

LIST P=12F508

**;Dobókocka program PIC12F508 mikrovezérlőre,**

#include <pl2F508.inc>

**Készítette: Skori 2009 dec.**

**\_config \_MCLR\_ON & \_CP\_OFF & \_WDT\_OFF & \_IntRC\_OSC**  
**; MCLR láb resetel, kódvédelem ki, watchdog ki, belső RC oszcillátor**

```
#define egyes      b'00010000' ; középső 1-es pont - GP4
#define kettes     b'00000001' ; kettes átló - GP0
#define kettes_    b'00000100' ; kettes másik átló - GP2
#define kettes_k   b'00000010' ; közp kettes vonal - GP1
#define harmas     (egyed | kettes)
#define harmas_    (egyed | kettes_)
#define negyes     (kettes | kettes_)
#define otos       (egyed | kettes | kettes_)
#define hatos      (kettes | kettes_ | kettes_k)
#define hetes      (kettes | kettes_ | kettes_k | egyes)
#define hangsz     b'00100000' ;GP5-re van rákötve
#define hangsz_    d'5' ;GP5-re van rákötve
; kettest és hármast kétféle képpen lehet kijelezni (két átló van)
```

```
cblock 0x07 ; 7h-tól kezdődik a szabad memória, itt lesznek a program változói
ck ;ált. cikluszámológó
delay1 ;delay ciklusához
delay2 ;delay ciklusához
delay3 ;delay ciklusához
szam_0 ;változó a véletlenszám számolásához
kikapcsolo ;ha ki kell kapcsolni a következő reset-nél akkor =1
on_count ;bekapcsolások száma (ha később kellene valamire)
frekvencia ;periódusidő ciklusváltozó hanghoz (most már ez marad a neve)
timer ;a timerből elsőre kiolvasott számot elteszem ebbe (ez lesz a véletlenszám alapja)
endc
```

```
org 0
start:
;movwf OSCCAL ;oszcillátor kalibrálás gyári kalibrációs érték induláskor a W regisz
;
movlw b'11001000' ;
tris GPIO ; I/O portok beállítása (1:bemenet 0: kimenet)
clrf GPIO ; minden kimenet 0-ra
;
movlw b'11010000' ;7- Ébrvesztés bemenetről tiltva:1 eng:0
;6- felhúzó ell. GP0 GP1 GP3 tiltva:1 eng:0
;5- TIMERO órajel belső:0 külső:1 (GP2 lábön)
;4- TIMERO léptetőél felfutó:0 lefutó:1
;3- előosztó hozzárendelés: szam_0:0 watchdog:1
;2,1,0 Előosztó (osztás 2 ennyiedik hatványával) WDT-nél, ennek a duplá
option ;option reg beállítása
;
btfs STATUS,3 ;a harmadik bit 0?
goto poweronreset ;ha power-on volt akkor elugrik
;
btfs kikapcsolo,0 ;kikapcsolo alsó bit 0?
goto kikapcs ;ha a reset hamar jött az előző után akkor kikapcsra ugrik
;
call delay100ms ;késleltetés a reset nyomógomb esetleges pergése miatt
bsf kikapcsolo,0 ;ha a következő pár sorban jön újabb reset akkor kikapcs változó=1.
;
movfw TMRO ;TIMER olvasása véletlenszám generáláshoz
movwf timer ;az elsőnek olvasott valódi véletlenszám tárolása
```

```

call villogtat_1 ;ledek villogtatása, a játék kedvéért :) + késleltetés + hang
clrf kikapcsolo ;innenről nem kikapcs, hanem új számot generál a reset. kikapcs változ
call villogtat_1 ;ledek villogtatása, a játék kedvéért :) + késleltetés + hang

movwf timer ;előkészítés a véletlenszám osztásához
call osztas ;osztási maradék szam_0 változóban (mert nem 0...255 közöttit kell ki
movwf szam_0 ;w-ban (0...11)
call szamok ;konvertálás (egyes...hatos)
movwf GPIO ;a "tárolt érték" kijelzése (ledek állapotának beállítása)

movlw d'6' ;várakozás
movwf ck ;ennyiszer fut le a ciklus
call delay20s ;késleltetés
decfsz ck ;ck csökken, egészen 0-ig
goto $-2 ;visszaugrás (ha még nem 0)

goto kikapcs ;hosszú idő eltelt - kikapcsolás
;*****
beep_1: ;ezzel állítom elő a hangot

movlw d'200' ;200 periódus lesz
movwf delay1

bpl:
movlw d'80' ;ettől függ a periódusidő (belső ciklus)
addwf szam_0,w ;meg persze a számtól is hogy ne csak egyféle hangmagasságú sipolás le
movwf frekvencia ;eltesszük a változóba az aktuális értéket

swapf delay1,w ;kiszámolunk egy másik lehetséges frekit is
andlw d'15' ;ami nem a számtól, hanem csak a külső ciklustól függ
iorlw d'80'

;btfsz on_count,0
btfsz timer,6 ;ez dönti el, hogy a két kiszámolt frekvencia közül melyik szóljon
movwf frekvencia

decfsz frekvencia ;periódusidő várakozó ciklusa
goto $-1
movlw hangsz ;hangszóró bitje a GPIO porton
xorwf GPIO ;hangszóró bit invertálása
decfsz delay1 ;0-ig fogunk számlálni
goto bpl ;vissza a ciklus elejére

bcf GPIO,hangsz_ ;a végén hangszóró kikapcs (biztos ami biztos :)
retlw 0 ;visszatérés a függvényből.
;*****
villogtat_1:
movlw d'11' ;ez egy ennyiszer lefutó ciklushoz lesz
movwf ck ;szám a cilusváltozóba

jmp1:
;"véletlenszám" generálás (ez csak a villogás miatt kell, nem valódi, csak nagyjából-véletlen soro
call sorsol ;szám: 0...11 a szam_0 változóban
movwf szam_0 ;a "dobott érték" kijelzése
call szamok ;konvertálás a ledek portra kötése szerintire
movwf GPIO ;ledek bakapcsolása

;call delay100ms ;várakozás - ez volt korábban
call beep_1 ;de most hangot is adunk ki

decfsz ck ;ismétlés amíg a ciklusváltozó 0-ig csökken
goto jmp1 ;vissza a ciklus elejére

retlw 0 ;visszatérés a függvényből.
;*****

