr FIR</b></i>	<b>13</b>
<i><b>Licznik</b></i>	<b>17</b>
<i><b>Rotacja LED (odpytywanie)</b></i>	<b>18</b>
<i><b>Rotacja LED (przerwania)</b></i>	<b>19</b>
<i><b>PWM</b></i>	<b>20</b>
<i><b>Pomiar czasu reakcji</b></i>	<b>23</b>
<i><b>Regulator histerezowy</b></i>	<b>24</b>
<i><b>Stoper</b></i>	<b>25</b>
<i><b>GPIO 1</b></i>	<b>25</b>
<i><b>GPIO 2</b></i>	<b>25</b>
<i><b>Sterowanie X-Y</b></i>	<b>26</b>
<i><b>Filtr wykładniczy I-rzędu</b></i>	<b>27</b>
<i><b>Obsługa wyświetlacza LCD (HD447802)</b></i>	<b>28</b>
<b>Literatura</b>	<b>30</b>

**Wprowadzenie**

Przedstawione zadania obejmują tylko część przykładowych tematów stawianych przed studentami w Laboratorium Techniki Mikroprocesorowej. Są przeznaczone dla studentów niskich semestrów, mających podstawową wiedzę o technice mikroprocesorowej.

Skrypt omawia wybrane, przykładowe ćwiczenia wykonywane w laboratorium. Ich treść, jak i liczba są na bieżąco modyfikowane. Kolejność wykonywania ćwiczeń oraz zmiana ich treści jest określana przez prowadzących zajęcia.

Opis ćwiczeń zawiera tylko treść zadania do wykonania. Nie zawiera rozwiązań. W niektórych przypadkach przekazywane są informacje pomocnicze np. algorytmy cząstkowe, opisy peryferiów, procedury konfiguracyjne, ...

Zgodnie z Regulaminem treść zadania jest podawana studentom w tygodniu poprzedzającym spotkanie laboratoryjne.

Podstawowym warunkiem zaliczenia ćwiczenia jest napisanie odpowiedniego programu i jego uruchomienie w systemie laboratoryjnym. Pytania i dyskusja w trakcie realizacji ćwiczenia, będą związane ze specyficznymi, zastosowanymi przez studenta algorytmami i rozwiązaniami programistycznymi (np. metoda konwersji kodów, sposób ochrony danych przy przerwaniach, organizacja obsługi przerwań, etc.).

**Zadania****BIN  $\Rightarrow$  BCD (BIN2BCD)**

Napisać procedurę konwertującą liczbę 16-b przedstawioną w kodzie binarnym naturalnym na liczbę w kodzie BCD (postać upakowana). Liczby umieszczone są odpowiednio:

\* liczba wejściowa - BIN  $\Rightarrow$  R15

\* liczba wyjściowa - BCD  $\Rightarrow$  R13 MSB(0h, 0h, 0h, L\_dt)LSB  
R14 MSB(L\_t, L\_s, L\_d, L\_j)LSB  
gdzie: L\_dt - liczba dziesiątek tysięcy  
L\_t - liczba tysięcy  
L\_s - liczba setek  
L\_d - liczba dziesiątek  
L\_j - liczba jedności

Uwagi Procedura ma nosić nazwę BIN2BCD. W nagłówku należy umieścić krótką informację o zastosowanej metodzie konwersji. Należy również stosować komentarze wyjaśniających działania poszczególnych części programu.

Parametry oceny

- \* struktura procedury
- \* prawidłowe działanie
- \* czas wykonania [CM]

**Bit Banging I<sup>2</sup>C**

Należy napisać procedurę emulacyjną (*bit banging*), obsługującą transmisję 8-bitowych danych do ekspandera z komunikacyjnym interfejsem I<sup>2</sup>C.

Uwagi

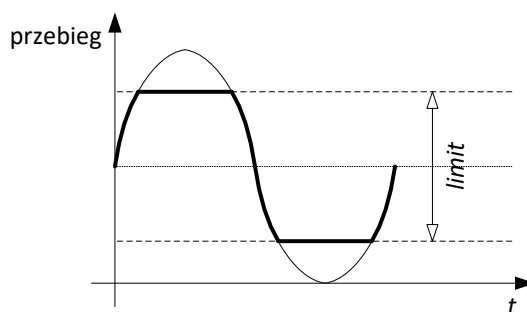
- procedura ma być wywoływana cyklicznie i przysyłać dane z wybranego rejestru  $\mu$ C do ekspandera
- należy skorzystać z szablonu procedury Bit Banging przedstawionego w skrypcie „*Interfejsy komunikacyjne mikrokontrolerów*”
- transmisja jednokierunkowa (od Master'a do Slave'a)
- każda transmisja ma przysyłać tylko jeden bajt danych
- przyporządkowanie linii transmisyjnych magistrali do linii GPIO  $\mu$ C wynika ze schematu ideowego – „*Mikroprocesorowy system laboratoryjny*”

**Generator przebiegu SIN**

Należy napisać i uruchomić program generujący w sposób ciągły przebieg sinusoidalny, na podstawie odczytywanej tablicy zawierającej zdigitalizowany, jeden okres funkcji sinusoidalnej. Tabela zawiera 256 próbek.

- Uwagi
1. należy użyć jednego z dwu przetworników DAC
    - przetwornika wewnętrznego procesora MSP430F169 - DAC1
    - przetwornika zewnętrznego dołączonego do portu P2 procesora – AD7524
  2. częstotliwość sinusoidalnego przebiegu wyjściowego regulowana przy użyciu enkodera
  3. należy wprowadzić ograniczenie (*limit*) wartości sinusoidalnego przebiegu (Rys.1) wyjściowego przy użyciu przycisków (P4). Zmiana wartości ograniczenia nie może wpływać na częstotliwość.
- W każdym wariancie program musi być realizowany w przerwaniach (nie używający zasobów programu głównego)

Rys.1



- Ocennik
- realizacja pkt.1+2 – max. 40 pkt
  - realizacja pkt.1+2+3 – max. 50 pkt

Tablica	sin:	db 128d, 131d, 134d, 137d, 140d, 143d, 146d, 149d
		db 152d, 156d, 159d, 162d, 165d, 168d, 171d, 174d
		db 176d, 179d, 182d, 185d, 188d, 191d, 193d, 196d
		db 199d, 201d, 204d, 206d, 209d, 211d, 213d, 216d
		db 218d, 220d, 222d, 224d, 226d, 228d, 230d, 232d
		db 234d, 235d, 237d, 239d, 240d, 242d, 243d, 244d
		db 246d, 247d, 248d, 249d, 250d, 251d, 251d, 252d
		db 253d, 253d, 254d, 254d, 254d, 255d, 255d, 255d
		db 255d, 255d, 255d, 255d, 254d, 254d, 253d, 253d
		db 252d, 252d, 251d, 250d, 249d, 248d, 247d, 246d
		db 245d, 244d, 242d, 241d, 239d, 238d, 236d, 235d
		db 233d, 231d, 229d, 227d, 225d, 223d, 221d, 219d
		db 217d, 215d, 212d, 210d, 207d, 205d, 202d, 200d
		db 197d, 195d, 192d, 189d, 186d, 184d, 181d, 178d
		db 175d, 172d, 169d, 166d, 163d, 160d, 157d, 154d
		db 151d, 148d, 145d, 142d, 138d, 135d, 132d, 129d
		db 126d, 123d, 120d, 117d, 113d, 110d, 107d, 104d
		db 101d, 98d, 95d, 92d, 89d, 86d, 83d, 80d
		db 77d, 74d, 71d, 69d, 66d, 63d, 60d, 58d
		db 55d, 53d, 50d, 48d, 45d, 43d, 40d, 38d
		db 36d, 34d, 32d, 30d, 28d, 26d, 24d, 22d
		db 20d, 19d, 17d, 16d, 14d, 13d, 11d, 10d
		db 9d, 8d, 7d, 6d, 5d, 4d, 3d, 3d
		db 2d, 2d, 1d, 1d, 0d, 0d, 0d, 0d
		db 0d, 0d, 0d, 1d, 1d, 1d, 2d, 2d
		db 3d, 4d, 4d, 5d, 6d, 7d, 8d, 9d
		db 11d, 12d, 13d, 15d, 16d, 18d, 20d, 21d
		db 23d, 25d, 27d, 29d, 31d, 33d, 35d, 37d
		db 39d, 42d, 44d, 46d, 49d, 51d, 54d, 56d
		db 59d, 62d, 64d, 67d, 70d, 73d, 76d, 79d
		db 81d, 84d, 87d, 90d, 93d, 96d, 99d, 103d
		db 106d, 109d, 112d, 115d, 118d, 121d, 124d, 128d

**Generator przebiegów liniowych 1**

Należy napisać i uruchomić program generujący w sposób ciągły przebieg trójkątny lub piłokształtny opadający lub piłokształtny narastający lub prostokątny.

