

Hardver alapok 5-6. laborgyakorlat

A laborgyakorlat célja:

Egyszerű sorrendi feladatok megvalósítása szimulátoros környezetben, majd kipróbálása mérőpanelen. Megszakítás kezelés gyakorlati alkalmazása. Időzítés és IT kezelés szimulátoros ellenőrzése.

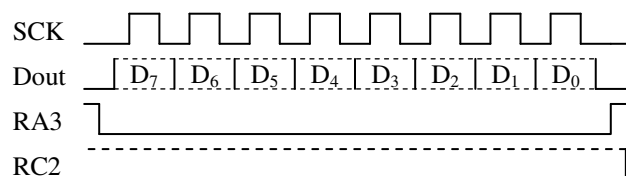
1. Szoftveres léptetőregiszter készítése

Feladat:

Írjunk assembly szubrutint (**SENDBYTE**), amely a W regiszterben kapott 8 bites értéket bitsorosan továbbítja a RB1-SCK, RB3-Dout portbiteken. A szubrutin minden adatbit után egy adott hosszúságú SCK órajel pulzust állítson elő. Az órajel „1” értéke alatt az adat kimenet értéke ne változzon. A szubrutin elsőként a legmagasabb helyi értékű bitet továbbítsa. Az időzítéshez használjuk az előző laborfoglalkozáson elkészített **Delay10us** szubrutint.

Ellenőrizzük a működést a szimulátor logikai analízátor funkciójával.

Egészítsük ki a feladatot úgy, hogy a léptetés elején az RA3 portbitet kapcsoljuk 0-ba, majd a küldés végén vissza 1-be. Végül kapcsoljuk az RC2 portbitet 1-be.



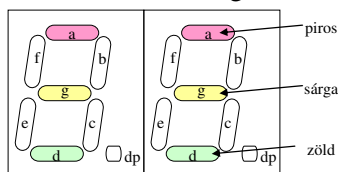
Segítség a megoldáshoz

Az előző foglalkozáson elkészített impulzus előállító programrészlet alapötletét felhasználva több pulzust is generálhatunk. Az előírt idődiagramnak megfelelően vezéreljük a portbiteket, ahol pedig időzítésre van szükség, ott hívjuk meg a megfelelő Delay szubrutint. A mérőpanelen található hétszegmens kijelző egység két egymás után láncolt léptető regisztert tartalmaz, melynek működését már a 3. laborfoglalkozás alkalmával megismertük. A most készített szubrutin arra szolgál, hogy ezeket a léptető regisztereket töltsse fel a W regiszterben megadott adattal.

2. Jelzőlámpa vezérlés

Feladat:

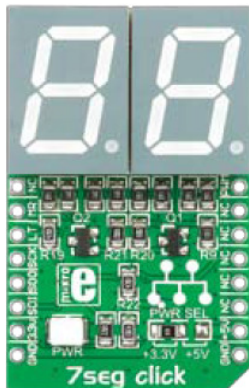
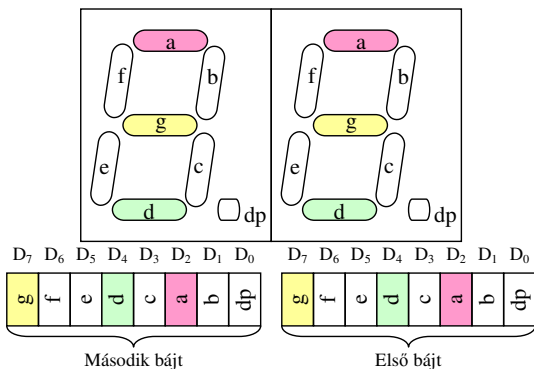
A 2. mérésen megismert feladat szoftveres megvalósítása. A kereszteződés két jelzőlámpáját a hétszegmens kijelzők középső szegmensei alkotják az alábbi ábrának megfelelően.



Az előző feladatban elkészített (**SENDBYTE**) szubrutin felhasználásával készítsünk olyan szubrutint (**SETLAMP**), amely a W regiszterben kapott 0..7 közötti szám alapján beállítja az eredeti feladatnak megfelelő kijelzést a panelen található hétszegmens kijelzőkön.

Segítség a megoldáshoz:

A kijelző vezérléséhez két bájtot kell kiküldeni egymást követően, melyeknek egyes bitjei az alábbi jelentésűek:



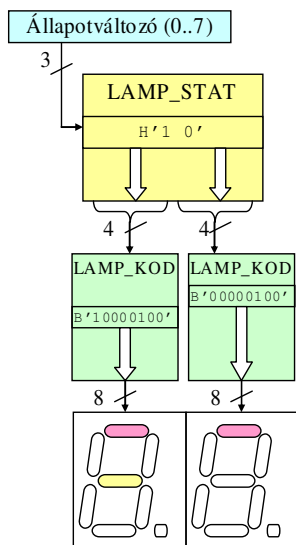
Szubrutin a kódok előállításához, W=0..3 állapot

LAMP_KOD:

```
ANDLW H'03'
BRW
RETLW B'00000100';P
RETLW B'10000100';PS
RETLW B'00010000';Z
RETLW B'10000000';S
```

A kijelzendő értékek előállítását egy állapotváltozó alapján végezzük két táblázat (**LAMP_STAT** és **LAMP_KOD**) segítségével a mellékelt táblázat szerint.

