

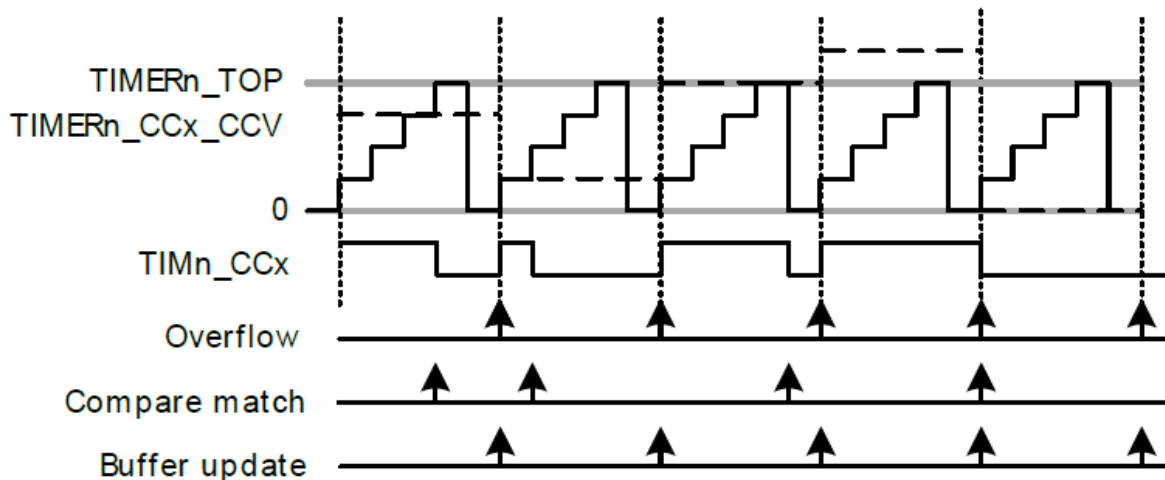
PROJEKTOWANIE ENERGOOSZCZĘDNYCH SYSTEMÓW WBUDOWANYCH

ĆWICZENIE 4

LOW ENERGY TIMER, BURTC

1. Low Energy Timer – tryb PWM

Modulacja szerokości impulsu (PWM) jest często stosowana przy sterowaniu silnikami DC, jasnością źródła światła, etc. Do tego celu można użyć liczników z zaimplementowaną funkcją PWM w które wyposażone są mikrokontrolery. Podstawowa zasada działania PWN przedstawiona jest na rysunku 1.



Rys. 1. Sposób generowania impulsów o zmiennej szerokości

Za pomocą rejestru TOP ustawia się okres impulsu, natomiast rejestrem porównawczym (CC lub COMP) reguluje się szerokość impulsu.

W naszym przypadku w mikrokontrolerze EFM32 do wygenerowania impulsów o zmiennej szerokości można użyć licznika Low Energy Timer (LETIMER), który pozwala na znaczną optymalizację zużycia energii podczas pracy w trybie PWM.

W celu przetestowania działania licznika należy otworzyć projekt Low Energy Timer dla płytki STK-3700 ([AN0026](#)). Kod pętli main jest następujący:

```

1. /*****
2.  * @brief Main function
3.  * Main is called from __iar_program_start, see assembly startup file
4.  *****/
5. int main(void)
6. {
7.     /* Align different chip revisions */
8.     CHIP_Init();
9.
10.    /* Initialize LETIMER */
11.    LETIMER_setup();
12.
13.    //-----
14.    // THE INTERRUPT IS SIMPLY TO DECREASE THE VALUE OF COMP1 TO VARY THE PWM DUTY CYCLE
15.    //-----
16.    /* Enable underflow interrupt */
17.    LETIMER_IntEnable(LETIMER0, LETIMER_IF_UF);

```

```

18.
19. /* Enable LETIMER0 interrupt vector in NVIC*/
20. NVIC_EnableIRQ(LETIMER0_IRQn);
21.
22. while (1)
23. {
24.     /* Go to EM2 */
25.     EMU_EnterEM2(false);
26. }
27.}

```

Algorytm działania powyższego kodu jest następujący: konfigurowany jest licznik LETIMER, aktywowane zostaje przerwanie od porównania (underflow) i na końcu układ wchodzi w tryb energetyczny EM2.

