

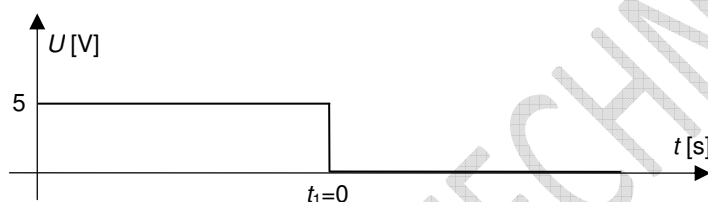
Miroslaw Łazoryszczak

LABORATORIUM nr 3

Temat: Programowa eliminacja drgań zestyków

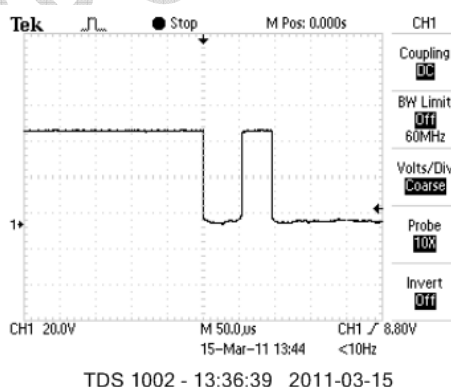
1. PROBLEM DRGAŃ ZESTYKÓW

W ćwiczeniu nr 2 zadanie polegało na obsłudze wyświetlacza siedmiosegmentowego z elementarną obsługą przycisku. Na płytce uruchomieniowej ZL2MCS51 przyciski dołączone są do portu P3. Jeśli dany przycisk jest zwolniony („wyciśnięty”), to na odpowiadającym mu porcie mikrokontrolera panuje stan wysoki. Po wciśnięciu przycisku w chwili $t_1 = 0$ na wejściu mikrokontrolera powinien pojawić się natychmiast stan niski (rys. 1).



Rys. 1. Wykres czasowy napięcia na wejściu mikrokontrolera w chwili wciśnięcia przycisku idealnego.

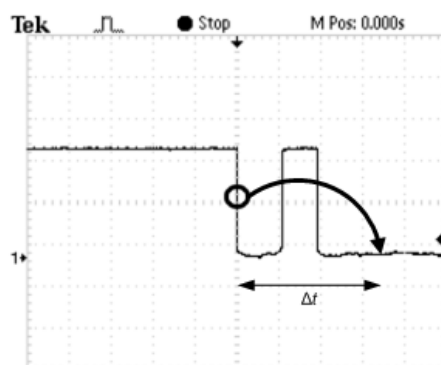
Niestety w praktyce sytuacja występuje rzadko. Najczęściej procesowi włączenia bądź wyłączenia przycisku towarzyszą oscylacje, które mogą powodować błędną interpretację klawiatury. Zjawisko to nazywane jest drganiem zestyków. W ćwiczeniu realizowanym w ramach zajęć laboratoryjnych drgania zestyków przycisków mogą powodować wielokrotne zwiększenie stanu wyświetlacza przy pojedynczym wciśnięciu przycisku. Przykładowy przebieg czasowy towarzyszący wciśnięciu klawisza przedstawiono na rys. 2. Przebieg ten został zarejestrowany za pomocą oscyloskopu w rzeczywistym układzie ZL2MCS51.



Rys. 2. Przebieg czasowy drgań zestyków

Drgania zestyków są uciążliwe w przypadku tak mało skomplikowanego pod względem funkcjonalnym układu. W sytuacji, gdy wymagana jest większa pewność, że nie nastąpi samoczynne powtórzenie akcji wywołanej wciśnięciem klawisza niezbędne jest zastosowanie mechanizmów eliminacji drgań zestyków (ang. *debounce*). Istnieje szereg metod redukcji niepożądanych oscylacji. Wyróżnić należy metody sprzętowe oraz programowe. Ponieważ zestaw uruchomieniowy ZL2MCS51 posiada przyciski, które są na stałe podłączone do portów mikrokontrolera, wariant sprzętowy eliminacji drgań zestyków jest niemożliwy do zrealizowania. W ramach metod programowych również istnieje szereg możliwości rozwiązania problemu. Ogólna idea polega na odliczeniu określonego czasu od momentu zarejestrowania zmiany stanu na porcie odpowiadającym wciśniętemu klawiszowi i ponownym sprawdzeniu wartości tego portu. Jeśli oba odczyty będą zgodne i ich stan

