

Miniaturowy przypomniacz

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5638

Podstawowe parametry:

- Zasilanie z baterii litowej 3 V.
- Pobór prądu: w trybie czuwania 0,5 μ A, w stanie aktywnym 8 μ A, w stanie alarmu 7 mA.
- Odmierzanie jednego z 4 predefiniowanych czasów: 15 minut, 30 minut, 1 godzina, 2 godziny.
- Odmierzanie czasu wyzwalane przyciskiem.
- Czas ustawiany przełącznikiem DIP-SWITCH.
- Mikrokontroler ATtiny12A, oprogramowanie w języku C.

Projekty pokrewne na www.media.avt.pl:

AVT-1862 Sygnalizator akustyczny (EP 8/2015)
AVT-2900 Elektroniczny przypomniacz (Edw 5/2009)
AVT-2894 Zegar-przypomniacz VFD (Edw 4/2009)

Wykaz elementów:

Rezystory: (SMD 0805)

R1, R6, R7, R11 10 k Ω
R2, R3: 100 Ω
R4, R5, R9: 2,2 k Ω
R8: 4,7 k Ω
R10: 1 k Ω

Kondensatory:

C1, C2: 10 μ F/10 V (SMD 0805)
C3: 100 nF (SMD 0805)

Półprzewodniki:

LED1, LED2: niebieskie (SMD 1206)
T1, T2: BC847 lub podobne
U1: ATtiny13A-SSU

Pozostałe:

B1: koszyk poziomy CR2032 THT + bateria
BUZ1: piezo z generatorem 3 V
SW1: przycisk (6x6/17)mm
SW2: DIP SWITCH, 4 sekcje
Obudowa KM-10N

Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie K1Tem (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym UK) - jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wlutowane w płytkę PCB)
- wersja [A] płytka drukowana bez elementów i dokumentacja Kitu w których występuje układ scalony wymagający zaprogramowania, posiadają następujące dodatkowe wersje:

- wersja [A*] płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB), prosimy o kontakt via email: kity@avt.pl.

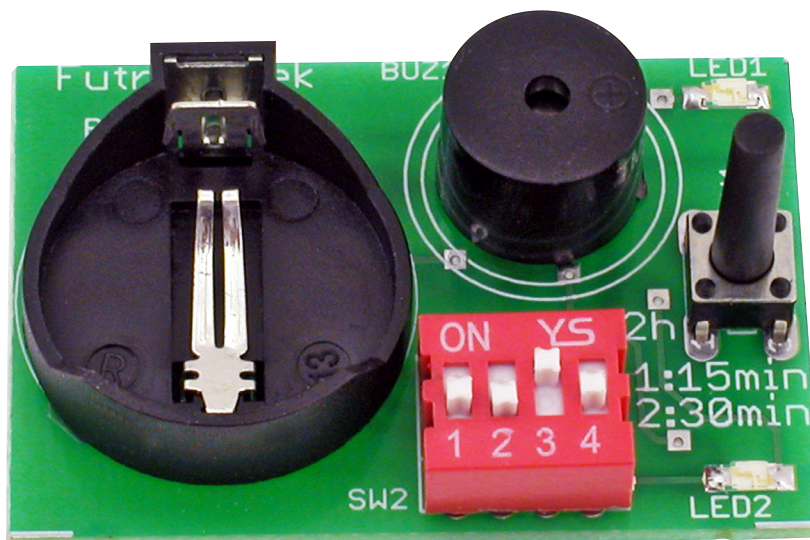
Opisywany projekt urządzenia, które służy do odmierzenia jednego z czterech możliwych do wyboru czasów: 15 minut, 30 minut, 1 godzina lub 2 godziny. Odmierzanie czasu jest sygnalizowane szybkimi błysnięciami, a koniec odliczania błyskami i dźwiękiem. Do zasilania przewidziano jedną baterię typu CR2032, przez co przypomniacz może być naprawdę lekki i niewielki.

