

AGH, WIET	Techniki Obliczeniowe
Kierunek: EiT	Temat 33: Obliczanie wartości początkowych liczników (prędkość transmisji, interwały czasowe) dla wybranych mikroprocesorów np. rodziny 8051, ATMEGA, dla zadanej wartości częstotliwości MCLK.
Data wykonania: 19.01.2019	Imię i nazwisko: 1. Przybyło Wojciech 2. Nowak Artur

1. Wstęp teoretyczny

Nasz projekt rozpoczęliśmy od analizy not katalogowych odpowiednich mikroprocesorów rodziny 8051, ATMEGA. Wybraliśmy mikroprocesor AT89LP52, posiadający liczniki, które można skonfigurować w dwóch interesujących nas konfiguracjach:

1. *Mode 0* - licznik 13 bitowy
2. *Mode 1* - licznik 16 bitowy

Liczniki te posłużyły nam do wyznaczania pożądanego czasu i prędkości transmisji danych. Do realizacji projektu posłużyliśmy się prostymi przekształceniami równań zawartych w notach katalogowych.

2. Napotkane problemy

W trakcie realizacji projektu napotkaliśmy dwa znaczące problemy:

1. Odliczanie konkretnego czasu podanego przez użytkownika gdy wprowadzi on wartość przekraczającą całkowity czas wszystkich inkrementacji licznika.

Przykład:

Dla licznika 13-bitowego chcąc odmierzyć $4,5\text{ ms}$ przy czasie przeładowania od stanu 0 licznika do pojawienia się flagi równym 1 ms .

2. Odliczanie konkretnego czasu podanego przez użytkownika gdy wprowadzi on wartość niewiele różniącą się od wartości cyklu maszynowego licznika.

Przykład:

Dla licznika 16-bitowego chcąc odmierzyć $4,537\text{ }\mu\text{s}$ przy wartości cyklu maszynowego wynoszącej $1\text{ }\mu\text{s}$

Przy wyznaczaniu stanów początkowych dla czasu transmisji nie napotkaliśmy na poważniejsze problemy. Wszystkie zostały skorygowane prostymi funkcjami warunkowymi.

3.1 Realizacja projektu - Odmierzanie czasu

Projekt składa się z jednego głównego skryptu i kilku plików typu "function", które wcześniej wymieniony skrypt wykorzystuje, ze względu na dużo lepszą czytelność kodu. Wszystkie komentarze informujące użytkownika o kolejnych potrzebnych krokach znajdują się w plikach "function" by niepotrzebnie nie cieniować kodu. W plikach "function" znajdują się również

wszystkie obliczenia powiązane z każdym licznikiem osobno np. licznik 16-bitowy jest w innym pliku function niż licznik 13-bitowy. Oto lista nazw plików wraz z ich krótkim opisem:

1. Projekt.m - Skrypt będący filarem naszego projektu
2. Powitanie.m - Plik "function" w którym znajduje się krótkie powitanie użytkownika informującego go o np. temacie projektu, dacie jego wykonania itp.
3. Wybor_dzialania.m - Plik "function" w którym znajdują się 4 linie informacji dla użytkownika jak ma wybrać interesujące go działania. (Dwie opcje: ustalenie stanu początkowego licznika by odmierzyć czas lub wyznaczyć prędkość transmisji)
4. Wybor_licznika.m - Plik "function" w którym znajdują się 3 linie informacji dla użytkownika jak ma wybrać interesujący go licznik będący w trybie odmierzania czasu.
5. Wybor_transmisji - Plik "function" w którym znajdują się 3 linie informacji dla użytkownika jak ma wybrać interesujący go licznik będący w trybie wyznaczania transmisji danych.
6. Czas_AT89LP52_Liczniki_01_Mode0.m - Plik "function" służący do odmierzenia czasu na licznikach w trybie 0. Zawiera komentarz nt. licznika, ostrzeżenia o maksymalnej częstotliwości, wybór zegara zewnętrznego lub wewnętrznego, niezbędne obliczenia i pokazanie wyniku.
7. Czas_AT89LP52_Licznik_01_Mode1.m - Plik "function" zawierający informacje takie jak w punkcie szóstym jednak dostosowane do liczników w trybie 1.
8. Transmisja_AT89LP52_Licznik_Mode_0.m - Plik "function" służący do wyznaczenia stanu początkowego liczników gdy chcemy uzyskać pożądaną prędkość transmisji danych.
9. Transmisja_AT89LP52_Licznik_Mode_1 - Zawiera informacje podobne do tych z punktu ósmego jednak z obliczeniami i komentarzem dostosowanym do wyznaczania prędkości transmisji w trybie 1.