będzie odpowiadał niskiemu poziomowi napięcia, można uznać, że klawisz został rzeczywiście wciśnięty i można wykonać akcję odpowiadającą wciśnięciu przycisku (rys. 3). W zastosowaniach praktycznych przyjmuje się, że czas Δt wystarczający do pominięcia efektu drgań zestyków wynosi w zależności od rodzaju, jakości i stopnia zużycia przycisków od kilku do kilkudziesięciu milisekund.



Rys. 3. Idea mechanizmu eliminacji drgań zestyków

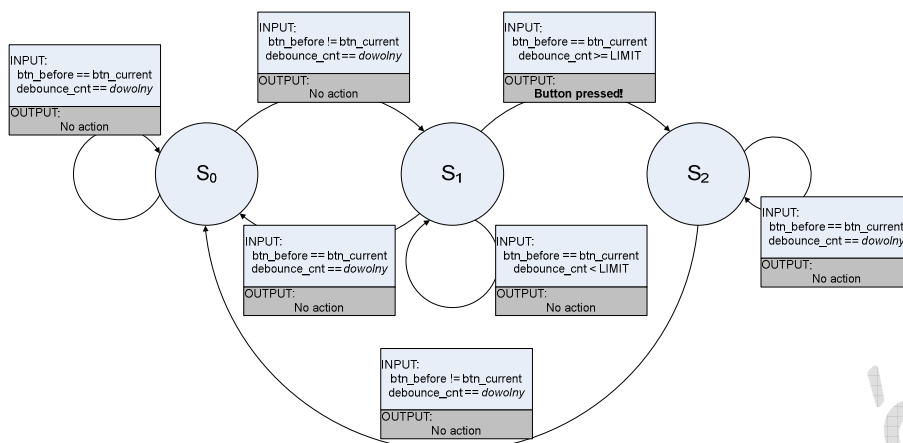
Najprostsza realizacja wynika wprost z powyższego opisu. Wymaga ona zatem sprawdzenia stanu przycisku, następnie implementacji funkcji realizującej programowe opóźnienie i ponownego sprawdzenia stanu przycisku. Zaletą tej metody jest niewątpliwie niewielki stopień komplikacji programu. Wadą natomiast fakt, że w trakcie realizowania opóźnienia procesor realizuje rozkazy, które są nieprzydatne (dodatkowo niepotrzebnie obciążające procesor) z punktu widzenia zadań programu mikrokontrolera wykonywanych np. przez pętlę główną.

Inny sposób może polegać np. na wykorzystaniu kolejnego układu czasowo-licznikowego. W przypadku, gdy pierwszy układ czasowo-licznikowy (TIMER 0) odpowiada za obsługę wyświetlacza, do obsługi drgań zestyków można wykorzystać układ drugi (TIMER 1) wraz z obsługą odpowiedniego przerwania. Algorytm mógłby wówczas zostać opisany w sposób następujący:

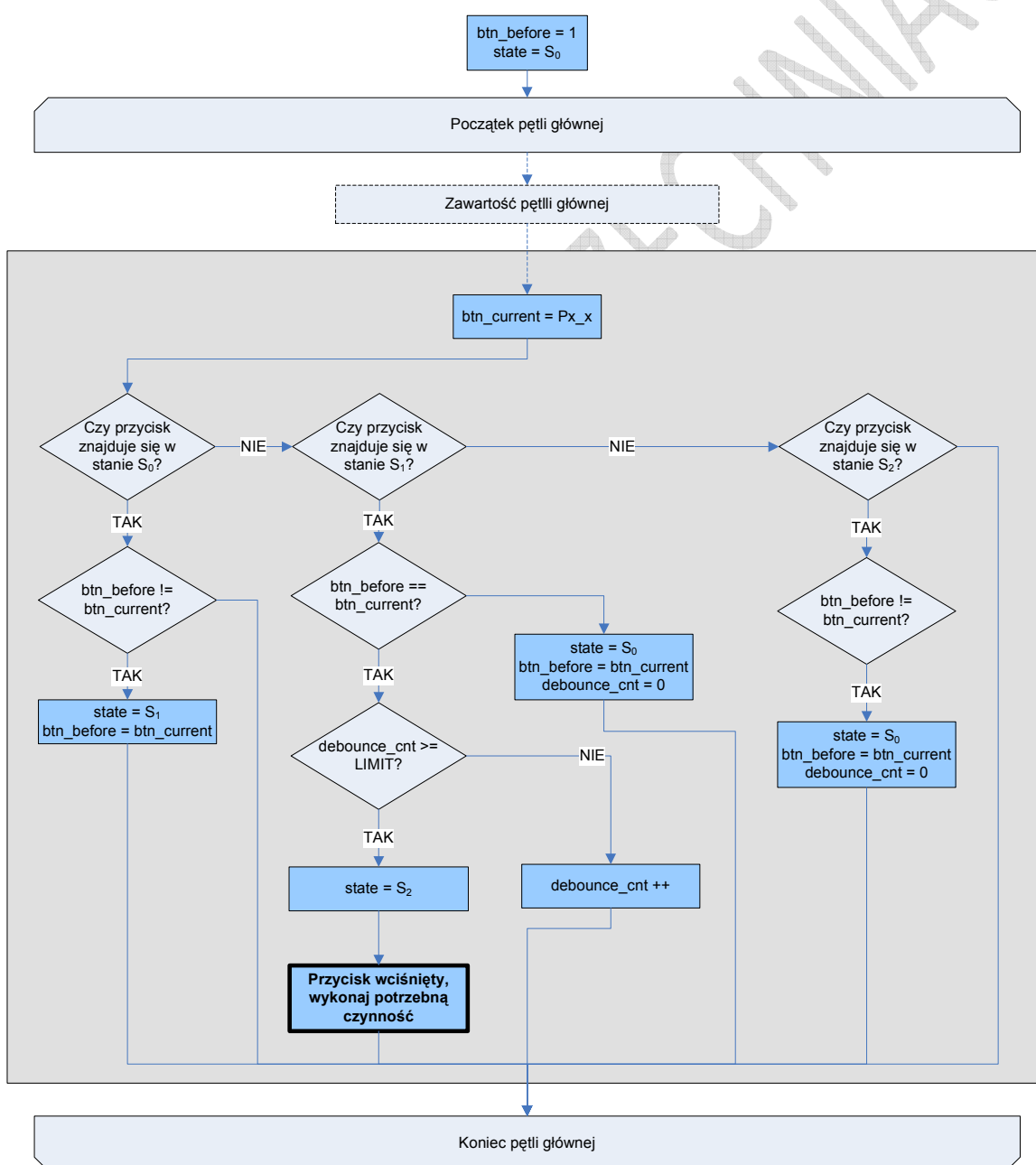
- sprawdzanie cykliczne (w pętli głównej), czy doszło do zmiany stanu portu połączonego z przyciskiem ze stanu wysokiego na niski,
- włączenie timera,
- ponowne sprawdzenie stanu przycisku,
- wykonanie akcji odpowiadającej wciśnięciu klawisza po osiągnięciu założonego czasu,
- wyłączenie timera.

W niniejszym ćwiczeniu wykorzystania zostanie metoda, którą można umiejscowić pomiędzy dwoma rozwiązaniami przedstawionymi powyżej. Tzn. procesor nie będzie pracował w sposób bezużyteczny (brak dedykowanej funkcji oczekiwania typu *delay*), ale z drugiej strony do odmierzenia czasu użyta zostanie sama pętla główna zamiast układu czasowo-licznikowego. Rozwiązanie takie również posiada pewne wady. Należą do nich: stopień komplikacji oraz konieczność korekty czasu oczekiwania na powtórne próbkowanie stanu klawisza wraz ze wzrostem rozmiaru pętli głównej rozwijanego programu. Metoda ta zostanie jednak zaimplementowana z uwagi na walory poznawcze.