Budowa

Schemat ideowy proponowanego rozwiązania pokazano na **rysunku 1**. Zastosowanie

To niewielkie urządzenie zostało zaprojektowane, przede wszystkim, z myślą o diabetykach, lecz można dla niego znaleźć znacznie szersze spektrum zastosowań. Jego zadaniem jest zasygnalizowanie upływu wcześniej ustalonego odcinka czasu, mierzonych od wciśnięcia przycisku. Można w ten sposób przypomnieć o konieczności wykonania pomiaru poziomu glukozy we krwi po pewnym czasie od posiłku, albo o konieczności zajrzenia do kotłowni po pewnym czasie od rozpalenia w piecu.

Rekomendacje: użyteczny gadżet – rodzaj elektronicznego węzełka z timerem, który przyda się diabetykom, kucharzom i zapominalskim (tylko czy oni będą pamiętać, po co ten węzełek?).



mikrokontrolera typu ATtiny13A firmy Microchip pozwoliło na znaczne uproszczenie budowy urządzenia. Identyfikacja układu zrealizowany wyłącznie przy użyciu bramek logicznych byłby zdecydowanie bardziej złożony. Ten typ mikrokontrolera idealnie nadaje się do zastosowania w opisywanej aplikacji. Ma niewielką obudowę SO8, możliwość zasilania napięciem już od 1,8 V oraz różne tryby obniżonego poboru energii. Dodatkowo, jest to jeden z najbardziej popularnych oraz najtańszych układów tej klasy dostępny praktycznie w każdym sklepie elektronicznym.

Do zasilania służy bateria litowa typu CR2032. Dostarcza napięcie 3 V, co jest wystarczające do zasilania podzespołów służących do sygnalizacji. Kondensatory C1...C3 zmniejszają impedancję źródła energii. To istotne zwłaszcza dla baterii będących na granicy rozładowania, w których wzrasta rezystancja wewnętrzna.

Informacja optyczna jest prezentowana za pomocą dwóch niebieskich diod LED. Wprawdzie

ich napięcie przewodzenia jest zbliżone do napięcia zasilającego, lecz to wystarcza, aby świeciły dostatecznie jasno. Jednocześnie ten kolor jest dobrze widoczny dla ludzkiego oka, zwłaszcza w warunkach niedostatecznego natężenia oświetlenia. Do ich załączania służy klucz z tranzystorem T1. Alarmowanie o odmierzeniu

