

Tallinna Ülikool  
Matemaatika-loodusteaduskond  
Informaatika osakond

Janno Pugi

# Õppematerial arvuti poolt juhitud lisaseadmete loomiseks

Bakalaureusetöö

Juhendaja: Jaagup Kippar

Autor: ..... „“ ..... 2006. a.  
Juhendaja: ..... „“ ..... 2006. a.  
Osakonna juhataja: ..... „“ ..... 2006. a.

Tallinn 2005



# 1. Sisukord

1. Sisukord.....	3
2. Sissejuhatus.....	4
3. Parallel ehk LPT port.....	5
3.1 Ülevaade.....	5
3.2 Aadressid ja registrid.....	8
3.3 LPT monitorid.....	11
3.4 Userport.....	13
4. Programmeerimiskeeled.....	15
4.1 Java.....	15
4.1.1 Kompilaator.....	15
4.1.2 Jnpout32_v10 teek.....	15
4.1.3 Näiteprogramm.....	16
4.2 C++.....	17
4.2.1 Kompilaator.....	17
4.2.2 Inpout32 pakett.....	17
4.2.3 C++ näiteprogramm.....	17
5. Teemad.....	18
5.1 Valgusdiodide vilgutamine.....	18
5.2 Sisendite kasutamine.....	25
5.3 Sisendid ja väljundid koos.....	29
5.4 Välist toiteallikat kasutavate seadmete lülitamine.....	31
5.5 Võrgutoidet kasutavate seadmete lülitamine.....	34
5.6 Välist toiteallikat kasutavate seadmete lülitamine mikroskeemiga.....	35
5.7 Steppermootori juhtimine.....	38
5.7.1 Ülevaade steppermootoritest.....	38
5.7.2 Unipolaarse steppermootori juhtimine.....	39
5.7.3 Juhtprogrammid.....	46
5.8 LCD-mooduli juhtimine.....	47
5.8.1 Ülevaade LCD'st.....	47
5.8.2 LCD-ekraani juhtimine.....	49
5.8.3 Juhtprogrammid.....	56
5.9 Otsilloskoop.....	58
5.10 Täienduseks.....	62
6. Kokkuvõte.....	63
7. Summary.....	64
8. Kasutatud allikad.....	65

## 2. Sissejuhatus

Arvutid ja muud tehnilised lahendused, mis aitavad meil igapäevast tööd teha, on muutunud tavalisteks tööriistadeks. Erinevatel inimestel on erinevad oskused ja teadmised arvutist. Tavakasutaja teadmised piirduvad enamasti mingite kindlate programmide kasutamise oskamisega, kuid spetsialistid peaksid tundma süsteeme süvitsi.

Tallinna Ülikooliski on informaatika eriala, kus saab õppida arvutite hingeelu, programmeerimist ja tutvuda riistvaraga, kuid kahjuks on see küllaltki pinnapealne ja teoreetiline. Saadakse hägune ettekujutus, kuidas liiguvad andmed arvutis ja kuidas neid töödeldakse. Enamik üliõpilasi, kes ei tunne arvutite sügavamat hingeelu, sellega lepib ja lõpetab ülikooli edukalt. Kuid iga endast lugupidav informaatik peaks teadma ja suuremat tundma huvi, kuidas töötavad süsteemid madalamal tasemel – elektrooniliselt.

Loodud õppematerjal on suunatud eelkõige huvilistele, kellele meeldib eksperimenteerida ja katsetada ning tutvuda süsteemide ülesehitusega. Neile, kes ei piirduks väljapakutud ülesannete lahendamise ja vaid püstitaks ka ise endale ülesandeid ja eesmärke ning näeks vaeva nende saavutamise ja – ainult nii on võimalik olla hingega asja juures.

Käesolev õppematerjal keskendub eelkõige LPT-porti ühendatavate lisaseadmete loomisele. Olgugi, et LPT-port on iseenesest hakanud juba vaikselt välja surema, on teda märksa lihtsam ja arusaadavam juhtida, kui näiteks USB-porti. Seega oleks erinevate seadmetevaheliste ühendustega tutvumiseks just hea printeriporti pruukida. Paralleelpordi saab puudumise korral kerge vaevaga arvutile lisada ja sellest tulenev ühekordne paarisajakroonine investeering ei tohiks selle tehnikaõppe juures lootusetult suur olla. Lisaks kõigele muule, on ka printeriporti ühendatavate seadmete detailid odavad ja lihtsad, mis teeb algajate elu kergemaks.

Algust tehakse väga lihtsate skeemidega, mis peaksid olema jõukohased ka puuduvate elektroonikakogemustega tudengitele. Õppematerjali läbitöötaval tudengil võiksid siiski olla mõningad varasemad teadmised elektroonikast ja skeemide lugemisest ning huvi otsida ise lisa või täiendavaid teadmisi nii internetist kui ka raamatutest. Programmeerimisalaseid teadmisi läheks samuti tarvis.

Programmeerimisülesanded on lahendatud Javas ja skeemid koostatud programmiga Eagle. Kõik mainitud ja kasutatud programmid on vabavaralised.

## 3. Parallel ehk LPT-port<sup>1</sup>

### 3.1 Ülevaade

LPT-porti (*LPT – Line Printer Terminal*) on kasutatud läbi aegade, et arvutiga ühendada<sup>2</sup>:

- printereid
- skännereid
- CD-kirjutajaid
- väliseid kõvakettaid
- võrguadaptereid
- lindiseadmeid

Paralleelport (*DB25*) leiutati algselt IBM'i poolt printeri ühendamiseks arvutiga, kui too firma personaalarvutit arendas<sup>3</sup>. Kuid tänapäeval on LPT-port rohkem unustustehõlma vajunud ning kasutusel on juba palju kiiremad ja mugavamad standardid nagu näiteks USB(Universal Serial Bus), firewire jne... Võrreldes USB-pordiga on LPT-porti märksa lihtsam juhtida ning alustada võikski just sellega tutvumisega.

Kutsutakse just paralleelpordiks, kuna andmevahetus toimub baidi kaupa (erinevalt USB-st ja jadapordist): 8 bitti edastatakse paralleelselt ja üheaegselt. Seetõttu on andmeedastus jadapordist kiirem, kuid kaabli pikkus on piiratud 5-10 meetriga ning kaabel on ebamugavalt paks, sest sisaldab 25 juhet.

Standard paralleelport on võimeline edastama 50 kuni 100 kilobaiti andmeid sekundis.

---

1 Epanorama.net - [http://www.epanorama.net/circuits/parallel\\_output.html](http://www.epanorama.net/circuits/parallel_output.html) (11.01.2006)

2 Howstuffworks.com - <http://computer.howstuffworks.com/parallel-port.htm> (11.01.2006)

3 Epanorama.net - [http://www.epanorama.net/circuits/parallel\\_output.html](http://www.epanorama.net/circuits/parallel_output.html) (21.04.2006)

Vaatamata pordi üldisele unustusehõlma vajumisele on seda siiski väga hea ja mugav kasutada elektroonikaga tutvaval tudengil endatehtud lisaseadmete ühendamiseks arvutiga. Võimalused on pea piiramatud, kuid erinevate katsetuste käigus peab rangelt jälgima, et skeemides lühiseid ja muid vigu ei tekiks, sest nii võib arvuti LPT-pordi ära rikkuda.

Targem oleks muretseda mõni vanem ja odavam I/O kaart, mis sisaldab lisa LPT-porti ning kui sellega peaks midagi juhtuma, siis ei pea tervet emaplaati ära vahetama, et pordi jälle korda saaks.

LPT-pordist voolu võttes tuleks arvestada sellega, et voolutugevus ei tohiks minna kõrgemaks kui mõned kümned milliamprid (~ 10-15mA), sest muidu võib pordi läbi põletada.

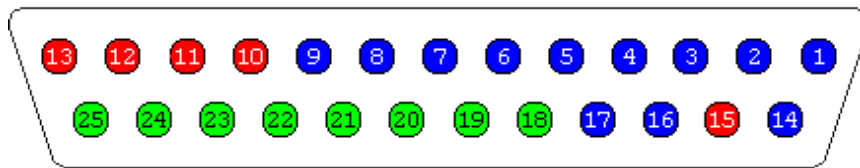
**Tabel 1 kirjeldab LPT-pordi 25 väljaviiku<sup>4</sup>:**

<i>Viiik</i>	<i>Nimi</i>	<i>Selgitus</i>
1	Strobe	Strobeerimisimpulss, mis ütleb printerile, millal on 8 bitti valmis lugemiseks. Omab pinget vahemikus 2,8 – 5 volti, kuid langeb 0,5 voldi peale, kui arvuti saadab baidi jagu infot. Pingelangusest saab printer aru, millal on õige aeg andmeid lugeda.
2-9	Data0 - Data1	8 bitti ühe andmebaidi jaoks. Vastava väljaviigu pinge ~5 volti tähistab loogilist väärtust 1 ja pinge 0 volti tähistab loogilist väärtust 0.
10	Acknowledge	Kinnitus, et andmed võeti korralikult vastu. Analoogne viiguga 1, kuid vastust printerilt arvutile.
11	Busy	Printer hõivatud(mälu täis või toimub andmete töötlemine). Arvuti ootab uute andmete saatmisega.
12	Paper-Out	Printeril paber otsas.
13	Select	Teatatakse arvutile, et printer on küljes.
14	Auto-Linefeed	Palutakse printeril trükkida tühi rida.
15	Error	Vea korral väärtus loogiline 0 ja arvuti lõpetab andmete saatmise.
16	Initialize	Algväärtustab printeri.
17	Select-Printer	Valib printeri
18-25	Ground	Maandus.

*Tabel 1: LPT-pordi väljaviigud selgitustega*

4 Howstuffworks.com - <http://computer.howstuffworks.com/parallel-port1.htm> (11.01.2006)

Joonisel 1 näeme LPT-pordi sisendeid, väljundeid ja maandusi <sup>5</sup>:



Joonis 1: LPT-port

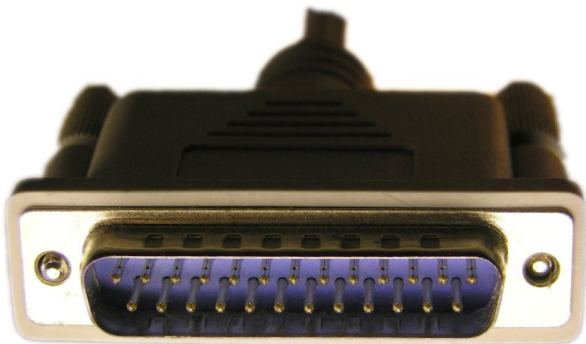
väljund: 12 viiku, joonisel sinised

sisend: 5 viiku, joonisel punased

maa: 8 viiku, joonisel rohelised

On olemas kahte tüüpi LPT pordi pistikuid. Paljudel printerite juhtmetel on arvutipoolses otsas D-Type pistik ja printeripoolses Centronics pistik.

#### D-Type 25:



#### Centronics 34<sup>6</sup>:



Pistikute lähemal vaatlemisel leiab väljaviikude juurest nummerduse, mis teeb hõlpsaks vajaliku viigu leidmise.

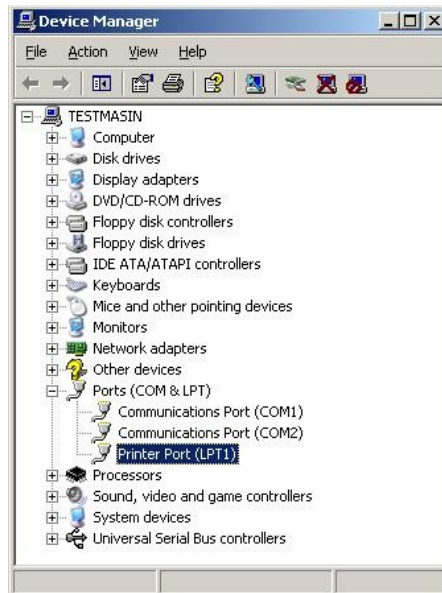
Tabelis 2 lehekülj 9 on pistikute viikude kirjeldused ja võrdlused.

<sup>5</sup> Linuxgazette.net - [http://linuxgazette.net/118/misc/chong/parallel\\_port\\_diagram.jpg](http://linuxgazette.net/118/misc/chong/parallel_port_diagram.jpg) (05.01.2006)

<sup>6</sup> Wikipedia.org - <http://de.wikipedia.org/wiki/Bild:Centronics.jpg> (05.01.2006)

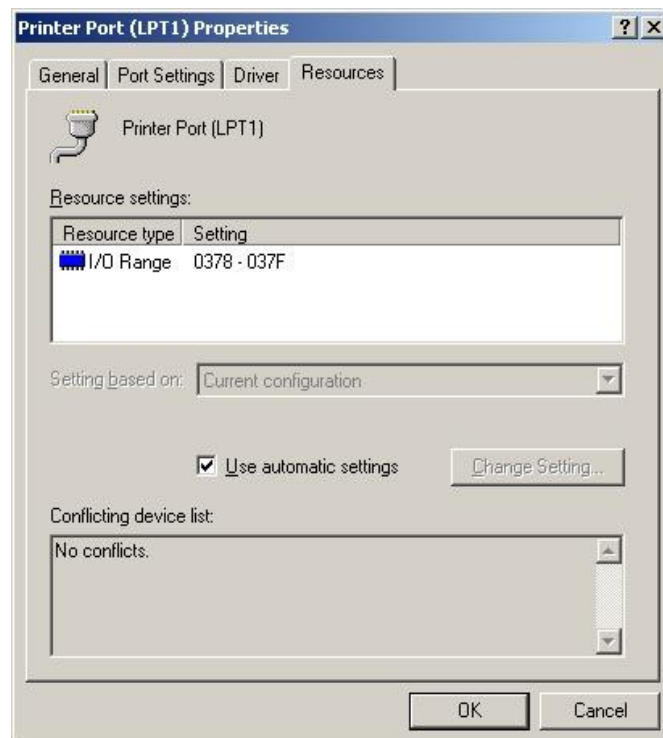
### 3.2 Aadressid ja registrid

Saamaks teada, mis aadresse LPT-port kasutab, võib avada Windows XP all *Device Manageri*(Kuvapildistus 1), kus portide loetelu all asub ka printeriport (*Printer Port (LPT 1)*):



Kuvapildistus 1: Device Manager

*Printer Port (LPT1) Properties*'is *Resource settings* all on näha *I/O Range*, mis märgib sisend/väljund ala ehk aadresside piirkonda, milleks antud näites(Kuvapildistus 2) on 0378 – 037F.



Kuvapildistus 2: Printer Port (LPT1) Properties

Tüüpilise personaalarvuti LPT1 baas-aadress on 0x378 ja LPT2 baas-aadressiks 0x278<sup>7</sup>.

Et kirjutada või lugeda andmeid väljaviikudest, mis on mõeldud kas andmete edastuseks, staatuse kuvamiseks või seadme juhtimiseks, tuleb seda teha kasutades registreid. Ühendused registrite ja väljaviikude vahel ei ole füüsilised vaid virtuaalsed.

### Registreid on printeripordil 3<sup>8</sup>:

Andmeregister (*Data Register*) – ühenduses andmeviikudega

Staatuseregister (*Status Register*) – ühenduses staatuseviikudega

Juhtregister (*Control Register*) – ühenduses kontrollviikudega

Näiteks saates andmeid andmeregistrisse (0x378), ilmuvad need andmed ainult väljaviikudel (LPT2-LPT9). Soovides juhtida väljaviikusid, mis asuvad juhtregisris, tuleb kasutada registri aadressi 0x37A.

Kui on teada LPT pordi baasaadress, mis on ühtlasi andmeregistri aadress, saab arvutada ülejäänud registrite aadressid:

Andmeregister = baas-aadress

Staatuseregister = baas-aadress + 1.

Juhtregister = baas-aadress + 2.

