Akademia Górniczo-Hutnicza
w Krakowie
Instytut Elektroniki
WIET

Laboratorium
Technika Mikroprocesorowa 2

Ćwiczenie 4

Liczniki TPM0 i TPM1
PWM

Autor: Mariusz Sokołowski

wer. 28.09.2021

1. WSTĘP

1.1. CEL

Celem ćwiczenia jest:

- zapoznanie się z możliwościami liczników TPM0 i TPM1,
- konfiguracja i wykorzystanie licznika TPMx do pracy w trybie PWM jako regulatora jasności świecenia diody LED – współpraca z polem dotykowym,
- konfiguracja i wykorzystanie licznika TPMx do pracy w trybie PWM jako źródła sygnału akustycznego, o regulowanej częstotliwości – współpraca z głośnikiem,
- współpraca licznika systemowego SysTick i klawiatury, z licznikiem TPM0 – generowanie dźwięków temperowanych dla różnych oktaw,
- elektroniczna pozytywka.

1.2. WYMAGANIA

Sprzętowe:

- komputer klasy PC, spełniający wymagania sprzętowe aplikacji KEIL v5,
- zestaw FRDMKL05Z

Programowe:

- system operacyjny Windows 7 lub wyższy (wszystkie instrukcje powstały w oparciu o Windows 7 Pro x64),
- środowisko Keil / uVision 5 MDK-ARM

Doświadczenie:

- podstawowa umiejętność obsługi komputera klasy PC,
- podstawowa znajomość systemów operacyjnych rodziny Windows,
- umiejętność programowania w języku C.

Literatura:

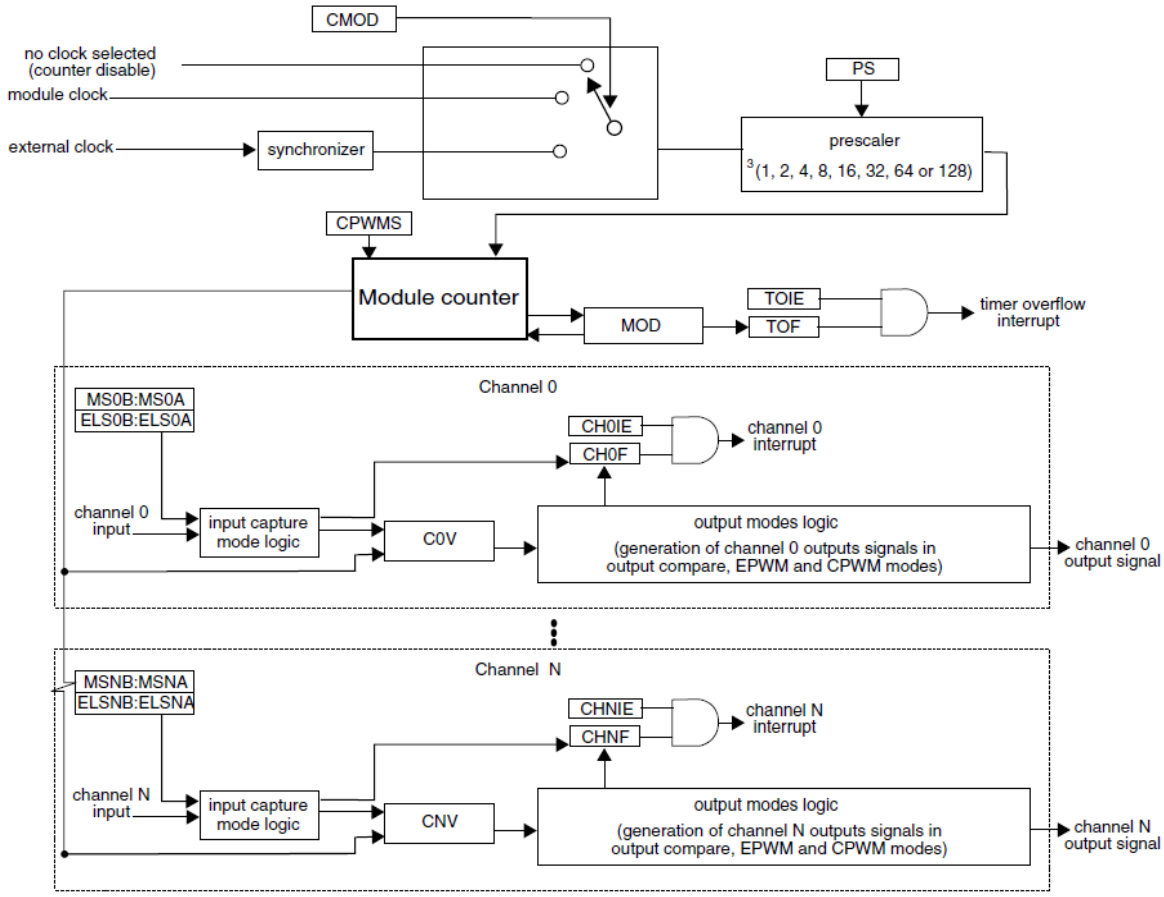
- KL05 Sub-Family Reference Manual, Freescale Semiconductor
- Kinetis L Peripheral Module Quick Reference, Freescale Semiconductor
- Joseph Yiu, The Definitive Guide to the ARM Cortex-M0, Elsevier, 2011
- ARM „Cortex-M0+ Devices Generic User Guide”
- schemat modułu głośnika ze wzmacniaczem WSR-04489