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków

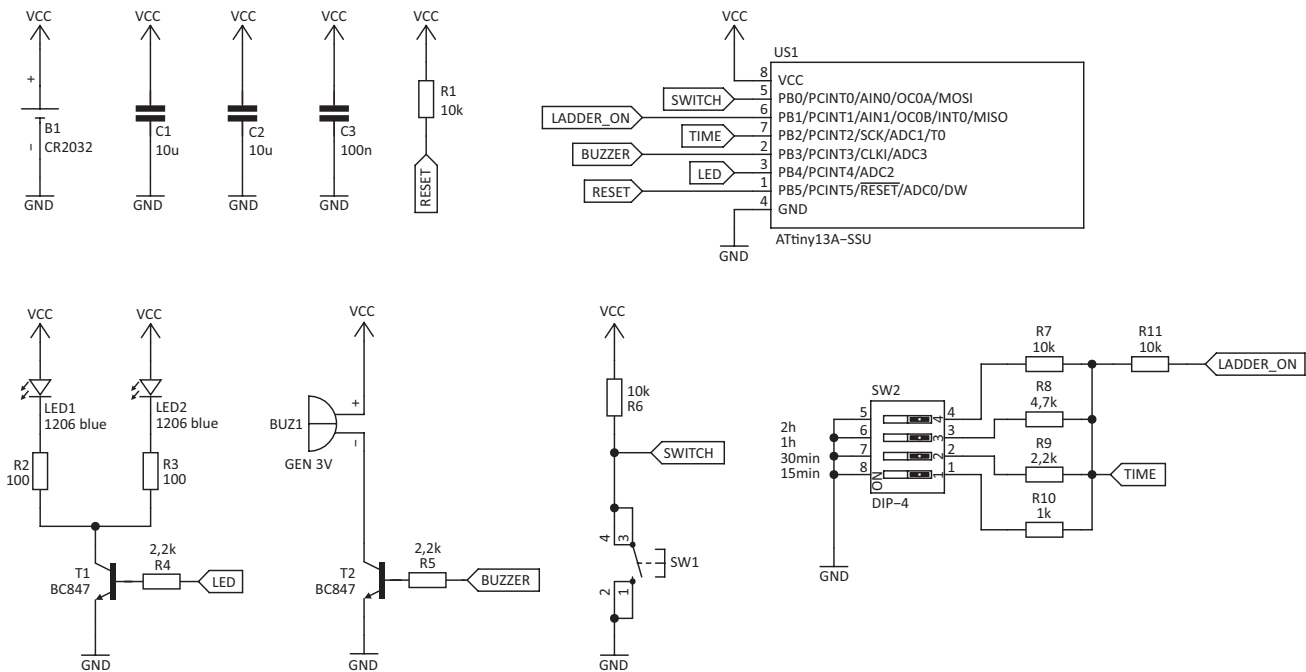
STM32

STM32

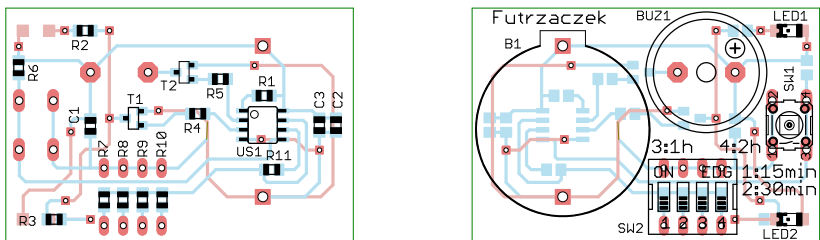
TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

ST
IIS. augmented
CERTYFIKOWANY PARTNER SZKOLENIOWY



Rysunek 1. Schemat ideowy przypominacza



Rysunek 2. Schemat montażowy przypominacza

zadanego czasu odbywa się też za pomocą sygnalizatora piezoelektrycznego z wbudowanym generatorem. Jest on wystarczająco głośny, aby usłyszeć go nawet wtedy, kiedy układ zostanie schowany np. do kieszeni. Jego załączenie odbywa się poprzez nasycenie tranzystora T2.

Uruchamianie odliczania oraz zatrzymywanie sygnalizacji odbywa się poprzez jednorazowe wciśnięcie przycisku SW1, którego styki są wstępnie polaryzowane przez rezystor podciągający R6. Taki sam rezystor zamontowano przy wyprowadzeniu RESET mikrokontrolera, co zapobiega przypadkowej zmianie jego poziomu logicznego na niski. Takie zaburzenie mogłoby być spowodowane zewnętrznym polem elektromagnetycznym lub osadzeniem się na nim zbyt dużej liczby ładunków elektrostatycznych.

Ustalenie długości odmierzanego czasu odbywa się za pomocą załączenia jednej z czterech sekcji przełącznika SW2 typu DIP-SWITCH. Załączona sekcja zwiera do masy jeden rezystor (od R7 do R10), co z rezystorem R11 tworzy dzielnik napięciowy. Mikrokontroler, podając wysoki poziom logiczny na rezystor R11, może zmierzyć przetwornikiem A/C napięcie odkładające się na owym dzielniku. W ten sposób, do zadawania czasu użyto jedynie dwóch wyprowadzeń mikrokontrolera. Dodatkowo, ten obwód można wyłączyć, zmieniając poziom wyprowadzenia PB1 na niski, co znacznie

obniża zapotrzebowanie układu na cenną energię baterii.

