	Mechatronika, Optika és Gépészeti Informatika Tanszék	M2
	Léptetőmotor vezérlése	D528
A mérés célja:	Léptetőmotor vezérlése bipoláris, unipoláris esetben Párhuzamos port kezelése Free-Pascal vagy LabVIEW fejlesztői környezetben	
A mérés során felhasznált eszközök:	<ol style="list-style-type: none"> 1. Léptetőmotor 2. Párhuzamos porthoz illesztett áramkör 3. Személyi számítógép 4. Tápegység 	
A mérés során elvégzendő feladatok:	<ol style="list-style-type: none"> 1. Léptetőmotor egész és fél lépéses üzemmódjának elkészítése 2. Adott lépésszám elvégzésére alkalmas program elkészítése 3. Ping-Pong program elkészítése 	

1. A feladatról

Ebben a mérésben a számítógép nyomtató portjáról fogjuk vezérelni a léptetőmotort egy „interfészen” keresztül. Bevezető kísérletként a léptetőmotort a PORT.EXE nevű általános port-indikáló programmal vezéreljük lépésenként. Először kipróbáljuk mi is történik a megfelelő portcím (\$378) egyes biteinek állítgatásával.

E kísérlet közben határozzuk meg a léptetőmotor léptetéséhez szükséges bitkombinációkat, valamint melyik portcím melyik bitein láthatóak az opto- kapuk jelei, ha az opto- kapukba befordítjuk a zászlót. Mivel PASCAL-ban az egyes portcímekre csak egy egész BYTE-ot lehet írni, ezért a kísérletezés közben meghatározott bináris bitkombinációkat, át kell számolni decimális vagy hexadecimális számmá. Ezeket fel kell jegyezni papírra, hogy később a programban használhassuk. Ezután fogjuk megírni Free-Pascal-ban vagy LabVIEW-ban az előre elkészített programváz alapján a léptetőmotor vezérlőprogramját.

