Mikrokontrolery ARM

Laboratorium 2

Krzysztof Pierczyk

17 kwietnia 2020

Wstęp

Na kolejnych zajęciach laboratoryjnych zajęliśmy się układami czasowo-licznikowymi obecnymi w układach STM32. Przygotowane zadania wymagały wykorzystania większości funkcjonalności oferowanych przez te układy począwszy od możliwości cyklicznego zgłaszania przerwań, przez generowanie przebiegów prostokątnych, na pomiarze czasu trwania impulsu kończąc. Nie wymuszały one skorzystania z mechanizmów synchronizacji liczników czy użycia zewnętrznych sygnałów taktujących, ale i te moduły można było przy odrobinie wysiłku zastosować.

Zadanie 1

Pierwsze z zadań polegało na generowaniu przez układ czasowo-licznikowy TIM1 przerwań, które zmieniałyby stany diod LD3 – LD6 w taki sposób, aby uzyskać efekt „biegającego punktu”. Przerwania te miały być generowane z okresem 200m. Główną część programu stanowiła sama konfiguracja układu TIM1. Jako, że pozostałe elementy, wykorzystywane przez program, stanowiły część poprzednich zajęć laboratoryjnych, poniżej umieszczono jedynie ten fragment kodu.

Procedura obsługi przerwania (*ang. Interrupt Service Routine, ISR*) umieszczona w wektorze tim1\_up\_tim10\_isr\_vector sprawdza się do sprawdzenia wartości licznika (inkrementowanego co przerwanie mod(3) ), zgaszeniu diody poprzedniej i zapaleniu następnej. Obsługa portów GPIO odbywa się poprzez API systemu ISIX.

W ramach ciekawostki warto zwrócić uwagę na sytuację, która przydarzyła się w czasie pisania programu do zadania pierwszego. Przy pierwszym podejściu do pisania udało się w poprawny sposób skonfigurować układ licznikowy, lecz program zachowywał się tak, jak gdyby wyzwalane były zawsze dwa przerwania z rzędu, tj. aktywowane były tylko dwie wybrane diody, a okres ich świecenia równy był przewidywanym 200ms. Po długich zmaganiach przyczyną błędu okazało się **zbyt późne gaszenie flagi przerwania generowanej wewnątrz modułu TIM1**.



Jak wiadomo, flagi przerwań w kontrolerze NVIC gaszone są automatycznie w ramach obsługi procedury przerwania. Jednak flagi generowane wewnątrz układów peryferyjnych mikrokontrolera gaszone automatycznie nie są. Okazuje się, że jeśli instrukcja czyszczenia flagi zostanie wykonana zbyt późno, wyjście z procedury obsługi przerwania może nastąpić przed następnym cyklem odpowiedniej szyny danych co spowoduje, że NVIC potraktuje starą wartość flagi jako wystąpienie kolejnego przerwania i rozpocznie ponowne wykonanie ISR. Zatem jedną z dodatkowych nauk płynących z tych zajęć laboratoryjnych jest to, że flagi przerwań należy gasi jak najwcześniej (przynajmniej w przypadku mikrokontrolerów z rodziny STM32).

Zadanie 2

Zadanie drugie wymagało nie tylko skonfigurowaniu układu podstawy czasu licznika TIM4, ale również modułu *Capture/Compare* w taki sposób, aby generował on na pinach podłączonych do diod LED sygnał prostokątny o częstotliwości 1Hz oraz wypełnieniu równym 50%. Sygnał taki miał pojawić się na diodach LED4, LED3 oraz LED5 i w każdym przypadku miał być przesunięty w fazie o kolejno 0°, 180° oraz 270°.

Aby urozmaicić zadanie i poprawić efekty wizualne będące wynikiem działania programu do tego zestawu dołączono diodę LED6 oraz zmieniono konfigurację przesunięć fazowych. W efekcie otrzymano:

* Przesunięcie 0° na diodzie LED3
* Przesunięcie 90° na diodzie LED5
* Przesunięcie 180° na diodzie LED6
* Przesunięcie 270° na diodzie LED4

Konfiguracja taka skutkuje powstanie efekt „biegającego punktu” z „zakładkami”, tj. z momentami, w których dwie sąsiednie diody świecą się jednocześnie. Okresy te równe są połowie czasu świecenia dioda, a zatem ¼ okresu sygnału prostokątnego.

