Széchenyi István Főiskola
Győr

Informatikai És Villamosmérnöki Fakultáns
Automatizálási Tanszék

Tanszéki konzulens: Dr. Hodossy László Főiskolai docens

Mérnöki project

Pozícionálási feladatok léptetőmotorral PLC, vagy számítógép vezérléssel

Győr, 1998. December 2.

Tartalomjegyzék

A kidolgozandó feladat

Léptetőmotor pozícionálása PLC vagy számítógép segítségével.

Tanszéki konzulens: Dr. Hodossy László főiskolai docens

A feladatban résztvevő diákok névsora:

1, Boros Géza

2, Dián Tibor

3, Gombás István

4, Mészáros Árpád

5, Németh Gábor

6, Sipkovits Szabolcs

7, Torma József

A részfeladatok :

Boros Géza : A PLC programozása. A vezérlőkártya angol nyelvű leírásának fordításában is segített. A léptető-motor forgásának szabályozása PLC programmal a kitalált feladat szerint. A dokumentumok PLC-s részének írása.

Dián Tibor: A PLC programozása. A vezérlőkártya konfigurálása, illetve a léptető-motor irányítása a PLC programmal a kitalált feladat szerint. A motor és a PLC összekapcsolása. A dokumentumok PLC-s részének írása.

Gombás István: A PLC léptető-motor kártyájának idegen nyelvű leírásának fordítása, illetve a program beírása az István feladata. A motor és a PLC összekötése. A dokumentumokba a léptető-motrokról szóló rész írása.

Mészáros Árpád: A lépető-motor vezérlése számítógép segítségével. Program írás és az illesztőegység tervezése illetve kivitelezése, a motor üzembe helyezése. A számítógépekről szóló fejezetek elkészítése a dokumentációkba.

Németh Gábor: A PLC programozása. A vezérlőkártya magyar nyelvű leírásának értelmezése. A motor és a PLC összekötése. A motorállvány elkészítése, összeszerelése.

Sipkovits Szabolcs: Számítógép program írásában illetve az illesztőegység elkészítésében vesz részt. A motor és a kiegészítő részegységek tartószerkezetének elkészítése. A részdokumentációk egyesítése.

Torma József: Az illesztőegység elkészítésében segít. A léptető motor lehetséges feladatának kitalálása, illetve a menet közben fellépő problémák megoldása.

BEVEZETÉS

Az eddigi rendszerek fejlesztése során az elmúlt években egyre nagyobb terület nyílt meg a léptetőmotorok alkalmazása előtt. A léptetőmotorok egyre fontosabb szerephez jutnak az automatizálás területén, felhasználása sokrétű; a legjellegzetesebb felhasználási területek a következők:

- digitális rendszerekben beavatkozó elemként;
- digitális-analóg átalakítóként (pl.: adószelszin forgatása);
- impulzustárolóként;
- matematikai műveletek végzésére alkalmazzák.

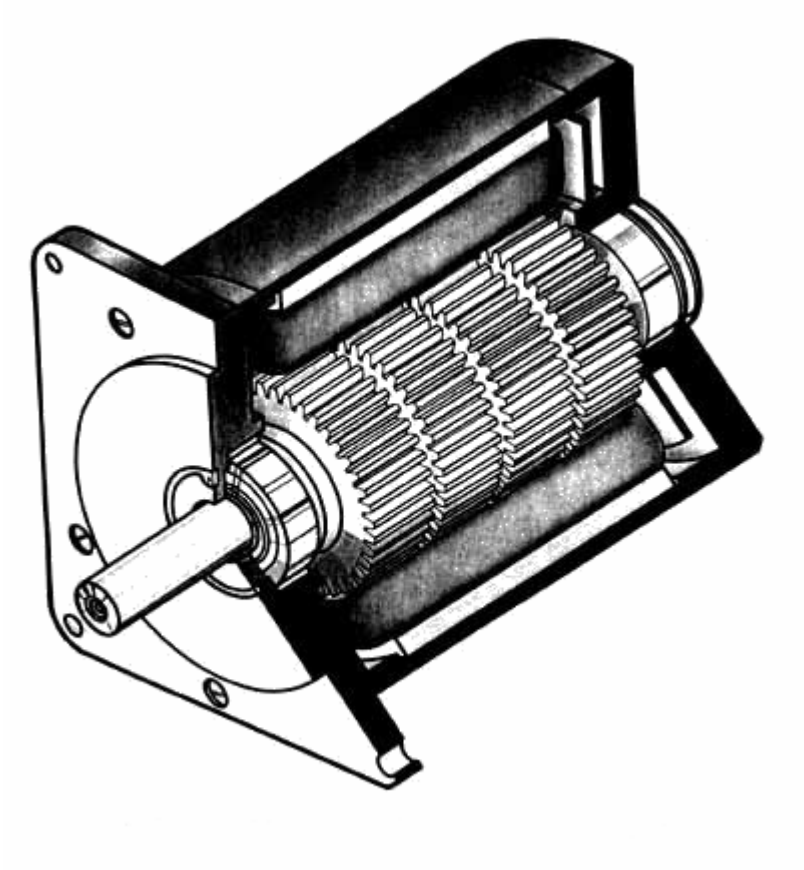
Ezek közül mégis a beavatkozó elemként való használata a legjelentősebb. A műszertechnikai alkalmazások közül a legfontosabb a távjelzés és a távmérés. Ezen kívül alkalmazzák számítástechnikai mellékhajtásoknál, X-Y rajzolók, regisztráló-és nyomtatóberendezések hajtásainál. A robottechnikában, NC, CNC hajtásoknál és az űrhajózás területén egyre nagyobb mértékben alkalmazzák. A léptetőmotoroknak az elterjedése leginkább az elektronika nagymértékű fejlődésével magyarázható, mivel a léptetőmotor felépítése egyszerű és viszonylag olcsón megvalósítható, a rendszerköltségek nagy hányadát a vezérlőelektronika teszi ki.

Léptetőmotorok működése, típusai

Léptetőmotoroknak nevezzük azokat a szinkron gépeket, amelyek szakaszosan érkező jelekkel, impulzusokkal táplálva, meghatározott nagyságú (nem folytonos) szögelfordulást vagy haladó mozgást végeznek.

A léptetőmotorok forgórészére nyugalmi helyzetben (impulzus-szünetben) meghatározott helyeken rögzítőnyomaték hat.

Ha egy beérkező impulzus a szomszédos állórész-pólusait gerjeszti, akkor a forgórész beáll az új, részére mágneses szempontból kedvezőbb irányba, vagyis α szöggel elfordul, és ott ismét rögzített állapotban marad az impulzus megszűnése után is. Egy következő, ellentétes polaritású impulzus hatására, amit az előző tekercsekre kapcsolunk, a motor forgórésze α szöggel tovább fordul.



5 fázisú léptetőmotor metszeti ábrája

<u>Léptetőmotor jellegzetes üzemállapotai</u>:

A léptetőmotoroknál a következő üzemállapotokat találjuk:

- az impulzusok szünetében a forgórész áll, villamos vagy ritkábban mechanikus úton rögzítik
- üzem a határfrekvenciánál kisebb frekvencián
- üzem a határfrekvenciánál nagyobb frekvencián

Egy-egy beérkező impulzus hatására a gép forgórésze egy-egy lépésnek megfelelő szöggel elfordul. Tekintettel arra, hogy a motor forgórészének tehetetlenségi nyomatéka, a vezérlőtekercseknek pedig induktivitásuk van (ez két energiatárolót jelent), egy-egy lépést ez a gép periodikus vagy aperiodikus lengések közepette végez. (Az örvényáramok vagy csillapítótekercsek jelenléte miatt az időállandók száma általában nagyobb kettőnél.)
Növeljük folyamatosan a motort tápláló impulzusok frekvenciáját, ekkor a motor forgórésze először a vezérlőimpulzusok ritmusára lépked. Egy bizonyos frekvencián felül a motor

gyakorlatilag a szinkron géphez hasonlóan egyenletes fordulatszámmal üzemmel. A tengelyen nyomatékokkal terhelt léptetőmotor forgórésze kissé - a terhelési szögnek megfelelően - lemarad a forgómezőhöz képest, épp úgy, mint egy nyomatékkal terhelt szinkron gép forgórésze.

Határfrekvenciának nevezzük azt a legnagyobb frekvenciát, amelynél a forgórész saját inerciája felével mereven kapcsolva képes lépéskihagyás nélkül indulási és megállási parancsokat végrehajtani.

A léptetőmotor üzemszerűen működhet a határfrekvenciánál nagyobb frekvencián is. Ilyenkor azonban annak érdekében, hogy gyorsítás és fékezés idején lépéskihagyás (lépésveszteség) ne legyen, gondoskodnunk kell a gyorsítás és fékezés programozásáról. Induláskor és leálláskor a határfrekvenciánál kisebb frekvenciájú impulzusokkal tápláljuk a motort.

<u>Léptetőmotorok felosztása</u>:

Működési elvük szerint vannak

⇒ elektromechanikus
⇒ elektromágneses
⇒ mágneses léptetőmotorok.

- aktív
- reaktív
- reluktancia-forgórészűekre.

a) Elektromechanikus léptetőmotoroknak nevezzük azokat a gépeket, amelyek mechanikus kilincsművet tartalmaznak. Üzemük lassú, zajos, és élettartamuk korlátozott. Jellemzőjük a viszonylag nagy nyomaték (40 Nm) és kis másodpercenkénti lépésszám. Táplálásuk megfelelő egyfázisú egyirányú impulzusokkal történik. Nagy előnyük, hogy nem szükséges impulzus-átalakító egység.

b) Az elektromágneses léptetőmotorok határfrekvenciája eléri a néhány kHz-et. Viszonylag jó üzemi paramétereik (súly, méret, frekvencia-jelleggörbe) vannak. Működésük zajtalan. Hátrányuk, hogy táplálásukhoz külön impulzus-átalakító egységet és a belengés ellen külső csillapítást igényelnek.

A forgóasztal kivitele

Forgóasztal a léptetőmotor tápegységével

A próbaasztal egy alumíniumból mart vastag lap, amin helyet kapott középen a léptetőmotor, ami alulról van felfogatva a lapra, valamint a referencia pont érzékelő reed-cső, amit a forgórúd végére szerelt mágnes huzat meg. A stabilitás, és önhordás miatt a négy sarkán alá van támasztva aluhengerekkel, amiknek a végein gumilábak vannak, amik elnyelik a keletkező rezgéseket. Az előlapon a kezelőszervek kaptak helyet (program, start/stop, vészkigomb) amil a hátlapra szerelt sorkapcsokhoz csatlakoznak, innen vannak elvezetve az irányítórendszer bemeneteire, valamint a léptetőmotor tápegységét is ezek kötik össze a motorral.

Léptetőmotor vezérlése számítógéppel

A számítástechnika rohamos terjedése a számítógépek sokoldalúságának köszönhető. Egy adott számítógép felhasználási területe szinte csak a rajta futtatott programtól függ, ezért az eddig analóg berendezéssel, vagy egyéb fix huzalozású logikával szerelt rendszereket igen sok helyről kiszorították a mikroprocesszoros rendszerek. A számítógép nagy előnye, hogy tömeggyártása miatt igen olcsó egy hasonló tudású mikroprocesszoros rendszer, vagy PLC árához képest, valamint programtámogatottsága is igen széles, ezért az alkalmazások fejlesztése igen egyszerű feladat. A beruházások csökkentése megoldható egy már létező számítógépes rendszer átállításával, minimális költséggel.
Az iparban is, ahol ideiglenes jelleggel, vagy kevés ráfordítással kell irányítást végezni, az ipari PC-k nyújtotta megbízhatóság mellett komolyabb feladatok is elvégezhetők.