W pliku Transmisja_AT89LP52_Licznik_01_Mode0.m zajmujemy się wyznaczaniem stanu początkowego dla liczników mikroprocesora w trybie 0 czyli 13-bitowym. Użytkownik podaje dwie zmienne: czas i częstotliwość.

Problem nr 1 z punktu "Napotkane problemy" czyli:

„Dla licznika 13-bitowego chcąc odmierzyć 4,5 ms przy czasie inkrementacji od stanu 0 licznika do pojawienia się flagi równym 1 ms.”

został rozwiązany następująco:

1. Plik wyznacza wartość cyklu maszynowego w sekundach dla zadanej częstotliwości (zapisana pod zmienną cykl_maszynowy_w_sekundach). Używamy w tym miejscu dodatkowo wcześniej podanej przez użytkownika zmiennej "flaga_X" która mówi nam ile taktów zegara potrzeba nam na jeden cykl maszynowy. Zmienna "flaga_X" może przyjąć tylko i wyłącznie wartości 1, 6 lub 12.
2. Plik wyznacza czas przejścia od stanu 0 do przejścia z 8192 na 0 licznika (zapisana pod zmienną time_out_period).
3. Dalej przechodzimy do adekwatnej funkcji warunkowej "if" której warunkiem wykonania jest czas > time_out_period. (czas to nazwa zmiennej- podawana jest przez użytkownika)
4. Plik oblicza ile razy zmienna time_out_period mieści się w zmiennej czas i zokrągla ją w dół, zmienna zostaje zapisana pod nazwą "liczba_flag_minus_jeden". Nazwa zmiennej została dobrana tak by nie zapomnieć, że do odmierzenia czasu zawsze potrzebna jest

co najmniej jedna flaga, zatem aby ustalić całkowitą liczbę flag należy tą zmienną podnieść o jeden na końcu działania programu.

5. Za nowy czas podstawiony zostaje czas poprzedni pomniejszony o ilość pełnych flag przemnożonych przez zmienną `time_out_period`.
6. W tym momencie algorytm posiada nowy czas dla którego musi ustalić stan początkowy licznika. W tej sytuacji mogą zdarzyć się 3 rzeczy: nowy czas może być bliski co do wartości czasowi cyklu maszynowego, co jest jednocześnie naszym problemem numer 2, nowy czas może idealnie pokryć się z wartością ze zmiennej `time_out_period`, nowy czas może przyjąć jakąś inną wartość nie odpowiadającą żadnemu z dwóch wyżej wymienionych przypadków.

6.1.1 Jeśli nowy czas jest bliski wartości cyklu maszynowego w sekundach (Problem numer 2 z punktu "Napotkane problemy") w takiej sytuacji powinien zadziałać algorytm znajdujący się pod funkcją warunkową o która posiada następujące warunki by ją wykonać:

1. $(\text{cykl_maszynowy_w_sekundach}/\text{czas}) > 0.00001$
2. $\text{mod}(\text{czas}, (\text{cykl_maszynowy_w_sekundach} * 100)) \sim 0$

Pierwszy warunek sprawdza czy czas podany przez użytkownika ma wartość bliską wartości cyklu maszynowego. Im dysproporcja między wartością cyklu maszynowego a czasem podanym przez użytkownika jest większa tym mniej musimy martwić się o możliwe błędy w odliczeniu czasu. Drugi warunek sprawdza czy przypadkiem nowy czas wyznaczony w punkcie 5 nie jest równy zmiennej `time_out_period`. Drugi argument warunku numer dwa jest przemnożony przez liczbę sto. Podczas testowania algorytmu dla wartości cyklu maszynowego równego $1\mu s$ i czas ok. $20,56ms$ program pokazywał błąd przy zamianie stanu początkowego z liczby dziesiętnej na szesnastkową ponieważ na piętnastym miejscu po przecinku pojawiała się liczba jeden wynikająca z wysokiej precyzji zaokrągleń. Przemnożenie przez sto skutecznie wyeliminowało ten problem.

