

8-bitowa kontrofensywa (4)

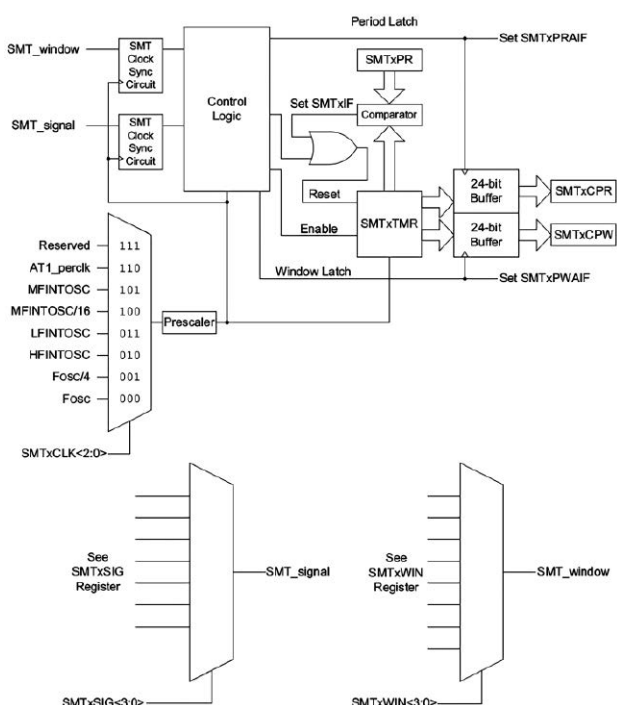
Układy licznikowe i generatory w mikrokontrolerach PIC16

Wydawać by się mogło, że mikrokontrolery 8-bitowe odeszły już do lamusa. Nic bardziej mylącego! Firma Microchip stale pracuje nad tym, aby podnieść ich atrakcyjność i zachęcić konstruktorów do korzystania z jej produktów. Wszak nie zawsze trzeba „wytaczać armatę na muchę”.

Układy licznikowe są wbudowane w mikrokontrolery „od zawsze”. Typowe liczniki/timery są stale rozwijane: zwiększa się pojemność rejestrów licznika, rozbudowywane są mechanizmy wyzwiania zliczania, wprowadzana liczne powiązania z innymi modułami peryferyjnymi. W mikrokontrolerach PIC16F1xx wprowadzono kilka unikalnych modułów licznikowych, które trudno spotkać (przynajmniej na razie) u innych producentów.

Licznik SMT – Signal Measurement Timer

Moduł SMT jest 24-bitowym licznikiem konfigurowanym do pomiarów szerokości impulsu, częstotliwości, współczynnika wypełnienia, różnicy czasu pomiędzy zboczami dwóch sygnałów. Schemat blokowy tego licznika pokazano na **rysunku 1**. Ma on 3 wejścia: *SMT_window*, *SMT_signal* i źródła referencyjnego sygnału zegarowego. Na każdym z tych wejść jest umieszczony multiplexer umożliwiający wybranie jednej z kilku możliwości. Wbudowany rejestr 24-bitowy (*period register*) i komparator cyfrowy mogą generować zdarzenia typu równe wartości (*match*), przekroczony czas (*timeout*). Możliwość podawania sygnałów z różnych źródeł na 3 wejścia powoduje, że SMT może pracować w aż 11 unikalnych trybach!



Rysunek 1. Schemat blokowy licznika SMT

Źródłem zegarowego sygnału referencyjnego są wewnętrzne oscylatory RC lub zewnętrzny oscylator kwarcowy. W celu dokładnych pomiarów częstotliwości referencyjne powinny się charakteryzować dokładnością i stabilnością. W takim wypadku trzeba wybrać oscylator kwarcowy Primary Oscillator z zewnętrznym rezonatorem kwarcowym lub tryb EC (External Clock).

W trybie EC można taktować mikrokontroler z zewnętrznego źródła tak dokładnego i stabilnego, jak to potrzebne. W wielu zastosowaniach wystarczy dokładność wbudowanego oscylatora RC. Dla oscylatora HFINTOSC gwarantowana dokładność przy napięciu zasilania 2,3...5,5 V i w temperaturze otoczenia 0...60°C wynosi $\pm 2\%$, ale w temperaturze ujemnej już tylko $\pm 5\%$. **Wszystkie wymienione tryby pracy można konfigurować za pomocą wtyczki MCC MPLAB Code Configurator.** Moduł SMT może pracować w następujących trybach:

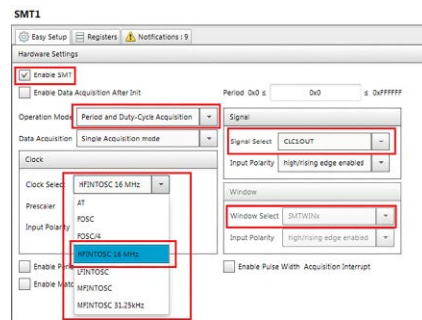
TRYB POMIARU OKRESU I WSPÓŁCZYNNIKA WYPEŁNIENIA (PERIOD AND DUTY CYCLE MODE).