- **A megvalósítandó feladat:**

Számítógéppel vezérelni egy léptetőmotoros rendszert, ami egy ipari folyamat részfeladatát látja el előre meghatározott fix pozícionálási pontokkal, és sorrendben. A rendszer referencia pontba állítható. A számítógépes vezérlőprogram külső forrásból veszi a sorrendi vezérlési adatokat, tehát bármikor, bármire kicserélhető a program amit végrehajt.

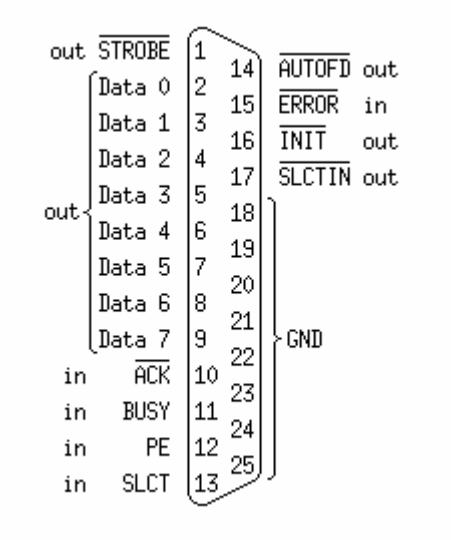
- **Kapcsolatteremtés a folyamattal**

A motorok és a számítógép közé kell egy olyan motorvezérlő elektronika, ami elvégzi a számítógép BUS-rendszerének illesztését, és leválasztását, valamint egy teljesítmény meghajtó, ami a motort már közvetlenül hajtja. A BUS illesztőt általában nem kell kivitelezni, mert a számítógépekbe már gyárilag be szoktak építeni olyan I/O kártyákat, amivel kommunikálni tudunk a külvilággal, amik főleg a ki/beviteli portok. Ezeknél lehetnek soros, és párhuzamos adatáramlást támogató típusok. A nagy reakciósebesség, és az egyszerű felépítés miatt ajánlott a párhuzamos port használata. A léptetőmotorok tekercsei érzékenyek a túlterhelésre, valamint a gyár által garantált nyomaték, és fordulatszám értékeket is csak megfelelő teljesítmény-vezérlés mellett lehet elérni, ezért egyes motorokhoz a gyár készít teljesítmény-tápegységet, ami tartalmazza a motor áramellátását szolgáló áramköröket, és a tekercsek vezérlési sorrendjét adó vezérlőt is. Ennél a számítógépet igen könnyen illeszthetjük a motorvezérlőhöz, mivel annak a digitális bemenetei általában TTL-szint kompatibilisek. Ilyen motor, és vezérlő a japán STD202 -es rendszer.

Az illesztő interfésznek csak egy jelmeghajtót, és egy optoelektronikusan megvalósított galvanikus leválasztást kell tartalmaznia, ahogy a gyár javasolja. A folyamatból érkező jeleket is a számítógép fogadja egy bemeneti porton, ezért igen biztonságos, és precíz irányítás valósítható meg

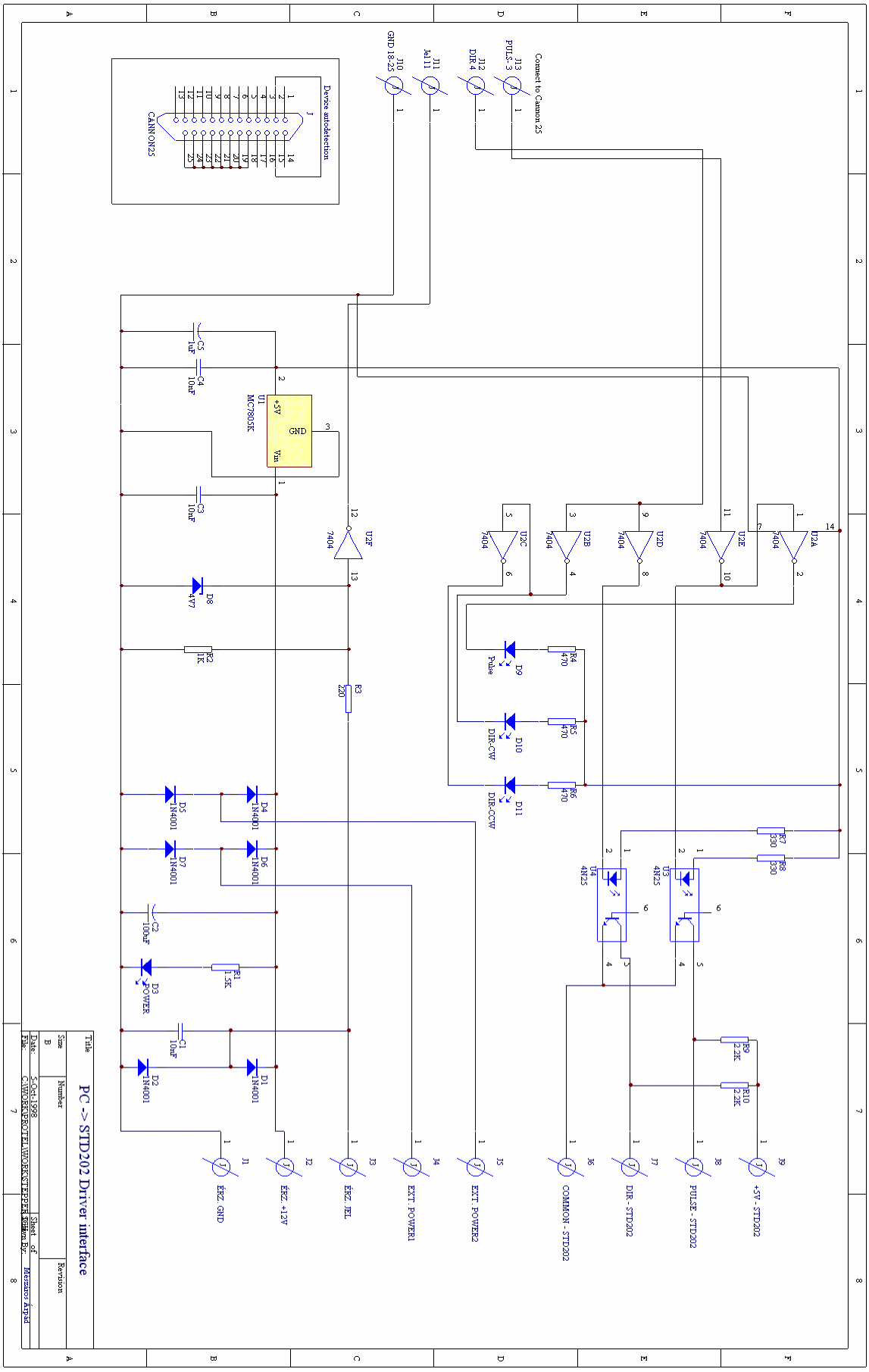
A project feladathoz egy IBM PC kompatibilis számítógéptípust választottunk, mivel ez igen elterjedt típus, szinte mindenhol megtalálható, így a program hordozhatósága garantált. Természetesen lehet más számítógéppel is vezérelni, de ott típusfüggő szoftvert kell rá csinálni.

Az illesztő egység felépítése, működése



Az IBM PC kompatíbilis gépeken szériatartozéknak számít a (serial) soros port, ami egy egyszerű perifériaillesztő, amin van egy egyirányú kimenő 8 bites adatport, egy 4 bites bejövő státuszport, és egy 5 bites kimenő controll port. Az illesztő valósítja meg a PC buszrendszere, és a külvilág között a kapcsolatot. A programozása is igen egyszerű, mivel rendelkezik három I/O címmel, ami a 3 portot közvetlenül vezérli, tulajdonképpen egy puffer. Mivel szabványos, ezért kiváló lehetőséget biztosít különböző külső egységek vezérlésére, holott eredetileg csak nyomtatóillesztőnek szánták. Az illesztő egységet is erről a portról vezéreljük, mivel TTL kompatíbilis jelekkel dolgozik, ezért szinte bármilyen digitális egység meghajtható róla.

Az illesztő funkciója alapvetően az, hogy leválassza a ki, és bemenő jeleket a számítógép és az irányított folyamat között. Tartalmaz minden jelhez egy meghajtó puffert, valamint ahol szükséges ott a jelek galvanikus leválasztását is elvégzi. Egyéb szolgáltatása a folyamat jeleinek vizuális megjelenítése a könnyebb követhetőség, és hibakeresés könnyítése céljából.



Az illesztő interface kapcsolása

Működése:

Mivel a számítógépből nincsen gyárilag tápfeszültség kivezetve, ezért az illesztő külső tápfeszültséget igényel, melynek értéke 12..18V. A bemenő tápfeszültséget először D4-D7 diódákból kialakított Graetz hídra vezetjük (Ext power1, Ext power2), ezzel univerzális táplálású a készülék, mivel a bemenő feszültség polaritása közömbös, sőt, még váltakozó áramú táplálás is lehetséges. A bejövő tápfeszültség szűretlenségéből adódó hibákat C2 szűrőkondenzátor küszöböli ki. Az R1-D3 LED a tápfeszültség meglétét jelzi ki. A TTL kapuknak 5V-os tápfeszültségre van szükségük, ezért ezt a feszültséget (12-18V) még le kell csökkenteni. A stabil 5V-ot U1 7805 pozitív stabilizátor IC állítja elő. C3 és C4 feladata az IC gerjedésének megakadályozása, C5 pedig utószűrő. A párhuzamos portról 2 vezérlő jelet, és egy bemenő jelet vezetünk ki, valamint a rendszerföldet. Még szükséges 2 további jel is, de ezek csak a program szempontjából fontosak, a szerepük a hardveregység meglétének ellenőrzése (csatlakoztatva van-e?) Ezek a D0 adatkimenet, és az ERR státuszbemenet összekötése, a dugóban. A számunkra szükséges jelek a D1 adatkimenet ami a léptetést vezérli (PULSE), a D2 adatkimenet ami a forgásirányt határozza meg (DIR), valamint a BUSY státuszbemenet amivel az irányított folyamat kezdőállapotának beállítását tudjuk kontrollálni egy érzékelő segítségével (Érz jel - motor null pozíció beállítása).

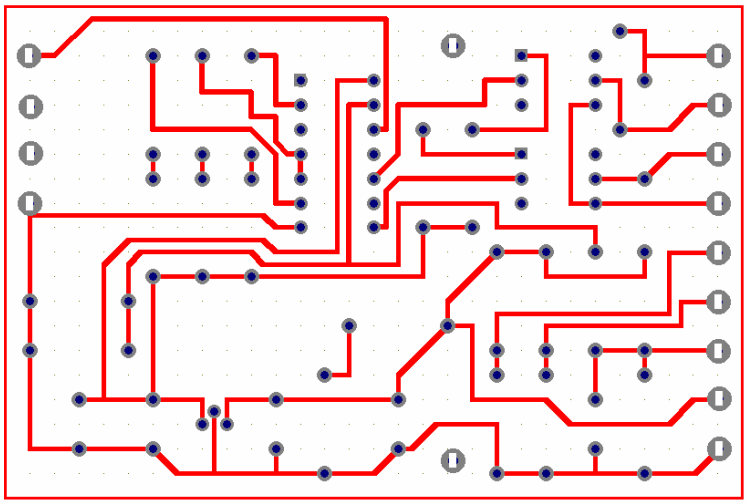
Az STD202 léptetőmotor teljesítmény meghajtót ezzel a 2 jellel kell vezérelni, mégpedig úgy, hogy a PULSE kimeneten egy pozitív impulzus jelent egy léptetést a motoron, ami egy 2 fázisú bipoláris hibrid léptetőmotor, tehát 200 lépés 1 teljes körülfordulás, azaz 1.8° egy lépés. A meghajtó viszont half stepping üzemmódban vezérli a motort, ami 400 lépés/körülfordulást jelent, amihez 400 db TTL szintű impulzus szükséges. A forgás iránya a DIR jellel állítható.

