

STM8S001J3 (8)



Timer w trybie PWM

Ósma część cyklu artykułów dotyczących 8-pinowego mikrokontrolera STM8S001J3 poświęcona jest układowi licznikowemu (timerom). W artykule opisano budowę wybranego układu licznikowego oraz pokazano jak wykonać krok po kroku przykładową aplikację używającą tego zasobu mikrokontrolera do wygenerowania sygnału PWM (Pulse-Width Modulation).

Powszechnie spotykanym w mikrokontrolerach zasobem są układy licznikowe. Zasada ich działania opiera się na zliczaniu impulsów sygnału taktującego (generowanego wewnątrz mikrokontrolera lub doprowadzanego do niego z zewnątrz). Funkcjonalność ta umożliwia układom licznikowym na realizowanie przeróżnych zadań. Główne z nich to:

- Wytwarzanie podstawy czasu dla aplikacji przez generowanie przerwania co określony interwał czasu. Jest to wygodny mechanizm dla implementacji zadań cyklicznych. Ponadto przerwanie układu licznikowego może wybudzać mikrokontroler z trybu uśpienia.
- Wytwarzanie sygnałów cyfrowych na wyjściu portu mikrokontrolera, w szczególności sygnału PWM. Tego typu sygnał przydatny jest np. w sterowaniu prędkością silników DC, w ustawianiu pozycji serwomechanizmu oraz w zmienianiu poziomu jasności oświetlenia opartego na diodach LED.
- Określanie parametrów sygnałów cyfrowych na wejściu portu mikrokontrolera np. długości impulsu lub liczby impulsów. Druga z wymienionych funkcjonalności używana jest np. w tradycyjnych (mechanicznych) licznikach wody.

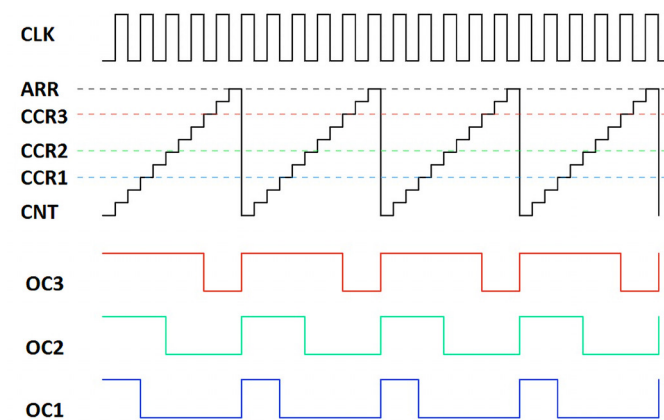
Cechą wspólną wszystkich sygnałów wytwarzanych przez układ licznikowy jest ich cyfrowy charakter. Oznacza to, że sygnał może przyjąć jeden z dwóch stanów: niski i wysoki. Wartość napięcia tych stanów odpowiada wartościom, jakie może przyjąć port mikrokontrolera: stan niski to 0 V, natomiast stan wysoki to typowo 3,3 V lub 5 V. Szczególnym przypadkiem sygnału generowanego przez układ licznikowy jest sygnał PWM. Jest to sygnał o zadanych: częstotliwości/okresie oraz poziomie wypełnienia (procentowo wyrażonym czasie trwania poziomu wysokiego w jednym okresie sygnału). Wytwarzanie sygnałów PWM przez układ licznikowy pokazano w praktyce na rysunku 1. CLK (clock) jest sygnałem taktującym układ licznikowy. Z każdym kolejnym zboczem sygnału zegarowego wartość licznika CNT (counter) zwiększa się o 1. Gdy na skutek zliczania wartość CNT osiągnie wartość równą CCR1 (capture/compare register 1), poziom