### LPT-pordite registrite aadresside tabel<sup>9</sup>:

<i>Register</i>	<i>LPT1</i>	<i>LPT2</i>
Andmeregister ( <i>Data Register</i> )	0x378	0x278
Staatuseregister ( <i>Status Register</i> )	0x379	0x279
Juhtregister ( <i>Control Register</i> )	0x37A	0x27A

7 Beyondlogic.org - <http://www.beyondlogic.org/spp/parallel.htm> (18.04.2006)

8 Logix4u.net - <http://www.logix4u.net/parallelport1.htm> (11.01.2006)

9 Beyondlogic.org - <http://www.beyondlogic.org/spp/parallel.htm#5> (19.04.2006)

Ülevaatlik tabel (Tabel 2) erinevate pistikute kohta ja nende väljaviikude kirjeldused<sup>10</sup>:

Väljaviik (D-Type 25)	Väljaviik (Centronic)	Signaal	Andmete suund Sisse/välja	Register	Riistvaraline inversioon
1	1	Strobeerimine	sisse/välja	Juht	JAH
2	2	Bitt 0	välja	Andmed	
3	3	Bitt 1	välja	Andmed	
4	4	Bitt 2	välja	Andmed	
5	5	Bitt 3	välja	Andmed	
6	6	Bitt 4	välja	Andmed	
7	7	Bitt 5	välja	Andmed	
8	8	Bitt 6	välja	Andmed	
9	9	Bitt 7	välja	Andmed	
10	10	Kinnitus	sisse	Staatus	
11	11	Hõivatud	sisse	Staatus	JAH
12	12	Paber otsas	sisse	Staatus	
13	13	Valima	sisse	Staatus	
14	14	Tühi rida	sisse/välja	Juht	JAH
15	32	Viga	sisse	Staatus	
16	31	Initsialiseeri- mine	sisse/välja	Juht	
17	36	Printeri valik	sisse/välja	Juht	JAH
18 - 25	19-30	Maa	Maa		

Tabel 2: Ülevaatlik tabel

Riistvaraline inversioon tähendab, et kui näiteks tarkvaraliselt saata väljaviigule nr 14 loogiline 1(+5V), siis pordis see inversioneeritakse ehk pööratakse vastupidiseks, seega jõuab väljaviigule loogiline 0.

10 Beyondlogic.org - <http://www.beyondlogic.org/spp/parallel.htm> (26.04.2006)

## LPT-pordi puudumine

Kui peaks juhtuma, et uuel arvutil ei ole LPT-porti enam vajalikuks peetud või arvutil on lihtsalt LPT-port rikkis, siis võib poest osta PCI liidesesse käiva kontrolleri, mis sisaldab LPT-porti ja mure on murtud. Printeripordiga seade maksab umbes 150 kuni 200 krooni.

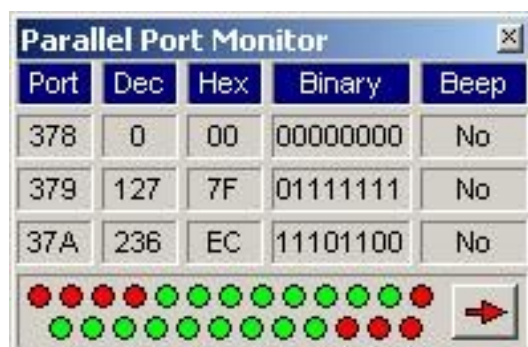
Sobib kõikidele arvutitele, millel on PCI liides ja on väga lihtsalt paigaldatav.

Lisaks PCI liidesesse käivatele kontrolleritele on olemas ka USB-porti käivad isendid.

## 3.3 LPT monitorid

LPT monitore, ehk programme, mis monitoorivad pordi seisundit ja lubavad seda ka muuta, on erinevate katsetuste käigus mugav kasutada.

**Paramon** (Kuvapildistus 3)



Kuvapildistus 3: Parallel Port Monitor

LPT-pordi väljaviikude seisundi jälgimiseks/muutmiseks sobib hästi programm nimega Paramon, mida on võimalik internetist alla laadida Geekhideout lehelt: <http://www.geekhideout.com/parmon.shtml>

Paramon näitab LPT-pordi väljaviikude loogilisi väärtusi ja neid on ka võimalik muuta, klikates hiirega vastava väljaviigu mumm peal. Biti 0 korral on vastav mumm roheline ja pordis vastav viik pingega 0V, biti 1 korral on vastav mumm punane ja pordis vastav viik pingestatud +5 voldiga.

Jooniselt 3 on visuaalselt näha, et pordi andme väljaviigud 2-9 on hetkel kõik nullid, mida näitab ka binaarkood 00000000.

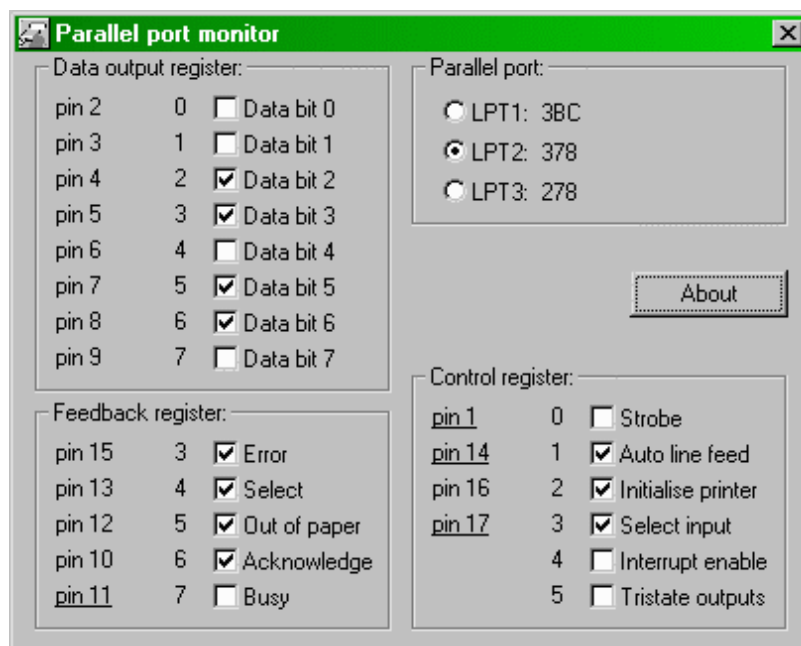
Oluline on tähele panna, et väljaviigud 1, 11, 14 ja 17 on riistvaraliselt inversionsioneeritud ehk kui näiteks pordi viigul nr 1 on reaalselt pinge +5V, siis tarkvaraliselt on seal loogiline 0. Paramoni punased ja rohelised mummud näitavad reaalselt hetkeseisu pordis – kas vastav väljaviik on pingestatud või ei.

Näiteks joonisel olev alumine parem mumm ehk pordi väljaviik nr 14, mis kuulub juhtimisregistrisse, on hetkel pingestatud olekus, kuid tarkvaraliselt omab see väärtust loogiline 0, mida on näha 37A binaarkoodi parempoolsest bitist.

### Parallel port monitor (Kuvapildistusel 4)

Välimuselt palju suurema, kus väljaviigud on nimedega märgitud, programmi LPT pordi seisundi jälgimiseks ja muutmiseks saab aadressilt: <http://neil.fraser.name/software/lpt/>

On mõnes mõttes ülevaatlikum, kuna erinevad väljaviigud jagatud registrite järgi, kuid tundub Paramoni kõrval natuke kohmakas ja suur.



Kuvapildistus 4: Parallel port monitor

Allajoonitud väljaviigud tähistavad riistvaralist inversiooni.

### 3.4 Userport

Lihtne kernelipõhine draiver WindowsNT/2000/XP jaoks, mis tagab programmidele loa otse sisend/väljund(I/O) portide poole pöördumiseks, mis muidu on NT põhistes süsteemides keelatud<sup>11</sup>.

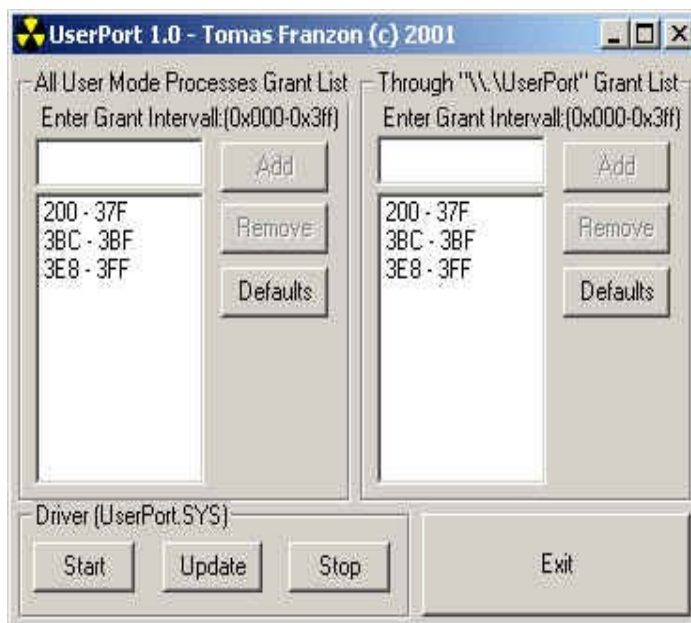
Internetis saadaval:

<http://www.embeddedtronics.com/public/Electronics/minidaq/userport/UserPort.zip>

Pakist läheb vaja ainult kahte faili: **UserPort.sys** ja **UserPort.exe**.

**UserPort.SYS** on vaja kopeerida „C:\Windows\System32\Drivers“ kausta.

Käivitada Userport.exe, mis toob nähtavale akna (Kuvapildistusel 5):

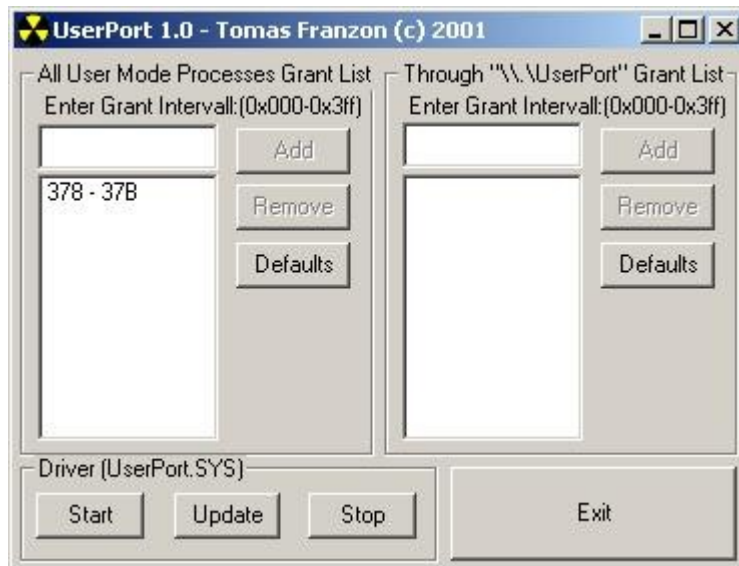


Kuvapildistus 5: UserPort alguses

Vajalik on kustutada nimekirjadest kõik kirjed, kasutades nuppu Remove.

Lisada uus aadresside vahemik 387 – 37B(nähtav Kuvapildistusel 6), kasutades nuppu Add.

<sup>11</sup> Writelog.com - [http://www.writelog.com/support/lpt\\_port\\_support\\_on\\_windows\\_nt.htm](http://www.writelog.com/support/lpt_port_support_on_windows_nt.htm) (20.01.2006)



*Kuvapildistus 6: UserPort pärast*

Järgmisena on vaja klõpsata nupul Update, Start ja Exit.

Arvutile tuleb teha seadete jõustumiseks alglaadimine.

### **UserPort eemaldamine**

UserPort'i kustutamiseks süsteemist on vaja eemaldada Drivers kataloogist userport.sys, mida võib teha käsuga: `del %windir\system32\drivers\userport.sys`

Kustutada registrist sissekanded:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\UserPort

Arvutile tuleb teha seadete jõustumiseks alglaadimine.

## 4. Programmeerimiskeeled

Järgnevalt on ära toodud näiteprogrammid LPT-pordi juhtimiseks kahes populaarses keeles:

Java ja C. Ülesannete lahendused on keeles Java

### 4.1 Java

#### 4.1.1 Kompilaator

Kompilaatori(JDK) javas kirjutatud programmide kompileerimiseks saab internetist Sun'i kodulehelt: <http://java.sun.com/j2se/1.5.0/download.jsp>

#### 4.1.2 Jnpout32\_v10 teek

Tegemist java teegiga, mis võimaldab javas kirjutatud programmidel juhtida(kirjutada ja lugeda väärtusi) LPT-porti. Saadaval internetis Douglas Beattie leheküljel: <http://hytherion.com/beattidp/comput/pport.htm>

**Lahtipakkimisel sisaldab kaust nelja faili:**

**ioPort.java** - mähisklass Jnpout32.DLL jaoks

**pPort.java** - sisaldab juhtimisfunktsioone ioPort klassi jaoks

**ioTest.java** - lühike ja kerge testprogramm, mis kasutab LPT1-porti aadressiga 0x378. Näha on täpselt, kuidas toimub kirjutamine ja lugemine LPT-pordist.

**run.bat** - kompileerib ja käivitab ioTest.java

Jnpout32.dll tuleb lisada PATH'i või kopeerida näiteks C:\Windows\System32 kataloogi.

ioPort.class ja pPort.class peavad asuma samas kataloogis, kus käivitatakse java programm!

### 4.1.3 Näiteprogramm

ioTest.java põhjal koostatud näiteprogramm, mis kirjutab porti ja loeb sealt:

```
public class ioTest
{
    static short datum; //muutuja andmete jaoks, mis kirjutatakse porti
    static short Addr; //muutuja pordi aadressi jaoks
    static pPort lpt; //LPT pordile ligipääsuks vajaliku objekti koostamine

    static void do_read()
    {
        datum = (short) lpt.input(Addr); //Lugemine
        System.out.println("Loen pordist: " + Integer.toHexString(Addr) +
                           " = " + Integer.toHexString(datum));
    }
    static void do_write()
    {
        System.out.println("Kirjutan porti: " + Integer.toHexString(Addr) +
                           " andmed = " + Integer.toHexString(datum));
        lpt.output(Addr,datum); //Kirjutamine
    }

    //Siit algab programmi täitmine
    public static void main( String args[] )
    {
        lpt = new pPort(); //uus objekt LPT pordi poole pöördumiseks
        Addr=0x378; //Pordi aadress
        datum=0x77; //16nd süsteemis andmed, mis kirjutatakse porti
        //Järgnev väljakommenteeritud rida esitab samu andmeid, mis üleminegi,
        //kuid 2nd süsteemis, et inimesel oleks parem näha, millised väärtused
        //pordi väljaviikudel millises järjekorras on. 2nd süsteemi kasutamisel
        //tuleks ülemine rida välja kommenteerida ja alumine rida sisse
        //kommenteerida.
        //datum = (short)Integer.parseInt("01110111",2);
        do_write(); //Kirjutatakse andmed porti
        do_read(); //Loetakse pordist andmed
    }
}
```

## 4.2 C++<sup>12</sup>

### 4.2.1 Kompilaator

Tasuta kompilaator Dev-C++ asub internetis <http://www.bloodshed.net/devcpp.html>

Sisaldab endas lisaks kompilaatorile ka arenduskeskkonda.

### 4.2.2 Inpout32 pakett

Inpout32.dll töötab Win98SE, Win2K ja WinXP'ga. Võimaldab teostada kirjutamis- ja lugemisoperatsioone paralleel porti. Lubab C's kirjutatud programmil juhtida LPT porti. Saadaval internetis Douglas Beattie leheküljel koos näiteprogrammidega, mis kasutavad Inpout32.dll'i: <http://hytherion.com/beattidp/comput/pport.htm>

### 4.2.3 C++ näiteprogramm

```
//Antud näiteprogramm ei vaja Inpout32 paketti.
#include <conio.h>
#include <stdio.h>

int main()
{
    int andmed = 1; //Andmed kümnendsüsteemis
    while (andmed<129)
    {
        _outp(0x378,andmed); //Registrisse 0x378 kirjutatakse andmed x
        _sleep(500);
        andmed = andmed * 2; //Peale korrutamist liigub bitt 1 ühe koha võrra edasi
    }
    _outp(0x378,0); //Andmeregistrisse kirjutatakse 0
    return 0;
}
```

---

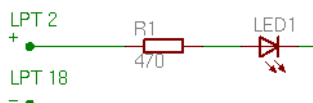
12 Douglas Beattie - <http://hytherion.com/beattidp/comput/pport.htm> (11.02.2006)

## 5. Teemad

Teemad erinevatest valdkondadest peaksid andma hea ettekujutse LPT-pordiga erinevate seadmete juhtimisest. Alustatud on võimalikult lihtsate teemade ja skeemidega ning lõpetatud märksa raskematega, mis võiksid huvi pakkuma ka juba kogunud elektroonikule. Lisaks võib iseseisvalt teha erinevaid pinge ja voolutugevuse mõõtmisi.

### 5.1 Valgusdiodide vilgutamine

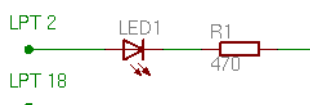
Esimeseks katsetuseks sobiks kõige paremini valgusdiodide (*LED – Light-emitting diode*) ühendamine otse LPT-porti. Sellise seadme skeemist enam midagi lihtsamat olla ei saagi, sest puudub väline toiteallikas – voolu võetakse otse pordist ja detailide arv on väga väike. Tarvis läheb ainult takistit voolu piiramiseks ja valgusdiodi ennast.<sup>13</sup>



Skeem 1: Valgusdiod

Skeemi 1 kohaselt ühendatakse valgusdiod jadamisi 470 $\Omega$  takistiga, mis piirab voolu diodile ja pordile parajale ning ohutule tasemele. Takisti ots ühendub LPT väljaviik nr 2 ehk esimese andmeviigu (andme bitt 0) külge ja valgusdiodi ots ühendub LPT väljaviigu 18 külge, mis on maanduseks.