A D1 pulse jel az U2E inverterre érkezik, ami egy open kollektoros teljesítményinverter. Mivel az STD202 meghajtót galvanikusan le kell választani a számítógépről, ez az inverter hajtja meg az U3 optocsatoló LED-jét. A DIR jel U2D inverterre érkezik, ami az U4 optocsatoló LED-jét hajtja meg. A szekunder oldalon a fotótranzisztorok gyári ajánlás szerint vannak normál aktív közös emitteres kapcsoló üzemű kapcsolásban. A kollektorellenállások magából a meghajtóegységből vannak táplálva, aminek a közös vezetéke megy az emitterekre, és a kollektorokról veszi ki a jelet. Mivel az optocsatoló ebben a kapcsolásban fázist fordít, ezért az inverterek fázisfordítását kiküszöböljük. U2A hajtja meg az R4-D9 LED-et, ami minden impulzusra felvillan. U2B, és U2C az R5-D10, R6-D11 LED-eket hajtják meg, ahol a LED-ek az aktuális forgásirányt jelzik ki.

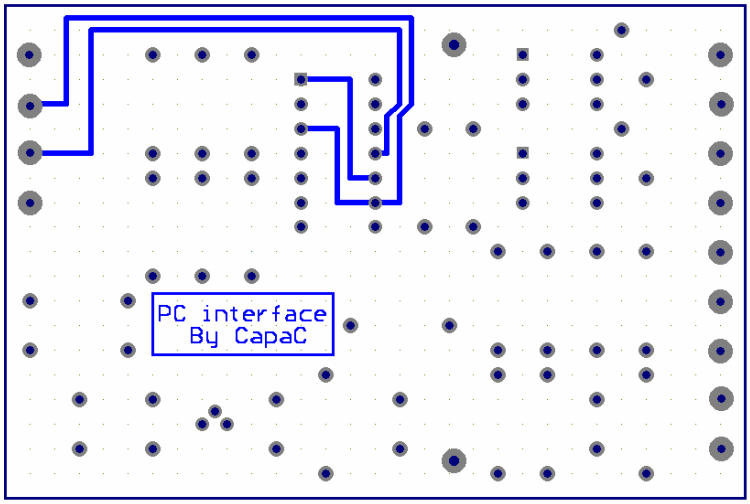
A nullpozíció érzékelőnek egy +12V-os jelet kell a bemenet, és a test közé juttatnia. Ez lehet végálláskapcsoló, reed cső, vagy induktív/kapacitív közeledésérzékelő. A bemeneten kívül ki van vezetve még a rendszerföld, és a +12V is amiről üzemelhetnek az érzékelő egységek. A zajos környezetet figyelembe véve a bemenetnél szűrő egységek találhatóak. A D1-D2 dióda páros a túlfeszültséget vágja, ami a bemenetre kerül. A C1 a kisebb amplitúdójú feszültségtüskéket szűri le a bemenetről. Az U2F inverter illeszti a bemenő jelet a státuszporthoz, de mivel ez TTL kapu, ezért a bemenetén vágni kell a jelet, amit R3-D8 Zenner dióda végez 4.7V-ra vágva a jelet. R2 az inverter bemenetének testrehúzását végzi. Mivel a BUSY bemenet invertált, ezért az inverter visszafordítja a jelet, azaz nincs negáció.

Megvalósítás

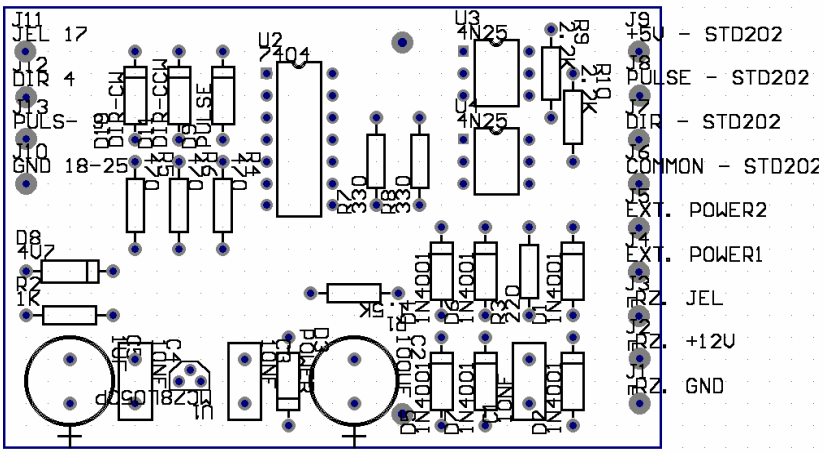
Az áramkört kétoldalon fóliázott nyomtatott áramköri kártyára kell szerelni. A kártya huzalozási tervét számítógép segítségével nyomtatható alakra kell hozni és pausz papírra igen erős tónussal lézer nyomtatóval kinyomtatni. Ezzel elkészült a mesterfilm, ami a fotorezisztív eljáráshoz a maszkot adja. A kártya fotorezisztív eljárással készül majd el. A letisztított panelt mindkét oldalán befújjuk P20-as fotolakkal, majd száradni hagyjuk 1 órán át. Ezután ráillesszük a maszkot pontosan mindkét oldalra, és rögzítjük üveglapok között. Ezután erős UV sugárzás alá tesszük kb. 5 percre, majd megfordítjuk, és újabb 5 percig világítjuk. Világítás után nátriumhidroxidos oldatban előhívjuk a lakkot. Előhívás után maratóoldatban lemaratjuk a felesleges részeket, majd kifúrjuk a lyukakat rajta, és védőlakkal vonjuk be. Ezután beforrasszuk az alkatrészeket, és élesztjük az egységet. Élesztés után műanyagdobozba szereljük.



Alsó oldali fóliarajzolat

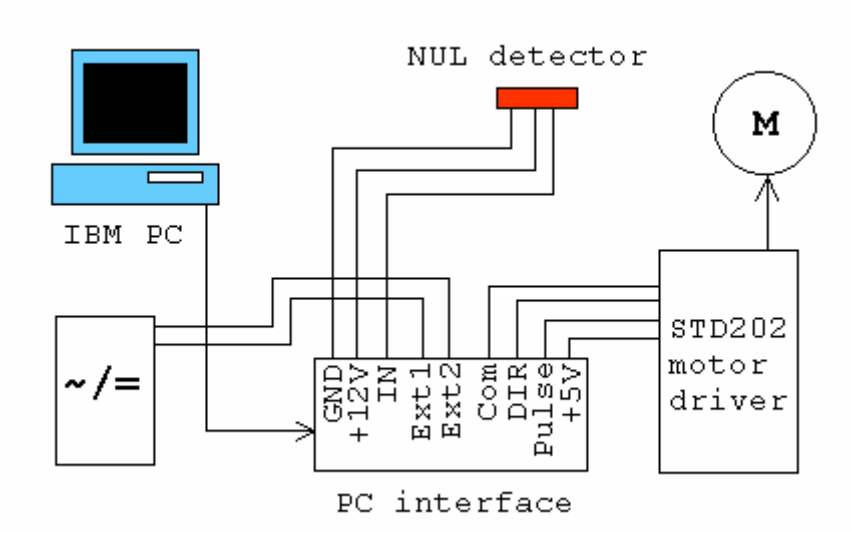


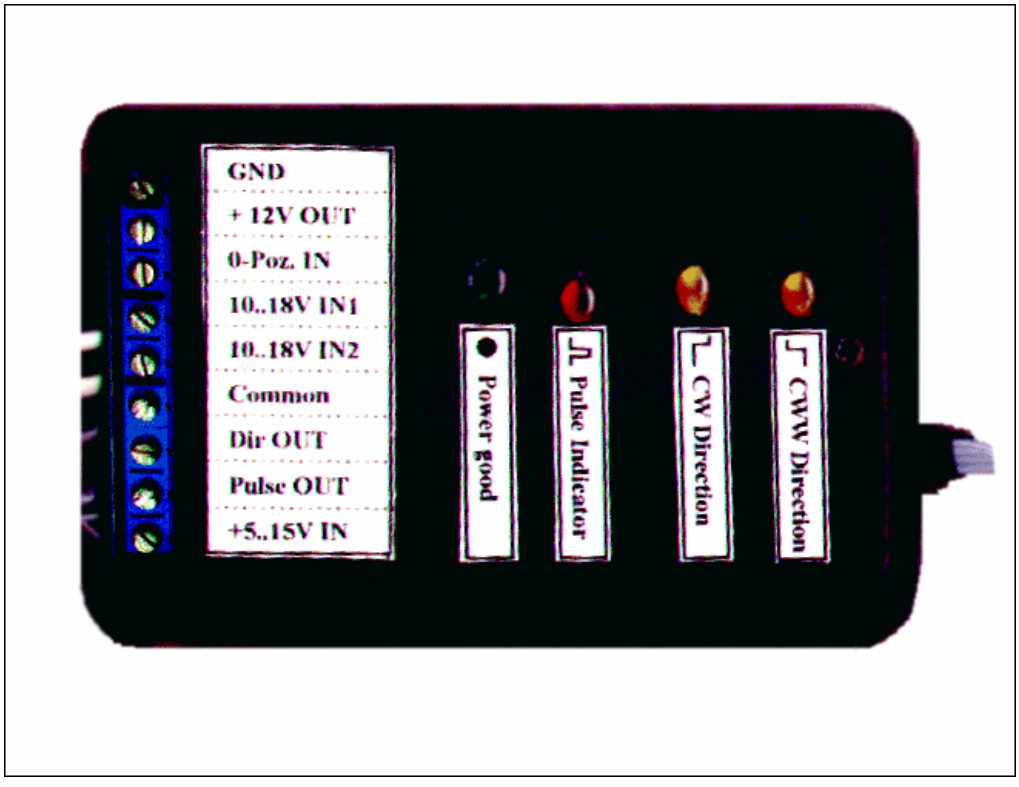
Felső oldali fóliarajzolat



Alkatrész beültetési rajz

Az egység bekötése:





A készre szerelt illesztődoboz

A számítógépes program

Az illesztő vezérlését egy program valósítja meg. A program interaktív módon kéri be az adatot, mely egy speciális szövegfájl, ami a program script nyelvén íródott, és tartalmazza a vezérlési feladatot (lépések száma, sebesség, irány, ismétlés). A vezérlési szövegfájl egy nem formázó szövegszerkesztővel elkészíthető (lásd - példa script fájl). A program magja egy ütemadó, ami a gép időzítő megszakítására ül rá, és szabályos időközökben 1/2000 másodpercenként ad egy ütemet. A sebesség érték határozza meg, hogy hány ütem elteltével adjon ki egy impulzust. Az impulzus szélessége 100 mikroszekundum amit passzív időzítéssel a rendszerbusz konstans időzítését figyelembe véve egy másik rutin ad ki. A nullhelyzet-érzékelőn bekapcsoláskor beállítja a referencia pozíciót. A végálláskapcsoló pergése miatt szoftveres pergésmentesítés van. A folyamat befejezése után várja a további adatok bevitelét. Ez a program már alkalmas komplex, összetett feladatok elvégzésére, mivel tartalmaz egy valós idejű fordítót, és így már csak a megírt script határozza meg hogy milyen feladatra akarjuk használni.

