

Jméno: **Kateřina Hrnecková** Měřeno: **11.11.2022**

Klasifikace:

## 1 Pracovní úkoly

1. Seznamte se s překladačem MPLAB XC8: Datové typy, operátory, standardní knihovny, knihovní funkce, ...
2. Vytvořte jednoduchý zdrojový soubor v jazyce C, např. ovládání LED diody na desce PVK40. Program přeložte a vložte do paměti mikrokontroléru.
3. Přepište program z úlohy č. 3 (časovač a přerušení) nebo č.4 (A/D převodník) do jazyka C a porovnejte počet instrukcí.

## 2 Vypracování

Řešená úloha: Voltmetr

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

#define _XTAL_FREQ 3276800

// 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)
{
    //...
}

uint8_t Table(uint8_t c)
{
    /*Table:  addwf    pcl,f          ;Display segments table
    retlw    B'11000000'      ;0
    retlw    B'11111001'      ;1
    retlw    B'10100100'      ;2
    retlw    B'10110000'      ;3
    retlw    B'10011001'      ;4
    retlw    B'10010010'      ;5
    retlw    B'10000010'      ;6
```

```

        retlw    B'11111000'    ;7
        retlw    B'10000000'    ;8
        retlw    B'10010000'    ;9
        retlw    B'11111111'    ;display off
;*/

switch(c)
{
    case 0:
        return 0b11000000;
    case 1:
        return 0b11111001;
    case 2:
        return 0b10100100;
    case 3:
        return 0b10110000;
    case 4:
        return 0b10011001;
    case 5:
        return 0b10010010;
    case 6:
        return 0b10000010;
    case 7:
        return 0b11111000;
    case 8:
        return 0b10000000;
    case 9:
        return 0b10010000;
    default:
        return 0b11111111;
}

}

void main(void)
{
    uint8_t cislice = 0;
    uint16_t vysledekAD = 0;

    TRISAbits.TRISA2 = 1;
    ADCON1 = 0b11000000;
    ADCON0 = 0b10010001;
    TRISD = 0;
    TRISB = 0b11101000;

    PORTB = 0xFF;

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

        vysledekAD = (ADRESH << 8) + ADRESL;
        vysledekAD = vysledekAD*5;
        // PRVNI CISLICE

```

```

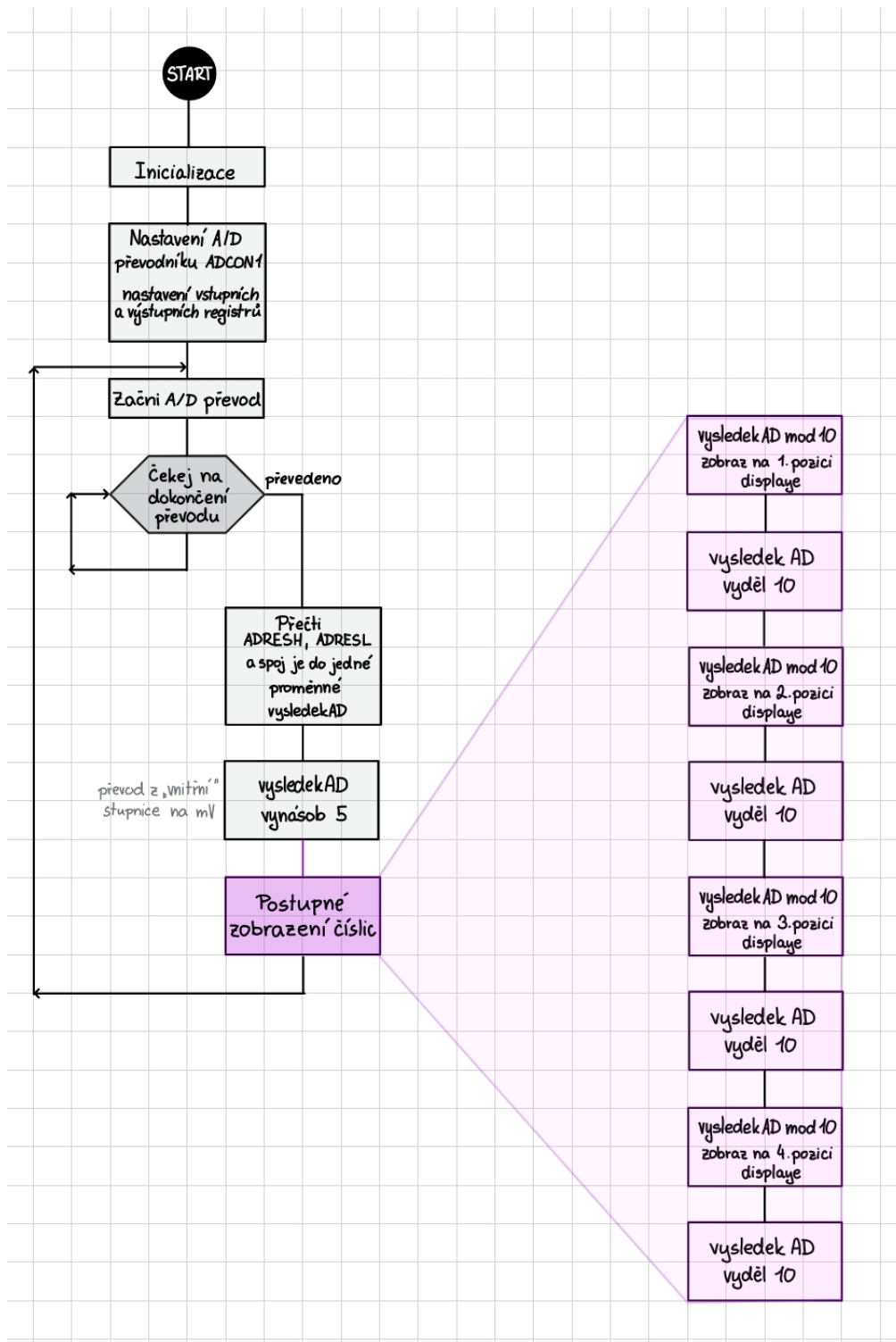
        cislice = vysledekAD%10;
        PORTD = Table(cislice);
        vysledekAD = vysledekAD/10;
        PORTBbits.RB0 = 0;
        __delay_ms(5);
        PORTBbits.RB0 = 1;

        //DRUHA CISLICE
        cislice = vysledekAD%10;
        PORTD = Table(cislice);
        vysledekAD = vysledekAD/10;
        PORTBbits.RB1 = 0;
        __delay_ms(5);
        PORTBbits.RB1 = 1;

        //TRETI CISLICE
        cislice = vysledekAD%10;
        PORTD = Table(cislice);
        vysledekAD = vysledekAD/10;
        PORTBbits.RB2 = 0;
        __delay_ms(5);
        PORTBbits.RB2 = 1;

        //CTVRTA CISLICE
        cislice = vysledekAD%10;
        PORTD = Table(cislice);
        vysledekAD = vysledekAD/10;
        PORTBbits.RB4 = 0;
        __delay_ms(5);
        PORTBbits.RB4 = 1;
    }
    return;
}

```



Obr. 1: Diagram vypracování úlohy – Voltmetr.