Ponownie fragment kodu odpowiedzialnego za konfigurację układu czasowo licznikowego został przedstawiony na poniższym listingu. Zarówno do inicjalizacji układu podstawy czasu jak i modułu Capture/Compare wykorzystane zostały struktury inicjalizacyjne z biblioteki *Low Level Library* (LL).

Podstawa czasu została skonfigurowana tak, aby **przepełnienie następowało z częstotliwością 2Hz**, co w połączeniu z trybem **trybem *Toggle on Match***modułu C/C pozwoliło uzyskać generację fal prostokątnych o częstotliwości 1Hz. Kanały podpięte do diod LED3 i LED5 zostały ustawione na **polaryzację prostą**, a te podłączone na diod LED 6 i LED4 na **polaryzację odwróconą**. Po ustawieniu wartości rejestrów CCRx dla diod LED3 i LED5 na 0, a dla diod LED6 i LED4 na połowę wartości rejestru ARR zagwarantowało to pożądane przesunięci fazowe.



Zadanie 3

Zadanie trzecie również poruszało kwestię generacji fali prostokątnej z użyciem układu TIM4, jednak tym razem nacisk położony został na manipulację stopniem wypełnienia fali, a nie jej przesunięciem fazowym. Zgodnie z treścią zadania 3 z 4 diod LED zostały zasilone falą prostokątną o wypełnieniu kolejno 20%, 40% oraz 60%. Ostatni z diod została zainicjalizowana sygnałem prostokątnym o wypełnieniu 0 (tj. stałą wartością napięcia równą 0).

W sposób analogiczny do przedstawionego w sprawozdaniu z laboratorium pierwszego skonfigurowano przerwania od przycisku USER poprzez moduł przerwań zewnętrznych *EXTI*. Procedura obsługi przerwania została wykorzystana do realizacji programowej eliminacji drgań styków. Wykrycie faktycznego wciśnięcia przycisku powoduje zwiększenie wartości rejestru CCR4 o 50, co stanowi 1/10 wartości rejestru ARR powiększonej o 1. W praktyce skutkuje to **zwiększeniem procentowego wypełnienia sygnału prostokątnego o 10 punktów procentowych**. Gdy liczba w rejestrze osiągnie wartość (ARR + 1), kolejne wciśnięcia przycisku skutkują obniżeniem wartości o 50. Kod wykorzystany w czasie realizacji tego zadania jest analogiczny do kodu z zadania drugiego poza dwiema różnicami:

* Polaryzacje wszystkich kanałów *CO* zostały ustawione na proste
* Tryb pracy wyjść kanałów zmieniono z *Toggle on Match* na *PWM1*/*PWM2*

# Wnioski dotyczące trybów PWM

Badania przeprowadzono przy ustawieniu modułu czasowo-licznikowego w trybie zliczania do góry. W takim przypadku oba tryby PWM wykazują identyczne możliwości w kontekście sterowania średnią wartością napięcia generowaną na obciążeniu.

* Z użyciem trybu PWM1 można osiągnąć wartości wypełniania z zakresu [0%; 100%] manipulując wartością rejestru CCRx pomiędzy wartościami 0 i (ARR+1) zgodnie ze wzorem
* W trybie PWM2 powyższa zależność jest odwrócona – wypełnienie 0% uzyskujemy przy CCRx ≥ ARR + 1, a 100% przy CCRx = 0. Poziom wypełnienia określa wzór

Oczywiście w obu przypadkach, gdy wartość ARR będzie wynosiła (lub dla układów TIM2 i TIM5) wpisanie do CCRx wartości (ARR + 1) nie będzie możliwe, zatem w trybie PWM1 nie będzie możliwe osiągnięcie wypełnieni 100%, a w trybie PWM2 wypełnienia 0%. Rozdzielczość wypełnienia jest niezależna od trybu i wynosi

