

ÁSZÁMBÚÍ,

avagy gyorsalpaló a káttesért



ATmega128-as processzorhoz

Mérési laboratórium 2.

Készítette: Döme

Észrevétel, hibajelentés: demeter.333@gmail.com

2013. május 19.

Ellenőrző méréshez írtam, így feltételezem, hogy már mindenki találkozott assembly kóddal, így csak azt részletezem, hogy mire kell figyelni, gyakori hibák stb.

Ez a dokumentum nem helyettesíti a tanszék adta „ATmega128 assembly segédlet” című dokumentumot és nem mentesít ismerete alól.

INTERRUPT

Kétféle interruptot kell tudni írni fejből: gombnyomásos és időzítő.

Felmerülhet a kérdés, hogy ezt most magoljam be? Majd a gyors válasz: magolja be, akinek két anyja van. Ha megérted mi történik, akkor nagy valószínűleg könnyebben fogsz írni egyet.

Az interrupt „táblázat” helyén egy reti nop féle szörnyűséget fogsz találni. ott egy reti nop alkot egy egységet, tehát a megszakításhoz ki kell törölni egy retit és egy nopot.

Gomb megszakítás, mintakód:

```
/* Reset & Interrupt Vectors
.cseg
.org $0000 ; Define start of Code segment
    jmp RESET
    reti
    nop
    reti
    nop
    reti
    nop
    reti
    nop
    reti
    nop
    jmp BTNO
    reti
    nop
    reti
```

Ez után következnek az inicializálások. Az I meghajtón már ott figyel az atmega128.pdf nevű fájlod, mivel senki nem várhatja el tőled, hogy fejből tudd a bitkiosztást, majd az 1000redik programodnál menni fog, ha eljutsz odáig 😊

```

;***** BTNO inicializálás*****
ldi temp, 0 ; beállítjuk az E portot bemenetként (nyomógomb)
out DDRE, temp
ldi temp, 0b00001000 ; megszakítási esemény kiválasztása
; . . . . 10 . . ; ISC5=10 (INT5 lefutó él)
;ldi temp, (1<<ISC51) ; előző bitminta másképp
out EICRB, temp
ldi temp, 0b00100000 ; az INT5-höz tartozó megszakítást engedélyezzük
; . . 1 . . . . . ; INT5=1 - megszakítás engedélyezve
;ldi temp, (1<<INT5) ; előző bitminta másképp
out EIMSK, temp
sei ; globális IT engedélyezése

```

Itt mondd meg, hogy te szeretnéd, ha figyelne a külső megszakításokat is...de a processzor nem gondolatolvasó: lövése sincs, hogy te mire gondolsz. Ezért megmondod neki, hogy te processzor kérlek nézd már...nem is tudom...mondjuk az E portot, ott mondjuk legyen lefutó élén a nekem kellő gomb(EICRB) így, processzor tudni fogja, hogyha annak a gombnak hirtelen van egy lefutó éle, akkor engedélyezni kéne vmit...de mit? legjobb lesz, ha iszom a vizeletemből és megkérdezem EIMSK-t. Aztán EIMSK megmondta, hogy melyik gomb megszakításáról van szó. Majd írtam egy sei-t, amivel közöltem processzornak, hogy ennyit gondoltam az interruptokról.

Majd a megfelelő címke alá megírjuk a megszakításunkat:

```

;*****
;IT kezelés
;*****
BTNO:
    push temp ; elmentjük temp aktuális értékét
    in temp, SREG ; elmentük SREG aktuális értékét
    push temp

;ide írok mindenféle dolgokat

    pop temp
    out SREG, temp
    pop temp ; visszatöltjük SREG-et és temp-et a veremből
reti

```

Ezt a keretet akkor kell használnunk, ha valami olyasmit írunk az interruptba, ami a temp és/vagy SREG-et bántja. Azért van rá szükségünk, hogy akárhol is járunk a főprogramunkban, ne befolyásolja a számításokat az interrupt. Legjobb lesz, ha mindig oda írod, főleg ha nem tudod eldönteni. (Mellesleg a push, pop a veremkezelés.) A reti kell minden interrupt szubrutin után, ezzel térünk vissza.

Hardveres időzítő, mintakód:

inicializálás:

```
;***** Timer 0 inicializálása *****
ldi    temp,0b00001111
      ;      0.....      ; FOC=0
      ;      .0..1...     ; WGM=10 (CTC mod)
      ;      ..00....     ; COM=00 (kimenet tiltva)
      ;      .....111    ; CS0=111 (CLK/1024)
      out    TCCR0,temp    ; Timer 0 TCCR0 regiszter
      ldi    temp,108      ; 11059200Hz/1024 = 108*100
      out    OCR0,temp     ; Timer 0 OCR0 regiszter
      ldi    temp,0b00000010
      ;      000000..     ; Timer2,1 IT tiltva
      ;      .....1.     ; OCIE0=1
      ;      .....0      ; TOIE0=0
      out    TIMSK,temp    ; Timer IT Mask regiszter
      sei    ; globális IT engedélyezve
```

A hardveres időzítő inicializálásának lényege, hogy fogalma sincs, hogy mennyi idő 1 sec, arról meg végképp, hogy te mennyi időnként akarsz az ő interuptját meghívni. **TCCR0**: több időzítő van benne, ezzel megnevezed, hogy te 0.-ról beszélsz. Itt állíthatod be neki az utolsó 3 biten, hogy hány részre szedje szét a processzor órajelét, jelen esetben 1024 az jó nekünk, megtanuljuk, örülünk. A negyedik bit ha jól tudom az időzítő komparálását engedélyezi.

