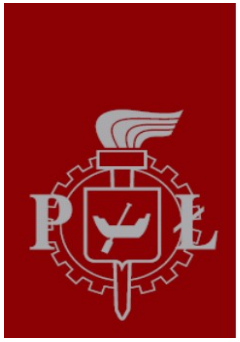


Politechnika Łódzka
Instytut Elektroniki

Programowanie Mikrokontrolerów]

Architektura układu ATmega128(L), część 2.

mgr inż. Paweł Poryzała
Zakład Elektroniki Medycznej

Zagadnienia

- Przerwania zewnętrzne
  - INT
  - PCINT
- Timer/Counters
- USART
- Komparator analogowy
- Przetwornik ADC

Przerwania zewnętrzne INTx

- Końcówki INT0 .. INT7.
- Jeśli włączone działają niezależnie od kierunku danej linii portu (ew. przerwania software'owe).
- Opcje wyzwalania przerwań:
  - Zbocza – opadające i narastające, ew. dowolne,
  - Niski poziom logiczny.
- Kontrola przerwań zewnętrznych – rejestry: EICRA, EICRB, EIMSK, EIFR.

Przerwania zewnętrzne INTx

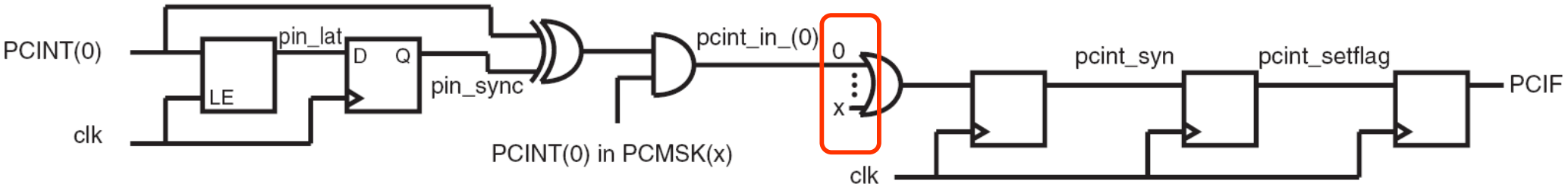
- Aby przerwanie zewnętrzne mogło wyprowadzić układ ze stanu *Sleep Mode* innego niż *Idle* musi ono działać w sposób asynchroniczny!

Table 50. Interrupt Sense Control(1)

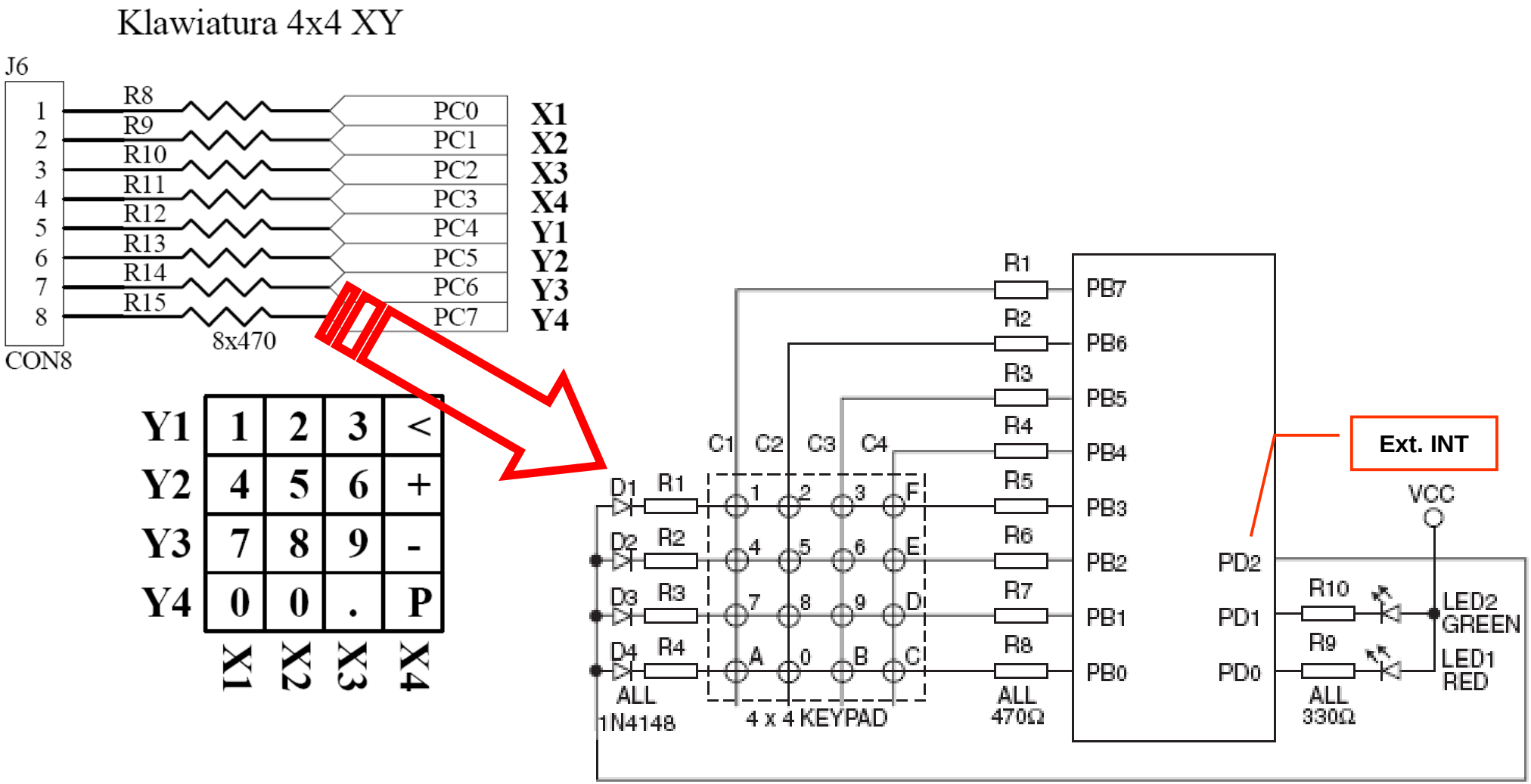
ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any logical change on INTn generates an interrupt request
1	0	The falling edge between two samples of INTn generates an interrupt request.
1	1	The rising edge between two samples of INTn generates an interrupt request.

Przerwania zewnętrzne PCINTx

- Umożliwiają użycie znacznie większej liczby przerwań zewnętrznych
- Kilka źródeł przerwań (typowo po 8 pinów, z możliwością maskowania) połączonych jest wspólnie, wyzwala jedno wspólne ISR
- Opcje wyzwalania – dowolna zmiana stanu logicznego
- Kontrola przerwań PCINT – PCICR, PCIFR, PCMSKx

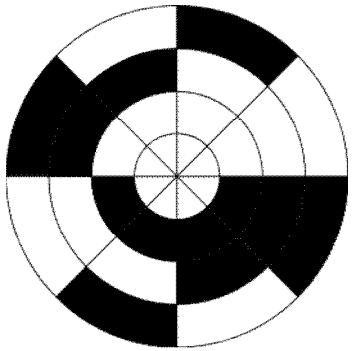


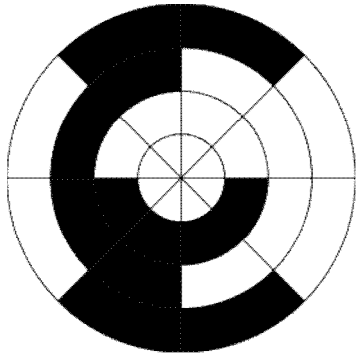
Wykorzystanie przerwań zewnętrznych

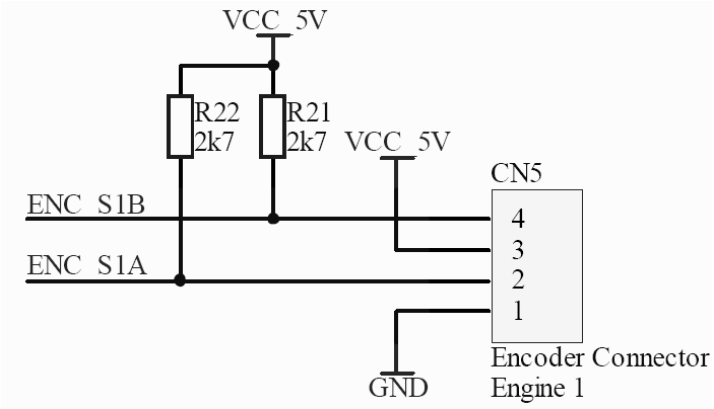


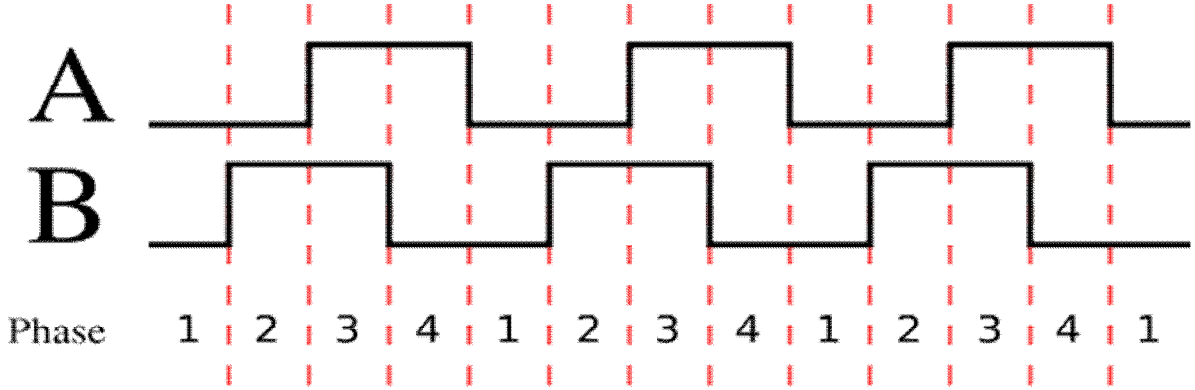
Wykorzystanie przerwań zewnętrznych

- Enkoder
  - Absolutny,
  - Inkrementalny (impulsator, enkoder kwadraturowy).

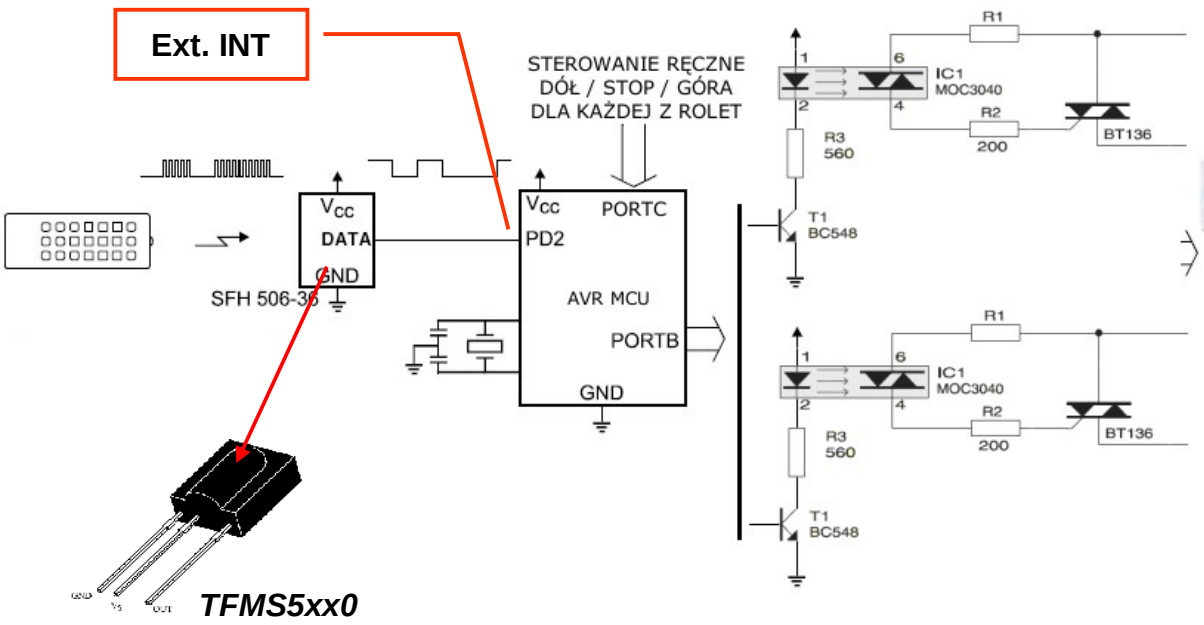


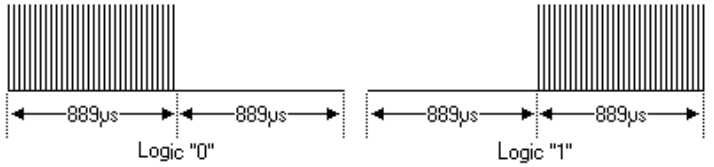


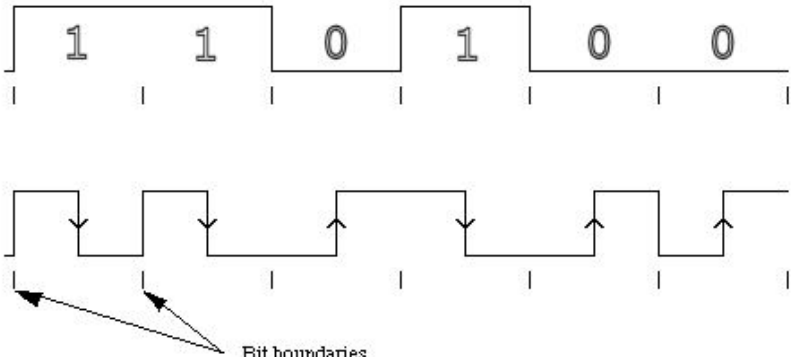


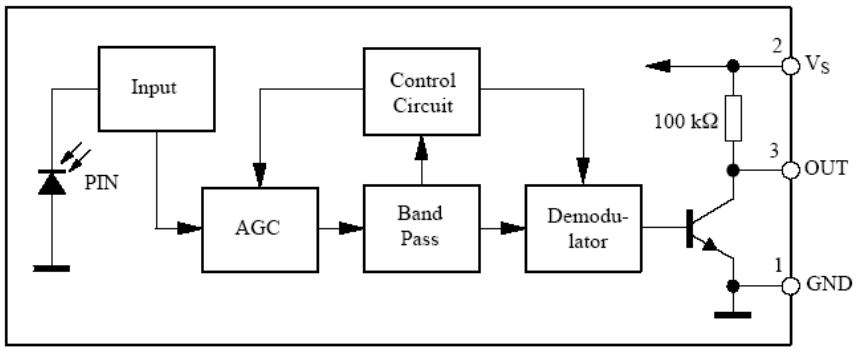


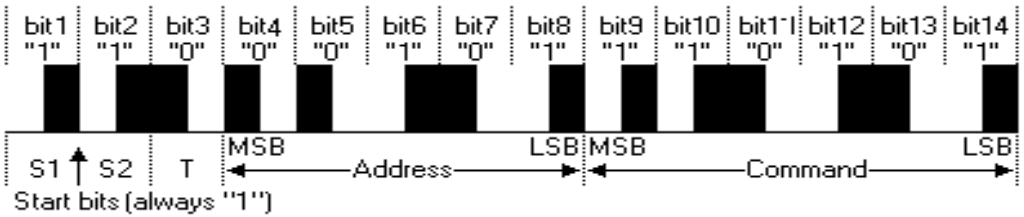
Wykorzystanie przerwań zewnętrznych



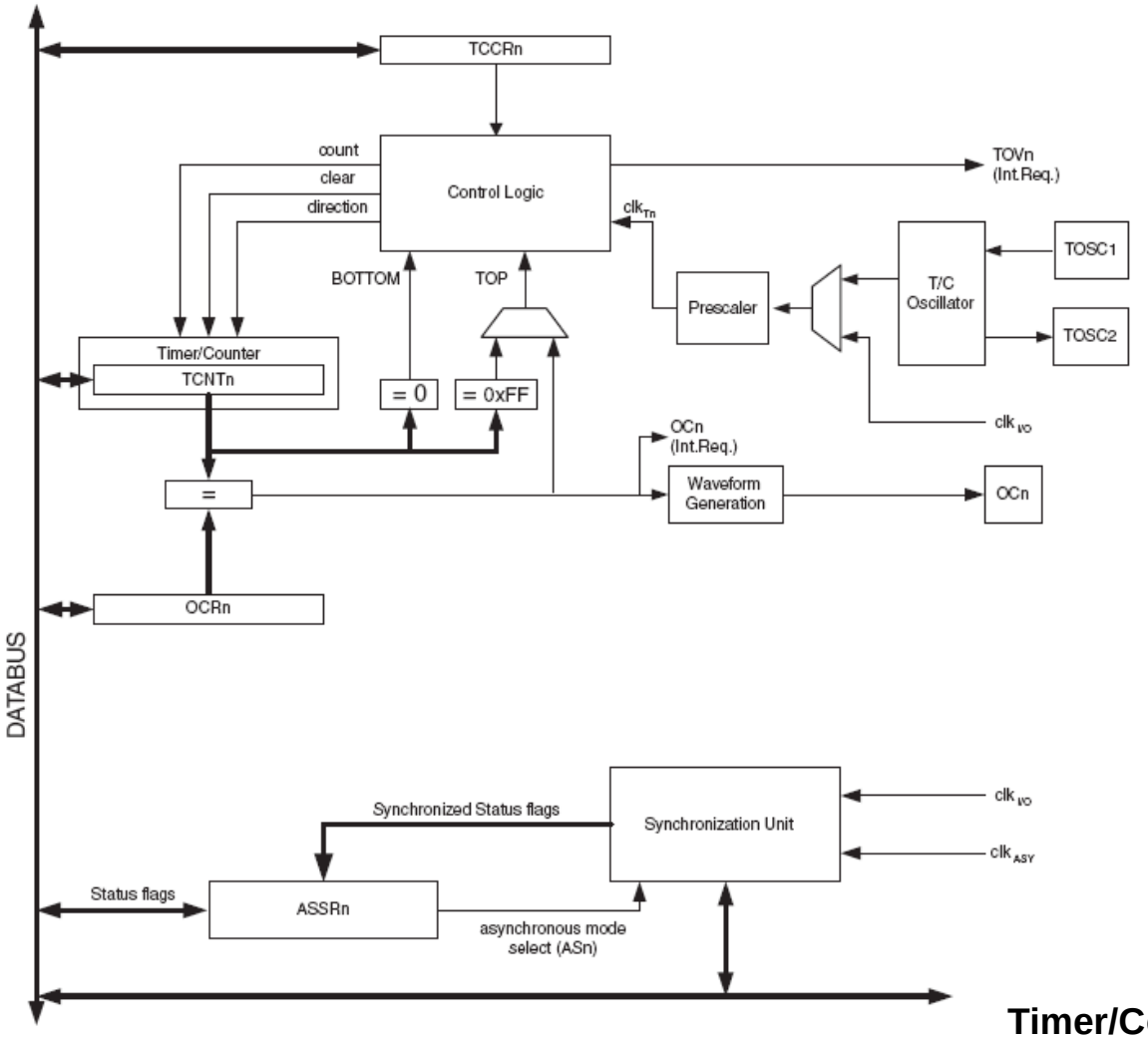








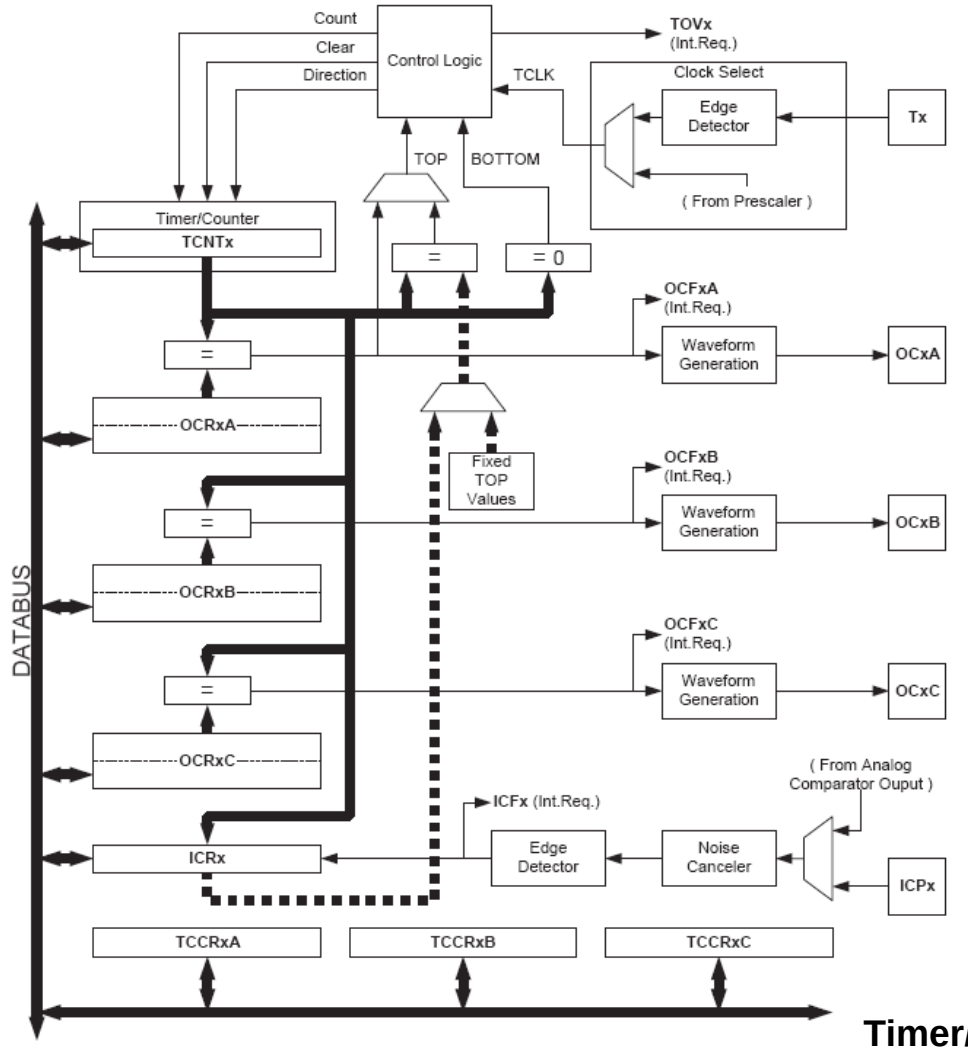
Układy Timer/Counters



Timer/Counter 0 (8-bit) Block Diagram

- Timer/Counter 0
  - 8-bit
  - Clear Timer On Compare Match (auto reload)
  - Phase Correct PWM
  - 10-bit prescaler
  - Async. Operation (External 32kHz Watch Crystal)

Układy Timer/Counters



- Timer/Counter 1 and 3
  - 16-bit
  - 3 niezależne bloki Output Compare
  - 1 blok Input Capture
  - Clear Timer On Compare Match (auto reload)
  - Phase Correct PWM, Phase and Frequency Correct PWM

Timer/Counter 1, 3 (16-bit) Block Diagram

Układy Timer/Counters

- Tryby pracy:
  - Normal Mode
  - Clear Timer on Compare Match (CTC) Mode
  - Fast PWM Mode
  - Phase Correct PWM Mode
  - Phase and Frequency Correct PWM Mode

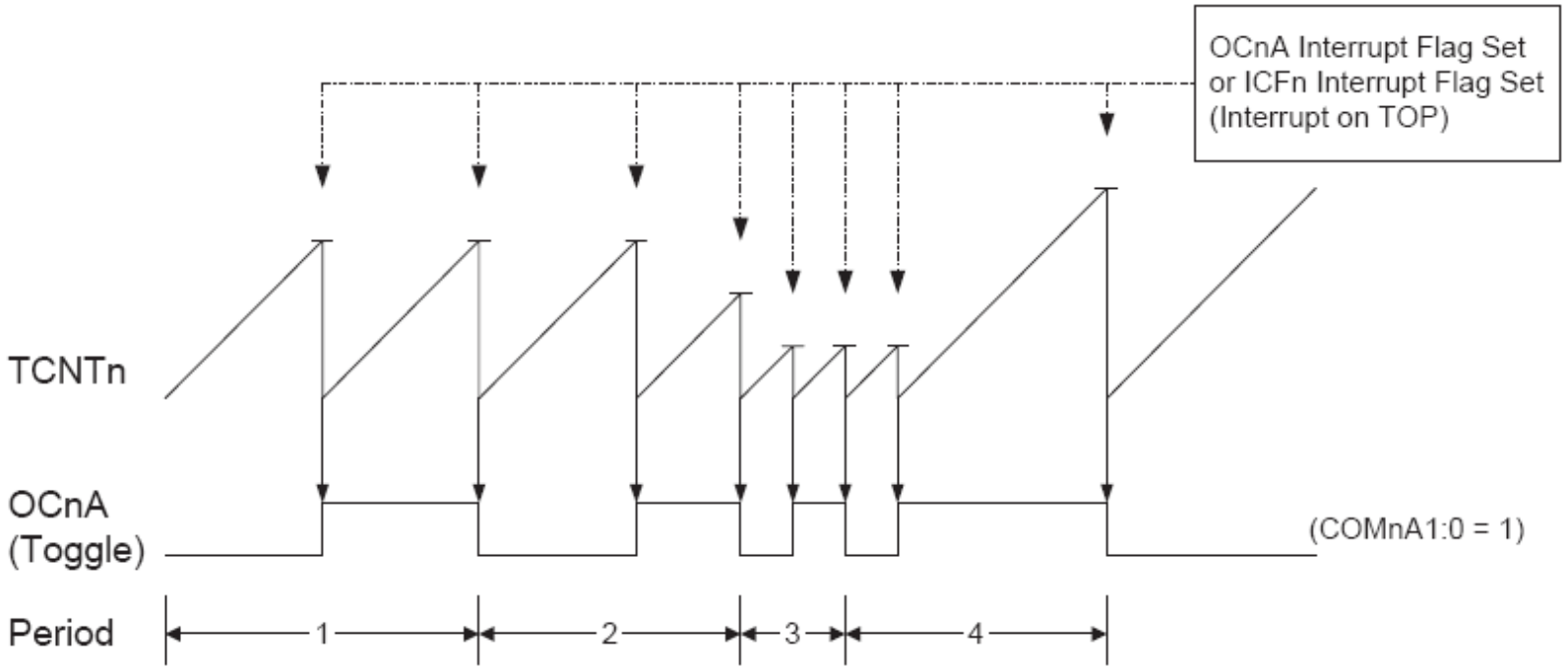
Różne tryby pracy do różnych zastosowań…
Jak one wyglądają…

Układy Timer/Counters – tryby pracy

- Normal Mode
  - Najprostszy
  - Zliczanie zawsze do góry, bez automatycznego zerowania (tylko przepełnienie)
  - Flaga przerwania TOVn ustawiana w momencie przepełnienia licznika
  - Możliwość zwiększania rozdzielczości licznika / odmierzanych czasów przez software (np. ustawianie wartości początkowej).