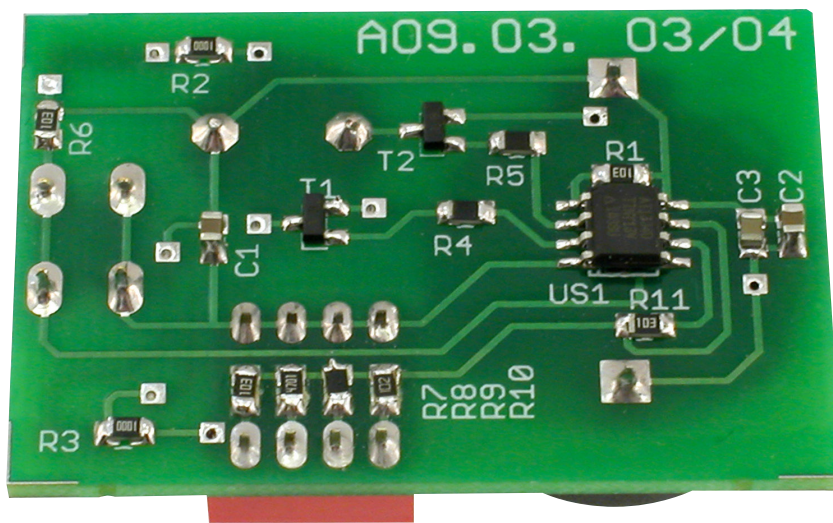
Montaż i uruchomienie

Układ przypominacza zmontowano na dwustronnej płytce drukowanej o wymiarach 46 mm×31 mm, której schemat montażowy pokazano na **rysunku 2**. W pierwszej kolejności należy przylutować elementy montowane powierzchniowo, które znajdują się po spodniej stronie płytki – mikrokontroler, tranzystory i inne (**fotografia 3**). Następne w kolejności

są dwie diody LED po stronie przeciwnej. Na sam koniec polecam włutować elementy przewlekane, po tej samej stronie, co diody LED1 i LED2. Do mikrokontrolera należy wgrać dedykowany wsad pamięci Flash, przy fabrycznej konfiguracji bitów zabezpieczających (tzw. fusebitów). Jeżeli wszystko zostało zmontowane poprawnie, układ jest gotowy do działania po włożeniu baterii do koszyka. Wystarczy ustawić żądany do odmierzenia czas za pomocą przełącznika SW2.

Eksplatacja

Będąc w stanie uśpienia, układ nie daje żadnych oznak działania. Jednorazowe, krótkie wciśnięcie przycisku powoduje pobranie informacji z przełącznika i rozpoczęcie odliczania czasu, co jest sygnalizowane pojedynczym, dłuższym mignięciem diod LED. Od tej pory będą one błyskały co 1 sekundę, aż do zakończenia odliczania. Jeżeli na przełączniku nie wybrano żadnej opcji (wszystkie przełączniki w pozycji OFF),



Fotografia 3. Spodnia strona płytki z zamontowanymi elementami

diody nie będą sygnalizowały odmierzenia czasu i układ powróci do stanu spoczynkowego.

Po upływie zadanego czasu, diody zaczynają błyskać intensywnie, natomiast sygnalizator dźwiękowy wydaje z siebie krótkie dźwięki. Zatrzymanie alarmu odbywa się po krótkim wciśnięciu przycisku SW1. Co istotne, raz uruchomionego odliczania nie można zatrzymać,

ani zmodyfikować czasu jego trwania. Ma to na celu uniknięcie sytuacji niechcianego wyłączenia odmierzenia czasu, które mogłoby się zdarzyć przy przypadkowym wciskaniu przycisku SW1.

Cały układ mieści się w obudowie KM-10N. Jest wykonana z niebieskiego, półprzezroczystego tworzywa sztucznego, przez które widać

światło diod. Należy wywiercić w niej jedynie dwa niewielkie otworki: jeden dla przycisku SW1, a drugi dla sygnalizatora piezoelektrycznego. Przełącznik SW2 będzie wtedy schowany wewnątrz, lecz był to zabieg celowy – odmierzony czas ustawia się zazwyczaj na początku eksploatacji, a potem korzysta cały czas z tego samego ustawienia.

