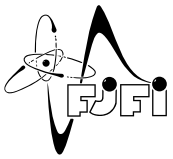
Mikroprocesorové praktikum **FJFI ČVUT v Praze**

Úloha 10 – Jednočipový mikrokontrolér PIC18F45K20




Jméno: **Kateřina Hrnečková** Měřeno: **19.5.2023** Klasifikace:

---

1 Pracovní úkoly

1. Seznamte se s vlastnostmi a vnitřní strukturou mikrokontroléru PIC18F45K20.
2. Seznamte se s vývojovou deskou PICkit 44-pin Demo Board.
3. Vytvořte program v jazyce C využívající periferie na desce (LED, tlačítko, trimer).
4. Program přeložte a vložte do paměti mikrokontroléru.

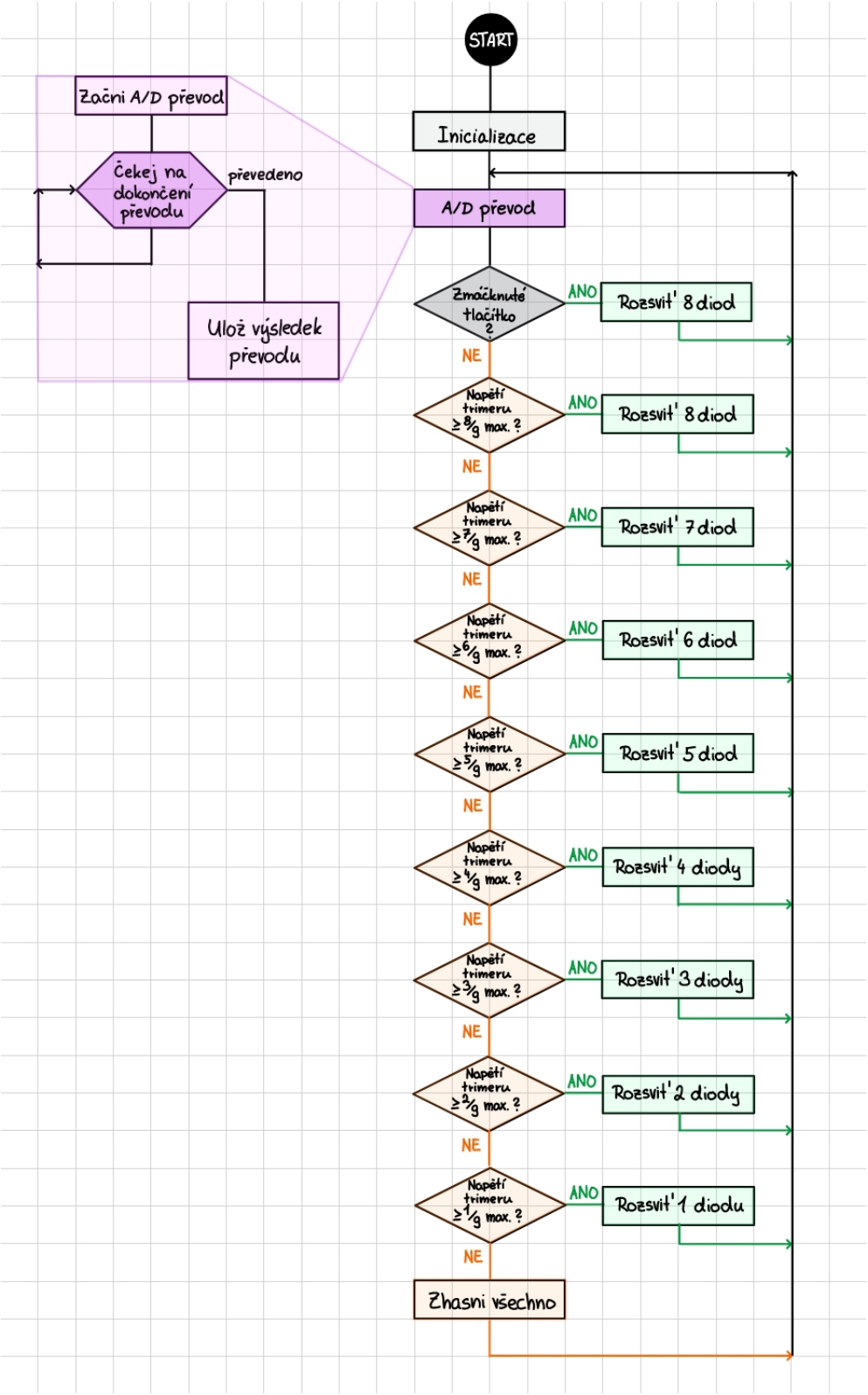
2 Vypracování

Řešená úloha: Rozsvícení diod podle napětí na trimeru.

```
// PIC18F45K20 Configuration Bit Settings

// 'C' source line config statements

// CONFIG1H
#pragma config FOSC = INTIO67   // Oscillator Selection bits (Internal oscillator block, port function on
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover bit (Oscillator Switchover mod

// CONFIG2L
#pragma config PWRT = ON        // Power-up Timer Enable bit (PWRT enabled)
#pragma config BOREN = OFF      // Brown-out Reset Enable bits (Brown-out Reset disabled in hardware and s
#pragma config BORV = 18        // Brown Out Reset Voltage bits (VBOR set to 1.8 V nominal)

// CONFIG2H
#pragma config WDTEN = OFF      // Watchdog Timer Enable bit (WDT is controlled by SWDTEN bit of the WDTC
#pragma config WDTPS = 32768    // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = PORTC   // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = OFF     // PORTB A/D Enable bit (PORTB<4:0> pins are configured as digital I/O on
#pragma config LPT1OSC = OFF    // Low-Power Timer1 Oscillator Enable bit (Timer1 configured for higher p
#pragma config HFOFST = ON      // HFINTOSC Fast Start-up (HFINTOSC starts clocking the CPU without waiti
#pragma config MCLRE = ON       // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON      // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause
#pragma config LVP = ON         // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config XINST = OFF      // Extended Instruction Set Enable bit (Instruction set extension and Ind

// CONFIG5L
#pragma config CP0 = OFF        // Code Protection Block 0 (Block 0 (000800-001FFFh) not code-protected)
```

```
#pragma config CP1 = OFF        // Code Protection Block 1 (Block 1 (002000-003FFFh) not code-protected)
#pragma config CP2 = OFF        // Code Protection Block 2 (Block 2 (004000-005FFFh) not code-protected)
#pragma config CP3 = OFF        // Code Protection Block 3 (Block 3 (006000-007FFFh) not code-protected)

// CONFIG5H
#pragma config CPB = OFF        // Boot Block Code Protection bit (Boot block (000000-0007FFh) not code-pr
#pragma config CPD = OFF        // Data EEPROM Code Protection bit (Data EEPROM not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF       // Write Protection Block 0 (Block 0 (000800-001FFFh) not write-protected)