logiczny sygnał na wyjściu portu skojarzonego z kanałem pierwszym zmienia się. Zazwyczaj układ licznikowy dysponuje więcej niż jednym kanałem. Wtedy ich użycie jest możliwe w analogiczny sposób do kanału pierwszego. W przypadku trzech kanałów zdefiniowane są zatem wartości CCR2 i CCR3. Gdy CNT zrówna się najpierw z CCR2, a potem z CCR3, poziomy logiczny sygnałów na wyjściu portów skojarzonych z kanałami drugim i trzecim również zmieniają się. Wartość graniczną zliczania wyznacza ARR (auto-reload register). Osiągnięcie tej wartości przez CNT skutkuje jego wyzerowaniem. W tym momencie czasu poziomy logiczne sygnałów wszystkich trzech kanałów zmieniają się, a licznik rozpoczyna zliczanie od początku. Zatem jest tak, że wartości CCR1, CCR2 oraz CCR3 określają poziom wypełnienia sygnałów PWM odpowiednio dla kanału pierwszego, drugiego i trzeciego, natomiast wartość ARR określa częstotliwość/okres wszystkich trzech sygnałów PWM.

Układy licznikowe w STM8S001J3

Mikrokontrolery STM8S mogą być wyposażone (w zależności od modelu) w trzy różne typy układów licznikowych: zaawansowanego sterowania (TIM1), ogólnego przeznaczenia (TIM2/TIM3/TIM5) oraz podstawowe (TIM4/TIM6). Wszystkie te peryferia oparte są na tej samej architekturze (rejstry, podstawowe funkcje), natomiast różnią się wybranymi cechami i parametrami. Zestawienie porównawcze zaprezentowano w tabeli 1.

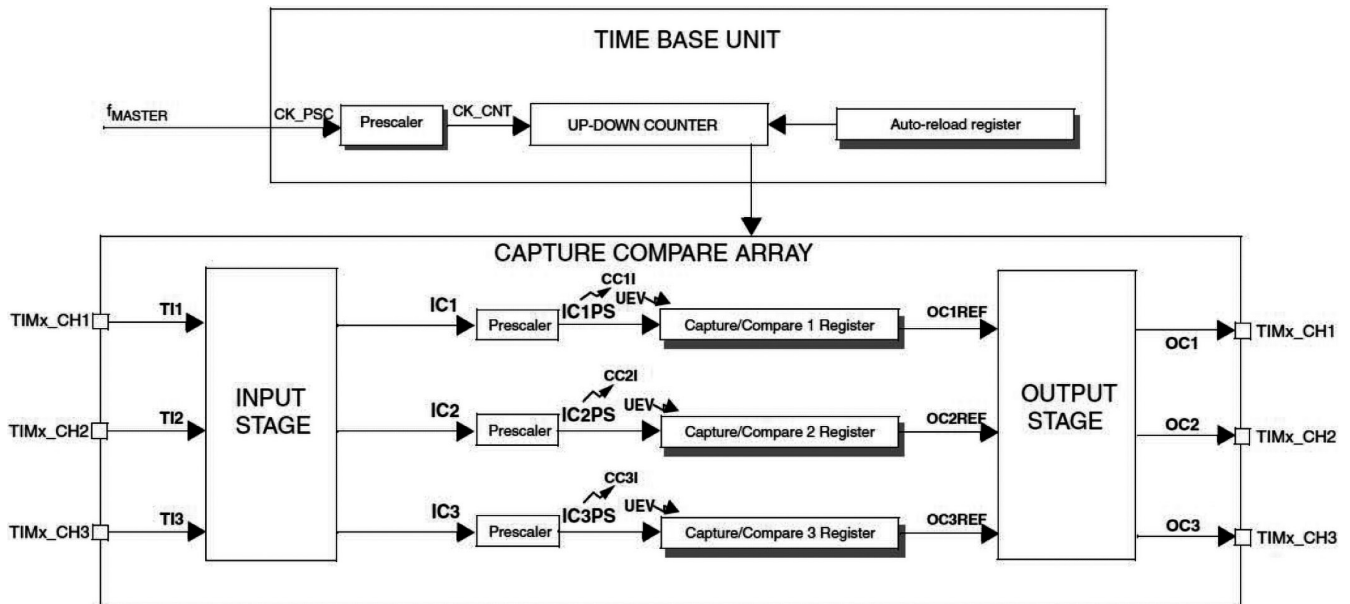
Mikrokontroler STM8S001J3 wyposażony jest w dwa układy licznikowe: TIM1 oraz TIM2. Schemat blokowy układu licznikowego