2. LICZNIK TPMX

Układ MKL05Z32VLC4, będący "sercem" zestawu FRDM-KL05Z, posiada w swoich zasobach sprzętowych, dwa, 16-bitowe liczniki:

- TPM0 - sześciokanałowy,
- TPM1 - dwukanałowy.

Schemat blokowy modułu TPMx jest przedstawiony na rysunku Rys. 1



Rys. 1 Schemat blokowy modułu TPMx

Licznik główny (Module Counter) jest wspólny dla wszystkich kanałów, i to w oparciu o jego zawartość są kreowane wszystkie tryby pracy:

- zwykły licznik, zliczający zegarowe impulsy wejściowe aż do przepełnienia,
- „Input Capture" - sygnał podany na wejście „channel N input" lub pochodzący z wyjścia komparatora CMP0 (tylko TPM1, ustawiane w SIM_SOPT4[TPM1CH0SRC]), zapamiętuje („zatrzaskuje"), aktualny stan licznika głównego, w rejestrze CNV. Aktywne może być zbocze narastające lub opadające, względnie obydwa zbocza sygnału zatrzaskującego,
- „Output Compare" - następuje ciągłe porównywanie zawartości licznika głównego z wcześniej zapisaną zawartością rejestru CNV. W momencie, gdy obydwie wartości się zrównają, na wyjściu „channel N output signal" pojawia się stan, wynikający z zaprogramowanego trybu pracy:

- ustawienie wartości przeciwnej (negacja) na wyjściu,
- ustawienie wartości „1” na wyjściu,
- ustawienie wartości „0” na wyjściu,
- wygenerowanie na wyjściu impulsu dodatniego, o długości jednego cyklu zegarowego,
- wygenerowanie na wyjściu impulsu ujemnego, o długości jednego cyklu zegarowego.

➢ PWM - szczególny przypadek „Output Compare”. Są dwa rodzaje pracy PWM:
- impuls pozycjonowany względem początku okresu,
- impuls pozycjonowany względem środka okresu.

Obydwa tryby mogą generować na wyjściu dwa rodzaje polaryzacji sygnału:

✓ „High-true pulses” - zerowanie sygnału wyjściowego („channel N output signal”), w przypadku zrównania się wartości licznika głównego z zawartością rejestru CNV, a ustawianie na wartość „1”, w przypadku początku okresu (przeładowanie licznika głównego, po doliczeniu do wartości maksymalnej - patrz rejestr MOD).