W trybie pomiaru okresu i współczynnika wypełnienia używa się tylko zegara referencyjnego i sygnału wejściowego *SMT_signal*. Po zakończeniu pomiaru, w rejestrze CPR znajduje się wartość odpowiadająca długości okresu mierzonego sygnału, a w rejestrze CPW wartość odpowiadająca długości poziomu wysokiego *Ton* (z dokładnością do i w jednostkach okresu przebiegu referencyjnego). Rejestr *CPW* jest zapisywany przy zboczu opadającym, a rejestr *CPR* przy narastającym mierzonego przebiegu. Zbocze narastające ustawia flagę przerwania i inicjuje jego obsługę (jeżeli jest odblokowane). Jednocześnie inicjuje kolejny pomiar w trybie pomiaru automatycznego. Dzięki podwójnemu buforowaniu poprzedni pomiar jest dostępny do odczytania z rejestrów *CPR* i *CPW* w trakcie kolejnego mierzonego okresu. Konfigurację MCC pokazano na **rysunku 2**. Wybieramy tu:

- Odblokowanie modułu SMT.
- Wybór trybu *Period and Duty* – okno *Operation Mode*.
- *Data Acquisition* (pomiar pojedynczy lub cykliczny).
- Źródło sygnału referencyjnego – okno *Clock*.
- Źródło sygnału mierzonego – okno *Signal*.

Można też odblokować zgłaszanie przerwania po zakończeniu każdego z cykli pomiaru. Jest to doskonała metoda pomiaru współczynnika wypełnienia sygnału PWM. Taki pomiar może być użyty, na przykład, do odczytywania niektórych typów sensorów z wyjściem PWM, lub w innych układach automatyki. Na **listingu 1** zamieszczono procedurę inicjalizacji modułu SMT1 skonfigurowanego, jak to zostało pokazane na rys. 2.

TRYB HIGH AND LOW TIME MEASUREMENT. Ten tryb jest podobny do opisywanego wcześniej i jego zastosowanie jest takie samo. Różnica polega na tym, że zamiast mierzyć długość okresu, mierzy się długość



Rysunek 2. Konfiguracja trybu Period and Duty-Cycle Acquisition

```

Listing 1. Konfiguracja modułu SMT1 do pracy w trybie Period And
Duty Cycle Acquisition
void SMT1_Initialize(void)
{
    /* WPOL high/rising edge enabled; STP rolls over to 24'h000000;
       SPOL high/rising edge enabled; EN enabled;
       SMT1PS 1:1 Prescaler; CPOL rising edge; */
    SMT1CON0 = 0x80;
    /* SMT1REPEAT Single Acquisition mode;
       SMT1MODE Period and Duty-Cycle Acquisition;
       SMT1GO disabled; */
    SMT1CON1 = 0x02;
    // SMT1CPWUP SMT1CPW1 update complete; SMT1CPRUP SMT1PR1 update
    // complete; RST SMT1TMR1 update complete;
    SMT1STAT = 0x00;
    // SMT1CSEL HFINTOSC 16 MHz;
    SMT1CLK = 0x02;
    // SMT1WSEL CLC2OUT;
    SMT1WIN = 0x0A;
    // SMT1SSEL CLC1OUT;
    SMT1SIG = 0x07;
    // SMT1PR 0;
    SMT1PRU = 0x00;
    // SMT1PR 0;
    SMT1PRH = 0x00;
    // SMT1PR 0;
    SMT1PRL = 0x00;
}

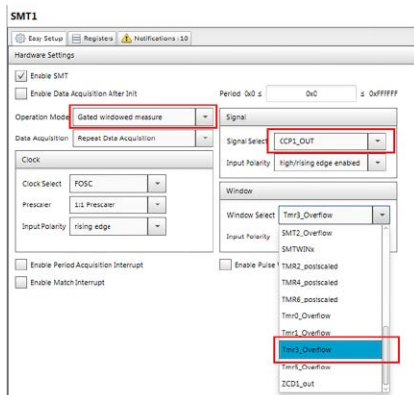
```

poziomu niskiego *Toff*. Wynik pomiaru *Toff* jest zapisywany w rejestrze *CPR*.

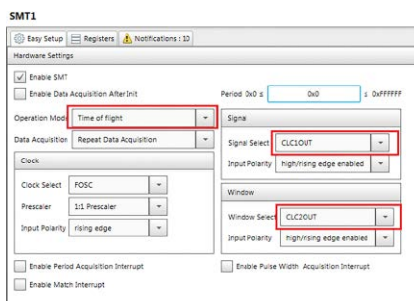
TRYB GATED WINDOWED MEASURE. W tym trybie jest mierzony czas trwania poziomu wysokiego *Ton*, ale pomiar jest wykonywany w „oknie”, którego czas trwania jest określony przez sygnał na wejściu *SMT1_window*. Czas trwania pomiaru może być dłuższy niż jeden okres mierzonego sygnału i dlatego możemy mówić o uśrednionym pomiarze czasu trwania *Ton* za czas trwania okna pomiarowego. Może to być przydatne do pomiaru sygnałów z fluktuacją fazy (z jitterem), w przypadku, kiedy pomiar uśredniony jest dopuszczalny. Sygnał „okna” może być wybierany z wielu różnych źródeł, na przykład może to być sygnał określony przez przepełnienie licznika, z układu generowania sygnału PWM, z wyjścia CLC itp. Odpowiednią konfigurację licznika pokazano na **rysunku 3**.