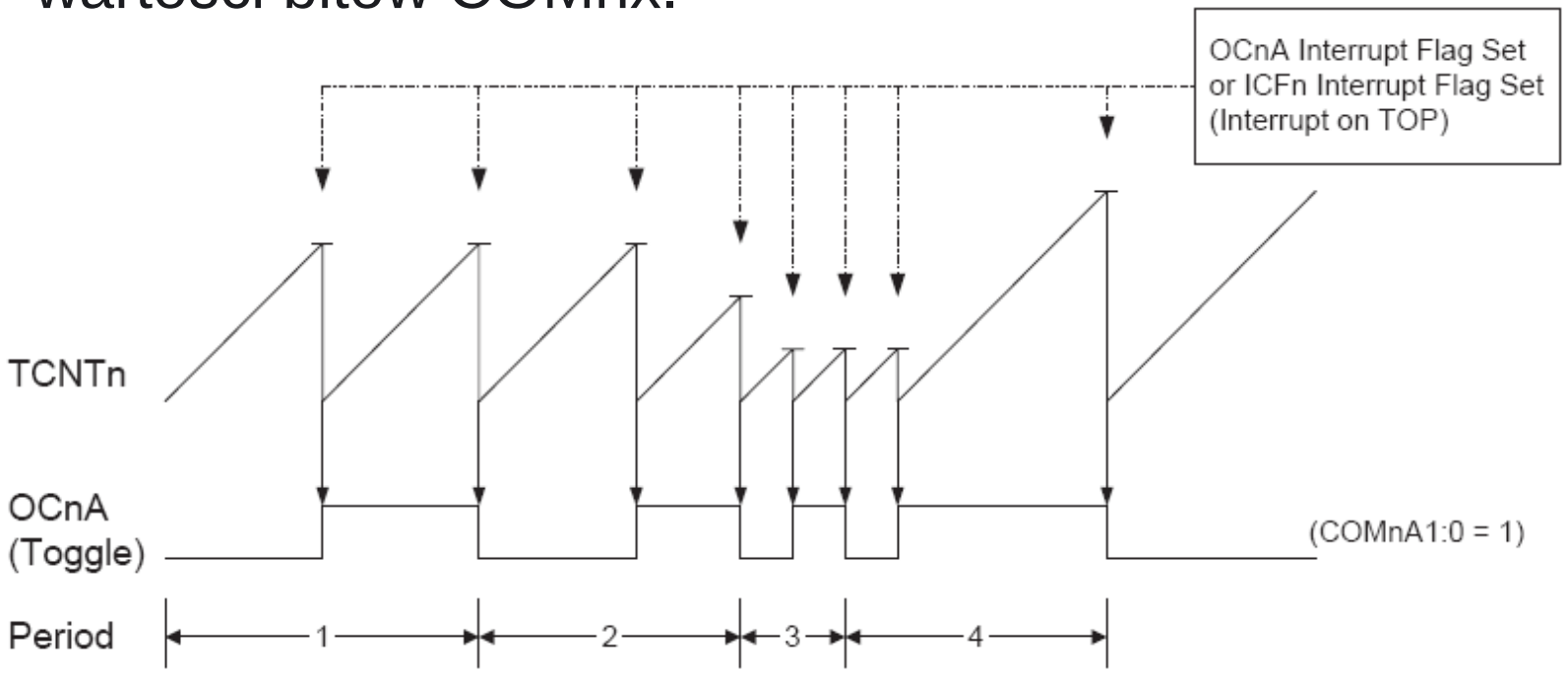
Układy Timer/Counters – tryby pracy

- Clear Timer on Compare Match (CTC) Mode
  - Rejestry OCRnx lub ICRn używane do kontroli rozdzielczości licznika (wartości maksymalnej),
  - Przerwania OCFnx lub ICFn w momencie osiągnięcia wartości maksymalnej,
  - Rejestry definiujące TOP nie są podwójnie buforowane – problem, gdy nowa wartość < aktualnej wartości TCNT,
  - Możliwość generowania przebiegów na wyjściu OCnx.



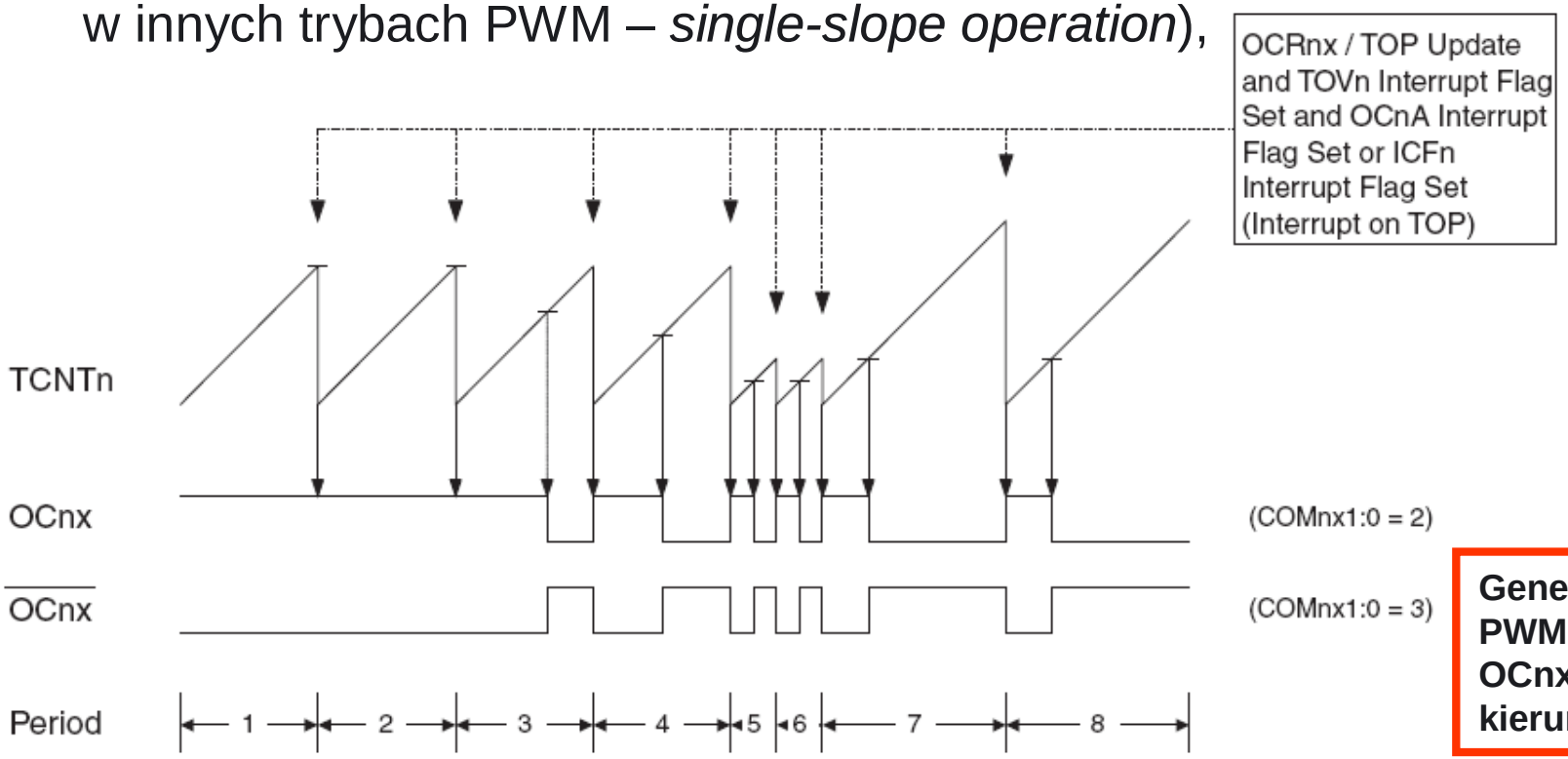
Układy Timer/Counters – tryby pracy

- Clear Timer on Compare Match (CTC) Mode
  - Możliwość generowania przebiegów na wyjściu OCnx (uwaga na kierunek linii!!!),
  - Zmiana sposobu działania linii OCnx poprzez definicje wartości bitów COMnx.



Układy Timer/Counters – tryby pracy

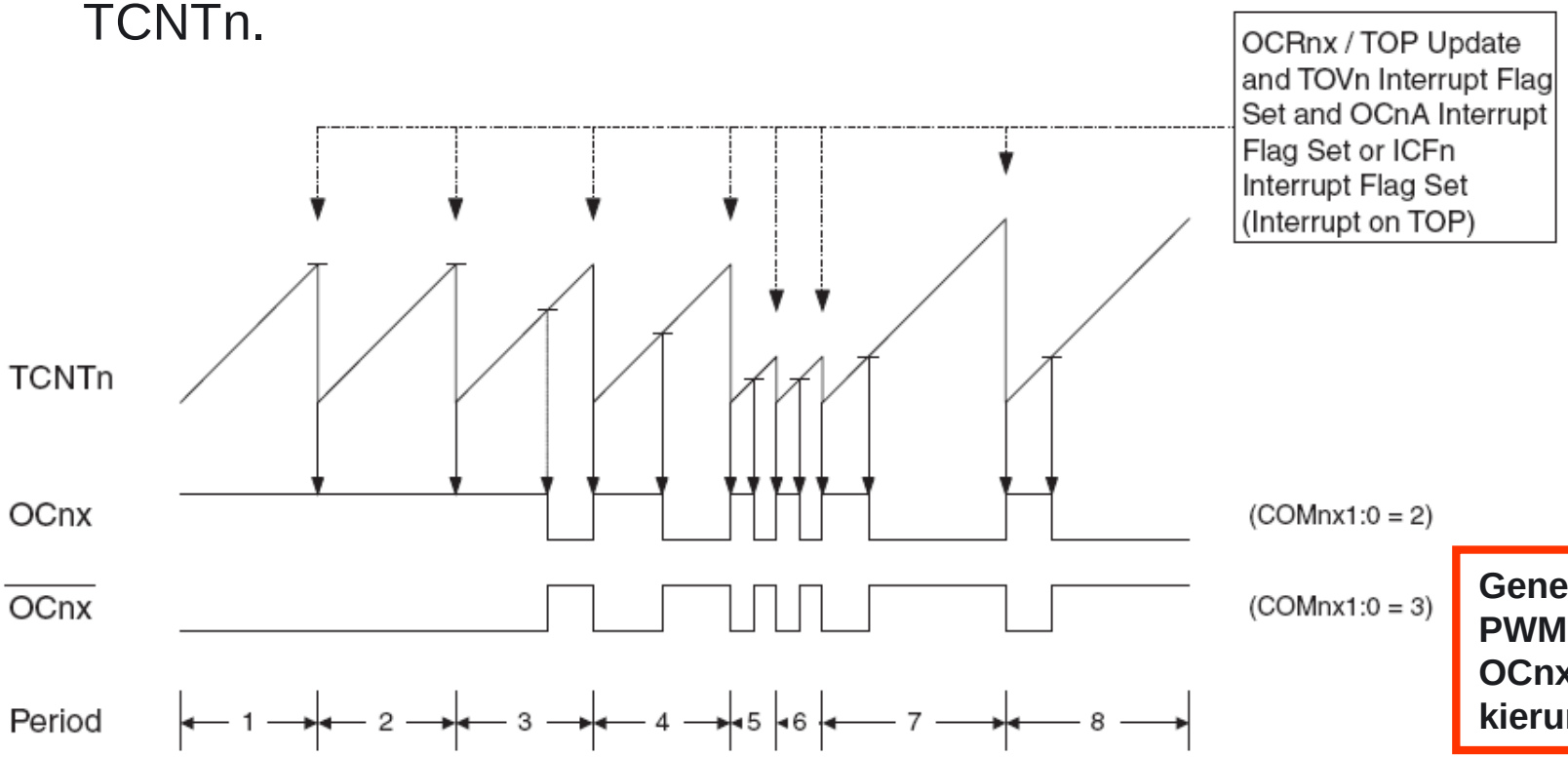
- Fast PWM Mode
  - Rozdzielczość 8-, 9-, 10-bit lub definiowana przez wartość rejestrów ICRn / OCRnx,
  - Generowanie przebiegu PWM wysokiej częstotliwości (2x większa niż w innych trybach PWM – *single-slope operation*),



Generowanie PWM na wyjściu OCnx (uwaga na kierunek linii!!!),

Układy Timer/Counters – tryby pracy

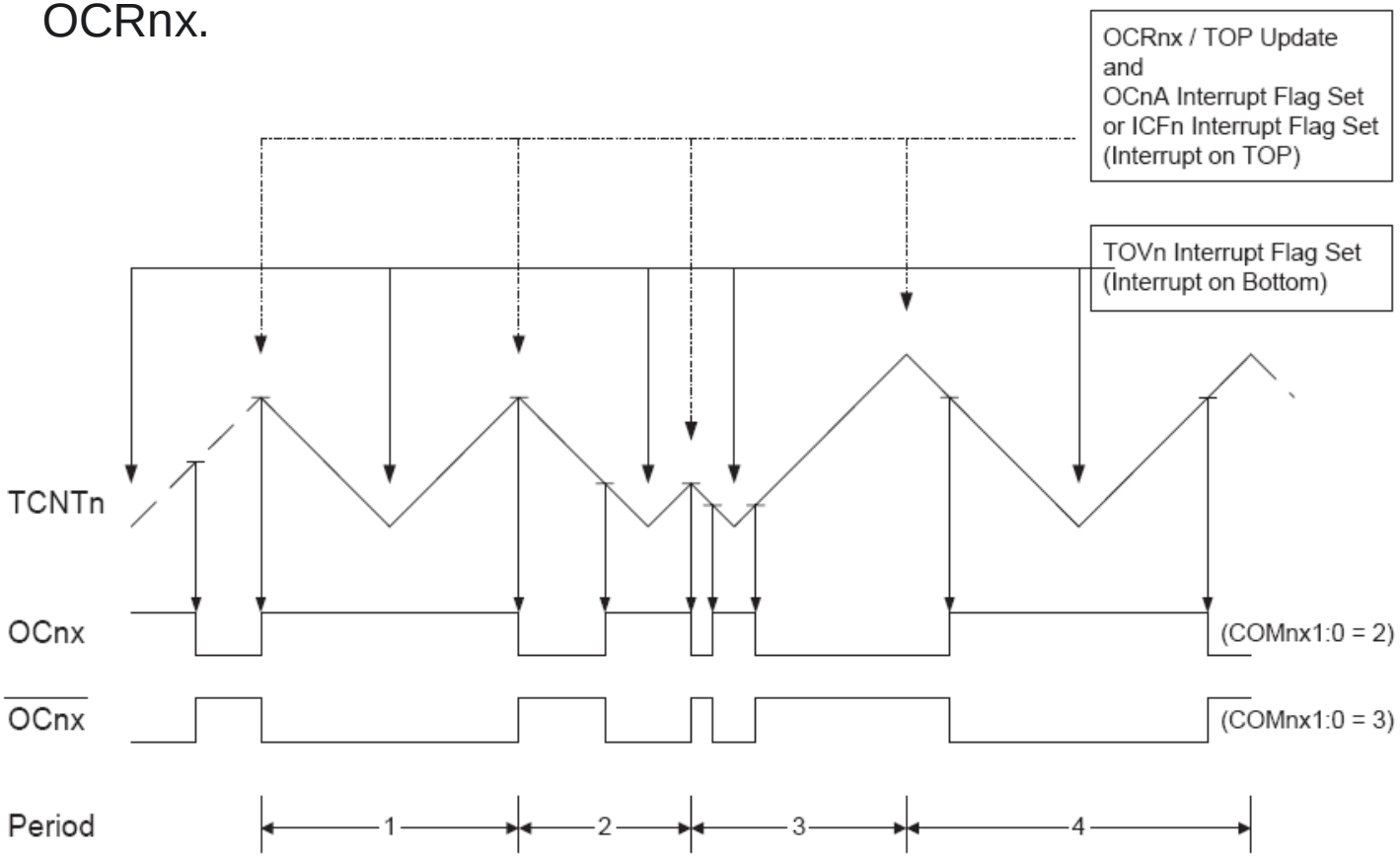
- Fast PWM Mode
  - Wysoka częstotliwość – regulacja mocy odbiorników, aplikacji PWM jako DAC (małe rozmiary el. zewnętrznych),
  - Flaga OCnx ustawiana w momencie zrównania wartości OCRnx oraz TCNTn.



Generowanie PWM na wyjściu OCnx (uwaga na kierunek linii!!!),

Układy Timer/Counters – tryby pracy

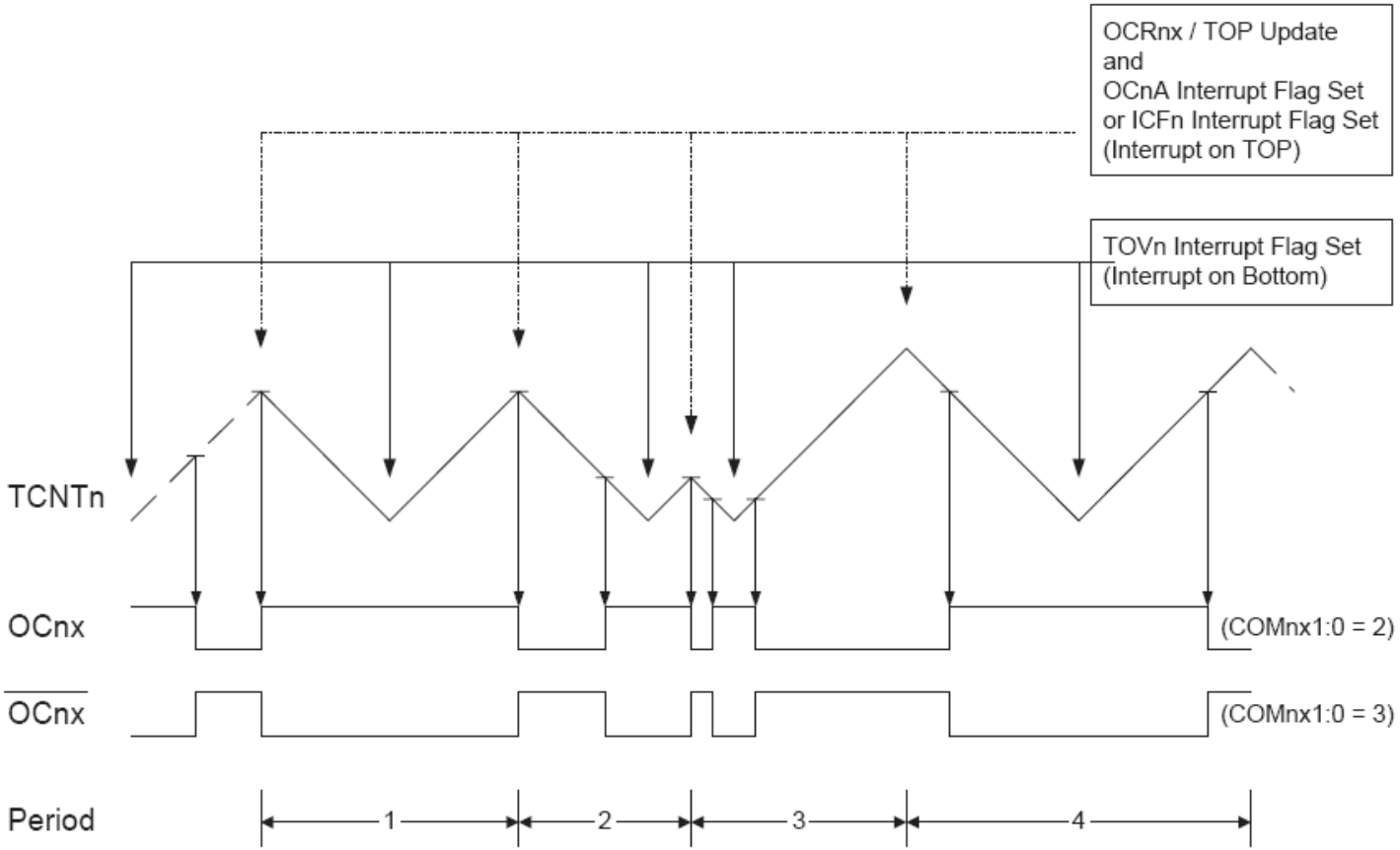
- Phase Correct PWM Mode
  - Phase Correct PWM wysokiej rozdzielczości, *dual-slope operation*,
  - Rozdzielczość 8-, 9-, 10-bit lub definiowana przez wartość rejestrów ICRn / OCRnx.



