

ÓBUDAI EGYETEM
Kandó Kálmán Villamosmérnöki Kar

Távoktatás

Programozható irányítások I

Tanulási útmutató

és követelmények

Villamosmérnök hallgatók részére

Tantárgyfelelős:

Zalotay Péter
főiskolai docens

1. A tantárgy célja

A **hallgatók** ismereteket szerezzenek a digitális, programozható irányító berendezések rendszertechnikai felépítéséről, elméleti alapjairól, a hardver és szoftver tervezés módszereiről.

A feladatok, valamint a laboratóriumi gyakorlatokon **jártasságot** biztosítani a digitális hardver, mikroprocesszoros-mikrokontrolleres berendezések (mikrogépek, PLC-k) programozásában, illesztésében, élesztésben, és a hibakeresésben.

Az irányító-, és technológiai berendezések illesztési szoftvereinek, a programozási eljárások alkalmazásának, a feladatmegoldások tipikus algoritmusainak megismertetése, és **begyakoroltatása**.

2. A tantárgy témái

1. Ipari irányítási rendszerek

Az elektronikus vezérlések helye, szerepe az ipari gyártástechnológiák automatizálásának korszerű megoldásaiban;

- mikroprocesszoros célberendezések
- PLC-k

Vezérlőberendezések felépítése

- egyszerű strukturálatlan rendszer
- hierarchikus rendszer
- irányítási rendszerek csoportosítása
- Rendszerleírási módszerek;

A 8031/51 mikrokontroller család C nyelvű programozása, és alkalmazása irányítási feladatokat megoldó programokban.

2. A MODICON 340 és SIEMENS S7 330 PLC-k programozása, és alkalmazása

- rendszertechnikai felépítésük, működésük,
- diszkrét, és analóg illesztők,
- programozási nyelvek,
- alkalmazási lehetőségek,
- kiválasztási szempontok.

3. Kombinációs vezérlési feladatok programozása

A kombinációs feladatok leírási módszerei,

Egyszerűsítési eljárások,

Algoritmizálás,

- Bitműveletes (assembly, utasításlista),
- táblázatos megoldások (tömbök, táblázat-műveletek).

Kombinációs feladatok programozása

- mikroprocesszoros rendszerek, ill. PLC-k alkalmazásával

4. Sorrendi vezérlési feladatok programozása

Rendszertechnikai ismeretek, a sorrendi feladatok csoportosítása, leírási módszerek. Sorrendi feladatokat megvalósító programok tervezése,

- ütemtáblázat, kódolás, vezérlő függvények, kimeneti függvények felírása,
- a programszerkezet kialakítása,
- programozás függvénymegoldással, táblázat használatával.

Sorrendi feladatok programozása

- mikroprocesszoros rendszerek, ill. PLC-k alkalmazásával

3. Követelmények

A távoktatás rendszerének megfelelően, a hallgatók *önálló tanulással* kell, elsajátítsák a mellékelt CD-n rendelkezésükre bocsátott, illetve az *Automatika Intézet honlapjáról* (www.aut.bmf.hu) letölthető tananyagot.

Az önálló munkát segítik

- a *konzultációk*,
- a kidolgozott *mintapéldák* elemzése,
- a példák és
- a kötelezően beadandó *feladatok* önálló megoldása, valamint
- a *mérési* feladatok elvégzése.

a. Az órarend szerinti *konzultációkon* a megadott témakörök önálló feldolgozása során felmerülő kérdéseket tárgyaljuk meg.

b. A félév során beadandó három *önálló feladat* otthon kell megoldani, s azt a következő konferenciát megelőző héten kell beküldeni a tantárgyért felelős oktatónak. A feladatok elbírálása pontozással történik. (A maximális pontszám 10.)

c. A félév végén - az előírt tananyag teljes anyagát átfogó kérdésekből – *szóbeli vizsgát* kell tenni. A kérdések két témacsoportot fognak át.

- A hardver ismeretekből példa, és elméleti anyag.
- Programozási ismeretekből példa, és elméleti anyag.