Licznik LETIMER posiada 2 rejestry porównawcze – COMP0 i COMP1. Licznik ten zlicza w dół. Począwszy od COMP0 (#define COMPMAX 500). Rejestr porównawczy COMP1 również ustawiony jest początkowo na COMPMAX 500. Przy każdym przerwaniu od porównania wykonuje się następujący kod:

```

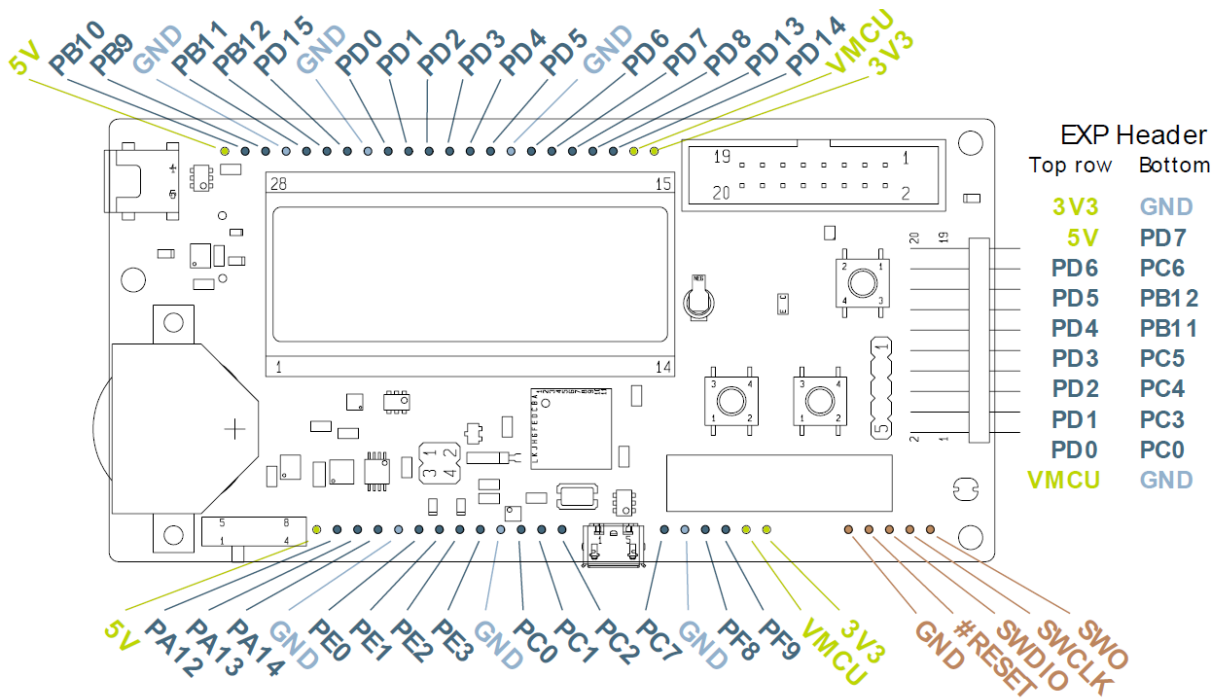
1. /******//**
2. * @brief LETIMER0_IRQHandler
3. * Interrupt Service Routine for LETIMER
4. *****/
5. void LETIMER0_IRQHandler(void)
6. {
7.     /* Clear LETIMER0 underflow interrupt flag */
8.     LETIMER_IntClear(LETIMER0, LETIMER_IF_UF);
9.
10.    /* If the value of comp1 is over 0, decrement it
11.       or bring comp1 back to COMPMAX (500) */
12.    if(comp1 != 0)
13.        comp1--;
14.    else
15.        comp1 = COMPMAX;
16.
17.    /* Write the new compare value to COMP1 */
18.    LETIMER_CompareSet(LETIMER0, 1, comp1);
19.}

```

Zatem przy każdym impulsie następuje dekrementacja rejestru porównawczego COMP1 powodująca płynną zmianę szerokości impulsu. Przy COMP1 równym 0 następuje przepisanie wartości początkowej (COMPMAX).