Generowanie PWM na wyjściu OCnx (uwaga na kierunek linii!!!),

Układy Timer/Counters – tryby pracy

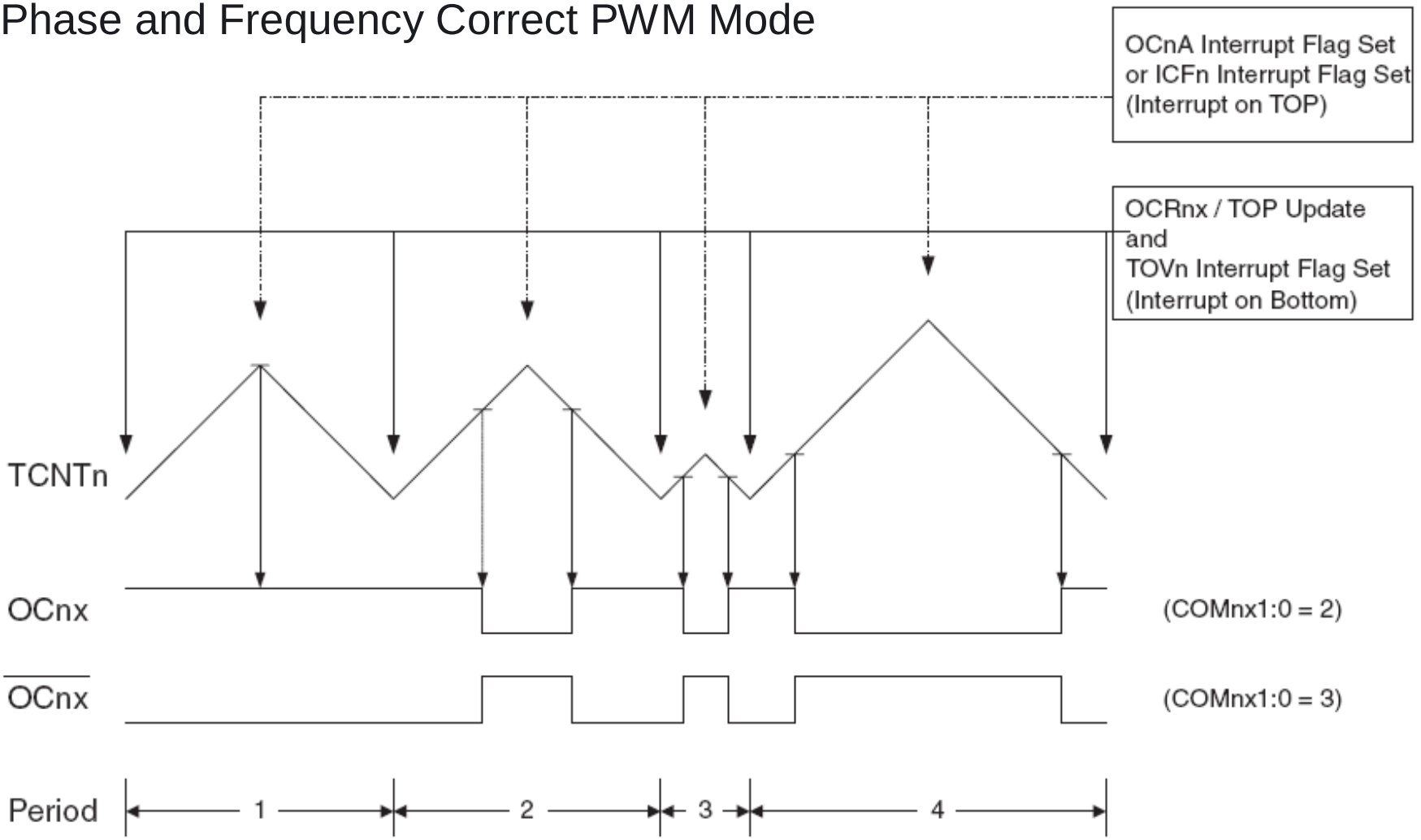
- Phase Correct PWM Mode
  - Flaga OCnx ustawiana w momencie zrównania wartości OCRnx oraz TCNTn,
  - Zmiana sposobu działania linii OCnx poprzez definicje wartości bitów COMnx.



Generowanie PWM na wyjściu OCnx (uwaga na kierunek linii!!!),

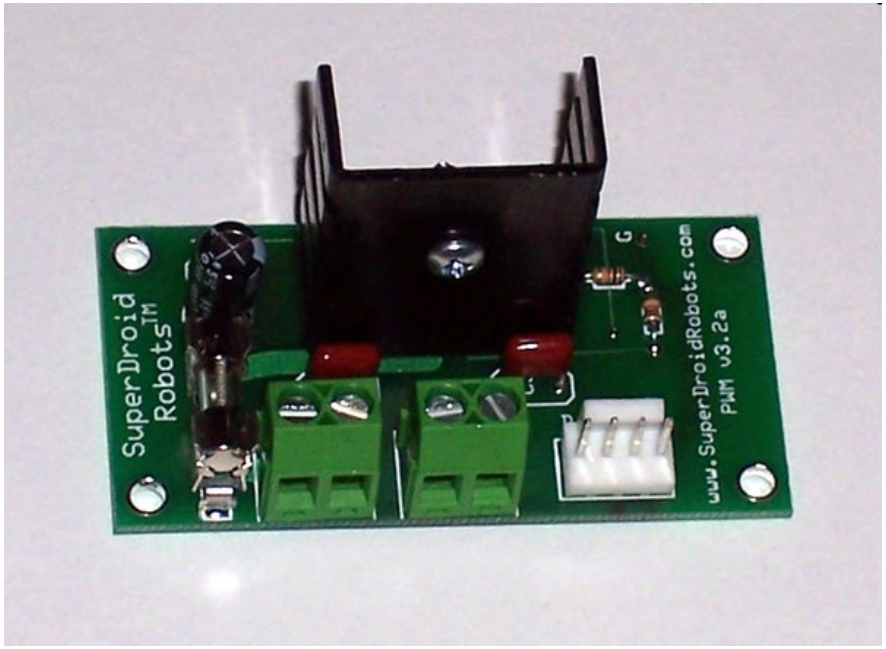
Układy Timer/Counters – tryby pracy

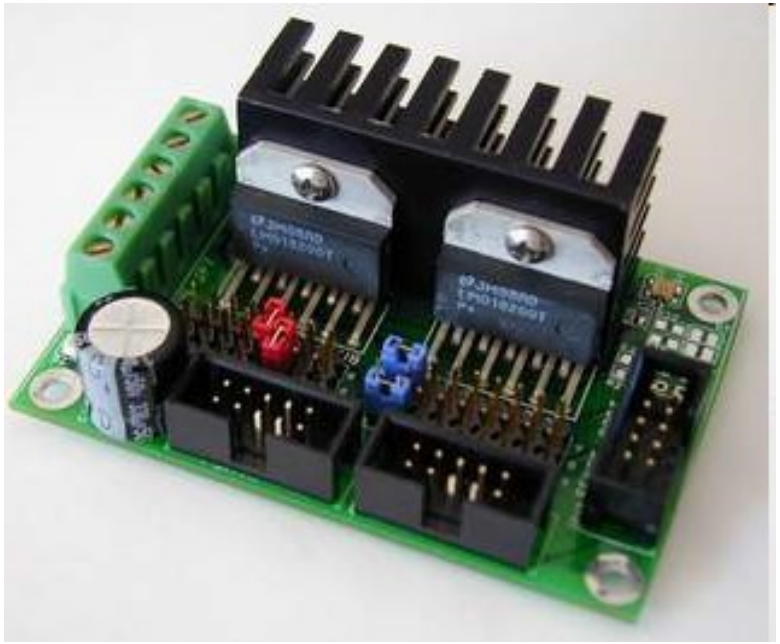
- Phase and Frequency Correct PWM Mode



Układy Timer/Counters – tryby pracy

- Phase and Frequency Correct PWM Mode
  - Phase Correct PWM wysokiej rozdzielczości, *dual-slope operation*,
  - Odpowiedni do sterowania silników,
  - Rozdzielczość definiowana przez wartość rejestrów ICRn / OCRnx – od 2 do 16 bit,
  - Zmiana sposobu działania linii OCnx poprzez definicje wartości bitów COMnx.





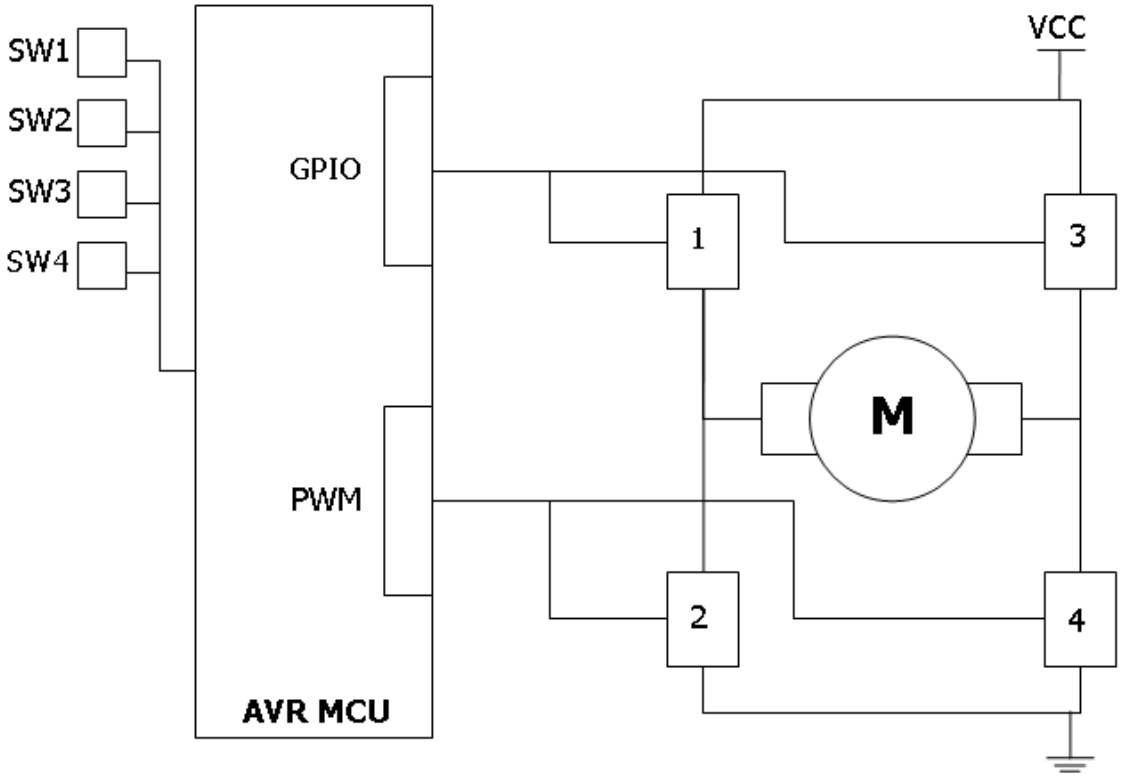
Układy Timer/Counters – tryby pracy, podsumowanie

Table 61. Waveform Generation Mode Bit Description

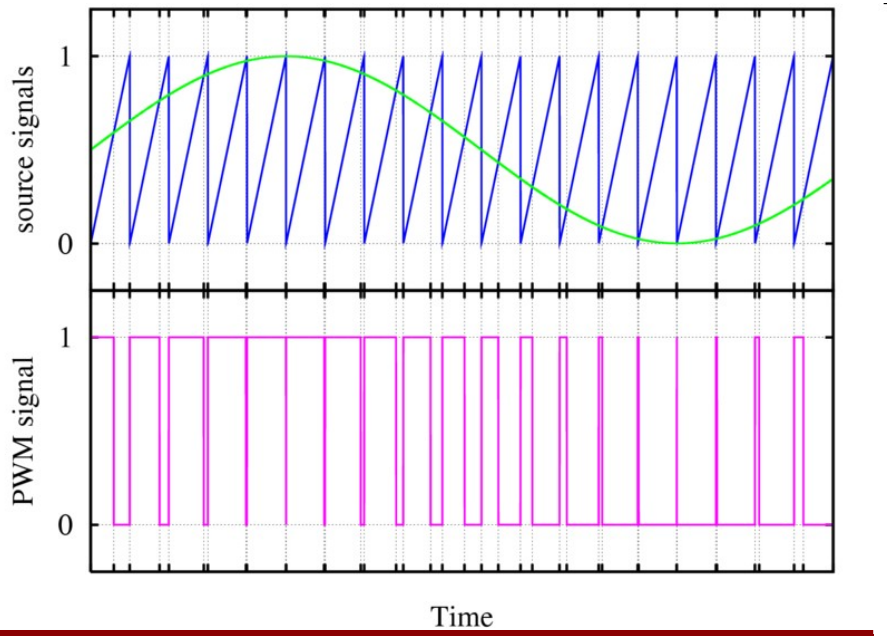
Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation[1]	TOP	Update of OCRnx at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

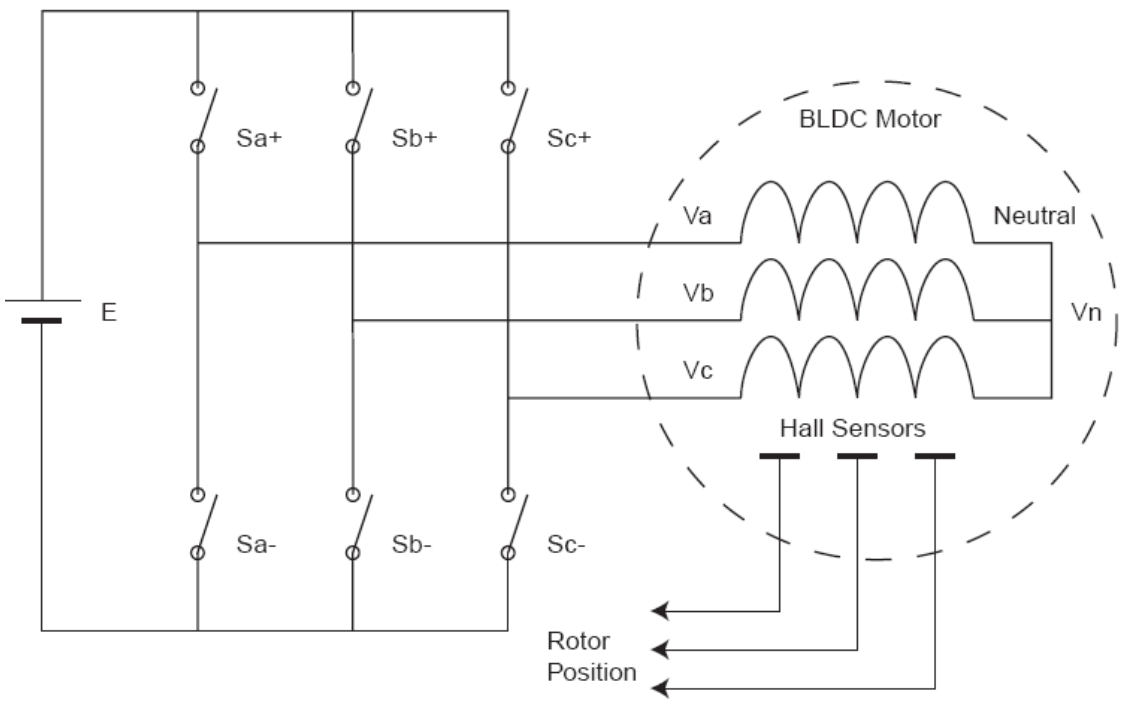
Wykorzystanie układów Timer/Counters



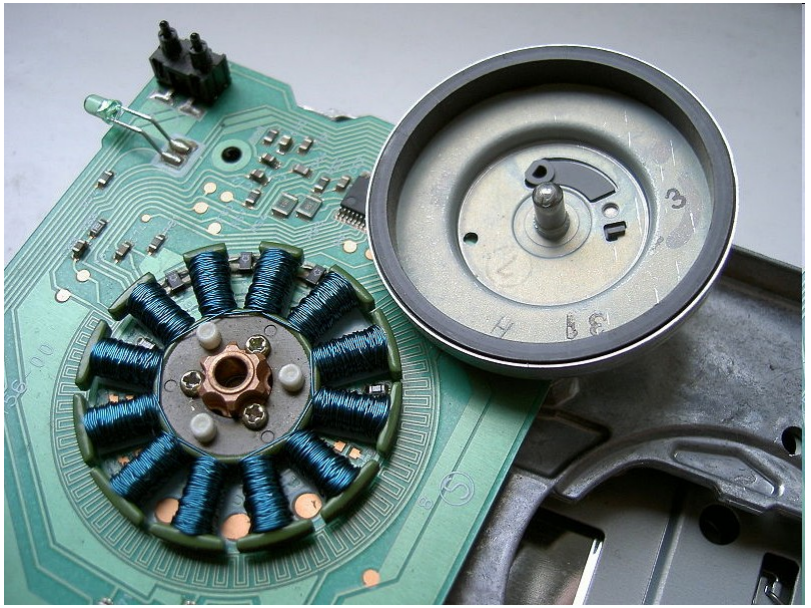
- GPIO – sterują kierunkiem obrotów oraz hamowaniem,
- PWM – steruje prędkością obrotową.



Wykorzystanie układów Timer/Counters



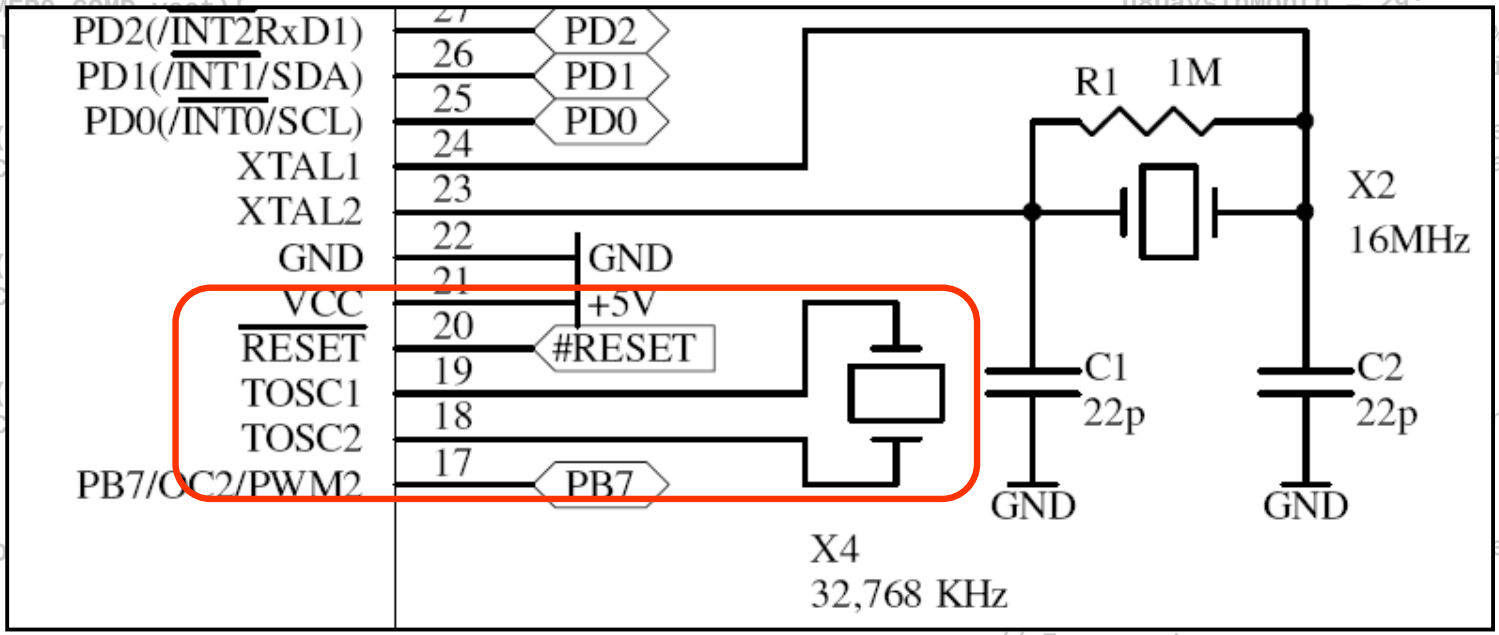
- BLDC – *BrushLess DC*
- Komutacja elektroniczna
- Budowa silnika DC / BLDC



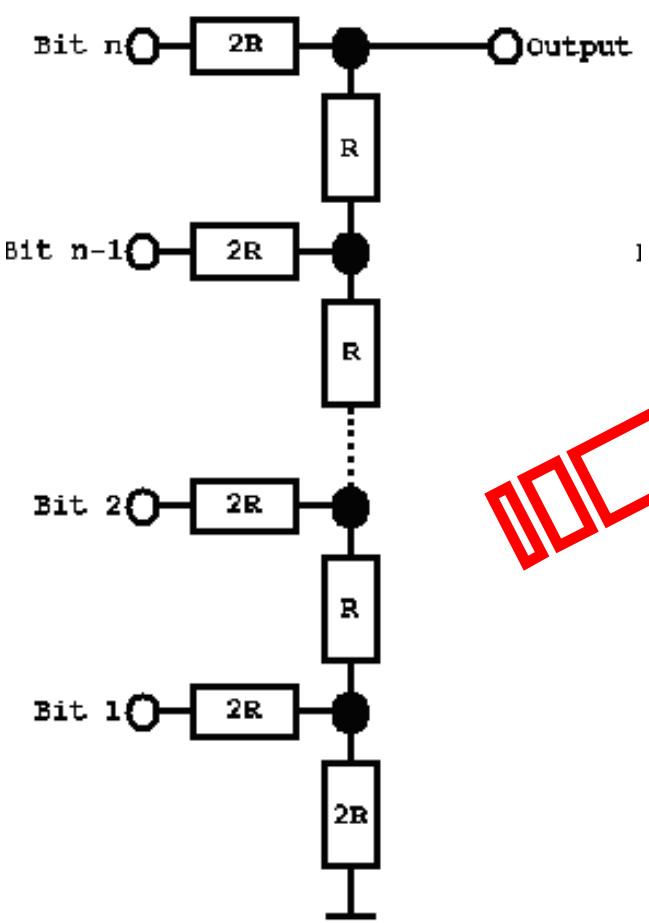
- Napięcia na uzwojeniach silnika,
- Silniki BLAC (*Brush Less AC*)

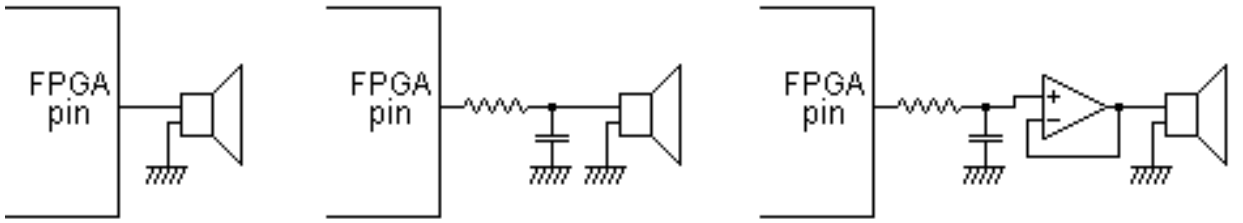
Wykorzystanie układów Timer/Counters

- Timer/Counter jako wbudowany RTC
  - Brak konieczności używania zewnętrznych układów RTC
  - Oszczędność pieniędzy oraz miejsca na PCB