Pobór prądu w stanie uśpienia wynosi ok. 0,5 μ A, więc teoretycznie bateria o nominalnej pojemności 200 mAh wystarczy na 45 lat czuwania. W praktyce można liczyć na czas porównywalny z datą trwałości danej baterii. Podczas odliczania (kiedy diody błyskają cyklicznie), średni pobór prądu to ok. 8 μ A. W stanie alarmu wzrasta on do 7 mA. W tabeli 1 umieszczono wykaz nominalnych oraz zmierzonych czasów. Błąd jest rzędu 3%, a to przez użycie wzorca RC, o czym napisano dalej.

Dla ciekawskich

Tak mały pobór prądu udało uzyskać się dzięki temu, że mikrokontroler niemal cały czas przebywa w stanie uśpienia. Odmierzaniem czasu zajmuje się wbudowany układ Watchdog, który jest wyposażony w niezależny oscylator 128 kHz o bardzo małym zapotrzebowaniu na moc. Stopnie podziału zostały tak dobrane przez producenta, że jest możliwe generowanie przerwania co 1 s (rysunek 4). Użyty w tym układzie oscylator jest typu RC, toteż dokładność odmierzanego przezeń czasu może pozostawiać trochę do życzenia. Postanowiłem to sprawdzić poprzez pomiar oscyloskopem częstotliwości wywoływania przerwań. W procedurze obsługi przerwania poziom logiczny jednego wyprowadzenia był na chwilę zmieniany na wysoki, co skutkowało pojawieniem się impulsu szpilkowego. Na rysunku 5 pokazano oscylogram przy ustawionym interwale 16 ms, a na rysunku 6 przy interwale 1 s (zasilanie 3 V).

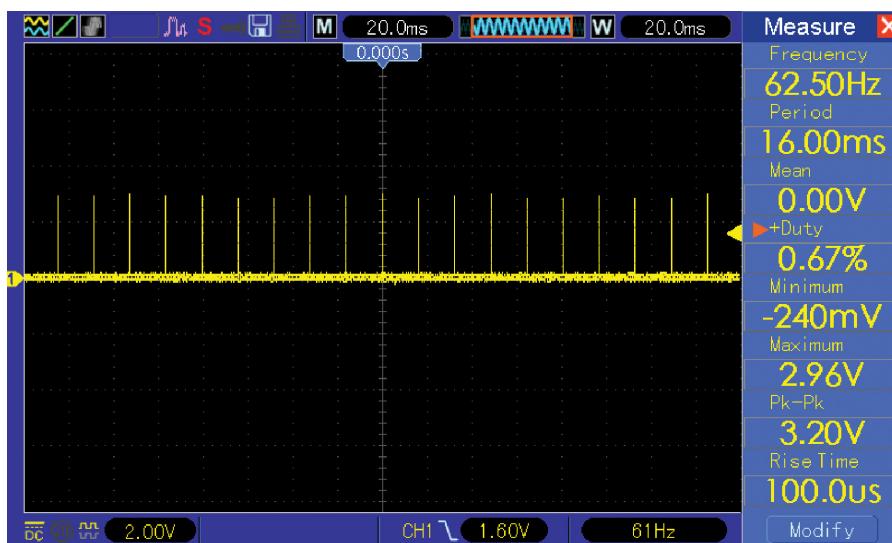
Uzyskane rezultaty są zachęcające: przy odmierzanym czasie rzędu 16 ms, błąd był mniejszy niż rozdzielczość pomiaru. Przy 1 s, wyniósł jedynie niecałe 3%, co ma potwierdzenie w wynikach zawartych w tabeli 1. Taki wynik byłby nie do zaakceptowania w zegarze, lecz w ręcznym „przypomniaczu” można go uznać za dopuszczalny.

Kod programu, który steruje pracą niniejszego układu i bazuje na tych przerwaniach, znajduje się na listingu 1. Został w całości napisany w języku C.