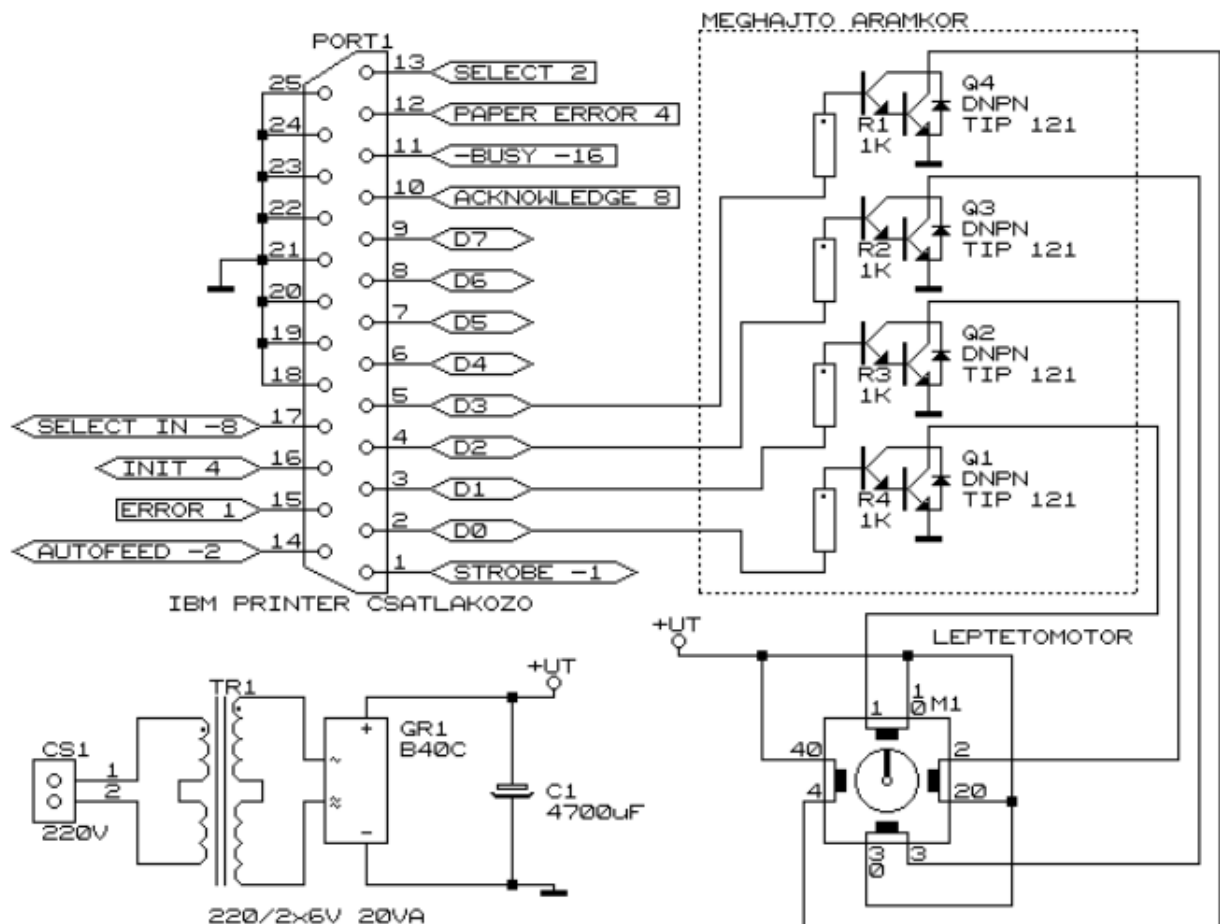
2. Az áramkör leírása

Ehhez a léptetőmotorhoz alkalmazott meghajtó áramkör az 1. ábrán látható. A számítógép és a meghajtó közti interfészt a számítógép printerportja alkotja.

A csatlakozópontok elnevezései természetesen a printer funkcióit jelentik. Számunkra ezeknek az elnevezéseknek nincs jelentőségük, csak a konvenció kedvéért tartottuk meg. Az adatvezetékek : D0 .. D7 – írhatók. Ezzel vezéreljük jelen esetben a motor egyes tekercseit.

A státuszvezetékek : SEL; PE; BUSY; ACK – csak olvashatók. Ide valamelyik két vezetékre van kötve a két opto-kapu, amivel a motor két szöghelyzetét érzékelhetjük.

A parancsvezetékek : SEL-IN; INIT; ERR; AUTOFEED – írható-olvasható vezetékcsoportok. Ezeket most nem használjuk.



1. ábra A léptetőmotor illesztése a párhuzamos porthoz

3. A Free-Pascal környezetéről

A párhuzamos port kezelését WindowsXP rendszerben az UserPort program segítségével tudjuk végrehajtani, amely engedélyezi a portok I/O műveleteinek elérését az operációs rendszerben. Ez a Free-Pascal port-műveleteihez szükséges.

A Free-Pascalban a portokat a következőképpen tudjuk olvasni vagy írni:

- Az olvasás az `inportb` függvénnyel történik: {go32 unitban található}

```
function inportb(
  port: Word
):Byte;
```

- Az írás az `outportb` eljárással történik: {go32 unitban található}

```
procedure outportb(
  port: Word;
  data: Byte
);
```

Példa a használatra:

```
uses      crt,
          go32;          {portkezeléshez}

var in, out: integer;

begin
    out:=8;
    outportb($378, out); {kiíratás}
    delay(50);
    in:=inportb($379);   {beolvasás}
end.
```