TRYB TIME OF FLIGHT MEASUREMENT. W aplikacjach, w których jest konieczny precyzyjny pomiar czasu pomiędzy dwoma różnymi zdarzeniami, można użyć trybu *Time of Flight Measurement*. Dwa mierzone sygnały są doprowadzone do wejść *SMT1_signal* i *SMT1_window*. Wejście sygnałowe może inicjować zliczanie impulsów zegara referencyjnego, a wejście „okna” kończy zliczanie sygnału referencyjnego. Zakończenie zliczania może zgłaszać przerwanie. Przerwanie może też być zgłaszane, jeśli zliczana wartość jest równa zapisanej w rejestrze *SMT1PR*. Na **rysunku 4** pokazano konfigurację trybu *Time of Flight* mierzącego czas pomiędzy zdarzeniami (zbozcami) na wyjściach modułów *CLC1* i *CLC2*. Zbozcza określające wystąpienie zdarzenia są konfigurowane w oknie *Input Polarity*.

TRYB TIMER. Moduł SMT pracuje tutaj w nim jak „zwykły” licznik zliczający impulsy z wejścia



Rysunek 3. Konfiguracja trybu Gated Windowed Measure



Rysunek 4. Tryb Time of flight

SMT1_Input. Jedyna różnica jest taka, że mamy do dyspozycji aż 24-bitowy rejestr licznika, niespotykany w standardowych licznikach innych mikrokontrolerów, w których, aby zwiększyć pojemność licznika trzeba stosować różne „sztuczki”.

TRYB WINDOWED MEASURE. Tryb *Windowed Measure* jest przeznaczony do pomiaru czasu trwania okresu sygnału podawanego na wejście *SMT1_window*. Każde narastające zbocze uruchamia zliczanie impulsów zegara referencyjnego. Kolejne narastające zbocze zeruje licznik impulsów i cykl pomiaru zaczyna się od nowa. Podwójne buforowanie umożliwia odczytanie wyniku poprzedniego pomiaru w trakcie kolejnego cyklu. Koniec każdego cyklu może zgłosić przerwanie. Możliwe jest też zgłoszenie przerwania, kiedy licznik zliczanych impulsów jest równy zawartości rejestru *SMT1PR*.

TRYB GATED TIMER. Tryb *Gated* jest spotykany w standardowych timerach mikrokontrolerów. Różnica polega na większej rozdzielczości pomiaru (24 bity). Kiedy sygnał na wejściu *SMT1_signal* jest ustawiony, to licznik zlicza impulsy referencyjne. Wynik pomiaru jest zapisywany do rejestru *CPW*.

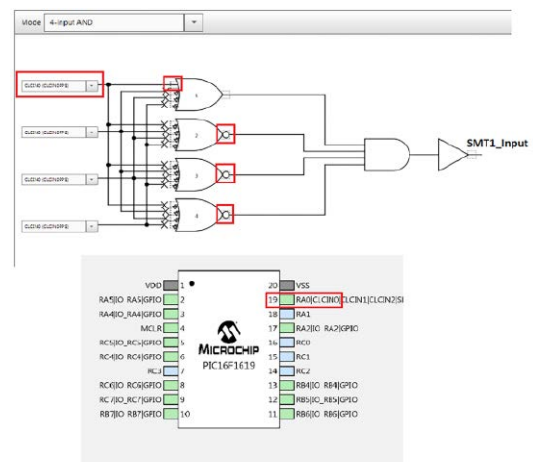
TRYB CAPTURE. Moduł SMT pracuje jak standardowy licznik (najczęściej 16-bitowy) w połączeniu z modułem *CCP* pracującym w trybie *Capture*. Trzeba pamiętać, że licznik nie jest zerowany i musi być zerowany programowo.

Opisane tryby pracują synchronicznie z rdzeniem mikrokontrolera. Wynika to z faktu zliczania impulsów zegara referencyjnego pracującego synchronicznie z rdzeniem, a w szczególności taktującego rdzeń. Można też skonfigurować 3 dodatkowe tryby pracujące asynchronicznie: *Counter Mode*, *Gated Counter Mode*, *Windowed Counter Mode*. W tych trybach są zliczane impulsy podawane na wejście *SMT1_signal*.

W praktyce przetestowałem tylko jeden tryb pracy: *Period and Duty Cycle Acquisition*. Żeby podać sygnał z zewnętrznego generatora trzeba go dołączyć przez moduł CLC – tu będzie to moduł *CLC1*, który skonfigurowano, jak na **rysunku 5**.

