

# Mikrokontroller III. mérési jegyzőkönyv

Mátyás ANTAL

(Supervisor: Attila TIHANYI)

Pázmány Péter Catholic University, Faculty of Information Technology and Bionics

50/a Práter street, 1083 Budapest, Hungary

antal.matyas.gergely@hallgato.ppke.hu

**Abstract**—A mérés célja volt a mikrokontrollerek gyakorlati megismerése, az MSP 430-196 mikrokontroller használata, egyszerű alpműveletek végrehajtása, ezzel a flag bitek működésének gyakorlati vizsgálata.

## I. MÉRENDŐ OBJEKTUMOK

A mérés során az MSP 430-196 mikrokontrollert, valamint egy számítógépet és az ezen futó programozási környezetet használtunk, ennek segítségével végeztünk a mikrokontrolleren egyszerű alpműveleteket, összeadást és kivonást, ezzel ismerkedve a műveletvégzéssel, valamint a flagek használatával. A mérés során a műveletek elvégzésére használt programrészleteket a jegyzőkönyvbe illeszttem, valamint az elkészült programot az emailhez csatolom.

## II. AD ÁTALAKÍTÓ KEZELÉSE I.

Az első feladat a mikrokontroller bal oldali potmétere által beállított feszültség kiírása volt a kijelzőre, ezt a *SetupADC12* parancs hívásával tudtuk megvalósítani, mely a potméterek értékét a *RightValue* illetve *LeftValue* regiszterekbe tölti. Innen a *mov.w* paranccsal helyeztük át az értéket az *R12* regiszterbe, ahonnan a *hexdraw* függvény hívásával a képernyőre rajzoltuk.

```
feladat1:
    call #SetupADC12
    mov.w LeftValue,R12
    call #hexdraw
    jmp feladat1
    ret
```

## III. AD ÁTALAKÍTÓ KEZELÉSE II.

A feladat az előzőhöz hasonlóan a potméter által beállított érték kiírása volt, ez esetben azonban a jobb oldali potméter értékét kellett a jobb felső sarokba. Mivel a *hexdraw* függvény csak a bal felső sarokba tud rajzolni, a korábbi mérésen már használt *LCDChrXY* függvény hívásával rajzoltunk megfelelő koordinátákra. Jelen függvény azonban csak egy karakter kiírására alkalmas, így a beállított számot helyiértékenként kellett különböző koordinátákra kirajzolnunk, ezt a jobb oldalon található módszerrel oldottuk meg:

## IV. HELYRE TÖRTÉNŐ KIÍRATÁS

A feladat a korábbi méréshez hasonlóan a kijelző adott pontjába való kirajzolás, illetve a karakter mozgatása volt, ez esetben nem a joystick, hanem a két potméter segítségével. A bal oldali potméterrel az X, a jobb oldalival az Y koordináta értékét tudtuk módosítani. A feladat megvalósításához itt is az *LCDChrXY* valamint az *LCDUpdate* függvényeket használtuk a jobb oldali potméter értékét az *R13*, a bal oldalit az *R12* regiszterbe töltve, a kiírandó karakter kódját pedig az 'O' betűre állítva. Minden mozgatás után az előző karaktert

```
feladat2:

    call #SetupADC12
    mov.w RightValue,R6
    call #hexdraw
    mov #1, R14
    mov R6, R12
    call #multiply

    mov R12, R14
    add #0x30, R14
    mov #0, R13
    mov #13, R12
    call #LCDChrXY

    mov R6, R12
    mov #8, R14
    mov #0, R13
    call #divide

    mov R12, R6
    mov #256, R14
    mov R6, R12
    call #multiply

    mov R12, R14
    add #0x30, R14
    mov #0, R13
    mov #12, R12
    call #LCDChrXY

    mov R6, R12
    mov #10, R14
    mov #0, R13
    call #divide

    mov R6, R12
    mov R12, R14
    add #0x30, R14
    mov #0, R13
    mov #11, R12
    call #LCDChrXY

    mov R6, R12
    mov #10, R14
    mov #0, R13
    call #divide

    mov R12, R6
    mov R12, R14
    add #0x30, R14
    mov #0, R13
    mov #10, R12
    call #LCDChrXY

    call #LCDUpdate
    mov R5, R12
    call #hexdraw

    jmp feladat2
    ret
```

kitöröltük egy 'space'-t írunk. A kódot a következő oldal tartalmazza.

## V. KÉT ÜTŐ ELKÉSZÍTÉSE

A pong játékhoz tartozó két ütő elkészítése az előző feladathoz hasonlóan történt, a két potméter értékét beolvasva határoztuk meg a jobb illetve bal oldali ütő csak függőleges irányú elmozdulását, majd ugyancsak az *LCDChrXY*

függvény segítségével rajzoltuk ki a két ütt alkotó egymás  
alatti két 0x23 karakterkódú ütt.

```
        mov.b #2, R4
        mov.b #7, R5
        mov.b #0x4F, R6

feladat3:
        mov.b R6, R14
        mov.w RightValue, R13
        mov.w LeftValue, R12
        call #LCDChrXY
        call #LCDUpdate
        mov.b #0x20, R14
        mov.b R4, R13
        mov.b R5, R12
        call #LCDChrXY
        jmp feladat3
```