Ka on lubatud takisti ja diodi kohtade vahetamine skeemis (nähtav skeemil 2) – kõik toimib endiselt. Tuleb ainult jälgida, et diodi skeemitähis ja diod skeemis oleksid õiget pidi, sest diodid juhivad voolu ainult ühes suunas.



Skeem 2: Valgusdiod 2

13 AndyWorld - <http://www.hot.ee/andy00/projektid/ele/lpt01.htm> (11.01.2006)




Vastavalt Ohmi seadusele, mis ütleb, et voolutugevus on võrdeline pingega ja pöördvõrdeline takistusega, siis pinge 5V ja takistuse 470Ω korral on voolutugevus ~10mA. See on valgusdiodi ja ka LPT-pordi jaoks täpselt paras voolutugevus.

**Ohmi seadus ütleb:**  $I = U / R$  ehk voolutugevus = pinge / takistus

$$5V / 470\Omega = \sim 10mA$$

Oluline on jälgida, et valgusdiodid oleksid skeemis alati õiget pidi ühendatud, sest vastasel juhul ei hakka nad helendama, kuna juhivad voolu ainult ühes kindlas suunas, nagu diodid ikka.

**Selgitav tabel diodi pluss- ja miinuspoole leidmiseks (Tabel 3):**

<i>Joonis/pilt</i>	<i>Selgitus</i>
	Valgusdiodi skeemitähis paigutatakse skeemi nii, et tähises olev kolmnurk suunab oma teravikuga vertikaalse triibu ehk miinus poole suunas.
	Pildilt on näha, et valgusdiodi pluss poole ehk anoodi traat on lühem kui katood.
	Pildilt on näha, et uue valgusdiodi pluss-jalg on natukene pikem. Kasutatud või kuskilt skeemist lahti joodetud valgusdiodide puhul tõenäoliselt sellist jalgade pikkuse erinevust märgata enam ei ole. Küll aga võib anoodi ja katoodi traate võrrelda.

*Tabel 3: Diodide kirjeldus*

Andes LPT väljaviigule nr 2 väärtuseks loogiline 1(+5V), hakkabki diodid helendama, väärtuse loogiline 0 (0V) korral on diodid kustunud.

**Ülesanne:** Koostada programm, mis paneks väljaviigu nr 2 külge ühendatud valgusdiodi helendama.

**Lahendus:**

```
import java.io.*;
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write() { lpt.output(Addr,datum); }
    public static void main( String args[] )
    {
        lpt = new pPort(); //Uus objekt pordi poole pöördumiseks.
        Addr=0x378; //Kasutatakse LPT pordi baasaadressi, ehk andmeregistrit.
        //Andmed esitatakse kahendsüsteemis, kuna nii on visuaalselt parem
        näha, millistel väljaviikudel parasjagu millised väärtused on. Andmed
        muudetakse programmi poolt 16nd koodi.
        datum = (short)Integer.parseInt("00000001",2);
        do_write(); //Kirjutab andmed reaalselt porti
    }
}
```

**Ülesanne:** Koostada programm, mis vilgutaks valgusdiodi sagedusega 1Hz.

**Lahendus:**

```
import java.io.*;
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;
    static void do_write() { lpt.output(Addr,datum); }

    public static void main( String args[] ){
        lpt = new pPort();
        Addr=0x378;
```

```

while (true) //Lõpmatu tsükkel
{
    datum = (short)Integer.parseInt("00000000",2); //Diod kustus
    do_write();
    try{Thread.sleep(500);}catch(Exception e){} //Oodatakse pool sekundit
    datum = (short)Integer.parseInt("00000001",2); //Diod helendab
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
}
}
}

```

**Ülesanne:** Ühendada diod väljaviigu nr 9 (andmebitt nr 8) külge ja panna eelnevad programmid seda arvestama.

**Ülesanne:** Ühendada valgusdiod tagurpidi ja veenduda, et see ei hakka helendama.

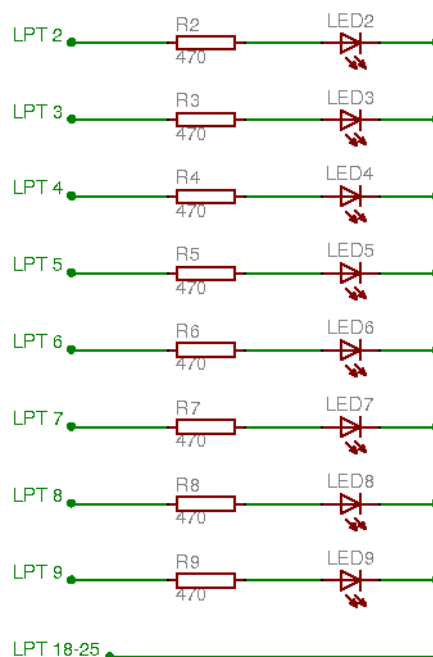
Kui ühe diodi vilgutamine on selgeks saadud, võib neid korraga rohkem kasutusele võtta.

Järgnevas skeemis (Skeem 3) on valgusdiodid (*LED2-LED9*) ühendatud LPT-pordi kõikidesse andme väljaviikudesse (LPT2-LPT9) ja iga diodiga on järjestikku takistid (R2-R9) väärtustega  $470\Omega$ , mis toimivad voolupiirajatena, et voolutugevus lubatust suuremaks ei läheks ja niimoodi porti läbi ei põletaks.

Valgusdiodide miinusklemmid ühendatakse LPT väljaviikude 18-25 külge. Skeemides võib pea alati kõik maandused omavahel kokku ühendada - nii on tehtud ka skeemis 3, kus on omavahel kokku ühendatud LPT väljaviigud 18-25 ja diodide miinusklemmid.

LPT-pordil on iga andmebiti kohta üks eraldi maandusklemm, kuna niimoodi andmeid edastades peaks olema vähem häireid. Käesolevas õppematerjalis esitatud skeemid ei ole nii tundlikud, et iga bitt peaks eraldi maandust kasutama, seega võibki LPT-porti kõik maandusklemmid omavahel kokku ühendada ja kasutada skeemi maandusena. Kaheksa maandusjuhtme asemel saab kasutada ainult ühte.

### Skeemil 3 kaheksa valgusdiodi ühendamine LPT-porti:



Skeem 3: Valgusdiodid

Oluline on jälgida, et kõik valgusdiodid oleksid ühendatud õiget pidi – diodi pikem jalg jäägu takisti poole. Vastasel juhul ei hakka lihtsalt diod helendama, sest diodid juhivad voolu ainult ühes suunas.

Tulukeste juhtimine käib lihtsa binaarloogika abil tarkvaraliselt. Kui vastavale LPT väljaviigule ilmub binaarväärtus 1, hakkab antud valgusdiod helendama ja väärtuse 0 korral on diod kustus.

**Ülesanne:** Koostada programm, mis paneks kõik valgusdiodid helendama.

**Lahendus:**

```
import java.io.*;
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write() { lpt.output(Addr,datum); }
    public static void main( String args[] )
    {
```

```

lpt = new pPort();
Addr=0x378;

    //Andmed esitatakse 16nd süsteemis, kus väärtus 0xFF on suurim, ehk
    2nd süsteemis 11111111 ja 10nd süsteemis 255

datum = 0xFF;

    //Kui peetakse mugavamaks inimese jaoks kasutada 2nd süsteemi, siis
    võib eelneva koodirea välja kommenteerida ja järgneva sisse
    kommenteerida.

//datum = (short)Integer.parseInt("11111111",2); //2nd süsteemis
do_write(); //Kirjutab andmed reaalselt porti
}
}

```

**Ülesanne:** Koostada programm, mis paneks helendama LPT väljaviigu nr 9 külge ühendatud diodi, siis kustutaks selle ära ja paneks väljaviigu nr 8 küljes oleva diodi helendama jne...

**Lahendus:**

```

import java.io.*;

public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;
    static void do_write() { lpt.output(Addr,datum); }

    public static void main( String args[] )
    {
        lpt = new pPort();
        Addr=0x378;
        while (true)
        {
            datum = (short)Integer.parseInt("10000000",2);
            do_write();
            try{Thread.sleep(500);}catch(Exception e){}
            datum = (short)Integer.parseInt("01000000",2);
            do_write();
            try{Thread.sleep(500);}catch(Exception e){}
            datum = (short)Integer.parseInt("00100000",2);
            do_write();
            try{Thread.sleep(500);}catch(Exception e){}
        }
    }
}

```

```

    datum = (short)Integer.parseInt("00010000",2);
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
    datum = (short)Integer.parseInt("00001000",2);
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
    datum = (short)Integer.parseInt("00000100",2);
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
    datum = (short)Integer.parseInt("00000010",2);
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
    datum = (short)Integer.parseInt("00000001",2);
    do_write();
    try{Thread.sleep(500);}catch(Exception e){}
}
}
}

```

**Ülesanne:** Koostada eelnev programm tsükli abil.

**Lahendus:**

```

import java.io.*;

public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;
    static void do_write() { lpt.output(Addr,datum); }

    public static void main( String args[] )
    {
        lpt = new pPort();
        Addr=0x378;
        int arv = 128; //10nd süsteemi arv 128, 2nd süsteemis 10000000
        while (true)
        {
            datum = (short)arv; //arv teisendatakse 16nd süsteemi
            do_write();
        }
    }
}

```

```

try{Thread.sleep(500);}catch(Exception e){}
if (arv==1) { arv = 128*2; } //Kui jõutakse arvuni 1(2nd süsteemis
  00000001), siis alustatakse otsast peale 128'ga. Kahega korrutamine
  on vajalik selleks, et järgnev koodirida jagab arvu kahega.
arv = arv/2; //Kui arv oli enne 128, siis jagatakse kahega, ning
  saadakse 10nd süsteemi arv 64, mis 2nd süsteemis on 01000000
}
}
}

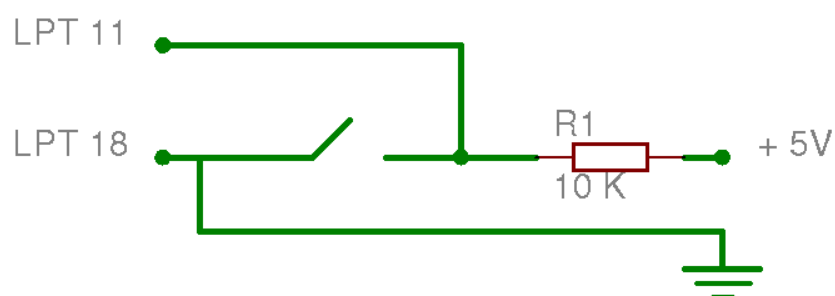
```

**Ülesanne:** Koostada programm, mis kuvaks programmi sisestatud kümnendsüsteemis arvu valgusdiodidel kahendsüsteemis.

## 5.2 Sisendite kasutamine

Kuna arvuti funktsiooniks peale andmete töötamise ja väljastuse on ka andmete lugemine, siis on võimalik LPT-portigi kasutada andmete sisestamiseks. Näiteks võib panna lüliti uste külge ja programmi statistikat arvutama, millal ja kui kaua teatud ukseid lahti on või palju inimesi sealt käib.

Sisendeid on standardprinteripordil ainult 5(staatus registreeritud 0x379), kuid nende otstarbeka kasutamisega saab peaaegu kõik vajalikud andmed sisestatud. Sisendite lugemiseks on mitmeid meetodeid. Neist lihtsaim ja kindlaim ning ainult ühe sisendiga on toodud allpool oleval skeemil 4.<sup>14</sup>



Lüliti lahtioleku ajal on LPT väljaviik 11 peal +5volti ja takisti R1 on voolu piiramiseks, et see liialt tugevaks ei läheks. Väljaviigus 11 on seega loogiline 1. Olenevalt LPT-pordi iseärasustest võib

<sup>14</sup> AndyWorld - <http://www.hot.ee/andy00/projektid/ele/lpt02.htm> (04.03.2006)

hakkama saada ka ilma välise toiteallikata ning pordis võib juba vaikimisi nullist natuke kõrgem pinge olla ning seega loogiline 1 ning maandamise korral muutub see loogiliseks nulliks ja pinge kaob vastavast väljaviigust. Mis juhtub ka lüliti kinnioleku ajal: maandatakse viik 11 viiku 18 ning viigu 11 loogiline nivoo läheb ühelt nulliks.

Tähelepanu tasuks pöörata asjaolule, et skeem kasutab lisatoiteallikat, mistõttu peab väga hoolikalt jälgima, kas kokkupandud seadme juures on ikka kõik juhtmed õigestes kohtades, sest pordi läbipõletamine lühise korral ei ole üldse mitte raske.

Voolub võib võtta ka arvuti toiteploki, kust +5V tulev juhe on punane ja maandus on musta värvi.

**Ülesanne:** Koostada programm, mis teataks iga sekundi tagant, kas väljaviiku 11 ühendatud lüliti on kinni või lahti.

**Lahendus:**

```
public class Ylesanne
{
    static short datum;
    static short datum_abi;
    static short Addr;
    static pPort lpt;

    static void do_read(){ datum = (short) lpt.input(Addr);}

    public static void main( String args[] )
    {
        lpt = new pPort();

        while (true)
        {
            Addr=0x379;
            do_read();

            if (datum == 0xFF) //Kuna pordis on algselt 0x7F (011111112) ning peale
                viigu 11 maandamist peab seega olema 0xFF (111111112). Tähele
                tuleks panna, et kuna viik 11 on invertteeritud, siis maandamise
                korral muutub seal väärtus nullist üheks.

                System.out.println("Lyliti kinni");
            else
                System.out.println("Lyliti lahti");
```

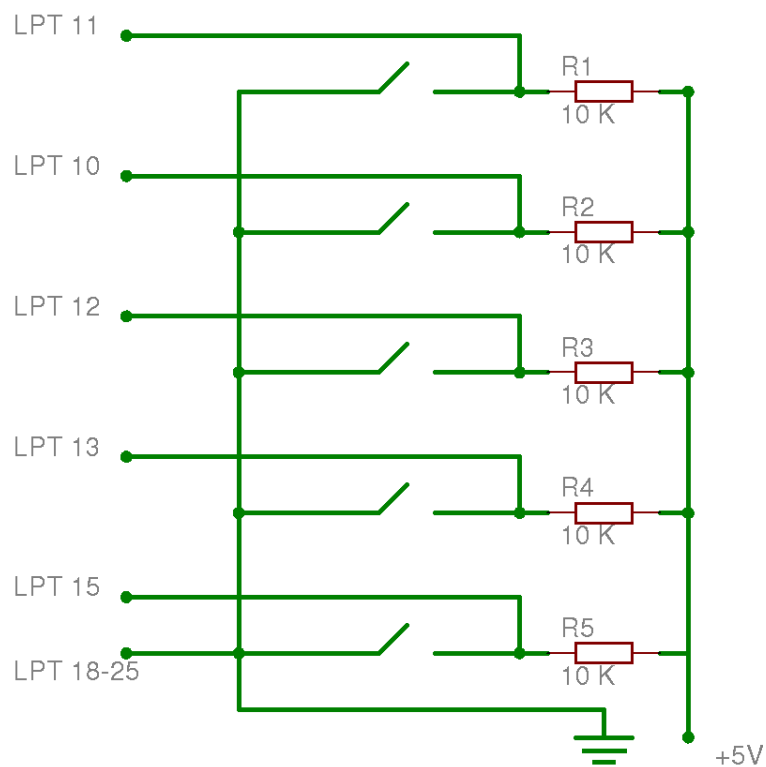
```

    try{Thread.sleep(1000);}catch(Exception e){}
  }
}
}

```

**Ülesanne:** Kasutada väljaviigu 11 asemel väljaviiku 13 ja modifitseerida programmi vastavalt.

Rohkemate sisendite kasutamise korral on skeem põhimõtteliselt samasugune, kuid kasutatakse rohkem sisendeid (Skeem 5):



Skeem 5: Sisendid

Antud skeemis kasutatakse iga sisendi oleku säätamiseks ühte lüliti ja ühte takistit. Takisti on kasutusel, et tekitada sisendisse kõrge potentsiaaliga pinge (loogiline nivoo 1) ja samuti voolu piiramiseks, mis satub sisendisse, et kaitsta porti ennast ja selle taga peituvaid kiviid.

Sisendis on loogiline nivoo 1, kui lüliti on avatud ja loogiline nivoo 0, kui lüliti on suletud ehk sisend ühendatud maanduse ehk miinusklemmiga.