Rysunek 1. Sposób wytwarzania sygnałów PWM przez układ licznikowy

Tabela 1. Zestawienie porównawcze układów licznikowych dostępnych w układach STM8S

Układ licznikowy	Rozdzielczość licznika	Rodzaj zliczania	Podzielnik	Liczba kanałów	Kanały komplementarne	Funkcja repetition counter	Zewnętrzne wyzwolenie timera	Funkcja break	Synchronizacja timerów
TIM1	16 bitów	w górę/ w dół	1...65536	4	3	Tak	Tak	Tak	z TIM5/ TIM6
TIM2			1...32768	3	0	Nie	Nie	Nie	Nie
TIM3	w górę	1...128	2	0					
TIM4		8 bitów	1...128	0	0				
TIM5	16 bitów	w górę	1...32768	3	0	Tak	Tak	Tak	Tak
TIM6	8 bitów		1...128	0	0	Nie	Nie	Nie	Nie



Rysunek 2. Schemat blokowy układu licznikowego TIM2

TIM2 pokazano na **rysunku 2**. Układ ten taktowany jest (podobnie jak wszystkie zasoby mikrokontrolera) przez sygnał f_{MASTER} , który wytwarzany jest z wewnętrznego lub zewnętrznego źródła sygnału zegarowego (HSI, LSI lub HSE). Maksymalna częstotliwość tego sygnału to 16 MHz. f_{MASTER} doprowadzony jest do modułu układu licznikowego o nazwie *TIME BASE UNIT*. Istnieje możliwość zredukowania częstotliwości tego sygnału przez 4-bitowy podzielnik. Następnie impulsy sygnału taktującego są zliczane przez 16-bitowy licznik (w zależności od konfiguracji w górę lub w dół). Wartość licznika nieustannie porównywana jest z wartością ustawioną w rejestrze *Auto-reload*, aby w momencie gdy wielkości te są takie same nastąpiło przywrócenie licznikowi wartości początkowej. Wartość licznika używana jest przez drugi z modułów układu licznikowego – *CAPTURE/COMPARE ARRAY*. Składa się on głównie z podzielników i rejestrów *Capture/Compare*, sekcji *OUTPUT STAGE* odpowiedzialnej za wytwarzanie sygnałów cyfrowych na wyjściach kanałów i portów oraz sekcji *INPUT STAGE* odczytującej sygnały cyfrowe z wejść kanałów i portów.

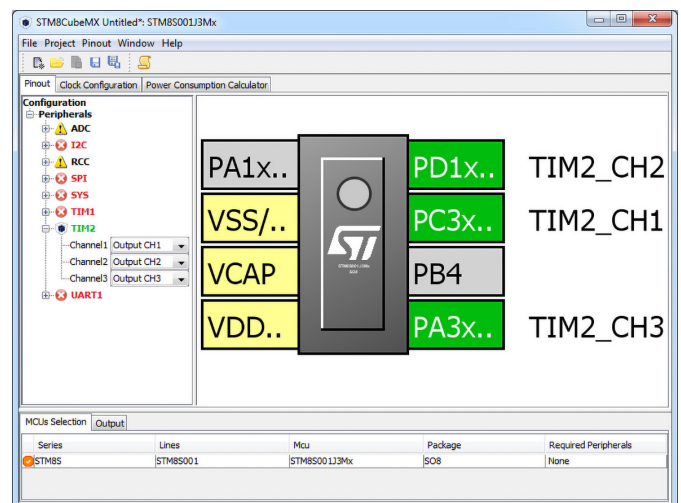
W **tabeli 2** zamieszczono wykaz wszystkich portów mikrokontrolera STM8S001J3, które mogą zostać wykorzystane jako kanały układu licznikowego TIM2. Każdy z tych trzech kanałów może zostać skonfigurowany w wejściowy lub wyjściowy tryb pracy.