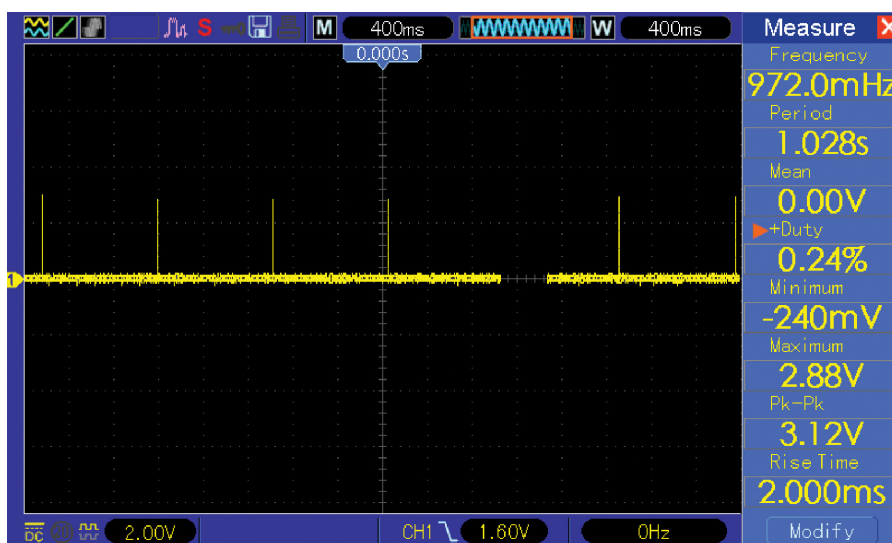
Michał Kurzela, EP

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V _{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s

Rysunek 4. Dostępne interwały przerwań generowanych przez układ Watchdog (źródło: nota katalogowa firmy Microchip, dawniej Atmel)



Rysunek 5. Oscylogram obrazujący częstotliwość przerwań Watchdog przy interwale 16 ms



Rysunek 6. Oscylogram obrazujący częstotliwość przerwań Watchdog przy interwale 1 s

Tabela 1. Pomiary odmierzanych czasów	
Czas nominalny	Czas zmierzony
15 min	15 min 30 s
30 min	30 min 50 s
1 h	1 h 1 min 45 s
2 h	2 h 4 min