- Warunki zadania
1. należy użyć jednego z dwu przetworników DAC
    - a) przetwornika wewnętrznego procesora MSP430F169 - DAC\_0 (rozdzielczość 12-b)
    - b) przetwornika zewnętrznego dołączonego do portu P2 procesora – AD7524 (rozdzielczość 8-b)
  2. zmiana generowanego przebiegu przyciskami (P4)
  3. program realizowany wyłącznie w przerwaniach (Timer\_A lub Timer\_A+P1)
  4. częstotliwości generowanych przebiegów po przełączeniu muszą być identyczne

Zalecenia W przerwaniach od Timer\_A należy modyfikować zawartość arbitralnie wybranego rejestru i przesłać tę zawartość do rejestru wejściowego przetwornika DAC.

Inicjalizacja  
przetwornik DAC\_0  
(strona materiałów  
slau049f.pdf)

- rejestr konfiguracyjny DAC12\_OCTL – należy ustawić poniższe wartości:
  - Vref=VREF+
  - 12-bit resolution
  - Load mode 0
  - Full-Scale=1xVref
  - High speed amplifier output
  - Data format - straight binary
  - Interrupt disabled
  - conversion enabled
- rejestr konfiguracyjny ADC12CTL0 – należy ustawić poniższe wartości:
  - Reference generator ON
  - VRef+=2.5V

Inicjalizacja  
przetwornik AD7524  
(strona materiałów  
AD7524.pdf)

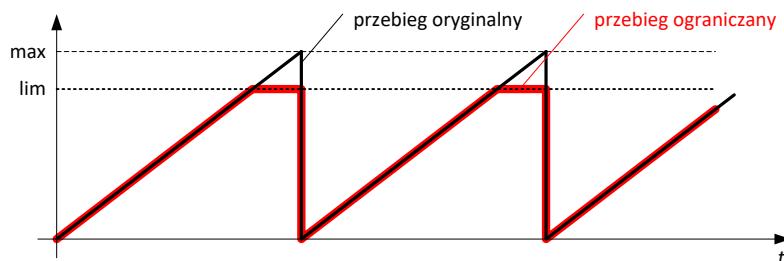
Należy ustawić wejścia sterujące **CS/WR** odpowiednio do trybu pracy przetwornika „DAC responds to data bus (DB0–DB7) inputs”.

## Generator przebiegów liniowych 2

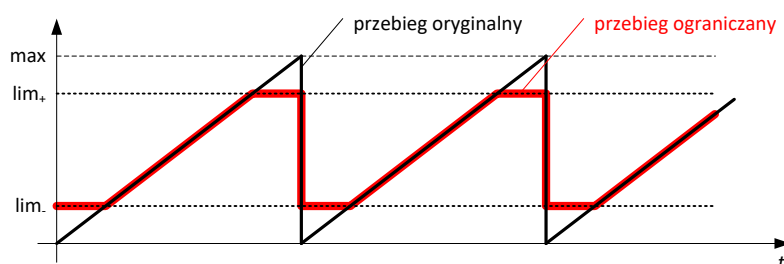
Rozszerzenie (modyfikacja) poprzedniego programu generującego przebiegi piła/trójkąt/prostokąt.

Należy napisać i uruchomić program ograniczający amplitudę generowanego przebiegu w sposób pokazany na Rys.1 **lub** Rys.2

Rys.1



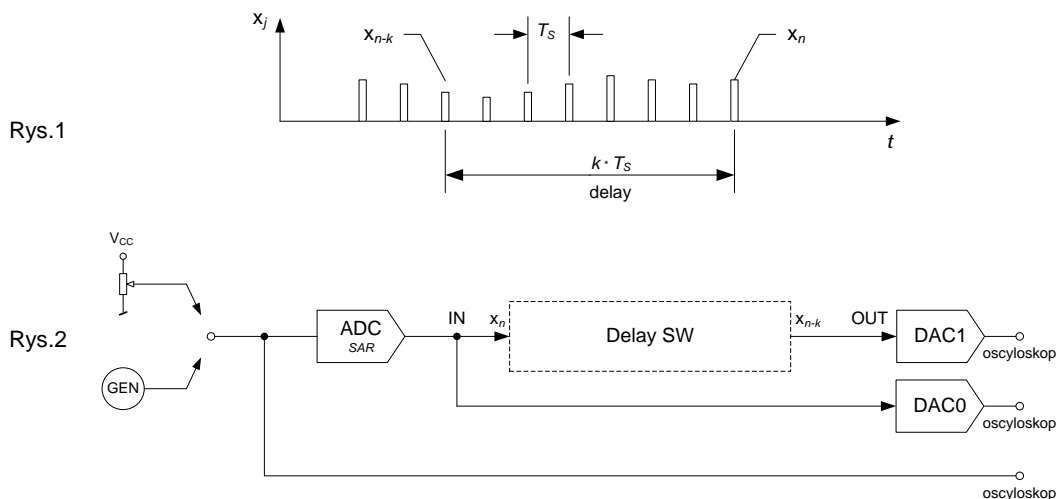
Rys.2



- Warunki zadania
1. do zmiany wartości limitu można używać:
    - enkodera
    - dwu wybranych przycisków (up/down) dołączonych do portu P4
  2. wartość limitu i jego zmiana nie mogą zmieniać częstotliwości przebiegu
  3. program realizowany wyłącznie w przerwaniach (Timer\_A lub Timer\_A+P1)

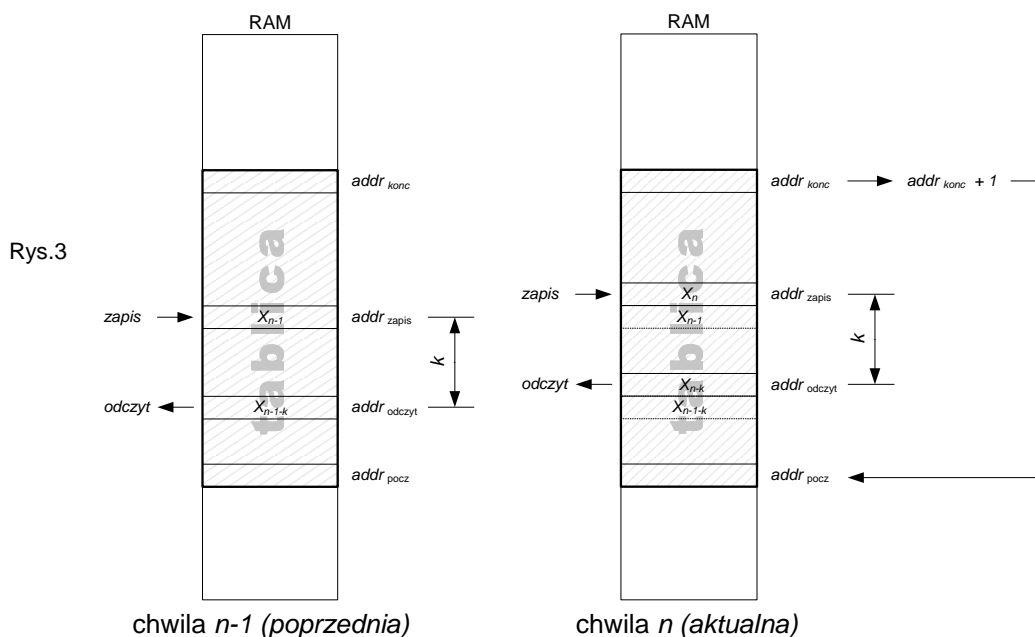
## Opóźnienie sygnału analogowego

Należy napisać program (Delay) opóźniający przebieg analogowy. Poniżej przedstawiono podstawowy schemat blokowy układu (Rys.2) i zasadę opóźniania (Rys.1) digitalizowanego przebiegu analogowego.



Realizację rejestru FILO, składającego pobierane próbki i zapewniającego oczekiwane opóźnienie  $k \cdot T_s$  (gdzie  $T_s$  jest okresem próbkowania) można zrealizować na wiele sposobów. Poniżej dwa:

- konsekwentne przesuwanie zawartości komórek pamięci tworzących rejestr FILO.
- umieszczanie aktualnie pobieranych próbek w tablicy i odpowiedni adresowo odczyt tych, które były umieszczone w tablicy uprzednio. Zasadę tej metody przedstawiono na rysunkach poniżej – adres zapisu aktualnie wprowadzanej próbki i odczytu opóźnionej (wprowadzonej uprzednio) jest w każdym cyklu inkrementowany. To zapewnia stałość  $k$ , a więc stałość czasu opóźnienia.





