Mikrokontroller III. jegyzőkönyv

A mérés adatai

A mérést végző személyek: Ekart Csaba, Kincs Boglárka
A mérés helye, ideje: PPKE-ITK 420-as mérőlabor, 2017.05.11. 12:15-15:00
Felhasznált mérőeszköz: MSP430F169 mikrokontroller

A mérés menete

Az előző mérésen elkészített projektet megnyitottuk az IAR Embedded Workbench-csel, a korábbi forráskódot kitöröltük, és a megfelelő részben elkezdtük írni az új programokat. A környezet debugger funkcióját alkalmaztuk a futtatás során, és folyamatosan ellenőriztük az egyes regiszterek értékét a jobboldali regiszter ablakban.

1. feladat – Az AD átalakító kezelése I.

A feladat szerint meg kellett valósítanunk egy olyan programot, amely a mikrokontroller panelen található baloldali potméter által beállított analóg feszültséget kiírja a grafikus kijelzőre.

KÓDRÉSZLET

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

Először meghívtuk a szükséges SetupADC12 függvényt, az AD konverter kezelőprogramját. A potméterek értéke ekkor a LeftValue, illetve RightValue címszavakkal kérhető le, ezt az R12-es regiszterbe másoltuk, majd a hexdraw függvénnyel kirajzoltuk a balfelső sarokba.

2. feladat – Az AD átalakító kezelése II.

A feladat nagyban hasonlított az előzőhöz, de a jobb oldali potméter kiírását is végre kellett hajtani a kijelző jobb felső részén.

KÓDRÉSZLET

```
asmmain:
    call #SetupADC12
    mov.w LeftValue,R6
    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 R12,R6
    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 asmmain
ret
```

A probléma természetesen azzal volt, hogy a hexdraw függvény csak a kijelző bal felső sarkába alkalmas kiíratásra, így a jobb oldali potméter értékének kiírásához nem használhattuk ezt. Az előző mérés során mikor egy tetszőleges karaktert kellett mozgatnunk a mikrokontroller kijelzőjén, már részben elkészítettük az ehhez szükséges kódokat, melyhez az LCDChrXY és az LCDUpdate függvényekre van szükség. A program gyakorlatilag az egyes helyiértéken lévő értékeket rajzolja ki a jobb felső saroknak megfelelő koordinátákra.

3. feladat – Potméterrel vezérelhető helyre történő kiírás

A mikrokontroller 2 mérésen megvalósított joystick kiírás alapján el kellet készíteni a potméterrel vezérelhető helyre történő kiírást. A baloldali potméterrel X, a jobb oldali potméterrel Y irányba mozgathatóvá kellet tenni.

KÓDRÉSZLET

```
    asmmain:
        mov.b #2,R4 ;kezdeti y koordináta
        mov.b #7,R5 ;kezdeti x koordináta
        mov.b #0x4F,R6 ;a kirajzolandó O betű kódja
    minta:
```

```
            mov.b R6,R14
            mov.w RightValue,R13
            mov.w LeftValue,R12
            call #LCDChrXY
            call #LCDUpdate
            mov.b #0x20,R14 ;törlés (space beírása a helyére)
            mov.b R4,R13
            mov.b R5,R12
            call #LCDChrXY
            jmp minta
        ret
```

A feladat megvalósítása nagyban hasonlított a mikrokontroller II. mérésen megvalósított feladathoz, azzal a különbséggel, hogy nem a joystick mozgatására, hanem a potméter tekerése változtattuk az x és y koordinátákat. Az LcdChrXY és az LCDUpdate függvényekez használtuk, hasonló logika mentén mint legutóbb. A jobb oldali potméter értékét az R13-as a baloldali potméter értékét az R12-es regiszterbe másoltuk, illetve a kiíratandó karakter kódját az O betűre állítottuk. Minden elmozdulás után az előző karaktert „kitöröljük”, vagyis a helyére egy space karaktert írunk.