✓ „Low-true pulses” - ustawianie wartości „1” na wyjściu („channel N output signal”), w przypadku zrównania się wartości licznika głównego z zawartością rejestru CNV, a zerowanie, w przypadku początku okresu (przeładowanie licznika głównego, po doliczeniu do wartości maksymalnej - patrz rejestr MOD).

Wejście „channel N input” i wyjście „channel N output signal” to jedna i ta sama końcówka, której rola zależy od wybranego trybu pracy.

Licznik główny może liczyć „w przód” lub „w przód, a następnie w tył”. W ćwiczeniu będzie wykorzystywany ten pierwszy przypadek.

Istnieją cztery źródła sygnałów zegarowych:

- zewnętrzny (TPM_EXTCLK, ustawiany w TPMx_SC[CMOD], SIM_SOPT4[TPMxCLKSEL]),
- wewnętrzne (ustawiane w SIM_SOPT2[TPMSRC]):
  - MCGFLLCLK (używany w niniejszym ćwiczeniu),
  - OSCERCLK,
  - MCGIRCLK.

Sygnał zegarowy, po przejściu przez dzielnik (Prescaler), jest zliczany przez licznik główny. Wyzwolenie zliczania może odbyć się na parę sposobów:

- programowo - wyzwalanie sprzętowe zablokowane (TPMx_CONF[CSOT=0]), licznik główny włączony (TPMx_SC[CMOD=1] {2 dla zegara zewnętrznego})
- sprzętowo - wyzwalanie sprzętowe odblokowane (TPMx_CONF[CSOT=1]), licznik główny włączony (TPMx_SC[CMOD=1] {2 dla zegara zewnętrznego}).

Źródłem wyzwalania sprzętowego może być (zbocze narastające):

- sygnał zewnętrzny, podawany na wejście EXTRG_IN,
- CMP0,
- PITx,
- TPMx (overflow),
- RTC,
- LPTMR.

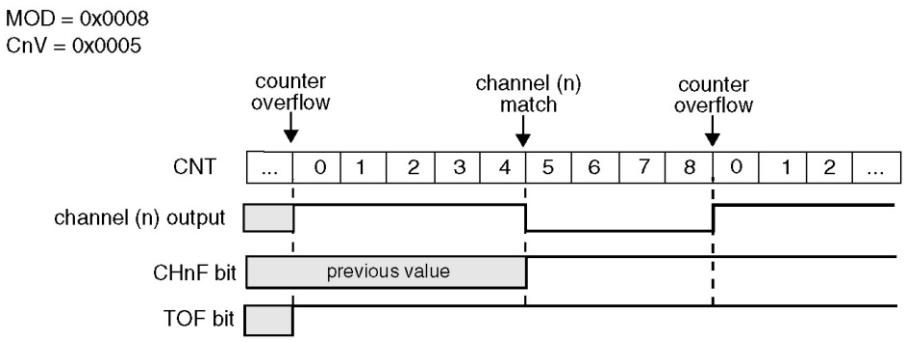
Źródło wyzwalania sprzętowego wybiera się w rejestrze TPMx_CONF[TRGSEL].

Rejestr MOD służy do „skracania” 16-bitowego licznika głównego. Ponieważ licznik główny liczy od wartości 0000 do wartości maksymalnej (przepełnienia), określonej przez rejestr MOD, więc w trybie zliczania „w przód” może zliczyć maksymalnie MOD+1 impulsów wejściowych. W trybie zliczania „w przód, a następnie w tył” może zliczyć maksymalnie 2·MOD impulsów wejściowych.