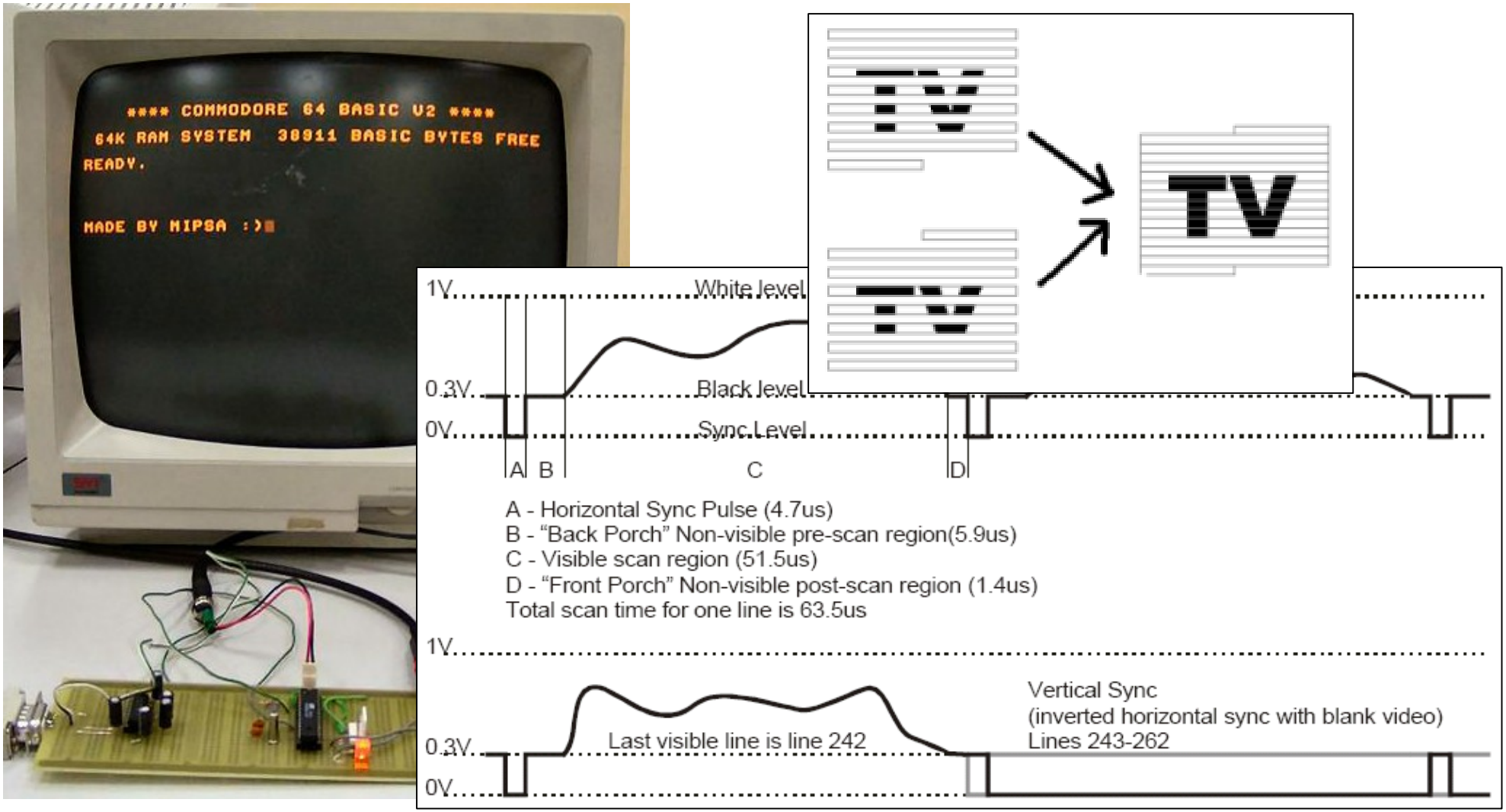
Wykorzystanie układów Timer/Counters



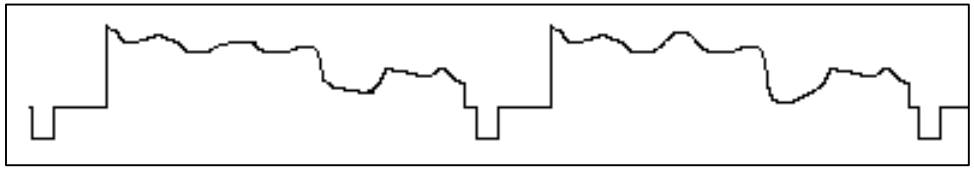


- Brak wbudowanego DAC:
  - Budowa drabinki R-2R (n-bitowy przetwornik wymaga n-GPIO)
  - Budowa PWM DAC

Wykorzystanie układów Timer/Counters

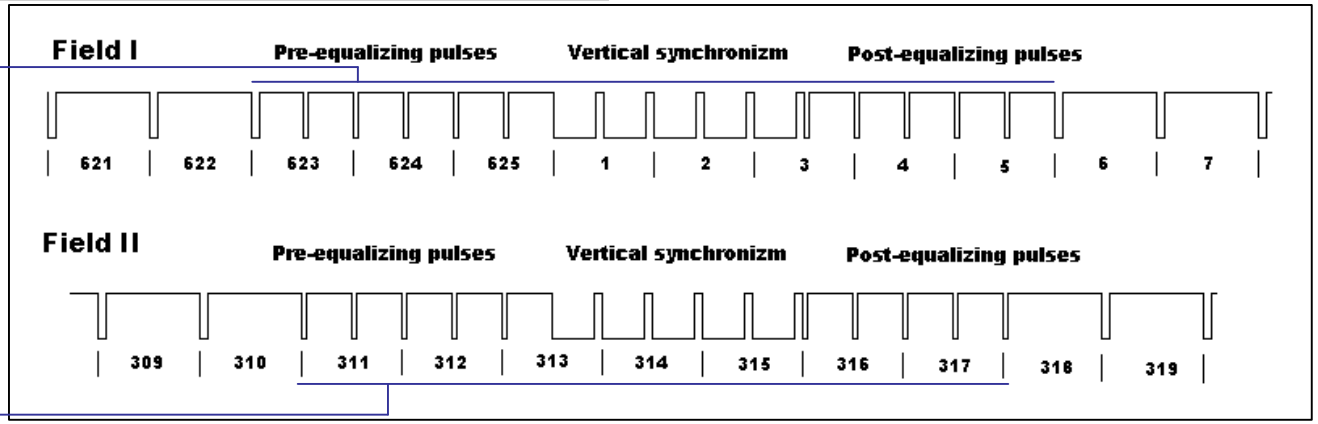


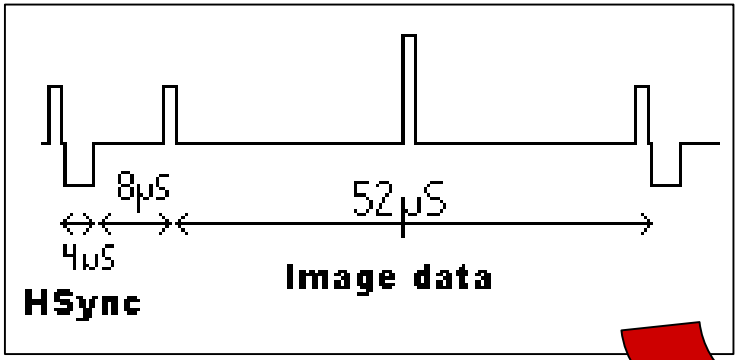
Wykorzystanie układów Timer/Counters

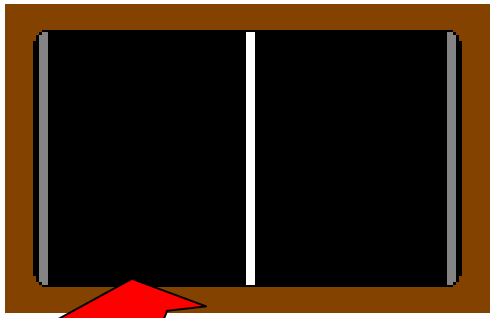


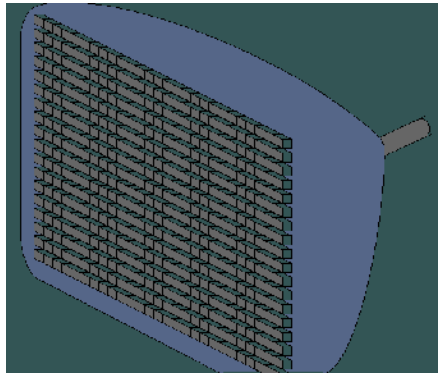
Wysyłanie kilku linii obrazu.

Impulsy synchronizacji pionowej po pierwszym półobrazie i całym obrazie - poziomy 0 i 0,3V.

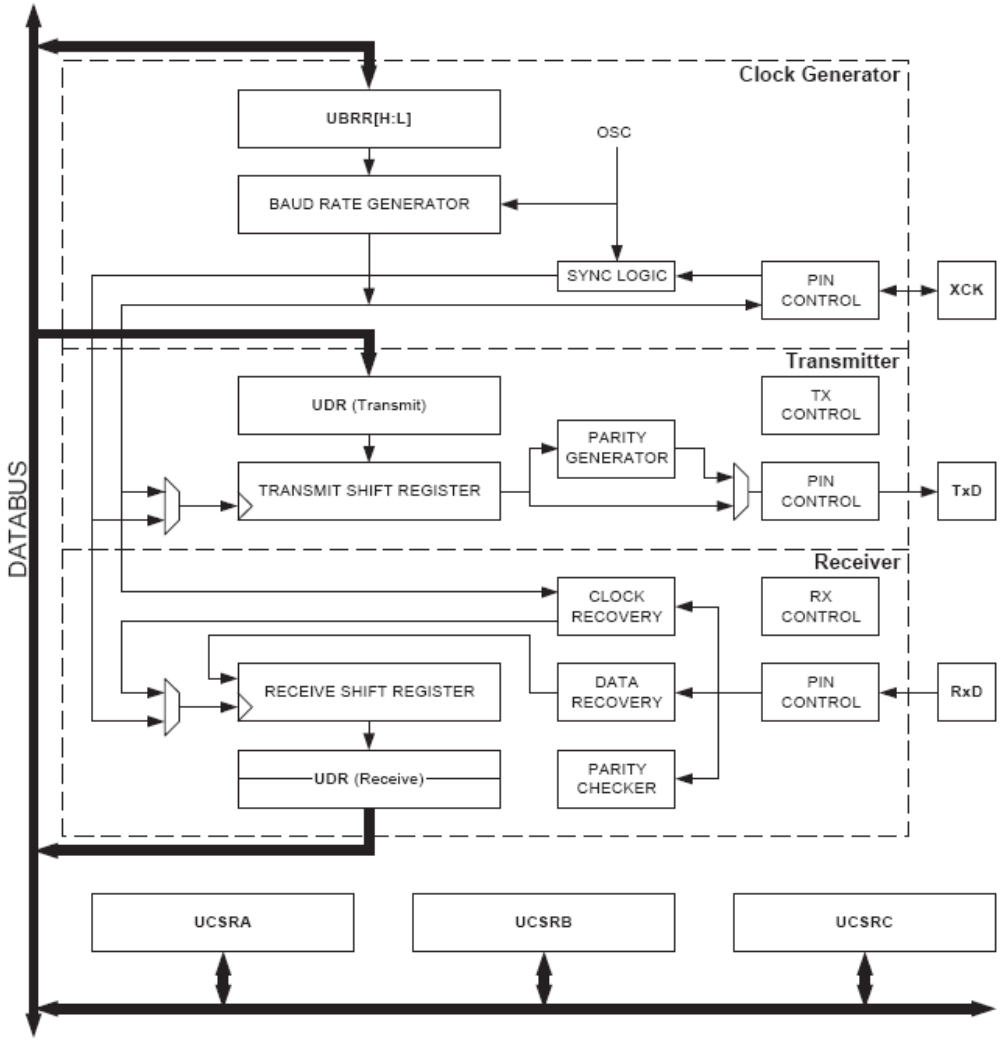








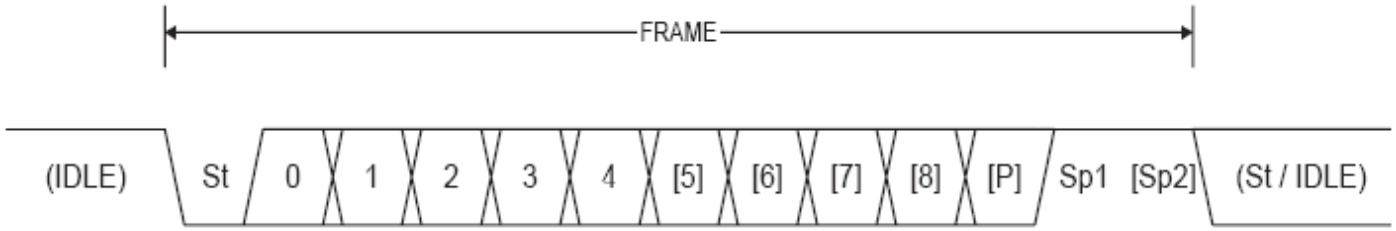
USART



USART Block Diagram

- USART0, USART1
  - Full Duplex
  - Praca asynchroniczna oraz synchroniczna
  - Obsługa ramek o długościach 5 .. 9-bit
  - Generator parzystości
  - Detekcja błędów ramki
  - Detekcja Data OverRun

USART – formaty ramek



- Format ramki ustala się w rejestrze UCSRnC
  - Tryb parzystości (*even*, *odd*)
  - Ilość bitów stop
  - Ilość bitów danych

  - Komunikacja asynchroniczna / synchroniczna (z wyborem polaryzacji zbocza zegara dla operacji odczytu / zapisu)

USART – źródła przerwań

- *RX Complete Interrupt*
  - w buforze odbiorczym znajdują się nieodczytane dane,
- *TX Complete Interrupt*
  - cała ramka (start, dane, parzystość, stop) została wysłana z *Transmitt Shift Register*, w rejestrze UDR nie ma więcej danych do wysłania,
- *USART Data Register Empty Interrupt*
  - bufor nadawczy UDR jest gotów do przyjęcia kolejnych danych.

USART – Baud Rate Generator

Table 84. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 8.0000 MHz				f_{osc} = 11.0592 MHz				f_{osc} = 14.7456 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25									
57.6k	8	-3.5%	16									
76.8k	6	-7.0%	12									
115.2k	3	8.5%	8									
230.4k	1	8.5%	3									
250k	1	0.0%	3									
0.5M	0	0.0%	1									
1M	–	–	0									
Max (1)	0.5 Mbps		1 Mbps									

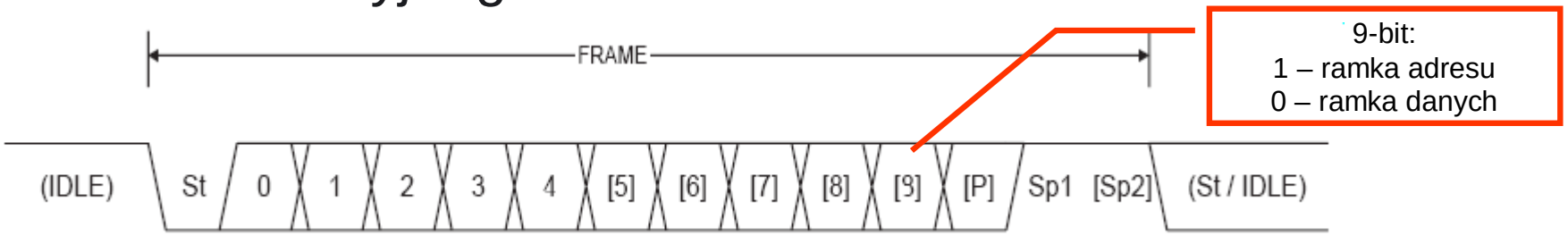
1. UBRR = 0, Error = 0.0%

Table 74. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate (1)	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

USART – tryb wieloprocesorowy

- Możliwość budowy prostej magistrali wieloprocesorowej
  - Jeden układ master, wiele układów slave,
  - Adresowanie układów podrzędnych (wybór adresata transmisji),
  - Zaadresowany układ przechodzi do normalnego trybu komunikacji (wyłącza tryb komunikacji wieloprocesorowej), reszta układów ignoruje transmisję,
  - Odróżnienie ramek adresowych / danych – typowo 9-bit słowa transmisyjnego



USART – kod przykładowy: Inicjalizacja

```
// !-- 1 --!
#define FOSC 14745600   // Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1

int main(void) {
   ...
   USART0_Init ( MYUBRR );
   ...
}

void USART0_Init(uint16_t ubrr) {

   // !-- 2 --!
   UBRR0H = (uint8_t)(ubrr>>8);
   UBRR0L = (uint8_t)ubrr;

   // !-- 3 --!
   UCSR0B = (1 << RXEN0)|(1 << TXEN0);

   // !-- 4 --!
   UCSRC = (3 << UCSZ00);
}
```

Komentarz do kodu:

- Definiując odpowiednie **FOSC** (podane w Hz) oraz **BAUD** unikamy szukania odpowiedniej wartości w tabelach na str. 194,

Inicjalizacja USART0:

2. ustawiamy odpowiednią prędkość transmisji (przekazaną w definicji **MYUBRR** przy wywołaniu funkcji inicjalizacji),
3. włączamy odbiornik oraz nadajnik USARTu,
4. konfigurujemy parametry ramki interfejsu: 8 bitów danych, jeden bit stopu brak parzystości

źródło: ATmega128(L) Datasheet

USART – kod przykładowy: Nadawanie

```
void USART0_Tx (uint8_t data)
{
   // !-- 1 --!
   while ( !( UCSR0A&(1<<UDRE0)) )
        ;

   // !-- 2 --!
   UDR0 = data;
}
```

Funkcja wysyłająca pojedynczy bajt:

1. *Pooling* bitu UDRE0 z rejestru UCSR0A – oczekiwanie na zwolnienie bufora danych mikrokontrolera (zabezpieczenie przed zamazaniem nie wysłanych jeszcze danych),
2. Zapisanie danych do rejestru UDR0 – transmisja następuje automatycznie.

źródło: ATmega128(L) Datasheet

USART – kod przykładowy: Odbiór

```
uint8_t USART_Rx(void)
{
   // !-- 1 --!
   while ( !(UCSR0A & (1<<RXC0)) )
        ;

    // !-- 2 --!
   return UDR0;
}
```

Funkcja odbierająca pojedynczy bajt:

1. *Pooling* bitu RXC0 z rejestru UCSR0A – oczekiwanie na zakończenie odbioru danych przez USART0
2. Odczyt odebranych danych z rejestru UDR0 oraz zwrócenie ich jako wartość funkcji.

źródło: ATmega128(L) Datasheet

Zastosowania

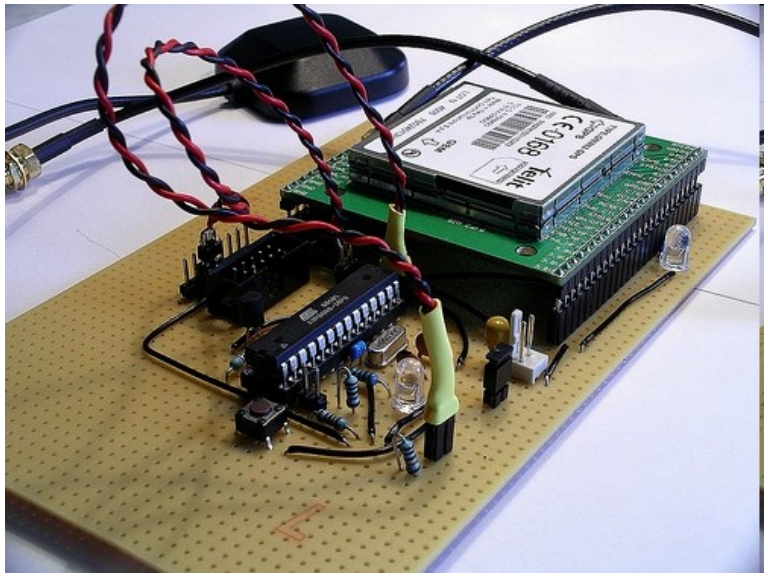
GPS

PC (serial port)

PDA

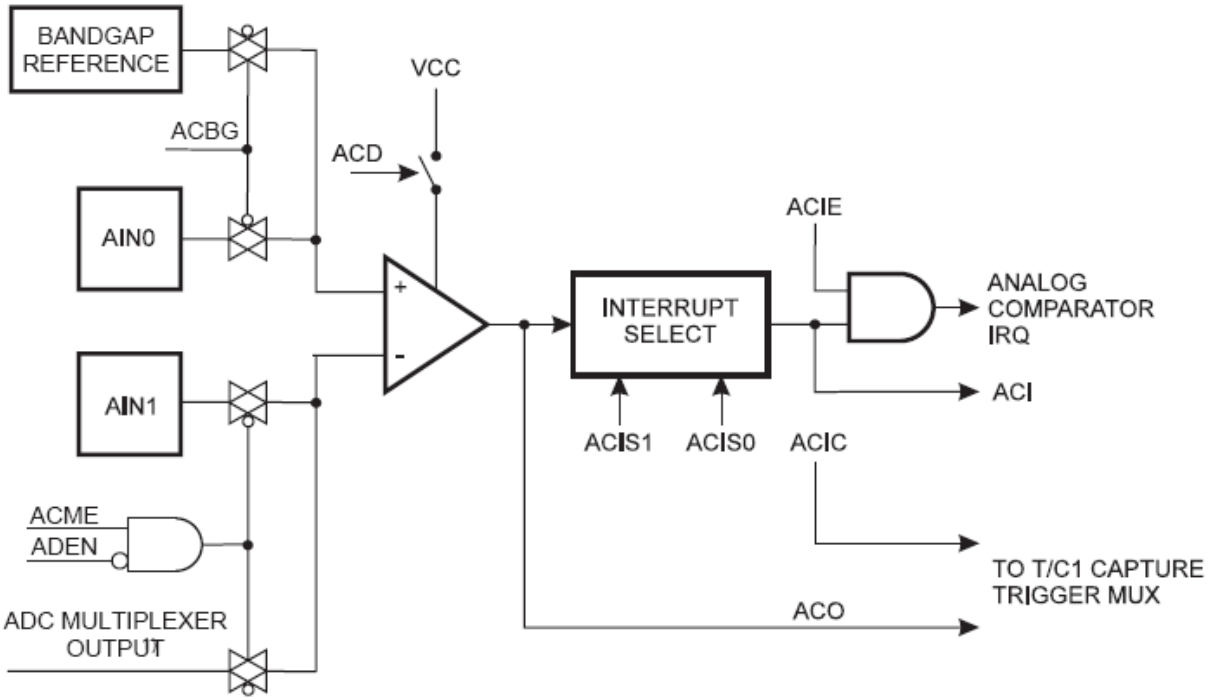
Wireless RC232 Modules

GSM



Bootloader'y
Interfejsy debugowania
I inne…

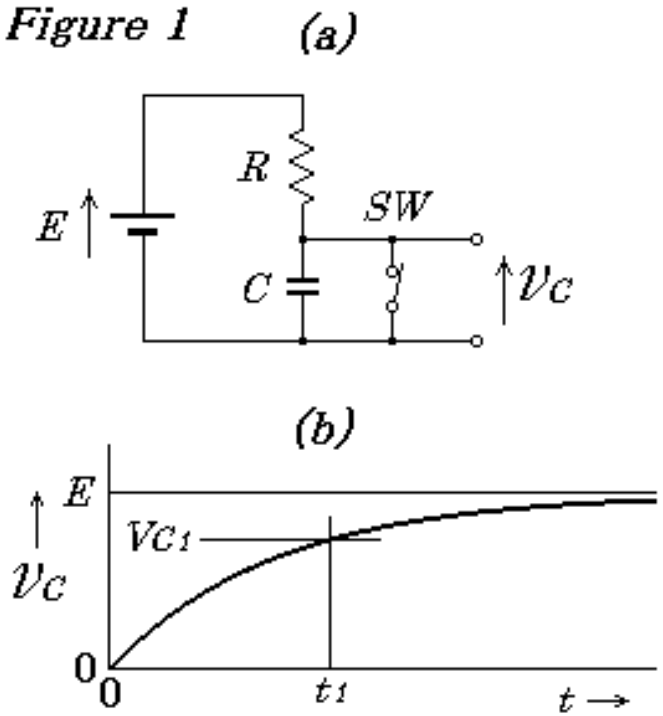
Komparator analogowy



Analog Comparator Block Diagram

- Analog Comparator:
  - Porównanie wartości z wejść AIN0 (+) oraz AIN1 (-)
  - Umożliwia wyzwolenie Input Capture Timer/Counter1
  - Przerwanie w chwili zbocza narastającego, opadającego lub zmiany stanu wyjścia komparatora