A program forráskódja:

```
Program PC_Interface_for_Stepper_Motor;

Uses Crt,Dos,Graph;

Type
     CountBuf    =       array[1..10000] of longint;
     PCountBuf   =       ^CountBuf;       {Script szekvenciatároló}

Const
                                          {* LPT bázisport bitek *}
     TEST        =       1;               {D0 teszt bit maszk}
     PULSE       =       2;               {D1 Pulse bit maszk}
     DIR         =       4;               {D2 Direction bit maszk}
                                          {* LPT státusz bitek *}
     SENSOR      =       128;             {-Busy input státusz bit maszk}
     ERR         =       8;               {Error autódetektálás bemenet maszk}
                                          {* Egyéb munkaregiszterek *}
     Speed       :       word    = 1;     {Impulzus periódus: 500us*Speed}
     PulseNr     :       word    = 1;     {Motor lépésszám}
     OnWork      :       boolean = False; {Folyamat végrehajtás engedélyező}
     Count       :       Word    = 0;     {megszakítás számláló}

     ON          =       True;
     OFF         =       False;
     PulseW      :       word    = 20;  {Impulzusszélesség : 200us}
     ProgRep     :       integer = 1;   {Script ciklus számláló0=végtelen}
                                        {Grafikus keret poligónjai}
     Poly        :       array[0..15,0..3] of PointType =
                 (((x:0;y:0),(x:40;y:0),(x:60;y:40),(x:20;y:40)),
                  ((x:80;y:0),(x:120;y:0),(x:140;y:40),(x:100;y:40)),
                  ((x:160;y:0),(x:200;y:0),(x:220;y:40),(x:180;y:40)),
                  ((x:240;y:0),(x:280;y:0),(x:300;y:40),(x:260;y:40)),
                  ((x:320;y:0),(x:360;y:0),(x:380;y:40),(x:340;y:40)),
                  ((x:400;y:0),(x:440;y:0),(x:460;y:40),(x:420;y:40)),
                  ((x:480;y:0),(x:520;y:0),(x:540;y:40),(x:500;y:40)),
                  ((x:560;y:0),(x:600;y:0),(x:620;y:40),(x:580;y:40)),
                  ((x:0;y:440),(x:40;y:440),(x:60;y:480),(x:20;y:480)),
                  ((x:80;y:440),(x:120;y:440),(x:140;y:480),(x:100;y:480)),
                  ((x:160;y:440),(x:200;y:440),(x:220;y:480),(x:180;y:480)),
                  ((x:240;y:440),(x:280;y:440),(x:300;y:480),(x:260;y:480)),
                  ((x:320;y:440),(x:360;y:440),(x:380;y:480),(x:340;y:480)),
                  ((x:400;y:440),(x:440;y:440),(x:460;y:480),(x:420;y:480)),
                  ((x:480;y:440),(x:520;y:440),(x:540;y:480),(x:500;y:480)),
                  ((x:560;y:440),(x:600;y:440),(x:620;y:480),(x:580;y:480)));

VAR
     LptBase             : word;          {Párhuzamos port báziscíme}
     LptStatus           : word;          {Párhuzamos port státuszportja}
     LptCommand          : word;          {Párhuzamos port parancsportja}
     OldInt08            : Procedure;     {eredeti 08h megszakításcím}
     Ch                  : Char;
     Bevitel             : string;
     NulCount            : word;          {referencia pont beállítás segéd}
     Error,i,j           : integer;
     Ut                  : PCountBuf;              {Script szekvenciatömb}
     Inputbuf            : array[0..13] of char;   {adatbeviteli puffer}
```

```
{----------------------------Grafikus driver beépítése--------------------}
{$L EGAVGA.OBJ}
procedure egavgadriver;     external;

{----------------------------Végzetes hiba kezelő-------------------------}
Procedure CriticalError(s:string);
Begin
       CloseGraph;
       Textmode(lastmode);
       Writeln;
       Writeln('------------------------- ERROR --------------------',#7);
       Writeln(s);
       Writeln('Üss Entert...');
       readln;
       halt;
End;

{----------------------LPT portok ellenőrzése, beállítása----------------}
Procedure INITLPT;
var
   lp1,lp2,lp3 : boolean;
   C           : Char;
Begin
     lp1:=true; lp2:=true; lp3:=true;
     LptBase:=0;
     if memw[$0000:$408] = 0 then lp1:=false;      {port létezik-e?}
     if memw[$0000:$40A] = 0 then lp2:=false;
     if memw[$0000:$40C] = 0 then lp3:=false;
     if not (lp1 or lp2 or lp3) then               {ha nem létezik kilépés}
       CriticalError('Printer port not found !');
     if (lp1 and (lp2 or lp3)) or (lp2 and (lp1 or lp3)) or
        (lp3 and (lp1 or lp2)) then begin          {ha több van, választás}
       writeln;
       writeln('$ Melyik LPT portot akarod használni?');
       writeln;
       if lp1 then writeln('1.  LPT1');
       if lp2 then writeln('2.  LPT2');
       if lp3 then writeln('3.  LPT3');
      repeat
       C:=readkey;
       if (C='1') and lp1 then LptBase:=memw[$0000:$408];
       if (C='2') and lp2 then LptBase:=memw[$0000:$40A];
       if (C='3') and lp3 then LptBase:=memw[$0000:$40C];
      until LptBase>0;
     end
     else begin
       if lp1 then LptBase:=memw[$0000:$408];        {port beállítása}
       if lp2 then LptBase:=memw[$0000:$40A];
       if lp3 then LptBase:=memw[$0000:$40C];
     end;
     LptStatus:=LptBase+1;
     LptCommand:=LptBase+2;
     port[LptCommand]:=192;                          {init port}
     Delay(60);
     port[LptCommand]:=196;
     Delay(10);
     port[LptBase]:=0;                               {vezérlö detektálása}
     if (port[LptStatus] and ERR) = ERR then
       CriticalError('Interface hardvert nem találom!');
     port[LptBase]:=1;                             {ha nincs rádugva kilépés}
     if (port[LptStatus] and ERR) = 0 then
       CriticalError('Interface hardvert nem találom!');
End;
```

```
{-------------------------Nullpozició kapcsoló beolvasása---------------}
Function ReadSensor : Boolean;
begin
      if (port[LptStatus] and SENSOR) = 128 then ReadSensor:=True
      else ReadSensor:=False;
end;

{-------------------------Irány kimenet állítása------------------------}
Procedure SetDir(Dire : Boolean);
Begin
    if dire then Port[LptBase]:=Port[LptBase] AND (255-DIR)
    else Port[LptBase]:=Port[LptBase] OR DIR;
End;

{-------------------------Pulzus kimenet állítása (teszt)---------------}
Procedure SetPulse(Dire : Boolean);
Begin
    if dire then Port[LptBase]:=Port[LptBase] AND (255-PULSE)
    else Port[LptBase]:=Port[LptBase] OR PULSE;
End;

{-----------------------------1 impulzus kiadása-------------------------}
Procedure Impulzus; Assembler;
asm
   mov dx,lptbase            {LPT port beolvasása}
   in  al,dx
   and al,255-PULSE          {maszkolás a Pulse bitre}
   out dx,al                 {visszaírás a portra}
   xor ax,ax
   mov dx,0
   mov cx,PulseW             {várakozás PulseW ciklusszor}
@ide:
   in  ax,dx
   loop @ide
   mov dx,lptbase            {LPT port beolvasása}
   in  al,dx
   or  al,PULSE              {unmaszkolás a Pulse bitre}
   out dx,al                 {visszaírás a portra}
end;

{----------------------------RTC időzítő megszakítás beállítás------------}
Procedure SetIntTick(Divid : Word); assembler;
asm
   cli                {CTC timer 18.2 interrupt/sec => X interrupt/sec}
   mov al,00110110b
   out 43h,al         {0-ás rendszeridőzítő kiválasztva}
   mov ax,Divid
   out 40h,al         {osztási arány beállítása 1.147.800/divid}
   mov al,ah
   out 40h,al
   sti                {megszakítás engedélyezése}
 end;

{--------------------------Windows ellenőrző-----------------------------}
function CheckWindows : boolean; assembler;
asm
   mov  ax,1600h
   int  2Fh
   cmp  al,1
   jbe  @check2
   cmp  al,80h
   jae  @check2
   mov  al,0
   jmp  @windows_detected
@check2:
   mov  ax,4680h
```

```
   int  2Fh
   or   ax,ax
   jnz  @no_windows_detected
   jmp  @windows_detected
@no_windows_detected:
   mov al,$FF
@windows_detected:
   not al
end;

{---------------------------Ütemvezérlő megszakításkezelö----------------}
Procedure WorkIRQ; Assembler;
Asm
   push DS
   push ax
   push bx
   mov  ax, Seg @Data
   mov  DS,ax
                        {Init interrupt}
   mov  al,OnWork       {ha OnWork True akkor lefut a megszakítás}
   xor  al,0
   jz   @kilepes
   mov  ax,Count        {Count novelese}
   inc  ax
   mov  Count,ax
   xor  ax,Speed        {összehasonlítás a Speed-del}
   jnz  @kilepes
   mov  Count,0         {időzíts lejárt, vezérlés!}
   call impulzus        {impulzus kiadása}
   mov  ax,PulseNr      {PulseNr csökkentése}
   dec  ax
   mov  PulseNr,ax
   xor  ax,0
   jnz  @kilepes        {ha még van lépés továbbfut}
   mov  OnWork,0        {ha nincs lépés leállítja a folyamatot}
@kilepes:
                        {Exit interrupt}
   mov  al,20h
   out  20h,al
   pop  bx
   pop  ax
   pop  DS
   iret
End;

{---------------------------Script feldolgozás/forditás-----------------}
Procedure Script(var Buf : PCountBuf);
var Bev  : string;
    Scr1 : Text;
    i    : integer;
    cod  : integer;

begin
    New(Buf);                                 {memóriafoglalás a szkriptnek}
    for i:=1 to 10000 do buf^[i]:=100000;  {szekvenciatömb alapállapotba}
    i:=1;
    Settextbuf(input,inputbuf);               {bevihető karakterek 8+3}
                                              {* fájlbekérés *}
      repeat
        Setfillstyle(1,yellow);               {üzenetdoboz kirajzolása}
        Bar(50,70,310,100);
        Setcolor(black);
        Rectangle(50,70,310,100);
        outtextxy(60,85,'Script fájl neve: ');
        gotoxy(26,6);
        Readln(bev);                          {fájlnév beolvasása}
        assign(scr1,bev);
```

```
      {$I-}reset(scr1);{$I+}             {fájl meglétének ellenőrzése}
    until ((IOresult=0) and (length(bev)<>0)) or (bev='*');
    if bev='*' then begin                {* feldolgozás *}
      SetIntVec($08,@OldInt08);
      SetIntTick(65535);
      Dispose(buf);
      criticalerror('Felhasználói megszakítás');
    end;
  Readln(scr1,bev);
  if bev<>'Script For Stepping Controll By CapaC v1.0' then begin
    SetIntVec($08,@OldInt08);        {Időzítés megszakítás kikapcsolás}
    SetIntTick(65535);               {Sebesség vissza 18/sec-ra}
    close(scr1);                     {fájl lezárás}
    Dispose(buf);                    {memória felszabadítás}
    criticalerror('Ez nem script fájl!');
  end;
  While not eof(scr1) do begin       {fájlvég jelig dolgoz fel}
    Readln(scr1,bev);                {köv. sor beolvasása}
    if bev[1]<>';' then begin          {ha pontosvessző van elöl kihagy}
      if pos('SET',bev)>0 then begin   {SET sor feldolgozása}
        bev:=copy(bev,5,length(bev));  {lépésszám, és irány tárolása}
        val(bev,buf^[i],cod);
        if cod<>0 then begin           {hibás adat kilépés}
          SetIntVec($08,@OldInt08);
          SetIntTick(65535);
          close(scr1);
          Dispose(buf);
          criticalerror('Script hiba! - SET');
        end;
        inc(i);
      end;
      if pos('SPD',bev)>0 then begin   {SPD sor feldolgozása}
        bev:=copy(bev,5,length(bev));
        val(bev,buf^[i],cod);          {sebesség osztó tárolása}
        if cod<>0 then begin           {hibás adat kilépés}
          SetIntVec($08,@OldInt08);
          SetIntTick(65535);
          close(scr1);
          Dispose(buf);
          criticalerror('Script hiba! - SPD');
        end;
        inc(i);
      end;
      if pos('REP',bev)>0 then begin   {REP sor feldolgozása}
        bev:=copy(bev,5,length(bev));
        val(bev,ProgRep,cod);          {ciklusismétlés tárolása}
        if cod<>0 then begin           {hibás adat kilépés}
          SetIntVec($08,@OldInt08);
          SetIntTick(65535);
          close(scr1);
          Dispose(buf);
          criticalerror('Script hiba! - REP');
        end;
      end;
    end;
  end;
  close(scr1);
end;
```

```
{---------------------------VGA 640x480 16c bekapcsolás-------------------}
Procedure Graphon;
const
   driver : integer =VGA;
   mode   : integer =VGAHI;
begin
    if registerbgidriver(addr(egavgadriver)) <0 then
      Criticalerror(grapherrormsg(graphresult));
      initgraph(driver,mode,' ');
      if graphresult<>0 then criticalerror('Grafikus hiba!');
end;

{---------------------------üzemelés kijelző-----------------------------}
Procedure WorkDisplay(sta : boolean);
begin
   if sta=OFF then begin
     Setfillstyle(1,lightred);
     fillellipse(210,250,10,10);
     Setfillstyle(1,yellow);
     fillellipse(110,250,10,10);
   end
   else begin
     Setfillstyle(1,lightred);
     fillellipse(110,250,10,10);
     Setfillstyle(1,yellow);
     fillellipse(210,250,10,10);
   end;
end;

{---------------------------állapot kijelző------------------------------}
Procedure StatusDisplay(cnt,dir,spd,actrep : longint);
var
   st   :       string;
   err  :       integer;
begin
    Setcolor(black);
    Setfillstyle(1,yellow);
    Bar(490,71,548,200);
    str(cnt,st);
    outtextxy(345,110,'Összes lépés     : '+st);
    if dir=0 then st:='CW '+#25
    else st:='CWW '+#24;
    outtextxy(345,130,'Irány            : '+st);
    str(spd,st);
    outtextxy(345,150,'Sebesség         : '+st);
    str(actrep,st);
    outtextxy(345,190,'Aktuális ismétlés: '+st);
end;

{---------------------------Státuszablak kirajzolás----------------------}
Procedure Statusz(s : string);
Begin
    Setcolor(black);
    Setfillstyle(1,white);
    Bar(332,280,548,300);
    Rectangle(332,280,548,300);
    Outtextxy(335,290,s);
end;
```

```
{------------------------------Főprogram----------------------------------}
BEGIN
    Textmode(CO80);                              {képernyő beállítása}
    directvideo:=false;
    ClrScr;
    TextColor(cyan);
    Writeln('$ Windows ellenőrzése...');         {Windows alól nem indul!}
    if checkwindows then
      CriticalError('Ez a program nem fut Windows alól!')
    else Writeln('$ OK!');
    Writeln('$ Párhuzamos potrok vizsgálata...');
    InitLpt;                                     {LPT port bekonfigurálása}
    Writeln('$ OK!');
    Writeln('$ Inicializálás...');               {vezérlő alapállapotba}
    SetDir(OFF);
    SetPulse(OFF);
    NulCount:=0;                                 {mullpozíció beállítása}
    While (not ReadSensor) and (NulCount<=400) do begin
      Impulzus;
      inc(NulCount);
      delay(5);
    end;
    if NulCount>400 then
     CriticalError('$ Kifutott a tartományból! Null pozíció nincs
beállítva!')
    else writeln('$ Null pozíció beállítva');
    delay(2000);
                                            {grafikus képernyő kirakása}
    GraphON;
    Setbkcolor(black);                      {ablakok kirajzolása}
    Setcolor(black);
    Setfillstyle(1,yellow);
    Bar(0,0,640,480);
    Setfillstyle(9,lightgray);
    Bar(40,40,600,440);
    Rectangle(0,40,640,440);
    Setfillstyle(1,black);
    for i:=0 to 15 do fillpoly(4,poly[i]);
    Setfillstyle(1,yellow);
    Setcolor(black);
    Bar(330,50,550,400);
    Rectangle(330,50,550,400);
    Outtextxy(370,60,'Státusz display');
    Line(330,70,550,70);
    StatusDisplay(0,0,0,0);
    Bar(80,200,250,300);
    Rectangle(80,200,250,300);
    Outtextxy(100,210,'Végrehajtás alatt');
    Line(80,220,250,220);
    Outtextxy(100,280,'IGEN');
    Outtextxy(200,280,'NEM');
    WorkDisplay(OFF);
    Setcolor(white);
    Setfillstyle(1,lightred);
    Bar(80,350,250,400);
    Outtextxy(100,370,'LEALLITAS: ESC');
                                        {* IRQ ütemadó indítása *}
    SetIntTick(600);                    {1193180/tick=>2000/sec}
    GetIntVec($08,@OldInt08);
    SetIntVec($08,@WorkIrq);            {Időzítés megszakítás bekapcsolás}
  Repeat
                                        {* Script feldolgozás *}
    Statusz('Írja be a Script nevét!');
    Script(ut);
    Statusz('Script OK! Space-indit');  {Space indítja a folyamatot}
```

```
Repeat
     ch:=readkey;
    until ch=#32;
    ch:=#0;
    Statusz('Script futtatás...');
                                    {ha 0 a rep, akkor végtelen ismétlés}
    if ProgRep=0 then ProgRep:=-1;
    Repeat
     i:=1;
     Repeat
       PulseNr:=abs(ut^[i]);
       if abs(ut^[i])=ut^[i] then begin
         SetDir(ON);                {negatív lépésnél fordított irány}
         j:=0;
       end
       else begin
         SetDir(OFF);
         j:=1;
       end;
       inc(i);
       Speed:=ut^[i];
       WorkDisplay(ON);                    {képernyő frissítés adatokkal}
       StatusDisplay(PulseNr,j,Speed,ProgRep);
       OnWork:=True;                         {folyamat indul}
       repeat
         if keypressed then ch:=readkey;  {ESC és a folyamat vége figyel}
       until (OnWork=false) or (ch=#27);
       WorkDisplay(OFF);                   {leállás, köv. folyamat betöltés}
       inc(i);
     until (ch=#27) or (ut^[i]=100000);    {ESC vagy szekvenciatömb végéig}
     if ProgRep>0 then dec(progrep);       {ciklus csökk. ha nem végtelen}
    until (ch=#27) or (Progrep=0);         {ESC vagy ciklus végéig megy}
    onwork:=false;                         {folyamat tiltás}
    Statusz('Script befejezve. Kilép?');
    repeat
      Ch:=readkey;
      Ch:=upcase(Ch);
    until Ch in ['I','N'];                 {Enter-re vár}
  until Ch='I';
    Dispose(ut);
    SetIntVec($08,@OldInt08);              {Időzítés megszakítás kikapcsolás}
    SetIntTick(65535);                     {Sebesség vissza 18/sec-ra}
    closegraph;
    writeln('Bye!');
END.of program                             {kilépés.}
```