Przepełnienie licznika głównego (Overflow) oraz zdarzenia w poszczególnych kanałach mogą być źródłem przerwań. Rządzi nimi jeden wektor, wspólny w ramach jednego modułu TPMx. Dlatego, aby dowiedzieć się, które przerwanie było aktywne, należy przeanalizować zawartość rejestru TPMx_STATUS, który zawiera wartości wszystkich flag przerwania: od licznika głównego i wszystkich aktywnych kanałów. Po ustaleniu źródła przerwania, należy programowo skasować odpowiedni bit w powyższym rejestrze, poprzez wpisanie wartości „1” na danej pozycji.

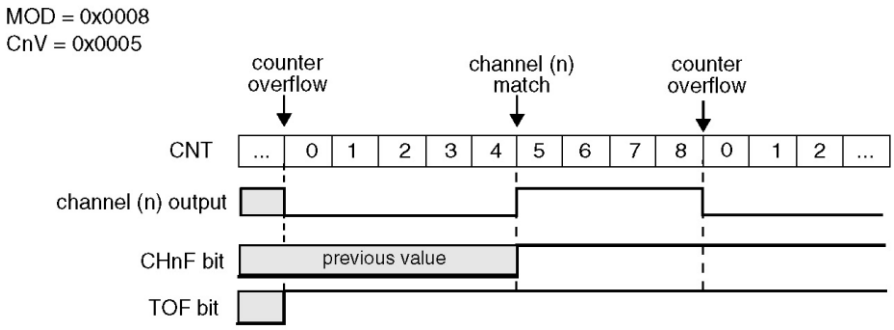
2.1.PWM

W niniejszym ćwiczeniu skupimy się na trybie pracy z pozycjonowaniem impulsu PWM względem początku okresu (Edge-aligned PWM). Jak już wcześniej wspominano, w trybie tym można wybrać jedną z dwóch polaryzacji sygnału wyjściowego. Na rysunku Rys. 2 pokazano zależności czasowe pomiędzy wartością rejestru MOD, CNV, zawartością licznika głównego i wyjściem, dla polaryzacji dodatniej („High-true pulses”). Proszę zwrócić uwagę na zachowanie flag przepełnienia (TOF) licznika głównego i zdarzenia „kanałowego” CHnF.



Rys. 2 Edge-aligned PWM - High-true pulses

Na rysunku Rys. 3 pokazano zależności czasowe pomiędzy wartością rejestru MOD, CNV, zawartością licznika głównego i wyjściem, dla polaryzacji ujemnej („Low-true pulses”).



Rys. 3 Edge-aligned PWM - Low-true pulses.

Analizując powyższe przebiegi czasowe, można zauważyć, że aby uzyskać wypełnienie 100%, należy do rejestru MOD wpisać wartość o 1 mniejszą, niż maksymalna wartość, która może się pojawić w CNV. W stosunku do trybu „Low-true pulses" współczynnik wypełnienia liczy się dla impulsu, który ma wartość „0" (polaryzacja ujemna).

Programowanie trybu PWM zostanie opisane na przykładzie kanału 3, licznika TPM0. Wyjście tego kanału wyprowadzone jest na końcówkę PTB8, która jest równocześnie podłączona do katody czerwonej diody LED. Taki sposób podłączenia diody pociąga za sobą pewne wymogi, jeżeli chodzi o polaryzację sygnału wyjściowego. Chcielibyśmy, aby jasność świecenia diody rosła wraz ze wzrostem wartości, wpisywanej do rejestru C3V (dana pobierana z pola dotykowego - Slider). Powyższe postulaty spełni tryb „Low-true pulses", ponieważ stan niski (który zaświeca diodę) przebiegu PWM jest proporcjonalny do zawartości C3V. Kolejność programowania wygląda następująco:

1. włączyć „zasilanie" portu B, w SIM_SCGC5[PORTB=1] - wszystkie diody LED, a tym samym kanały PWM, są dołączone tylko do tego portu,
2. zdefiniować rolę końcówki PTB8 jako TPM0_CH3, w PORTB_PCR[8][MUX=2],
3. włączyć „zasilanie" licznika TPM0, w SIM_SCGC6[TPM0=1],
4. wybrać źródło zegara taktującego licznik główny TPM0, w SIM_SOPT2[TPMSRC=1] - MCGFLLCLK=41943040Hz,
5. ustawić tryb zliczania „ w przód", licznika TPM0, w TPM0_SC[CPWMS=0] (jest to wartość domyślna po RESET, więc można ten krok pominąć),
6. wybrać wartość podzielnika sygnału zegarowego (tego z pkt. 4), w TPM0_SC[PS=6] (dzielnik zegara wejściowego równy 64; zegar = 655360Hz),
7. ustawić wartość rejestru MOD, w TPM0_MOD = 0xFFFF -
8. ustawić tryb PWM „Low-true pulses", kanału nr 3, TPM0, w TPM0_C3SC[MSB=1 i ELSA=1],
9. ustawić wartość początkową rejestru C3V (np. 0000), w TPM0_C3V=0 (dioda zgaszona),
10. włączyć licznik TPM0, w TPM0_SC[CMOD=1].

Od tego momentu na końcówce PTB8 pojawi się przebieg PWM („Low-true pulses"), pozycjonowany względem początku okresu, o wypełnieniu proporcjonalnym do zawartości rejestru C3V (współczynnik wypełnienia będzie dotyczył impulsu „ujemnego", czyli wartości „0" w stosunku do całego okresu).

Jak określić częstotliwość sygnału wyjściowego?

Licznik główny TPM0 zlicza impulsy zegara, który wybrano w SIM_SOPT2[TPMSRC=1], czyli 41943040Hz (Core Clock), a jego częstotliwość została podzielona prze wartość, określoną w TPM0_SC[PS=6], czyli 64. Daje to w efekcie przebieg o częstotliwości 655360Hz. W trybie liczenia „do przodu", licznik zliczy MOD+1 takich „podzielonych" impulsów zegarowych. Częstotliwość tak powstałego przebiegu to:

$$f = \frac{655360}{MOD + 1}\ [Hz]$$

Ogólnie, dla różnych podzielników zegara Core Clock można zapisać:

$$f = \frac{Core\ Clock}{2^{PS}(MOD + 1)}\ [Hz]$$

Gdyby odwrócić sytuację i chcieć zapytać: jaką wartość należy załadować do MOD, aby otrzymać częstotliwość f, to wzór miałby postać:

$$MOD = \frac{Core\ Clock}{f 2^{PS}} - 1$$

Wzór ten przyda się, gdy będziemy wyliczać temperowaną skalę muzyczną dla dowolnej oktawy.

Jak określić wartość średnią napięcia wyjściowego PWM?

Średnia wartość napięcia wyjściowego jest zależna od maksymalnej wartości, która odpowiada stanowi „1" na wyjściu pinu (u nas 3.0V) oraz współczynnika wypełnienia k, i wyraża się wzorem:

$$U_{śr} = 3V\,\frac{100\% - k[\%]}{100\%}$$

Proszę zauważyć, że wybrany jest tryb PWM „Low-true pulses". Dlatego dla k=100% napięcie średnie jest równe 0V (impuls „0" trwa 100% czasu, czyli cały okres), a średnia z wartości 0 to 0.