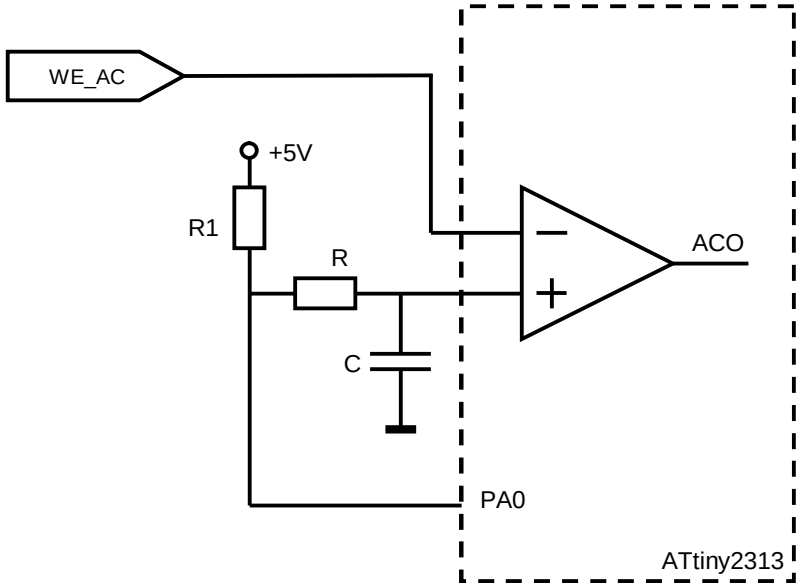
Zastosowanie komparatora analogowego



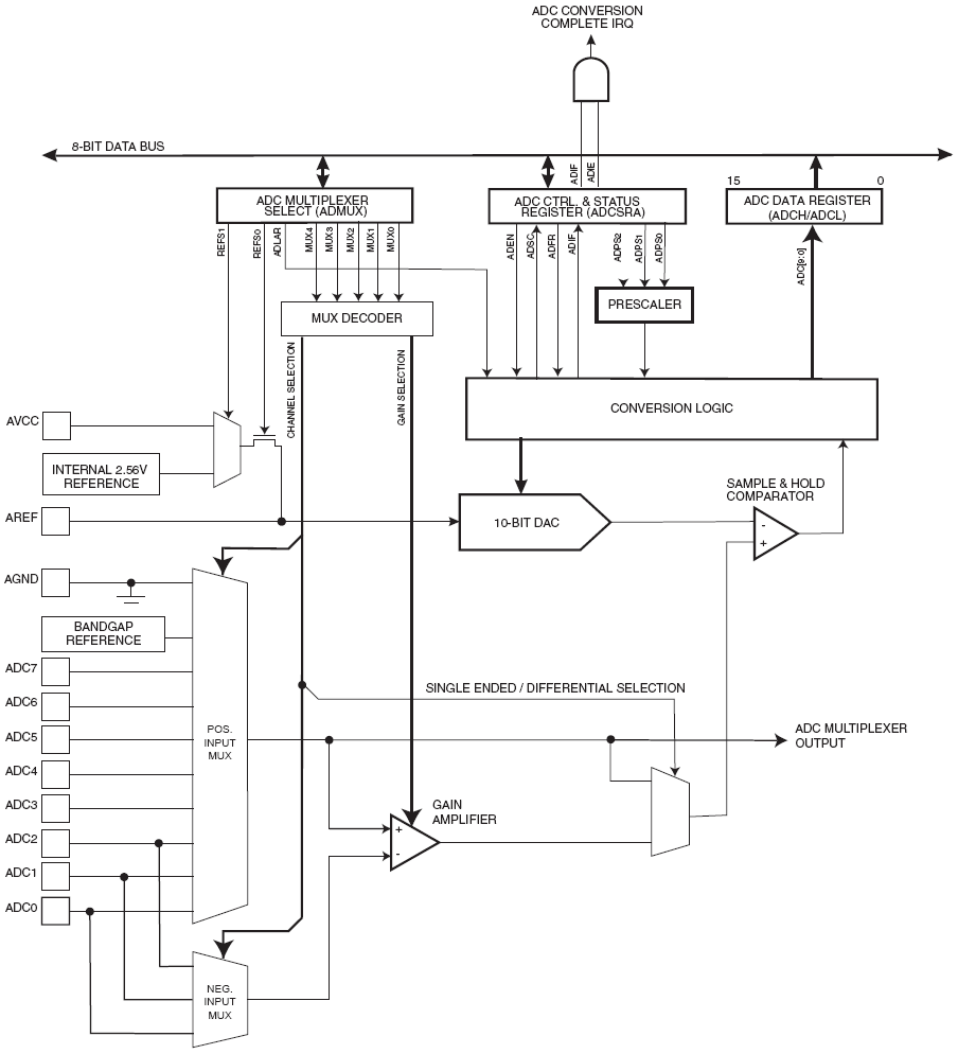
- Komparator jako przetwornik ADC (tylko w sytuacjach, gdy brak go w układzie)

$$V_C = E\left(1 - \exp\left(-t/RC\right)\right)$$

$$t = -RC\left(1 - \ln\left(\frac{V_C}{E}\right)\right)$$



Przetwornik ADC



- Przetwornik ADC (SAR):
  - Rozdzielczość 10-bit (**dopuszczalny błąd +/- 2 LSB**)
  - Czas konwersji w zakresie 13-260 us (76,9 – 15 kSPS)
  - 8 multipleksowanych wejść unipolarnych (odniesionych do poziomu AGND)
  - 7 wejść różnicowych
  - 2 wejścia różnicowe ze wzmocnieniem 10x (20dB) lub 200x (46dB) – spada jednak dokładność przetwarzania
  - Zakres napięć wejściowych 0 .. AVCC
  - Wewnętrzne źródło napięcia odniesienia 2,56V, ew. wartość AVCC
  - Tryb *Free Running* / *Single Conversion*
  - Przerwania po zakończeniu konwersji

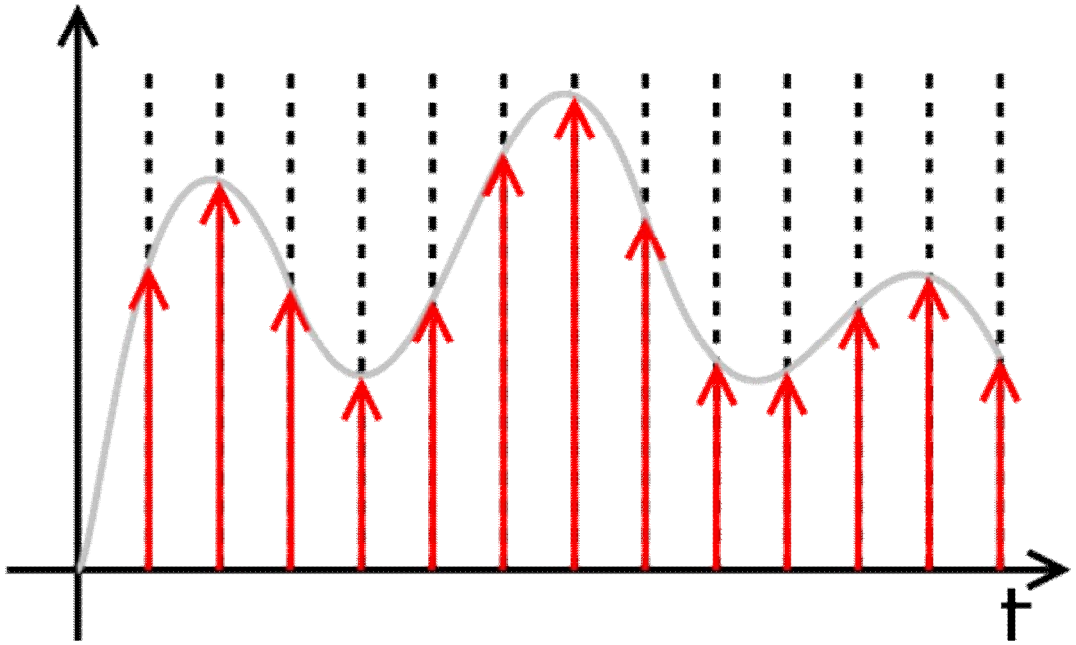
Przetwornik ADC

- próbkowanie – dyskretyzacja sygnału w czasie
- kwantowanie – dyskretyzacja wartości sygnału
- kodowanie uzyskanego sygnału dyskretnego

ELECTRICAL SYMBOL FOR ANALOG TO DIGITAL CONVERTER (ADC)

Próbkowanie

- dyskretyzacja argumentu funkcji x(t) – kolejne pobieranie próbek wartości sygnału w pewnych odstępach czasu w taki sposób aby ciąg próbek umożliwiał jak najwierniejsze odtworzenie całego przebiegu funkcji



<= dyskretyzacja w dziedzinie czasu

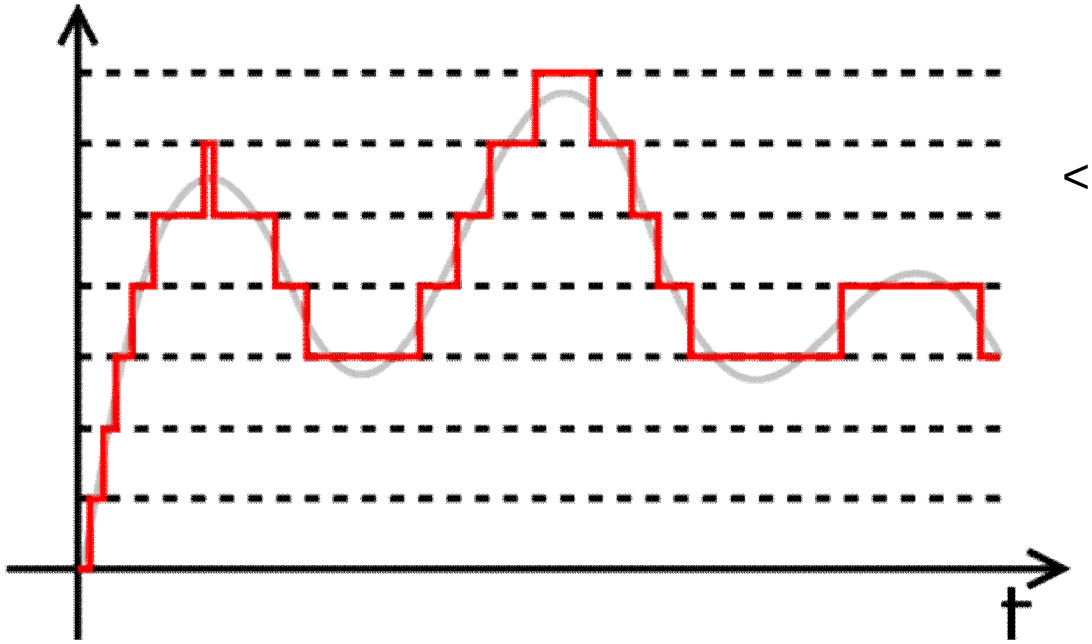
zwykle stosuje się próbkowanie okresowe, równomierne o okresie próbkowania T_S i częstotliwości próbkowania $f_S = 1/T_S$

Próbkowanie

- minimalna niezbędna częstotliwość próbkowania, przy której możliwe jest pełne odtworzenie sygnału ciągłego x(t) na podstawie pobranych próbek określa teoria Nyquista (Shannona-Kotielnikowa):
  - *przebieg dolnopasmowy jest ściśle określony przez próbki pobierane z częstotliwością co namjniej dwukrotnie większą od maksymalnej częstotliwości występującej w widmie próbkowanego przebiegu.*

Kwantyzacja

- dyskretyzacja wartości x(t) – przyporządkowanie każdej próbce skończonej liczby poziomów amplitudy, odpowiadających dyskretnym poziomom od zera do pełnego zakresu.

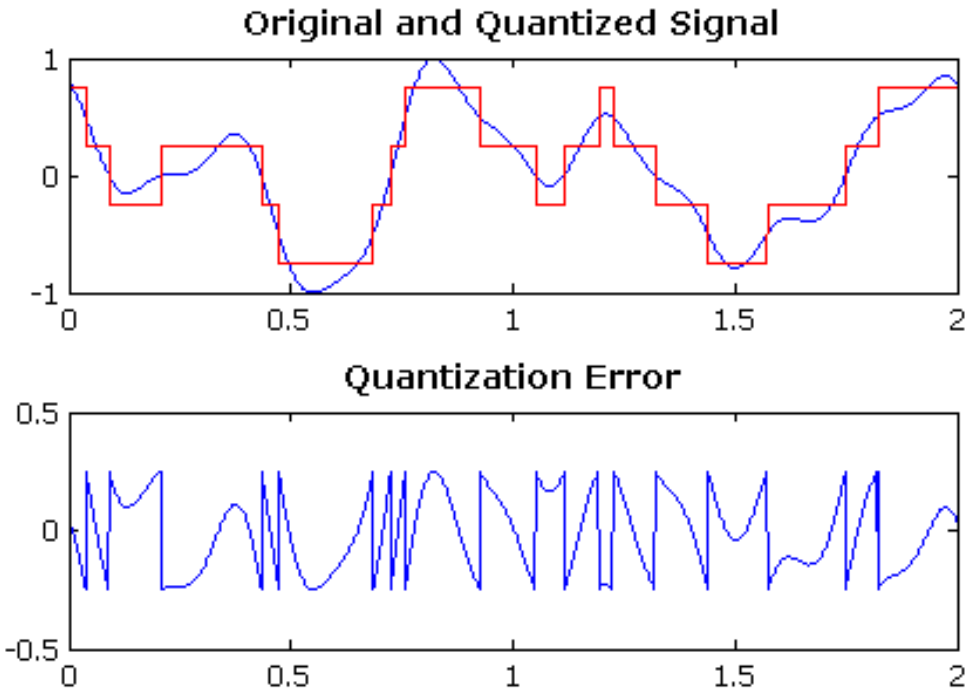


<= dyskretyzacja w dziedzinie amplitudy

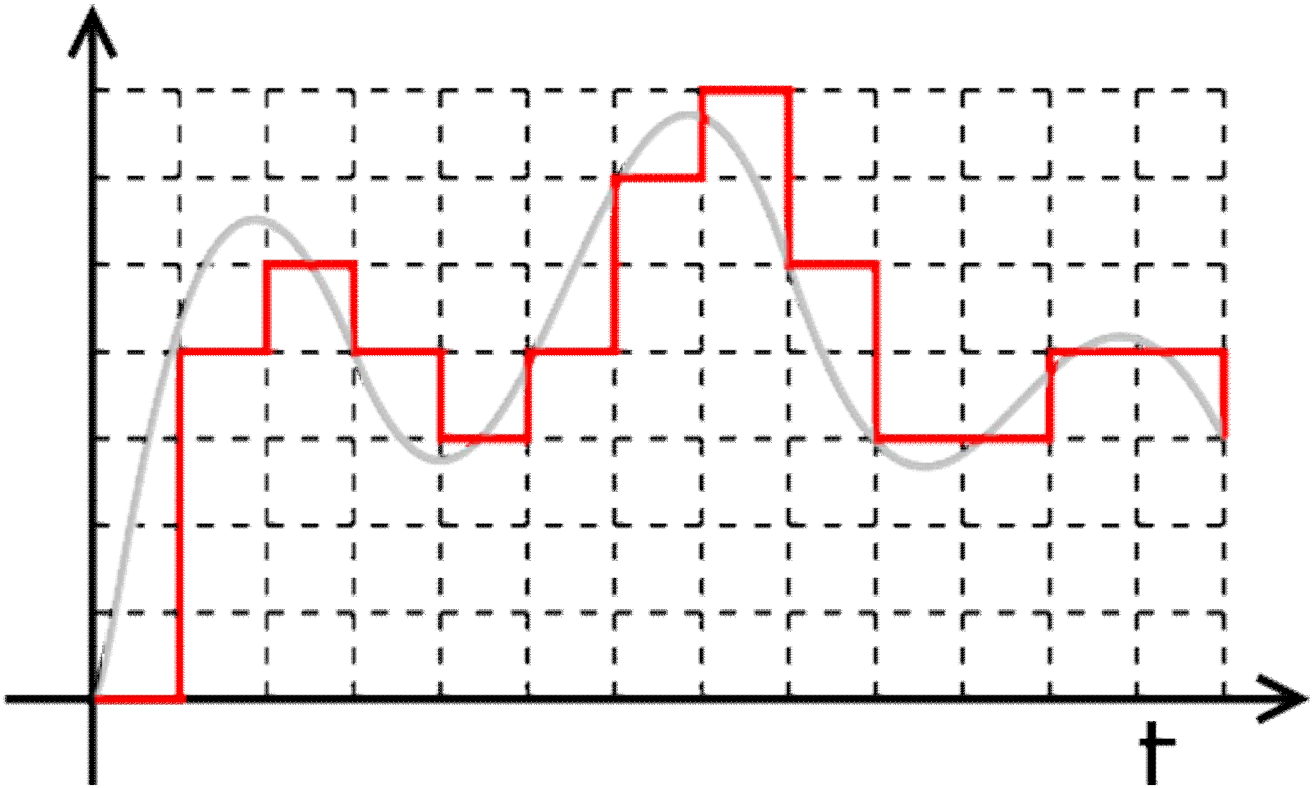
zwykle stosuje się kwantowanie równomierne w poziomach dyskretnych wynikających z rozdzielczości przetwornika oraz zakresu przetwarzania

Kwantyzacja

- błąd kwantyzacji – w układach z kwantowaniem równomiernym zawiera się zawsze w granicach +/- q/2, gdzie q jest rozdzielczością układu (najmniejszą wartością zmiennej X rozróżnialną przez układ kwantujący).



Sygnał cyfrowy

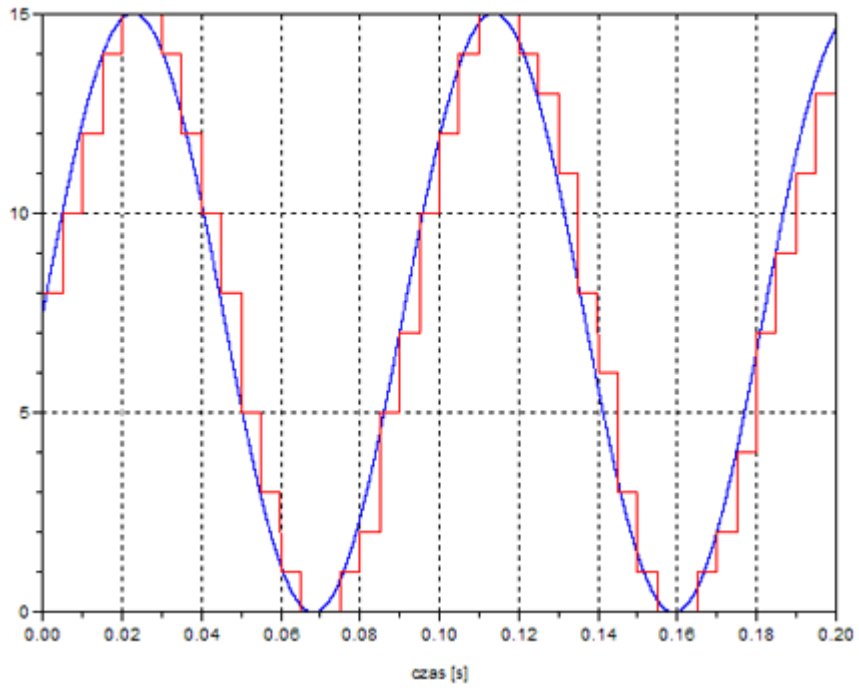


/\ dyskretyzacja w dziedzinie czasu oraz amplitudy /\

Kodowanie

- wartość dyskretyzowanej amplitudy zostaje wyrażona za pomocą odpowiednio dobranego kodu cyfrowego
  - kody unipolarne
    - naturalny kod dwójkowy
    - zapis dziesiętny kodowany dwójkowo (BCD)
  - kody komplementarne
  - kody bipolarne
    - zapis zank-moduł
    - przesunięty kod dwójkowy
    - zapis uzupełnień do 2 (U2)
    - zapis uzupełnień do 1 (U1)

Przetwornik ADC



- rozdzielczość: 4 bity
- częstotliwość próbkowania: 250 Hz
- kodowanie: NKB

Parametry przetworników ADC

- błąd kwantyzacji (ang. *quantization error)*
  - napięciu analogowemu (mogącemu przyjmować w zakresie przetwarzania nieskończenie wiele wartości) zostaje przyporządkowany cyfrowy sygnał N, ze zbiory skończonej liczby przedziałów kwantowania o wielkości q.
- nominalny pełny zakres przetwarzania (ang. *full scale range*)
  - wartość napięcia przetwarzanego $U_{FS} = q*2^n$ odpowiadająca wartości maksymalnej słowa wyjściowego powiększonej o 1.
- rzeczywisty zakres przetwarzania
  - wartość napięcia przetwarzanego $U_{FS} = q*(2^n – 1)$ odpowiadająca wartości maksymalnej słowa wyjściowego

Parametry przetworników ADC

- zakres dynamiczny przetwornika
  - stosunek pełnego nominalnego zakresu przetwarzania q*2^n do wartości przedziału kwantowania q

$$\left.\frac{U_{FS}}{q}\right|_{dB} = \left.\frac{2^n q}{q}\right|_{dB} = \left.2^n\right|_{dB} = 20\log 2^n = 6{,}02n$$

  - zakres dynamiczny wzrasta o ok. 6dB przy wzroście liczby bitów o jeden.

Parametry przetworników ADC

- błąd kwantyzacji można traktować jako szum związany z procesem przetwarzania A/C
- stosunek sygnał / szum kwantyzacji S/N wyrażany jest jako stosunek wartości maksymalnej sygnału równej 2^n*q do wartości skutecznej szumy kwantyzacji:

$$\left.\frac{S}{N}\right|_{dB} = 20\log\frac{2^n q}{\frac{q}{\sqrt{12}}} = 20\left(\log 2^n + \log\sqrt{12}\right) = 6{,}02n + 10{,}8$$

- wartość S/N wzrasta o ok. 6dB przy wzroście słowa wyjściowego przetwornika o jeden bit.

Parametry przetworników ADC

- rozdzielczość (zdolność rozdzielcza) – najmniejsza wartość sygnału wejściowego rozróżnialna przez przetwornik.
  - wyrażana w bitach:
    - np. 8-bit => 2^8 = 256 poziomów dyskretnych
  - wyrażona w woltach (wartość napięcia odpowiadająca 1LSB): $$q = \frac{U_{FS}}{2^n}$$
- Rozdzielczość najczęściej podawana jest jako liczba bitów *n* słowa wyjściowego przetwornika.

Parametry przetworników ADC

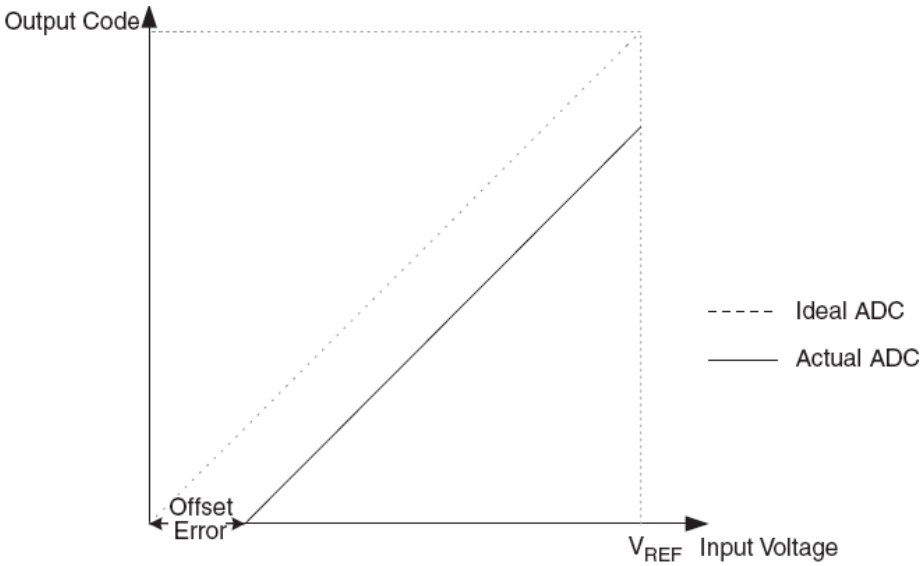
- rozdzielczość przetwornika A/C stanowi granicę jego dokładności (wynika ona z samej istoty procesu kwantowania),
- w prawidłowo zaprojektowanym układzie A/C liczba bitów słowa wyjściowego jest tak dobrana, że wartość błędu analogowego (wyrażona względnie bądź bezwzględnie) jest mniejsza od błędu kwantyzacji,
- zwiększanie zdolności rozdzielczej ponad tą granicą jest niecelowe (nie poprawia dokładności przetwarzania)
- w normalnym przypadku wartość zdolności rozdzielczej powinna określać też dokładność.

Parametry przetworników ADC

- czas przetwarzania – czas potrzebny do jednego całkowitego przetworzenia na wielkość cyfrową, z określoną rozdzielczością, sygnału analogowego o wartości równej pełnemu zakresowi przetwarzania,
- częstotliwość przetwarzania – maksymalna częstotliwość, z jaką mogą odbywać się kolejne przetworzenia sygnału z zachowaniem określonej rozdzielczości i dokładności w pełnym zakresie,
- szybkość bitowa – liczba bitów rezultatu przetwarzania uzyskiwana w jednostce czasu.