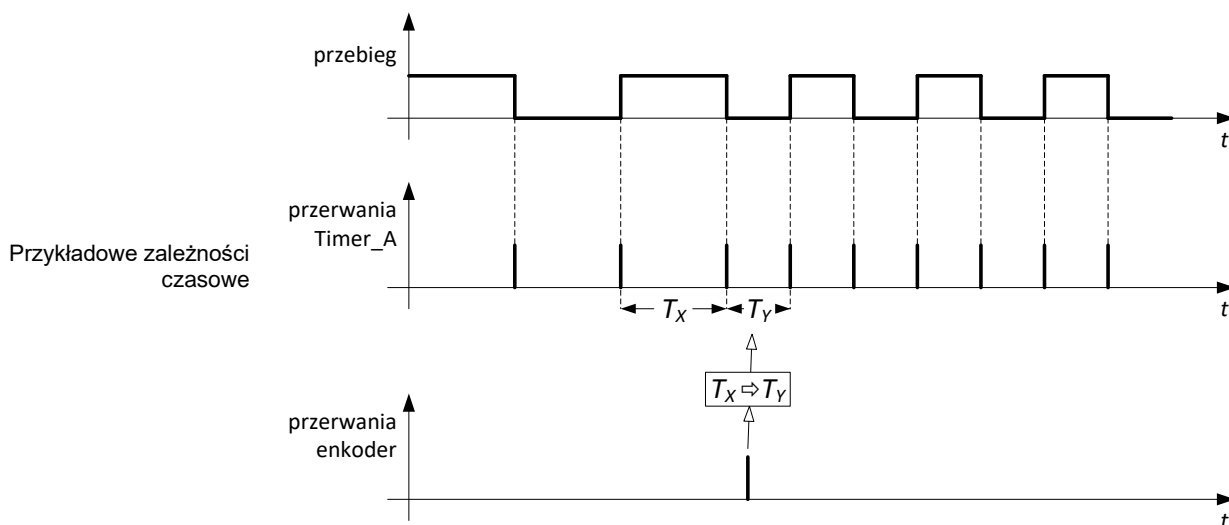
Na rysunku (Rys.3) przedstawiono również sposób ograniczenia wielkości tablicy. Oczywistym warunkiem poprawności takiego ograniczenia jest  $k < (addr_{konc} - addr_{pocz})$ . W systemie laboratoryjnym tablica powinna być umieszczona w przedziale adresowym RAM 0200h-0A00h.

- Uwagi
- zalecane jest stosowanie tablicowania
  - opóźnienie przebiegu w przedziale 0,1-1,0s
  - program realizowany wyłącznie w przerwaniach od Timer\_A (nie używający zasobów programu głównego)
  - wersja podstawowa procedury DELAY – stała wartość czasu opóźnienia
  - zalecana, rozszerzona wersja procedury DELAY - nastawianie czasu opóźnienia (czyli wartości **k**) przyciskami dołączonymi do portu P4

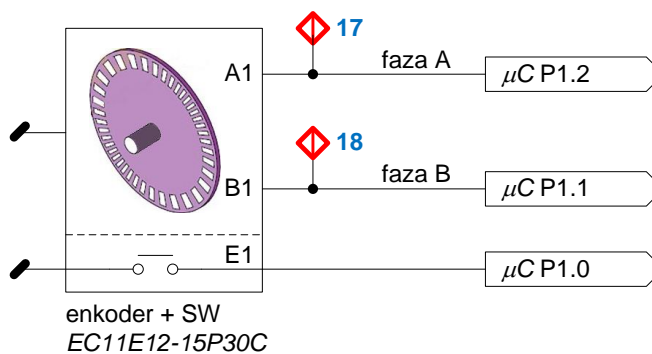
**Generator przebiegu prostokątnego (enkoder)**

Należy napisać i uruchomić program generujący przebieg prostokątny na jednej z wolnych linii wybranego arbitralnie portu (np. P5.0 lub P3.7 lub ...) zgodnie z tabelą *Piny pomiarowe* w dokumencie „Mikroprocesorowy system laboratoryjny”.

- Warunki zadania
- zakres zmian częstotliwości generowanego przebiegu: od 500Hz do 2000Hz
  - częstotliwość przebiegu zależna od nastaw w Timer\_A
  - przebieg tworzony wyłącznie w przerwaniach od Timer\_A
  - okres przerwań od Timer\_A zmieniany enkoderem
  - **program generatora i program główny** są od siebie **niezależne** i nie mają żadnych wspólnych zasobów



Schemat połączeń enkodera



## Linijka świetlna (enkoder)

Należy napisać i uruchomić program zmieniający długość linijki świetlnej (P2)

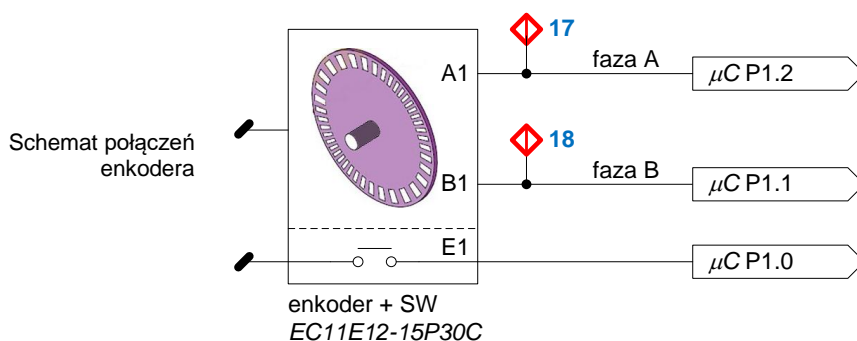
Długość linijki ma być ustalana enkoderem odpowiednio:

- obrót enkoderem w jedną stronę zwiększa długość linijki
- obrót enkoderem w przeciwną stronę zmniejsza długość linijki
- zmiana pozycji enkodera o jedną pozycję zmienia długość linijki o jeden punkt świetlny (LED)
- maksymalna długość linijki – 8 punktów świetlnych
- minimalna długość linijki – 0 punktów świetlnych

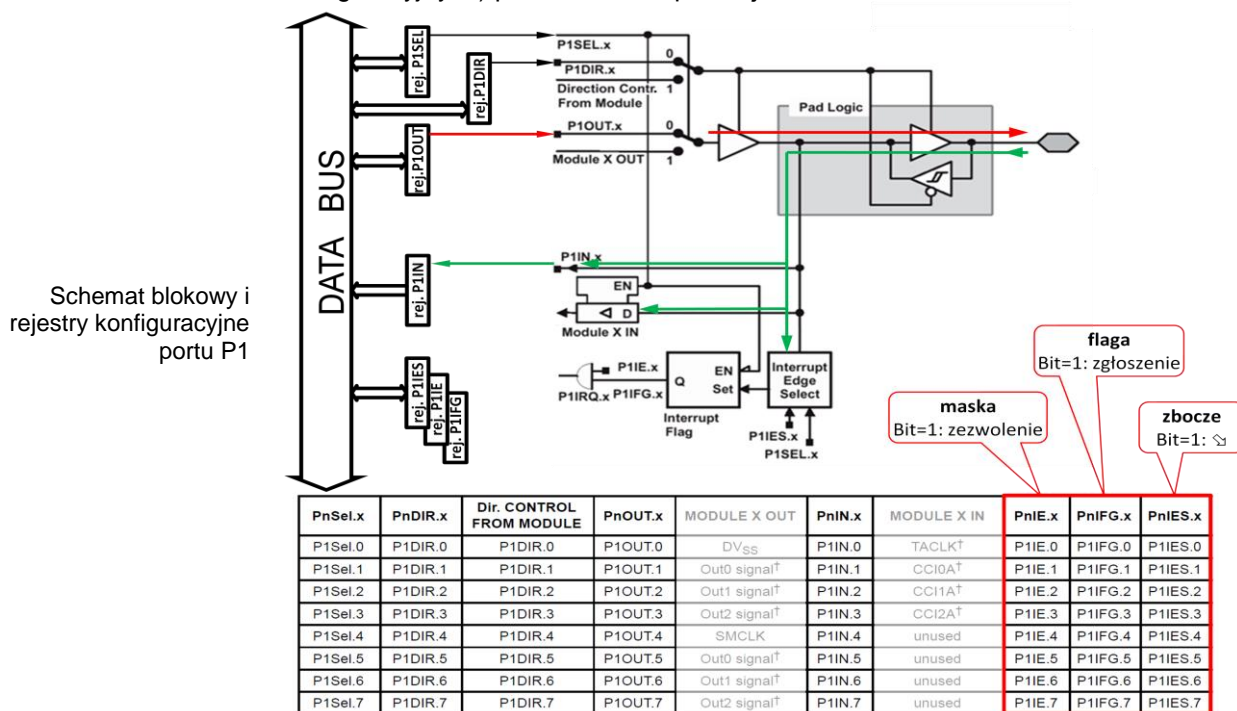
Uwagi **Program ma być procedurą obsługi przerwania od poru P1 i nie posiadać zasobów wspólnych z programem głównym.**

Schemat połączeń przedstawiony jest w dokumentacji „Mikroprocesorowy system laboratoryjny” – enkoder (blok #7), linijka świetlna (P2 blok #5).

Połączenia enkodera są również pokazane poniżej. Wyjścia faz A i B są dołączone odpowiednio do P1.2 i P1.1. Przycisk (E1) może być wykorzystany do innych funkcjonalności.



Zasady wykorzystania portu P1, jako źródła przerwań (ustawienie rejestrów konfiguracyjnych) przedstawiono poniżej.



