
Ćwiczenie 9: Pomiar czasu względnego za pomocą timera

Instrukcja laboratorium

Mariusz Chilmon <mariusz.chilmon@ctm.gdynia.pl>



2024-01-30

Any fool can write code that a computer can understand. Good programmers write code that humans can understand.

— *Martin Fowler*

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z:

- odczytem czasu odmierzonego przez timer,
- wykorzystaniem przerwań GPIO w interakcji z timerem,
- dekompozycji złożonego programu do bloków funkcjonalnych.

Uruchomienie programu wyjściowego

1. Podłącz płytkę *WPSH209* do *Arduino Uno*.
2. Wyświetlacz wskazuje wartość 8 . 888.

Zadanie podstawowe

Celem zadania podstawowego jest zaprogramowanie miernika refleksu. Urządzenie ma co około 10 sekund rozpoczynać odliczanie sygnalizowane sygnałem świetlnym lub dźwiękowym, które użytkownik ma jak najszybciej zatrzymać wciśnięciem przycisku *S1*. Czas między sygnałem a wciśnięciem przycisku ma być mierzony z rozdzielczością 1 ms.

Wymagania funkcjonalne

1. Po kilku–kilkunastu sekundach od uruchomienia zaświecana jest dioda *D1* i/lub uruchamiany jest buzzer *LS1*. Rozpoczyna się odmierzenie czasu na wyświetlaczu.
2. Po chwili (np. 20 ms) dioda i/lub buzzer są wyłączone.
3. Po wciśnięciu przycisku *S1* odliczanie czasu zostaje zatrzymane i można go odczytać z wyświetlacza.
4. Po kilku–kilkunastu sekundach cykl pracy rozpoczyna się od nowa.

Modyfikacja programu

Implementację programu najlepiej podzielić na kroki, pozwalające stopniowo realizować żadaną funkcjonalność.

Do odmierzania czasu można użyć timera *Timer/Counter1*, który dzięki 16-bitowej rozdzielczości pozwoli zachować odpowiednią rozdzielczość pomiaru.

Wyświetlanie stanu timera

W funkcji `currentTime()` zwróć wartość licznika `TCNT1`.

Na wyświetlaczu będzie widoczna domyślna wartość tego rejestru, tj. 0.

Uruchomienie timera

W celu uruchomienia timera należy za pomocą bitów `CS10...CS12` podać sygnał zegarowy o częstotliwości, która umożliwi odliczenie czasu rzędu kilku sekund. Możesz zrobić to w pętli głównej `mainLoop()`, dzięki czemu odliczanie będzie mogło być cyklicznie wznowiane.

Na wyświetlaczu będzie widoczne odliczanie licznika `TCNT1`.

Przeskalowanie czasu

Przelicz wartość zwracaną przez `currentTime()`, tak by była wyrażona w milisekundach.

Opóźnienie startu

Opóźnij start odliczania dodając wywołanie funkcji `_delay_ms()` w pętli głównej.



Ponieważ pomiar czasu następuje za pomocą timera, a przycisk obsługiwany jest w przerwaniu, umieszczanie w pętli głównej odliczania czasu za pomocą `_delay_ms()` (co blokuje program na czas odliczania interwału) nie stanowi problemu.

Zatrzymanie odliczania

Aby po wciśnięciu przycisku jak najszybciej zatrzymać odliczanie, należy użyć przerwania.

Aby włączyć przerwanie od przycisku S1 należy w funkcji `interruptsInitialize()` włączyć przerwanie `PCINT1` dla wejścia `PCINT9`¹:

```
1 PCMSK1 |= _BV(PCINT9);  
2 PCICR  |= _BV(PCIE1);
```

W obsłudze tego przerwania (`ISR(PCINT1_vect)`) zatrzymaj pracę timera przez wyłączenie sygnału zegarowego.

Resetowanie timera

Na początku każdego cyklu (po odmierzeniu opóźnienia za pomocą `_delay_ms()`) wyzeruj licznik timera, tak by pomiar czasu zaczynał się zawsze od zera.

Sygnalizacja rozpoczęcia odliczania

Bezpośrednio po lub bezpośrednio przed włączeniem timera² w pętli głównej włącz buzzer i/lub LED. Wyłącz je po krótkiej chwili, do odmierzenia której też możesz użyć `_delay_ms()`.

Zadanie rozszerzone

Celem zadania rozszerzonego jest usunięcie możliwości oszukiwania urządzenia przez użytkownika.

Wymagania funkcjonalne

1. Po przepiętleniu timera odliczanie jest zatrzymywane.
2. Ciągłe wciskanie i puszczenie przycisku opóźnia start odliczania, żeby nie było możliwe w ten sposób uzyskanie niskiego wyniku (falstart).

Modyfikacja programu

Przepiętlenie timera można obsłużyć w przerwaniu `TIMER1_OVF_vect`. Włącza się je za pomocą flagi `TOIE1`.

Obsłużenie falstartu można zrealizować przez ustawianie dodatkowej zmiennej w przerwaniu od przycisku i sprawdzaniu jej w pętli głównej.

¹Tak, w tym mikrokontrolerze `PCINT` raz oznacza przerwanie od GPIO, a raz pin wywołujący to przerwanie...

²Nie da się tego zrobić jednocześnie, gdyż nie da się w jednej instrukcji zapisać danych do kilku rejestrów.