Mierzony sygnał jest podawany na wejście *CLCIN0*, a następnie przez skonfigurowany tor na wewnętrzne wejście *SMT1_input*. Po podaniu sygnału o zmiennym współczynniku wypełnienia z generatora, pojedynczy pomiar jest odblokowywany po wykonaniu funkcji *SMT1_DataAcquisitionEnable()* pokazanej na **listingu 2**. Pomiar rozpocznie się po ustawieniu bitu *SMT1GO*, a następnie po pierwszym narastającym zboczku na wejściu *SMT1_input*. Zakończenie pomiaru jest wykrywane przez testowanie bitu *AS* w rejestrze *SMT1STAT*. Jeżeli *AS* jest ustawiony, to *SMT* jest w trakcie pomiaru (**listing 3**). Po wyzerowaniu bitu *AS* pomiar jest zakończony i można odczytać rejestry *CPR* i *CPW* (**listing 4**).

Wszystkie pokazane tu funkcje zostały wygenerowane automatycznie przez wtyczkę MCC.



Rysunek 5. Konfiguracja CLC1

```
Listing 2. Odblokowanie pojedynczego pomiaru
void SMT1_DataAcquisitionEnable(void)
{
    // Start the SMT module by writing to SMTxGO bit
    SMT1CON1bits.SMT1GO = 1;
}
```

```
Listing 3. Testowanie zakończenia pomiaru
bool SMT1_IsSignalAcquisitionInProgress(void)
{
    return (SMT1STATbits.SMT1AS);
}
```

```
Listing 4. Odczytanie rejestrów CPR i CPW
uint32_t SMT1_GetCapturedPulseWidth()
{
    return (SMT1CPW);
}

uint32_t SMT1_GetCapturedPeriod()
{
    return (SMT1CPR);
}
```

AT Licznik Angular Timer

Licznik *Angular Timer* (rysunek 6) jest przeznaczony do pomiaru powtarzalnych sygnałów reprezentujących ruch obrotowy. Załóżmy, że na wale silnika elektrycznego mamy zamontowany jeden czujnik pomiarowy i każdy obrót wału powoduje wygenerowanie jednego impulsu. Czas pomiędzy tymi impulsami jest czasem jednego pełnego obrotu wału. Jeżeli podzielimy ten czas przez 360 stopni, to otrzymamy czas potrzebny na obrót wału o 1 stopień. Licznik *Angular Timer* wykonuje automatycznie generuje impulsy odpowiadające czasowi obrotu o 1 stopień.

Tak duża rozdzielczość może nie być potrzebna i dlatego można zdefiniować, co jaki kąt obrotu na wyjściu licznika będzie się pojawiał impuls lub będzie zgłaszane przerwanie. Na **rysunku 7** pokazano zasadę działania licznika. Na jego wejście są podawane impulsy z czujnika położenia wału silnika (*Time Base Input*), a na wyjściu występują impulsy odpowiadające zmianie położenia wału o 90°, bo tak zaprogramowano licznik.

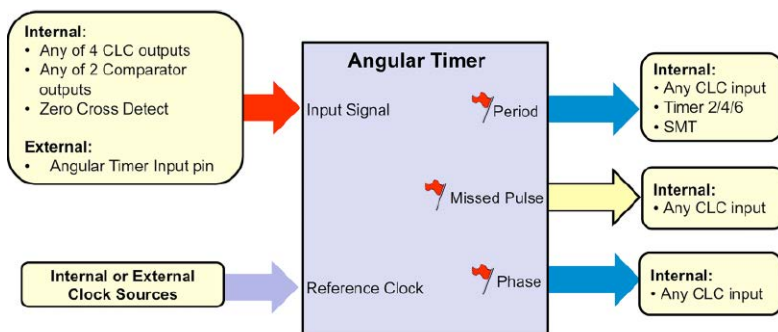
Do działania modułu są potrzebne dwa sygnały wejściowe: zegar referencyjny i sygnał wejściowy z czujnika położenia. Za wybór sygnału referencyjnego odpowiada rejestr *Angular Timer Clock (AT1CLK)*, a za wybór źródła sygnału wejściowego rejestr *Angular Timer Signal (AT1SIG)*. Z tych sygnałów są tworzone 3 sygnały wyjściowe:

- Period Clock.
- Phase Clock (impuls generowany co zaprogramowany kąt obrotu wału).
- Missing Pulse detector (impuls generowany po wykryciu zagubienia impulsu wejściowego).

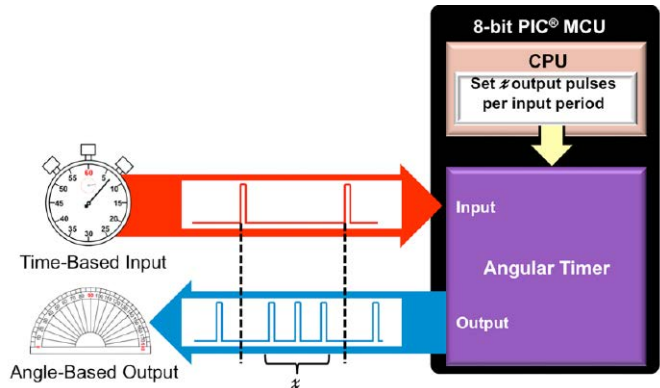
Licznik AT może pracować w dwóch trybach:

- Single Pulse Mode (jeden impuls na obrót).
- Multiple Pulse Mode (wiele impulsów na obrót).