```

```

poweronreset:                                ;ide ugrik a főprogram, ha bekapcsolási reset volt (pl. elemcsere)
        clrf      on_count                    ;bekapcsolásszámláló nullázás
        movlw    hetes                       ;az összes led
        movwf    GPIO                        ;felvillantjuk a ledeket

kikapcs:
        call     delay100ms                  ;kis késleltetés prellezés és hasonlók miatt
        call     beep_1                      ;hangszóró csippanás (+még egy kis késleltetés)
        incf     on_count                    ;bekapcsolás számlálása (valójában kikapcsolást számol :))
        clrf     kikapcsolo                  ;kikapcsoló nullázása (hiszen mindjárt kikapcsolunk)
        clrf     GPIO                        ;kimenetek nullázása (ledék és hangszóró kikapcsolása)
        sleep   ;alvó üzemmód (kikapcsolás) a következő reset-ig
;*****

sorsol:
        ;véletlenszám generálás
        movfw    TMRO                        ;beépített timer értékét olvassuk véletlenszámnak

osztas:
        movwf    szam_0
        movlw    d'12'                      ;12-vel fogunk osztani
        bcf     STATUS,C                    ;C-bit nullázása (biztos ami biztos)
        subwf    szam_0,w                   ;levonjuk az osztót
        BTFSC   STATUS,C                    ;annyiszor amíg az osztandó számunk el nem fogy
        goto    osztas                     ;vissza a ciklus elejére
        ;osztási maradék a szam_0 változóban
        retlw   0
;*****

szamok:
        ;a W regiszter értékétől függő utasításra ugrik és azzal az értékkel tér v
        incf     W                           ;mivel W=(0...11) -> 1...12
        addwf   PCL,f                        ;programszámláló növelése (ettől fog urani)
        retlw   egyes                        ;innenről visszatérés az ugrás szerint
        retlw   kettes                       ;ezek valósítják meg a (szám) - (ledék állapota) konverziót
        retlw   harmas
        retlw   negyes
        retlw   otos
        retlw   hatos
        retlw   egyes_
        retlw   kettes_
        retlw   harmas_
        retlw   negyes_
        retlw   otos_
        retlw   hatos

;*****

delay20s:
        ;ez a késleltető rutin semmit nem csinál csak futnak a ciklusok és telik az idő ve
        clrf    delay3
        goto    d2

delay100ms:
        movlw   1
        movwf   delay3

d2: movlw    d'70'
    movwf    delay2

d1: clrf    delay1
    decfsz  delay1,f
    goto    $-1

        decfsz  delay2,f
        goto    d1
        decfsz  delay3,f
        goto    d2

        retlw   0
;*****
end

```