

# Raport do zadania 4

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 <string.h>
#include "lcd.h"
```

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

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

#include <stdbool.h> - Biblioteka standardowa C dla typu bool

#include <string.h> - Biblioteka do obsługi stringów

#include "lcd.h" - Biblioteka do obsługi LCD

## Definicje i zmienne globalne

```
#define FCY 4000000UL
```

#define FCY 4000000UL - Częstotliwość procesora 4 MHz

## Funkcja czekaj

```
void czekaj(unsigned int ms) {  
    unsigned int i, j;  
    for(i = 0; i < ms; i++) {  
        for(j = 0; j < 1000; j++) {  
            Nop();  
        }  
    }  
}
```

void czekaj(unsigned int ms) - Funkcja realizująca opóźnienie w milisekundach

for(i = 0; i < ms; i++) - Pętla zewnętrzna dla liczby milisekund

for(j = 0; j < 1000; j++) - Pętla wewnętrzna dla generowania opóźnienia

Nop(); - Instrukcja pustej operacji dla wypełnienia czasu

## Funkcja przewijanie

```
void przewijanie(const char* tekst_gora, const char* tekst_dol, unsigned int szybkosc, unsigned int cykle) {
    char bufor_dol[17];
    int rozmiar_dol = strlen(tekst_dol);
    int pozycja;
    unsigned int i;

    char spacja[200];
    strcpy(spacja, "                ");
    strcat(spacja, tekst_dol);
    strcat(spacja, "                ");
    int pelny_rozmiar = strlen(spacja);

    for(i = 0; i < cykle; i++) {
        for (pozycja = 0; pozycja <= pelny_rozmiar - 16; pozycja++) {
            LCD_ClearScreen();

            LCD_PutString((char*)tekst_gora, strlen(tekst_gora));

            LCD_PutChar('\n');

            int j;
            for (j = 0; j < 16; j++) {
                if (pozycja + j < pelny_rozmiar) {
                    bufor_dol[j] = spacja[pozycja + j];
                } else {
                    bufor_dol[j] = ' ';
                }
            }
            bufor_dol[16] = '\0';

            LCD_PutString(bufor_dol, 16);

            czeka(j(szybkosc));
        }
    }
}
```

void przewijanie - Funkcja realizująca animację przewijania tekstu na wyświetlaczu

bufor\_dol[17]; - Bufor dla dolnej linii wyświetlacza

rozmiar\_dol = strlen(tekst\_dol); - Obliczenie długości tekstu do przewijania

pozycja; - Zmienna pozycji aktualnego okna przewijania

spacja[200]; - Bufor dla tekstu z dodanymi spacjami

strcpy(spacja, " "); - Dodanie 16 spacji na początku

strcat(spacja, tekst\_dol); - Dołączenie tekstu do przewijania

strcat(spacja, " "); - Dodanie 16 spacji na końcu

pelny\_rozmiar = strlen(spacja); - Obliczenie całkowitej długości bufora

**for(i = 0; i < cykle; i++) - Pętla dla określonej liczby cykli przewijania**

**for (pozycja = 0; pozycja <= pelny\_rozmiar - 16; pozycja++) - Pętla przewijania przez wszystkie pozycje**

- LCD\_ClearScreen(); - Wyczyszczenie ekranu wyświetlacza
- LCD\_PutString((char)tekst\_gora, strlen(tekst\_gora)); - Wyświetlenie tekstu górnej linii
- LCD\_PutChar('\n'); - Przejście do nowej linii
- for (j = 0; j < 16; j++) - Pętla kopiowania 16 znaków do bufora dolnej linii
- if (pozycja + j < pelny\_rozmiar) - Sprawdzenie czy nie wykraczamy poza bufor
- bufor\_dol[j] = spacja[pozycja + j]; - Kopiowanie znaku z odpowiedniej pozycji
- bufor\_dol[j] = ' '; - Wypełnienie spacją jeśli brak znaku
- bufor\_dol[16] = '\0'; - Dodanie znaku końca ciągu
- LCD\_PutString(bufor\_dol, 16); - Wyświetlenie dolnej linii
- czekaj(szybkosc); - Opóźnienie między klatkami animacji

## Funkcja wyświetlenia

```
void wyswietlanie1(void) {  
    przewijanie("RASHED KEBAB!", "KEBAB TYLKO 15 ZLOTYCH! SWIEZE MIESO!", 200, 2);  
}  
  
void wyswietlanie2(void) {  
    przewijanie(">> PROMOCJA <<", "KEBAB+COLA 20ZL - ul. Kowalskiego obok Biedronki", 180, 2);  
}  
  
void wyswietlanie3(void) {  
    przewijanie("CZYNNE 10-3", "ZAMOW: 123-456-789 - DOSTAWA GRATIS!", 220, 2);  
}
```

wyswietlanie1(void) - Funkcja wyświetlająca pierwszą reklamę

- Wywołanie przewijania z nazwą lokalu, ofertą, opóźnieniem 200ms i 2 cyklami

wyswietlanie2(void) - Funkcja wyświetlająca pierwszą reklamę

- Wywołanie przewijania z nagłówkiem promocji, szczegółami oferty, opóźnieniem 180ms i 2 cyklami

wyswietlanie3(void) - Funkcja wyświetlająca pierwszą reklamę

- Wywołanie przewijania z godzinami otwarcia, kontaktem, opóźnieniem 220ms i 2 cyklami

## Główna funkcja main

```
int main(void) {  
    unsigned int i;  
    for(i = 0; i < 10000; i++) {  
        Nop();  
    }  
  
    LCD_Initialize();  
    LCD_CursorEnable(false);  
  
    while (1) {  
        wyswietlanie1();  
        wyswietlanie2();  
        wyswietlanie3();  
        LCD_ClearScreen();  
        czekaj(1000);  
    }  
  
    return 0;  
}
```

int main(void) - Punkt wejścia do programu,

**for(i = 0; i < 10000; i++) – Wykonuje 10000 pustych cykli, żeby dać czas mikrokontrolerowi na ustabilizowanie się**

- Nop(); - Każda iteracja pętli wykonuje jedną pustą operację, która nie robi nic ale zabiera czas procesora

LCD\_Initialize(); - Uruchomienie i skonfigurowanie wyświetlacza LCD

LCD\_CursorEnable(false); - Ukrycie migającego kursora na ekranie

**while (1) - Rozpoczęcie pętli, która będzie działać bez końca**

- wyswietlanie1(); - Pokazanie pierwszej reklamy
- wyswietlanie2(); - Przejście do drugiej
- wyswietlanie3(); - Wyświetlenie trzeciej reklamy
- LCD\_ClearScreen(); - Wyczyszczenie całego ekranu po pokazaniu wszystkich trzech reklam
- czekaj(1000); - Odczekanie jednej sekundy przerwy przed rozpoczęciem kolejnego cyklu reklam