TIM2 w STM8CubeMX

Narzędziem ułatwiającym pracę z portami wejścia/wyjścia mikrokontrolera STM8S001J3 jest program komputerowy STM8CubeMX. Dzięki niemu programista może w prosty sposób (za pomocą graficznego interfejsu użytkownika) sprawdzić jakie są możliwe konfiguracje dla wszystkich portów oraz peryferiów mikrokontrolera. Tak jest

Tabela 2. Wykaz Zestawienie wszystkich portów mikrokontrolera STM8S001J3, które mogą zostać wykorzystane na potrzeby układu licznikowego TIM2

Numer pinu na obudowie S08	Nazwa pinu	Funkcja alternatywna	Funkcja alternatywna „remap”
5	PA3/TIM2_CH3/ SPI_NSS/UART1_TX	Timer 2 channel 3	
7	PC5/SPI_SCK/ TIM2_CH1		Timer 2 channel 1
8	PD3/AIN4/ TIM2_CH2/ADC_ETR	Timer 2 channel 2	



Rysunek 3. Przykładowa konfiguracja mikrokontrolera STM8S001J3 w programie STM8CubeMX

również w przypadku układów licznikowych. Przykładowy scenariusz z trzema kanałami TIM2 pokazano na **rysunku 3**.

Funkcje SPL do sterowania układem licznikowym TIM2

Aby w prosty sposób skonfigurować i używać licznika TIM2 w układzie STM8S001J3, warto w aplikacji użyć bibliotek SPL (*Standard Peripheral Library*) przygotowanych dla mikrokontrolerów z rodziny STM8S. Pliki *stm8s_tim2.h* oraz *stm8s_tim2.c* udostępniają szereg funkcji do tego celu. Zestawienie najważniejszych funkcji zaprezentowano w **tabeli 3**.

Dodatkowo, każdy kanał TIM2 dysponuje osobnym zestawem funkcji, różniącym się tylko indeksem w nazwie funkcji. Funkcje te, na przykładzie kanału pierwszego, przedstawiono w **tabeli 4**.

Przykładowa aplikacja sterująca układem licznikowym TIM2

W celu stworzenia przykładowej aplikacji użyte zostało środowisko programistyczne STVD (ST Visual Develop) oraz kompilator Cosmic CXSTM8. Opis tych narzędzi, jak również instrukcja jak stworzyć za ich pomocą szablon nowego projektu wraz z dodaniem bibliotek SPL dostępne są w artykule numer 3 z tej serii (EP 2/2018). Korzystając ze wspomnianego szablonu projektu należy edytować kod pliku *main.c*, w którym umieszczony zostanie kod aplikacji.

Tabela 3. Zestawienie wybranych funkcji z biblioteki dla układu licznikowego TIM2	
Nazwa funkcji	Opis działania funkcji
TIM2_DeInit(void)	Konfiguracja domyślna układu licznikowego TIM2
TIM2_TimeBaseInit(TIM2_Prescaler_TypeDef TIM2_Prescaler, uint16_t TIM2_Period)	Konfiguracja podstawy czasu układu licznikowego TIM2 według ustawień użytkownika
TIM2_ICInit(TIM2_Channel_TypeDef TIM2_Channel, TIM2_ICPolarity_TypeDef TIM2_ICPolarity, TIM2_ICSelection_TypeDef TIM2_ICSelection, TIM2_ICPSC_TypeDef TIM2_ICPrescaler, uint8_t TIM2_ICFilter)	Konfiguracja trybu Input Capture układu licznikowego TIM2 według ustawień użytkownika
TIM2_PWMConfig(TIM2_Channel_TypeDef TIM2_Channel, TIM2_ICPolarity_TypeDef TIM2_ICPolarity, TIM2_ICSelection_TypeDef TIM2_ICSelection, TIM2_ICPSC_TypeDef TIM2_ICPrescaler, uint8_t TIM2_ICFilter)	Konfiguracja trybu PWM Input układu licznikowego TIM2 według ustawień użytkownika
TIM2_Cmd(FunctionalState NewState)	Włączenie lub wyłączenie układu licznikowego TIM2
TIM2_ITConfig(TIM2_IT_TypeDef TIM2_IT, FunctionalState NewState)	Włączenie lub wyłączenie wybranych przerw dla układu licznikowego TIM2
TIM2_PrescalerConfig(TIM2_Prescaler_TypeDef Prescaler, TIM2_PSCReloadMode_TypeDef TIM2_PSCReloadMode)	Konfiguracja podzielnika sygnału zegarowego dla układu licznikowego TIM2
TIM2_SetCounter(uint16_t Counter)	Ustawienie nowej wartości licznika układu licznikowego TIM2
TIM2_SetAutoreload(uint16_t Autoreload)	Ustawienie nowej wartości Auto-reload układu licznikowego TIM2
TIM2_GetCounter(void)	Odczytanie aktualnej wartości licznika układu licznikowego TIM2
TIM2_GetPrescaler(void)	Odczytanie wartości podzielnika licznika układu licznikowego TIM2
TIM2_GetFlagStatus(TIM2_FLAG_TypeDef TIM2_FLAG)	Odczytanie stanu wybranej flagi układu licznikowego TIM2
TIM2_ClearFlag(TIM2_FLAG_TypeDef TIM2_FLAG)	Wyzerowanie wybranej flagi układu licznikowego TIM2
TIM2_GetITStatus(TIM2_IT_TypeDef TIM2_IT)	Odczytanie stanu wybranego przerwania układu licznikowego TIM2
TIM2_ClearITPendingBit(TIM2_IT_TypeDef TIM2_IT)	Wyzerowanie wybranego przerwania układu licznikowego TIM2