**Ülesanne:** Koostada programm, mis annaks teada sisendite seisust iga sekundi tagant.

**Lahendus:**

```
public class Ylesanne
{
    static short datum;
    static short datum_abi;
    static short Addr;
    static pPort lpt;

    static void do_read(){
        datum = (short) lpt.input(Addr);
    }

    public static void main( String args[] )
    {
        lpt = new pPort();

        while (true)
        {
            datum_abi = datum; //Eelmine kord loetud väärtus jäetakse meelde
            Addr=0x379;
            do_read(); //Loetakse uus väärtus

            if (datum != datum_abi) //Kui seis on endine, uut teavet ei kuvata.
            {
                System.out.println("Loen LPT pordist sisendit: " +
                    Integer.toHexString(Addr) + " = " + Integer.toHexString(datum));
            }
            try{Thread.sleep(1000);}catch(Exception e){}
        }
    }
}
```

**Ülesanne:** Koostada programm, mis teataks sõnadega kõikide lülitite seisust. Näiteks: „Lüliti nr 3 on suletud“.

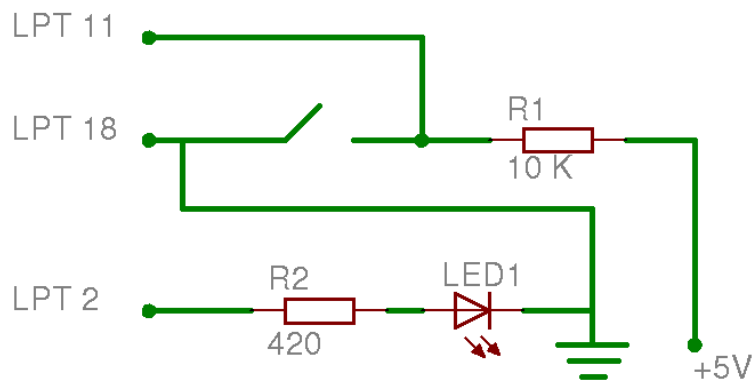
**Ülesanne:** Koostada kahe sisendiga skeem ja kirjutada virtuaalset autot juhtiv programm, mis esimese sisendi korral pidurdaks autot hoogu maha ja teise sisendi korral kiirendaks auto liikumist.

**Ülesanne:** Koostada nelja sisendiga skeem ja kirjutada programm, mis liigutaks ekraanil olevat mummy vastavalt sisenditele paremale-vasakule ning üles-alla.

### 5.3 Sisendid ja väljundid koos

Arvuti võib panna ka reageerima teatud sisendite muutustele teatud väljundite muutustega – sisendandmete töölemine ja väljundi kuvamine ongi ju arvuti funktsiooniks, kuid üldlevinud klaviatuur ja monitor on asendatud LPT-porti ühenduva skeemiga.

#### Skeem sisendi ja väljundi koos kasutamiseks (Skeem 6):



Skeem 6: Sisend ja väljund

Skeemi kohaselt on algselt lüliti lahti ning seega LPT väljaviik nr 11 pingestatud ja väljaviiku nr 2 ühendatud diod kustus. Lüliti sulgemisel maandatakse väljaviik nr 11 ning sellelt kaob pinge, kuid tarkvaraliselt võib selle väljaviigu väärtuseks lugeda ühe, kuna too viik on riistvaraliselt invertteeritud.

LPT väljaviiku nr 2 ühendatud diod hakkab helendama, kui pordi esimese andmebiti väärtuseks anda 1.

**Ülesanne:** Koostada programm, mis LPT väljaviik nr 11 loogilise nivoo ühest muutmisel nulliks paneks helendama valgusdiodi, mis asub LPT väljaviigus nr 2.

#### Lahendus:

```
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;
```

```

static void do_read()
{
    datum = (short) lpt.input(Addr);
    System.out.println("Loen pordist: " + Integer.toHexString(Addr) +
        " = " + Integer.toHexString(datum));
}

static void do_write()
{
    System.out.println("Kirjutatan porti: " + Integer.toHexString(Addr) +
        " with data = " + Integer.toHexString(datum));
    lpt.output(Addr,datum);
}

public static void main( String args[] )
{
    lpt = new pPort();
    Addr=0x378;

    while (true)
    {
        Addr=0x379; //Sisendite registri aadress
        do_read(); //Loeb sisendeid

        if (datum == 0xFF) //Väljaviik 11 lülitatud(maandatud)
        {
            Addr=0x378; // Andmeregistri väljaviiku nr 2
            datum=0x01; // kirjutatakse 1
            do_write(); //
        }else //Väljaviik 11 lülitamata (maandamata)
        {
            Addr=0x378; // Andmeregistri väljaviiku nr 2
            datum=0x00; // kirjutatakse 0
            do_write(); //
        }
        try{Thread.sleep(1000);}catch(Exception e){}
    }
}
}

```

**Ülesanne:** Võtta kasutusele 2 sisendit ja kaks väljundit.

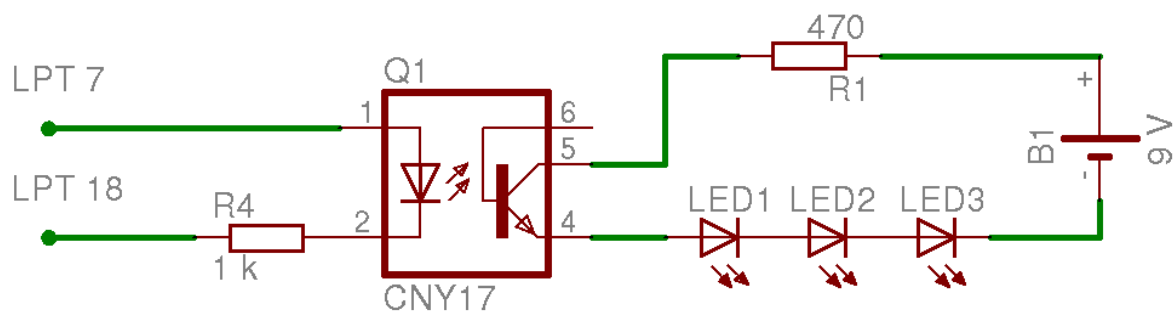
## 5.4 Välist toiteallikat kasutavate seadmete lülitamine

Lisaks lihtsale valgusdiodide helendama panemisele on võimalik LPT pordi abil sisse lülitada ka seadmeid, mis kasutavad välist toiteallikat – alalisvooluga töötavad pirnid, mikromootorid ...

Sellised skeemid on juba tunduvalt keerulisemad, kuna LPT port on võimeline välja andma ainult kuni mõnikümmend milliamprit, siis suuremate seadmete lülitamiseks on vaja voolu võimendada kasutades välist toiteallikat.

Järgnev autori koostatud skeem kasutab välist toiteallikat 9-voldise patarei näol ja optronit kolme valgusdiodi juhtimiseks:

**Optroniga vooluvõimendi skeemil 7:**



Skeem 7: Optroniga vooluvõimendi

Optron CNY17 kujutab endast releetaolist mikroskeemi, kuid erinevalt releest on tegu kontaktivaba lülitiga. Optroni korpuses on valgusallikas ja fototundlik element. Valgusallika süttides mõjutab valgusvoog valgustundlikku elementi, mille tulemusel muutub selle takistus väikeseks, ehk teisisõnu väljaviigud 4 ja 5 hakkavad omavahel voolu juhtima.. Valgustundlikuks elemendiks on kasutusel fototransistor. Võrreldes releega on optronil tunduvalt väiksemad mõõtmed, liikuvate osade puudumine ja tunduvalt väiksem voolutarve.<sup>15</sup> CNY17 talub lülitamisel maksimaalselt kuni 60mA voolu.

Takisti R4 on optroni valgusallika voolu piiramiseks ja takisti R1 piirab valgusdiodide ahela voolu.

Kui LPT pordi väljaviik 7 pingestatakse loogilise 1 näol, siis optroni valgusallikas hakkab kiirgama

<sup>15</sup> AndyWorld - <http://www.hot.ee/andy00/projektid/ele/spikker.htm> (17.03.2006)

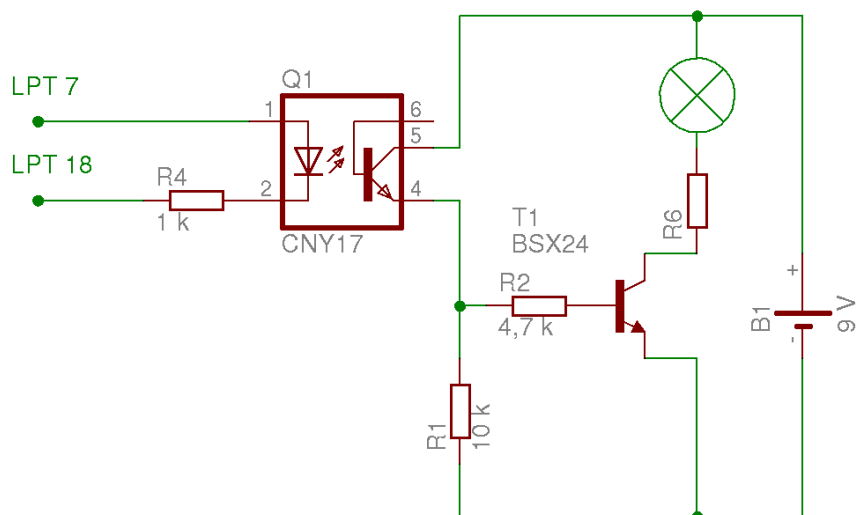
ja fototransistor reageerib sellele väljaviikude 5 ja 4 „kokkuühendamise“ga.

Optron pakub kaitset pordile, kuna välist toidet kasutatav skeemiosa lülitatakse tööle valguse toime tõttu ning füüsiline ühendus pordi ja valgusdioodide vahel puudub – skeem on galvaaniliselt lahtisidestatud.

**Ülesanne:** Koostada programm, mis vilgutaks eelmises skeemis olevaid valgusdiodide.

Kui on vaja lülitada seadmeid, mis tarbivad rohkem voolu, kui optron lülitada suudab, võib kasutusele võtta transistorid.

**Ühe transistoriga vooluvõimendi skeemil 8:**



*Skeem 8: Transistoriga vooluvõimendi*

Antud skeem ühendatakse LPT-pordi väljaviikude 7 ja 18 külge.

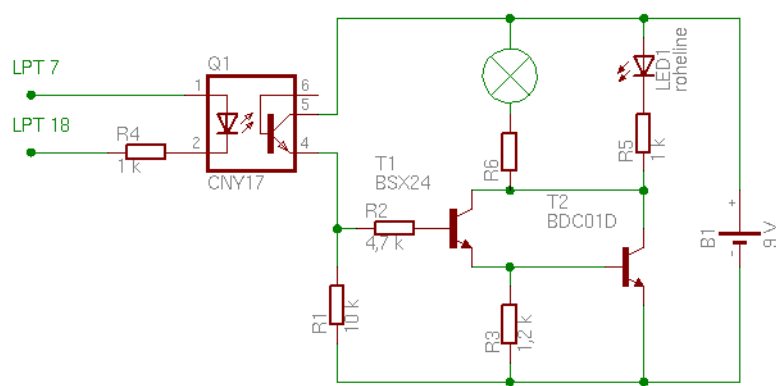
Skeemis olev transistor T1 talub voolu kuni 100mA, mis võimaldab lülitada rohkem voolu vajavaid seadmeid, kui optron üksinda. Antud skeemis lülitatakse pirn põlema.

Optroni lülitamise korral jõuab vool transistori baasile läbi takisti R2, mis piirab baasvoolu ohutule tasemele, ja transistor avaneb ning pirn hakkab helendama. Takisti R1 kaitseb transistori iseenesliku avanemise ees ja hoiab transistori kindlalt suletuna – baasi maandatuna.

R6 tuleb valida vastavalt sellele, millist pirni või muud seadet kasutatakse. Kui kasutatav seade võib töötada otse 9 voldi pealt, siis võib R6 takisti ära jätta.

Kahe või enama transistoriga lülitusi kasutatakse kohtades, kus ühe transistori kasutamine ei ole suure voolu tõttu võimalik ja võimsat transistorit on vaja juhtida nõrga vooluga. Sel juhul pannakse järjestikku mitu transistorit, nii et, nõrgem neist juhib võimsamat.

### Kahe transistoriga vooluvõimendi skeemil 9:



Skeem 9: Kahe transistoriga vooluvõimendi

Antud skeem ühendatakse LPT pordi väljaviikude 7 ja 18 külge.

Kui LPT väljaviik nr 7 on pingestamata (väärtus loogiline 0), siis optroni (CNY17) transistor on suletud ning BSX24 baasil on pinget null: BSX24 on suletud ja seega ka BDC01D.

Kui LPT väljaviik nr 7 saab väärtuseks loogiline 1, siis pingestatakse optron ning tema transistor avaneb: väljaviigud nr 4 ja nr 5 hakkavad voolu juhtima. Takistile R1 langeb kogu toitepinge, mis avab transistori T1. Teoreetiliselt võiks R1 ka puududa, sest ränitransistorid püsivad piisavalt hästi kinni, aga töökindluse mõttes võiks siiski skeemis olla: hoiab T1 paremini suletuna. R2 piirab T1 baasvoolu ohutule tasemele.

T1 avaneb ja toitepinge läheb läbi pirni ja R6 transistori T2 baasile ning avab selle. R3 on samal otstarbel nagu R1 - hoiab transistori kindlalt suletuna kui juhtpinge puudub. Pirn hakkab helendama ja R6 piirab voolu läbi selle. LED 1 (roheline) põleb siis, kui ka pirn on voolu all ja R5 piirab selle LEDi voolu.

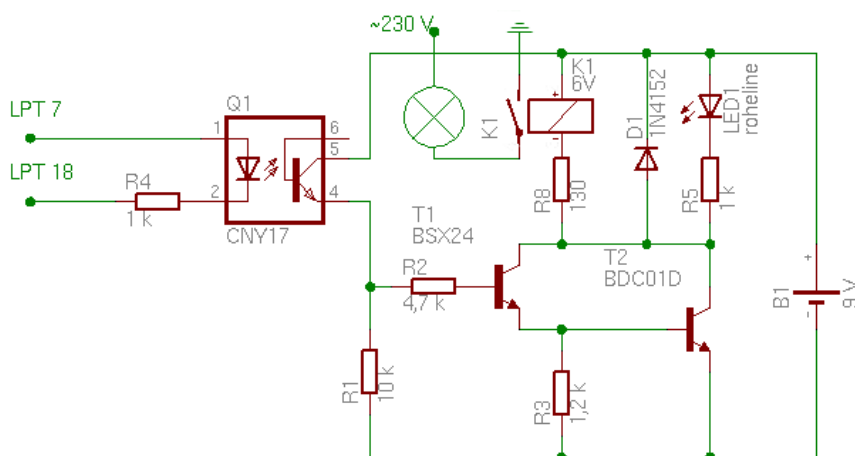
Pane tähele:

- R6 tuleb arvutada vastavalt sellele, kui palju pirn helendamiseks voolu vajab.
- Pirni võib asendada ka mõne muu seadmega, mis töötab 9V patarei pealt.
- Transistori T2 kollektorvool võib olla maksimaalselt 500mA.
- Kui seade töötab alla 100mA, siis võib transistori T2 ära jätta, sest transistori T1 maksimaalne kollektorvool võib olla kuni 100mA.

## 5.5 Võrgutoidet kasutavate seadmete lülitamine

Lisaks lihtsale valgusdiodivilgutamisele on võimalik LPT pordi abil ka juhtida seadmeid, mis kasutavad võrgutoidet – röster, raadio, laelamp...

**Võrgutoidet kasutavate seadmete lülitamine skeemil 10:**



Skeem 10: Võrgutoite lülitamine

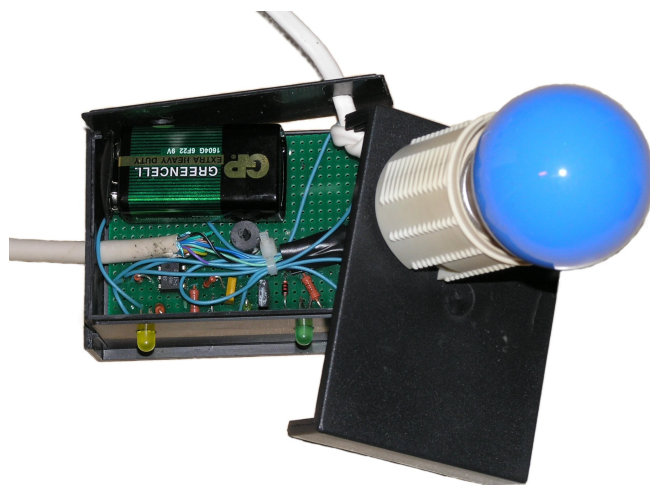
Antud skeem (Skeem 10) ühendatakse LPT pordi väljaviikude 7 ja 18 külge ja on küllaltki sarnane eelnevate skeemidega.

Kui T1 avaneb, siis toitepinge läheb läbi relee ja R8 transistori T2 baasile ning avab selle. R3 on samal otstarbel nagu R1 - hoiab transistori kindlalt suletuna kui juhtpinge puudub. Relee rakendub ja R8 piirab voolu läbi relee. D1 kaitseb transistorit T2 relee tagastumisel (releed toitva ahela

katkemisel) mähises tekkiva eneseinduktsiooni elektromotoorjõu ees. Selle elektromotoorjõu pinge on vastupidine toitepingele ning sõltub relee induktiivsusest, mis võib olla päris kõrge, sest analoogselt saadakse pinge televiisorites 9kV kineskoobi tarvis. Nii võiks T1 läbi põleda. LED 1 (roheline) põleb siis, kui ka relee on voolu all ja R5 piirab selle LEDi voolu.