6.1.2 Jeśli oba warunki są spełnione to algorytm oblicza za pomocą proporcji stan początkowy licznika, odpowiednio zaokrąglając proporcję w dół jeśli zajdzie taka potrzeba dodając liczbę 1. To zaokrąglenie wynika jedynie z tego, że rozpatrujemy nowy czas bliski czasowi cyklu maszynowego, w innych przypadkach to zaokrąglenie nie będzie mieć miejsca.

$$\text{proporcja} = 8192 * \text{czas} / \text{time_out_period}$$
$$\text{stan_poczatkowy_licznika_dec} = 8192 - \text{ceil}(\text{proporcja}) + 1$$

Następnie dokonujemy konwersji na system heksadecymalny. Oprócz tego w kodzie znajdują się dwie dodatkowe linie odpowiadające za sprawdzenie otrzymanych wyników, które były pomocne na etapie projektowania, ale nie są już ważne dla naszych potrzeb dlatego w tym punkcie je pominiemy.

6.2 Jeśli nowy czas jest równy wartości zmiennej `time_out_period` to stan początkowy licznika ustali się na 0, a ilość flag będzie równa tej zapisanej w zmiennej `liczba_flag_minus_jeden`.

6.3 Jeśli nowy czas nie spełnia żadnych z dwóch wyżej wymienionych warunków to zostanie obliczona zwykła proporcja bez zaokrągleń:

$$\text{proporcja} = 8192 * \text{czas/time_out_period}$$

$$\text{stan_poczatkowy_licznika_dec} = 8192 - \text{proporcja} + 1$$

Następnie dokonujemy konwersji na system heksadecymalny.

7. Jeśli czas podany przez użytkownika będzie blisko wartości cyklu maszynowego to wykonany zostanie punkt 6.1.2 z tą różnicą, że zostanie podany błąd względny wynikający z zaokrąglenia.
8. Jeśli czas podany przez użytkownika jest mniejszy niż wartość zmiennej time_out_period a zarazem nie będzie bliski co do wartości zmiennej cykl_maszynowy_w_sekundach (prawdopodobieństwo zaistnienia takiej sytuacji oceniamy na małe ale możliwe) to obliczona zostanie zwykła proporcja jak w punkcie 6.3
9. Na samym końcu pojawiają się interesujące użytkownika wyniki zmienione za pomocą funkcji "num2str" do postaci tekstowej co czyni wyniki bardziej czytelnymi.

Analogicznie zadziała algorytm dla licznika 16-bitowego jednak wartość 8192 w kodzie zostanie zastąpiona wartością 65536.

3.2 Realizacja projektu - Prędkość transmisji

Oba tryby do pomiaru transmisji różnią się jedynie jedną wartością w jednym iloczynie, wynika to z różnych wzorów na "Baud rate", które odczytane zostały z noty katalogowej str. 58-59. Użytkownik podaje dane zewnętrzne takie jak: pożądaną ilość bodów na sekundę, częstotliwość MCLK i flaga X, która wyznacza ile taktów zegara przypada na jeden cykl maszynowy. Streścimy teraz najważniejsze punkty obliczeń:

1. Wyznaczenie cyklu maszynowego.
2. Wyznaczenie czasu przeładowania liczników w zależności od dwóch trybów SMOD0 i SMOD1.
3. Wyznaczenie stanu początkowego licznika. Znając ilość bitów licznika, czas jednego cyklu maszynowego i pożądaną czas przeładowania możemy bez większych problemów wyznaczyć ten parametr.
4. Deklaracja zmiennej kontrolnej i funkcja warunkowa "if". Jeśli funkcja "if" się wykona zmienna kontrolna przyjmie wartość jeden. Taka wartość tej zmiennej jest niezbędna w dalszej części kodu by móc wyświetlić poprawny wynik. Zapobiega to sytuacjom w których algorytm chciałby wykonać operację zamiany na system heksadecymalny liczby ujemnej lub zmiennoprzecinkowej liczby dziesiętnej, która może pojawić się przy nierozsądnie dobranych wejściowych.

4. Wnioski

Środowisko "Matlab" nie miało większych problemów z bardzo szybką realizacją naszych wzorów. Poprawnie zaimplementowane wzory umożliwiły nam uzyskanie szybkich odpowiedzi na interesujący nas temat.