#pragma config WRT1 = OFF       // Write Protection Block 1 (Block 1 (002000-003FFFh) not write-protected)
#pragma config WRT2 = OFF       // Write Protection Block 2 (Block 2 (004000-005FFFh) not write-protected)
#pragma config WRT3 = OFF       // Write Protection Block 3 (Block 3 (006000-007FFFh) not write-protected)

// CONFIG6H
#pragma config WRTC = OFF       // Configuration Register Write Protection bit (Configuration registers (3
#pragma config WRTB = OFF       // Boot Block Write Protection bit (Boot Block (000000-0007FFh) not write-
#pragma config WRTD = OFF       // Data EEPROM Write Protection bit (Data EEPROM not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF      // Table Read Protection Block 0 (Block 0 (000800-001FFFh) not protected f
#pragma config EBTR1 = OFF      // Table Read Protection Block 1 (Block 1 (002000-003FFFh) not protected f
#pragma config EBTR2 = OFF      // Table Read Protection Block 2 (Block 2 (004000-005FFFh) not protected f
#pragma config EBTR3 = OFF      // Table Read Protection Block 3 (Block 3 (006000-007FFFh) not protected f

// CONFIG7H
#pragma config EBTRB = OFF      // Boot Block Table Read Protection bit (Boot Block (000000-0007FFh) not p

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#define _XTAL_FREQ 16000000                // Frekvence krystalu

#include <xc.h>

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

void main(void)
{
        uint16_t vysledekAD = 0;
        uint16_t max = 0b0000001111111111;

        ADCON0 = 0b00000001;
        ADCON1 = 0b00000000;
        ADCON2 = 0b10111110;

        TRISD = 0;
        while(1)
        {

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

```
                ;
            vysledekAD = (ADRESH << 8) + ADRESL;

            if (PORTBbits.RB0 == 0)
                LATD=0b11111111;
            else
            {
                if (vysledekAD >= (8*max)/9)
                    LATD=0b11111111;
                else if (vysledekAD >= (7*max)/9)
                    LATD=0b11111110;
                else if (vysledekAD >= (6*max)/9)
                    LATD=0b11111100;
                else if (vysledekAD >= (5*max)/9)
                    LATD=0b11111000;
                else if (vysledekAD >= (4*max)/9)
                    LATD=0b11110000;
                else if (vysledekAD >= (3*max)/9)
                    LATD=0b11100000;
                else if (vysledekAD >= (2*max)/9)
                    LATD=0b11000000;
                else if(vysledekAD >= max/9)
                    LATD=0b10000000;
                else LATD = 0;
            }

        }
        return;
}
```



Obr. 1: Diagram vypracování úlohy – Rozsvícení diod podle napětí na trimeru.