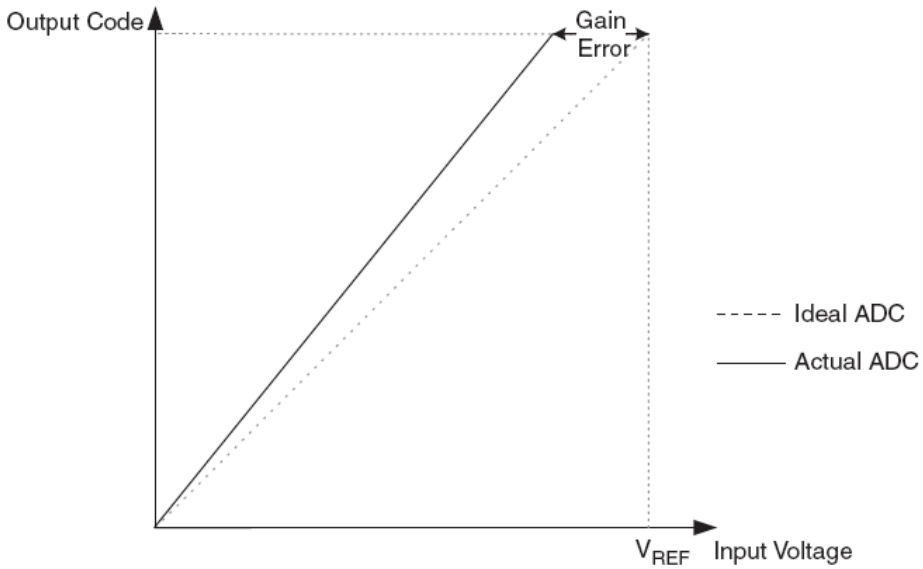
Błędy przetwornika ADC

Błąd przesunięcia zera (ang. *offset error*)



The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB

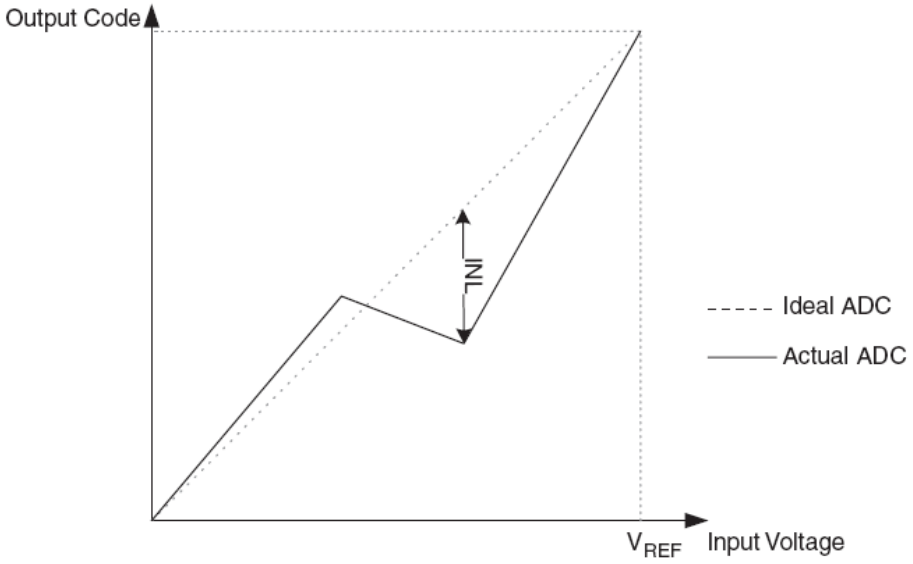
Błąd wzmocnienia lub skalowania (ang. *gain error*)



After adjusting for offset, the gain error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

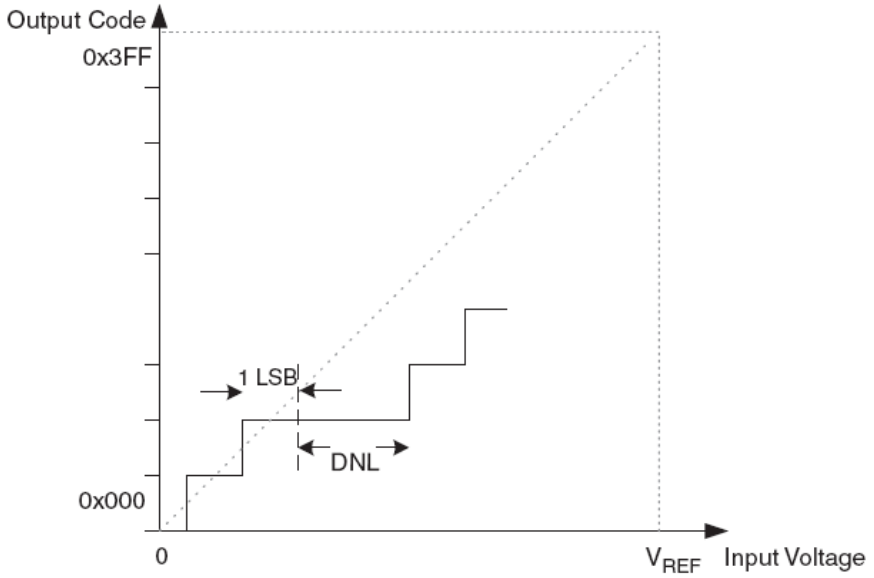
Błędy przetwornika ADC

Nieliniowość całkowa
(ang. *Integral Non-linearity (INL)*)



After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB

Nieliniowość różniczkowa
(ang. *Differential Non-linearity (DNL)*)



The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB

Błędy przetwornika ADC

- Dokładność bezwzględna (ang. *absolute accuracy*):
  - określana jako różnica między teoretyczną i rzeczywistą wartością napięci U_{IN}, powodującą powstanie na wyjściu określonej wartości cyfrowej.
- Dokładność względna (ang. *relative accuracy*)
  - dokładność bezwzględna odniesiona do pełnego nominalnego zakresu przetwarzania i wyrażona w procentach lub częściach wartości najmniej znaczącego bitu.
- Wartości obu dokładności są uwarunkowane przede wszystkim przez nieliniowość całkową i różniczkową, błąd przesunięcia zera i błąd skalowania oraz błąd kwantyzacji.

Błędy przetwornika ADC w ATmega128(L)

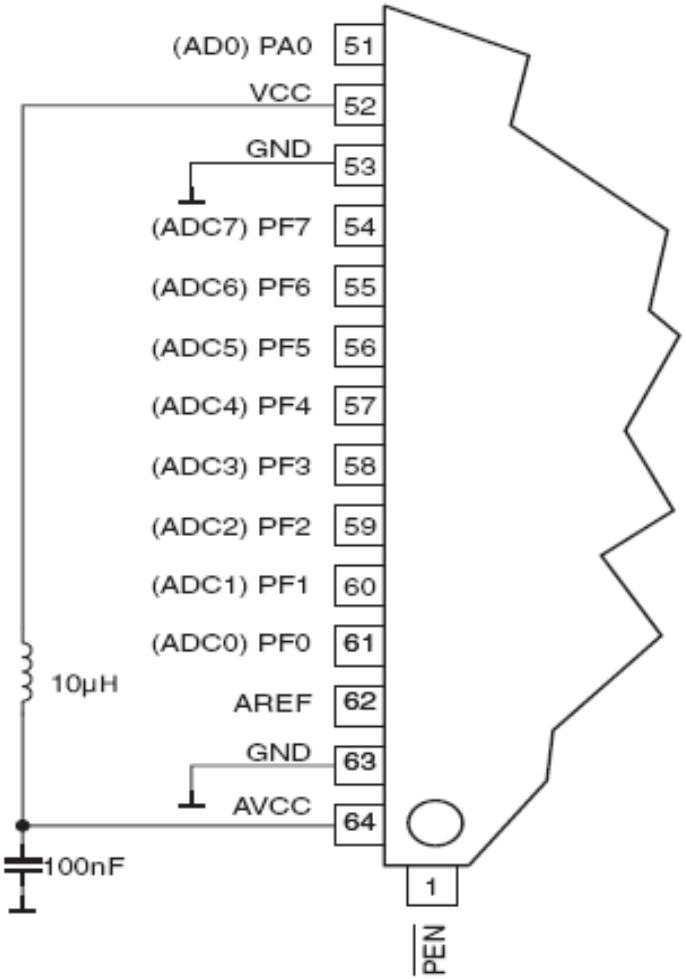
- podsumowane w rozdziale:
  - Electrical Characteristics -> ADC Characteristics

	Resolution	Single Ended Conversion			10	Bits
	Absolute Accuracy (Including INL, DNL, Quantization Error, Gain and Offset Error)	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz		1.5		LSB
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 1 MHz		3.25		LSB
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz Noise Reduction mode		1.5		LSB
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 1 MHz Noise Reduction mode		3.75		LSB

tryb pracy single-ended

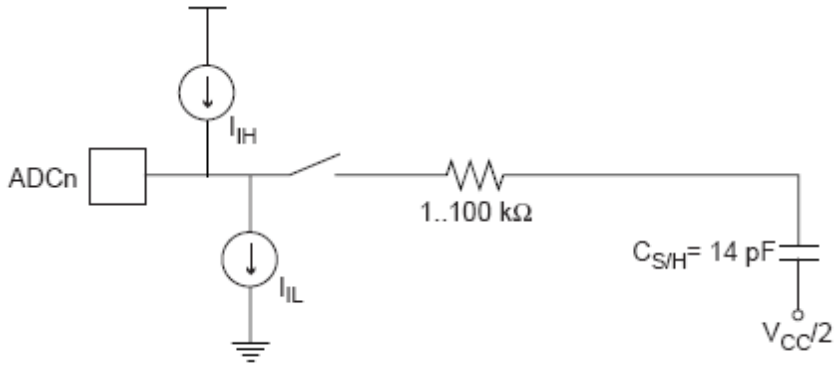
Przetwornik ADC

- Wyprowadzenia przetwornika
  - AVCC – napięcie zasilające część analogową
  - AREF – napięcie odniesienia dla przetwornika, końcówka pozwala jednak na odkłócenie dowolnego napięcia odniesienia
  - ADC0 .. ADC7 – wejścia pomiarowe



Przetwornik ADC

- Model końcówki ADCn układu
  - Impedancja zastępcza widziana z zewnątrz to rezystancja szeregowa (1 .. 100 kOhm) oraz równoległa pojemność na poziomie 14pF
  - Tworzy to w najgorszym przypadku filtr dolnoprzepustowy o cz. odcięcia na poziomie 110kHz
  - Możliwie mała rezystancja wewnętrzna źródła sygnału mierzonego

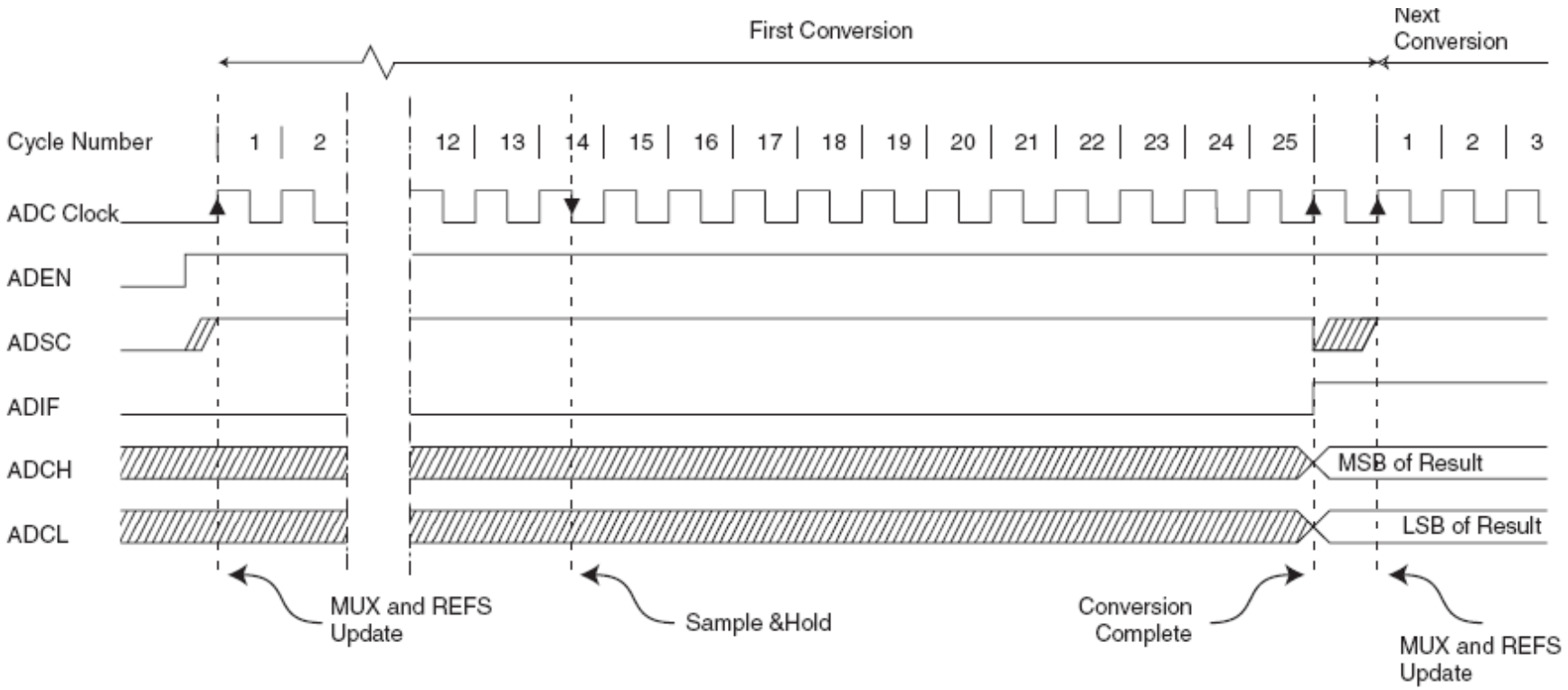


Przetwornik ADC

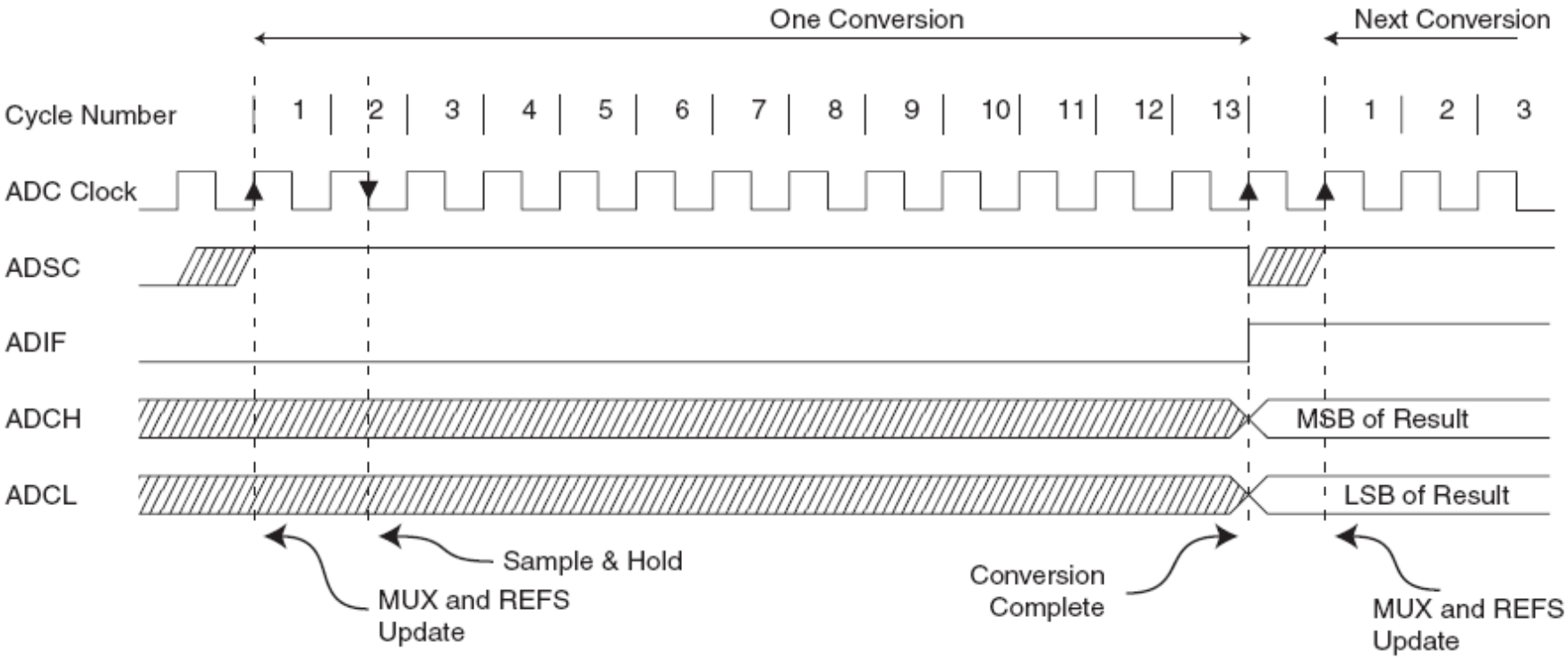
- Tryb *Free Running*
  - Przetwornik nieustannie dokonuje konwersji, której ostatni wynik znajduje się zawsze w rej. ADC Data Register,
  - Konieczność manualnego wyzwolenia pierwszej konwersji (ADSC w rej. ADCSRA).

- Tryb *Single Conversion*
  - Manualne wyzwalanie konwersji ADC (ADSC w rej. ADCSRA),
  - Wyzerowanie bitu ADSC (lub przerwanie) oznacza koniec konwersji.

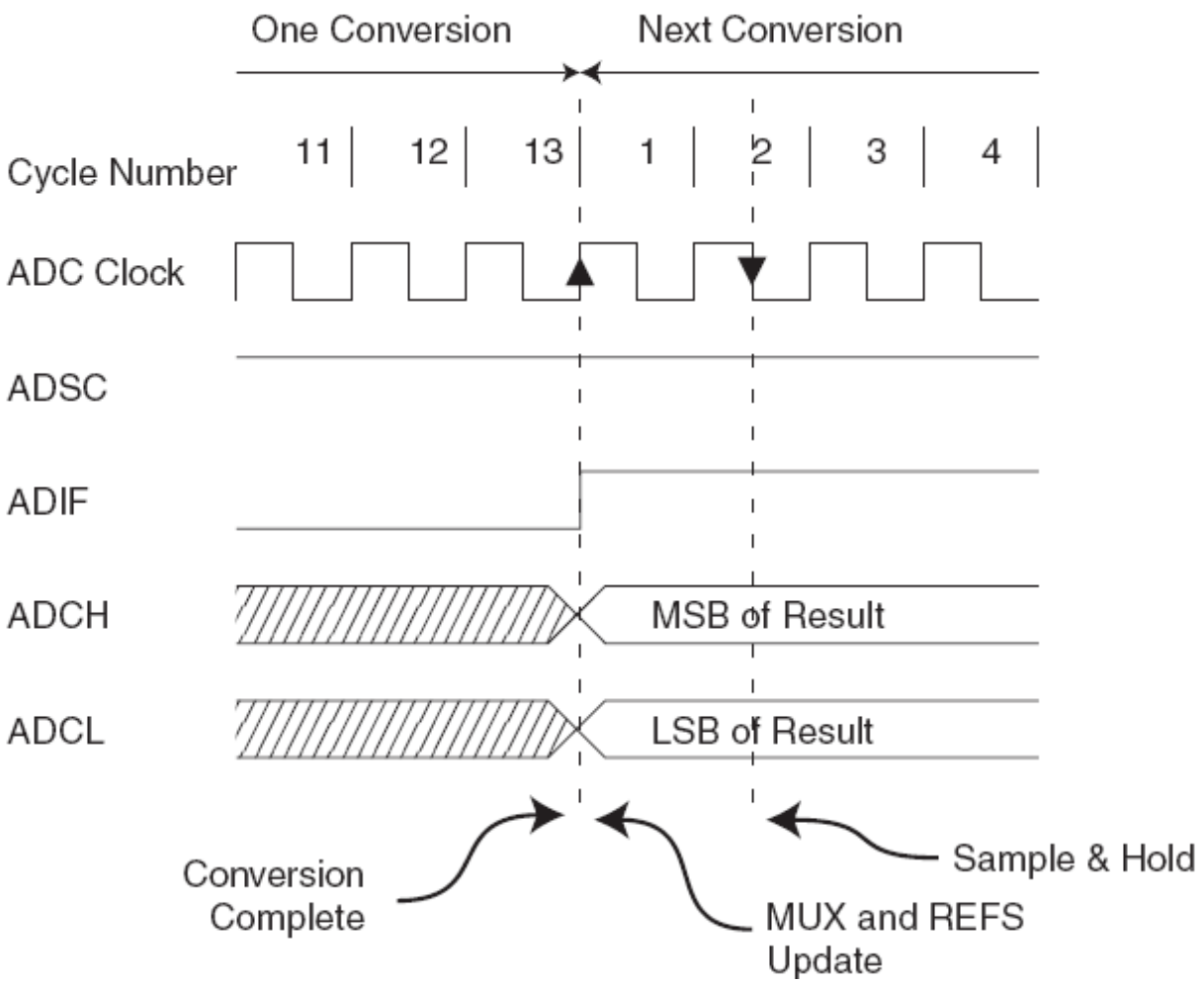
ADC Timing Diagram, First Conversion (Single Conversion Mode)



ADC Timing Diagram, Single Conversion



ADC Timing Diagram, Free Running Conversion



Przetwornik ADC – wynik konwersji

- Single Ended Conversion:

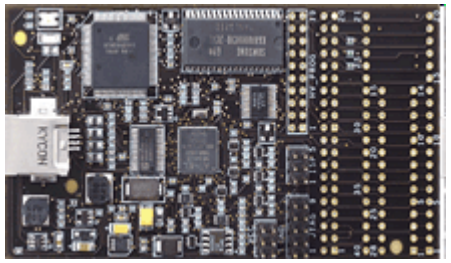
$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

- Differential Measurement:

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

Programowanie układów rodziny AVR

- Narzędzia f. Atmel Corporation
  - AVR Dragon
    - supports all programming modes, complete emulation for devices with 32kB or less Flash memory
  - AVR ISP mkII
    - [illegible]
  - AV[illegible]
    - JTAG for all 32-bit AVR32 MCU/DSP and JTAG/debugWIRE for all 8-bit AVR MCUs
  - AVR ONE!
    - programming / debugging of all AVR, AVR XMEGA and AVR32 devices
  - ATSTK500, ATSTK600, ATSTK1000



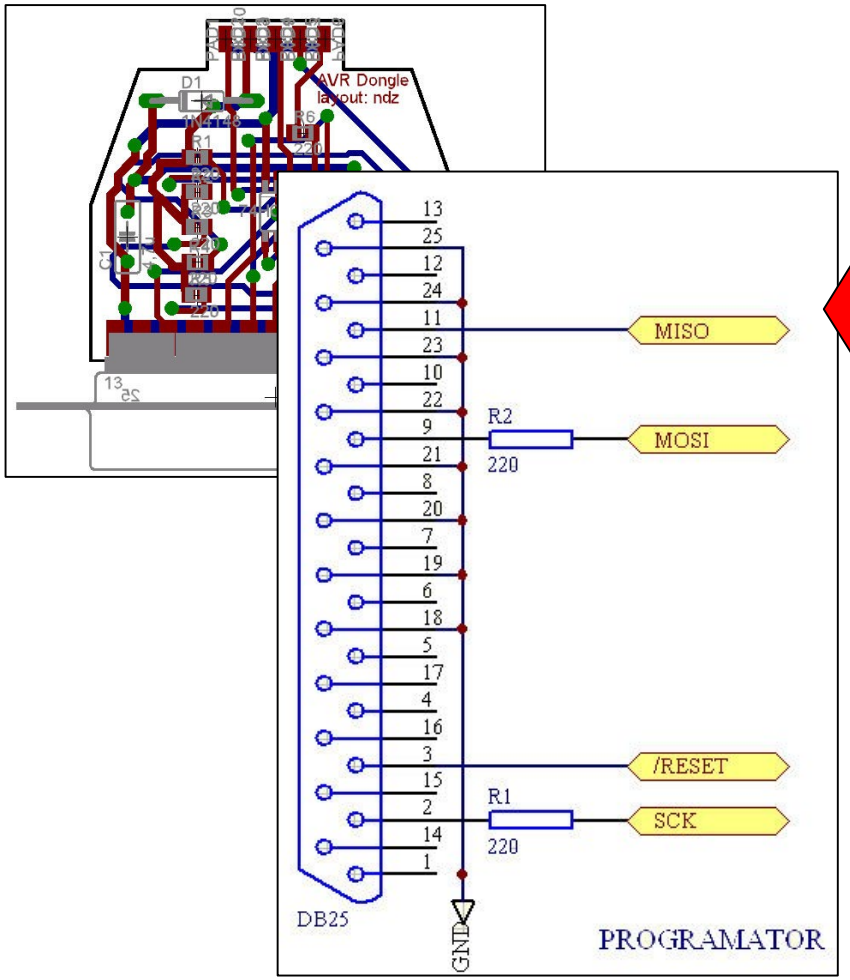
zaleta: **integracja z AVRStudio, możliwość pracy z linii komend, współpraca z avr-dude**

wada: **cena…**

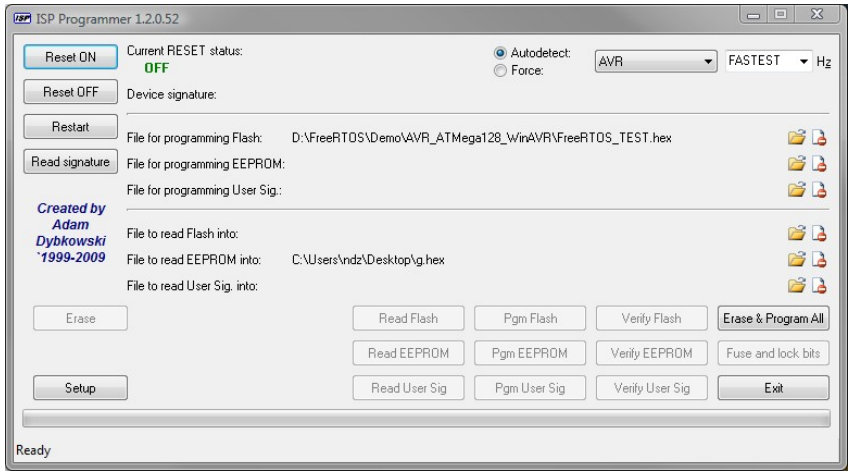
Programowanie układów rodziny AVR

- Programatory alternatywne
  - programator LPT (w różnych wydaniach)
    - MIPSA PROG
  - programatory COM
    - AVRprog
  - programatory USB
  - bootloadery
    - AVR109
    - MegaLoad

Programator LPT MIPSA PROG

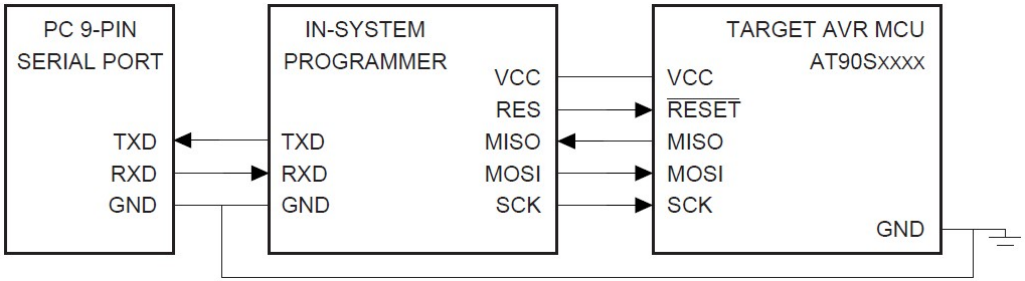


- Najprostsza wersja programatora – niezawodna ale nie buforowana!!!
- Całość zamknąć można we wtyczce DB25 (złącze równoległe LPT).