Na rys. 4 przedstawiono diagram stanów ilustrujący koncepcję zastosowanej metody eliminacji drgań zestyków. Węzły diagramu oznaczają stan układu. Stan S_0 oznacza, że układ oczekuje na wciśnięcie klawisza. W stanie S_1 następuje odliczanie czasu w celu potwierdzenia wciśnięcia klawisza (właściwa eliminacja drgań zestyków). W stanie S_2 układ oczekuje na zwolnienie przycisku. W tabelkach przy węzłach grafu zamieszczono „wejścia” czyli warunki zmiany stanu oraz „wyjście” układu czyli potwierdzenie wciśnięcia przycisku. Zmienna *btn_current* przechowuje stan przycisku w bieżącym obiegu pętli głównej, zaś zmienna *btn_before* przechowuje stan przycisku w poprzednim obiegu pętli głównej. Zmienna *debounce_cnt* służy do odliczania przebiegów pętli głównej. W ten sposób czas wykonania pętli głównej staje się jednostką czasu, w której mierzony jest okres eliminacji drgań zestyków. Rozwiązanie przedstawione powinno zostać zaimplementowane jako część pętli głównej programu. Przycisk uznaje się za wciśnięty przy przejściu ze stanu S_1 do S_2 , w sytuacji, gdy *btn_before* == *btn_current* oraz została przekroczona graniczna wartość czasu, zdefiniowana za pomocą stałej LIMIT, oznaczającej liczbę przebiegów pętli głównej.



Rys. 4. Diagram stanów ilustrujący działanie mechanizmu eliminacji drgań zestyków z wykorzystaniem pętli głównej



Rys. 5. Algorytm programowej eliminacji drgań zestyków jako fragment pętli głównej

Algorytm programowej eliminacji drgań zestyków jako część głównej pętli programu przedstawiono na rys. 5. Należy zwrócić uwagę, iż algorytm dotyczy obsługi drgań zestyków dla pojedynczego przycisku. Jego adaptacja do obsługi większej liczby klawiszy nie powinna być jednak szczególnie utrudniona.

2. ZADANIA

1. Zmodyfikuj kod programu będącego treścią zadania drugiego z ćwiczenia nr 2 (program obsługi wyświetlacza siedmiosegmentowego oraz klawiszy) dodając do niego programową eliminację drgań zestyków wg wytycznych przedstawionych w rozdziale 1 niniejszej instrukcji opierając się zwłaszcza na rys. nr 5.
2. Dobierz eksperymentalnie wartość stałej LIMIT w sposób zapewniający poprawną pracę przycisków.
3. Na podstawie analizy gotowego kodu źródłowego oszacuj czas wystarczający do eliminacji drgań zestyków w konkretnym zestawie uruchomieniowym.



WSKAZÓWKA:

W celu obsługi większej liczby przycisków wygodnie jest przechowywać zmienne odpowiadające za obsługę drgań zestyków w tablicy.

3. LITERATURA

- [1] BTC Korporacja: Dokumentacja techniczna zestawu uruchomieniowego ZL2MCS51, <http://www.btc.pl/pdf/zl2mcs51.pdf> (dostęp: luty 2011).
- [2] Intel, MCS51 Microcontroller Family User's Manual, 1994 (dokumentacja dostępna pod różnymi adresami w sieci Internet).
- [3] Systemy wbudowane - Laboratorium nr 1 – 2, instrukcja do ćwiczeń, 2011.
- [4] Majewski J.: Programowanie mikrokontrolerów 8051 w języku C. Pierwsze kroki, BTC, Warszawa, 2005.
- [5] Stępień C. (red.): Mikroprocesory firmy Intel, PWN, Warszawa 1992.