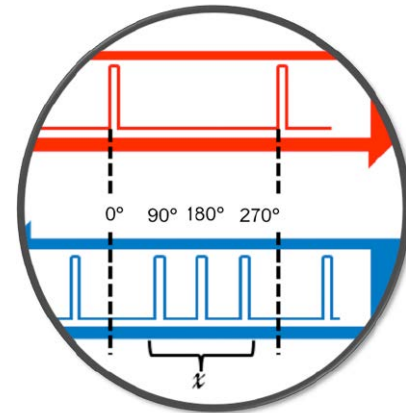
Tryb *Single Pulse* rozpoczyna zliczanie licznika po wykryciu impulsu z czujnika położenia wału. Kiedy zostanie wykryty kolejny impuls (po pełnym obrocie) w liczniku *Current Period Counter* jest zapisywana wartość odpowiadająca pełnemu obrotowi wału silnika (podwójne buforowanie). Jednocześnie licznik odliczający czas pomiędzy impulsami jest zerowany.



Rysunek 6. Schemat blokowy licznika Angular Timer



Rysunek 7. Działanie licznika Angular Timer



Rysunek 8. Generowanie dodatkowych impulsów co 90 stopni

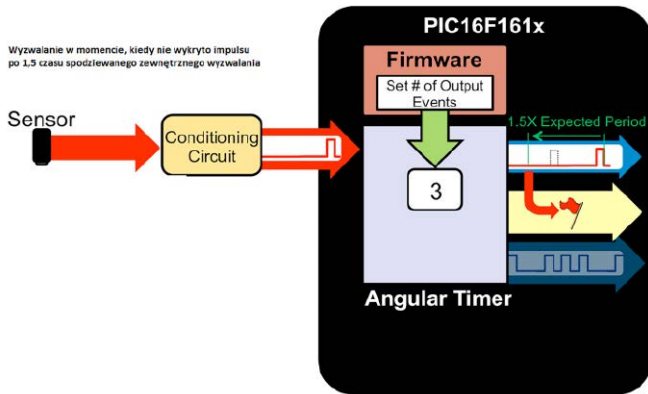
Licznik *Angular Timer* może generować ciąg impulsów bazujących na pojedynczych impulsach wejściowych generowanych co 1 obrót. Na **rysunku 8** pokazano powiększony przebieg z rys. 7.

Licznik *Angular Timer* wykrywa „zagubione” impulsy. Algorytm sterujący silnikiem może używać tej informacji i odpowiednio reagować. Licznik oczekuje, że co pełny obrót wystąpi się impuls wejściowy. W pewnych przypadkach, na przykład z powodu zatrzymania się silnika lub błędnego zadziałania czujnika na wale, impuls może się nie pojawić. Układ wejściowy jest wyzwalany, jeśli różnica pomiędzy zapamiętanym okresem poprzedniego, poprawnego wyzwalania zapisana w rejestrze *ATxPER* i bieżąca wartość licznika mierzącego czas pomiędzy impulsami (*Current Period Counter*) jest równa wartości zapisanej w rejestrach *ATxMISS*. Wartość *ATxMISS* jest zapisywana przez algorytm sterowania silnikiem i jest stała niezależnie od zmian częstotliwości wirowania wału silnika.

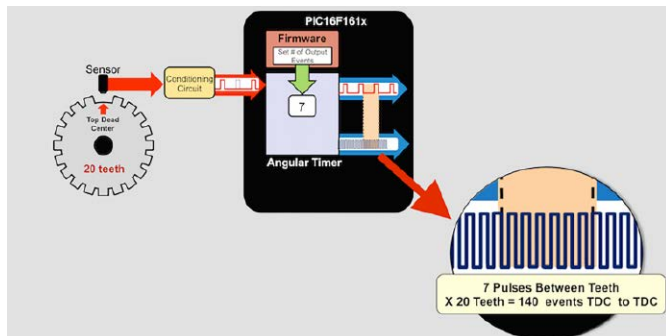
Istnieje też adaptacyjna metoda wykrywania brakujących impulsów. Układ wejściowy jest zawsze wyzwalany w czasie o 50% dłuższym, niż spodziewany zewnętrzny impuls wyzwalający o długości równej ostatniemu, zapamiętanemu okresowi wyzwalania. Metoda adaptacyjna automatycznie dostosowuje się do prędkości obrotowej silnika.

Licznik AT może też pracować w trybie *Multiple Pulse Mode*. Na wale silnika można umieścić czujnik generujący na przykład 20 impulsów na jeden obrót (**rysunek 10**). Jeżeli potrzebujemy do sterowania 140 impulsów na 1 obrót, to licznik *Angular Timer* może dodać 7 dodatkowych impulsów pomiędzy każdym impulsem wejściowym.