Zdecydowaną większość czasu poświęciliśmy na zapoznanie się z tematem, wyprowadzanie poprawnych wzorów i testowanie ich. Nasz projekt zaczęliśmy od implementacji obliczeń, potem przeszliśmy do zabezpieczanie projektu przed błędami, niepoprawnymi wartościami wpisywanymi przez użytkowników itp. Zakończyliśmy go poprawianiem wyglądu kodu i umieszczeniem stosownych komentarzy w najważniejszych miejscach.

5. Przykładowe obliczenia

1. Odmierzanie czasu - wartość czasu mniejsza niż czas przeładowania i większa od cyklu maszynowego

Dane wejściowe: Licznik 13-bitowy, częstotliwość 12MHz, 1 cykl maszynowy to 6 cykli zegara, czas $1ms$

Wynik:

Cykl maszynowy trwa: $0,5\mu s$

Stan początkowy licznika: 5193

Sprawdzenie: $(8193-5193)*0,5\mu s=0,0015$

2. Odmierzanie czasu - wartość czasu bliska cyklu maszynowego

Dane wejściowe: Licznik 16-bitowy, częstotliwość 12MHz, 1 cykl maszynowy to 12 cykli zegara, czas $4,537\mu s$

Wynik:

Cykl maszynowy trwa: $1\mu s$

Stan początkowy licznika: 65532

Błąd odmierzenia: 0,0926

Sprawdzenie: $(65537-65532)*1\mu s=5\mu s$, $1-4,537/5=0,0926$

3. Odmierzanie czasu - wartość dużo większa od czasu przeładowania

Dane wejściowe: Licznik 13-bitowy, częstotliwość 12MHz, 1 cykl maszynowy to 1 cykl zegara, czas $3s$

Wynik:

Cykl maszynowy trwa: $83,333ns$

Stan początkowy licznika: 3841

Łączna liczba flag potrzebna do odmierzenia czasu: 4395

Sprawdzenie: $4395*8192*83,333ns+(8193-3841)*83,333ns=3$

4. Prędkość transmisji - czas przeładowania licznika daleki od czasu cyklu maszynowego

Dane wejściowe: Licznik 16-bitowy, częstotliwość 11,059MHz, 1 cykl maszynowy to 1 cykl zegara, prędkość transmisji 128000 Bodów/sekundę

Dane wyjściowe:

Cykl maszynowy w sekundach wynosi: 9.0424e-08

Czas przeładowania licznika przy SMOD0 ma wynieść: 1.9531e-06

Stan początkowy licznika 1 przy SMOD0 w systemie dziesiętnym: 65516

Stan początkowy licznika 1 przy SMOD0 w systemie heksadecymalnym: FFEC

Rzeczywisty czas przeładowania licznika: 1.8989e-06

Błąd względny generacji: 0.02776

Realizacja ma szansę powodzenia

Czas przeładowania licznika przy SMOD1 ma wynieść: 3.9063e-06

Stan początkowy licznika 1 przy SMOD1 w systemie dziesiętnym: 65494

Stan początkowy licznika 1 przy SMOD1 w systemie heksadecymalnym: FFD6

Rzeczywisty czas przeładowania licznika: 3.8882e-06

Błąd względny generacji: 0.0046116

Realizacja ma szansę powodzenia

5. Prędkość transmisji - czas przeładowania licznika bliski od czasu cyklu maszynowego

Dane wejściowe: Licznik 16-bitowy, częstotliwość 11,059MHz, 1 cykl maszynowy to 12 cykli zegara, prędkość transmisji 128000 Bodów/sekundę

Dane wyjściowe:

Cykl maszynowy w sekundach wynosi: 1.0851e-06

Czas przeładowania licznika przy SMOD0 ma wynieść: 1.9531e-06

Stan początkowy licznika 1 przy SMOD0 w systemie dziesiętnym: 65536

Stan początkowy licznika 1 przy SMOD0 w systemie heksadecymalnym: 10000

Rzeczywisty czas przeładowania licznika: 1.0851e-06

Błąd względny generacji: 0.44443

Przekroczono 3% prog

Czas przeładowania licznika przy SMOD1 ma wynieść: 3.9063e-06

Stan początkowy licznika 1 przy SMOD1 w systemie dziesiętnym: 65534

Stan początkowy licznika 1 przy SMOD1 w systemie heksadecymalnym: FFFE

Rzeczywisty czas przeładowania licznika: 3.2553e-06

Błąd względny generacji: 0.16665

Przekroczono 3% prog