Minta script a script nyelv leírására

```
Script For Stepping Controll By CapaC v1.0
;Fejléc, típus azonosító MINDIG KELL a legelső sorba!
;
;      *******        Példa script funkciók bemutatása      *********
;
;a pontosvesszős sort nem veszi figyelembe ez a komment
SET -400
;400 lépés hátra, relatív elfordulás megadással ami teljes fordulat
;előjel az irányt adja, kis/nagybetű számit!
SPD 6
;sebesség osztó értéke az 1 a legkisebb ami a leggyorsabb 2000 lépés/sec
;a 2 utasíttáspár együtt van! SET-SPD alkot párt
SET 400
SPD 3
;vissza a referencia pontba, dupla sebességgel
SET 2000
SPD 2
;öt fordulat gyorsan
SET 1
SPD 2000
;vár 1 másodpercet, a köv. SET-nél 1-el kevesebbet kell megadni
REP 5
;eddigiek ismétlése X-szer ha X=0 akkor végtelen ismétlés
END
;vége
```

Léptetőmotor vezérlése PLC-vel

A PLC - s project leírása :

Egy olyan pozicionálási feladat elmélet és gyakorlati megvalósítása, amely konkrét pozícionálási programsorozatot hajt végre, amely egy az ipar területén elképzelt részfolyamathoz illeszkedik. A program több lehetséges programsorozatot hajt végre, amely kapcsolók segítségével választható ki, indítható, leállítható, és tartalmaz egy kezdőpont érzékelést.

A léptetőmotor egy állványba van beépítve ahol egy áthelyezőkart forgat, melynek öt pozícionálási pontja van, amelyek a következők:

1. A referencia pont és a munkadarab behozatali pontja.
2. Az "A" munkagép pozícionálási pontja.
3. Az "B" munkagép pozícionálási pontja.
4. Az "C" munkagép pozícionálási pontja.
5. A munkadarab kiviteli pontja.

A program két programsorozatot tud végrehajtani, melyeket kapcsolók segítségével választhatjuk ki. A programnak képes egyszeri és folyamatos futásra. A motor mindig várakozik míg a munkagép elvégzi feladatait.

"A" program leírása: Az 1-től végighalad az összes ponton keresztül az 5-ig majd visszatér az 1-es pontba.

"B" program leírása: 1-es pont, 2-es pont, 3-as pont, 4-es pont, majd visszatér a 2-es pontba ezután megy az 5-ös pontba, majd visszatér az 1-es pontba.