Csak az a vizsga *értékelhető*, amelynél a vizsgázó mindkét témakörből *eredményesen vizsgázik*.

Vizsgára bocsájtás feltétele mindhárom feladat beadása és elfogadása..

d. A gyakorlati ismeretek megszerzését segítik az elvégzendő *mérési gyakorlatok*. (A mérési gyakorlatok helyét és rendjét az utolsó konferencián kapják meg.)

4. Útmutató a tantárgy tanulásához

A tananyag feldolgozását, megértését a konzultációkon megbeszélte témák és Zalotay Péter: *Programozható irányítások I* című elektronikus jegyzete alapján végezhetik.

A tananyag feldolgozását célszerű a konzultációk ütemezése szerint végezni. A folyamatos tanulás, megértés, begyakorlás részeit célszerű tovább bontani *heti ütemezésre*.

Az egyes témaköröket ismertető *írásos anyag* elolvasása után szükséges *példák* megoldása.

A félév során beadandó feladatok elkészítését mindenképpen előzze meg az elméleti-, és példaanyag megismerése.

A tanulási program részletezésében kiemeljük azokat az *alapfogalmakat, jelöléseket*, amelyek *ismerete* feltétlen szükséges mind az anyag megértéséhez mind pedig az eredményes vizsgázáshoz.

5. RÉSZLETES TANULÁSI PROGRAM

3. év 2. Trimeszter

1. Konzultáció:

1.1. Felkészülés a konferenciára:

A *Digitális technika*, valamint az *Automatika* tantárgyakban tanultak felelevenítése. Segítséget nyújt ebben a megadott jegyzet 1. és 2. fejezete.

1.2. A konzultáción megtárgyalandó témakörök:

Az automatizálás feladatainak rendszertechnikai szempont szerint ismételése.

A 8031/51 mikrokontroller családról, illetve a programozási alapokról tanultak felelevenítése.

A mikrokontroller család C nyelvű programozásának alapismeretei.

1.3. Tervezési Feladat:

Tervezze meg azt a négy bemenetű és egy kimenetű kombinációs hálózatot, amely az alábbi – egyszerűsített felírású - logikai függvényt valósítja meg.

$$F = \sum_{i=0}^4 (0,4,5,10,14,15)$$

$$X = (1,2,6,7)$$

(az X a közömbös bemeneti kombinációk indexe)

A bemeneti változók jelei és súlyozásuk: $J \div 2^0$; $K \div 2^1$; $L \div 2^2$; $M \div 2^3$

Feladatok:

- Írja fel a függvény mindkét **kanonikus** alakját.
- Határozza meg a **konjunktív** alakú legegyszerűbb függvényt Karnaugh-diagram használatával.
- Végezze el a lehetséges összevonásokat.
- Írja meg a megoldás assembly, valamint C nyelvű forrásnyelvi programját az XE251 fejlesztői környezetben! Végezze el a programok fordítását, szerkesztését, és letölthető alakú változatát! Szimulátorral ellenőrizze a program működésének helyességét!

- e. A bemeneti változókat a 0C000H címen elérhető nyomógombsor alsó négy bitjéről kell beolvasni. A függvény eredményét a 0C001H címen elérhető LED-sor MSB bitje jelezze! **Figyelem!** A készüléken mind a be-, mind, pedig a kimenetek **aktív szintje 0!**
- f. A feladat tervezését írásos dokumentumban kell beküldeni. A program forrásnyelvű, és letölthető alakjait adathordozón a konferenciára hozzák!

Beadási határidő: a 2 konzultáció előtti péntek 12 óra.

2. Konzultáció:

2.1. Felkészülés a konferenciára:

A C nyelvű programozás megismerése példák alapján. Oldjon meg minél többet a példákból.

Olvassa el a jegyzet 4.1. fejezetét, amely a PB80C552 típusú mikrokontrollerrel kialakított mikrogép leírása.

Mindezek után készítse el az 1. Tervezési feladatot.

2.2. A konzultáción megtárgyalandó témakörök:

A PB80C552 típusú mikrokontrollerrel kialakított mikrogép hardver felépítése.