**Uwaga** ➤ Zapis do PxIES.x może ustawić flagę przerwania PxIFG.x odpowiednio:

PxIESx	PxINx	PxIFGx
0→1	0	may be set
0→1	1	unchanged
1→0	0	unchanged
1→0	1	may be set

Zapis do PxOUT, PxDIR może ustawić flagi w PxIFG.

Wniosek: przed zezwoleniem na przerwanie sprawdzić (lub odpowiednio ustawić) PxIFG.

- Jest tylko jeden wektor przerwania dla P1 (0FFE8h) i odpowiednio dla P2 (FFE2h). Dla stwierdzenia, który pin jest źródłem przerwania należy przetestować rejestr PxIFG.
- Flagi PxIFG nie są kasowane automatycznie. Muszą być kasowane przez program.

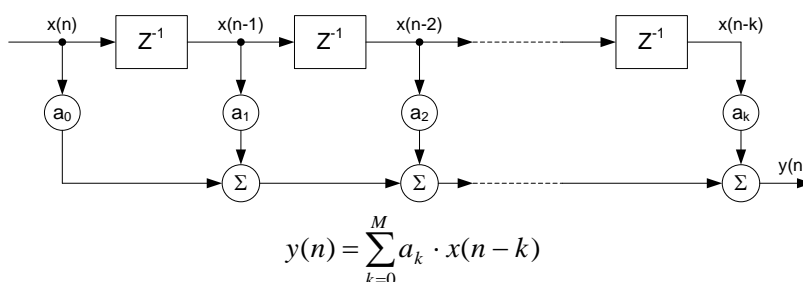
## Filtr FIR

Należy napisać procedurę filtru FIR i zbadać poprawność działania filtru (odpowiedź impulsowa i skokowa).

Filtry typu FIR są jednym z dwu podstawowych typów filtrów cyfrowych i są najczęściej stosowanymi programowymi filtrami przebiegów pomiarowych i sterujących w urządzeniach energoelektronicznych. Ta pozycja aplikacyjna filtrów FIR wynika z kilku czynników:

- układy pomiarów i sterowania oparte są przede wszystkim na technice cyfrowej predestynowanej do „naturalnej” implementacji zależności dyskretnych
- względna łatwość kształtowania cech charakterystycznych filtrów w domenie czasowej i częstotliwościowej
- stałość parametrów filtru
- ograniczenie sprzętowej części analogowej systemu sterującego/pomiarowego

Podstawowy schemat blokowy i równanie opisujące filtr FIR

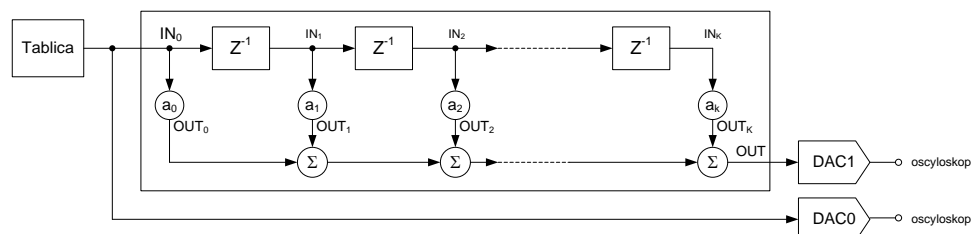


Proponowana literatura – R.G. Lyons „Wprowadzenie do cyfrowego przetwarzania sygnałów”, WKŁ

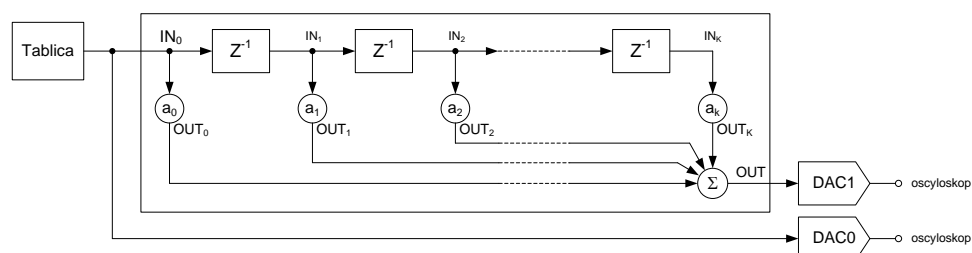
Poniżej dwa schematy realizacji oraz uwagi do nich:

- rząd filtru: min 4, max 8
- sugerowany schemat: ver 2
- współczynniki  $a_0$ - $a_K$  ułamkowe (odwrotności potęgi liczby 2)
- suma współczynników  $a_0$ - $a_K$  równa 1
- tablice 256x8-b zawierają sygnał wejściowy impulsowy/skokowy/sinusoidalny. Można przygotować inne, dodatkowe tablice wejściowe
- sugerowana zmiana tablicy wejściowej przyciskami (P4) - obsługa w przerwaniu Timer'a lub przerwaniu zewnętrznym
- procedura filtru powinna być wołana w każdym przerwaniu od wybranego timera. Z tym samym taktem odczytywane muszą być kolejne wyrazy tablicy i wprowadzane na wejście procedury filtru FIR

Blokowy schemat realizacji ver 1



Blokowy schemat realizacji ver 1



## Wskazówki do procedury FIR

- wartości próbek  $IN_0$ - $IN_k$  oraz wyników obliczeń cząstkowych  $OUT_0$ - $OUT_k$  muszą być umieszczone w oddzielnych komórkach pamięci RAM.
- kolejność działań w procedurze FIR powinna być następująca:
  - przesunięcie wartości próbek  $IN_0$ - $IN_k$  w prawo tzn.  $IN_n$  do  $IN_{n+1}$  począwszy od najstarszej próbki. Na pozycję  $IN_0$  należy wprowadzić próbkę pobraną aktualnie z tablicy. Podana metoda jest efektywna (czas) tylko w przypadku małej ilości próbek.
  - obliczenie wartości cząstkowych  $OUT_0$ - $OUT_k$  oraz ich sumy.
- zgodnie z założeniem współczynniki  $a_0$ - $a_k$  są ułamkowe (odwrotności potęg liczby 2). Oznacza to, że aby otrzymać wartość  $OUT_N$  należy podzielić  $IN_k$  przez odpowiednią potęgę liczby 2. Można to zrealizować używając instrukcji rotacji lub przesunięcia.

[illegible]

Tablica  
skok

db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 255d,255d,255d,255d,255d,255d,255d,255d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d  
db 0d,0d,0d,0d,0d,0d,0d,0d

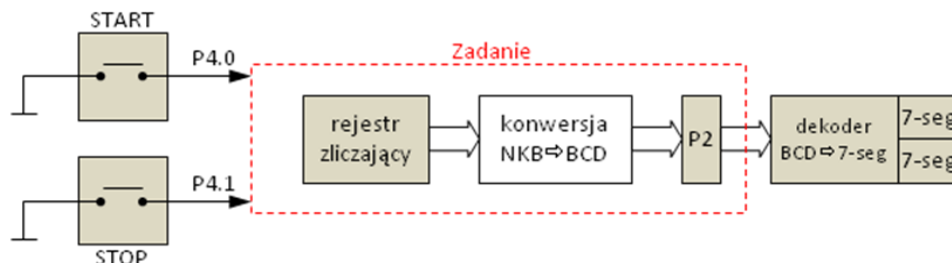
Tablica  
sin16

db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d  
 db 128d,152d,176d,198d,218d,234d,245d,253d  
 db 255d,253d,245d,234d,218d,198d,176d,152d  
 db 128d,103d,079d,057d,037d,021d,010d,002d  
 db 000d,002d,010d,021d,037d,057d,079d,103d



## Licznik

Należy napisać i uruchomić program odmierzający czas z rozdzielczością **1s** i wyświetlający zliczoną wartość na wyświetlaczach 7-mio segmentowych (P2).



Wyświetlacze 7-mio segmentowe pokazują prawidłowe wartości (0, ..., 9) tylko wówczas, gdy dane na porcie P2 przedstawione są w kodzie BCD. To oznacza, że zliczane w rejestrze zliczającym naciśnięcia muszą być przekształcone z naturalnego kodu binarnego na kod BCD i w tej postaci przesłane do portu P2. Ponieważ użyte są dwa wyświetlacze 7-seg, to słowo wysyłane do P2 musi być przedstawione w **upakowanym formacie BCD**.