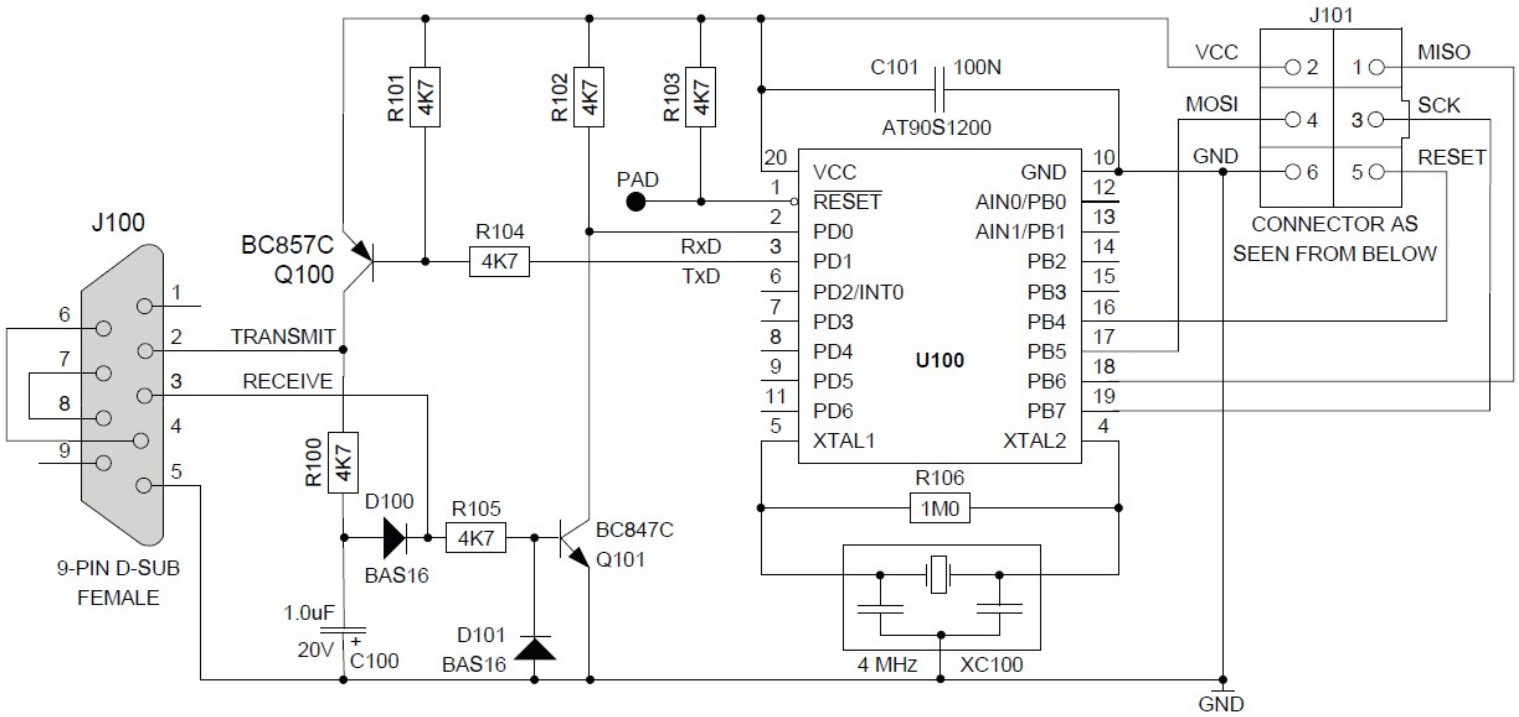
współpraca z ISP Programmer (http://dybkowski.net/isp) lub avr-dude

AVRprog (AN: AVR910)



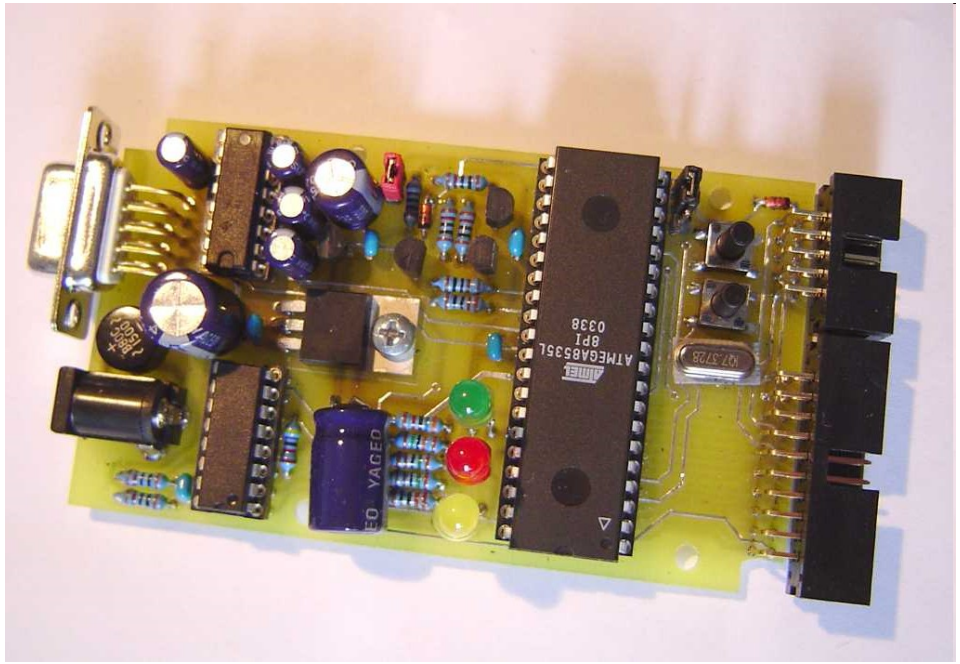
- opisany w nocie aplikacyjnej AVR910
- współpraca z AVRStudio

- brak konieczności posiadania porty LPT



HVprog

- kompatybilny z STK500
- współpraca z AVRStudio
- wsparcie dla wszystkich kontrolerów AVR
- możliwość programowania równoległego (HVP)
- dostępna pełna dokumentacja, możliwość wyposażenia w USB (konwerter FTDI)



http://www.der-hammer.info/hvprog/index_en.htm

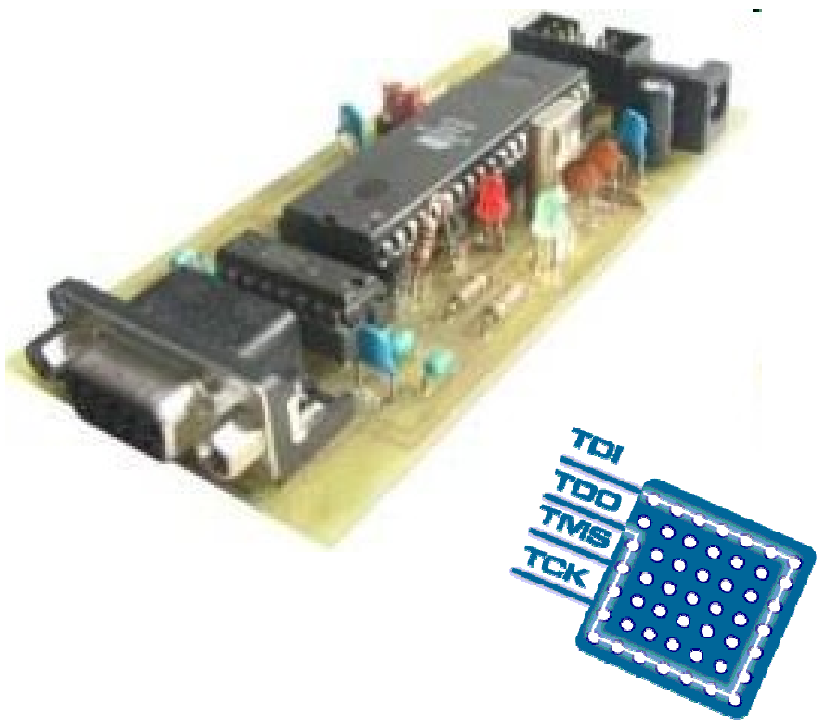
USBasp

- dostępny firmware, schematy oraz wzory PCB
- szybkość (programowanie do 5kB/s)
- współpraca z avr-dude
- alternatywnie GUI (Khazama AVR Programmer / eXtreme Burner)

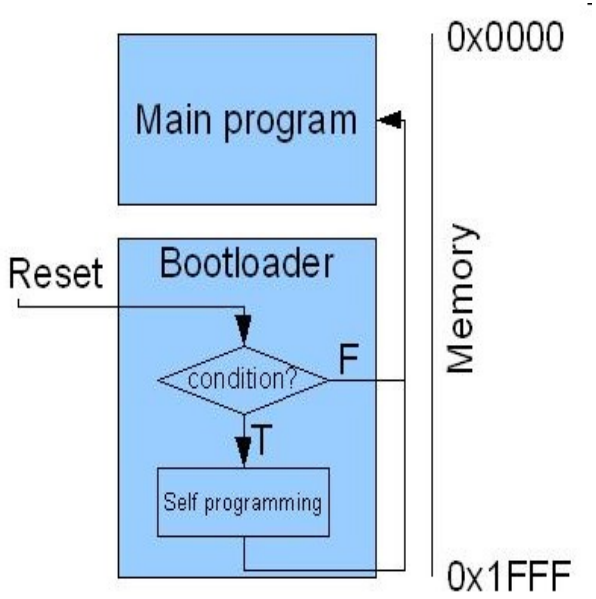
http://www.fischl.de/usbasp/

JTAG TWICE (JTAG ICE + STK500)

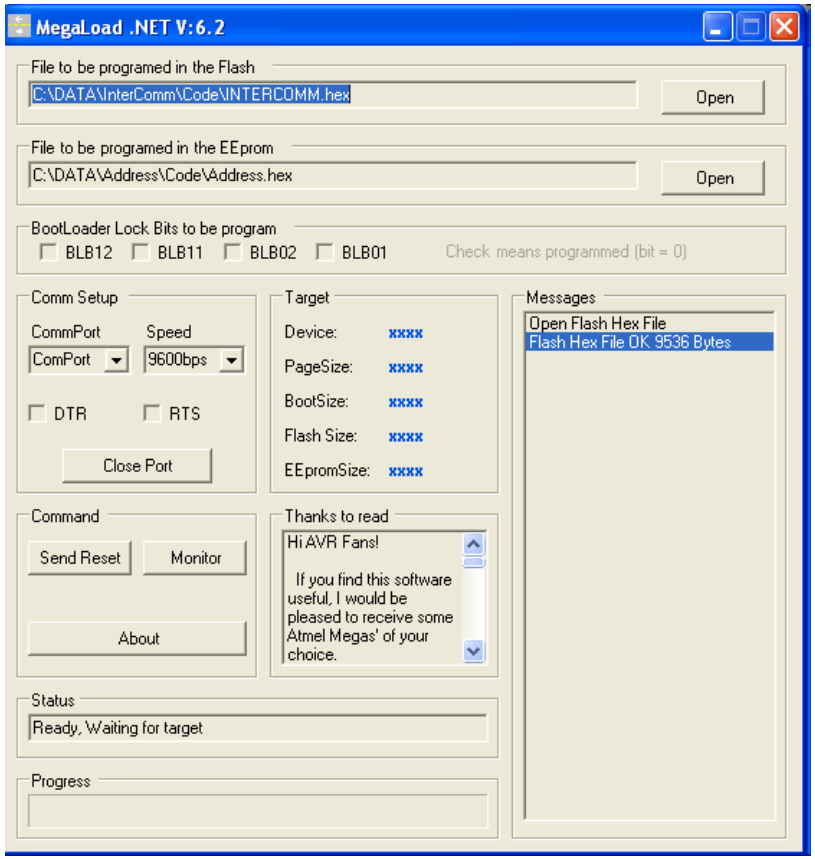
- kompatybilny z JTAG ICE
- kompatybilny z STK500 v2
- automatyczne przełączanie funkcji
- interfejs RS232
- możliwość upgrade oprogramowania



Bootloader: AVR109 i MegaLoad

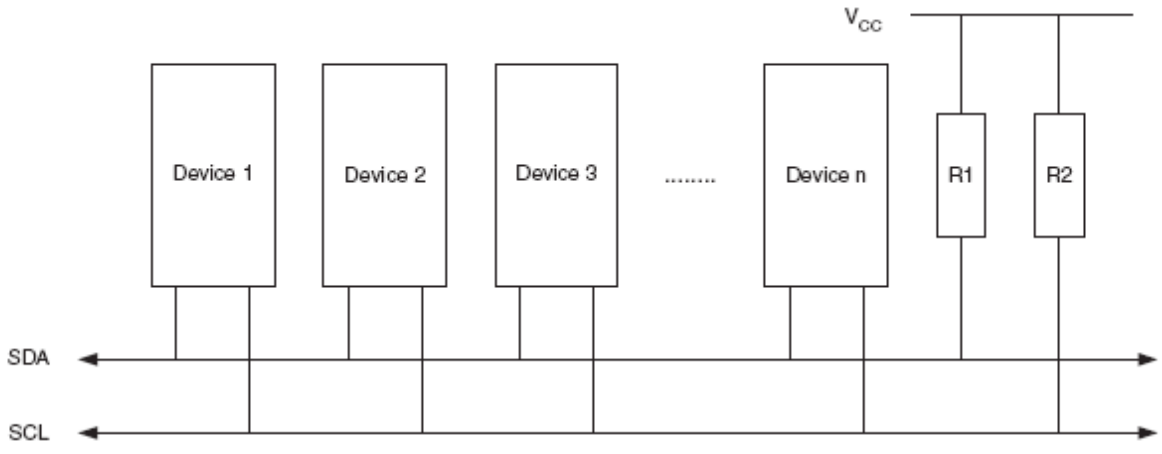


- AVR109 opisany w nocie aplikacyjnej
- ładowanie programu przez port szeregowy
- implementacja dla większości mikrokontrolerów rodziny AVR



http://www.microsyl.com/megaload/megaload.html

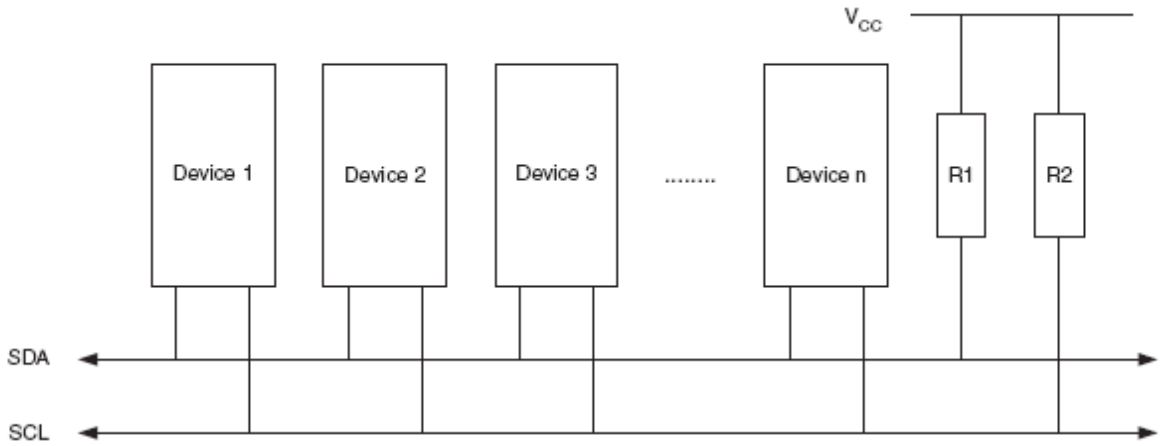
Interfejs *Two-wire Serial Interface* (TWI)



TWI Bus Interconnection

- TWI jest implementacją magistrali I2C dla mikrokontrolerów produkowanych przez f. Atmel.
- Właściwości:
  - Komunikacja dwuprzewodowa,
  - Tryb komunikacji Master oraz Slave, Transmitter oraz Receiver możliwy w obu konfiguracjach
  - Adresowanie 7-bit (max. 128 różnych urządzeń Slave)

Interfejs *Two-wire Serial Interface* (TWI)



TWI Bus Interconnection

- Właściwości (c.d.):
  - Tryb pracy Multi-Master z arbitrażem
  - Prędkość pracy do 400kbps,
  - Programowalny adres Slave, obsługa General Call,
  - Rozpoznanie własnego adresu na magistrali może wyprowadzić układ ze Sleep Mode.

Szybkość komunikacji TWI

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

- TWBR = Value of the TWI Bit Rate Register
- TWPS = Value of the prescaler bits in the TWI Status Register

Bit	7	6	5	4	3	2	1	0	
	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0	TWBR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