A felhasználandó programváz:

```
uses go32,          {a portkezeléshez szükséges}
    crt;

const pdata      : integer=$378; {printerport adat-vezetékeinek címe}
      pstat      : integer=$379; {printerport státusz-vezetékeinek címe}

var kesleltet : integer;          {léptetőmotor két lépése közt eltelt idő}

{-----vvvv-----meghajtó rész-----vvvv-----}

    {konstansok és változók a léptetőmotor vezérléséhez}
const Tmax=7;

      MASZK1 = ■;          {csak a 4. vezetékek olvasását engedi}
      MASZK2 = ■;          {csak a 7. vezetékek olvasását engedi}

      ELORE  = 1;          {léptetőmotor léptetésének iránya}
      HATRA  = -1;         {léptetőmotor léptetésének iránya}
      KI     = 0;          {léptetőmotor gerjesztése kikapcsolva}
      BE     = 1;          {léptetőmotor gerjesztése bekapcsolva}

var lep_stat : integer;          {lépéshelyzet regisztere}
    lep      : array[0..Tmax] of byte; {lépés-szekvencia}

procedure init;
begin
    kesleltet:=8;                {miliszekundumban minimum a 3ms}
    lep_stat:=0;                  {alaphelyzetbe hozzuk a léptetőregisztert}
    lep[0]:=■;
    lep[1]:=■;
    lep[2]:=■;
    lep[3]:=■;
    lep[4]:=■;
    lep[5]:=■;
    lep[6]:=■;
    lep[7]:=■;
end;

{ez a függvény igazat ad vissza, ha a zászló az 1-es optokapuban van}
function optol:boolean;
begin
    optol:=(inportb(pstat) and MASZK1)=0;
end;
```

```

{ez a függvény igazat ad vissza, ha a zászló a 2-es optokapuban van}
function opto2:boolean;
begin
  opto2:=(inportb(pstat) and MASZK2)=0;
end;

{ez az eljárás a megadott irányban lépteti 1-et a léptetőmotort}
procedure leptet(irany:integer);
begin
  lep_stat:=lep_stat+irany;
  if lep_stat>Tmax then
    lep_stat:=0;
  if lep_stat<0 then
    lep_stat:=Tmax;
  outportb(pdata,lep[lep_stat]);           {a motor léptetése}
  delay(kesleltet); {itt lehet sima késleltetés vagy valamilyen egyéb
                                     hasznos tevékenység pl. mérés}
end;

{ez az eljárás elforgatja a tárcsát opto1-ig}
procedure gohome;
begin
  while opto1 do begin
    leptet(HATRA);
    delay(kesleltet);
  end;
end;

{ez az eljárás a motor gerjesztését kapcsolja ki, illetve be}
procedure tartas(kapcs:integer);
begin
  if(kapcs=BE) then
    outportb(pdata,lep[lep_stat])           {az aktuális bit-et kiírja}
  else
    outportb(pdata,0);                     {az összes bit 0}
end;
{-----^^^-----meghajtó rész vége-----^^^-----}

{-----vvv-----felhasználói rész-----vvv-----}
{tegyen meg a léptetőmotor 1 teljes körülfordulást}
procedure feladat1;
begin
  {a feladat megvalósítása}
end;

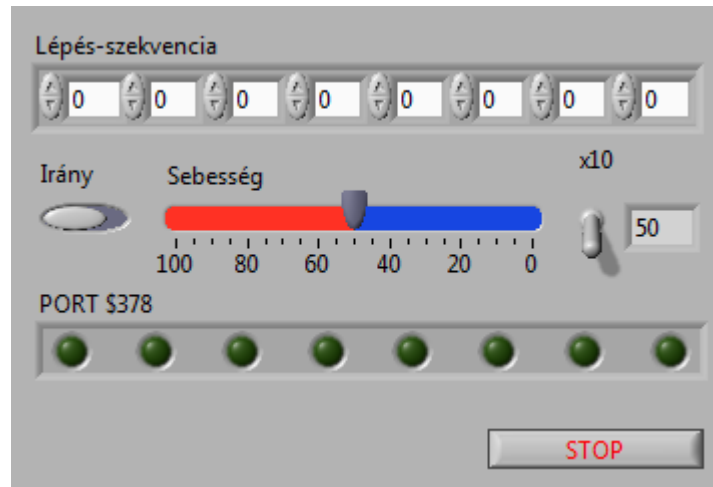
{a két opto-kapu között ping-pong}
procedure feladat2;
begin
  {a feladat megvalósítása}
end;
{-----^^^-----felhasználói rész vége---^^^-----}

{innen hívjuk meg a szükséges eljárásokat}
begin
  init;           {ne írjuk át}
  gohome;
  feladatX;       {az X-et a megfelelő (1-,vagy 2-re kell átírni)}
  tartas(KI);     {ne írjuk át}
end.

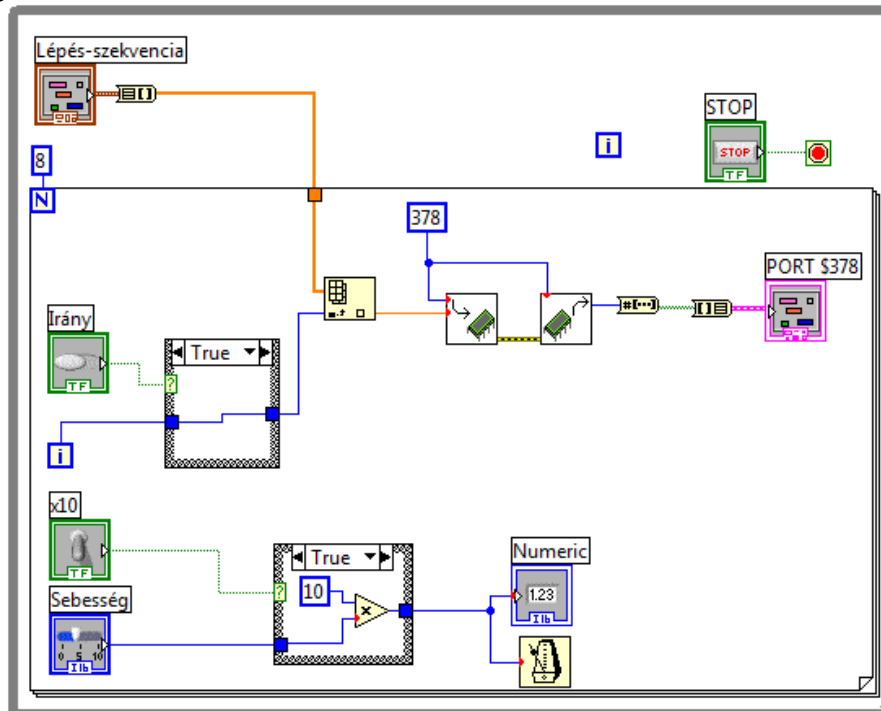
```