Antud skeemi on autor kasutanud ka seminaritöös „ID-kaardi rakendus“ ja realiseeritud nõndaviisi id-kaardiga autentimisele riistvaralise toiminguga vastamine.

**Pilt „ID-kaardi rakenduse“ riistvaralisest seadmest (Pilt 1):**



*Pilt 1: ID-kaardi rakenduse riistvaraline seade*

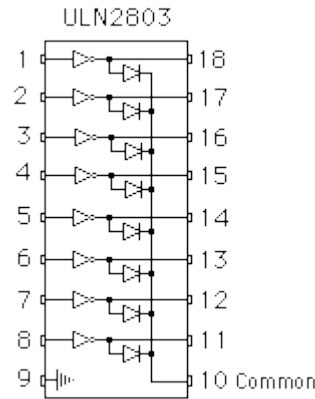
## **5.6 Välist toiteallikat kasutavate seadmete lülitamine mikroskeemiga**

Kui on vaja lülitada mitut seadet, mis kasutavad välist toiteallikat, siis eelnevate transistoritel põhinevate skeemide mitmes eksemplaris valmistamine muutub igavaks ja võtab palju aega ning raha. Suur hulk detaile võtab enda alla ka suure hulga trükiplaadi pinda.

Lahenduseks võib olla mikroskeem **ULN2803**<sup>16</sup> (Joonis 2):

---

<sup>16</sup> LCDprojects - <http://www.alessioviti.com/lcdprojects/datasheet/2803.gif> (20.04.2006)



Joonis 2: ULN2803

Tegemist on mikroskeemiga, mis on 8-bitine ehk 1-baidine. NPN tüüpi transistorite darlingtoni jada, kus iga sisendit võimendatakse kahe transistoriga.

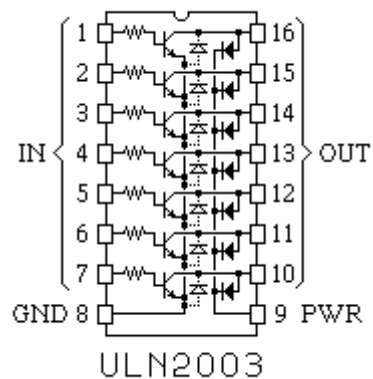
Kannatab lülitamisel voolu kuni 500mA ja pinget 50V iga lülituse kohta. Sobib hästi kohtadesse, kus nõrga vooluga on vaja juhtida erinevaid lisaseadmeid. Näiteks LPT-pordiga releesid, stepper-mootoreid, magneteid, LED ekraane jne ...

Andmeleht asub: <http://impressolibre.sourceforge.net/miniplotter/ULN2803-D.PDF>

Kuid ULN2803 omapära on selles, et ta on invertteeritud, mis tähendab seda, et kui näiteks sisend 1 ei ole pingestatud, siis väljund 18 on. Kui sisend 1 on pingestatud, siis väljund 18 ei ole pingestatud.

Kui ei ole vaja just nimelt 8-„kanalit“ kasutada, võib pruukida ULN2003'e, millel on küll seitse kanalit ehk 7 bitti (16 väljaviiku kokku), kuid seda on mugavam kasutada. Ehk siis kui sisendis 1 on pinge, on väljund 16 ühendatud maha, ehk juhib voolu (on lülituses).

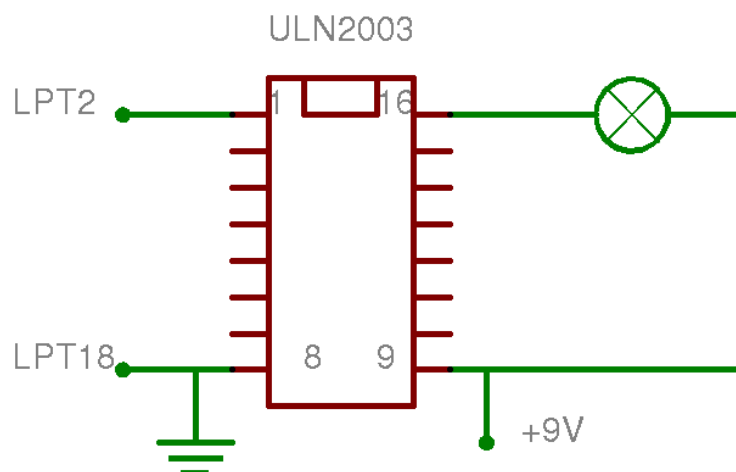
**ULN2003 väljaviigud**<sup>17</sup> (Joonis 3):



Joonis 3: ULN2003

17 PPMPs - [http://ppmps.zesoi.fer.hr/projekt/video\\_nadzor/data/motor\\_files/image008.gif](http://ppmps.zesoi.fer.hr/projekt/video_nadzor/data/motor_files/image008.gif) (18.04.2006)

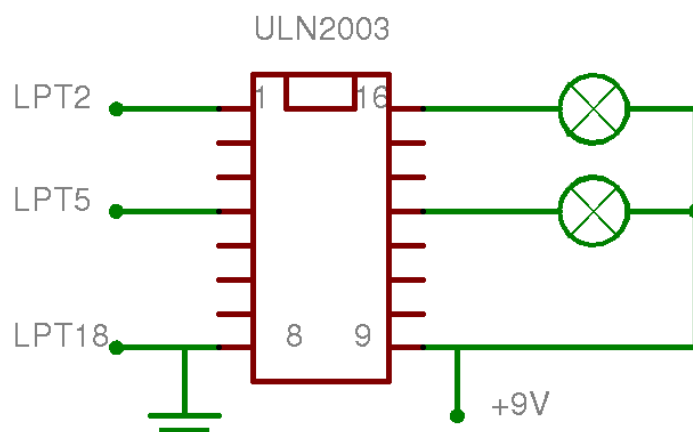
**Skeem 9-voldise pirni helendama panemiseks LPT-pordist saadava signaali abil (Skeem 11):**



*Skeem 11: ULN2003*

Antud skeem (Skeem 11) on küllaltki põhimõtteline. ULN2003e väljaviik nr 1 ühendatakse LPT-pordi väljaviigu 2 külge, millega juhitakse seadme tööd. Kui ULN2003 väljaviigu nr 1 pingestatakse, siis väljaviik 16 maandatakse ja pirn hakkab helendama.

Sisendeid on ULN2003 kivil 7 ning seega on võimalik juhtida ühe kiviga kuni seitsme seadme tööd. Järgneva skeemi kohaselt juhitakse LPT väljaviikudega 2 ja 5 kahte erinevat pirni – sama loogika (Skeem 12) kohaselt võib seitse pirni rööbiti ühendada:



*Skeem 12: ULN2003*

**Ülesanne:** Realiseerida eespool olevad skeemid (Skeem 11, Skeem 12) ja kirjutada programm pirnide juhtimiseks.

**Ülesanne:** Ühendada rööbiti 7 pirni ja juhtida neid programmeeritavalt nagu valgusdioode.

**Ülesanne:** Koostada programm, mille abil saaks klaviatuuri numbriklahvidega (1-8) süüdata ja kustutada pirne.

**Ülesanne:** Koostada programm, mis näitaks arvuti helitugevuse nivood pirnidel.

## 5.7 Steppermootori juhtimine

### 5.7.1 Ülevaade steppermootoritest

Stepper- ehk samm-mootorid ei ole nagu tavalised alalisvoolumootorid, millede võll lihtsalt ringi pöörleb pingestamisel. Samm-mootorite võll liigub sammude kaupa ehk sammub.

Steppermootoreid kasutatakse kohtades, kus on vaja liigutada midagi sammude kaupa ja täpselt.

Steppermootoreid leidub maatriksprinterites, disketi seadmetes, erinevates robotites ...

#### Steppermootoreid on kahte liiki:

1. Biopolaar – magnetitevahelisi tõmbeid ja tõukeid juhitakse erinevate mähiste pingestamisega teatud suunalise vooluga.
2. Unipolaar – magnetitevaheseid tõmbeid ja tõukeid juhitakse erinevate mähiste pingestamisega.

Unipolaarseid steppermootoreid juhitakse enamasti nelja juhtmega, mis lähevad erinevate mähiste külge ning viies või viies ja kuues on maanduseks. Unipolaarseid steppermootoreid on lihtsam juhtida kui biopolaarseid.

Steppermootorite kohta võib täiendavalt lugeda

Douglas W. Jones ja Ian Harrise veebilehtedelt

<http://www.cs.uiowa.edu/~jones/step/> ja

<http://www.doc.ic.ac.uk/~ih/doc/stepper/>

Pildil 2 olev steppermootor on pärit maatriksprinterist ja sammuks on tal 7,5 kraadi.

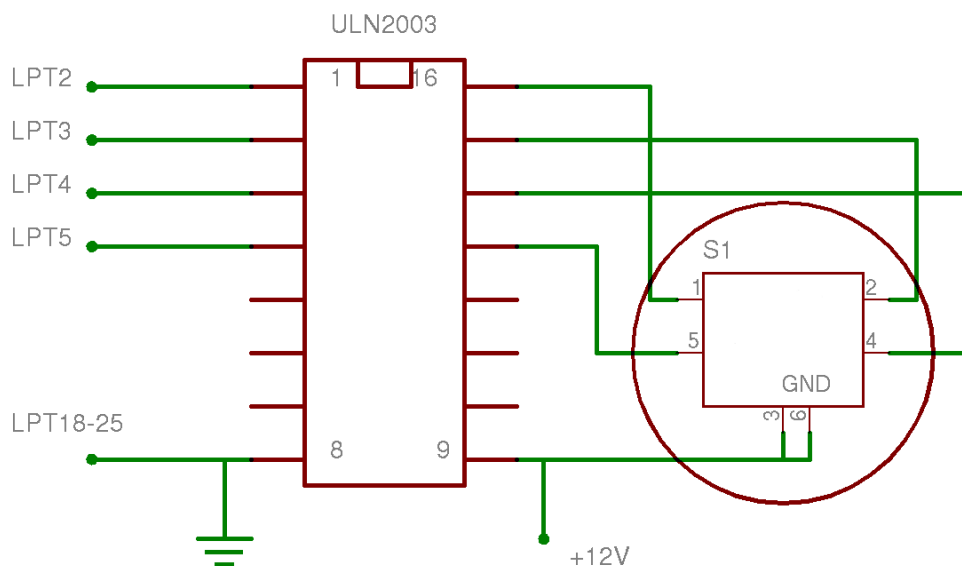


Pilt 2: Steppermootor

## 5.7.2 Unipolaarse steppermotori juhtimine

Unipolaarsed steppermotorid juhitakse kõrvuti olevate mähiste järgimööda pingestamise teel.

**Skeem LPT-pordi abil steppermotori juhtimiseks (Skeem 13):**



Skeem 13: Steppermotori juhtimine

ULN2003 on LPT-pordist tulevate signaalide võimendamiseks, kasutades välist toiteallikat. Siinses skeemis tarvivad nii ULN2003 kui ka steppermotor +12 volti. Kuid pinge võib olla paar kuni 24 volti, oluline on vaid jälgida, et mootori tarbitav vool liiga suureks ei läheks ehk mitte üle 0,5A. Selle skeemi võib realiseerida ka optroni ja transistoritega.

Stepperootoreid võib juhtida kasutades erinevaid loogikaid, mida siin ka käsitletakse.<sup>18</sup>

### 1. Steppermotori liigutamine täissammuga (*single stepping*)

Kõige lihtsam ja tavalisem on mootori liigutamine täissammuga (Tabel 4). Korruga on pingestatud olekus üks mähis, mis tõmbab enda poole mootori võlli küljes olevat magnetit ja seega liigutab ka võlli.

Kui järgimööda kõrvalasuvaid mähiseid pingestada ühes kindlas suunas teatud kiirusega, siis hakkabki mootor selles samas suunas pöörlema ehk sammuma.

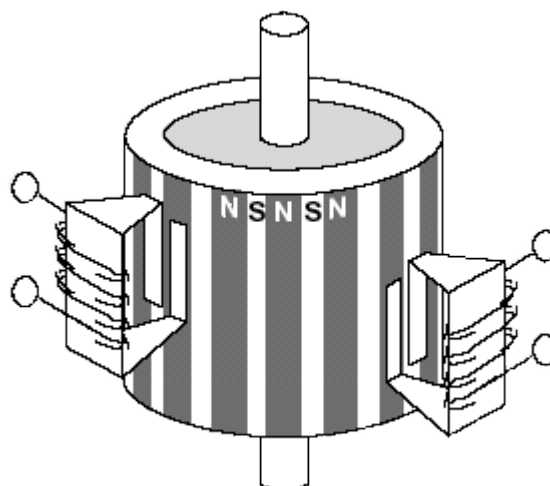
<sup>18</sup> Ian Harries - <http://www.doc.ic.ac.uk/~ih/doc/stepper/control2/sequence.html> (04.03.2006)

Samm	Juht 4	Juht 3	Juht 2	Juht 1	
a.1	on	off	off	off	
a.2	off	on	off	off	
a.3	off	off	on	off	
a.4	off	off	off	on	

Tabel 4: Stepermootori täis samm

Oluline on panna tähele, et enamasti nelja sammuga ei tee mootor täis pööret nagu seda eelnevatelt piltidelt näha on, sest mootori võlli ümber paikneb palju magnetiteid nagu seda ka jooniselt 4 näha on.

Põhimõtteline joonis stepermootori sisemusest<sup>19</sup>:



Joonis 4: Stepermootori sisemus

„N“ ja „S“ tähistavad vastavalt magnetite põhja- ja lõunapoolust.

<sup>19</sup> Electronics DIY-Team - [http://electronics-diy.com/electronics/stepper\\_motors.php](http://electronics-diy.com/electronics/stepper_motors.php) (07.03.2006)

Kui stepper mootori sammuks on näiteks 1,8 kraadi, siis täispöörde tegemiseks kulub 200 sammu.

**Ülesanne:** Koostada programm, paneks stepper mootori pöörlema ülalkirjeldatud viisil (täissammuga).

**Lahendus:**

```
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write() {lpt.output(Addr,datum); }

    public static void main( String args[] )
    {
        lpt = new pPort();
        Addr=0x378;

        while (true)
        {
            datum=(short)Integer.parseInt("00000001",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00000010",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00000100",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00001000",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
        }
    }
}
```

## 2. Steppermootori liigutamine täissammuga võimendatult (*high torque stepping*)

Antud loogika kohaselt (Tabel 5) on pingestatud korruga kõrvuti asetsevad mähised ja mootorit juhitakse kaks korda võimsamalt, kui eelmise skeemi puhul. Antud idee puhul on miinuseks see, et mootor tarvitab liikumisel rohkem voolu, kuid võlli pöörlemine osutus katsete käigus oluliselt jõulisemaks.

Samm	Juhe 4	Juhe 3	Juhe 2	Juhe1	
b.1	on	on	off	off	
b.2	off	on	on	off	
b.3	off	off	on	on	
b.4	on	off	off	on	

Tabel 5: Steppermootori täissamm võimendatult

**Ülesanne:** Koostada programm, paneks stepper mootori pöörlema ülalkirjeldatud viisil (täissammuga võimendatult).

**Lahendus:**

```
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write()
    {
        lpt.output(Addr, datum);
    }

    public static void main( String args[] )
    {
        lpt = new pPort();
        Addr=0x378;

        while (true)
        {
            datum=(short)Integer.parseInt("00001001",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00000011",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00000110",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}

            datum=(short)Integer.parseInt("00001100",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
        }
    }
}
```

### 3. Steppermootori liigutamine pooliku sammuga(*half stepping*)

Pooliku sammuga sammumist (tabel 6) võib osutada kasulikuks, kui on vaja, et mootori samm oleks poole väiksem kui terve. Sisuliselt on tegemist esimese ja teise liikumisloogika kombinatsiooniga. Pooled sammud tehakse võimendatult ja kasutades korraka kahte mähist. Ülejäänud vahepealsed sammud tehakse ühe mähisega.