- wszystkie funkcje programu mają być realizowany **wyłącznie w procedurze obsługi przerwania od Timer A**
- po wyświetleniu maksymalnej liczby '99' ma nastąpić przepełnienie do liczby '00' i od niej musi następować dalsze zliczanie
- przycisk P4.0 (START) inicjalizuje start odmierzania czasu (reakcja na naciśnięcie przycisku)
- przycisk P4.1 (STOP) inicjalizuje zatrzymanie odmierzania czasu (reakcja na naciśnięcie przycisku)

Uwagi Sugestia kolejności rozwiązywania i sprawdzania funkcji programu:

- zliczanie co 1s<sup>\*)</sup> i wysyłanie na P2
- dołożenie bloku konwersji NKB2BCD (BIN2BCD)
- wprowadzenie ograniczenia zliczania i wyświetlania do liczby 99d
- reakcja na przyciski START i STOP

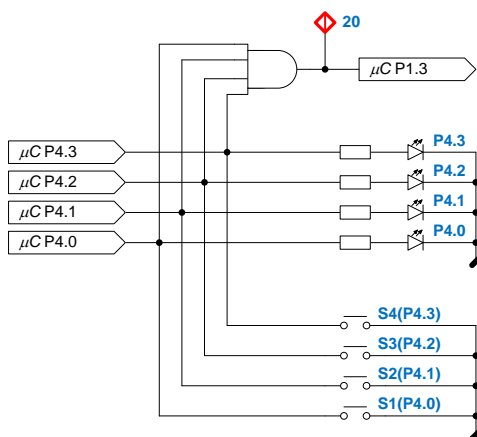
<sup>\*)</sup> zegar  $\mu C$  (DCO) po załączeniu zasilania generuje takt o częstotliwości 800kHz

## Rotacja LED (odpytywanie)

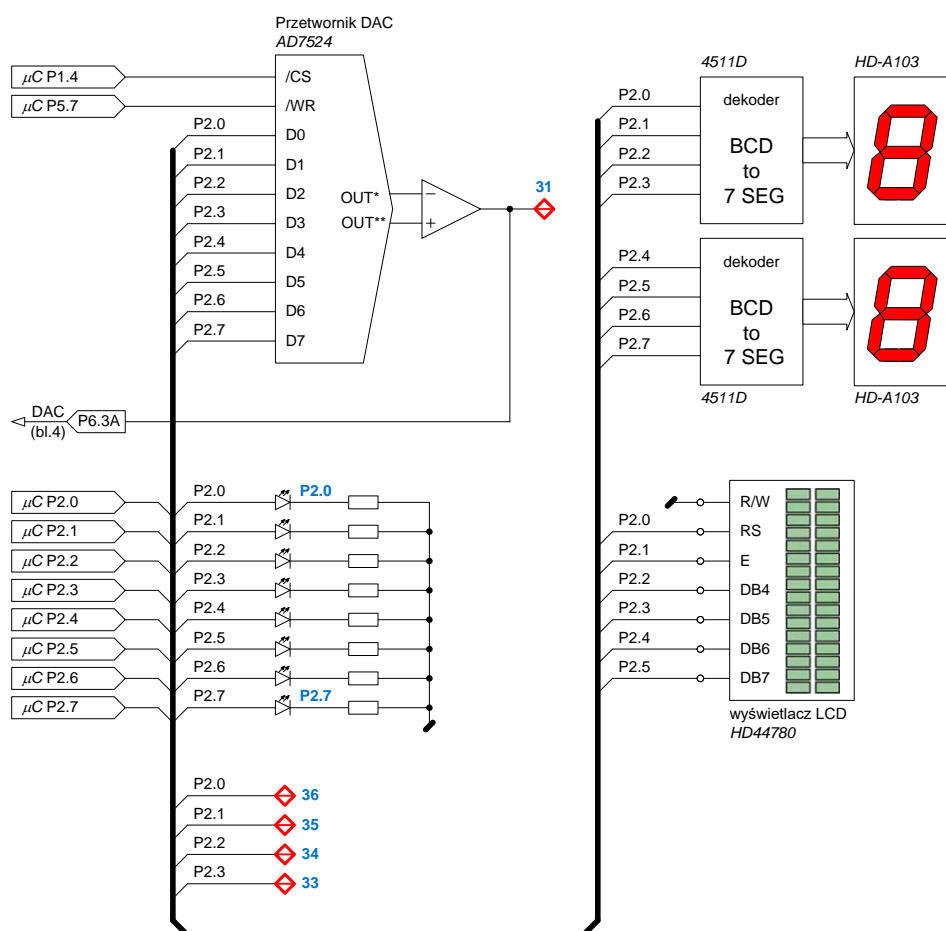
Należy napisać program przesuwający o jedną pozycję punkt świetlny na linii światłowej (P2) przy jednokrotnym **naciśnięciu** przycisku (każde naciśnięcie przycisku przesuwa punkt świetlny o jedną pozycję). Należy wybrać jeden z przycisków dołączonych do portu P4 (P4.0 - P4.3).

Uwagi Program przesuwania ma być programem głównym, zorganizowanym w pętli.  
**Użycie przerw jest niedozwolone.**

**Schemat blokowy połączeń  
linii P4.0-P4.3**  
„Mikroprocesorowy system  
laboratoryjny” (blok #7)



**Schemat blokowy połączeń  
linii P2.0-P2.7**  
„Mikroprocesorowy system  
laboratoryjny” (blok #5)



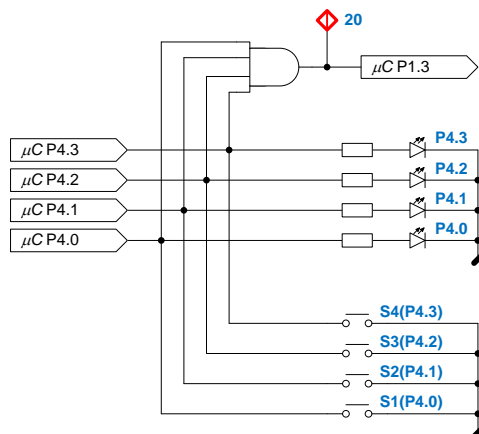
## Rotacja LED (przerwania)

Należy napisać program przesuwający o jedną pozycję punkt świetlny na linijce świetlnej (P2) przy jednokrotnym **naciśnięciu** przycisku (każde naciśnięcie przycisku przesuwa punkt świetlny o jedną pozycję). Należy wybrać jeden z przycisków dołączonych do portu P4 (P4.0 - P4.3).

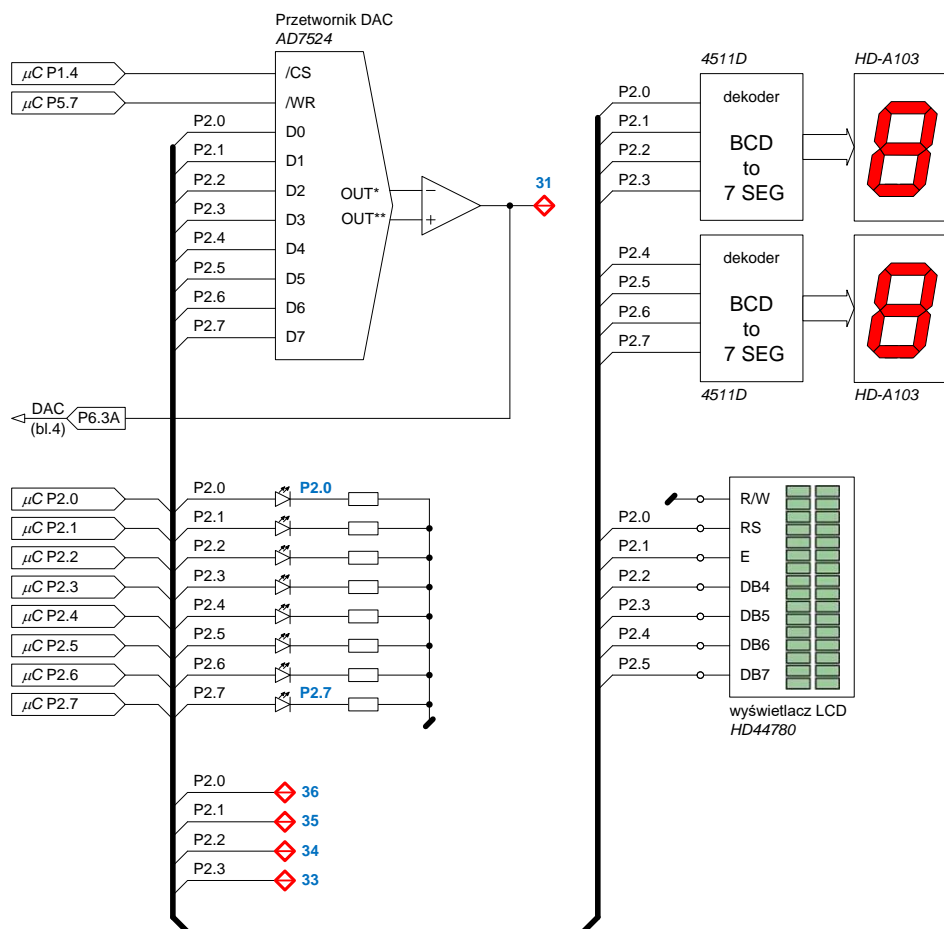
Uwagi **Program przesuwania ma być procedurą obsługi przerwania od portu P1, całkowicie niezależną od programu głównego i niemającą z nim wspólnych zasobów**

Komentarz Zgodnie ze schematem („Mikroprocesorowy system laboratoryjny” blok #7) naciśnięcie dowolnego przycisku P4.0 - P4.3 może generować żądanie przerwania od P1.3.