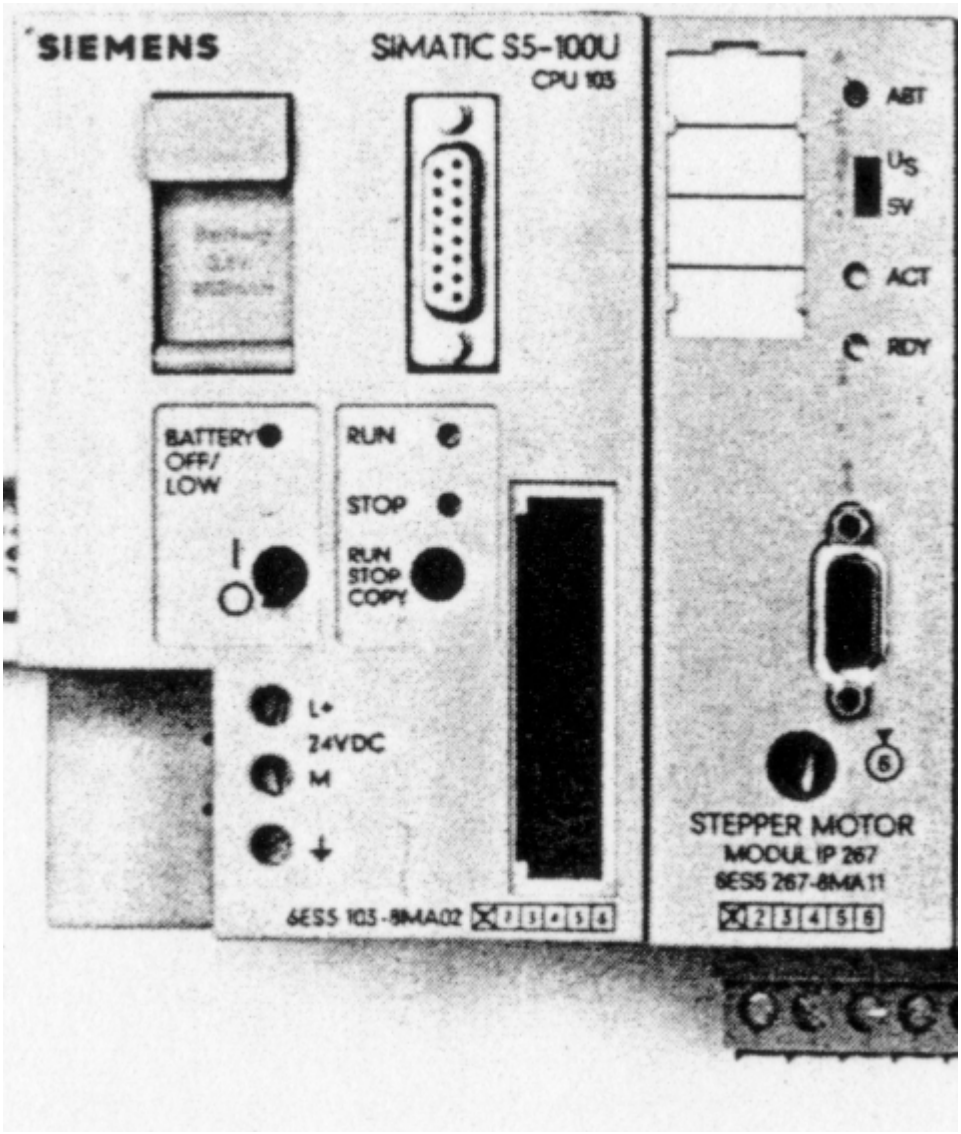
Programozható logikai vezérlők

A PLC mint irányítástechnikai eszköz

A programozható logikai vezérlő, egy olyan feladatorientált célszámítógép, amely tárolt program alapján képes kombinációs és szekvenciális hálózatok megvalósítására. Alkalmas ipari környezetben megfelelő vezérlési feladatok ellátására. Az irányítási algoritmus könnyen, gyorsan módosítható, így rendkívül rugalmas eszköz. Ez a tény is indokolja a
PLC - k rendkívül gyors elterjedését. Egy korszerű PLC képes a hagyományos vezérlési feladatok ellátásán túl több szabályozó kört kezelni, aritmetikai műveleteket elvégezni, analóg jeleket fogadni és feldolgozni, alá vagy mellé rendelt viszonyban hálózatban összekapcsoltan együttműködni, tehát intelligens folyamatirányító gépnek tekintendő.

SIMATIC S5-90U, S5-95U

Az S5-90U és az S5-95U mini kompakt kialakítású PLC - k. Két PLC közül az S5-95U a nagyobb teljesítményű, ez a nagyobb memória méretben, több aritmetikai funkcióban, több bemeneti és kimeneti csatlakozásban és több I/O modul alkalmazásában mutatkozik meg (max. 6 ill. 32 modul). Ciklikus elven működnek, a ciklus első lépéseként valamennyi bemenet állapotát lekérdezi és az operatív memóriában tárolja. Ezután a programmemória egyes címein található utasítások sorban egymás utáni beolvasása, értelmezése, végrehajtása következik, majd a kimenetek megfelelő beállítása történik meg. Soros logikáról beszélünk, mivel a logikai függvények lépésről lépésre haladva, időben sorosan épülnek fel.



Simatic S5-100U PLC és a hozzá csatolt IP267-es vezérlő

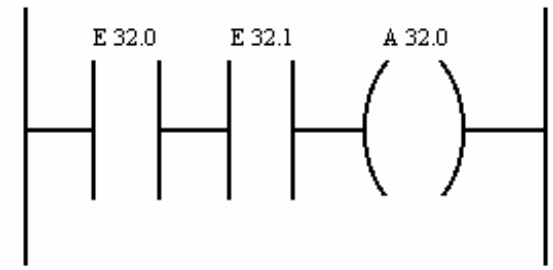
A PLC programozása

A PLC - t vagy külön erre a célra kifejlesztett programozó készülékkel vagy személyi számítógéppel a STEP-5-ös program segítségével lehet programozni.

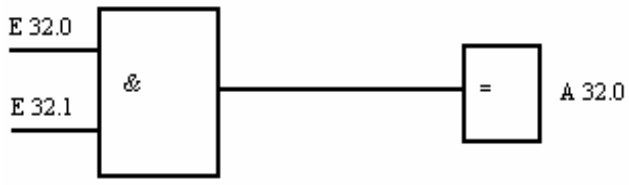
Üzemmódok : a programozó készülékkel (vagy PC - vel) kétféle üzemmódban lehet dolgozni, PLC - vel összekötve (on-line) vagy anélkül (off-line).
On-line üzemmódban a programozó készülék legelőször megvizsgálja a kapcsolatot önmaga és a PLC között. Ha a PLC nincs bekapcsolva, vagy ha nincs megfelelő kapcsolat a két eszköz között akkor az off-line üzemmód lép életbe.

A programozható logikai vezérlő összeköttetésben áll a be és kimeneti kártyái révén a jeladókkal és a végrehajtó szervekkel, amelyeket a program segítségével irányít, amit a felhasználó a STEP-5 program segítségével készített.
A PLC programozása PC segítségével történhet a SIMATIC S5 nevű program révén. A program írását egy szövegszerkesztőhöz hasonló alkalmazással valósíthatjuk meg, amelynek kezelése viszonylag körülményes, ezen problémák kiküszöbölésére alkották meg a Windows alatt futó SIMATIC programcsaládot.
A STEP-5 révén lehetőség nyílik arra hogy egy feladat meghatározását programmá alakítsunk. Ezt a programot háromféle megjelenítési módban kezelhetjük. A DIN 19239 alapján háromféle módot különböztetünk meg : létradiagram (KOP), funkcióterv (FUP) és utasításlista (AWL).

A létradiagram (KOP): a programot az USA-ban használatos szimbólumokkal jeleníti meg. A szimbólumok egy jeladó vagy végrehajtó szerv feszültség alatti illetve feszültségmentes állapotának lekérdezését szimbolizálják.



A funkcióterv (FUP): olyan módon ábrázolja a programot, hogy az alapfunkciók (pl.: VAGY művelet) ábrázolásához szimbólumokat használ, melyek bal oldalán a bemenetek, jobb oldalán pedig a kimenetek találhatók.



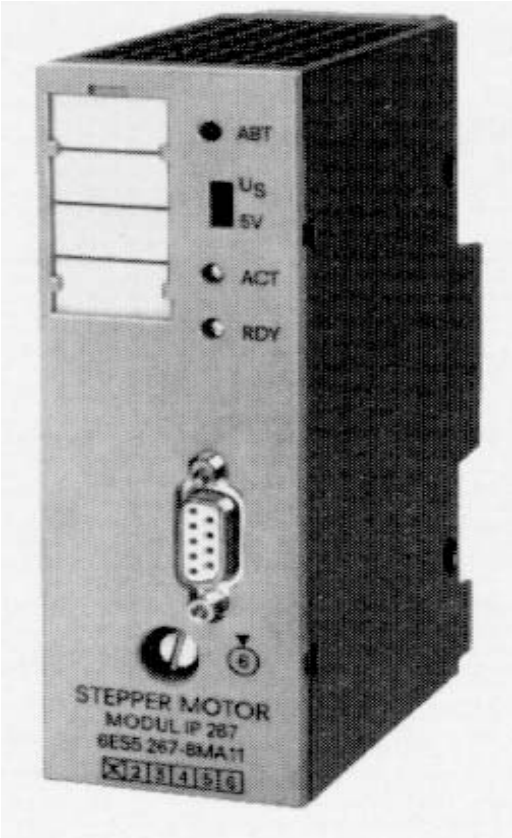
Az utasításlista (AWL) : a programot mnemotechnikai rövidítésekkel jeleníti meg. Ennél a megjelenítési módnál a STEP-5-ben elképzelhető valamennyi funkció programozható és ábrázolható.

U E 32.0
U E 32.1
= A 32.0

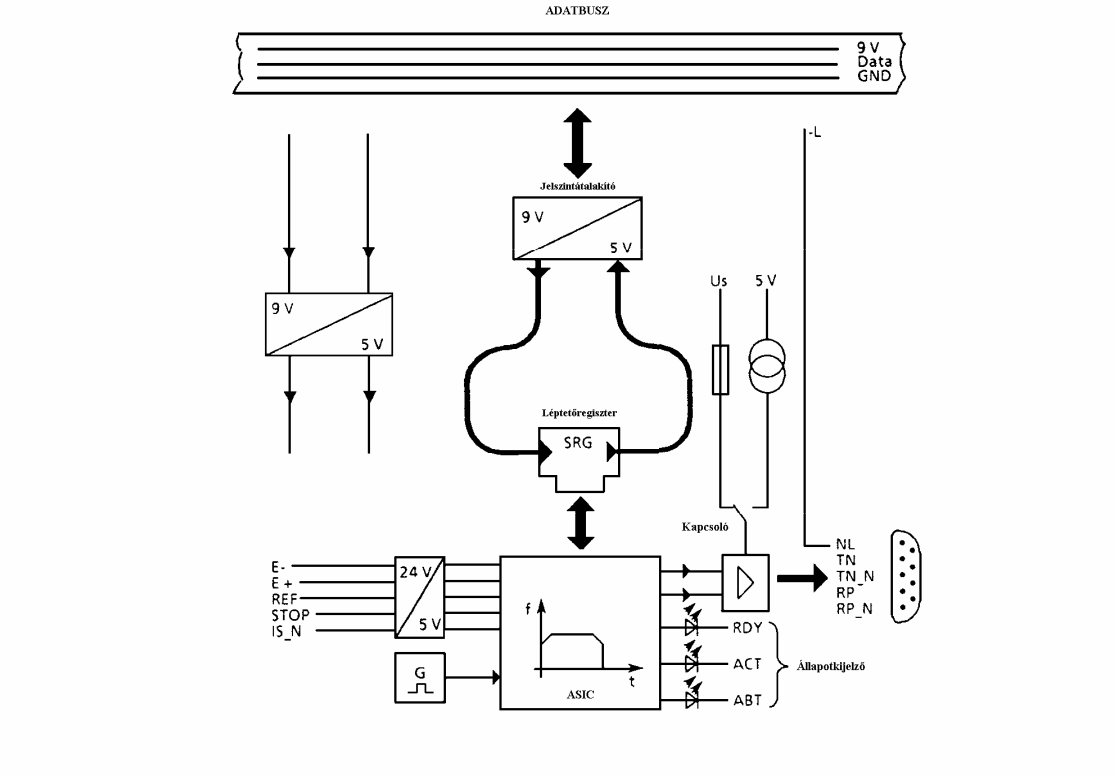
A programozásnál lehetőség van az egyes megjelenítési módokat a másikba konvertálni. AWL - ben programozott funkcióknál előfordulhat, hogy a konvertálás nem lehetséges, ezzel szemben minden KOP vagy FUP program átfordítható AWL - re.

