Mikrokontroller III. mérési jegyzőkönyv

Mérést végezte:
Csutak Balázs, Farkas Viktória

Mérés helye és ideje: ITK 320. terem, 2016.05.02-09

1. A mérés célja:

Ismerkedés a mikrokontroller nyújtotta lehetőségekkel. Perifériák értékeinek vizsgálata. Potméterek értékeinek feldolgozása. Pingpong játék elkészítése.

2. Felhasznált eszközök:

- MSP430F169 Texas-alapú mikrokontroller
- IAR Embedded Workbench programcsomag

3. A mérés menete:

Összekötöttük a mikrokontrollert a számítógéppel, átmásoltuk a megadott könyvtárat a Dokumentumok mappába, majd megnyitottuk a projektet. A kódban megkerestük a szerkesztésre kijelölt részt, és ide rendre beírtuk a feladatok megoldásához szükséges néhány sornyi programot. A környezet debugger funkcióját használva lépésenként végignéztük a program futását, figyelve az egyes regiszterek értékét a jobboldali ablakban. A potmétereket használva kipróbáltuk (és javítottuk) a megírt programokat.

4. Feladatok megoldása:

4.1. AD átalakító kezelése

A feladat során a mikrokontroller potméter feszültségeit kellett vizsgálnunk, ennek értékét a kijelzőn megjelenítenünk. Ez lényegében három utasítással megvalósítható:

```
asmmain:
eleje:
        call #SetupADC12          ; betolti a potmeterek
                                  ; pillanatnyi allasat
        mov.w LeftValue,R12       ; kiirja az R12-bol
                                  ; a potmeter erteket
        call #hexdraw
        jmp eleje                 ; ismetles: megnezzuk,
                                  ; hogy valtozott-e az ertek
ret
```

4.2. AD átalakító kezelése II.

A jobb oldali potméter értékét sikerült lekérdezni (RightValue értéke), de a kiírás csak a bal felső sarokba tudtuk meghívni. A jobb sarokba íráshoz elkezdtünk egy saját LCDStrXY függvényt, amely a teljes regiszter értékét tudná magadott koordinátára írni, de ezt nem sikerült befejezni a gyakorlat alatt - helyette inkább a pingpong játékra koncentráltunk. Az alapelv az volt, hogy a regiszter értéket mindig 16-al osztva, sorban megkapjuk a számjegyeket, melyeket aztán az (x,y), (x+1,y), (x+2,y), (x+3, y) pontokra írunk ki az LCDChrXY függvénnyel.

4.3. Potmétervezérelt kiírás

```
asmmain:
      call #SetupADC12  ; potmeter ertekeinek lekerdezese

      mov.w RightValue,R12       ; jobb ertek osztasa 683-al
      mov #683,R14       ; a max ertek 0xFFF, ezt kepezzuk a [0,5]-be
      call #divide
      mov R12,R4

      call #SetupADC12
      mov.w LeftValue,R12
      mov #293,R14 ; hasonloan az elozohoz, igazodunk az LCD meretehez
      call #divide
      mov R12,R5

      mov R4,R13
      mov R5,R12
      mov #79,R14
      call #LCDChrXY     ; kiirjuk a karaktert
      call #LCDUpdate    ; frissitjuk a kiirt kepet

      mov R4,R13         ; kitoroljuk a karaktert
      mov R5,R12         ; nincs update, nem tunik el,
      mov #0x20,R14      ; csak mikor mar kiirtuk a kovetkezot
      call #LCDChrXY
```

```
jmp asmmain
ret
```

4.4. Ütők elkészítése

Az ütők kirajzolása két-két LCDChrXY függvényhívásból áll, a megfelelő koordinátákra. A program az ütő felső elemének koordinátáját tárolja, a második (alsó) elem kirajzolása az Y koordináta növelésével történik. Az assembly kód a következő feladatban látható.

4.5. Pingpong

Itt következik a feladatban kért pingpong játék kódja.

A program alapvető lépései:

- Előző állás törlése: labda és ütők
- Potméter adatok betöltése és átalakítása (képernyő méret)
- Labda helyének kiszámolása (pozíció += sebesség)
- Felső és alsó ütközés kezelése (függőleges sebesség *= -1)
- Bal és jobb oldali fallal ütközés: megfelelő pont növelése
- Ütőkvel való ütközés ellenőrzése: (vízszintes sebesség *= -1)
- Labda kirajzolása
- Ütők kirajzolása
- Pontok kiírása

Hiányosságok és megjegyzések:

- Csak egyszámjegyű pont kiírása (mert nem tudunk többet írni a jobb sarokba)
- Nem lehet újrakezdeni a játékot
- A kijelző tulajdonságai miatt a rövid ideig megjelenített labda nagyon halványan látszik
- Szokatlan módon, a függőleges az X, a vízszintes az Y tengely

asmmain:

```
; "valtozonevek" az atlathatosag erdekeben
#define bal_x R10
#define jobb_x R9
#define labda_x R8
#define labda_y R7
#define labda_dx R6
#define labda_dy R5
#define bal_pont R4
```

```
#define jobb_pont R11

; kezdeti ertekek feltoltese
mov.w #0,bal_x
mov.w #0,jobb_x
mov.w #2,labda_x
mov.w #12,labda_y
mov.w #1,labda_dx
mov.w #1,labda_dy
mov.w #0,bal_pont
mov.w #0,jobb_pont

main:
       //labda torlese
       mov labda_x,R13
       mov labda_y,R12
       mov #0x20,R14
       call #LCDChrXY

      //regi utok torlese
       mov bal_x,R13
       mov #0,R12
            mov #0x20,R14
       call #LCDChrXY

       mov bal_x,R13
       inc R13
       mov #0,R12
       mov #0x20,R14
       call #LCDChrXY

       mov jobb_x,R13
       mov #13,R12
       mov #0x20,R14
       call #LCDChrXY

       mov jobb_x,R13
       inc R13
       mov #13,R12
       mov #0x20,R14
       call #LCDChrXY

       //potmetrek adatainak betoltese es atalakitasa poziciova
       mov.w R5,R12
       call #SetupADC12
       mov.w R12,R5
       mov.w RightValue,R12
       mov #820,R14
       call #divide
       mov R12,jobb_x
```

```
        mov.w R5,R12
        call #SetupADC12
        mov.w R12,R5
        mov.w LeftValue,R12
        mov #820,R14
        call #divide
        mov R12,bal_x

        //a ket uto kirajzolasa
        mov bal_x,R13
        mov #0,R12
        mov #0x23,R14
        call #LCDChrXY

        mov bal_x,R13
        inc R13
        mov #0,R12
        mov #0x23,R14
        call #LCDChrXY

        mov jobb_x,R13
        mov #13,R12
        mov #0x23,R14
        call #LCDChrXY

        mov jobb_x,R13
        inc R13
        mov #13,R12
        mov #0x23,R14
        call #LCDChrXY


        //labda helyenek kiszamolasa
        add.w labda_dx,labda_x
        add.w labda_dy,labda_y

            // utkozes lent
        cmp.w #5,labda_x
        jge alul_felul_utkozes

        // utkozes fent
        cmp.w #0,labda_x
        jl alul_felul_utkozes

        cmp.w #14,labda_y
        jge jobb_oldal_utkozes
        cmp.w #0,labda_y
        jl bal_oldal_utkozes
;bal uto
        cmp.w labda_x,bal_x
```

```
        jeq  bal_ellenoriz01
        mov  bal_x,R12
        inc  R12
        cmp.w labda_x,R12

        jeq  bal_ellenoriz01
 jobb:
        cmp.w labda_x,jobb_x
        jeq  jobb_ellenoriz01

        mov jobb_x,R12
        inc  R12

        cmp.w labda_x,R12
        jeq  jobb_ellenoriz01

labda_rajz:

        mov labda_x,R13
        mov labda_y,R12
        mov #0x30,R14
        call #LCDChrXY

        //pontok kiirasa
        mov.w #1,R12
        mov.w #0,R13
        mov.w bal_pont,R14
        add.w #0x30,R14
        call #LCDChrXY

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


        call #LCDUpdate
        jmp main

alul_felul_utkozes:
        mov.w #0,R12
        sub.w  labda_dx,R12
        mov.w R12,labda_dx
        jmp labda_rajz

bal_oldal_utkozes:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
```

```
        inc jobb_pont
        mov.w #2,labda_x
        mov.w #7,labda_y
        jmp labda_rajz

jobb_oldal_utkozes:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
        inc bal_pont
        mov.w #2,labda_x
        mov.w #7,labda_y
        jmp labda_rajz

bal_ellenoriz01:
        cmp.w #0,labda_y
        jeq bal_ellenoriz02
        jmp jobb

bal_ellenoriz02:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
        jmp labda_rajz

bal_ellenoriz03:
        dec labda_x
        cmp.w #0,labda_y
        jeq bal_ellenoriz04
        ;jmp jobb2

bal_ellenoriz04:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
        jmp labda_rajz

jobb_ellenoriz01:
        cmp.w #12,labda_y
        jeq jobb_ellenoriz02
        jmp labda_rajz

jobb_ellenoriz02:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
        jmp labda_rajz

jobb_ellenoriz03:
        ;dec labda_x
```

```
        cmp.w #12,labda_y
        jeq jobb_ellenoriz04
        jmp labda_rajz

jobb_ellenoriz04:
        mov.w #0,R12
        sub.w  labda_dy,R12
        mov.w R12,labda_dy
        jmp labda_rajz


        ret
```

4.6. Megjegyzések

A feladat bonyolultsága miatt a mérést több részletben sikerült elvégezni. A feladatot végül Laczkó Hunorral közösen fejeztük be, ezért az ő jegyzőkönyve és ez a jegyzőkönyv (helyenként) ugyanazt az assembly kódot tartalmazza.