Tabela 4. Zestawienie wybranych funkcji z biblioteki dla kanału pierwszego TIM2	
Nazwa funkcji	Opis działania funkcji
TIM2_OC1Init(TIM2_OCMode_TypeDef TIM2_OCMode, TIM2_OutputState_TypeDef TIM2_OutputState, uint16_t TIM2_Pulse, TIM2_OCPolarity_TypeDef TIM2_OCPolarity)	Konfiguracja trybu Output Compare dla kanału pierwszego układu licznikowego TIM2 według ustawień użytkownika
TIM2_SetCompare1(uint16_t Compare1);	Ustawienie nowej wartości Compare dla kanału pierwszego układu licznikowego TIM2
TIM2_GetCapture1(void)	Odczytanie aktualnej wartości Compare dla kanału pierwszego układu licznikowego TIM2
TIM2_SetIC1Prescaler(TIM2_ICPSC_TypeDef TIM2_IC1Prescaler)	Konfiguracja podzielnika sygnału zegarowego dla kanału pierwszego układu licznikowego TIM2 w trybie Input Capture

Przykładowa aplikacja wykorzystuje układ licznikowy TIM2 do generowania sygnału PWM. Wybrany do tego celu został kanał trzeci dostępny na porcie mikrokontrolera PA3 (pin numer 5). Aplikacja zwiększa wypełnienie sygnału od 0 do 99% z. Zwiększanie odbywa się stopniowa – z krokiem 1%. Po osiągnięciu wypełnienia maksymalnego czynność zaczyna się od początku (nieskończona pętla). W celu zaimplementowania opisanej aplikacji wykonane zostaną następujące kroki w pliku main.c:

- Utworzenie zmiennej *Pulse*, która użyta zostanie do przechowywania poziomu wypełnienia sygnału PWM (wartość początkowa 0).
- Utworzenie funkcji opóźniającej *delay()*.
- Wywołanie funkcji opóźniającej *delay()*, co przeciwdziała przez kilka sekund ewentualnemu późniejszemu wyłączeniu interfejsu programowania i debugowania SWIM będącego efektem rekonfiguracji portów.
- Wykonanie kodu konfiguracji portów wejścia/wyjścia, które nie są połączone z wyprowadzonymi mikrokontrolera (kod wzięty z noty aplikacyjnej AN5047: *Getting started with the STM8S001j3 microcontroller*).
- Wywołanie funkcji *TIM2_Deinit()* w celu wykonania konfiguracji domyślnej układu licznikowego TIM2.
- Wywołanie funkcji *TIM2_TimeBaseInit()* w celu skonfigurowania podstawy czasu dla układu licznikowego TIM2.