OCR0: megint csak azért kell a 0 a végére, hogy tudjuk melyik időzítőről szólunk. Ez megint egy komparálás, a lényeg az, hogy még tovább osztjuk a processzor időt, így még pontosabban megmondjuk, hogy mikor adjon megszakítást.

TIMSK: ha már mindent beállítottunk, akkor engedélyezzük is az időzítő megszakítását, mégpedig a TIMERO-nak és a megfelelő módon.

Seiről megint ne feledkezzünk meg.

Az időzítő megszakításnál se feledkezzünk meg a temp, sreg vermes gondolatról sem, meg a visszatérésről.

Fényemittáló Dióda (LED)

Gyors inicializálás, mert nem triviális, hogy kimenet.

```
ldi temp, 0xff
out DDRC, temp ; C port kimenet, ledek meghajtása
```

Használata:

```
out PORTC, ledek
```

Kapcsolók

A kapcsolók „különleges” porton vannak, így másképp kell őket kezelni, mert ők a memóriában vannak.

Gyors inicializálás:

```
ldi    temp, 0x00          ; bemeneti irányt szeretnék
sts    DDRG, temp         ; irány beállítás
lds    temp, PING         ; a PING-ből a kapcsolók temp-be
```

Használata:

```
lds sw, PING              ;mivel nincsenek sorban a kapcsolók, így rendezem őket
bst sw, 4                 ;T tárolóba teszem az sw2-t
cbr sw, 0b11110100       ;kitörlöm a nem kellő biteket
bld sw, 2                 ;T tárolóból visszatöltöm a megfelelő helyre az sw2-t
```

Most már sorba vannak rendezve.

Pegésmentesítés

Jogos kérdés, hogy mi az, hogy egy gomb pereg? És másik jogos, hogy miért foglalkozzam vele. Ez a nyomógomb is azon elv alapján működik megközelítőleg, mint egy átlagos nyomógombos golyóstoll: ha megnyomjuk a gombot, akkor a rugó oda-vissza mozog és ez olyan, mintha többször nyomtuk volna meg. Ez fájhat nekünk gombnyomás számláláskor, ezért vagyunk rásérülve a pegésmentesítésre, főleg ha kéri a feladat.

Hardveres pegésmentesítés, példakód:

ilyenkor kihasználjuk a processzor adta időzítőnk, jupijé.

timer_IT:

```
push temp
in temp, SREG
push temp

in temp, PINE                // beolvassuk az E portot
bst temp, 5                  // BTN0-t berakjuk a T-be

lsl pattern                  // shiteljuk a mintatarat
bld pattern, 0               // LSB-be berakjuk a gomb aktualis állapotát

andi pattern, 0x0F           // csak az also 4 bit kell nekünk
cpi pattern, 0b00001100 // 1-1-0-0 vizsgalat, ezzel edge detektelas is tortenik
brne not_pressed

com isBtnPushed              // gomblenyomas volt!
```

not_pressed:

```
pop temp
out SREG, temp
pop temp

reti
```

Szoftveres pergésmentesítés, példakód:

```
M_LOOP:
    in btn, PINE
    andi btn, 0b00010000           ;csekkoljuk a gombok állását, nem megfelelően
    breq GOMB                     ;arrol, hogy a gombok alacsony aktívak
    jmp M_LOOP

GOMB:
wait4ms:
    ldi temp, 0xAB                 ;171 * 255 + 2 * 255 + 2 * 39 = 44193
loop1:
    ldi temp2, 0xff               ;11059200 / 44193 = 250
loop2:
    dec temp2                     ;0,004 s - 250 Hz
    brne loop2
    dec temp
    brne loop1
    ret

wait20ms:
    ldi temp3, 25                 ;25-ször meghívjuk a 4 ms-os késleltetést,
loop3:                             ;így 100 ms-ot fogunk várni
    rcall wait4ms
    dec temp3
    brne loop3

    in btn, PINE
    andi btn, 0b00010000
    breq volt_lenyomva

volt_lenyomva:
    ldi gomb_nyomas, 0xff        ;ez lehet vminek engedélyező jele, mely már
    ret                           ;egy fix pergésmentesített jel
    jmp M_LOOP
```

Gyakori hibák

- A kód lefordul, de nincs interrupt.

Ellenőrizd, hogy jó helyen van-e a sei, megfelelően inicializáltad-e, könnyű elérni a biteket. Jó helyre írtad-e az interrupt táblázatban. Ezek okozhatnak olyan problémát, hogy más gombra működik az interrupt stb.

- A kód lefordul, de össze-vissza ugrál a programban.

Valószínűleg egy súlyos elvi hibát vétettél, debugolás közben nézd a Stack Pointert, ha túlcsozdul, akkor keresd meg az okát és ne szólj labveznek, mert akkor vége a dalnak. Gyakori hiba ilyenkor, hogy egy call-ból jmp-olsz, ezt old fel. (mellesleg, call-ból mindenképp ret-elj vissza, ez is lehet gond)

Jó tanács: az idő szűkében ne félj kérdezni a laborvezetőktől, ha tényleg megakadtál és mindenféle ihleted elszállt, ők nem azért vannak, hogy megbuktassanak.

Remélem segítettem egy-két embernek helyére tenni dolgokat. Sok sikert kívánok.