Samm	Juhe 4	Juhe 3	Juhe 2	Juhe 1	
a.1	on	off	off	off	
b.1	on	on	off	off	
a.2	off	on	off	off	
b.2	off	on	on	off	
a.3	off	off	on	off	
b.3	off	off	on	on	
a.4	off	off	off	on	
b.4	on	off	off	on	

Tabel 6: Steppermootor pooliku sammuga

**Ülesanne:** Koostada programm, mis paneks stepper mootori pöörlema ülalkirjeldatud viisil (pool sammuga).

**Lahendus:**

```
public class Ylesanne
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write()
    {
        lpt.output(Addr, datum);
    }

    public static void main( String args[] )
    {
        lpt = new pPort();
        Addr=0x378;

        while (true)
        {
            datum=(short)Integer.parseInt("00000001",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
            datum=(short)Integer.parseInt("00000011",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
            datum=(short)Integer.parseInt("00000010",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
            datum=(short)Integer.parseInt("00000110",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
            datum=(short)Integer.parseInt("00000100",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
            datum=(short)Integer.parseInt("00001100",2);
            do_write();
            try{Thread.sleep(50);}catch(Exception e){}
        }
    }
}
```

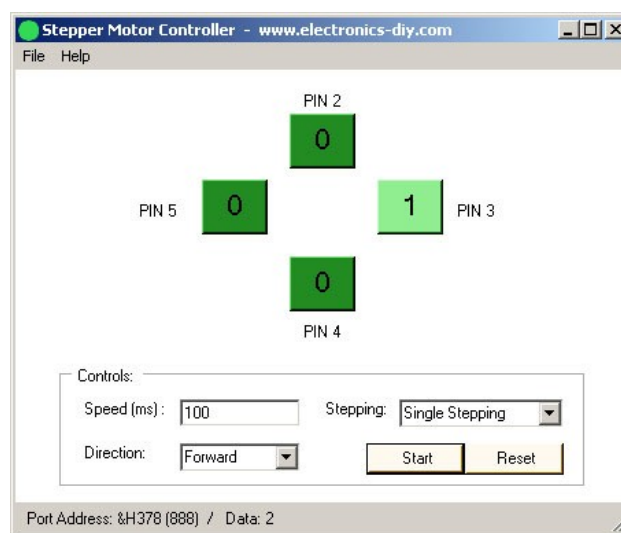
```

    datum=(short)Integer.parseInt("00001000",2);
    do_write();
    try{Thread.sleep(50);}catch(Exception e){}
    datum=(short)Integer.parseInt("00001001",2);
    do_write();
    try{Thread.sleep(50);}catch(Exception e){}
}
}
}

```

### 5.7.3 Juhtprogrammid

**Stepper Motor Controller**<sup>20</sup> (kuvapildistusel 7):



*Kuvapildistus 7: Stepper Motor Controller*

Stepper Motor Controller on programm stepper mootori juhtimiseks. Kasutajal lubatakse määrata suunda, kiirust ja sammumise viisi. Kasulik programm stepper mootoriga tutvumiseks ja ta omaduste väljaselgitamiseks. Näiteks võib teha mootoritega kaitseid, kui kiiresti nad on võimelised üldse pöörlema ning uurida erinevaid sammumisviise ja vaadelda, kuidas mootorid erinevates režiimides käituvad.

Internetist allalaetav leheküljel: <http://electronics-diy.com/electronics/downloads.php>

<sup>20</sup> Electronics DIY-team - [http://electronics-diy.com/electronics/schematics/stepper\\_motor\\_controller.jpg](http://electronics-diy.com/electronics/schematics/stepper_motor_controller.jpg) (02.02.2006)

## 5.8 LCD-mooduli juhtimine

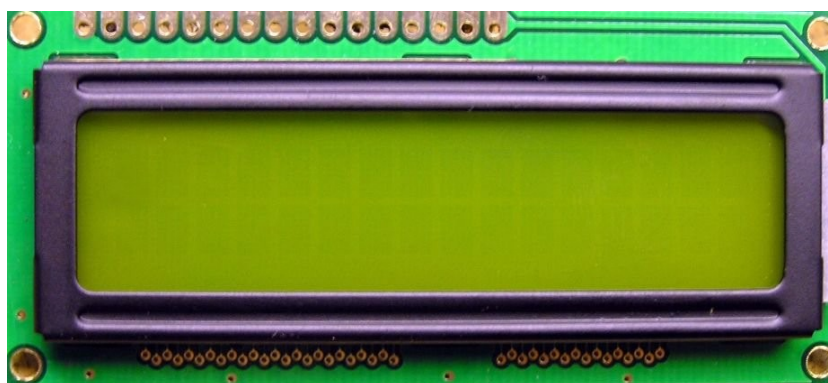
### 5.8.1 Ülevaade LCD'st

LCD-moduleid ehk vedelkristallekraane võib kohata kõikjal alates mobiiltelefonidest ning lõpetades külmkappide ja mikrolaineahjudega. Kuna tegemist on üldiselt suhteliselt standardsete seadmetega, siis on arvuti printeripordi abil erinevate LCD-moodulite juhtimine küllaltki lihtne, mikroprotsessoritega mõnevõrra keerulisem.

Tundub, et raalide tuunijatele meeldivad LCD-moodulid väga. Annab ju sinna kuvada parasjagu kasutatava mälu mahu, WinAmpis jooksva muusikapala pealkirja, protsessori temperatuuri ja koormuse, mälu hõivatuse ja palju muud huvitavat.<sup>21</sup>

LCD-ekraane on erinevate suurustega: 8x1, 8x2, 16x1, 16x2, 20x2, 20x4, 40x4. Esimene number märgib sümbolite arvu reas ja teine ridade arvu.

**Pildil 3 oleva LCD-ekraani suuruseks on 16x2:**

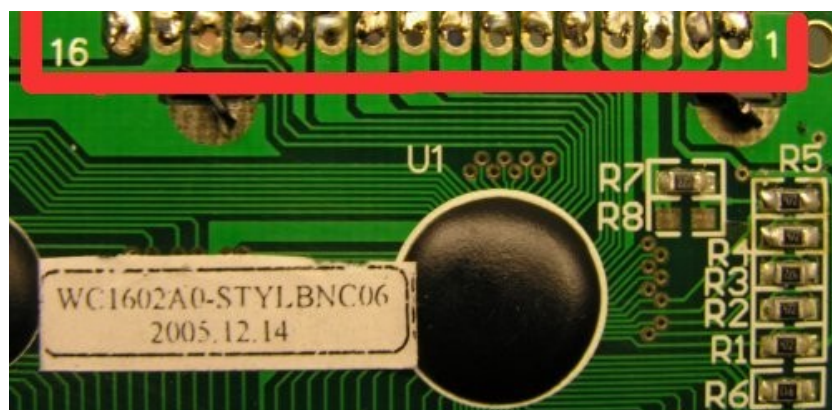


*Pilt 3: LCD-ekraan (16x2)*

#### **LCD-ekraani klemmid ja seletused:**

Heal juhul ja enamasti poest ostetud LCD-ekraanil on klemmid tähistatud (nagu pildil 4) 1..16. Järjekord on neil standartne, seega ühendamine vajaliku seadmega küllaltki lihtne.

<sup>21</sup> „Praktiline Arvutikasutaja“ 2003 aprill, lk 10



Pilt 4: LCD-ekraani klemmid

Peavalu võib valmistada mõni selline LCD-moodul, kus ei ole klemmidel numbreid märgitud. Sel juhul on kaks võimalust, kummalt poolt klemme ühendama hakata. Enamasti leiab maandusklemmi ehk esimese klemmi selle järgi üles, et ta on ka ühenduses korpusega.

### LCD-ekraani väljaviikude kirjeldused (Tabel 7) <sup>22</sup>:

Klemm	Seletus
1. Gnd	Maandus
2. Vcc	+5V toide
3. V0	Kontrasti määramiseks, V0 maandamise korral kontrast suurim.
4. RS	Register Select, loogilise väärtuse 1 ehk +5V korral kasutatakse sümbolite registrit(toimub sümbolite kuvamine ekraaniline) ja loogilise 0 korral kasutatakse käsu registrit(toimub käskude edastamine ekraanile).
5. R/W	Read/Write, loogilise 0 korral toimub ekraanile andmete kirjutamine, loogilise 1 korral andmete lugemine ekraanilt.
6. E	Enable, kui väärtus muutub 1 -> 0, siis ekraan võtab andmed vastu. Peale vajalike bittide paikaseadmist on vaja alati muuta Enable väärtus ühest nulliks, et ekraan taipaks ka andmed ilusti vastu võtta.
7-14. Data I/O	Andmete viigud, mida mööda toimub infovahetus LPT pordi ja LCD ekraani vahel.
15. Bl+	Taustvalguse +5V toide. Voolu on vaja piirata takistiga, kuid takisti väärtuse arvutamiseks on vaja teada valgusti maksimaalset lubatavat voolu ja Ohmi seadust( $I=U/R$ ). Kui on tegu tundmatu LCD'ga, siis võib katsetada erinevaid takisteid, alustades suurematest(~300 oomi) ning vähendades neid, kuni taustavalgus hakkab piisava tugevusega helendama. Kui ekraanil taustavalgust pole, siis puuduvad ka klemmid 15 ja 16.
16. Bl-	Taustvalguse maandusklemm.

Tabel 7: LCD-ekraani väljaviigud

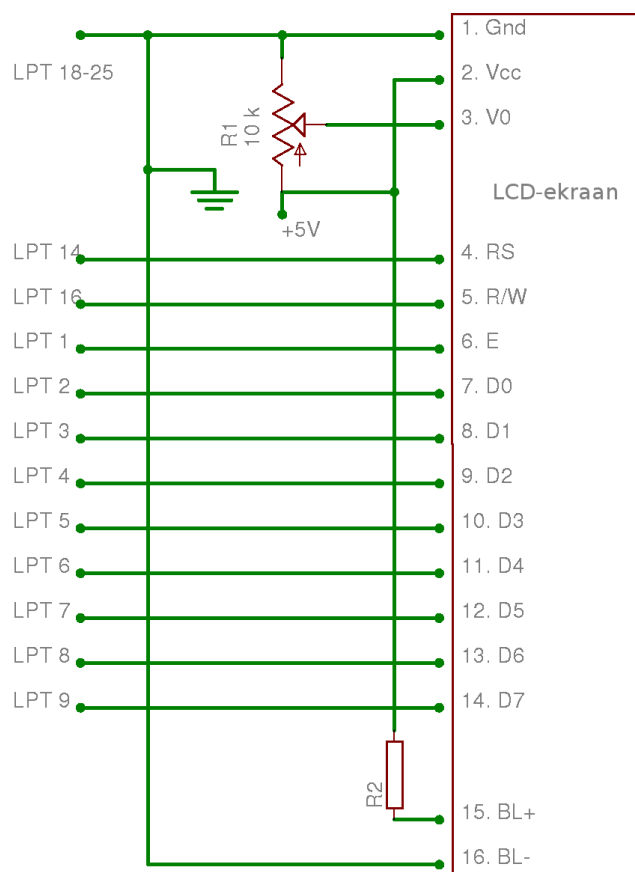
22 Fil - [http://www.repairfaq.org/filipg/LINK/F\\_LCD\\_progr.html](http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html) (30.04.2006)

## 5.8.2 LCD-ekraani juhtimine

Skeem LCD-ekraani arvuti printeriporti ühendamiseks on küllaltki lihtne. Piisab ainult kaabli õigesti ühendamisest, pole tarvis mikroskeeme, transistoreid, trükiplaati jne...

LCD-ekraaniga kommunikatsioon teeb lihtsamaks väga levinud Hitachi HD44780 kontrolleri, mis sisaldub enamustes LCD-ekraanides. Just tänu tolele kontrolleri on LCD-ekraanide juhtimine (sümbolite ning käskude edastamine) standardne. Hitachi kontrolleri spetsifikatsioon: <http://web.media.mit.edu/~ayah/documents/hd44780u.pdf>

**Skeem LCD-ekraani ühendamiseks LPT porti (Skeem 14)<sup>23</sup>:**



Skeem 14: LCD

Lisaks LPT-porti ühendamisele on vaja ka toitepinget +5V, mille saab otse arvuti toiteploki. Skeemis oleva potentsiomeetriga (R1) saab reguleerida pildi kontrasti, kuid klemmi 3 (V0) võib ühendada ka otse klemmi 2 (Vcc) külge, mis puhul on kontrastsus suurim.

<sup>23</sup> „Praktiline Arvutikasutaja“ 2003 aprill, lk 10

Teine skeemis olev takisti määrab LCD taustavalguse heleduse. Takisti väärtuse arvutamiseks on vaja teada Ohmi seadust ja valgusti maksimaalset lubatavat voolu. Kui takisti väärtuse arvutamine pole võimalik, siis võib proovida leida sobiva takisti katse-eksitus meetodil: alustada 500 oomisest takistist ja vähendada seda, kuni ekraan hakkab helendama.

LPT-pordi ja LCD-ekraani vaheline ühendus on 8 bitine – nii on kiirem ja mugavam andmeid edastada. Kui oleks tarvis ekraani mikroprotsessoriga juhtida, millel ei ole kaheksat väljaviiku, siis on ka võimalik 4 bitine kommunikatsioon.

Kui ekraanile tekkiv kiri peaks olema katkendlik, sisaldab arusaamatuid märke või muud sellist, on ühenduskaabel liiga pikk või toitepinge kehv. Häirete vältimiseks võib panna LCD viikude 3 ja 1 vahele kondensaatori mahtuvusega 0,1F ja viikude 1 ja 2 vahele elektrolüüt-kondensaatori mahtuvusega 10uF pingega 16V.<sup>24</sup>

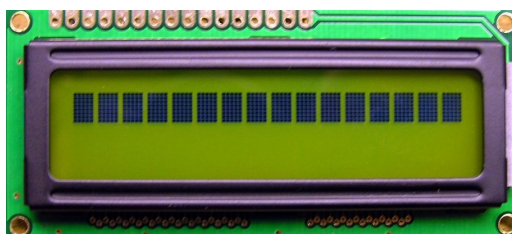
Enne ekraanile sümbolite kirjutamist on vaja LCD-ekraan initsialiseerida ehk algväärtustada.

Oluline on tähele panna, et peale iga sammu peab toimuma E muutmine ühest nulliks, et LCD saaks aru, millal on andmed valmis vastu võtmiseks.

Kuna LCD-moodulil kulub andmetele reageerimisele teatud aeg, siis ei tohi andmeid liiga kiiresti saata, sest muidu ei jõua kogu info kohale. Arvestada tuleks sellega, et pikemad operatsioonid(ekraani kustutamine) võtavad aega kuni 1,64ms ning lühemad(sümboli kirjutamine, mäluaadressi muutmine) umbes 40us.

### **Sammud LCD-ekraani initsialiseerimiseks<sup>25</sup>:**

1. **Toite sisselülitamine**, peale mida peaks kohe ekraanile ilmuma mustade ristkülikute rida (Pilt 5). See näitab ainult seda, et toide on peal.



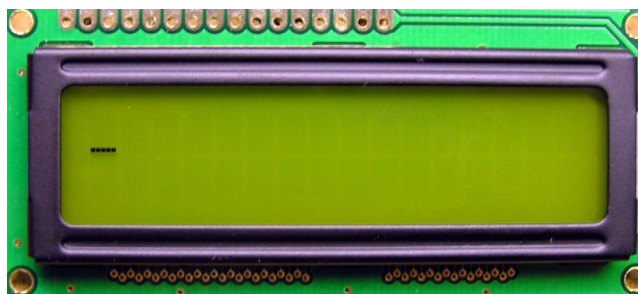
*Pilt 5: Toide peal*

<sup>24</sup> „Praktiline Arvutikasutaja“ 2003 aprill, lk 10

<sup>25</sup> Fil - [http://www.repairfaq.org/filipg/LINK/F\\_LCD\\_progr.html](http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html) (24.04.2006)

2. **R/W = 0** – Määrab ära andmete saatmise ekraanile, mitte lugemise. Peab edaspidi jäämaga nulliks.
3. **RS= 0** – Antakse teada, et ekraanile edastatakse käsklusi. Käskluste edastamise vältel peab RS olema kogu aeg null.
4. **Data=0x38** - Kuueteistkümnend kujul andmed 0x38 määravad, et kasutatakse 8 bitti, ekraanil on 2 rida ning 5x7 punkti formaadis toimub sümbolite kuvamine. Antud infot tuleb edastada 4 korda.
5. **Data = 0x06** - Sisestamisrežiim (Suurendus 1).
6. **Data = 0x0C** – Ekraan sees, kursor väljas, kursor ei plingi.
7. **Data = 0x01** – Ekraanilt kustutatakse kõik olemas olev tekst.
8. **Data = 0x80** – Ekraani esimese rea esimese sümboli aadress, kuhu võib kirjutada nüüd ühe märgi. Peale sümboli saatmist ekraanile muutub antud aadress automaatselt ühe võrra suuremaks ja seda ei ole vaja käsitsi muuta, kui tahetakse teksti järjest kirjutada. Ekraani teise rea esimese sümboli aadress on 0xC0 ja viimane 0xCF.