A kontroller *A/D*, *Capture*, *Compare*, *PWM*, valamint az *I²C* busz periféria elemeinek használata, programozása.

2.3. Példák:

1. A mikrokontrolleres gyakorló 0C000H címén elérhető LED-során jelezze ki binárisan a P1 porthoz csatlakoztatott peremkeres-kódkapcsolón beállított értékben lévő 1-ek számát.

A példában a beolvasott bájtot nyolcszor, a Cy –n keresztül kell forgatni, és valamelyik tárolóban számolni, hogy hányszor volt a tulcsordulás 1. Az így kapott eredményt jelezzük ki a LED soron.

A megoldás beolvasás után egy for ciklussal, majd az eredmény kijelzésével oldható meg.

A feladat megoldó program forrásanyagát, mintaként assembly, és C nyelven is megadjuk! A szimulátoron ellenőrizze a működést!

Assembly forrás:

```

;xxxxxxx fordítási parancsok xxxxxxxx
NAME          FO_MOD

;##### DEKLARÁCIÓK #####

;***** Gyakorló port címei *****

NGS           XDATA          0C000H           ;nyomógomb-sor címe
LEDS          XDATA          0C001H           ;LED-sor címe

;***** Szegmensek deklarálása *****

PROG          SEGMENT        CODE             ;program
VALT          SEGMENT        DATA           ;változók
STACK        SEGMENT        DATA           ;stack memória

;***** ADAT szegmens változóknak *****

BEM:          RSEG           VALT             ;bemeneti memória
              DS             1
KIM:          DS             1                ;kimeneti memória

;xxxxxxx STACK kijelölés xxxxxxxx

RSEG          STACK          ;STACK kezdet
DS            10H           ;16 bájtos STACK

;##### PROGRAMOK #####

;xxxxxxx Program szegmens xxxxxx
RSEG          PROG

```

```

START:    LJMP    INIT                                ;ugrás az inicializálásra
          ;***** Megszakítás ugrótábla *****
          DS     8                                ;helyfoglalás a megszakításoknak

          ;***** Inicializálás *****
INIT:     MOV     SP,#STACK-1
          MOV     R1,#BEM                          ;memória törlés
          MOV     R7,#10H
INIT_1:   MOV     @R1,#0
          INC     R1
          DJNZ   R7,INIT_1

          ;***** Fő program *****
FOPR:     MOV     BEM,P1                            ; beolvasás
          MOV     KIM,#0                            ;uj ciklis előtt törlés

          ; For ciklus

MOV       R2,#8                                    ;kezdő értékek beállítása
          MOV     A,BEM
          CLR     C

EL:       RLC     A                                ;ciklustörzs
          JNC     TOV
          INC     KIM
TOV:      DJNZ   R2,EL

          MOV     DPTR,#LEDS                        ;kiiratás
          MOV     A,KIM
          CPL     A
          MOVX   @DPTR,A
          JMP    FOPR

          END

```

C nyelvű forrás:

```

#include <REG552.h> // a mikrokontroller adatai
#include <Absacc.h> //periféria macro-k
#define LEDS XBYTE[0xC001] //LED-ek

void main (void)
{
    unsigned char data bem,kim,n; //deklarációk

    do
    {
        bem=P1; //beolvasás

        for (n=0,CY=0,kim=0; n<8; n++) //megoldás
        {
            bem*=2;
            if(CY) ++kim;
        }

        LEDS=kim; //kiiratás

    }while;
}

```

2. A mikrokontrolleres gyakorló 0C001H címén elérhető nyomógomb-sor alsó három gombjához rendeljük a hárombites bináris szám helyi értékeit. A gombok

lenyomásainak lehetséges kombinációinál a megfelelő helyiértékű LED (0C000H) világítson.

