

5.MÉRÉS

Időzítés és megszakítás programozása a mikrokontroller-bázisú mikrogépben

1. A mérés célja:

A 8051 mikrokontroller család minden elemébe beépített T0, T1 jelű programozható számlálók használatának, és a megszakítások kezelésének megismerése. A monitorprogram belső - terminál-kezelő programjainak használata.

2. A szükséges ismeretek:

A mérésben használt egységek (időzítő/számláló, megszakítás-vezérlő) belső fel-építéséről, működéséről, üzemmódjaikról, és programozásukról az ismereteket az előadás és az oktatási segédlet tartalmazza.

2.1. A programozható időzítő/számláló

A mérésekben használt gyakorlóban a **T1-es** időzítő/számláló a soros adatátvitel **Baud-rate generátora**. Ezért csak a **T0-ás** időzítő/számláló egységet szabad alkalmazni a **mérési** feladatokban.

Mivel az üzemmód beállítása mindkét egységnél a **TMOD** SFR regiszterben történik, és a regiszter csak bájtosan írható/olvasható, ezért ügyelni kell arra, hogy ne írjuk át a T1 üzemmód beállítását, vagyis nem szabad a regiszter felső négy bitjét megváltoztatni. A TMOD módosítása csak maszkolással végezhető az alábbi példa szerint:

Az új üzemmód állandójának deklarálása:

UJMOD EQU 0000XXXXB ; ahol XXXX az új üzemmód
;GATE, C/T, M1, M0 bitjeinek új értékei.

Az üzemmód beállítása az inicializálásban:

ANL TMOD,#0F0H ; a felső 4 bit maszkolása
ORL TMOD,#UJMOD ; az alsó 4 bit beillesztése

Ugyancsak egy regiszterben - a **TCON**-ban - vannak a **T0**, és a **T1** vezérlő-, valamint státusz bitjei (részletes kiosztást lásd az oktatási segédletben). Miután ez az SFR bit címezhető, ezért ezek a bitek a "**SETB bit**", és a "**CLEAR bit**" utasításokkal egyenként írhatók 1-be, vagy törölhetők.

2.2. A megszakítások és használatuk

A mikrokontroller család megszakításai fix címűek, ami azt jelenti, hogy minden megszakítás-forráshoz a code-memória első lapján - a 0003H-tól 8-as címtávolságonként - egy-egy cím (vektor-helyek) tartozik. Egy adott megszakítás elfogadása után a kiszolgáló rutint a CPU az adott megszakításhoz tartozó címen hívja.

Minden megszakításforráshoz tartozik egy megszakítás-kérő bit. Így például a **T0** időzítő/számláló túlszordulása a **TF0** bitet (TCON.4) írja 1-be, és ez kezdeményezhet megszakítást.

A megszakítás elfogadásának egyik feltétele, hogy a megfelelő engedélyező (maszk) bit 1 értékű legyen. A megszakításokat engedélyező bitek az **IEN0**, (interrupt enable) SFR-ben van. A regiszter bitei közül az **EA** (IEN0.7) az **általános engedélyező** bit, míg a többi - az egyes megszakításforrásokhoz tartozó - ún. **maszk** bitek, pl. a T0-hoz az **ET0** (IEN0.1) tartozik.

Megszakítás akkor jön létre, ha a forrás megszakítást kér és mind az EA és a megfelelő maszk bit is 1-ben van. Pl. A T0-al megvalósított megszakításos időzítésnél az ET0 és EA biteket 1-be kell írni, és a számláló minden túlsordulásakor lefut a megszakítás-rutin. **Fontos:** a megszakítás engedélyezés csak a inicializáló programrész végén történhet!

Egy megszakítás használatakor a megfelelő vektor helyre egy ugró utasítást kell írni, amely a megfelelő kiszolgáló-rutinra adja át a vezérlést.

