**Raport do zadania 2**

Imię i nazwisko: Jakub Dmochowski

Numer Albumu: 169236

**Konfiguracja sprzętowa**

**A table with black text

AI-generated content may be incorrect.**

POSCMOD = XT - wybór zewnętrznego kwarcu jako źródła zegara

OSCIOFNC = ON - funkcja wyjścia oscylatora włączona

FCKSM = CSDCMD - wyłączenie monitorowania zegara i przełączania

FNOSC = PRI - Podstawowe źródło zegara jako domyślne

IESO = ON - włączenie wewnętrznego/zewnętrznego przełączania oscylatora

**Watchdog timer, debugowanie**

**A table of text with different colored letters

AI-generated content may be incorrect.**

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**

**A group of yellow text

AI-generated content may be incorrect.**

#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 zmienne globalne**

**A close-up of a code

AI-generated content may be incorrect.**

#define FCY 4000000UL - Częstotliwość procesora 4 MHz

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

#define buttonnext PORTDbits.RD6 - Przycisk "następny"

#define buttonprev PORTDbits.RD13 - Przycisk "poprzedni"

unsigned char tryb = 1; - Tryb pracy (1 lub 2)

unsigned long delay\_value = 500000; - Zmienna wartość opóźnienia sterowana potencjometrem

**Konfiguracja wejść/wyjść i ADC**

**A white background with black text

AI-generated content may be incorrect.**

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

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

AD1PCFG = 0xFFDF; - Pin AN5 jako analogowy, pozostałe cyfrowe (bit 5 = 0)

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

**potencjometr() - Odczyt wartości z przetwornika ADC**

**A screen shot of a computer code

AI-generated content may be incorrect.**

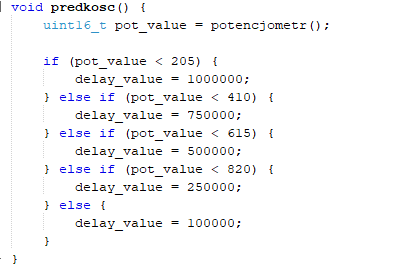
Definicje:

* 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 (0-1023)

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 (0-1023) z bufora

**predkosc() - Kontrola prędkości na podstawie potencjometru**

****

Zasada działania

* Odczytuje wartość z potencjometru (0-1023)
* Dzieli zakres na 5 przedziałów o równej szerokości (~20% każdy)
* Ustawia wartość opóźnienia odwrotnie proporcjonalnie do pozycji potencjometru
* Im wyższa wartość potencjometru, tym mniejsze opóźnienie (wyższa prędkość)

**funk3() - 8-bitowy licznik Gray z kontrolą prędkości**

**A computer code with black text

AI-generated content may be incorrect.**

Definicje:

* predkosc(); - Aktualizacja prędkości na podstawie potencjometru
* \_\_delay32(delay\_value); - Opóźnienie sterowane potencjometreu

Zasada działania:

* Implementuje 8-bitowy licznik Gray zliczający w górę
* W każdej iteracji wywołuje predkosc() dla aktualizacji tempa
* Konwertuje liczbę binarną do kodu Gray: wynik\_gray = licznik\_normalny ^ (licznik\_normalny >> 1)
* Opóźnienie zmienne w zakresie 100000-1000000 cykli w zależności od potencjometru
* Kończy działanie po wciśnięciu dowolnego przycisku

**funk7() - Wężyk z kontrolą prędkości**

**A screenshot of a computer code

AI-generated content may be incorrect.**

Definicje:

* predkosc(); - Aktualizacja prędkości na podstawie potencjometru
* \_\_delay32(delay\_value); - Opóźnienie sterowane potencjometrem
* wezyk = wezyk << 1; - Przesunięcie w lewo
* kierunek\_w\_prawo = 0; - Zmiana kierunku
* wezyk = wezyk >> 1- Przesunięcie w prawo
* kierunek\_w\_prawo = 1; - Zmiana kierunku

Zasada działania:

* Implementuje funkcje wężyk
* W każdej iteracji wywołuje predkosc() dla aktualizacji tempa
* Rozpoczyna z wzorem 0b00000111 (3 LED po prawej stronie)
* Przesuwa wzór w lewo do momentu osiągnięcia 0b11100000
* Następnie zmienia kierunek i przesuwa w prawo do 0b00000111
* Oscyluje między skrajnymi pozycjami
* Opóźnienie zmienne sterowane potencjometrem

**Pętla while w main()**

**A computer screen shot of a code

AI-generated content may be incorrect.**

Program działa w nieskończonej pętli, wykonując wybraną funkcję do momentu wciśnięcia przycisku, sprawdza który przycisk został wciśnięty po czym zmienia tryb.

* 1 - Licznik Gray
* 2 - Wężyk

**Debouncing**

A screenshot of a computer program

AI-generated content may be incorrect.

RD6:

* Przełącza z trybu 1 na 2, z trybu 2 na 1
* Implementuje debouncing z opóźnieniem 200000 cykli

RD13:

* Przełącza z trybu 2 na 1, z trybu 1 na 2
* Implementuje debouncing z opóźnieniem 200000 cykli