3. A mikrokontrolleres gyakorló 0C000H címén elérhető LED-során világítson a legnagyobb helyiértékű LED az $F=D(B+/C)+ABC$ logikai függvény IGAZ értékénél. Az A,B,C,D bemeneti változókat a 0C001H címén elérhető nyomógomb-sor legkisebb helyiértékű nyomógombjain lehessen beadni
4. Írjon programot amely a amely a 0C001H címén elérhető nyomógomb-sor legkisebb helyiértékű gombjának lenyomásakor jobbra, míg a legnagyobb helyiértékű nyomógomb lenyomásakor pedig balra lépő - 2n modulusú gyűrűs számláló alapciklusa szerint - világító sort állít elő a 0C000H címén elérhető LED-soron. A léptetést ciklusszámlálással ~ 0,5 sec-ra állítsa be!
5. Írjon programot, amely a képernyőre írja ki a saját nevét, címét és foglalkozását. Az egyes szövegrészek az ENTER lenyomás után jelenjenek meg
6. Írjon programot, amely a 0C001H címén elérhető 0-ás helyi értékű nyomógomb működtetésének idejét 0.1 sec pontossággal méri. A nyomógomb felengedésekor a mért érték a 0C000H címén elérhető LED - soron binárisan jelenjen meg
7. Írjon programot, amely villogtatja a gyakorló legkisebb helyiértékű ledje 0,5sec-es ismétlődéssel. A világos/sötét arány 75%os legyen. A legkisebb helyiértékű nyomógomb lenyomott állapotában a kitöltés inverze szerint villogjon a led.

2.4. Tervezési feladat:

Tervezze meg az ábra szerinti kétfázisú négyszögjelet előállító programot!

Mindkét jel frekvenciája $f=100\text{Hz}$, és a fázistolás 90° legyen.

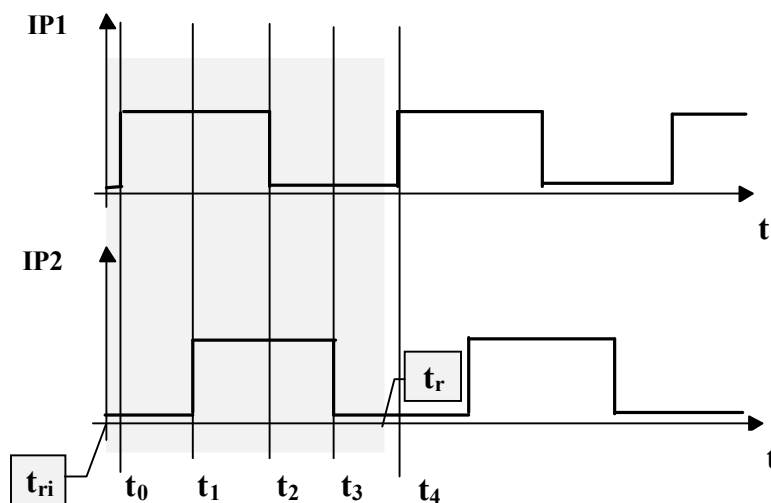
A feladatot a T2 számláló **compare** funkciójával oldja meg!

A jeleket a P4 Port következő bitjein jelenjenek meg

IP1 - P4.0

IP2 - P4.1

RT2 - P4.7



Feladatok:

- a. Határozza meg a váltások időpontjaihoz tartozó konstansokat!
- b. Tervezze meg a működtető program felépítését!
- c. Rajzolja meg a folyamatábrát!
- d. Írja meg a program c-nyelvű forrásállományát!
- e. Végezze el a program fordítását, szerkesztését!
- f. A feladat tervezését írásos dokumentumban kell beküldeni. A program forrásnyelvű, és letölthető alakjait adathordozón a konferenciára hozzák!

Beadási határidő: a 3. konzultáció előtti péntek 12 óra.

3. Konzultáció:

3.1. Felkészülés a konferenciára:

Az *irányítástechnikában* tanult *PLC* ismeretek, és a programozás átismétlése. Olvassák el a jegyzet 4.2. fejezetét. Készítse el a 2. Tervezési feladatot!