Peale ekraani initsialiseerimist on näha kursorit esimese rea esimese sümboli kohal (Pilt 6):



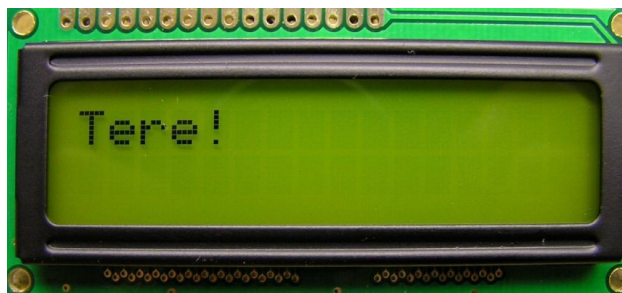
*Pilt 6: Kursor*

See tähendab, et nüüd võibki alustada sümbolite kirjutamist antud positsioonist. Selleks tuleb RS väärtus muuta üheks, et kasutataks sümbolite registrit andmete vastuvõtmisel. Peale seda võibki juba tähti või sümboleid andmetena ekraanile saata.

**„Tere!“ kuvamine ekraanile eeldab järgnevate sammude läbimist:**

1. **RS = 1, Data=0x54** – 0x54 on ASCII-tabeli „T“
2. **Data=0x65** - on ASCII-tabeli „e“
3. **Data=0x72** - on ASCII-tabeli „r“
4. **Data=0x65** - on ASCII-tabeli „e“
5. **Data=0x21** - on ASCII-tabeli „!“

**Ekraanipilt peale sammude läbimist (Pilt 7):**



*Pilt 7: Tere!*

**Järgnev autori väljatöötatud programmikood initsialiseerib LCD-mooduli ja võimaldab sinna teksti kuvada.**

```
public class Lcd
{
    static short datum;
    static short Addr;
    static pPort lpt;

    static void do_write() {lpt.output(Addr,datum);}

    static void inits() //initsialiseerib LCD ekraani
    {
        Kask((short)0x38); //8 data bitti, 2 rida, 5x7 punktine sümboli formaat
        Kask((short)0x38);
        Kask((short)0x38);
        Kask((short)0x38);
        Kask((short)0x06); //Kursori liigutamine ühe võrra
```

```

Kask((short)0x0E); //Ekraan sees, kursor sees, plinkimine väljas)
Kask((short)0x01); //Ekraani puhastus
Kask((short)0x80); //DD RAM Address esimese rea jaoks, 0xC0 on teine rida
}

```

//Võimaldab kerge vaevaga käske saata

```

static void Kask(short kask)
{
    Addr=0x378; //Pordi data address
    datum = kask; //käsu bait, 16nd süsteemis
    do_write(); //Pannakse valmis andmed
    Addr=0x37A; // Pordi kontroll register

    //Peale andmete valmispanekut määratakse ära, et tegu on käskudega ning
    toimub ühtlasi E muutus ühest nulliks, et LCD-moodul võtaks käsu vastu
    datum = 0xCA; //RS=0, R/W=0, E=1
    do_write();
    datum = 0xCB; //RS=0, R/W=0, E=0
    do_write();
    try{Thread.sleep(5);}catch(Exception e){} //Väike paus, et LCD-moodul
    jõuaks andmed vastu võtta
}

```

//Võimaldab kerge vaevaga üksikuid sümboleid kirjutada

```

static void Kirjuta(char symbol)
{
    Addr=0x378;
    datum=(short)symbol; //symbol
    do_write();

    Addr=0x37A;
    datum=0xCE; //RS=1, R/W=0, E=1
    do_write();
    datum=0xCF; //RS=1, R/W=0, E=0
    do_write();
    try{Thread.sleep(5);}catch(Exception e){}
}

```

//Pikema teksti lisamiseks, mis kasutab funktsiooni Kirjuta(char symbol) ning peale esimese rea täitumist(peale 16ndat sümbolit) alustab edasi kirjutamist teiselt realt

```

static void LisaTekst(String tekst)
{
    for (int u=0;u<32;u++ )
    {
        if (u+1>tekst.length()){break;}
        if (u==16){Kask((short)0xC0);} //liigutakse teisele reale
        Kirjuta(tekst.charAt(u));
        try{Thread.sleep(100);}catch(Exception e){} //tekitab väikse viivituse
        peale sümboli kuvamiseks. Jääb mulje nagu teksti trükitakse ekraanile.
    }
}

public static void main( String args[] )
{
    lpt = new pPort();

    inits(); //LCD mooduli initsialiseerimine

    LisaTekst("Esimene rida..."); //Jutumärkide vahel olev tekst kuvatakse
    ekraanile

    Kask((short)0xC0); //Viib kursori teise rea algusesse
    Kirjuta('1'); //Kirjutatakse teise rea algusesse „1“
    Kask((short)0xC7); //Viib kursori teise rea 8 sümboli kohale
    Kirjuta('2');
    Kask((short)0xCF); //Viib kursori teise rea viimase sümboli kohale
    Kirjuta('3');
}
}

```

**Programmi väljund LCD-ekraanil (Pilt 8):**



*Pilt 8: Programmi väljund*

Sümbolite jaoks on LCD-ekraanil olemas ASCII-tabel (Tabel 8) <sup>26</sup>:

Char. code	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
	0	0	0	1	1	1	1	0	0	1	1	1	1	1	
	0	1	1	0	0	1	1	1	1	0	0	1	1	1	
	0	0	1	0	1	0	1	0	1	0	1	0	1	1	
xxxx0000				0	a	P	\	P	-	9	E	o	P		
xxxx0001				!	1	A	Q	a	4	o	7	7	4	ä	Q
xxxx0010				"	2	B	R	b	r	"	ı	ı	ı	ı	ı
xxxx0011				#	3	C	S	c	s	ı	ı	ı	ı	ı	ı
xxxx0100				\$	4	O	T	d	t	\	ı	ı	ı	ı	ı
xxxx0101				%	5	E	U	e	u	ı	ı	ı	ı	ı	ı
xxxx0110				&	6	F	V	f	v	ı	ı	ı	ı	ı	ı
xxxx0111				'	7	G	W	w	7	7	ı	ı	ı	ı	ı
xxxx1000				(	8	H	X	h	x	ı	ı	ı	ı	ı	ı
xxxx1001				)	9	I	Y	y	ı	ı	ı	ı	ı	ı	ı
xxxx1010				*	:	J	Z	j	z	ı	ı	ı	ı	ı	ı
xxxx1011				+	;	K	[	k	[	ı	ı	ı	ı	ı	ı
xxxx1100				,	<	L	¥	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1101				-	=	M	]	m	)	ı	ı	ı	ı	ı	ı
xxxx1110				.	>	N	^	n	ı	ı	ı	ı	ı	ı	ı
xxxx1111				/	?	O	_	o	ı	ı	ı	ı	ı	ı	ı

Tabel 8: ASCII

„A“ on tuletatav järgmiselt:  
 Võtame „A“ tulba kohal bitid(0100) ja lisame neile „A“ rea ees olevad bitid(0001) ja saamegi „A“ = 01000001<sub>2</sub> ehk 0x41<sub>hex</sub>

„a“ on tuletatav järgmiselt:  
 Võtame „a“ tulba kohal bitid(0110) ja lisame neile „a“ rea ees olevad bitid(0001) ja saamegi „a“ = 01100001<sub>2</sub> ehk 0x61<sub>hex</sub>

Väiketäht on suurest tähest alati 0x20<sub>hex</sub> võrra suurem. Näiteks: „B“ = 0x42<sub>hex</sub> ja „b“ = 0x62<sub>hex</sub>

ASCII-tabeli olemasolu LCD-ekraanil teeb sümbolite kuvamise mitmeid kordi lihtsamaks, kuna personaalarvutigi sisaldab ASCII-tabelit. Pole vaja teha täiendavaid arvutusi: täht, mis on arvutis „a“ on sama koodiga ka LCD-ekraanil.

26 James Nevius - <http://www-inst.eecs.berkeley.edu/~ee40/calbot/webpage/ProgrammingLCD.htm> (30.04.2006)

## Käskude moodustamine tabelis 9 <sup>27</sup>:

Erinevaid käskusid on tabeli 9 järgi küllaltki lihtne moodustada. Kirjutamisoperatsioonide puhul peavad RS ja R/W olema nullid, ning ülejäänud andmebitid tuleb arvutada vastavalt tabelile 9.

RS R/W B7 B6 B5 B4 B3 B2 B1 B0	<i>Seletus</i>
0 0 0 0 0 0 0 0 0 1	Ekraani puhastamine
0 0 0 0 0 0 0 1 D S	Sisestamistüüp D: 0=suund -, 1=suund + S: 0=nihutuseta, 1=automaatne nihutus Kas üldse ja kuhu poole nihutatakse adressi, peale märgi sisestust. <b>0000111</b> <sub>2</sub> - andmete korral toimub automaatne nihutus ja liigutakse paremale poole.
0 0 0 0 0 0 1 D C B	Ekraan on/off D: 0=ekraan off, 1=on C: 0=kursor off, 1=on B: 0=vilkumine off, 1=on 1110 <sub>2</sub> - andmete korral on ekraan ja kursor sees, kuid kursor ei vilgu.
0 0 0 0 1 D N F X X	Andmeliides D: 0=8 bitt, 1=4 bitt N: 0=F works, 1=F is ignored F: 0=5x7 punkti, 1=5x10 punkti
0 1 B A A A A A A A	Loeb, kas on hõivatud ja adressi viida Kui B=1, ekraan on hõivatud.Tagastab viimase adressi viida.

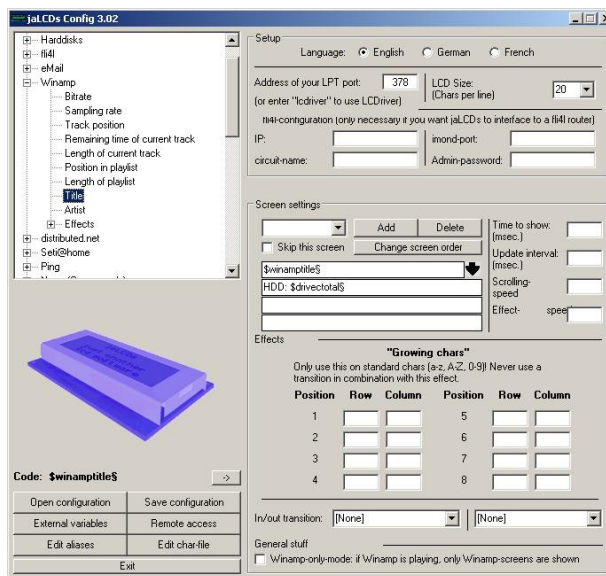
Tabel 9: Käskude moodustamine

### 5.8.3 Juhtprogrammid

Kes ise programmeerimisest mõnu ei tunne, võib kasutada juba valmisolevaid suurte võimalustega programme LCD-ekraanide juhtimiseks.

27 Fil - [http://www.repairfaq.org/filipg/LINK/F\\_LCD\\_progr.html](http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html) (30.04.2006)

## JALCD (Kuvapildistus 8)



Kuvapildistus 8: jalCD

Suudab kuvada ekraanile pea kõike, mida hing ihaldab: alates CPU kiirusest ja mälu hõivatuses ning lõpetades WinAmpi hetkel mängiva loo pealkirjaga.

Nii skeemi LCD-ekraani ühendamiseks LPT-porti kui ka programmi enda leiab aadressilt:

<http://jalcds.de/>

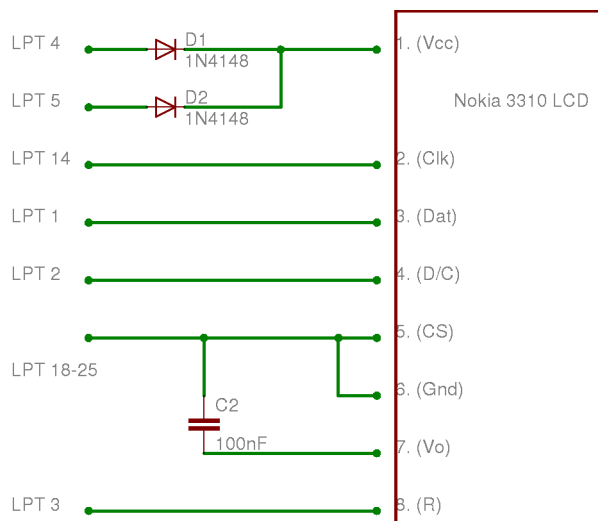
## LCDhype<sup>28</sup>

LCDhype nimeline vabavaraline programm saab hakkama ka vanast mobiiltelefonist (näiteks Nokia 3310) saadud ekraani juhtimisega, mistõttu pole tarvis osta poest hirmkallist LCD-ekraani.

Programmi uuema versiooni leiab aadressilt: <http://lcdhype.de/> Käivitamisel peab määrama ekraani tüübi (näiteks Nokia 3310).

Sõnumite manamiseks ekraanile on tarvis ühendada viimane LPT-porti ja käivitada programm.

Vajaliku juhtme skeem on küllaltki lihtne (Skeem 15):



Skeem 15: Nokia LCD

Skeem koosneb kahest diodist ja kondensaatorist ning lisaks printeripordi pistikupesast ja jupist kuuesoonelisest juhtmest.

## 5.9 Otsilloskoop

Otsilloskoop ehk otsillograaf on asendamatu abimees skeemis kulgevate signaalide kuju nägemisel. Näiteks saab temaga tavalise võimendi korrasolekut ja võimenduse kvaliteeti kontrollida: võimendi sisendisse lastakse signaaligeneraatori poolt genereeritud ühtlane ja puhas siinus-signaali ning otsillograafiga vaadatakse, kas võimendi väljundisse jõuab samasuguse kujuga signaal (nagu pildil 9<sup>29</sup>) või on tal moonutus.



Pilt 9: Siinus-singaal

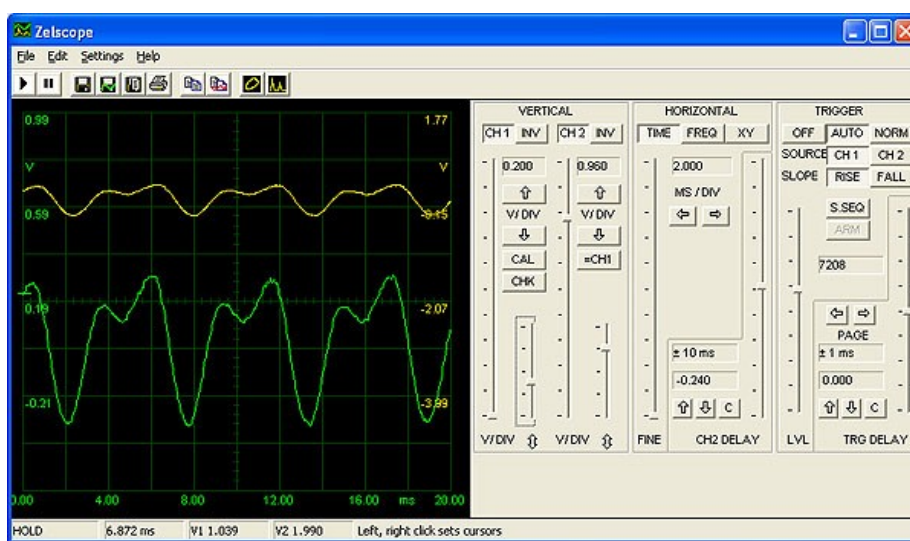
Peatükis tuleb juttu sellest, kuidas algajal on võimalik arvutiga töötav otsilloskoop ehitada, kuna poest ostetuna on nad väga kallid (mitukümmend tuhat krooni) ja tihtilugu pole algajal üldse

29 Elecp.com - <http://www.elecp.com/test/gos622g.jpg>

profiseadet vajagi. Otsilloskoobi näol ei ole niivõrd tegemist arvuti poolt juhitava seadmega, kui arvutit juhtiva seadmega.

<sup>30</sup>Lihtsama mäluga digitaalotsilloskoobi saab teha juba vähemalt 486 protsessoriga arvutist, millel on helikaart, sest just helikaardi kaudu liiguvad signaalid arvutisse. Vabavaralise programmi, mis teisendaks helikaardi mikrofoni sisendisse saabuval signaalid graafikuteks, saab aadressilt <http://www.zelscope.com>.

**Programm Zelscope (Kuvapildistus 9)<sup>31</sup>:**



Kuvapildistus 9: Zelscope

<sup>32</sup>Vaatamata suhtelisele lihtsusele on programm üsna võimas, lubades vaadata korraga kahte signaali, mis lastakse helikaardi paremasse ja vasakusse kanalisse, mõõta signaali pinget ja sagedust. Hoolimata programmi suurepärasest võimalusest on see siiski veidi aeglane ja vaadatav signaal ei saa olla suurema sagedusega kui umbes 22kHz - piiri seab kasutatav helikaart. Kindlasti ei tohi helikaardi sisendidesse anda pingeid üle 1V ning piirduda võiks näiteks 0,25 voldiga, et mitte helikaarti rikkuda.