A program a PLC memóriájában tárolódik MC 5 gépikód formájában.

Léptetőmotor vezérlése PLC - vel és IP267 - tel



A léptetőmotor vezérlését megoldhatjuk PLC - vel és egy intelligens perifériás részegységgel az IP267 - tel, amely programozható impulzussorozatokat ad a léptetőmotor számára. Az IP267 egy buszrendszeren keresztül kommunikál a PLC - vel. Az IP267-es egység együtt tud működni S5-90U-val,S5-95U/F-el és az S5-100U-val.



IP267 blokkvázlata

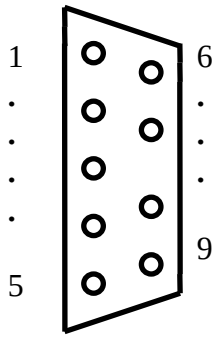
A rendszer rendelkezik egy adatbusszal ahonnan az információ egy jelszintátalakítón és egy léptetőregiszteren keresztül az ASIC áramkörbe kerül, amelynek feladata az impulzus előállítása és a kommunikációs feladatok ellátása. Az adatbusz és az ASIC között az információáramlás kétirányú. Az ASIC - hez kapcsolódik még egy jelszintátalakító(24V/5V), amelyhez különböző **bemenetek** tartoznak :

1. -
2. NL Referencia feszültség (Földelés).
3. E+ Digitális bemenet, pozitív végkapcsoló.
4. E- Digitális bemenet, negatív végkapcsoló.
5. REF Digitális bemenet, referenciakapcsoló.
6. STOP Digitális bemenet, stop végkapcsoló.
7. IS Digitális bemenet, vészkapcsoló.
8. -
9. VS Speciális feszültség (Vs) bemenet.
10. -

(A bemeneti kapcsolókat **24V** - ra kell kötni !)

Az IP267 - es kimenetei egy D csatlakozón keresztül tartanak kapcsolatot a léptetőmotorral. A **D csatlakozó pontjai** a következők :

1. -
2. TN órajel
3. -
4. TN_N inverz órajel
5. -
6. -
7. RP irányítási szint
8. RP_N inverz irányítási szint
9. NL Föld



Kijelzők :

1. ABT piros LED akkor világít ha pozícionálási feladat megszakad.
2. ACT zöld LED akkor világít ha a vezérlő aktív.
3. RDY zöld LED akkor világít ha a vezérlő egység pozícionálási feladatot kapott.

IP267 programozása :

Ahhoz hogy a léptetőmotor végrehajtson egy feladatot el kell küldeni neki egy komplett üzenetcsomagot.
Az üzenet két részből tevődik össze : konfigurálásból és pozicionálásból.

Az IP267 konfigurálása:

A konfigurációs üzenet négy byte-ból áll, amelyeket hexadecimális számként közlünk a modullal. Az első konfigurálásnál ki kell törölni a címeken lévő régi adatokat, és csak ezután küldhető el az új konfigurációs üzenet.

Byte-ok kiosztása:

0.byte	**start-stop frekvenciaszorzó (SS)**
1.byte	**1.bit EK konfigurációs bit** **4., 5. bit /KB1,KB0/** **többi bit nincs használva**
2.byte	**időintervallum gyorsításnál, lassításnál (ZI)**
3.byte	**0.bit FB0 Frekvencia** **1.bit FB1 bázisértékének** **2.bit FB2 beállítása** **többi bit nincs használva**

0.byte:

Fss(Hz)=BW(Hz) × SS × R
Fss - start-stop frekvencia
BW - Frekvencia bázisértéke
R - lassítási tényező (1 vagy 0.1)

1.byte:

EK bit = 0 E+, E- végkapcsoló 0-ra aktív
EK bit = 1 E+, E- végkapcsoló 1-re aktív
4., 5. bitnek konfiguráció esetén 0-nak kell lennie

2.byte:

a(Hz/ms)=(BW × R)÷(4 × 0.032ms × ZI)
a - Frekvencianövelés vagy csökkentés lépése

3.byte:

IP267-es frekvencia táblázata:

FB2	FB1	FB0	BW (HZ)	Gyors./Lassi. (HZ/ms) Z1=1..255	Max. frek. (kHz) G=255	Imp. Idő (µs)
0	0	0	800	6250..24,5	204	2
0	0	1	400	2125..12,25	102	3
0	1	0	200	1560..6,12	51	7
0	1	1	80	625..2,45	20,4	15
1	0	0	40	312..1,22	10,2	31
1	0	1	20	156..0,61	5,1	63
1	1	0	8	62,5..0,25	2,04	127
1	1	1	4	31,2..0,12	1,02	255

Ezek után a programban lévő egyik programsorozat első pozícionálásának konfigurációs adatait ismertetjük (FB 50 , Referenciapontból az 'A' munkagéphez) :

L KH 2802 => 0.byte = 00101000 1.byte = 00000010
T AW 64
L KH 0206 => 2.byte = 00000010 3.byte = 00000110
T AW 66

0. byte jelentése : SS = 40
1. byte jelentése : KB1 és KB0 = 0 , EK végkapcsolók 1-re aktívak.
2. byte jelentése : ZI = 2
3. byte jelentése : A táblázat alapján a BW = 8

Az IP267 pozícionálási üzenete :

0.byte	**Sebességszorzó (G)**
1.byte	**7. R bit ('1' => R = 0.1 ; '0' => R = 1)** **6. Referencia bit** **4., 5. üzemállapot bit (BA0, BA1)** **0., 1., 2., 3., útszakasz impulzusszáma (WS16 - WS19)**

2.byte	**útszakasz impulzusszáma (WS 8 - WS 15)**
3.byte	**útszakasz impulzusszáma (WS 0 - WS 7)**

0.byte:

Fa(Hz) = BW(Hz) × G × R
Fa - Kimeneti frekvencia

1.byte:

BA 1	BA 0	Értelmezés
0	**0**	**STOP**
0	**1**	**START Előre**
1	**0**	**START Hátra**
1	**1**	**Nyugalmi állapot**

Stop üzemállapot:: pozícionálási feladatot, úgy tudjuk megszakítani, hogy elküldjük a Stop parancsot a modulnak.
Start üzemállapot: ebben az esetben a modul csak akkor hajtja végre ezt az üzemmódot ha idáig nyugalmi üzemmódban volt. (előremenet esetén RP logikai 1- be kerül, hátramenet esetén RP logikai 0- ba kerül).
Nyugalmi üzemmód: egy pozícionálási feladatot többször is ki lehet küldeni az IP - nek, de csak az elsőt hajtja végre. Ha két pozícionálási feladatot egymás után egy irányba akarunk kiadni, akkor előtte nyugalmi üzemállapotba kell állítani a vezérlő egységet.

2., 3. byte :

Az útszakasz impulzusszámát határozza meg az első bájt alsó négy bitjével együtt.

Ezek után a programban lévő egyik programsorozat első pozícionálásának pozícionálási adatait ismertetjük (FB 50 , Referenciapontból az 'A' munkagéphez) :

```
L    KH 0110    =>  0.byte = 00000001    1.byte = 00010000
T    AW 64
L    KH 0050    =>  2.byte = 00000000    3.byte = 01010000
T    AW 66
```

0. byte jelentése : G = 1
1. byte jelentése : R=1 ; Start előre (BA0 = 1 ; BA1 = 0);
2., 3. byte jelentése : útszakasz impulzusszáma binárisan (decimálisan : 80);

A program futásának menete :

A program először OB21 és OB22 - t futtatja le majd az OB 1 -et hívja be.
Az OB 1 mint a program szervezőmodulja hívja be ezek után a **F**unction **B**austein - eket.

A Program AWL nyelvű leírása :

OB1

```
          O     M  50.0          ;ha fut valamelyik programsorozat és
          O     M  51.0          ;megnyomjuk a felfüggesztő gombot
          U     E  32.3    ; akkor a Z1 számláló értékéhez hozzáad
          ZV    Z  1             ; 1-et

          U     M  32.1          ;ha már másodszor is megnyomtuk a
          R     Z  1             ; gombot akkor ez a marker 1-es és ezért
                                 ; törli a számláló értékét

          L     Z  1
          L     KZ 001           ;ha a számláló értéke =1 akkor az
          !=F                    ; M32.0-s marker 1-es lesz.
          =     M  32.0

          U     M  32.0          ;ha a felfüggesztett állapot van
          S     A  32.3          ;érvényben akkor a kimenet (A32.3) 1-es


          L     Z  1
          L     KZ 002           ;ha a számláló értéke = 2 akkor az
          !=F                    ; M32.1-es marker 1-es lesz
          =     M  32.1

          U     M  32.1          ;ha megtörtént az újraengedélyezés
          R     A  32.3          ; akkor a kimenet 0-ás lesz


          U     M  53.0          ;ha az m53.0-ás marker =1 akkor ugrik
          SPB   FB 53            ;az FB 53-ba ,ahol végrehajtja a
                                 ; referenciapontba állást
NAME:     REFPONT

          U     M  53.0          ;innen nem megy tovább a prg. amíg az
          BEB                    ; FB nem törli az M53.0 - át


          UN    M  10.0          ;ha M10.0 = 0 akkor ugrik a valaszt-ba
          SPB   FB 10
NAME:     VALASZT

          UN    M  10.0          ;ha a választás megtörtént akkor megy
          BEB                    ; tovább
```

```
          U     M  60.0          ;az FB 50 és 51 állítja
          R     M  60.0
          BEB

          UN    M  32.0         ;ha nincs felfüggesztés és
          U     M  50.0         ;a PRG1 -et választottuk akkor ugrik
          SPB   FB 50           ; FB50-be

NAME:     PRG1

          U     M  50.0
          BEB

          UN    M  32.0         ;ha nincs felfüggesztés és
          U     M  51.0         ;a PRG2 -őt választottuk akkor ugrik
          SPB   FB 51           ; FB51-be

NAME:     PRG2

          U     M  51.0
          BEB

          BE                    ;Baustein Ende
```

OB21

```
          UN    M  53.0         ;ha 0 akkor 1 lesz az M53.0 ezzel
          S     M  53.0         ; biztosítva van hogy először az FB53 -
                                ; ra ugorjon
          U     M  10.0         ;ha 1 akkor 0 lesz az M10.0 így
          R     M  10.0         ; biztosítja a választást

          R     M  55.1         ;törlés
          R     M  55.0

          L     KB 0
          T     MB 50           ;törli az összes 50 és 51 -es markert
          T     MB 51

          R     M  60.0
          R     M  32.0         ;törlés
          R     M  32.1
          R     Z  1

          BE                    ;Baustein Ende
```

OB22

```
UN    M  53.0
S     M  53.0

U     M  50.0          ;ha 1 akkor 0 lesz az M50.0 és az M51.0
R     M  50.0          ;hogy a az kikapcsolás majd újraindítás
U     M  51.0          ; után ne folytassa az elkezdett feladatot
R     M  51.0

BE                     ;Baustein Ende
```