3.2. A konzultáción megtárgyalandó témakörök:

A PLC-k (programozható logikai vezérlők) *fejlődése*, *szerepük* az automatizálásban. Az *M340* (Schneider Electric), és az *S7-300* (Siemens) PLC családok jellemzői. Vezérlési feladatok programozása *LD*, és *FBD* programozási nyelveken. Logikai műveletek, időzítés, számlálás, és összehasonlítás megoldásai PLC programokban. Fejlesztői környezetek: a *Unity-XL*, és *Step 7* segítségével történő programozás megismerése.

3.3. Példák:

1. Az első Tervezési feladatban megadott logikai függvény programozása LD nyelven.

A függvény:

$$F = \sum (0,4,5,10,14,15) \quad X = (1,2,6,7)$$

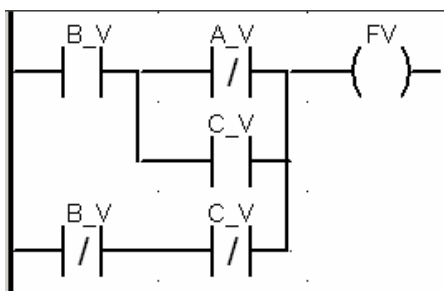
A logikai függvény legegyszerűbb alakja: $F = B(\bar{A} + C) + \bar{B}\bar{C}$

Első lépésként a *Változó-táblát* kell felvenni. Az itt megírt táblázat a legfontosabb *adatokat* tartalmazza, és *formailag* nem egyezik meg a használt fejlesztői környezetekben használtakkal.

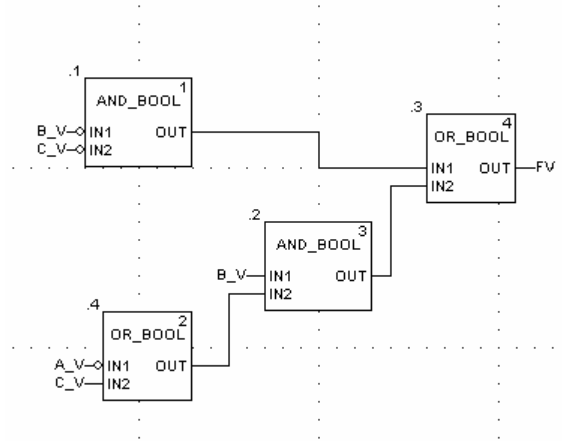
A Változó-tábla:

Szimbólum	típus	Cím	funkció
A_V	Boole	%I0.0	A változó
B_V	Boole	%I0.1	B változó
C_V	Boole	%I0.2	C változó
D_V	Boole	%I0.3	D változó
FV	Boole	%Q0.0	Függvényérték

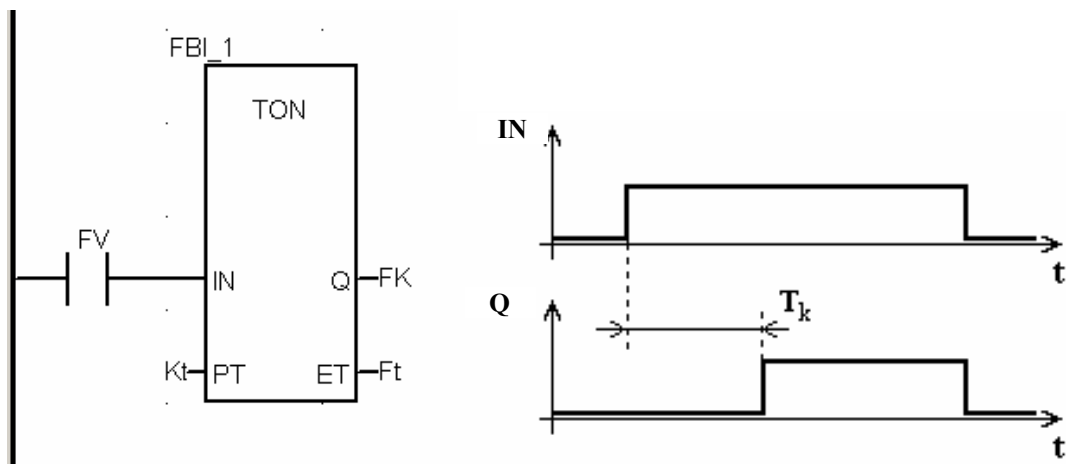
A függvényt megvalósító létradiagram:



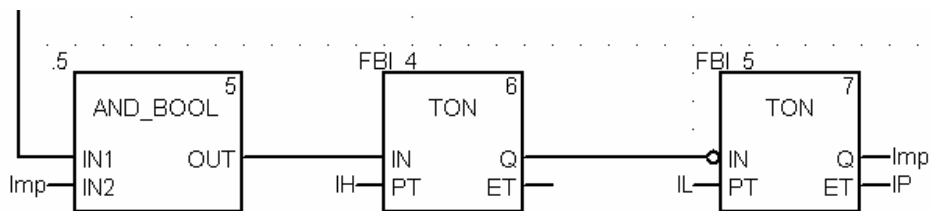
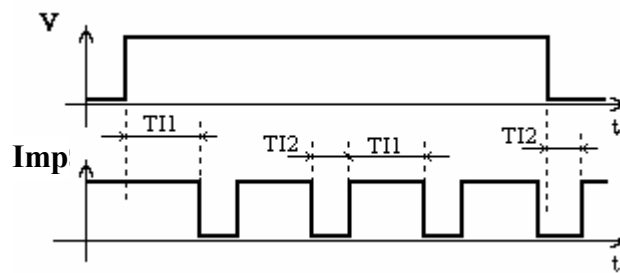
A függvény FBD nyelvű megvalósítása:



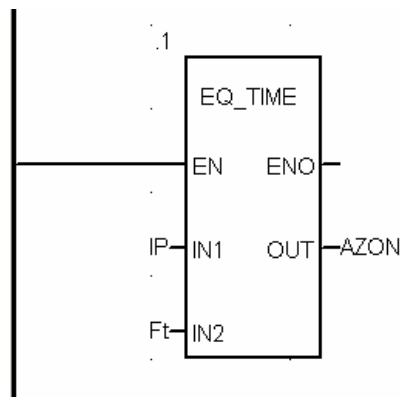
2. Bekapcsolási késleltetés (TON), és alkalmazása



Vezérelt impulzus-sorozat



3. Komparálás programozása

4. Készítse el a lépcsőházi **alternatív világításvezérlés** létradiagramos programját.

A lépcsőház **négyszintes**. A világítás váltson **ellenkező** állapotra - vagyis aludjon ki, ha világított, és fordítva - ha bármelyik szinten **működtetik** a világításvezérlő nyomógombot. A bekapcsolt lámpák **aludjanak** ki, ha 10 másodpercen belül nem vezérelnek - egyik szinten sem – állapotváltást.

5. Kapcsoló óra tervezése

A kapcsolás ideje **peremkeres** kódkapcsoló (kódkapcsoló hexadecimális értéket ad a PLC -nek 2x 4 biten!) segítségével **1 – 60** másodperc között legyen állítható. A beállított idő leteltét egy kimenet aktiválásával **jelezze**. A jelzés **nyugtázása** egy kapcsoló működtetésével történjen. Az órát egy **indító** kapcsolóval kell működésbe hozni. A jelzési idő csak a **nyugtázás**, és az új **indítás közötti** időtartam alatt lehessen állítani. Az aktuális időt két-karakteres **hétszegmensű** kijelzőn jelenítse meg. A kijelző bitkiosztása:

MSB	LSB	EN	X	d	c	b	a
-----	-----	----	---	---	---	---	---

ahol

- MSB és LSB a szegmensek kiválasztására
- EN az engedélyező bit
- X nem használjuk
- a,b,c,d a kiírandó – BCD kódolású - szám inverze.

