

Celem czwartego zadania laboratoryjnego było zaprojektowanie, złożenie, zaprogramowanie i przetestowanie układu z mikrokontrolerem MSP430F16x tak aby działał on jako miernik czasu reakcji – jeden przyciski rozpoczyna pomiar, drugi kończy pomiar. Zasada działania podobna jak w przypadku zwykłych stoperów. Wartość mierzonego czasu powinna być na bieżąco wyświetlana na wyświetlaczu dynamicznym, który obowiązkowo obsługiwany jest za pomocą przerwań (jedno przerwanie - jedno przełączenie pozycji).

UWAGI:

Częstotliwość odświeżania powinna gwarantować wyświetlenie każdej cyfry co najmniej 50 razy na sekundę. Do otrzymania satysfakcjonującej dokładności pomiaru długości trwania impulsów wymagane jest użycie trybu CAPTURE i obliczanie interwału z uwzględnieniem odczytów licznika przy zboczu początkowym i końcowym oraz liczby przerwań TxIFG, jakie zarejestrowano pomiędzy tymi zboczami. Kod programu został odpowiednio podzielony na część realizującą obsługę sprzętową i część aplikacyjną. Interakcję z systemem wzbogaciliśmy o wyświetlanie zer przy starcie systemu i przecinka, oddzielającego liczbę milisekund od sekund.

OPIS SPRZĘTU

Dzięki strukturze mikrokontrolera MSP43016x podłączenie urządzeń peryferyjnych nie było skomplikowanym zadaniem, jednak wymagało uwagi, aby nie pomylić portów obsługujących wyświetlacz.

///OBRAZEK///

Do portu pierwszego podłączyliśmy monostabilne wyłączniki (120_IN8), które posłużyły nam jako źródło sygnałów wejściowych, odpowiednio dla przycisku start – P1.2, stop – P1.3. Dla wyświetlacza zostały zarezerwowane porty P2.0 – P2.7 (obsługa drivera) i P3.0 – P3.7 (wybór wyświetlacza 7-segmentowego). W Timer_A wykorzystujemy wszystkie trzy kanały dwa w trybie capture i jeden w trybie compare.

OPIS OPROGRAMOWANIA

Do zmniejszenia poboru energii użyliśmy trybu LPM0, ponieważ tylko on gwarantował nam brak zmian częstotliwości pracy zegara SMCLK. Kod programu napisaliśmy w języku C, co było niewątpliwym ułatwieniem w porównaniu do poprzednich laboratoriów. Pozwoliło to na lepsze uporządkowanie funkcjonalności i zadań do wykonania w naszym programie. Kod stał się bardziej przejrzysty i zrozumiały.

Zliczanie czasu

Naciśnięcie przycisku może nastąpić w momencie gdy jesteśmy na zboczu zegara, dlatego do dokładnego obliczania zmierzonego czasu zastosowaliśmy przerwania w trybie capture, dzięki którym jesteśmy w stanie określić pozycję w jakiej byliśmy na tym zboczu. Zbocza liczników są opadające. Po uruchomieniu systemu i naciśnięciu przycisku – start, stoper zaczynał odliczanie. Zapamiętujemy miejsce na zboczu i zaczynamy zliczać ilość przepełnień licznika w celu obliczenia zmierzonego czasu. Przy naciśnięciu przycisku – stop, znowu zapamiętujemy miejsce na zboczu i używając wcześniej zapisanych danych obliczamy dokładny czas do wyświetlenia (dodajemy: czas od naciśnięcia przycisku start do przepełnienia, ilość przepełnień licznika, czas od naciśnięcia przycisku stop do przepełnienia) ??? TO W NAWIASIE CHYBA ZBĘDNE UZUPEŁNIENIE???

OBSŁUGA WYŚWIETLACZ

Przy odświeżaniu wyświetlacza stosowaliśmy przerwania w trybie compare, generowane z tego samego Timera_A co przerwania w trybie capture. Ustawiona częstotliwość odświeżania to 720 kHz. Najpierw wybieramy wyświetlacz, który będziemy używać i potem dokonujemy wpisania wartości, która ma być wyświetlona.

???Po naciśnięciu start (zliczanie czasu uruchomione) z każdym przerwaniem od compare dodajemy wartość do zmiennej, która jest na bieżąco wyświetlana. Mamy świadomość, że nie jest to idealny czas, ale różnice są rzędu ms, czyli nie zdążymy nawet tego zauważyć. ???W dodatku liczba milisekund zmienia się tak szybko, że ciężko jest zaobserwować jak jest liczba na miejscu milisekund ??? Dokładnie wyliczona wartość (z użyciem przerwań capture) jest wyświetlana po naciśnięciu przycisku – stop, czyli po zatrzymaniu stopera. Dzięki temu użytkownik jest pewien, że otrzymuje dokładnie zmierzony czas. ???

Aby czas wykonania przerwania nie był za długi przy wpisywaniu dokładnej wartości do tablicy stosujemy dzielenie modulo przez 10, co zapewnia nam satysfakcjonujące efekty i pozwala na szybkie wykonanie przerwania. Dodatkowo wyświetlanie wartości aktualnej jest aktualizowane co drugie przerwanie, żeby maksymalnie, oczywiście w granicach normy skrócić czas wykonania przerwania.