

1 Pracovní úkoly

1. Seznamte se s modulem pulzní šířkové modulace.
2. Vytvořte zdrojový soubor v jazyce C využívající tento modul.
3. Program přeložte a vložte do paměti mikrokontroléru.

2 Vypracování

Řešená úloha: Ovládání rychlosti otáček motoru větráčku

```
#include <xc.h>
#include<stdint.h>
#include <pic16f877a.h>

#define _XTAL_FREQ 3276800           // Frekvence krystalu

// CONFIG
#pragma config FOSC = XT           // Oscillator Selection bits (XT oscillator)
#pragma config WDTE = OFF          // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF         // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF         // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF           // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (I
#pragma config CPD = OFF           // Data EEPROM Memory Code Protection bit (Data EEPROM code protection of
#pragma config WRT = OFF           // Flash Program Memory Write Enable bits (Write protection off; all prog
#pragma config CP = OFF            // Flash Program Memory Code Protection bit (Code protection off)

void __interrupt() preruseni(void)
{
    //...
}

void set_duty_cycle(uint16_t knoflik)
{
    // podle hodnoty A/D nastavi bity stridy:

    uint16_t vysledek = (knoflik/(1023/200))*3.276800;

    CCP1CONbits.CCP1Y = vysledek % 2;           // 10. bit stridy
    vysledek = vysledek / 2;                     // odseknou posledni cislo
    CCP1CONbits.CCP1X = vysledek % 2;           // 9. bit stridy
    vysledek = vysledek / 2;
    CCPR1L = vysledek;                          // MSB stridy
```

```

}

void main(void)
{
    uint16_t vysledekAD = 0;

    TRISCbits.TRISC2 = 0;                // 2. bit TRISC = 0 -> CCP1 = output
    CCP1CONbits.CCP1M3 = 1;              //      nastaveni TMR2 na PWM mode
    CCP1CONbits.CCP1M2 = 1;              //      nastaveni TMR2 na PWM mode
    T2CONbits.TMR2ON = 1;                // zapnuti TMR2

    PR2 = 0xA4;                          // nastaveni frekvence
    T2CONbits.T2CKPS1 = T2CONbits.T2CKPS0 = 0; // TMR 2 prescale = 1
    /*=====*/
    TRISAbits.TRISA2 = 1;                // nastaveni A/D prevodniku
    ADCON1 = 0b11000000;
    ADCON0 = 0b10010001;
    TRISD = 0;
    TRISB = 0b11101000;

    PORTB = 0xFF;
    /*=====*/

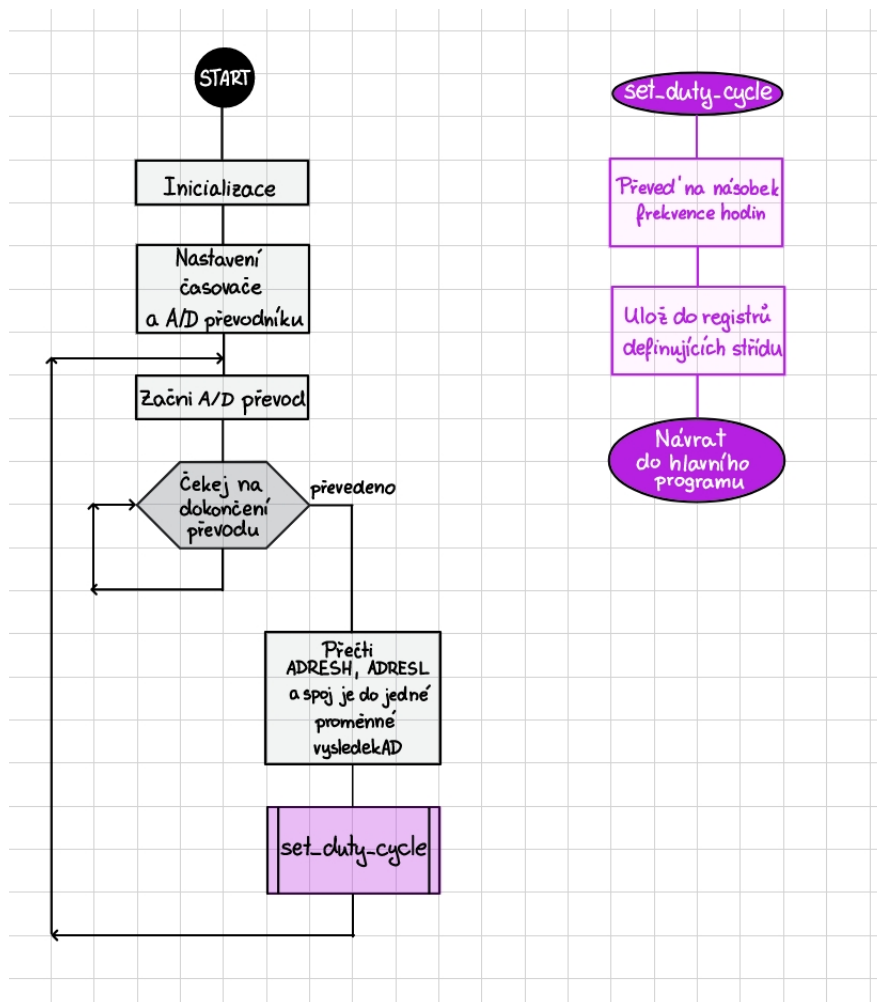
    while(1)
    {
        ADCON0bits.GO = 1;
        while(ADCON0bits.GO != 0)
            ;

        vysledekAD = (ADRESH << 8) + ADRESL;

        set_duty_cycle(vysledekAD);
    }

    return;
}

```



Obr. 1: Diagram vypracování úlohy – Ovládání rychlosti otáček motoru.