

# Raport do zadania 3

Imię i nazwisko: Jakub Dmochowski

Numer Albumu: 169236

## Konfiguracja sprzętowa

```
#pragma config POSCMOD = XT  
#pragma config OSCIOFNC = ON  
#pragma config FCKSM = CSDCMD  
#pragma config FNOSC = PRI  
#pragma config IESO = ON
```

POSCMOD = XT - wybór zewnętrznego kwarcu jako źródła zegara

OSCIOFNC = ON - funkcja wyjścia oscylatora włączona

FCKSM = CSDCMD - wyłączenie monitorowania zegara i przetaczania

FNOSC = PRI - Podstawowe źródło zegara jako domyślne

IESO = ON - włączenie wewnętrznego/zewnętrznego przetaczania oscylatora

## Watchdog timer, debugowanie

```
#pragma config WDTPS = PS32768
#pragma config FWPSA = PR128
#pragma config WINDIS = ON
#pragma config FWDTEN = OFF
#pragma config ICS = PGx2
#pragma config GWRP = OFF
#pragma config GCP = OFF
#pragma config JTAGEN = OFF
```

WDTPS = PS32768 - Prescaler watchdog timera ustawiony na 1:32768

FWPSA = PR128 - Prescaler A watchdog timera ustawiony na 1:128

WINDIS = ON - Windowed watchdog timer wyłączony

FWDTEN = OFF - Watchdog timer całkowicie wyłączony

ICS = PGx2 - Komunikacja z debuggerem przez piny PGC2/PGD2

GWRP = OFF - Ochrona zapisu do pamięci programu wyłączona

GCP = OFF - Ochrona odczytu kodu wyłączona

JTAGEN = OFF - Interfejs JTAG wyłączony

## Biblioteki

```
#include <xc.h>
#include <libpic30.h>
#include <stdbool.h>
```

#include <xc.h> - Główna biblioteka dla kompilatorów XC

#include <libpic30.h> - Biblioteka specyficzna dla PIC30, zawiera funkcje opóźnień

#include <stdbool.h> - Biblioteka standardowa C dla typu bool (true/false)

## Definicje i stałe

```
#define FCY 4000000UL
#define DEBOUNCE_DELAY 200000
#define button_wylacz PORTDbits.RD6
#define NASTAWA_ALARM 512
#define OPOZNIENIE_MRUGANIA 500000
```

#define FCY 4000000UL - Częstotliwość procesora 4 MHz

#define DEBOUNCE\_DELAY 200000 - Opóźnienie dla eliminacji drgań styków

#define button\_wylacz PORTDbits.RD6 - Przycisk wyłączenia alarmu

#define NASTAWA\_ALARM 512 - Próg załączenia alarmu (połowa zakresu ADC)

#define OPOZNIENIE\_MRUGANIA 500000 - Opóźnienie dla mrugania LED w fazie 1

## Konfiguracja wejść/wyjść i ADC

```
TRISA = 0x0000;
TRISD = 0xFFFF;

AD1PCFG = 0xFFDF;
AD1CON1 = 0x00E0;
AD1CON2 = 0;
AD1CON3 = 0x1F3F;
AD1CHS = 5;
AD1CON1bits.ADON = 1;
```

TRISA = 0x0000; - Port A jako wyjście (sterowanie LED)

TRISD = 0xFFFF; - Port D jako wejście (odczyt przycisków)

AD1PCFG = 0xFFDF; - Pin AN5 jako analogowy, pozostałe cyfrowe

AD1CON1 = 0x00E0; - Format danych, tryb próbkowania

AD1CON2 = 0; - Referencja napięcia i tryb skanowania

AD1CON3 = 0x1F3F; - Czas konwersji i próbkowania

AD1CHS = 5 - Wybór kanału AN5 dla konwersji

AD1CON1bits.ADON = 1 - Włączenie przetwornika ADC

\_\_delay32(1000); - STABILIZACJA ADC

## potencjometr() - Odczyt wartości z przetwornika ADC

```
uint16_t potencjometr() {  
    AD1CON1bits.SAMP = 1;  
    __delay32(100);  
    AD1CON1bits.SAMP = 0;  
    while (!AD1CON1bits.DONE);  
    return ADC1BUF0;  
}
```

Definicje:

- AD1CON1bits.SAMP = 1; - Rozpoczęcie próbkowania
- \_\_delay32(100); - Czas stabilizacji próbkowania
- AD1CON1bits.SAMP = 0; - Zakończenie próbkowania, start konwersji
- while (!AD1CON1bits.DONE); - Oczekiwanie na zakończenie konwersji
- return ADC1BUF0; - Zwrócenie wyniku konwersji

Zasada działania:

- Rozpoczyna próbkowanie sygnału analogowego z potencjometru
- Czeka 100 cykli na stabilizację
- Kończy próbkowanie i rozpoczyna konwersję ADC
- Oczekuje na zakończenie konwersji
- Zwraca 10-bitową wartość z bufora

## Algorytm systemu alarmowego

```
while (1) {  
    wartosc_pot = potencjometr();  
  
    if (stan_alarmu == 0 && wartosc_pot > NASTAWA_ALARM) {  
        stan_alarmu = 1;  
        licznik_5s = 0;  
        dioda_stan = 0;  
    }  
  
    if (stan_alarmu > 0 && wartosc_pot <= NASTAWA_ALARM) {  
        stan_alarmu = 0;  
        LATA = 0x0000; |  
        licznik_5s = 0;  
    }  
}
```

### Definicje

wartosc\_pot = potencjometr() - Odczyt aktualnej wartości ADC

Przejście ze stanu normalnego do ostrzegawczego

- stan\_alarmu = 1; - Aktywacja fazy ostrzegawczej
- licznik\_5s = 0; - Reset licznika czasu
- dioda\_stan = 0; - Reset stanu mrugania

Dezaktywacja alarmu przez potencjometr

- stan\_alarmu = 0; - Powrót do stanu normalnego
- LATA = 0x0000; - Wyłączenie wszystkich LED
- licznik\_5s = 0; - Reset licznika

## Wyłączenie przyciskiem z debouncing

```
if (!button_wylacz && stan_alarmu > 0) {  
    __delay32(DEBOUNCE_DELAY);  
    if (!button_wylacz) {  
        stan_alarmu = 0;  
        LATA = 0x0000;  
        licznik_5s = 0;  
        while (!button_wylacz);  
        __delay32(DEBOUNCE_DELAY);  
    }  
}
```

Definicje:

- \_\_delay32(DEBOUNCE\_DELAY); - Pierwsze opóźnienie 200000 cykli
- if (!button\_wylacz) - Potwierdzenie stanu przycisku
- stan\_alarmu = 0; - Reset automatu
- LATA = 0x0000; - Wyłączenie LED
- licznik\_5s = 0; - Reset licznika
- while (!button\_wylacz); - Oczekiwanie na zwolnienie
- \_\_delay32(DEBOUNCE\_DELAY); - Drugie opóźnienie

Zasada działania:

- Dwukrotna weryfikacja stanu przycisku z opóźnieniem
- Eliminacja fałszywych wyzwoleń spowodowanych drganiami styków
- Blokowanie do momentu zwolnienia przycisku
- Działa tylko gdy system jest w stanie alarmowym (stan\_alarmu > 0)

## Stan alarmu

```
switch (stan_alarmu) {
    case 0:
        LATA = 0x0000;
        break;

    case 1:
        if (dioda_stan == 0) {
            LATA = 0x0001;
            dioda_stan = 1;
        } else {
            LATA = 0x0000;
            dioda_stan = 0;
        }

        __delay32(OPOZNIENIE_MRUGANIA);
        licznik_5s++;

        if (licznik_5s >= 40) {
            stan_alarmu = 2;
        }
        break;

    case 2:
        LATA = 0x00FF;
        __delay32(100000);
        break;
}
```

### Stan 0 - Tryb normalny:

- LATA = 0x0000 - Wszystkie LED na porcie A wyłączone
- System oczekuje na przekroczenie progu potencjometru

### Stan 1 - Mruganie:

- Algorytm toggle: Zmienna dioda\_stan kontroluje stan LED
- dioda\_stan == 0 - Włącz LED, dioda\_stan = 1
- dioda\_stan == 1 - Wyłącz LED, dioda\_stan = 0
- OPOZNIENIE\_MRUGANIA = 500000 cykli
- Częstotliwość mrugania: około 4 Hz
- licznik\_5s++ po każdym cyklu mrugania
- Automatyczne przejście po 5 sekundach

### Stan 2 - Alarm:

- LATA = 0x00FF - Wszystkie 8 LED świecą jednocześnie (bity 0-7)
- \_\_delay32(100000) - Krótkie opóźnienie dla odciążenia procesora
- Stan utrzymuje się do wyłączenia