Ćwiczenie 2: Przerwania GPIO

Instrukcja laboratorium

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

2023-12-19

> Remember, things take time.
>
> — *Piet Hein*

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z:

- problemem odmierzania czasu w pętli głównej,
- korzyściami z wykorzystania przerwań,
- konfiguracją przerwań GPIO,
- uruchamianiem podsystemu przerwań,
- makrem `ISR()` do obsługi przerwań w AVR-GCC.

Uruchomienie programu wyjściowego

1. Podłącz płytkę *WPSH209* do *Arduino Uno*.
2. Załóż zworkę `J2`.
3. Zbuduj program i wgraj do mikrokontrolera.
4. Zweryfikuj, czy dioda LED `D1` mruga.
5. Zweryfikuj, czy dioda LED `D2` zaświeca się po wciśnięciu przycisku `A1`.

Zauważ, że dioda `D2` nie reaguje natychmiast na wciśnięcie przycisku. Wynika to z tego, że stan przycisku sprawdzany jest tylko co jakiś czas, między wykonaniami funkcji `heartBit()`.

Funkcja `heartBit()` symuluje wykonywanie przez pętlę główną złożonego programu. W rzeczywistości funkcja ta prawie całą moc obliczeniową mikrokontrolera zużywa na odmierzanie czasu w funkcji bibliotecznej `_delay_ms()`. Jak widzisz, odmierzanie czasu w ten sposób jest bardzo problematyczne.

Zadanie podstawowe

Modyfikacja programu

1. Przenieś wywołanie funkcji `handleKey()` z pętli głównej do obsługi przerwania `PCINT1` (zwanego też `PCI1`), czyli funkcji `ISR(PCINT1_vect)`.
2. W funkcji `interruptsInitialize()` umieść:
    1. włączenie przerwania `PCINT1`;
    2. aktywację tego przerwania przez pin, do którego jest podłączony przycisk `A1`;
    3. odblokowanie globalnej maski przerwań.

Zapoznaj się z rejestrami `PCICR` i `PCMSK1` oraz funkcją `sei()`.

Piny mikrokontrolera (wybrane lub dowolne, zależnie od możliwości mikrokontrolera) mogą przerywać działanie pętli głównej, np. po pojawieniu się zbocza (tu: zmianie stanu logicznego na przeciwy). Umożliwia to szybką reakcję na zdarzenie zewnętrzne.
Zazwyczaj mikrokontrolery z rodziny ATmega posiadają kilka pinów obsługujących przerwania `INT` (np. `INT0`, `INT1`...), które można skonfigurować w dowolny sposób (np. reakcja tylko na zbocze opadające) oraz zestaw przerwań `PCINT`, umożliwiających reakcję na zmianę stanu na pozostałych pinach, pogrupowanych w ośmioelementowe bloki.

Wymagania funkcjonalne

1. Dioda `D1` miga bez zmian.
2. Dioda `D2` reaguje natychmiast na wciśnięcia przycisku `A1`.

Zadanie rozszerzone

Modyfikacja programu

Zmodyfikuj pętlę główną i obsługę przerwania z zadania podstawowego tak, by po wciśnięciu przycisku była ustawiana zmienna, która zmieni w pętli głównej stan diody `D3` na przeciwny.

Do zmiany stanu diody na przeciwny możesz użyć funkcji `ledToogle(PIN_LED_TOGGLE)`.

Pamiętaj o kwalifikatorze typu **`volatile`**.

Wymagania funkcjonalne

1. Dioda `D1` miga bez zmian.
2. Dioda `D2` reaguje natychmiast na wciśnięcia przycisku `A1`.
3. Dioda `D3` zmienia stan na przeciwny z opóźnieniem, wynikającym z działania pętli głównej.

W tym zadaniu wykorzystujemy przerwanie nie do tego, by zareagować natychmiast na zdarzenie, ale tylko do tego, by je zarejestrować i obsłużyć później w pętli głównej. W ten sposób często obsługiwane są zdarzenia, które wymagają czasochłonnych operacji. Dzięki takiemu podejściu procesor nie utyka na dłuższy czas w obsłudze przerwania.