W sprawozdaniu proszę o wyliczenie wartości teoretycznej okresu wysyłania impulsów na podstawie częstotliwości użytego zegara oraz wartości COMPMAX.

Sygnal PWM generowany jest na pinie PD6. Za pomocą oscyloskopu proszę zaobserwować i umieścić w sprawozdaniu sygnał PWM. Proszę zmierzyć okres i porównać z obliczeniami teoretycznymi. Lokalizacja pinu PD6 na płytce STK3700 jest zaznaczona na rysunku 2.



Rys. 2. Schemat wyprowadzeń płytki STK3700

Istnieje również inna możliwość sprawdzenia poprawności generowania sygnału PWM za pomocą znajdującej się na płycie STK diody LED (LED0 lub LED1). Diody podłączone są pod piny PE2 oraz PE2. Za pomocą kabelka proszę połączyć pin PD6 z jednym z pinów diody. W sprawozdaniu proszę przedstawić swoje obserwacje, nt. jak zmienia się światło emitowane przez diodę sterowaną sygnałem PWM.

Dodatkowo w sprawozdaniu proszę przedstawić wykres zużycia prądu przez mikrokontroler podczas pracy (wykorzystać Energy Profiler) oraz zaznaczyć w których momentach znajduje się on w trybie energetycznym EM2 oraz w którym momencie następuje przerwanie z licznika LETIMER.

Na oscyloskopie proszę również zaobserwować i umieścić w sprawozdaniu kształt sygnału z pinu PD7. Co to jest za sygnał i jak jest on generowany?

Proszę o takie zmodyfikowanie kodu, aby zamiast licznika LETIMER został użyty zwykły licznik TIMER/COUNTER. Przy zachowaniu takiej samej funkcjonalności proszę przedstawić w sprawozdaniu wykres poboru prądu przez mikrokontroler i porównać go z poprzednim wykresem.

2. Backup Real Time Counter (BURTC)

Ciekawą funkcjonalnością cechuje się moduł BURTC w mikrokontrolerach serii Giant Gecko. Po ustawieniu odpowiedniego czasu w zegarze możemy aktywować domenę zasilania backup. Może ona być zasilana przez zapasową baterię (lub inne źródło zasilania) lub nawet zwykły kondensator. Przy zaniku podstawowego zasilania (domena VMCU) do jednego z rejestrów BURTC zostaje zapisany aktualny czas (tzw. timestamp) oraz uruchamiany zostaje licznik, który zasilany jest z domeny backup. W przypadku, gdy powróci zasilanie główne przed wyczerpaniem się zasilania backup można odczytać rejestr z timestamp'em oraz aktualną wartość licznika. Znając częstotliwość zegara tego licznika możemy odzyskać aktualny czas.

W celu przetestowania działania BURTC proszę skompilować oraz uruchomić na płytce STK3700 projekt Backup Power Domain ([AN0041](#)). Po uruchomieniu kodu proszę ustawić aktualny czas za pomocą przycisków PB0 oraz PB1. Następnie za pomocą suwaka znajdującego się w lewym dolnym rogu płytki proszę wyłączyć zasilanie domeny VMCU i odczekać chwilę. Następnie za pomocą suwaka proszę powrócić do odpowiedniego zasilania domeny VMCU i sprawdzić czy zegar wyświetla odpowiednią wartość.

Proszę przeanalizować kod programu oraz wykres poboru energii podczas działania programu. Wnioski proszę przedstawić w sprawozdaniu. Dodatkowo, proszę o wyliczenie na jaki czas wystarczy w pełni naładowany kondensator zastosowany na płytce STK w domenie backup, aby można było odtworzyć zegar po utracie zasilania głównego.