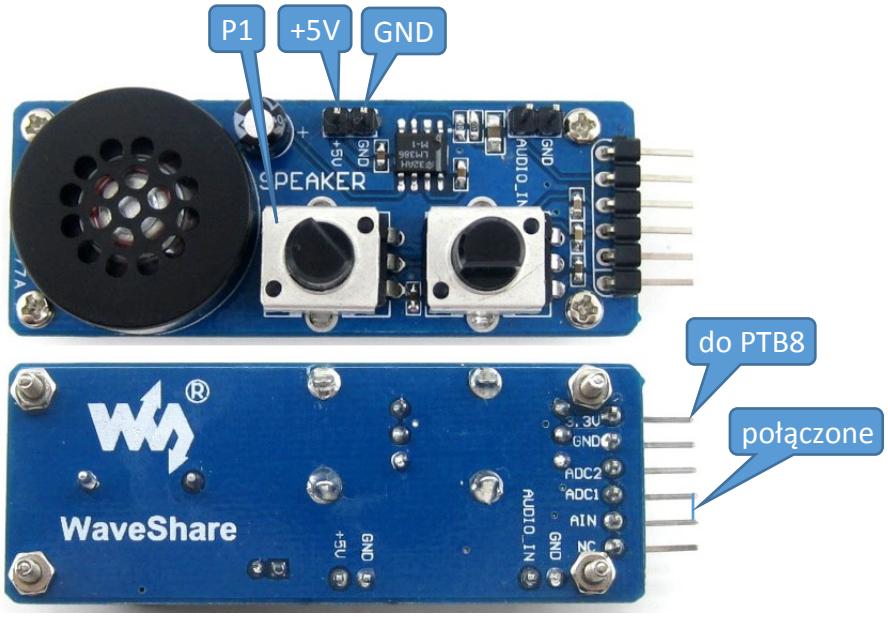
3. MODUŁ GŁOŚNIKA WSR-04489

Jeśli do końcówki PTB8 (dioda czerwona LED) dołączymy dodatkowo głośnik (Rys. 4), to oprócz efektów wizualnych będziemy mieć również dźwiękowe. Układ podłączamy do płytki KL05Z wg zaleceń z tabeli Tab. 1.

FRDM-KL05	**WSR-04489**
PTB8	VCC (3.3V)
+5V	+5V
GND	GND

Tab. 1 Tabela połączeń modułu WSR-04489 z KL05Z

Dodatkowo, na module WSR-04489, łączymy pojedynczym kabelkiem pin ADC1 z pinem AIN. Pozwoli to na regulację głośności sygnału dźwiękowego, dobieraną obrotowym potencjometrem P1.



Rys. 4 Głośnik WSR-04489 i jego podłączenie do systemu

Końcówka, która jest dołączana do PTB8, może mieć różne nazwy: „3.3V” lub „(VCC 2.5-5.5V) VCC”. W razie wątpliwości, należy popatrzeć na rysunek Rys. 4.

4. System równomiernie temperowany

Inaczej strój muzyczny. W (dwunastotonowym) systemie równomiernie temperowanym stosunek częstotliwości dwóch kolejnych dźwięków wynosi $\sqrt[12]{2}$, gdyż system ten zakłada podział oktawy na 12 równych części. Jako dźwięk podstawowy przyjęto „a1”, z oktawy razkreślnej, którego częstotliwość wynosi 440Hz. Na jego podstawie można określić częstotliwości wszystkich dźwięków, wszystkich oktaw. Sąsiednie dźwięki różnią się od siebie o pół tonu. Odległość (interwał) pomiędzy dźwiękami naturalnego szeregu diatonicznego wynosi cały ton dla par c – d, d – e, f – g, g – a i a – h. Odległość pomiędzy parami e – f i h – c wynosi półtonu.

Oktawa razkreślna

Dźwięk	częstotliwość cykli na sekundę
c^1	261,6
d^1	293,7
e^1	329,6
f^1	349,2
g^1	391,9
a^1	440,0
h^1	493,9
c^2	523,2

5. Ćwiczenie 1

Ponieważ będziemy wykorzystywać wyświetlanie liczb zmiennoprzecinkowych, została powiększona wielkość stosu, w zbiorze *startup_MKL05Z4.s*.

```
; <h> Stack Configuration
;   <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
; </h>

Stack_Size      EQU     0x00000300
```