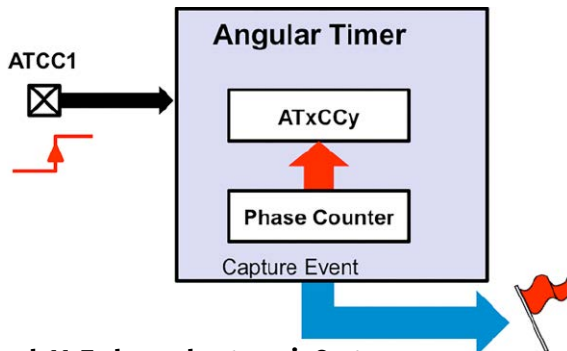
Do sprawdzania statusu (zawartości) rejestrów licznika AT, a także do wywoływania reakcji przy określonym kącie obrotu mogą być użyte tryby *Compare* i *Capture*. W trybie *Capture* procedury użytkownika mogą odczytywać (pobierać) stan licznika fazy po wyzwoleniu na wejściu *ATTCx* (**rysunek 11**). Może to mieć zastosowanie do określania kąta sygnału wejściowego do dalszych obliczeń w algorytmie sterowania. W trybie *Compare* (**rysunek 12**) jest porównywana wartość licznika fazy z wartością zapisaną



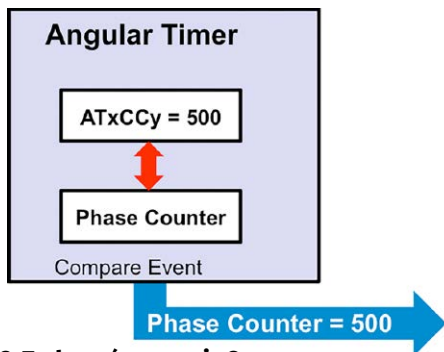
Rysunek 9. Działanie adaptacyjnej metody wykrywania zagubionych impulsów



Rysunek 10. Układ pracujący w trybie Multiple Pulse Mode



Rysunek 11. Tryb przechwytywania Capture



Rysunek 12. Tryb porównywania Compare

w rejestrze *ATxCCy*. Jeżeli obie liczby są równe, to moduł *Angular Timer* może zgłosić przerwanie. Konfigurację licznika najlepiej wykonać za pomocą wtyczki MCC (**rysunek 13**).

Przedstawiony tu opis timera zawiera podstawowe i z konieczności ogólne informacje pozwalające na zrozumienie zasady działania. Dokładny opis z dokładnym wyjaśnieniem działania wszystkich opcji i rejestrów licznika znajduje się w dokumentacji mikrokontrolera. Trzeba pamiętać, że *Angular Timer* to potężne wsparcie układów sterowania silnikami tym bardziej, że jego praca nie obciąża rdzenia mikrokontrolera. Nie zastąpi on jednak głównych algorytmów

sterowania, które użytkownik musi sobie utworzyć samodzielnie. W zależności od tych algorytmów, trzeba będzie wybrać odpowiednie tryby pracy i zaprogramować licznik.

Licznik NCO

Moduł *NCO* jest przeznaczony do użycia w aplikacjach, które wymagają generatora ściśle określonej, dokładnej częstotliwości. Może on nie tylko pełnić rolę generatora, ale również pozwala na precyzyjną regulację tej częstotliwości oraz współczynnika wypełnienia generowanego przebiegu.

Częstotliwość przebiegu cyfrowego można w prosty sposób dzielić używając do tego celu liczników. Moduł licznika jest jednym z podstawowych i umieszcza się go w każdym nawet najprostszym mikrokontrolerach PIC10F. Standardowo są to liczniki 8- lub 16-bitowe, wyposażane dodatkowo w preskalery i czasami w postskalery.

To czy uda nam się wygenerować potrzebną częstotliwość, zależy od częstotliwości przebiegu na wejściu i współczynnika podziału. Jeżeli wyliczony współczynnik podziału jest liczbą całkowitą, to można uzyskać dokładnie taki przebieg, jakiego oczekujemy. Na przykład, jeżeli dzielimy wejściową częstotliwość 2 MHz, a potrzebujemy 100 kHz, to 20 MHz/100 kHz=20. Problem pojawia się, gdy potrzebujemy np. 130 kHz. Wtedy 20 MHz/130 kHz=15,38461538... Po podzieleniu przez 15 otrzymujemy 133333,3333 Hz. Takie proste dzielenie albo jest obciążone sporym błędem, albo wymaga dobierania częstotliwości wejściowej, by współczynnik podziału był całkowity.

Microchip w module *NCO* zastosował sprytnie połączenie licznika z układem modyfikującym jego wartość w czasie każdego okresu zliczanego przebiegu (**rysunek 14**). Aby zapewnić odpowiednią rozdzielczość pomiaru, licznik ma długość 20 bitów, a modyfikowana wartość to 16 bitów. Układ sumatora działa niezależnie od zegara systemowego mikrokontrolera. Pobiera on wartość z akumulatora, dodaje zaprogramowaną liczbę i po dodaniu przesyła ponownie do akumulatora.