Listing 1. Kod programu przypomnacza

```
//Miniaturowy przypomnacz AVT
//Michał Kurzela @ 2018
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
//definicja długości odmierzanyc czasów [w minutach]
#define CZAS_1 15
#define CZAS_2 30
#define CZAS_3 60
#define CZAS_4 120
//progi ADC do wykrycia zworek
#define PROG_4 700 //nominalnie 512
#define PROG_3 380 //nominalnie 327
#define PROG_2 250 //nominalnie 185
#define PROG_1 130 //nominalnie 93
#define LADDER_ON PORTB|=(1<<PORTB1)
#define LADDER_OFF PORTB&=~(1<<PORTB1)
#define BUZZER_ON PORTB|=(1<<PORTB3)
#define BUZZER_OFF PORTB&=~(1<<PORTB3)
#define LED_ON PORTB|=(1<<PORTB4)
#define LED_OFF PORTB&=~(1<<PORTB4)
#define LICZBA_BLYSKOW 4 //liczba błysków w jednym cyklu
#define PRZERWA_KROTKA 50 //długość krótkiej przerwy w generowaniu błysków
#define PRZERWA_DLUGA 250 //długość przerw między seriami błysków

volatile uint8_t sekundy = 0; //odliczone sekundy
volatile uint8_t minuty = 0; //odliczone minuty
volatile uint8_t czy_odlicza = 0; //0 - pozostaje uspiomy, 1 - odlicza, 2 - wszczyna alarm
volatile uint8_t czas_zadany; //zadany do odliczenia czas w minutach

// Przerwanie od przetwornika A/C
ISR(ADC_vect)
{
    if((ADCL | (ADCH<<8)) <= PROG_1){ czas_zadany = CZAS_1; }
    if((ADCL | (ADCH<<8)) > PROG_1 && (ADCL | (ADCH<<8)) <= PROG_2){ czas_zadany = CZAS_2; }
    if((ADCL | (ADCH<<8)) > PROG_2 && (ADCL | (ADCH<<8)) <= PROG_3){ czas_zadany = CZAS_3; }
    if((ADCL | (ADCH<<8)) > PROG_3 && (ADCL | (ADCH<<8)) <= PROG_4){ czas_zadany = CZAS_4; }
    if((ADCL | (ADCH<<8)) > PROG_4){ czas_zadany = 0; czy_odlicza = 0; } //przejdź w stan uspienia, jeżeli nie ma załączonego DIP switcha
    ADCSRA = 0; LADDER_OFF; //wyłącz ADC i drabinkę
    if(czy_odlicza)
    { //jeżeli odliczanie nie zostało zatrzymane
        sekundy = 0; minuty = 0; //przygotowanie do odliczania
        WDTCR = (1 << WDTIE) | (1 << WDP1) | (1 << WDP2); //uruchomienie przerw od watchdoga + timeout 1s
    }
    LED_OFF; //wyłącz diody LED - użytkownik otrzyma mignięcie
}

// Przerwanie od przycisku SW1
ISR(PCINT0_vect)
{
    if(bit_is_clear(PINB,PINB0))
    {
        if(czy_odlicza == 0)
        { //jeżeli nie odliczał
            LADDER_ON; //załączenie drabinki rezystorowej
            LED_ON; //załączenie diod LED dla uzyskania mignięcia potwierdzającego start
            czy_odlicza = 1; //trwa odliczanie
            ADMUX = (1 << MUX0); //wejście ADC1
            ADCSRA = (1 << ADEN) | (1 << ADATE) | (1 << ADIE) | (1 << ADPS2); //uruchomienie ADC, wielokrotne powtarzanie, uruchomienie
            przerwania, preskaler przez 16
            ADCSRA |= (1 << ADSC); //rozpoczęcie konwersji
            _delay_ms(200); //aby wyeliminować drganie styków
        }
        if(czy_odlicza == 2)
        { //jeżeli został wszczęty alarm po odliczeniu czasu
            czy_odlicza = 0; //przejsie w stan uspienia
            WDTCR = 0; //watchdog jest już niepotrzebny
            LED_OFF; BUZZER_OFF; //wyłącz diody i buzzer
            _delay_ms(1000); //aby wyeliminować drganie styków
        }
    }
}

// Przerwanie od Watchdog
ISR(WDT_vect)
{
    if(czy_odlicza == 1)
    {
        sekundy++;
        LED_ON;
        _delay_ms(2);
        LED_OFF;
    }
    if(sekundy >= 60){minuty++; sekundy = 0; }
    if(minuty >= czas_zadany){ czy_odlicza = 2; } //rozpocznij alarm
}

// GŁÓWNA FUNKCJA PROGRAMU
int main(void)
{
    //konfiguracja wyprowadzeń
    DDRB = (1 << DDB4) | (1 << DDB3) | (1 << DDB1); //wyjścia LED, BUZZER i LADDER_ON
    PORTB = (1 << PORTB0); //wejście SWITCH w stanie wysokim
    //uruchomienie przerw od przycisku
    GIMSK = (1 << PCIE); //aktywacja systemu przerw PCINT
    PCMSK = (1 << PCINT0); //przerwanie od pinu PINB0
    sei(); //uruchomienie systemu przerw

    while(1)
    {
        while(czy_odlicza == 2){ //w stanie alarmu
            for(uint8_t i = 0 ; i < LICZBA_BLYSKOW ; i++)
            {
                LED_ON; BUZZER_ON;
                _delay_ms(PRZERWA_KROTKA);
                LED_OFF; BUZZER_OFF;
                _delay_ms(PRZERWA_KROTKA);
            }
            _delay_ms(PRZERWA_DLUGA);
        }
        set_sleep_mode(SLEEP_MODE_PWR_DOWN);
        sleep_enable();
        sleep_cpu();
    }
    sleep_disable();
}

```