3. A LabVIEW környezetéről

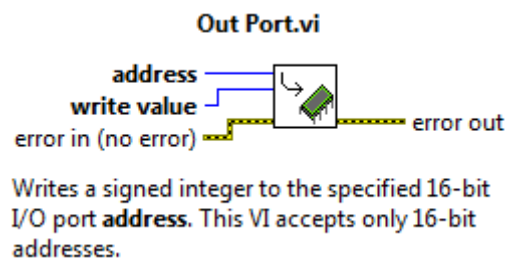
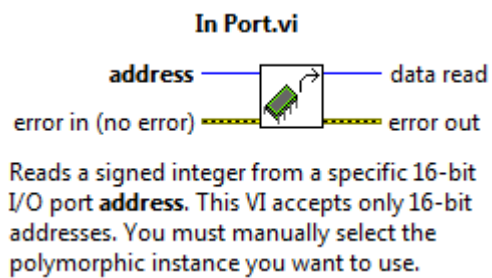
Front Panel:



Blokk diagram:



A portműveleteket a LabVIEW-ban az In Port és Out Pot VI-ok segítségével végezhetjük el.



Mérési feladat:

- *Írja fel, hogy milyen bitkombinációk szükségesek az egész és fél lépéses üzemmódokhoz!*
- *Töltse ki a lépés-szekvencia tömböt a megfelelő kombinációval!*
- *Készítsen el egy olyan programot, amely 1 teljes körfordulást végez, és utána elengedi a léptetőmotor tekercseit!*
- *Határozza meg, hogy mely vezetékeken találhatóak meg az opto-kapuk!*
- *Készítsen el egy olyan programot, amely a két opto-kapu között ide-oda jár leállításig!*

A méréshez felhasználható Free-Pascal vagy LabVIEW forrásfájlok, valamint a PORT-kezelő alkalmazás az Asztalon található M2 nevű mappában vannak előkészítve.

Senki ne írja felül a mintafájlokat! Mindenki készítsen egy mappát magának, amelybe másolja be a szükséges fájlokat!