Schemat blokowy generatora *NCO* pokazano na **rysunku 15**. Sygnał wejściowy jest wybierany selektorem programowanym bitami *NxCKS*. Można wybrać przebieg z dedykowanego wejścia mikrokontrolera *NCOCLK*, wyjścia wbudowanego układu logicznego (CLC) *LC1OUT*, oscylatora kwarcowego *Fosc* lub wewnętrznego oscylatora *HFINTOSC* o maksymalnej częstotliwości 16 MHz. Wybrany przebieg jest zliczany przez 20-bitowy licznik-akumulator. W czasie każdego narastającego zbocza jest do niego dodawana 16-bitowa wartość zapisana przez użytkownika w 16-bitowym rejestrze inkrementacji *NCOxINCL*. Częstotliwość przepełniania się licznika-akumulatora wylicza się z zależności:

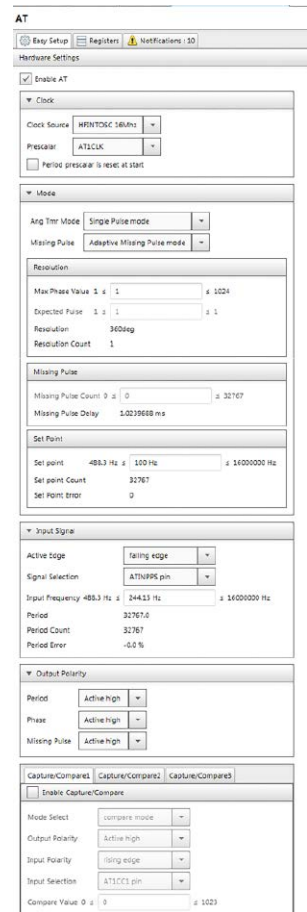
$$F_{OVERFLOW} = \frac{NCOClockFrequency \cdot IncrementValue}{2^n}$$

gdzie:

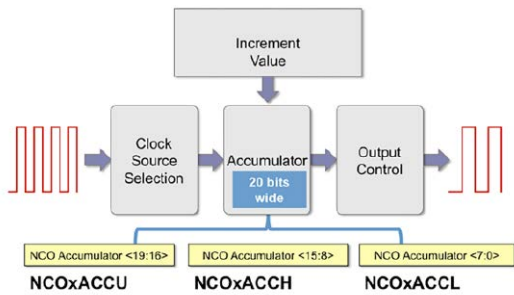
NCO Clock Frequency – częstotliwość wejściowa generatora *NCO*.

Increment Value – liczba dodawana do akumulatora.

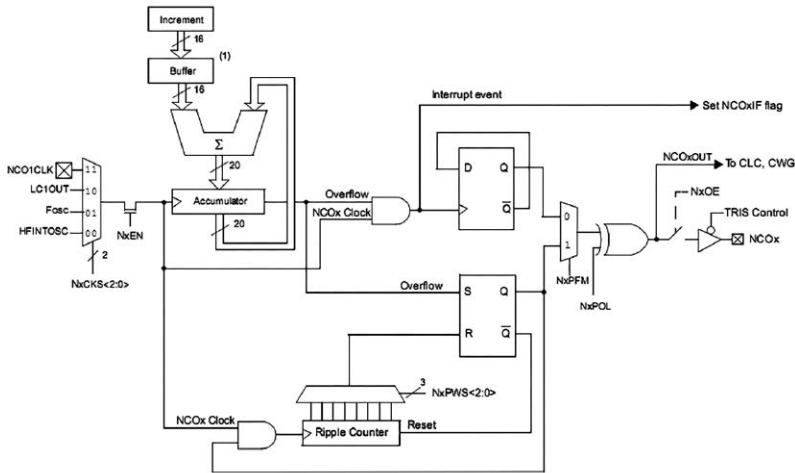
n – długość licznika akumulatora, tu 20 bitów.



Rysunek 13. Konfiguracja timera Angular Timer za pomocą wtyczki MCC



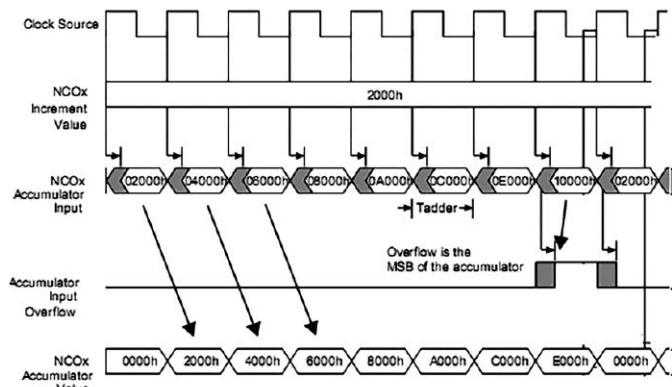
Rysunek 14. Zasada działania licznika NCO



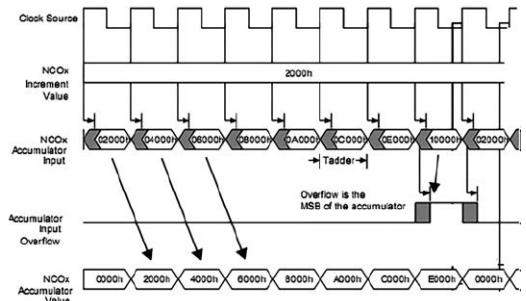
Rysunek 15. Schemat blokowy NCO

Im wartość zapisana w *Increment Value* będzie większa, tym licznik będzie się szybciej przepełniał i częstotliwość przebiegu F_{OVERFLOW} będzie większa. Na rysunku 16 pokazano przykład zliczania z dodawaniem wartości 0x2000 przy każdym narastającym zboczku przebiegu wejściowego. 20-bitowy licznik-akumulator przepełni się po 8 cyklach (okresach) przebiegu wejściowego. Wybierając kombinację częstotliwości wejściowej i zawartości rejestru *NCOxINCL* można precyzyjnie dobrać generowaną częstotliwość.

