

```

#include "relay_all.h"
#include "relay-14k50.h"

/-- Magas prioritású programmegszakítás kiszolgálása
#pragma interrupt hi_isr
void hi_isr() {
    USBDeviceTasks(); /--- USB programmegszakítás kiszolgálása
}
/-- Alacsony prioritású programmegszakítás kiszolgálása
#pragma interruptlow lo_isr
void lo_isr() {
}

/** A PIC18F14K50 mikrovezérlő AN4 - AN11 csatornájának engedélyezése, és az
 * ADC inicializálása (AN6 kiválasztása, órajel FOSC/64, TACQ = 20TAD,
 * konverziós eredmény jobbra igazítása).
 */
void InitADC(void) {
    CONFIG_AN4_AS_ANALOG();
    CONFIG_AN5_AS_ANALOG();
    CONFIG_AN6_AS_ANALOG();
    CONFIG_AN7_AS_ANALOG();
    CONFIG_AN8_AS_ANALOG();
    CONFIG_AN9_AS_ANALOG();
    CONFIG_AN10_AS_ANALOG(); //AN10(RB4) analóg bemenet beállítása
    CONFIG_AN11_AS_ANALOG(); //AN11(RB5) analóg bemenet beállítása
    ADCON0=0x18; //AN6 kiválasztás, ADC még letiltva
    ADCON1=0x8; //VREF+ = FVR, VREF- = VSS
    ADCON2=0xBE; //20TAD, FOSC/64, jobbra igazítás
    ADCON0bits.ADON=1; //Az ADC engedélyezése
}

/***** Egy analóg csatorna kiválasztása, és megmérése.
 * A függvény blokkoló típusú: kivárja a mérés végét (~46 us).
 * \param chan a kiválasztott csatorna AN0, AN1, ... stb. jelölés szerinti sorszáma
 *****/
unsigned int ReadADC(unsigned char chan)
{
    ChangeBits(ADCON0, chan<<2, 0b00111100);
    ADCON0bits.GO = 1; // Start AD conversion
    while (ADCON0bits.NOT_DONE); // Wait for conversion
    return (((unsigned int)ADRESH)<<8) | (ADRESL);
}

void main(void)
{
    unsigned int j,R1V,R1A,R2V,R2A,R3V,R3A,R4V,R4A;
    long A1V,A1A;

    InitializeSystem();

    REFCON0=FVR_ENABLE|FVR1S_1024; //FVR inicializálása 1.024 V-ra
    while(!FVR STATUS); //kivárjuk, amíg FVR stabil lesz
    InitADC(); //AN6 kiválasztása, 20TAD, FOSC/64, jobbra igazítás

    while (!usb cdc kbhit())
    {
        ProcessIO(); //--az elso karakter érkezésére várni
    }
    outString("Parancs karakterek a vezerleshez\n");
    outString("Csatorna\t1.\t2.\t3.\t4.\nBekapcs.\t1\t3\t5\t7\nKikapcs.\t2\t4\t6\t8\nMinden
csatorna Be->a Ki->t\nEz a tabla h\n");
    outString("Relays Panel Start\n");
    while (1)
    {
        A1A=0; A1V=0; // atlag nullazas A2A=0; A3A=0; A4A=0;A2V=0; A3V=0; A4V=0;

        for(j=0; j<256; j++)
        {
            R1A=ReadADC(6); //AN6 mérés
            R2A=ReadADC(5); //AN5 mérés
            R3A=ReadADC(4); //AN4 mérés
            R4A=ReadADC(7);
            R1V=ReadADC(8);
            R2V=ReadADC(9);
            R3V=ReadADC(11);
            R4V=ReadADC(10);

```

C:\Users\Szabolcs\Desktop\Usb\usb_cdc\Diploma\MPLab\relay-14k50.c

```
A1A +=ReadADC(6);
// A2A +=ReadADC(5);
// A3A +=ReadADC(4);
// A4A +=ReadADC(7);
A1V +=ReadADC(8);
// A2V +=ReadADC(9);
// A3V +=ReadADC(11);
// A4V +=ReadADC(10);

    outString("\n\n1A");
    outdec(R1A,3);           //Három tizedesre írjuk ki R1A-t

    outString("\t\t2A");
    outdec(R2A,3);           //Három tizedesre írjuk ki R2A-t

    outString("\t\t3A");
    outdec(R3A,3);

    outString("\t\t4A");
    outdec(R4A,3);

    R1V=R1V<<4;           //R1V mV-okra átszámítva
    outString("\n1V");
    outdec(R1V,3);           //Három tizedesre írjuk ki R1V-t

    R2V=R2V<<4;           //R2V mV-okra átszámítva
    outString("\t\t2V");
    outdec(R2V,3);

    R3V=R3V<<4;
    outString("\t\t3V");
    outdec(R3V,3);

    R4V=R4V<<4;
    outString("\t\t4V");
    outdec(R4V,3);

    delay ms(2000);

if(usb_cdc_kbhit())
    {
        switch (usb_cdc_getc())
        {
            case '1':
                LATCbits.LATC5 = 1;
            break;
            case '2':
                LATCbits.LATC5 = 0;
            break;
            case '3':
                LATCbits.LATC4 = 1;
            break;
            case '4':
                LATCbits.LATC4 = 0;
            break;
            case '5':
                LATBbits.LATB7 = 1;
            break;
            case '6':
                LATBbits.LATB7 = 0;
            break;
            case '7':
                LATBbits.LATB6 = 1;
            break;
            case '8':
                LATBbits.LATB6 = 0;
            break;
            case 'a':
                { LATC = 0B00110000;
                  LATB = 0B11000000; }
            break;
            case 't':
                { LATC = 0B00000000;
                  LATB = 0B00000000; }
            break;
            /** case 'h':

```

```
outString("\n\nCsatorna\t1.\t2.\t3.\t4.\nBekapcs.\t1\t3\t5\t7\nKikapcs.\t2\t4\t6\t8\nMinden csatorna
Be->a    Ki->t\n");
```

C:\Users\Szabolcs\Desktop\Usb\usb_cdc\Diploma\MPlab\relay-14k50.c

```
break;
*****/
    } //end (usb_cdc_getc())

    } //end (usb_cdc_kbhit())

} //end for

LATBbits.LATB6 = 0; // a periodus lefutását jelzi... később törölve lesz
delay ms(1000);
LATBbits.LATB6 = 1;

outString("\n\nAtlag");

    A1A=A1A>>4; //R1A mA-okra átszámítva
    outString("\n\n1A");
    outdec(A1A,3); //Három tizedesre írjuk ki R1A-t

    A1V=A1V>>4; // A1V mV-okra átszámítva
    outString("\n\n1V");
    outdec(A1V,3); //Három tizedesre írjuk ki A1V-t

} //while
} //main
```