

TAMPEREEN AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma  
Elektroniikan suuntautumisvaihtoehto

Tutkintotyö

Jarkko Tuomi

**INTERNETIN KAUTTA OHJATTAVA VALVONTAJÄRJESTELMÄ**

Työn ohjaaja:  
Tampere 2006

Yliopettaja Mauri Inha

# TAMPEREEN AMMATTIKORKEAKOULU

Tietotekniikka

Elektroniikka

Tuomi, Jarkko

Internetin kautta ohjattava valvontajärjestelmä

Tutkintotyö

31 sivua + 6 liitesivua

Työn ohjaaja

Yliopettaja Mauri Inha

Marraskuu 2006

Hakusanat

AVR, mikrokontrolleri, valvontajärjestelmä, askelmoottori, web-kamera

## TIIVISTELMÄ

Tässä työssä suunnitellaan ja toteutetaan mikrokontrolleripohjainen valvontajärjestelmä internet-sivulle, jonka kautta voidaan tarkastella valvontakamerakuvaa ja käännellä kameraa askelmoottorin avulla. Lisäksi järjestelmässä on mahdollisuus sammuttaa tai käynnistää erilaisia sähkölaitteita tai valvoa, ovatko laitteet kytkettyinä. Työ painottuu ensisijaisesti mikrokontrollerikortin suunnitteluun, ja siinä tutustutaan lähemmin Atmelin mikrokontrolleriin ATmega88. Lisäksi tarkastellaan askelmoottorin toimintaa ja sen ohjausta. Kortille suunnitellaan piirilevy ja valitaan sopivat oheiskomponentit. Kontrollerikortti on liitetty PC-tietokoneeseen, johon on asennettu Linux-käyttöjärjestelmä, ja työssä käydään pintapuolisesti läpi web-kameran asennus ja käyttö Linuxissa. Lisäksi työssä tutustutaan sarjaliikenteen toimintaan.

Mikrokontrollerikortin ohjelma toteutettiin C-kielellä ja kehitysohjelmistona käytettiin Codevision AVR C -kääntäjää. Kytkentä ja piirilevy toteutettiin Eagle Layout Editor -ohjelmistolla. Hallintasivu kirjoitettiin käyttämällä HTML- ja PHP-kieliä Linux-alustalle. Linuxista käytössä on Gentoo-jakeluversio ja WWW-palvelijana toimii Apache.

Myöhemmin laitetta on mahdollista helposti kehittää entistä monipuolisemmaksi liittämällä siihen esimerkiksi erilaisia antureita mittaamaan lämpötiloja tai antamaan hälytystietoja. Lisäksi laite on toteutettavissa itsenäiseksi laitteeksi niin, että se toimii verkossa WWW-palvelimena ilman erillistä PC-tietokonetta.

TAMPERE POLYTECHNIC  
Computer System Engineering  
Electronics

Tuomi, Jarkko  
Engineering Thesis  
Thesis Supervisor  
November 2006

Internet controlled monitor system  
31 pages + 6 appendicies  
Senior Lecturer Mauri Inha

Keywords

AVR, Microcontroller, Control System, Step Motor, Webcam

## ABSTRACT

The purpose of this work is to design and to process a microcontroller based on a control system for a web-page. On the web-page it is possible to turn round the surveillance camera with step motor. This system also has the possibility to turn on/off different electric apparatus. Additionally, the system is concentrating on designing the microcontroller card and we become familiar with the Atmel microcontroller ATmega88. It is also examined how the step motor works and how it is controlled. The circuit board is designed on to a card and suitable peripheral components are chosen. The controller card is connected to a PC which has a Linux operating system. The installation of the web-camera and the usage of the camera in Linux is introduced briefly in this work. The function of serial traffic is furthermore explained.

The program of the microcontroller card was accomplished with C-language and the development software was Codevision AVR C-compiler. The schematic and the circuit board was accomplished with Eagle Layout Editor Software. The controlling page was written by using the HTML- and PHP-languages on the Linux base system. The PC-computer uses the Gentoo Linux-distribution and the WWW-server is Apache.