Przepełnienie licznika powoduje zmianę stanu przerzutnika D. Na jego wyjściu Q występuje przebieg prostokątny o wypełnieniu 50% i częstotliwości równej połowie częstotliwości przepełnień licznika. Zapisując bit *NxPOL* można zmieniać polaryzację sygnału wyjściowego. Przez selektor wyjściowy jest on podawany na wyprowadzenie *NCOx*, ale jest też wewnętrznie dostępny dla bloków peryferyjnych *CLC* i *CWG*. Każde przepełnienie licznika-akumulatora powoduje ustawienie znacznika przerwania *NCOxIF*. Jeżeli przerwanie nie jest zablokowane, to zostanie zgłoszone i musi być obsłużone. Źródłem przebiegu taktującego *NCO* mogą być następujące przebiegi: wyjściowy *HFINTOSC*, wyjściowy *FOSC*, wyjściowy *LCxOUT* (*CLC*),



Rysunek 16. Przykład zliczania z dodawaniem 2000hex



Rysunek 17. Tryb Fixed Duty Cycle

wejściowy (doprowadzenie) *CLKIN*. Źródło wybiera się za pomocą bitów *NxCKS* <1:0>.

Moduł *NCO* może pracować w dwóch trybach: *Fixed Duty Cycle* i *Pulse Frequency Modulation*. Tryb wybiera się za pomocą bitu *NxPFM* w rejestrze *NCOxCON*:

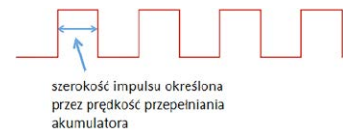
TRYB FDC (FIXED DUTY CYCLE). Ten tryb pracy wybierany wyzerowaniu bitu *NxPFM*. W trakcie pracy *NCO* cyklicznie zmienia stan na przeciwny po każdym przepełnieniu akumulatora (rysunek 17).

TRYB PULSE FREQUENCY MODULATION. W tym trybie wybieranym przez ustawienie bitu *NxPFM* po każdym przepełnieniu się licznika-akumulatora wyjście *NCOx* przechodzi w stan aktywny i pozostaje w tym stanie przez czas równy zaprogramowanej wielokrotności okresu zliczanego sygnału wejściowego (rysunek 18). Czas stanu aktywnego (poziom wysoki) i nieaktywnego (poziom niski) zależy od bitu określającego polaryzację *NxPOL* i bitów *NxPWS* rejestru konfiguracyjnego *NCOxCON*.

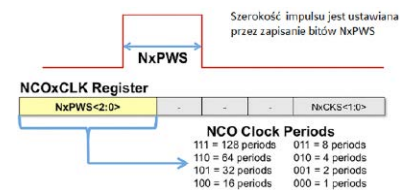
Do konfiguracji i testów generatora *NCO* wykorzystałem mikrokontroler PIC16F1509. Okno konfiguracji zostało pokazane na rysunku 19. Możemy wybrać tu:

- Odblokowanie modułu *NCO*.
- Tryb pracy.
- Polaryzację sygnału wyjściowego.

Silnym wsparciem konfiguracji modułu jest okno *Clock Settings*. Konfigurujemy tu źródło przebiegu taktującego i podajemy żądaną częstotliwość na wyjściu generatora. *MCC* oblicza rzeczywistą częstotliwość na wyjściu i generuje plik konfiguracyjny (listing 5). Testowanie polega na zmierzeniu generowanej częstotliwości na wyprowadzeniu *NCO* (*RC1* – pin 15).



Rysunek 18. Tryb Pulse Frequency Modulation



Rysunek 19. Konfiguracja NCO

```
Listing 5. Inicjalizacja NCO z parametrami pokazanymi na rys. 19
void NCO1_initialize (void)
{
    // Set the NCO to the options selected in the GUI
    // N1OE enabled; N1EN disabled; N1POL active_lo; N1PFM FDC_
    mode;
    NCO1CON = 0x40;
    // N1CKS HFINTOSC_16MHz; N1PWS 1_clk;
    NCO1CLK = 0x00;
    //
    NCO1ACCU = 0x00;
    //
    NCO1ACCH = 0x00;
    //
    NCO1ACCL = 0x00;
    // NCO1INCH 11
    NCO1INCH = 11;
    // NCO1INCL 199
    NCO1INCL = 199;
    // Enable the NCO module
    NCO1CONbits.N1EN = 1;
}
```