- Wywołanie funkcji *TIM2_OC3Init()* w celu skonfigurowania kanału trzeciego układu licznikowego TIM2 do pracy w trybie *Output Compare*.
- Wywołanie funkcji *TIM2_Cmd()* w celu włączenia układu licznikowego TIM2.
- Wewnątrz nieskończonej pętli *while()*:
 - Wywołanie funkcji *TIM2_SetCompare3 ()* w celu zmiany poziomu wypełnienia sygnału PWM zgodnie z wartością zmiennej *Pulse*.
 - Zwiększenie o 1 wartości zmiennej *Pulse*.
 - Sprawdzenie za pomocą instrukcji warunkowej *if()* czy zmienna *Pulse* osiągnęła wartość 99. Jeśli tak, wyzerowanie zmiennej.
 - Wywołanie funkcji opóźniającej *delay()* w celu wytworzenia odstępu czasu między kolejnymi zmianami poziomu wypełnienia sygnału PWM.

Kod zgodny z zaprezentowanym opisem pokazano w **listingu 1**. Działanie aplikacji obserwować można poprzez podłączenie diody LED do portu PA3 mikrokontrolera. Jasność świecenia diody LED odzwierciedla poziom wypełnienia sygnału PWM. Przy zerowym i niskim poziomie wypełnienia dioda LED nie emituje światła. Przy bardziej znacznym poziomie wypełnienia dioda zaczyna emitować światło. Wraz ze wzrostem poziomu wypełnienia poziom jasności diody LED również zwiększa się. Jasność maksymalna uzyskiwana

Listing 1. Stworzony w oparciu o funkcje SPL kod pliku main.c

```

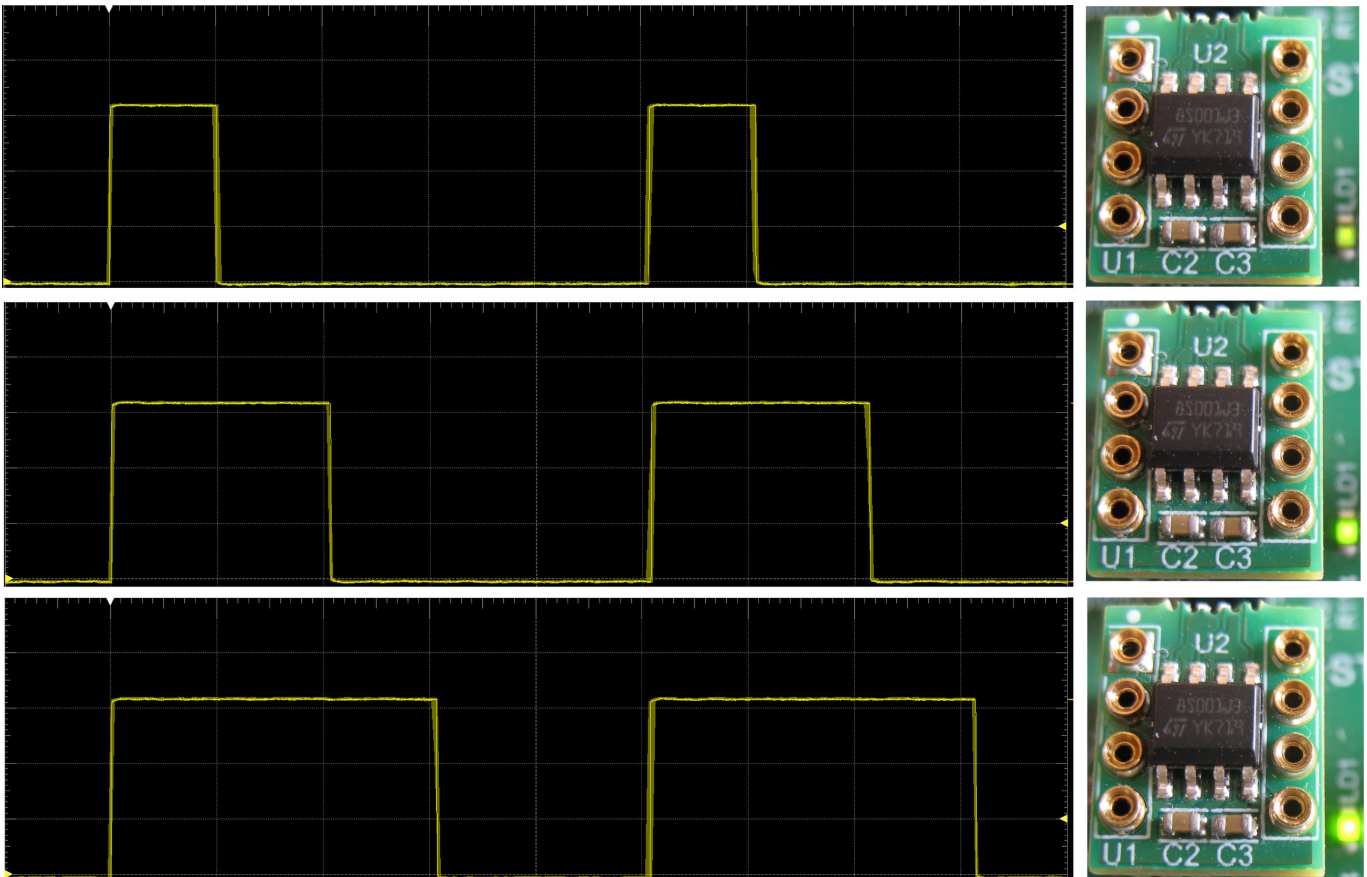
#include "stm8s.h"
uint16_t Pulse = 0;

void delay(unsigned long int how_long);
main()
{
    //delay to avoid irreversible blocking of SWIM
    delay(100000);
    //configuration of unused pins
    GPIOA->DDR |= GPIO_PIN_2;
    GPIOB->DDR |= GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_6 | GPIO_PIN_7;
    GPIOC->DDR |= GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_7;
    GPIOD->DDR |= GPIO_PIN_0 | GPIO_PIN_2 | GPIO_PIN_4 | GPIO_PIN_7;
    GPIOE->DDR |= GPIO_PIN_5;
    GPIOF->DDR |= GPIO_PIN_4;
    //TIM configuration
    TIM2_DeInit();
    TIM2_TimeBaseInit(TIM2_PRESCALER_1, 100);
    TIM2_OC3Init(TIM2_OCMODE_PWM1, TIM2_OUTPUTSTATE_ENABLE, Pulse, TIM2_OCPOLARITY_HIGH);
    TIM2_Cmd(ENABLE);

    while (1)
    {
        TIM2_SetCompare3(Pulse);
        Pulse++;
        if(Pulse == 99) Pulse = 0;
        delay(1000);
    }
}

void delay(unsigned long int how_long)
{
    unsigned long int i;
    for (i = 0; i < how_long; i++);
}