6. Tervezze meg - az alábbi feladatokat ellátó – **ellenőrző program** létradiagramos programját.

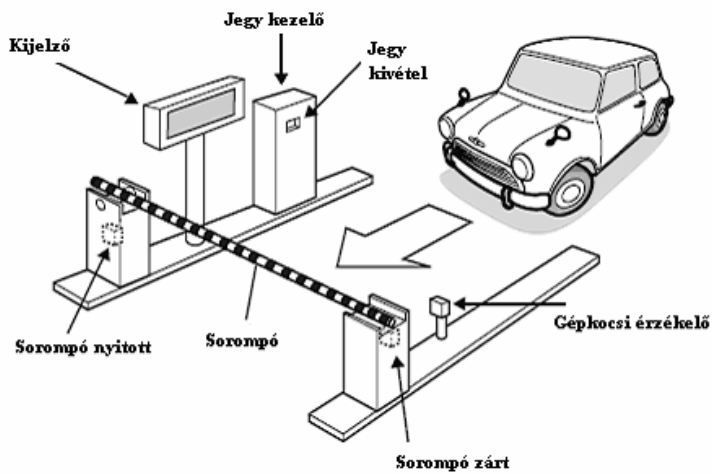
Az ellenőrzés célja három készülék – továbbiakban K1, K2, K3 - egyidejű működésének meghatározása, és számlálása.

A készülékek közül bármelyik kettő, vagy mindhárom legfeljebb $t_{\max} = 0,5$ másodpercig lehet egyidejűleg bekapcsolva. A t_{\max} értéket meghaladó együttes működés legfeljebb tízszer fordulhat elő. Ha többször érzékel a logika egyidejűséget, akkor le kell tiltani a működtetést. Az újbóli engedélyezés nyugtázással történhet.

3.4. Tervezési feladat:

Gépkocsi parkoló bejárati sorompó PLC-s vezérlő programjának megtervezése.

➤ *Sorompós gépkocsi behajtó működése:*



A gépkocsi beérkezését a **helyzetérzékelő** jelzi.

A **jegykiadó** nyílásában kerül az érvényesített jegy.

A kijelző oszlopon látható **három lámpa** jelzi a teendőket.

▪ *Az egység felépítése, és vezérlési feltételei:*

Alaphelyzete: a sorompó **zárt**, a kiadóban **nincs** jegy.

A **jegykiadóban** egy elektro-pneumatikus **szelep** akkor ad jegyet a nyílásba, ha a sorompó zárt, és gépkocsi érkezik a sorompóhoz. Ekkor a kijelző **piros** lámpája világít. A jegy kiadása 2 mp-ig tart, aminek befejezéskor egy **érintkező** zár. Utána a **sárga** lámpa villogása jelzi, hogy el lehet venni a jegyet

A **sorompót** forgásirányváltással vezérelt **aszinkronmotor** mozgatja. A nyitott, illetve a zárt állapotot egy-egy **záró** érintkező jelzi. A zárt sorompó nyitását akkor kell elkezdni, ha **van** gépkocsi a sorompó előtt, és **kivették** a jegyet az adagolóból. A sorompót teljes nyitásig kell mozgatni. A gépkocsi elhaladása után 3 mp-el a sorompó lecsukását kell indítani és a teljes bezárásig működtetni.

Feladatok:

- Határozza meg a vezérlési feladat logikai függvényeit!
- Határozza meg a szükséges változókat, és típusait!
- Készítse el a változó deklarációkat táblázatban!
- Tervezze meg a működtető program felépítését!
- Tervezze meg a programot LD nyelven!
- A feladat tervezését írásos dokumentumban kell beküldeni!

Beadási határidő: a 4. konzultáció előtti péntek 12 óra.

4. Konzultáció:

4.1. Felkészülés a konferenciára:

A megismert PLC típusok fejlesztői programjainak megismerése.
Olvassák el a jegyzet 5. fejezetét.
Készítse el a 3. Tervezési feladatot!

4.2. A konzultáción megtárgyalandó témakörök:

Kombinációs, és **sorrendi** feladatok programjainak tervezése

Kombinációs feladatok egyszerűsítése (Quin-Mc Closkey eljárás).

Sorrendi feladatok csoportosítása, leírási módszerek. Sorrendi feladatokat megvalósító programok tervezése,

- ütemtáblázat, kódolás, vezérlő függvények, kimeneti függvények felírása,
- a programszerkezet kialakítása,
- programozás függvénytáblázattal, táblázat használatával.