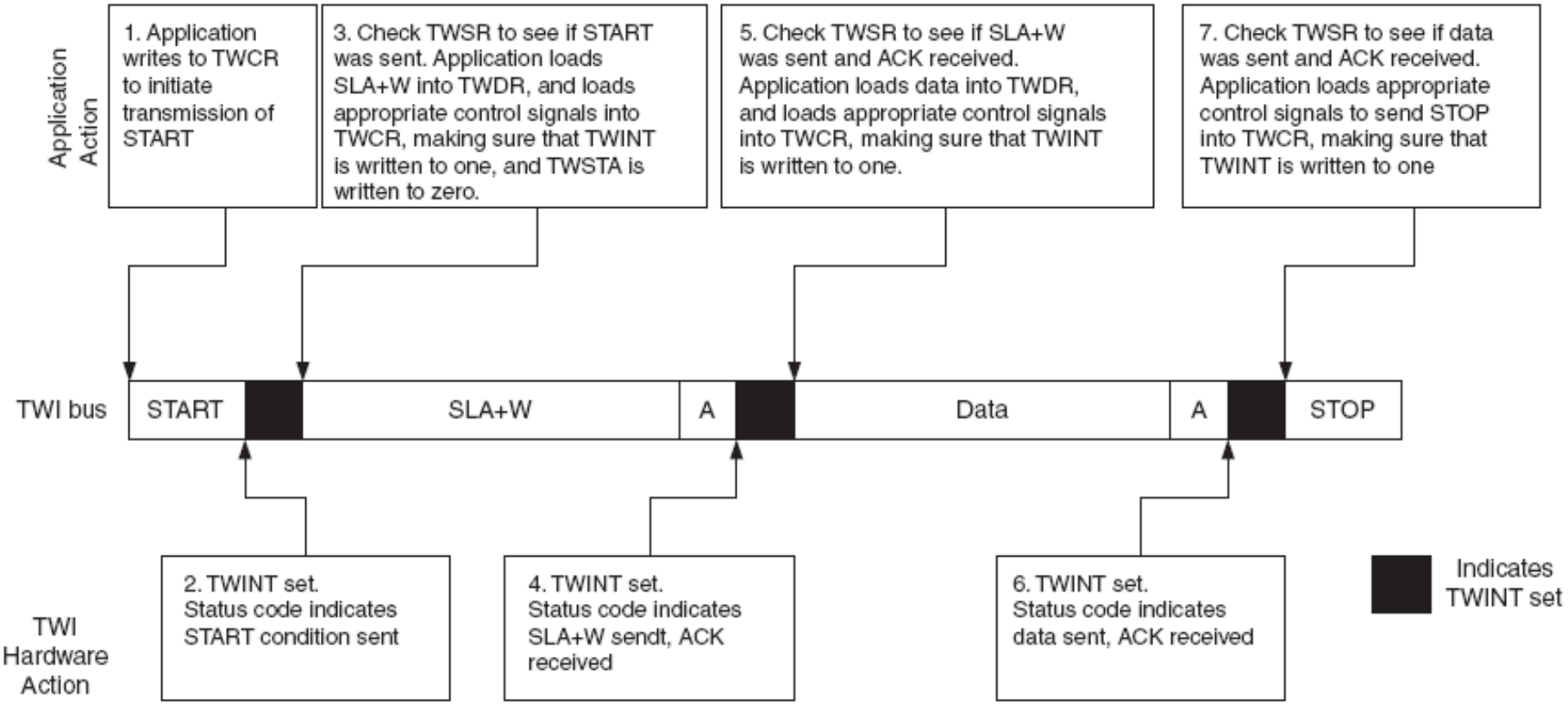
TWI Bit Rate Register

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

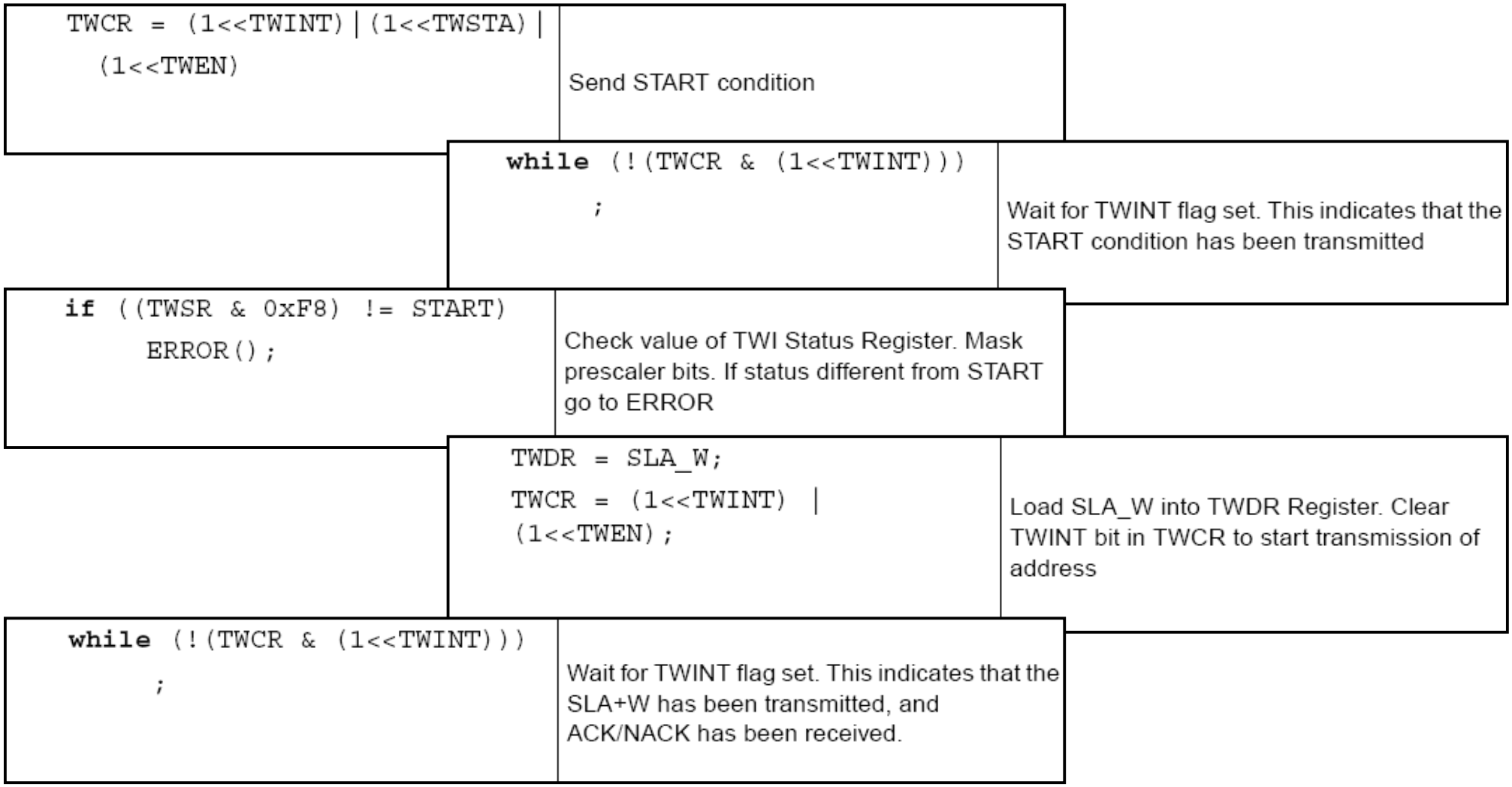
TWI Status Register

Zasada działania

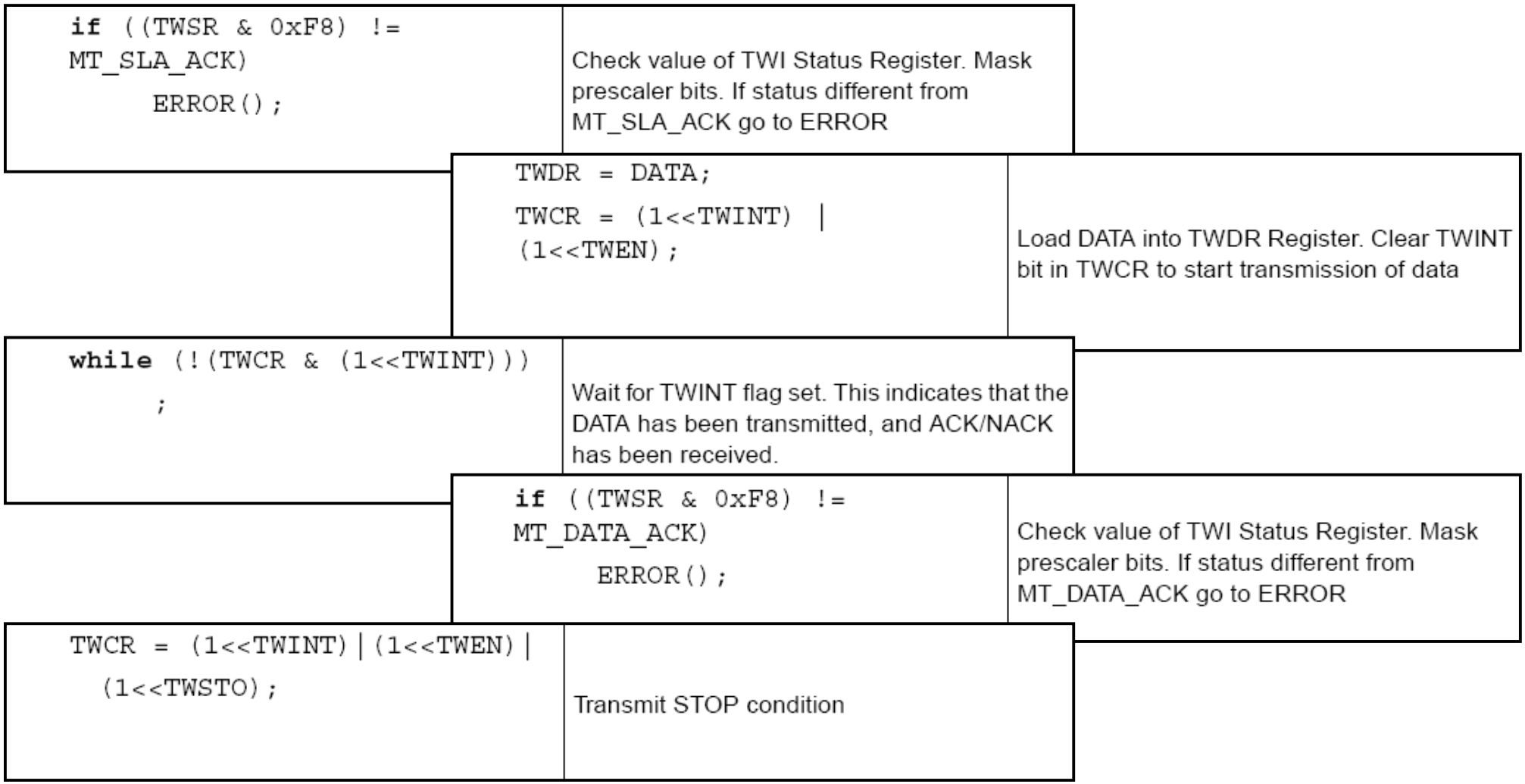
Figure 95. Interfacing the Application to the TWI in a Typical Transmission



TWI bez przerwań – *pooling*



TWI bez przerwań – *pooling*

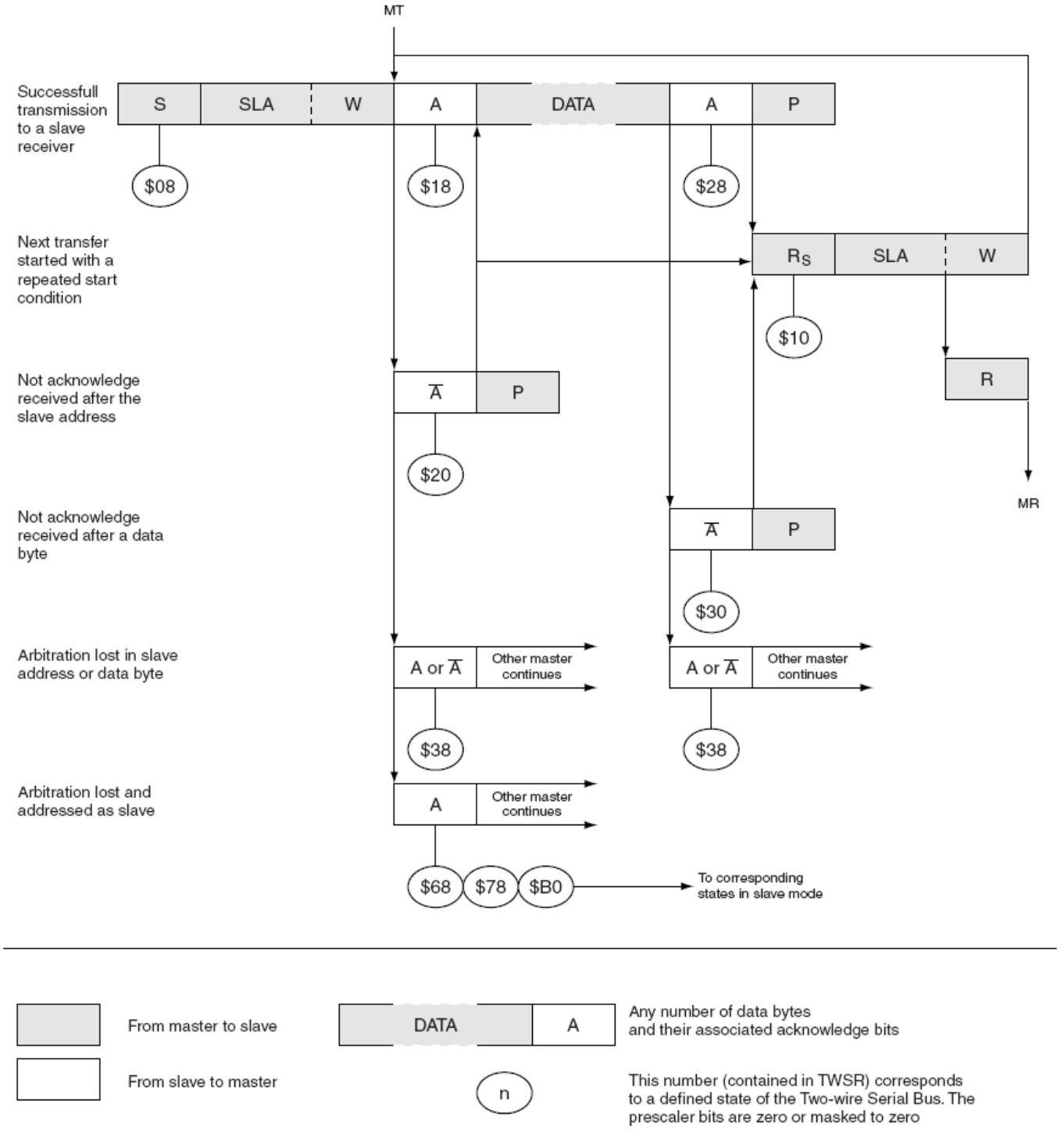


TWI bez przerwań – *pooling*

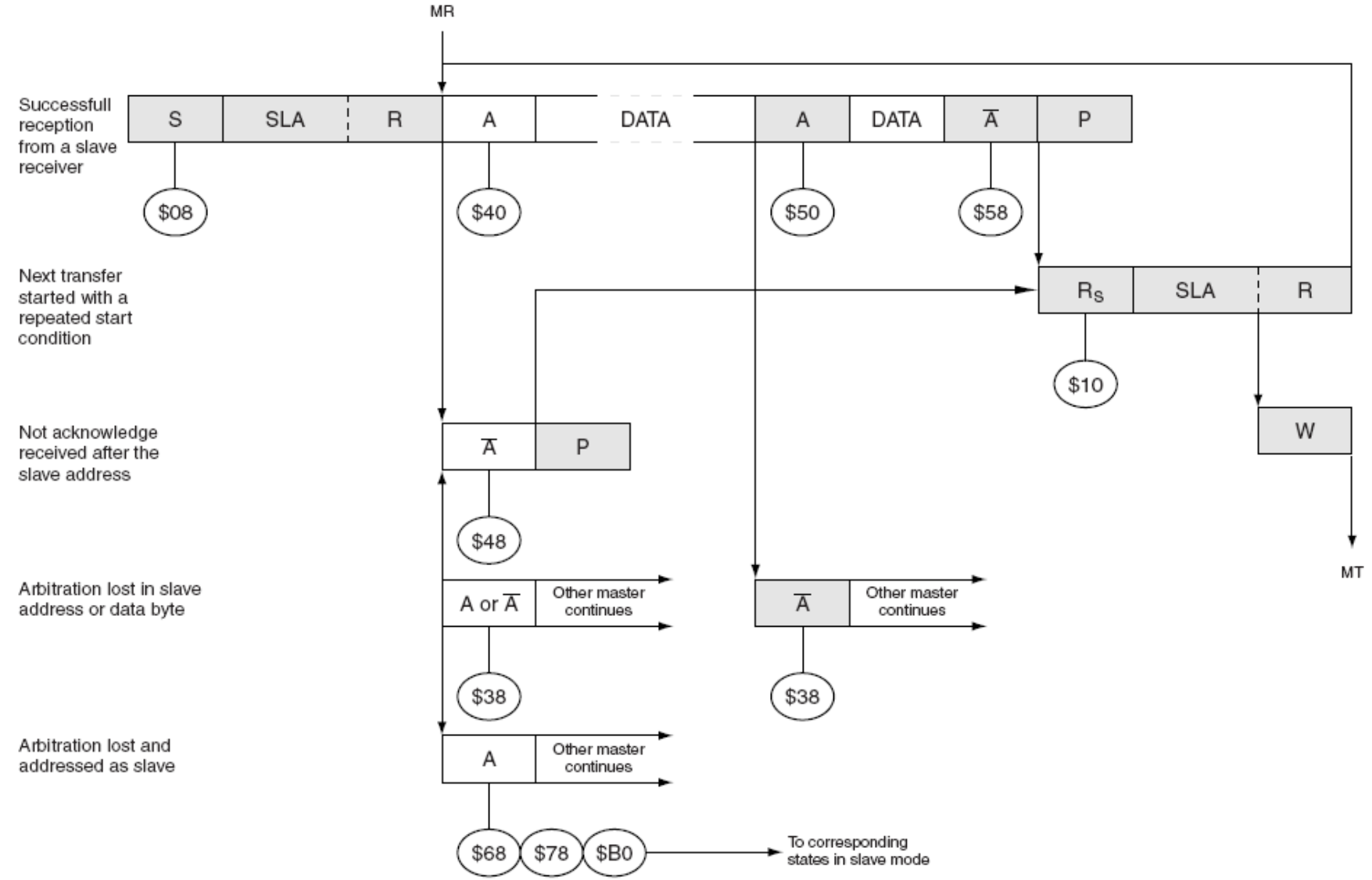
- Przykład komunikacji z pamięcią EEPROM serii 24Cxx – avr-libc examples
- SAMODZIELNA ANALIZA

- Przykład biblioteki – używana na laboratorium biblioteka Perera Fleury – i2cmaster
- SAMODZIELNA ANALIZA

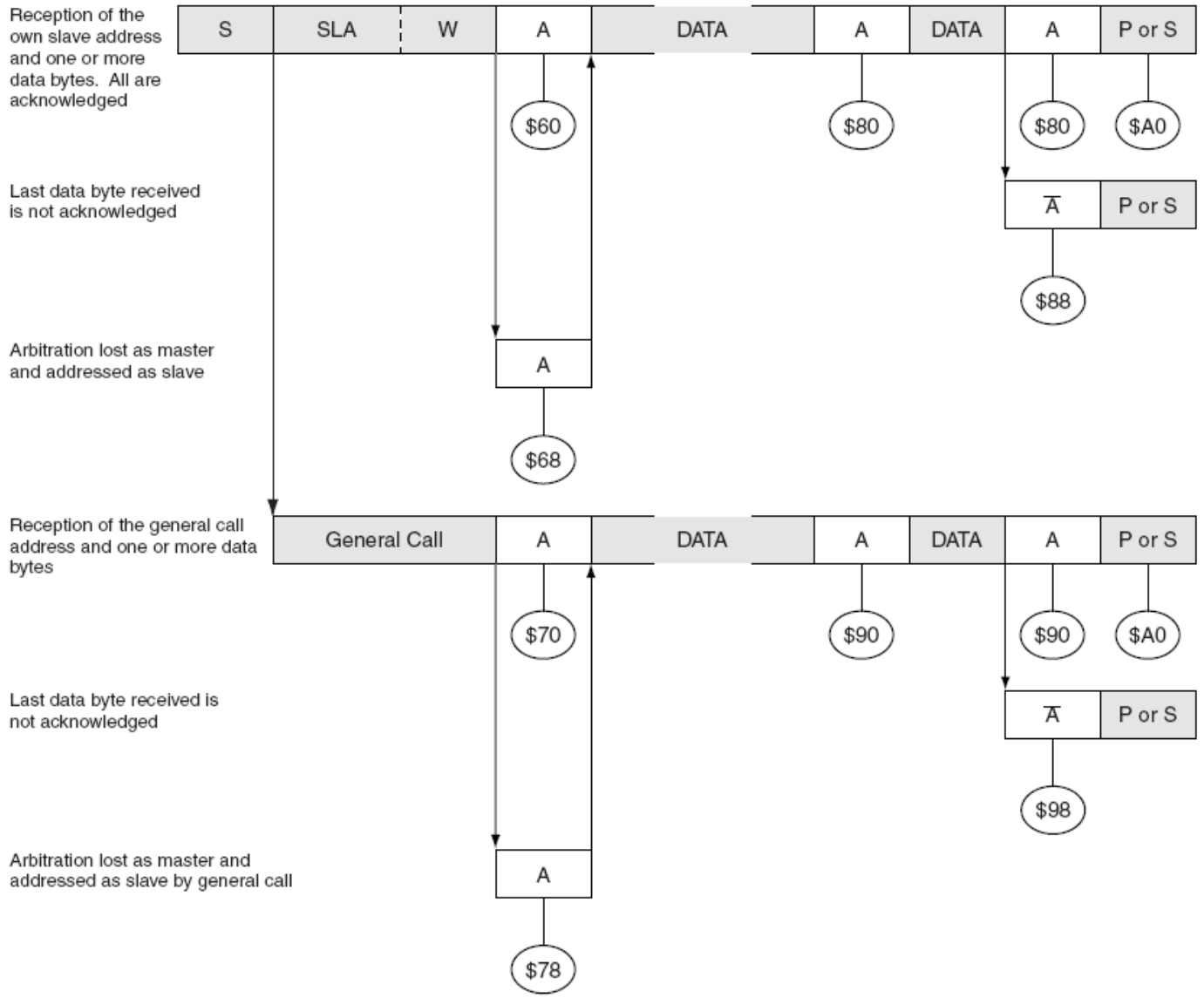
Master Transmitter Mode



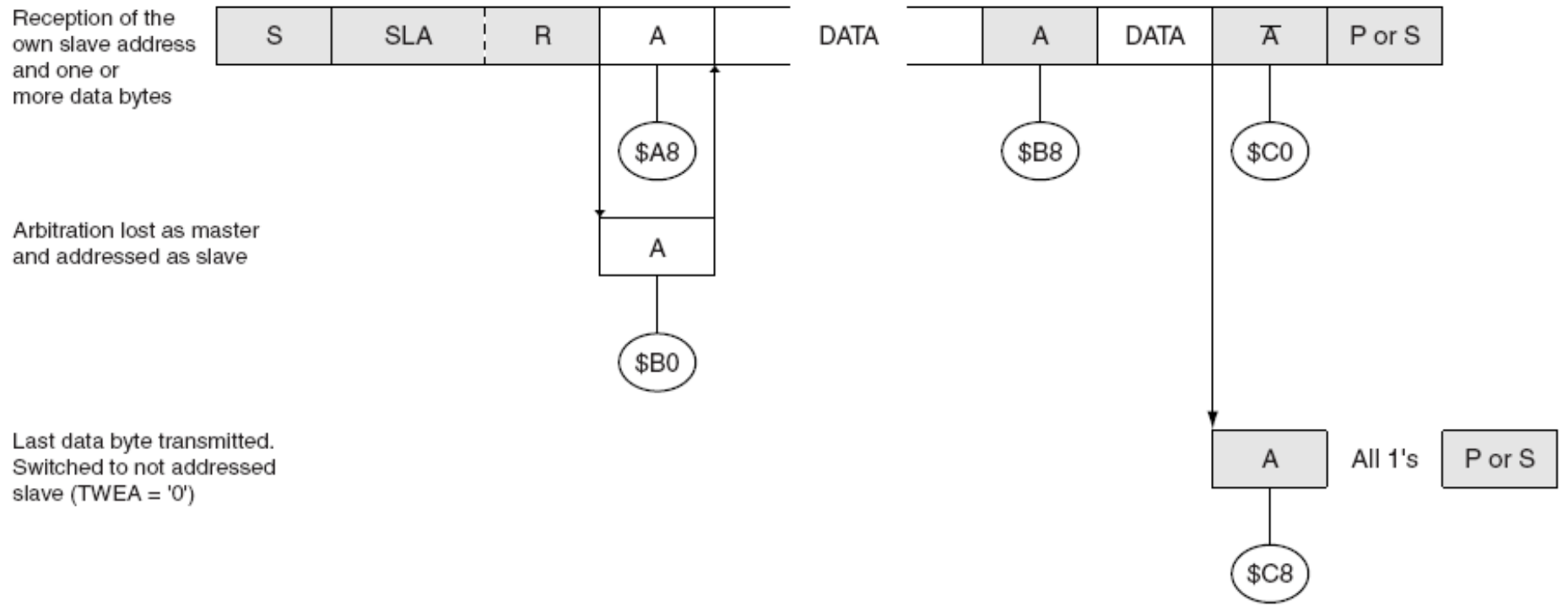
Master Receiver Mode

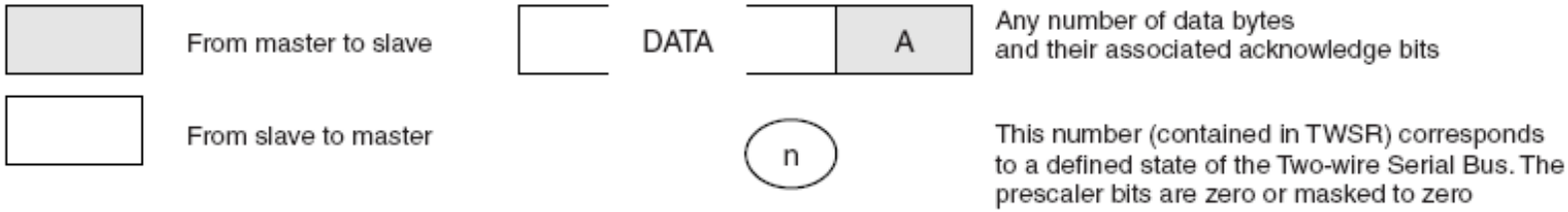


Slave Receiver Mode



Slave Transmitter Mode





TWI - przerwania

- Jedno przerwania bloku TWI, w którym należy kontrolować cały proces transmisji.

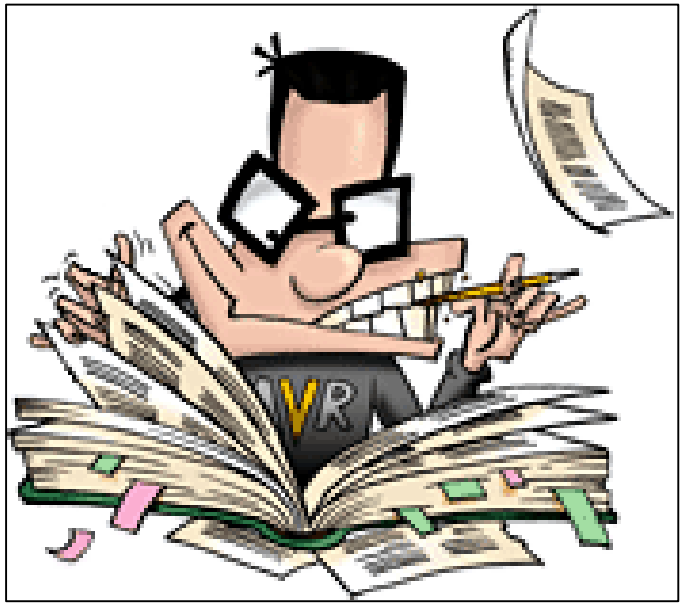
```
//! I2C (TWI) interrupt service routine
ISR(TWI_vect) {
    // read status bits
    uint8_t status = TWSR & TWSR_STATUS_MASK;

    switch(status) {
          // Master General
          case TW_START:      // 0x08: Sent start condition
          case TW_REP_START:  // 0x10: Sent repeated start condition
                    // (...)
          break;

          case TW_MT_SLA_ACK: // 0x18: Slave address acknowledged
          case TW_MT_DATA_ACK:          // 0x28: Data acknowledged
                    // (...)
          break;

          //(...)
    }
```

Pytania?



C.D.N.

Politechnika Łódzka

Instytut Elektroniki