```



Rysunek 4. Widok sygnału PWM wygenerowanego przez aplikację oraz zdjęcie mikrokontrolera i diody LED sterowanej sygnałem PWM. Od góry: poziom wypełnienia 20%, poziom wypełnienia 40%, poziom wypełnienia 60%

jest dla maksymalnego poziomu wypełnienia sygnału PWM. Bardziej szczegółowo sygnał PWM można obserwować za pomocą oscyloskopu. Wygenerowany przez aplikację sygnał PWM z kilkoma przykładowymi poziomami wypełnień pokazano na **rysunku 4**.

Podsumowanie

W artykule przekazano podstawowe informacje o układzie licznikowym TIM2 mikrokontrolera STM8S001J3 wraz z opisem przykładowej aplikacji. Osoby chcące dowiedzieć się bardziej szczegółowych informacji powinny sięgnąć do dokumentacji technicznej producenta.

Parametry i charakterystyka wszystkich układów licznikowych dostępne są w nocie katalogowej mikrokontrolera (*datasheet*). Z kolei schematy oraz opis wszystkich funkcjonalności i rejestrów znajduje się w podręczniku użytkownika mikrokontrolera (*user manual* RM0016). Dodatkowo szereg aplikacji testowych dostępny jest w dedykowanym podkatalogu bibliotek SPL: `..\STM8S_StdPeriph_Lib\Project\STM8S_StdPeriph_Examples\TIMx`, gdzie x oznacza numer licznika.

Szymon Panecki
szymon.panecki@st.com