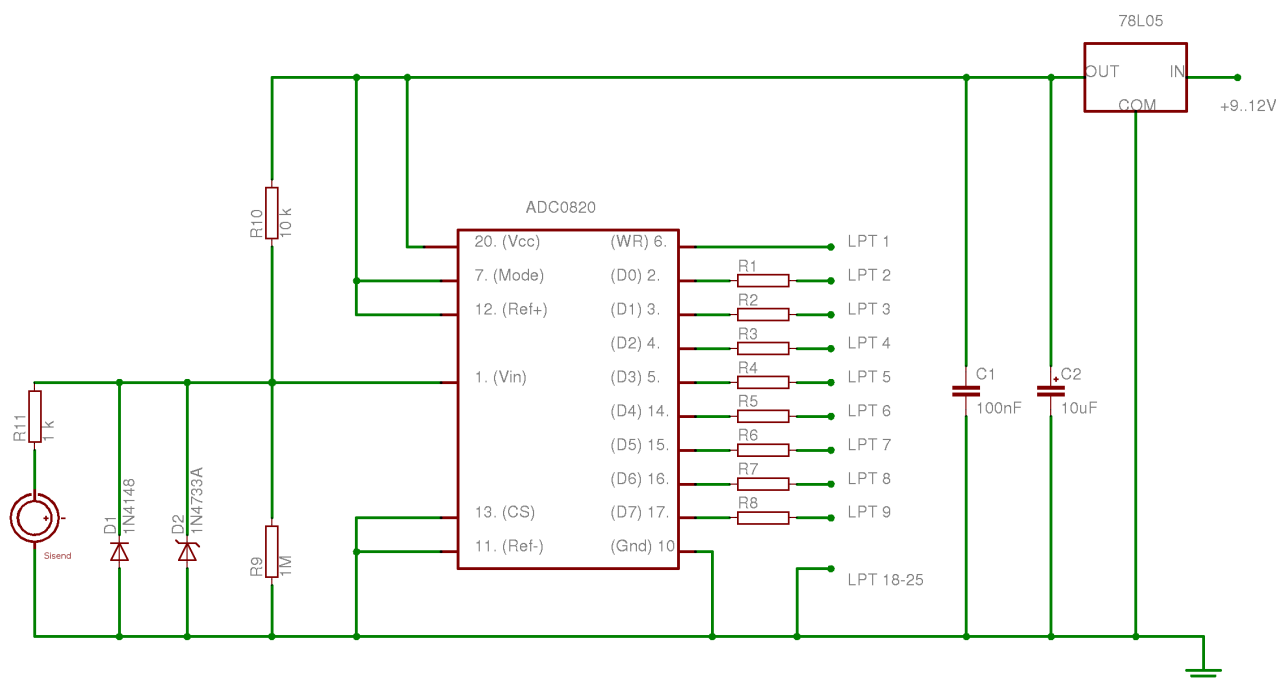
Kiirema otsilloskoopi saamiseks võib võtta kasutusele analoog-digitaalmuunduri mikroskeemi. Selle baasil ehitatud otsilloskoop on tunduvalt kiirem ja detailid odavad. Mõõteriista skeem ja tarkvara asuvad veebilehel [www.geocities.com/LPTScope/index.html](http://www.geocities.com/LPTScope/index.html).

30 „Praktiline Arvutikasutaja“ 2005 veebruar , lk 10

31 Zelscope - <http://www.zelscope.com/> (19.04.2006)

32 „Praktiline Arvutikasutaja“ 2005 veebruar , lk 10

## Analoog-digitaal muunduri skeem LPTScope programmi tarvis (Skeem 16) <sup>33</sup>(skeem ja kirjeldus):



Skeem 16: LPTScope

Skeem ehitatud analoog-digitaalmuunduri ADC0820 baasil ja on eelmisest(helikaarti kasutavast) tunduvalt kiirem. Skeem peaks olema jõu- ja taskukohane ka väheste kogemustega elektroonikule. Kõige kallim detail, mikroskeem, maksab umbes 75 krooni, mis on sadu kordi vähem, kui poest ostetud otsillograaf.

Skeem sisaldab põhimõtteliselt ainult muunduri mikroskeemi. Lisatud on sisendi kaitseahel takistitel R9, R10 ja R11, stabiltronil D2 ja diodil D1. Selle võib ka ära jätta, kuid peab olema kindel, et mitte kunagi ei hakata mõõtma 5V ületava amplituudiga signaale. Soovitav on siiski kaitseahel alles jätta.

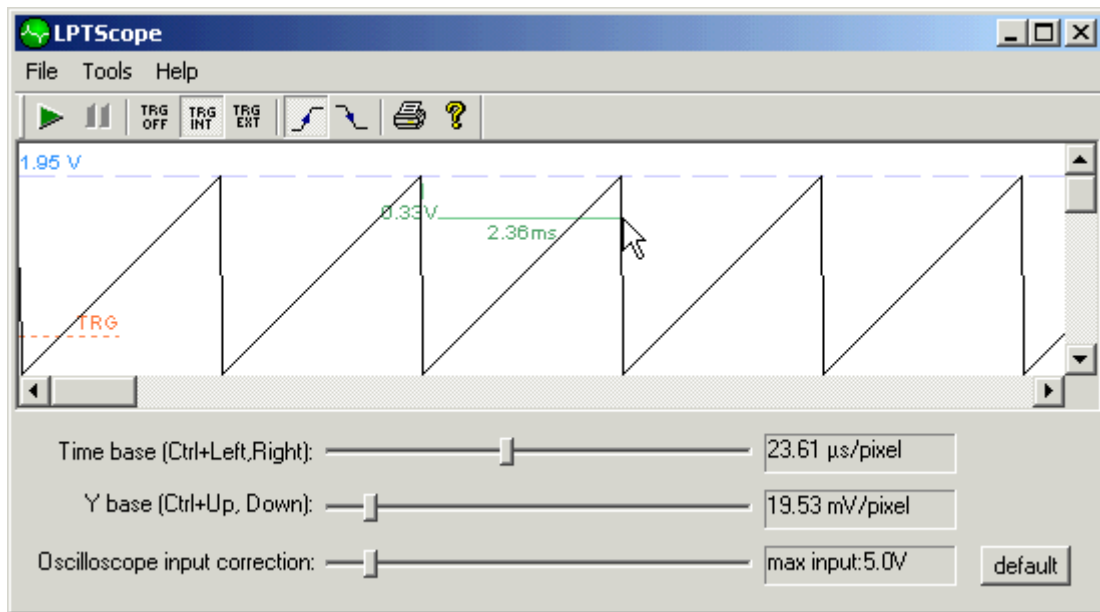
Takistid R1 ... R8 on arvuti printeripordi kaitseks juhul, kui see ei ole kahesuunaline – nii on vaid üsna vanadel arvutitel ja uutel siis, kui BIOS-ist vastavalt seada. 1 kilo-oomisest suurusest peaks piisama, et kaitsta porti ja muundurit, juhul kui viimane hakkab andmeid edastama ja port ei ole nende vastuvõtmiseks võimeline. Olles kindel, et arvutis on kahesuunaliseks sideks sobiv LPT-port, võib takistid skeemist välja jätta.

Otsilloskoopi sisendpinge ei tohi ületada 5V, sest rohkem mikroskeem ei talu. Loogikaskeemide jaoks sellest piisab, aga näiteks helivõimendis võib olla tunduvalt suurema amplituudiga signaale ning siis tuleks kasutada pingejagurit.

Stabilisaatorkivi 78L05 annab muundurkivile toidet 5 volti.

33 „Praktiline Arvutikasutaja“ 2005 veebruar , lk 10

## Programm LPTScope<sup>34</sup>:



Programm LPTScope on küll veidi tagasihoidlikuma tööekraaniga kui Zelscope, kuid tunduvalt kiirem. Võimaldab kuvada signaali pinget ja mõtte signaalilõikude kestvus.

34 LPTScope - <http://www.geocities.com/lptscope/hw.html> (19.04.2006)

## 5.10 Täienduseks

Lisaks käesolevas õppematerjalis olevatele skeemidele, võib hulgaliselt lisa ja õpetusi leida ka internetist, raamatutest ning ajakirjadest:

[www.tevalo.ee](http://www.tevalo.ee) – elektroonikakauplus

[www.opood.ee](http://www.opood.ee) – elektroonikakauplus

<http://www.tevalo.ee/foorum> – Eesti suurim elektroonikafoorum, kus leitakse vastus igale probleemile ja abistatakse lahkelt. Palju skeeme ja õpetusi.

<http://www.dcc.ttu.ee/Martin> - Skeemid, õpetused.

<http://www.hot.ee/evo14/> - Helitehnika skeemid.

<http://www.hot.ee/andy00/> - Skeemid ja õpetused.

<http://www.hot.ee/royalclub> - Skeemid ja lääne-vene asendused.

<http://www.epanorama.net> - Sisaldab hulgaliselt skeeme.

[www.howstuffworks.com](http://www.howstuffworks.com) – Erinevate seadmete tööpõhimõtete kirjeldused.

„Praktiline Arvutikasutaja“ - Ajakiri, kus ilmub üsna tihti elektroonikaskeeme ja õpetusi.

Jaak Pihlau „Digitaalelektronika käsiraamat“

Lembit Abo „Raadioseadmete üksikosad“

E. Aisberg „Transistor?...See on imelihtne!“ - Transistorite tööpõhimõtete lahkamine.

## 6. Kokkuvõte

Bakalaureusetöö raames sai koostatud LPT-porti ühendatavate lisaseadmete loomise õppematerjal, mis peaks olema heaks alustuseks riistvaraliste seadmete konstrueerimise ja nendevahelise suhtluse õppimiseks.

Õppematerjali läbinu peaks olema saanud piisavalt hea ettekujutuse arvutiga erinevate lisaseadmete juhtimise vallas ja oskama rakendada omandatud teadmisi mujalgi. Iseseisev edasiõppimine peaks minema ladusalt.

Tutvustatud ja kirjeldatud skeeme ning programmikoode saab edasi arendada ja mugandada endale käepärasteks – võib panna koduseadmeid juhtima või ehitada roboteid.

Kui algteadmised omandatud ning ennast ka täiendatud, võib hakata uurima näiteks mikroprotsessorite hingeelu. Nende programmeeriminegi käib valdavalt jada- või paralleelpordi kaudu. Vastava programmaatori võib ise meisterdada või poest osta. USB-porti või mõnda muusse moodsamasse porti käivaid odavaid programaatoreid teada ei ole ja enamuse neist kasutab siiski veel ainult LPT-porti, seega ei jäägi muud üle, kui kasutada toda vana head printeriporti.

Valminud õppematerjal peaks olema suureks abiks iseõppijale ja samuti võiks selle põhjal läbi viia valikkursusi nii ülikoolis kui ka keskkoolis või huvialaringides.

## 7. Summary

The following study material gives regulations and examples how to make and govern the devices that can be connected to the LPT Port. Despite the fact that the port of a printer is about to die out soon, it can still be used for studying how to control hardware equipment with a computer. Being obtained the basic knowledge, one can start attending more complicated, modern and faster interface such as USB, firewire etc. or start programming microprocessors with the help of LPT Port.

The following study material introduces LPT Port: its registers, pins and possibilities how to add LPT Port to a computer in case of its absence. Different circuits are described by starting with the governing of the Light-emitting diodes. It also displays the usage of input up to more difficult circuits that need external power.

The work also concentrates on governing stepper motors and LCD displays and the making of oscilloscope.

The thesis includes different programming tasks with answers and comments given. All the assignments are not solved in order to leave thinking space for the one who works through the paper. The student should have at least some basic knowledge in the field of electronics and the spirit to look up more material if necessary.

## 8. Kasutatud allikad

Epanorama.net - <a href="http://www.epanorama.net/circuits/parallel_output.html">http://www.epanorama.net/circuits/parallel_output.html</a> (11.01.2006).....	5
Howstuffworks.com - <a href="http://computer.howstuffworks.com/parallel-port.htm">http://computer.howstuffworks.com/parallel-port.htm</a> (11.01.2006).....	5
Epanorama.net - <a href="http://www.epanorama.net/circuits/parallel_output.html">http://www.epanorama.net/circuits/parallel_output.html</a> (21.04.2006).....	5
Howstuffworks.com - <a href="http://computer.howstuffworks.com/parallel-port1.htm">http://computer.howstuffworks.com/parallel-port1.htm</a> (11.01.2006).....	6
Linuxgazette.net - <a href="http://linuxgazette.net/118/misc/chong/parallel_port_diagram.jpg">http://linuxgazette.net/118/misc/chong/parallel_port_diagram.jpg</a> (05.01.2006).....	7
Wikipedia.org - <a href="http://de.wikipedia.org/wiki/Bild:Centronics.jpg">http://de.wikipedia.org/wiki/Bild:Centronics.jpg</a> (05.01.2006).....	7
Beyondlogic.org - <a href="http://www.beyondlogic.org/spp/parallel.htm">http://www.beyondlogic.org/spp/parallel.htm</a> (18.04.2006).....	9
Logix4u.net - <a href="http://www.logix4u.net/parallelport1.htm">http://www.logix4u.net/parallelport1.htm</a> (11.01.2006).....	9
Beyondlogic.org - <a href="http://www.beyondlogic.org/spp/parallel.htm#5">http://www.beyondlogic.org/spp/parallel.htm#5</a> (19.04.2006).....	9
Beyondlogic.org - <a href="http://www.beyondlogic.org/spp/parallel.htm">http://www.beyondlogic.org/spp/parallel.htm</a> (26.04.2006).....	10
Writelog.com - <a href="http://www.writelog.com/support/lpt_port_support_on_windows_nt.htm">http://www.writelog.com/support/lpt_port_support_on_windows_nt.htm</a> (20.01.2006).....	13
Douglas Beattie - <a href="http://hytherion.com/beattidp/comput/pport.htm">http://hytherion.com/beattidp/comput/pport.htm</a> (11.02.2006).....	17
AndyWorld - <a href="http://www.hot.ee/andy00/projektid/ele/lpt01.htm">http://www.hot.ee/andy00/projektid/ele/lpt01.htm</a> (11.01.2006).....	18
AndyWorld - <a href="http://www.hot.ee/andy00/projektid/ele/lpt02.htm">http://www.hot.ee/andy00/projektid/ele/lpt02.htm</a> (04.03.2006).....	25
AndyWorld - <a href="http://www.hot.ee/andy00/projektid/ele/spikker.htm">http://www.hot.ee/andy00/projektid/ele/spikker.htm</a> (17.03.2006).....	31
LCDprojects - <a href="http://www.alessioviti.com/lcdprojects/datasheet/2803.gif">http://www.alessioviti.com/lcdprojects/datasheet/2803.gif</a> (20.04.2006).....	35
PPMPS - <a href="http://ppmps.zesoi.fer.hr/projekt/video_nadzor/data/motor_files/image008.gif">http://ppmps.zesoi.fer.hr/projekt/video_nadzor/data/motor_files/image008.gif</a> (18.04.2006).....	36
Ian Harries - <a href="http://www.doc.ic.ac.uk/~ih/doc/stepper/control2/sequence.html">http://www.doc.ic.ac.uk/~ih/doc/stepper/control2/sequence.html</a> (04.03.2006).....	39
Electronics DIY-Team - <a href="http://electronics-diy.com/electronics/stepper_motors.php">http://electronics-diy.com/electronics/stepper_motors.php</a> (07.03.2006).....	40
Electronics DIY-team - <a href="http://electronics-diy.com/electronics/schematics/stepper_motor_controller.jpg">http://electronics-diy.com/electronics/schematics/stepper_motor_controller.jpg</a> (02.02.2006).....	46
„Praktiline Arvutikasutaja“ 2003 aprill, lk 10.....	47
Fil - <a href="http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html">http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html</a> (30.04.2006).....	48
„Praktiline Arvutikasutaja“ 2003 aprill, lk 10.....	49
„Praktiline Arvutikasutaja“ 2003 aprill, lk 10.....	50
Fil - <a href="http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html">http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html</a> (24.04.2006).....	50
James Nevius - <a href="http://www-inst.eecs.berkeley.edu/~ee40/calbot/webpage/ProgrammingLCD.htm">http://www-inst.eecs.berkeley.edu/~ee40/calbot/webpage/ProgrammingLCD.htm</a> (30.04.2006).....	55
Fil - <a href="http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html">http://www.repairfaq.org/filipg/LINK/F_LCD_progr.html</a> (30.04.2006).....	56
„Praktiline Arvutikasutaja“ 2005 august, lk 31.....	57
Elecp.com - <a href="http://www.elecp.com/test/gos622g.jpg">http://www.elecp.com/test/gos622g.jpg</a> .....	58
„Praktiline Arvutikasutaja“ 2005 veebruar, lk 10.....	59
Zelscope - <a href="http://www.zelscope.com/">http://www.zelscope.com/</a> (19.04.2006).....	59
„Praktiline Arvutikasutaja“ 2005 veebruar, lk 10.....	59
„Praktiline Arvutikasutaja“ 2005 veebruar, lk 10.....	60
LPTScope - <a href="http://www.geocities.com/lptscope/hw.html">http://www.geocities.com/lptscope/hw.html</a> (19.04.2006).....	61