Schemat blokowy połączeń  
linii P4.0-P4.3  
„Mikroprocesorowy system  
laboratoryjny” (blok #7)

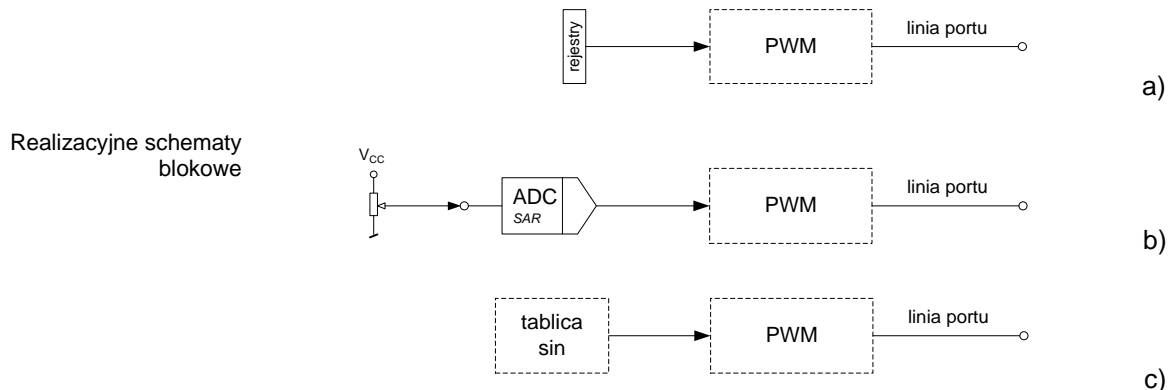


Schemat blokowy połączeń  
linii P2.0-P2.7  
„Mikroprocesorowy system  
laboratoryjny” (blok #5)



**PWM**

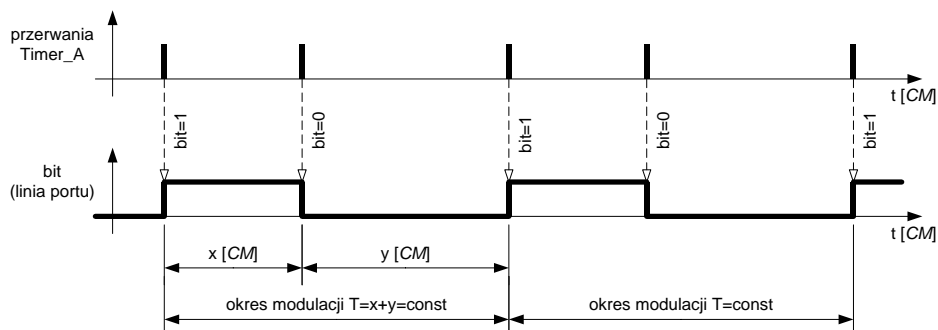
Należy zrealizować układ Modulacji Szerokości Impulsów (PWM). Schematy blokowe do sprawdzenia przyjętych rozwiązań przedstawiono na poniższym rysunku. Realizacja schematu (b) wymaga inicjalizacji przetwornika ADC natomiast w przypadku (c) należy napisać procedurę cyklicznego czytania tablicy zawierającej skwantowany przebieg sinusoidalny (przykładową tablicę i sposób jej obsługi przedstawiono w appendix'e).



Przykładowe rozwiązania przedstawiono w wariantach 1-2. Należy wybrać jeden z wariantów i zrealizować go. Wskazane jest rozpoczęcie od wariantu nr 1. We wszystkich przypadkach sugerowana jest rozdzielczość FFFh.

**Realizacja  
wariant 1**

Modulację zrealizować w oparciu o jeden timer ogólnego przeznaczenia. Zasadę wykorzystania timer'a przedstawiono na poniższym rysunku.

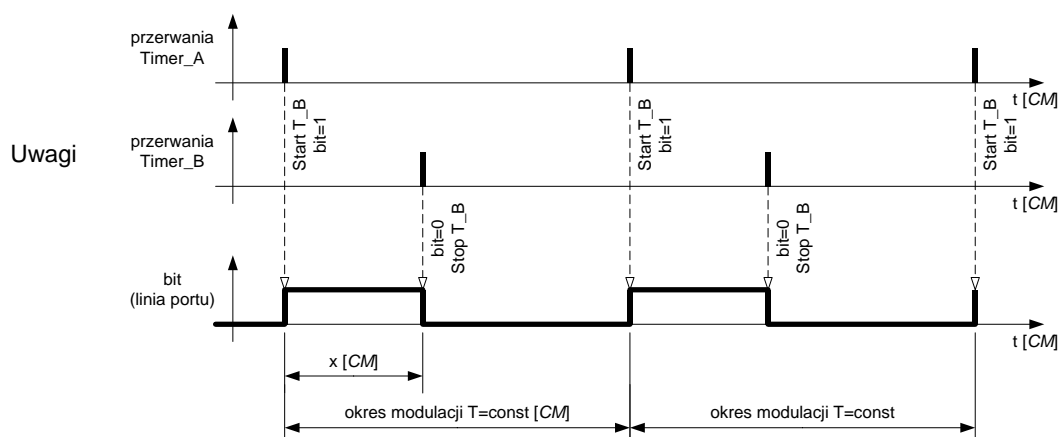


Wskazówki: Należy dokonać inicjalizacji Timer\_A, ustawiając go w jeden z trybów aktywnych (np. up\_mode). Przyjęty i stały okres T przebiegu modulowanego oznacza, że  $x+y=T=const$ . W każdym kolejnym przerwaniu (procedurze obsługi przerwania) należy ustawiać wartość przepełnienia rejestru liczącego timer'a odpowiednio na  $x$  i  $y$ , jednocześnie ustawiając właściwą wartość (0/1) bitu na wybranej linii portu.

Rejestry z danymi wejściowymi ( $T, x$  lub  $T, y$ ) wskazane na rys. a mogą być wybrane dowolnie np. któryś z rejestrów ogólnego przeznaczenia Rx.

Wskazane linie portu wyjściowego – P4.3 lub P4.4 lub P4.6

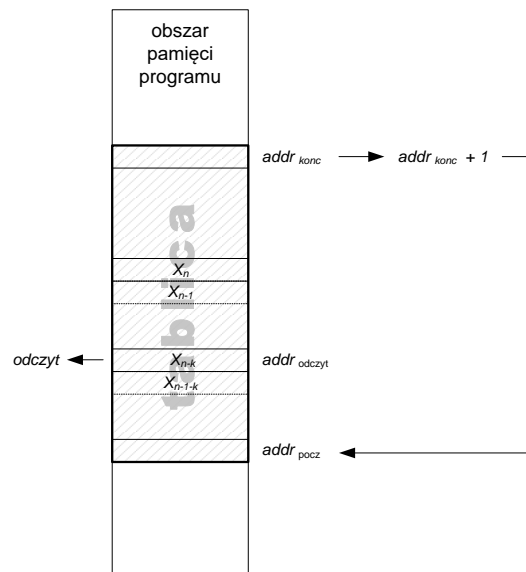
Realizacja variant 2 Modulację zrealizować w oparciu o dwa timer'y ogólnego przeznaczenia. Zasadę wykorzystania timer'ów przedstawiono na poniższym rysunku.



Tablica  
sinus  
(256x8-b)

sin: db 128d, 131d, 134d, 137d, 140d, 143d, 146d, 149d  
db 152d, 156d, 159d, 162d, 165d, 168d, 171d, 174d  
db 176d, 179d, 182d, 185d, 188d, 191d, 193d, 196d  
db 199d, 201d, 204d, 206d, 209d, 211d, 213d, 216d  
db 218d, 220d, 222d, 224d, 226d, 228d, 230d, 232d  
db 234d, 235d, 237d, 239d, 240d, 242d, 243d, 244d  
db 246d, 247d, 248d, 249d, 250d, 251d, 251d, 252d  
db 253d, 253d, 254d, 254d, 254d, 255d, 255d, 255d  
db 255d, 255d, 255d, 255d, 254d, 254d, 253d, 253d  
db 252d, 252d, 251d, 250d, 249d, 248d, 247d, 246d  
db 245d, 244d, 242d, 241d, 239d, 238d, 236d, 235d  
db 233d, 231d, 229d, 227d, 225d, 223d, 221d, 219d  
db 217d, 215d, 212d, 210d, 207d, 205d, 202d, 200d  
db 197d, 195d, 192d, 189d, 186d, 184d, 181d, 178d  
db 175d, 172d, 169d, 166d, 163d, 160d, 157d, 154d  
db 151d, 148d, 145d, 142d, 138d, 135d, 132d, 129d  
db 126d, 123d, 120d, 117d, 113d, 110d, 107d, 104d  
db 101d, 98d, 95d, 92d, 89d, 86d, 83d, 80d  
db 77d, 74d, 71d, 69d, 66d, 63d, 60d, 58d  
db 55d, 53d, 50d, 48d, 45d, 43d, 40d, 38d  
db 36d, 34d, 32d, 30d, 28d, 26d, 24d, 22d  
db 20d, 19d, 17d, 16d, 14d, 13d, 11d, 10d  
db 9d, 8d, 7d, 6d, 5d, 4d, 3d, 3d  
db 2d, 2d, 1d, 1d, 0d, 0d, 0d, 0d  
db 0d, 0d, 0d, 1d, 1d, 1d, 2d, 2d  
db 3d, 4d, 4d, 5d, 6d, 7d, 8d, 9d  
db 11d, 12d, 13d, 15d, 16d, 18d, 20d, 21d  
db 23d, 25d, 27d, 29d, 31d, 33d, 35d, 37d  
db 39d, 42d, 44d, 46d, 49d, 51d, 54d, 56d  
db 59d, 62d, 64d, 67d, 70d, 73d, 76d, 79d  
db 81d, 84d, 87d, 90d, 93d, 96d, 99d, 103d  
db 106d, 109d, 112d, 115d, 118d, 121d, 124d, 128d