Állapot	Bal kijelző	Jobb kijelző	Táblázat érték
0	Piros (0)	Piros (0)	B'0000 0000'
1	Piros+Sárga (1)	Piros (0)	B'0001 0000'
2	Zöld (2)	Piros (0)	B'0010 0000'
3	Sárga (3)	Piros (0)	B'0011 0000'
4	Piros (0)	Piros (0)	B'0000 0000'
5	Piros (0)	Piros+Sárga (1)	B'0000 0001'
6	Piros (0)	Zöld (2)	B'0000 0010'
7	Piros (0)	Sárga (3)	B'0000 0011'



LAMP_STAT:

```
ANDLW H'07' ; alsó 3 bit meghagyása
BRW
RETLW H'00' ; P - P
RETLW H'10' ; PS - P
RETLW H'20' ; Z - P
RETLW H'30' ; S - P
RETLW H'00' ; P - P
RETLW H'01' ; P - PS
RETLW H'02' ; P - Z
RETLW H'03' ; P - S
```

Az állapotváltozó (**LSTATE**) értéke alapján a **LAMP_STAT** táblázat megadja, hogy melyik mintát kell a bal- és melyik mintát kell a jobb oldali kijelzőre tenni.

Használjuk a **LAMP_STAT** táblázat 8 bites értékeinek alsó 4 bitjét a jobb szélső kijelzőhöz; a felső 4 bitet pedig a bal szélső kijelző értékének megadásához.

A két minta sorszám alapján a **LAMP_KOD** szubrutin segít a megfelelő nyolcbites adat előállításában.

Ezeket a 8 bites értékeket kell a kijelzőn megjeleníteni.

A főciklusban csak a **STATE** változót kell egyesével növelni és a szükséges késleltetést biztosítani.

3. Hardveres időzítő kezelése

Feladatok:

- Készítsünk késleltető szubrutint, amely a TIMERO időzítő segítségével 10ms idejű késleltetést valósít meg.
- Alakítsuk át úgy a programot, hogy megszakítást okozzon a TIMER túlcsordulása.
- Ellenőrizzük az időzítést szimulátorban is. Villogtassuk a megszakítási szubrutinból az RA7 kimeneten lévő LED-et 1Hz frekvenciával.

Segítség a megoldáshoz

A mikrokontroller 32MHz órajelének negyede ($F_{osc}/4$) rendelkezésre áll, mint választható órajel forrás. A feladat tehát az, hogy milyen osztásértékeket állítsunk be az előosztón, a számlálón és az utóosztón ahhoz, hogy 8MHz-ből 100Hz legyen. A PIC segédlet blokkvázlatát és a választható értékeket figyelembe véve egy lehetséges megoldás:

$$100\text{Hz} = 8000000\text{Hz} \cdot \frac{1}{32} \cdot \frac{1}{250} \cdot \frac{1}{10}$$

Ennek értelmében az előosztót 1:32-re kell állítani, a TMR0 számlálót 8 bites üzemmódba kell kapcsolni, és 1:250-es osztásra állítani, majd az utóosztót 1:10-re kell választani.

Ehhez a következő regiszterekbe a következő értékeket kell írunk:

Regiszter név	Érték (binárisan)	Megjegyzés
T0CON0	B'1000 1001'	Bekapcsolás + 1:10 utóosztás
T0CON1	B'0100 0101'	$F_{osc}/4$ + 1:32 előosztás
TMR0H	B'1111 1001'	0...249 között számol

Ellenőrző kérdések

Az alábbi ellenőrző kérdések megválaszolásához nézze át az előadáson elhangzott anyagokat és tanulmányozza a mérési útmutatót.

- Adott az alábbi szubrutin. Mi lesz a W regiszter értéke a szubrutinból kilépéskor, ha a belépéskor a.) 3
b.) 12
c.) 24

decimális érték van a W regiszterben

LAMP_STAT:

```
ANDLW H'07' ; alsó 3 bit meghagyása  
BRW  
RETLW H'00' ; P - P  
RETLW H'10' ; PS - P  
RETLW H'20' ; Z - P  
RETLW H'30' ; S - P  
RETLW H'00' ; P - P  
RETLW H'01' ; P - PS  
RETLW H'02' ; P - Z  
RETLW H'03' ; P - S
```

- Adott az alábbi programrészlet. Adja meg a C és DC flagek értékét a műveletek végrehajtása után.
MOVLW D'13'
ADDLW D'03'
- Adja meg a W regiszter értékét hexadecimálisan a fenti utasítás sorozat végrehajtása után.
- Adott az alábbi programrészlet. Milyen szélességű impulzus jelenik meg a portlábon, ha egy utasítás végrehajtási ideje 125ns?
BSF LATA, 5
NOP
NOP
BCF LATA, 5
- Rajzolja fel milyen jeleket használ az SPI interfész.
- Sorolja fel a TIMER0 főbb részegységeit (legalább 3).
- Hogyan lehet a mikrokotrollerben az összes megszakítást letiltani?
- Mi történik egy megszakítás érvényre jutásakor?
- Ha több IT forrás is van, hogyan lehet kideríteni az aktív forrást?
- Egy IT érvényre jutásakor a felsorolt regiszterek közül melyiket menti automatikusan a kontroller: **BSR, PCLATH, PORTA, STATUS, TMR0H, WREG**?
- Mi a különbség a **RETURN** és a **RETFIE** utasítások között?
- Mit állít vissza automatikusan a kontroller a **RETFIE** utasítás végrehajtásakor?