A 8051/31 típusú mikrokontroller megszakítás forrásai és kezdő címei:

Megszakítás-forrás	Kezdő-cím
0-ás külső megsz.	0003 H
Timer 0 megsz.	000B H
1-es külső megsz.	0013 H
Timer 1 megsz.	001B H
Soros vonal megsz.	0023 H

Az alábbi példa a T0 megszakításos használatát szemlélteti:

```

KEZD: JMP INIT           ; ugrás az inicializálásra
      DS 8                ; annyi hely foglalása, amennyi biztosítja hogy a
                           ; megfelelő címre kerüljön az ugró utasítás
      JMP MTO             ; T0 vektora
      DS 3                ; tetszőleges számú bájt lefoglalása

; xxxxxx Inicializálás xxxxx

INIT: MOV SP,STACK-1      ; Stack-pointer beállítása

      .                    ; inicializáló programok, pl. memória törlés,
      .                    ; T0 üzemmód beállítása, számláló indítása stb.

      SETB ET0            ; megszakítás engedélyezések
      SETB EA             ; Inicializálás vége

; xxxxxx Program xxxxxx

FOPR:
      .                    ; a feladatot megvalósító főprogram
      .

      JMP FOPR
; xxxxxx Szubrutinok xxxxxx

      ; A megszakítást kiszolgáló rutin általános felépítése a következő:

; xxxxxx A T0 késleltető rutin xxxxx

MT0:  PUSH ACC           ; regiszter mentések
      PUSH PSW

```

POP PSW ;kiszolgáló programblokk
 POP ACC ;regiszter visszatöltések
 RETI ; megszakítás vége

Megjegyzés: ha rutinban egyéb regisztereket is használunk, akkor azok mentéséről, visszatöltéséről is gondoskodni kell !

2.3. Megszakítási szintek

A mikrokontroller megszakításai között eleve van egy hardver prioritási sorrend. Ez az alábbi:

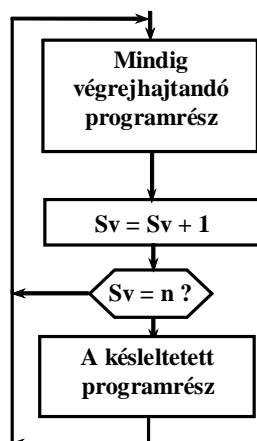
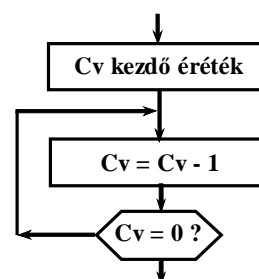
X0, T0, X1, T1, S0,

A sorrend lényege, hogy együttes megszakítás-kérés esetén a magasabb prioritású megszakítást szolgálja ki a kontroller. Ezen kívül mindegyik megszakítás egy alacsonyabb, vagy magasabb prioritási szintre is programozható. Az azonos szintre beállítottak között továbbra is érvényes a hardver sorrend. A 8051/31 típusú mikrokontroller 5 megszakítása az **IP0** prioritásvezérlő regiszterben állítható. (1 a magasabb, 0 az alacsonyabb szintre rendeli a megfelelő megszakításforrás kiszolgálását.) Hardver RESET hatására az összes megszakítás az alacsonyabb szintre kerül.

2.4. Késleltetési megoldások

A programokon belüli késleltetés megoldásai:

- **Szoftver-késleltetés** (2.mérés), amikor a program egy meghatározott futási-idejű ciklust hajt végre. A megoldás hátránya, hogy ameddig ki nem lép a ciklusból, addig a mikrogép egyéb eseményt - pl. bemeneti értékváltozást - nem észlelhet.