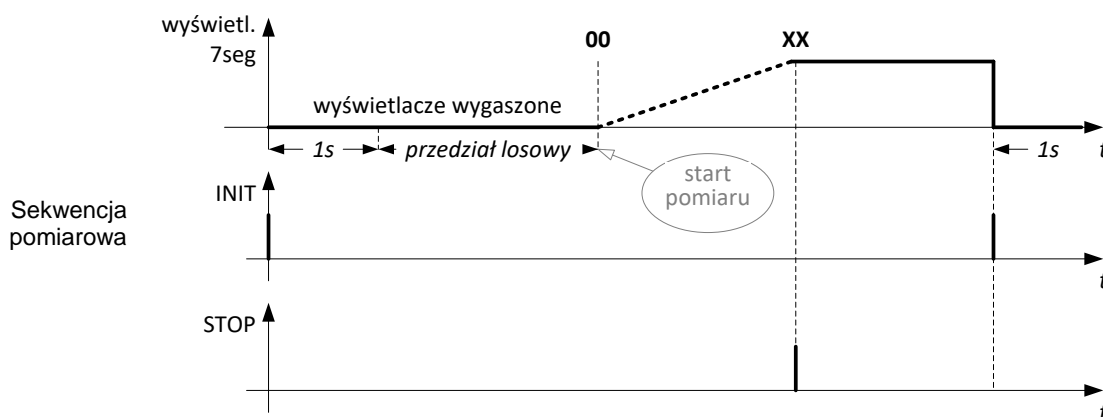
Sposób obsługi tablicy



**Pomiar czasu reakcji**

Należy napisać i uruchomić program mierzący szybkość reakcji. Program ma być zrealizowany z użyciem przerwań i być niezależny od programu głównego.

- Warunki zadania
- należy użyć dwu przycisków
    - P4.0 – (STOP) przycisk naciskany w reakcji na sygnał startu pomiaru
    - P4.1 – (INIT) przycisk rozpoczynający sekwencję pomiarową
  - dokładność pomiaru 0,01s
  - maksymalny czas pomiaru 1s
  - losowy czas oczekiwania na sygnał startu pomiaru (1s-7s) z rozdzielczością 0,01s
  - po naciśnięciu INIT wyświetlacze gasną
  - sygnałem startu pomiaru jest wyświetlenie liczby 00
  - STOP zatrzymuje pomiar
  - wynik pomiaru jest wyświetlany do chwili następnego naciśnięcia INIT

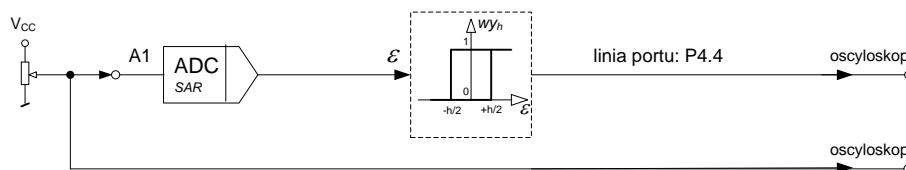


## Regulator histerezy

Jednym z najbardziej znanych, zamkniętych układów regulacji jest układ z regulatorem histerezy. W swojej wersji podstawowej oraz wielu zmodyfikowanych postaciach jest szeroko stosowany począwszy od aplikacji domowych (AGD) do przemysłowych (energoelektronika, napęd, energetyka, ...). Zadanie realizacji układu regulacji z regulatorem histerezy podzielone jest na sukcesywnie realizowane etapy. Przewidziany, łączny czas realizacji zadania to dwa spotkania laboratoryjne.

### 1. Komparator z histerezą (regulator histerezy)

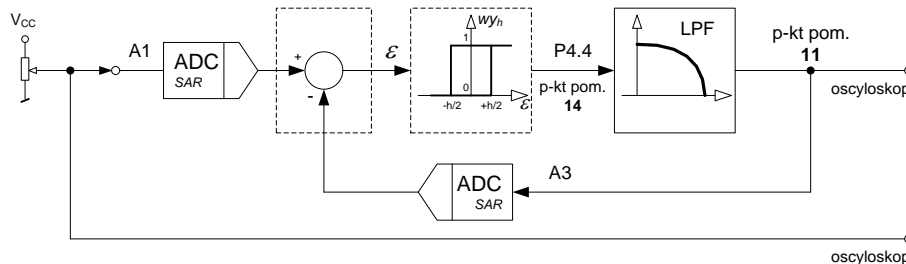
Należy napisać procedurę realizującą działanie komparatora z histerezą. Do kontroli poprawności działania napisanej procedury należy aplikować poniższy schemat.



- Uwagi
- procedura ma być umieszczona w przerwaniu od timer'a
  - szerokość strefy histerezy ma być określona wartością umieszczoną w jednym z rejestrów Rx

### 2. Zamknięty układ regulacji histerezy

Należy napisać procedurę realizującą działanie układu regulacji histerezy zgodnie z poniższym rysunkiem.



- Uwagi
- procedura ma być umieszczona w przerwaniu od timer'a
  - filtr dolnoprzepustowy LPF jest zrealizowany sprzętowo (schemat i opis w „Mikroprocesorowy system laboratoryjny” Blok #4)



**Stoper**

Należy napisać program tworzący w pełni funkcjonalny stoper. Sterowanie stoperem przyciskami uruchamiającymi trzy podstawowe funkcje: **Start**, **Stop**, **Reset**. Wszystkie funkcje stopera realizowane wyłącznie w przerwaniach.

- Jednostki czasu – sugerowane 0,1s z dokładnością min. 0,01s
  - Funkcje przycisków określone przez prowadzącego na zajęciach:
    - 1) trzy przyciski - **Start**, **Stop**, **Reset**
    - 2) dwa przyciski – wówczas funkcje **Start/Stop** są realizowane sekwencyjnie
    - 3) jeden przycisk – wszystkie funkcje są realizowane sekwencyjnie
  - Wyświetlane 2 cyfry na wyświetlaczach 7-mio segmentowych (dziesiątki i jedności)
  - Obsługa przycisków w przerwaniach od timer'a lub przerwań zewnętrznych
- Start** i **Stop** powinny być realizowane bezzwłocznie tzn. z opóźnieniem  $\max \leq 0,01s$

**GPIO 1**

Należy napisać program realizujący poniżej opisaną funkcjonalność.

Do portu P4 doprowadzone są zewnątrz dane (przyciski P4.0-P4.3) tak, że w każdej chwili czasowej tylko jedna linia portu odbiera sygnał L (zero logiczne). Na port P2 należy przekazywać bajt o wartości liczbowej odpowiadającej numerowi linii portu P4, na której aktualnie jest stan L.

**GPIO 2**

Należy napisać program realizujący poniżej opisaną funkcjonalność.

Na linii P2.0 należy generować przebieg prostokątny o stałej częstotliwości, ale wypełnieniu zależnym od zewnętrznie wymuszonego stanu linii P4.0:

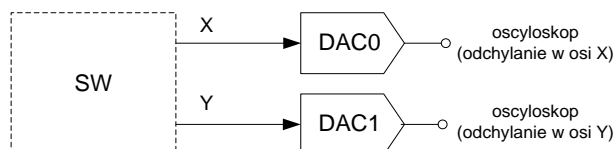
M=0.5 gdy P4.0 = L

M=0.25 gdy P4.0 = H

## Sterowanie X-Y

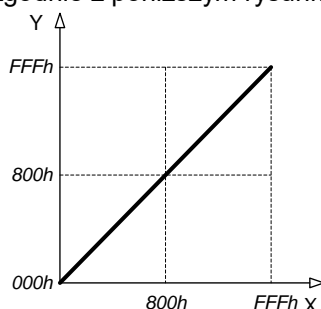
W układzie mikroprocesora znajdują się dwa 12-b przetworniki typu DAC. Przetworniki i ich obsługa opisane są w „User’s Guide” MSP430x1xx Family dostępnej na stronie materiałów laboratorium.

Należy napisać procedurę umożliwiającą narysowanie (plamka) na ekranie oscyloskopu wskazanej przez prowadzącego figury geometrycznej np. romb, trójkąt, inicjał itd.