Sorrendi feladatok csoportosítása:

- lefutó sorrendű (egy szekvenciájú) vezérlés:
 - idővezérelt,
 - eseményvezérelt,
- általános sorrendű (több szekvenciájú) vezérlés.

Sorrendi feladatok programozása

- mikroprocesszoros rendszerek, ill. PLC-k alkalmazásával

4.3. Példák:

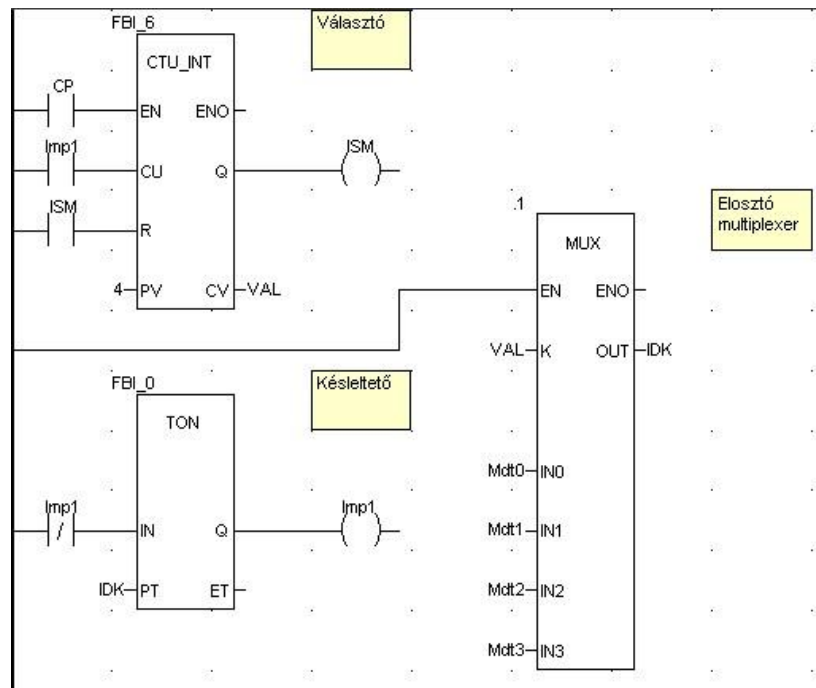
1. Ütemadó programozása

Az időfőggő lefutó típusú vezérléseknél az egyes ütemek időzítése változóak. Az ütemadó állítja elő az egymást követő időzítések előállítását. A példában egy – a mikrogépen futó - C-nyelvű, és egy – M340 típusú PLC-n futó - LD nyelvű program található.

a. C nyelvű

```
void MT0 (void) interrupt 1
{
  switch (utm)
  {
    case 1:
    case 3:   kesl2=50000; break;
    case 2:
    case 4:   kesl2=20000; break;
    default: kesl2=8000; break;
  }
  if(++kesl1>kesl2)
    {IDB=1;kesl1=0;}
}
```

b. LD nyelvű



2. Utkereszteződés jelzőlámpáinak vezérlése:

```
unsigned char code TABL []={0x0,0x21,0x13,0x0c,0x1a};
```

```
void main (void)
{
    inic (); // inicializálás
    do
    {
        be_olv(); //bemenetek frissítése
        ENG=(BE|ENG)&~KI; // bekapcsolás
        if (ENG) // a jelzőlámpák
        { // vezérlése
            if (IDB)
            {IDB=0;
                if (++utm==5) utm=1; //ütemváltás
                P4=TABL [utm];
            }
        }
        else // sárga lámpák vill.
        {utm=0;
            if (IDB) {IDB=0; kim^=0x12;P4=kim;}
        }
    } while (1);
}
```

3. Készítse el öt különböző időzítést kapcsoló ütemadó c nyelvű, és LD programját. Az időzítések sorozata: 5 sec, 12 sec, 10 sec, 12 sec, 5 sec.