Rozpakować zbiór *1_LED_PWM.zip*. Uruchomić projekt *LED_PWM.uvprojx*. Na diodę czerwoną LED zostaje podany sygnał PWM o współczynniku wypełnienia k=50% i częstotliwości f=10Hz. Za pomocą pola dotykowego można modyfikować wartość k. Dla małych wartości k, dioda wyraźnie mruga, jednak przygaszona. Dla dużych wartości k jasność jest większa, a mruganie ledwo dostrzegalne. Dzieje się tak dlatego, że dla bardzo dużych wartości k dioda przez większość czasu przewodzi, a tylko na krótko gaśnie. Czas stanu wyłączenia diody jest na tyle krótki, że nasze oko ledwo zauważa zmiany. Natomiast dla małych wartości k jest na odwrót. Dioda tylko na krótką chwilę zapala się, a większość czasu jest zgaszona i widzimy wyraźne mruganie.

Klawisz S2 ustawia częstotliwość sygnału PWM na 10Hz, a klawisz S3 na ok. 1002Hz. W przypadku częstotliwości 1002Hz nie obserwujemy już mrugania dla żadnej wartości k.

5.1. Zadanie

- Wyjaśnić, dlaczego w przypadku f=1002Hz nie obserwujemy mrugania, bez względu na wartość k?
- Wiedząc, że dioda niebieska LED jest podpięta do pinu PTB10, a ten do kanału nr 1 TPM0, dopisać obsługę klawisza S4, który będzie włączał sygnał PWM kanału nr 1 TPM0 do diody niebieskiej LED. Sygnał ma mieć częstotliwość ok. 100Hz i współczynnik wypełnienia regulowany jak dla pozostałych (wspólnie). W zbiorze *TPM.c* jest już wszystko ustawione dla tego kanału (jak również dla diody zielonej LED).

6. Ćwiczenie 2

Podłączyć głośnik jak w punkcie 3. Rozpakować zbiór *2_Gama_PWM.zip*. Uruchomić projekt *Gama_PWM.uvprojx*. Program odgrywa dźwięki podstawowe (bez półtonów) ośmiu oktaw. Klawisz S2 zwiększa numer oktawy, a klawisz S3 zmniejsza. Kolejne wciskanie klawisza S4 przełącza kolejne dźwięki w oktawie. Na wyświetlaczu jest wyświetlana wartość częstotliwości aktualnie generowanego dźwięku. Potencjometrem P1 należy wyregulować głośność odtwarzanych dźwięków.

6.1. ZADANIE

- Przeanalizować sposób generacji poszczególnych częstotliwości, w programie *main.c*.
- Pobawić się, nie wkurzając bliskich i sąsiadów.

7. ĆWICZENIE 3

Rozpakować zbiór *3_Kotek_PWM.zip*. Uruchomić projekt *Kotek.uvprojx*. Potencjometrem P1 należy wyregulować głośność odtwarzanych dźwięków. Program gra utwór „Wlazł kotek na płotek". Klawiszem S4 można zatrzymać odtwarzanie lub je wznowić. Klawisz S2 zwiększa numer oktawy, a klawisz S3 zmniejsza. Licznik SysTick jest wykorzystywany do nadawania tempa utworowi. Przyjęto, że cała nuta trwa 1s, półnuta 0.5s, ćwierćnuta 0.25s, a ósemka 0.125s. Są również generowane pauzy po każdej nucie. Wszystkim tym rządzi SysTick. Natomiast samym dźwiękiem (jego wysokością) steruje licznik TPM0. W tej wersji są również obecne półtony.

7.1. ZADANIE

- Grać aż się znudzi.
- Zlikwidować znak komentarza linii „// Fuga", a linie „// Kotek" zabezpieczyć znakiem komentarza. Skompilować program, uruchomić i cieszyć się Fugą Jana Sebastiana Bach'a.
- Spróbować „rozgryźć", jak są zapisywane nuty, wraz z pauzami.