Ponieważ wskazane figury składają się z odcinków linii prostych (równania liniowe 1-go rzędu) to nasuwającą się metodą rysowania jest obliczanie wartości Y na podstawie równania odpowiedniego odcinka  $Y=f(X)$  w odpowiednich przedziałach zmiennej X. Pełny zakres zmienności X i Y to przedział (000h; FFFh). Wynika on z rozdzielczość przetworników (12-b). Możliwe są też do zastosowania inne metody.

Pierwszym krokiem zadania laboratoryjnego będzie narysowanie linii prostej zgodnie z poniższym rysunkiem:



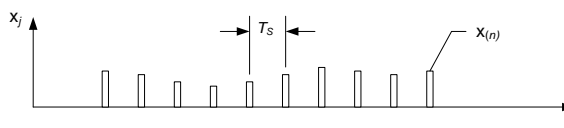
**Komentarz** Obraz na ekranie oscyloskopu będzie widoczny, jeśli plamka będzie się przesuwała po kreślonej trajektorii cyklicznie, z jak największą częstotliwością. Z tego powodu należy ustawić maksymalną częstotliwość używanego zegara systemowego DCO lub przełączyć sygnały zegarowe MCLK i SMCLK do uruchomionego oscylatora kwarcowego XT2 (8MHz), zgodnie z poniżej podaną procedurą. Procedura nie wyłącza oscylatora DCO.

Procedura uruchamiania oscylatora XT2 i przełączenia sygnałów zegarowych MCLK i SMCLK	bis.b	#OSCOFF,SR	;turn OFF osc.1
	bic.b	#XT2OFF,BCSCTL1	;turn ON osc.2
	BCM0	bic.b #OFIFG,&IFG1	;clear OFIFG
	mov	#0FFFFh,R15	;delay (waiting for oscilator start)
	BCM1	dec R15	;delay
	jnz	BCM1	;delay
	bit.b	#OFIFG,&IFG1	;test OFIFG
	jnz	BCM0	;repeat test if needed
		bic.b	#040h,&BCSCTL2
	bic.b	#080h,&BCSCTL2	;select XT2CLK as source
	bic.b	#030h,&BCSCTL2	;MCLK=source/1 (8MHz)
			;SMCLK
	bis.b	#SELS,&BCSCTL2	;select XT2CLK as source
	bic.b	#006h,&BCSCTL2	;SMCLK=source/1 (8MHz)

**Filtr wykładniczy I-rzędu**

- A. Należy napisać procedurę tzw. filtru wykładniczego I-rzędu umożliwiającego redukcję szumu sygnału wejściowego.

Zdigitalizowany przebieg analogowy



równanie filtra

$$y(n) = \alpha \cdot x(n) + (1 - \alpha) \cdot y(n-1)$$

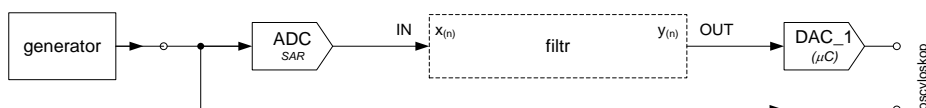
Zaletą filtru wykładniczego jest możliwość regulowania wielkości redukcji szumu poprzez zmianę współczynnika wygładzania  $\alpha$  ( $0 < \alpha < 1$ ). Im mniejsze  $\alpha$ , tym większa redukcja szumu oraz tym wolniej filtr odpowiada na zmiany w sygnale wejściowym.

- B. Należy ocenić poprawność działania filtru stosując schematy (Rys.1) przedstawione poniżej.

Rys.1 (a)



Rys.1 (b)



- C. Z badać odpowiedź impulsową i skokową filtru stosując schemat (Rys.2) przedstawiony poniżej. Wymaga on napisania programu generatora impulsów i skoków jednostkowych.

Rys.2



Uwaga współczynnik  $\alpha$  (0;1/4;1/2;3/4,1)

Oceńnik

- wykonanie punktów A+B – ocena 40 pkt. max
- wykonanie punktów A+B+C – ocena 50 pkt. max

Proponowana literatura – R.G. Lyons „Wprowadzenie do cyfrowego przetwarzania sygnałów”, WKŁ

**Obsługa wyświetlacza LCD (HD447802)**

Wykorzystując mikrokontroler MSP430 oraz wyświetlacz LCD (2x16) ze sterownikiem HD44780 należy napisać program (assembler), wyświetlający następujące napisy na wyświetlaczu:

Wyświetlany napis

Zespół nr : Z  
Napięcie : X.Y.V

W miejscu litery „Z” należy wstawić numer swojego zespołu laboratoryjnego, natomiast w drugiej linii w miejsce liter X.Y należy wstawić wartość napięcia z potencjometru, mierzonego na pinie pomiarowym TP9. Do realizacji zadania należy wykorzystać program, który był realizowany na jednym z poprzednich zajęć (woltomierz) oraz przykładowy program umieszczony na [stronie](#). Zmieniając położenie potencjometru wartość wyświetlanego napięcia powinna być aktualizowana.

Sposób podłączenia  
wyświetlacza LCD  
do wyjść mikrokontrolera

Na płytkach laboratoryjnych wykorzystywany jest 4 bitowy tryb przesyłania danych z wykorzystaniem 4 linii danych (P2.2, P2.3, P2.4, P2.5). Ponadto linia R/W jest na stałe podłączona do masy układu, a brak możliwości zmiany stanu tej linii skutkuje brakiem możliwości odczytu rejestrów wyświetlacza. Linia RS podłączona do wyjścia P2.0 odpowiada za przesyłanie komend (stan niski) lub danych (stan wysoki). Linia E podłączona do wyjścia P2.1 mikrokontrolera powoduje zatraskiwanie (na zboczu opadającym) danych w rejestrach wyświetlacza.

Opis przykładowego  
programu

W programie umieszczonym na [stronie](#) realizowana jest procedura inicjalizująca wyświetlacz a następnie wyświetlany są litery „ab” w środkowej części pierwszego wiersza. W pierwszych liniach programu deklarowane są stałe, które są wykorzystywane w dalszej części programu:

```
E          EQU BIT1
RS          EQU BIT0
DB4         EQU BIT2
DB5         EQU BIT3
DB6         EQU BIT4
DB7         EQU BIT5
DATA4       EQU 00111100b
FuncSet     EQU 00001000b
DDRAMSet    EQU 10000000b
```

Jedną z pierwszych procedur wywoływanych jest procedura inicjalizująca wyświetlacz LCD:

```
call #init_lcd
```

W procedurze tej wyświetlacz przestawiany jest w tryb 4 bitowy oraz ustawiana jest czcionka, a następnie ustawiany jest kursor i załączany wyświetlacz. Ponadto przestawiany jest adres pamięci DDRAM wyświetlacza na wartość 5, co powoduje przestawienie kursora na pozycję 5 i od tej pozycji będą wypisywane znaki. Ustawienie adresu pamięci DDRAM jest realizowane przez wysłanie odpowiedniego rozkazu (przy niskim stanie linii RS) do wyświetlacza. Jest to realizowane poprzez umieszczenie kodu rozkazu w rejestrze R12, a następnie wywołanie procedury *send4bit*, które realizuje właściwą transmisję, co zostało przedstawione poniżej.

```
;... set ddram adres .....;
mov.b #DDRAMSet+5, R12 ; ustawienie adresu 5
call  #send4bit
```

Procedura *send4bit* przesyła 8 bitowa dana/rozkaz w trybie 4 bitowym. Odbywa się to w dwóch cyklach: najpierw wysyłane są starsze 4 bity, a następnie młodsze 4 bity. Procedura wykorzystuje rejestry R12, R13 i R15. Przed wywołaniem procedury daną do wysłania należy umieścić w rejestrze R12. Po inicjalizacji wyświetlacza linia RS jest ustawiana w stan wysoki, aby umożliwić przesył danych i wysyłane są dwa znaki „ab” do wyświetlacza:

```
bis.b #RS, P2OUT
mov.b #'a', R12
call  #send4bit
mov.b #'b', R12
call  #send4bit
```

Ocennik    10 pkt – prawidłowa konfiguracja portów i przerwań  $\mu C$   
             10 pkt – uruchomienie przetwornika A/C oraz wyświetlacza LCD  
             15 pkt – wyświetlenie statycznego napisu w dwóch wierszach  
             15 pkt – aktualizacja wartości wyświetlanego napięcia

---

**Literatura**

---

- [1] materiały wykładowe
- [2] „MSP430x1xx Family User's guide” (slau049f.pdf), Texas Instruments, 2006
- [3] „Corrections to *MSP430x1xx Family User's Guide*” (slau049.pdf), Texas Instruments, 2016
- [4] „MSP430F15x, MSP430F16x, MSP430F161x Mixed Signal Microcontroller datasheet (Rev. G)”, Texas Instruments, 2011
- [5] „Wprowadzenie do cyfrowego przetwarzania sygnałów”, R.G. Lyons, WKŁ