W trybie zliczania *Center-aligned* rozdzielczość ta jest dwukrotnie zmniejszona, natomiast niezależnie od wartości ARR możliwe jest osiągnięcie dowolnego wypełnienia fali. Wzory to opisujące kolejno dla trybu PWM1 i PWM2 prezentują się następująco:

Zadanie 4

Ostatnie z zadań poruszało kwestię wykorzystania modułu *C/C* do pomiaru długości trwania impulsów, a konkretnie pomiaru częstotliwości fali prostokątnej podanej na jeden z pinów mikrokontrolera. Ze względu na brak dostępu do sprzętu laboratoryjnego, w tym generatora sygnałów, **kanał wejściowy CH1 układu TIM3 został podłączony do wyjścia CH2 układu TIM5** skonfigurowanego w trybie PWM1.

Preskaler licznika TIM5 ostawiony został tak, aby generować sygnał taktujący o częstotliwości 10MHz. Pozwala to na regulację częstotliwości sygnału prostokątnego w zakresie [152.59Hz; 5MHz]. Aby kontrolować częstotliwość sygnału PWM skonfigurowano również moduł przerwań *EXTI* do obsługi przycisku USER (z uwzględnioną programową procedura eliminacji drgań styku). Wykryte wciśnięcia powodują zmianę wartości w rejestrze ARR na jedną z trzech predefiniowanych. Wartości te określają **częstotliwości wyjściowej fali prostokątnej na poziomie 10kHz, 100kHz oraz 1MHz**. Przy każdej zmianie częstotliwości modyfikowany jest również rejestr CCR1 tak, aby utrzymać wypełnienie na poziomi 50%.

Konfigurację układu TIM3 rozpoczęto od ustawienia podstawy czasu na maksymalną możliwą częstotliwość, tj. 100MHz przy jednoczesnym ustawieniu wartości rejestru ARR na maksymalną (). Taki dobór parametrów pozwala osiągnąć największą dostępną rozdzielczość pomiaru na poziomie 10ns przy pomiarze okresu sygnału.



Następnym krokiem było uruchomienie modułu *C/C* w trybie *Input Capture* tak, aby obserwował on zmiany poziomu logicznego na kanale CH1. Przy wykryciu zbocza narastającego zapisuje on aktualną wartość rejestru CNT w rejestrze CCR1. Filtr sygnału wejściowego został wyłączony, aby uzyskać maksymalną dostępną rozdzielczość.

Aby pomiar okresu sygnału wejściowego nie był obciążony opóźnieniami programowymi**, skonfigurowano również moduł synchronizujący w trybie *Reset***. Sygnał *Trigger*, skonfigurowany jako wejście kanału CH1, powoduje dzięki temu nie tylko zapisanie wartości CNT do rejestru CCR1, ale również zresetowanie licznika. Dzięki temu, pomiar okresu realizowany jest **całkowicie sprzętowo**, a w rejestrze CCR1 znajduje się zawsze ostatnia zmierzona wartość, która może następnie posłużyć do wyznaczenia częstotliwości sygnału **poza procedurą pomiaru**.

Układ TIM1 skonfigurowany został tak, aby wyzwalać przerwania z okresem 1s. Procedura obsługi tego przerwania dokonuje odczytu wartości rejestru CCR1 i obliczenia na jej podstawie częstotliwości sygnału zgodnie ze wzorem

a następnie wypisuje rzeczywistą wartość częstotliwości i wartość obliczoną na podstawie pomiaru.

Poniższa tabela przedstawia wyniki pomiarów trzech częstotliwości testowych oraz związanych z nimi wartości błędów. Wartości pomiarów zostały uśrednione z 20 kolejnych próbek.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Częstotliwość  sygnału | Średnia wartość mierzona | Niepewność pomiarowa | Błąd bezwzględny [Hz] | Błąd względny [%] |
| 10 kHz |  |  |  |  |
| 100 kHz |  |  |  |  |
| 1 MHz |  |  |  |  |