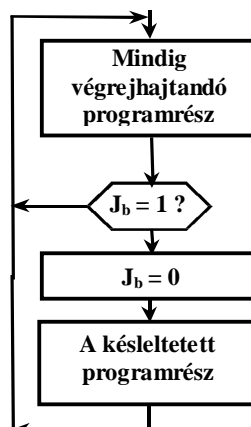


- A **főciklus számlálásával** a késleltetés azt jelenti, hogy a késleltetendő programrészt csak minden **n**-ik ciklusban hajtja végre a proceszor. A program az **Sv** változóban számlálja a ciklusokat. A késleltetés ideje:

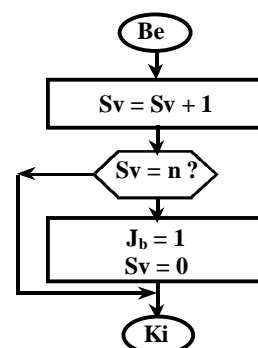
$$t_k = n \times t_c$$

ahol t_k a késleltetés, n a ciklusok száma és t_c a ciklusidő. A megoldásban már minden ciklusban értékelhetők a külső változások. Hátrány viszont, hogy a programciklus ideje t_c – a műveletektől függően változhat – így a késleltetés t_k sem lesz állandó.

- Pontos időalap-generátor megszakításainak számlálásával nagy-pontosságú késleltetés valósítható meg. A mikrokontrollerben található **T0**, vagy **T1** programozható időzítő/számláló nagy stabilitású időalap-generátor, amikor a quarc-al stabilizált órajelből osztott - **fo/12** frekvenciájú - jelet



Megszakítás rutin



számlálja. A megszakítást kiszolgáló rutinban az állandó időközönként fellépő megszakításokat kell számlálni (S_v számláló változóban). Az n -ik megszakítás-kiszolgáláskor egy jelzőbit 1-be állítása jelzi a késleltetési idő lejártát. A főprogramban csak akkor kell a késleltetendő programrészt végrehajtani, amikor a jelzőbit 1, majd a számlálót, és a bitet is törölni kell. Ekkor a késleltetés időtartama is állandó lesz.

3. Házi feladatok

- 3.1. Határozza meg, hogy mekkora számmal kell feltölteni a TH0 regisztert ahhoz, hogy a T0 időzítő reload (2-es üzemmódban) működtetve 0.1 msec-ként csorduljon túl. A kontroller 11,059 MHz-es órajelet kap.
- 3.2. Írjon megszakítás-rutint, amely a T0 időzítő megszakításait számlálja, és 0.5 sec elteltével egy időbitet beír.
- 3.3. Írjon programot amely villogtatja a gyakorló legkisebb helyiértékű LED-je 0,5 sec ismétlődéssel. A világos/sötét arány 75% legyen. A legkisebb helyiértékű nyomógomb lenyomott állapotában a kitöltés inverze szerint villogjon a LED.
- 3.4. Írjon programot amely méri a 0-ás nyomógomb működtetésének idejét 0,1 sec pontossággal. A nyomógomb felengedésekor a LED-soron jelenjen meg a mért érték.

4. Mérési feladatok:

- 4.1. Ellenőrizze a 3.3, 3.4 és 3.5 házi feladatokban megírt programok működését.
- 4.2. Módosítsa a MINTA1 programot úgy, hogy a szoftveridőzítés helyett a T0 időzítést (megszakítás segítségével) alkalmazzon.
 - illessze be a T0 megszakítás vektorát, és az inicializálását,
 - a házi feladatban elkészített megszakítás rutint szerkessze a programhoz
 - ellenőrizze a program futását.
- 4.3. Foglalja jegyzőkönyvbe a program folyamatábráját és a futtatás tapasztalatait.

5. Kérdések

- 5.1. Mit jelent a portok másodlagos funkciója?
- 5.2. Milyen üzemmódjai vannak a T0 és T1 időzítő/számláló áramköröknek?
- 5.3. Mit jelent az auto-reload működés?
- 5.4. Mi a feltétele a megszakításkérés elfogadásának?
- 5.5. Hogyan valósítható meg megszakításos időzítés?
- 5.6. Hogyan állítható be az időzítés pontos értéke?
- 5.7. Mit jelent a moduláris programozás? Melyek az előnyei?