FB10

```
NAME:    VALASZT
                       ;kapcsoló megfelelő beállításával majd a
                    ;nyomógomb megnyomásával a
                 ; megfelelő marker 1 értéket vesz fel
                       ;a programokhoz kijelzés is tartozik
UN    E  32.1
R     A  32.1

UN    E  32.2
R     A  32.2


U     E  32.1
S     A  32.1

U     E  32.2
S     A  32.2

U     E  32.1
U     E  32.0
S     M  50.0
S     M  10.0

U     E  32.2
U     E  32.0
S     M  51.0
S     M  10.0

U     M  50.0
R     M  51.0

U     M  51.0
R     M  50.0
```

```
      BE                          ;Baustein Ende


   FB50

NAME: PRG1

      U     M  55.0               ;a program végén indul ez az időzítés
      L     KT 005.2              ;és 5 s -ig tart ami alatt biztosan lefut az
      SE    T  1                  ;utolsó pozícionálás és ez biztosítja hogy
                                  ;ne lehessen ráindítani (R M10.0)

      U     T  1
      R     M  10.0
      R     M  55.0
      R     T  1

      U     M  55.0               ;amíg le nem jár az idő ennél a pontnál
                                  ;visszaugrik az OB1 -be onnét pedig BEB
                                  ;mindig megint ide
                                  ;BEB = Feltételes visszaugrás

      U     M  50.1
      SPB   =M001
      S     M  50.1
      L     KH 0000               ;első konfigurálásként kötelezően 0-t
      T     AW 64                 ; töltünk át
      BEA

M001: U     M  50.2               ;minden címkével ellátott rész csak
SPB   =M002                       ; egyszer hajtódik végre amikor az előző
      S     M  50.2               ; rész már végrehajtódott ezt pedig
      S     M  60.0               ; onnan tudja hogy minden részben
      L     KH 2802               ; beállít egy markert amit utána
      T     AW 64                 ; megvizsgálhat
      L     KH 0206               ; konfigurációs adatok
      T     AW 66
      BEA                         ;BEA = Feltétel nélküli visszaugrás

M002: U     M  50.3
      SPB   =KSL1
      S     M  50.3
      S     M  60.0
      S     M  56.0
      L     KH 0110               ;előrepozícionálás az 'A' munkapontba
```

```
      T     AW 64          ;01=> 64-es címre; 10=>65; 00=>66;
      L     KH 0050        ; 50=>67
      T     AW 66
      BEA

KSL1: U     M  56.0        ;minden pozicionálás után késleltet
      L     KT 005.2       ;(amíg a munkagép elvégzi feladatát)
      SE    T  3

      U     T  3
      R     M  56.0
      R     T  3

      U     M  56.0
      BEB

      U     M  50.4
      SPB   =M003
      S     M  50.4
      S     M  60.0        ;semleges állapot kiküldése
      L     KH 0130
      T     AW 64
      BEA

M003: U     M  50.5
      SPB   =KSL2
      S     M  50.5
      S     M  60.0
      S     M  56.1        ;előrepozícionálás a 'B' munkapontba
      L     KH 0110
      T     AW 64
      L     KH 0050
      T     AW 66
      BEA

KSL2: U     M  56.1
      L     KT 005.2
      SE    T  4

      U     T  4
      R     M  56.1
      R     T  4

      U     M  56.1
      BEB
```

```
        U     M  50.6
        SPB   =M004
        S     M  50.6
        S     M  60.0          ;semleges állapot kiküldése
        L     KH 0130
        T     AW 64
        BEA

M004:   U     M  50.7
        SPB   =KSL3
        S     M  50.7
        S     M  60.0
        S     M  56.2          ;előrepozícionálás a 'C' munkapontba
        L     KH 0110
        T     AW 64
        L     KH 0050
        T     AW 66
        BEA

KSL3:   U     M  56.2
        L     KT 005.2
        SE    T  5

        U     T  5
        R     M  56.2
        R     T  5

        U     M  56.2
        BEB


        U     M  150.0
        SPB   =M005
        S     M  150.0
        S     M  60.0          ;semleges állapot kiküldése
        L     KH 0130
        T     AW 64
        BEA

M005:   U     M  150.1
        SPB   =KSL4
        S     M  150.1
        S     M  56.3
        S     M  60.0
        L     KH 0110          ;előrepozícionálás a 'kiviteli'
        T     AW 64                  ; munkapontba
```

```
        L     KH 0050
        T     AW 66
        BEA


KSL4:   U     M  56.3
        L     KT 005.2
        SE    T  6

        U     T  6
        R     M  56.3
        R     T  6

        U     M  56.3
        BEB

        U     M  150.2
        SPB   =M006
        S     M  150.2
        S     M  60.0          ;semleges állapot kiküldése
        L     KH 0130
        T     AW 64
        BEA


M006:   U     M  150.3
        SPB   =M007
        S     M  150.3
        S     M  60.0
        S     M  55.0
        L     KH 0120          ;vissza kiinduló pontba
        T     AW 64
        L     KH 0140
        T     AW 66
        BEA


M007:   L     KB 0
        T     MB 50            ;törli ahasznált markereket

        L     KB 0
        T     MB 150

        R     M  60.0
        BE
```

FB51

```
NAME:    PRG2

     NETZWERK 1                    ;felépítése hasonlít az FB50-eshez

          U    M  55.1
          L    KT 005.2
          SE   T  2

          U    T  2
          R    M  10.0
          R    M  55.1
          R    T  2

          U    M  55.1
          BEB



          U    M  51.1
          SPB  =M001
          S    M  51.1
          L    KH 0000
          T    AW 64
          BEA

M001:     U    M  51.2
          SPB  =M002
          S    M  51.2
          S    M  60.0
          L    KH 2802               ;konfigurációs adatok elküldése
          T    AW 64
          L    KH 0206
          T    AW 66
          BEA

M002:     U    M  51.3
          SPB  =KLT1
          S    M  51.3
          S    M  60.0
          S    M  66.0               ;előrepozícionálás az 'A' munkapontba
          L    KH 0110
          T    AW 64
          L    KH 0050
          T    AW 66
          BEA

KLT1:     U    M  66.0
```

```
      L     KT 005.2
      SE    T  7

      U     T  7
      R     M  66.0
      R     T  7

      U     M  66.0
      BEB


      U     M  51.4
      SPB   =M003
      S     M  51.4
      S     M  60.0          ;semleges állapot kiküldése
      L     KH 0130
      T     AW 64
      BEA

M003: U     M  51.5
      SPB   =KLT2
      S     M  51.5
      S     M  60.0
      S     M  66.1
      L     KH 0110          ;előrepozícionálás a 'B' munkapontba
      T     AW 64
      L     KH 0050
      T     AW 66
      BEA

KLT2:       U     M  66.1
      L     KT 005.2
      SE    T  8

      U     T  8
      R     M  66.1
      R     T  8

      U     M  66.1
      BEB


      U     M  51.6
      SPB   =M004
      S     M  51.6
      S     M  60.0          ;semleges állapot kiküldése
      L     KH 0130
      T     AW 64
```

```
        BEA


M004:   U       M  51.7
        SPB     =KLT3
        S       M  51.7
        S       M  66.2
        S       M  60.0
        L       KH 0110             ;előrepozícionálás a 'C' munkapontba
        T       AW 64
        L       KH 0050
        T       AW 66
        BEA


KLT3:           U       M  66.2
        L       KT 005.2
        SE      T  9

        U       T  9
        R       M  66.2
        R       T  9

        U       M  66.2
        BEB

        U       M  151.0
        SPB     =M005
        S       M  151.0
        S       M  60.0             ;semleges állapot kiküldése
        L       KH 0130
        T       AW 64
        BEA


M005:   U       M  151.1
        SPB     =KLT4
        S       M  151.1
        S       M  60.0
        S       M  66.3
        L       KH 0120             ;vissza a 'B' munkapontba
        T       AW 64
        L       KH 00A0
        T       AW 66
        BEA


KLT4:           U       M  66.3
        L       KT 005.2
        SE      T  10
```

```
      U     T  10
      R     M  66.3
      R     T  10

      U     M  66.3
      BEB


      ***

NETZWERK 2

      U     M  151.2
      SPB   =M006
      S     M  151.2
      S     M  60.0                ;semleges állapot kiküldése
      L     KH 0130
      T     AW 64
      BEA


M006: U     M  151.3
      SPB   =KLT5
      S     M  151.3
      S     M  60.0
      S     M  66.4
      L     KH 0110                ;előrepozícionálás a 'kiviteli'
      T     AW 64                        ; munkapontba
      L     KH 00F0
      T     AW 66
      BEA


KLT5:       U     M  66.4
      L     KT 005.2
      SE    T  11

      U     T  11
      R     M  66.4
      R     T  11

      U     M  66.4
      BEB


      U     M  151.4
      SPB   =M007
      S     M  151.4
      S     M  60.0                ;semleges állapot kiküldése
```

```
          L     KH 0130
          T     AW 64
          BEA

M007:     U     M  151.5
          SPB   =M008
          S     M  151.5
          S     M  60.0
          S     M  55.1
          L     KH 0120          ;vissza kiindulópontba
          T     AW 64
          L     KH 0140
          T     AW 66
          BEA

M008:     L     KB 0
          T     MB 51

          L     KB 0             ;törli a használt markereket
          T     MB 151

          R     M  60.0

          BE


     FB53

NAME:     REFPONT                ;referenciapontba viszi vissza a motort

          U     M  66.0
          L     KT 002.2         ;késleltetés hogy legyen ideje visszatérni a
          SE    T  22            ;referenciapontba a motornak és
          U     T  22            ;eközben ne lehessen ráindítani
          R     M  66.0
          R     T  22

          U     M  66.0
          BEB

          U     M  53.1
          SPB   =M01             ;0-t tölt át
          S     M  53.1
          L     KH 0000
          T     AW 64
          BEA
```

```
M01 : U     M  53.2
      SPB   =M02
      S     M  53.2          ;konfigurálási adatok áttöltése
      L     KH 0502
      T     AW 64
      L     KH 1005
      T     AW 66
      BEA

M02 : U     M  53.3
      SPB   =M03
      S     M  53.3
      L     KH 105F
      T     AW 64
      L     KH FFFF
      T     AW 66
      BEA

M03 : U     M  53.4
      SPB   =M04
      S     M  53.4
      L     KH 586F
      T     AW 64
      L     KH FFFF
      T     AW 66
      BEA

M04 : U     M  53.5
      SPB   =M05
      S     M  53.5
      L     KH 2820
      T     AW 64
      L     KH 0300
      T     AW 66
      BEA

M05 : U     M  53.6
      SPB   =M06
      S     M  53.6
      S     M  66.0
      L     KH 0150
      T     AW 64
      L     KH 4000
      T     AW 66
```

```
          BEA


M06 :     L       KB 0
          T       MB 53                   ;felhasznált markerek törlése

          BE
```

Felhasznált irodalmak

- SIEMENS SIMATTIC S5/PC/TI505 AUTOMATION SYSTEM
- Automatisierungsgeraat SIMATIC S5-115U
- Helmut Moczala: Törpe villamos motorok és alkalmazásaik
- Habiger - R.Schönfeld - W. Stock - D. Zenkl : Villamos hajtások kézikönyve
- www.jpte.hu (Janus Pannonius Tudományegyetem) távoktatás segédanyag (Léptetőmotorok fajtái, működése)
- Douglas W. Jones - egyetemi jegyzet www.cs.uiowa.edu (University of Iowa)
- Sas Tibor - Vezérlések párhuzamos porton keresztül
- László József - Perifériák programozása
- Abonyi Zsolt - PC hardver kézikönyv