The device is easy to develop more extensive in the future by connecting for example different sensors to measure temperatures or to give alarming information. The device can also be developed to an independent device that functions as the server in the net without the separate PC-computer.

## ALKUSANAT

Työn tekemiseen kului noin kolmen kuukauden intensiivinen ajanjakso. Työn tekeminen oli mielenkiintoista ja se perehdytti syvällisesti siihen, millainen käytännön elektroniikkaprojekti on. Lisäksi työ opetti käytännön ja teorian yhteen sovittamista. Erityisesti työssä auttoivat koulutukseen sisältyneet mikroprosessorin ohjelmointi- ja elektroniikan kurssit.

Tampereella 21. marraskuuta 2006

---

Jarkko Tuomi

## SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

ALKUSANAT

|   |    |
|---|----|
| SISÄLLYSLUETTELO.....                           | 5  |
| KÄYTETYT TERMIT JA LYHENTEET.....               | 6  |
| 1 JOHDANTO .....                                | 7  |
| 2 LAITTEISTON KUVAUS.....                       | 8  |
| 2.1 PC-palvelin.....                            | 9  |
| 2.2 Web-kamera.....                             | 10 |
| 3 MIKROKONTROLLERIKORTTI.....                   | 11 |
| 3.1 Komponenttivalinnat.....                    | 12 |
| 3.2 Kytkenän piirtäminen.....                   | 14 |
| 3.3 Kytkenän käytännön testaus.....             | 15 |
| 3.4 Piirilevyn suunnittelu.....                 | 18 |
| 4 OHJAUSKORTTI.....                             | 19 |
| 4.1 Komponenttivalinnat.....                    | 20 |
| 4.2 Askelmoottorin johtimien selvittäminen..... | 21 |
| 4.3 Kytkenän piirtäminen.....                   | 22 |
| 5 OHJELMISTON KEHITYS.....                      | 23 |
| 5.1 Mikrokontrollerikortin ohjelma.....         | 24 |
| 5.2 Sarjaportin tarkkailu-skripti.....          | 27 |
| 5.3 Käyttöliittymä.....                         | 28 |
| 6 YHTEENVETO.....                               | 29 |
| LÄHTEET.....                                    | 30 |

## KÄYTETYT TERMIT JA LYHENTEET

|          |   |
|----------|---|
| ASCII    | American Standard Code for Information Interchange, tietokoneiden standardiksi muodostunut merkistö |
| ATmega   | Atmelin yksi mikrokontrollerityyppi   |
| AVR      | Atmelin 8-bittinen mikroprosessoriperhe   |
| Bash     | GNU Bourne again shell, yleinen komentotulkki Linuxille   |
| CMOS     | Complementary Metal Oxide Semiconductor, kanavatransistoreihin perustuva mikropiiritekniikka        |
| EEPROM   | Electrically Erasable PROM, sähköisesti tyhjennettävä muisti  |
| Ethernet | LAN, pakettipohjainen lähiverkkoratkaisu  |
| Flash    | Puolijohdemuistityyppi  |
| Gentoo   | Suosittu Linux-jakeluversio   |
| ISP      | In-System Programmer, ohjelmointiväylä  |
| PHP      | Hypertext Preprocessor, yleinen WWW-ohjelmointikieli  |
| PWM      | Pulse Width Modulation, pulssin leveysmodulaatio  |
| RS232    | Recommended Standard, yleinen sarjaväyläarkkitehtuuri   |
| Skripti  | Script, komentojonotiedosto   |
| USART    | Universal Synchronous Asynchronous Receiver Transmitter, synkroninen ja asynkroninen ohjauspiiri    |
| USB      | Universal Serial Bus, uudehko sarjaväyläarkkitehtuuri   |
| WLAN     | Wireless Local Area Network, langaton lähiverkko  |

## 1 JOHDANTO

Työssä on tarkoitus suunnitella ja toteuttaa kodin hyötykäyttöön valvontasovellus, jonka hallinta on toteutettu internet-sivulle. Tältä internet-sivulta voidaan millä tahansa internet-yhteydellä varustetulla tietokoneella paikasta riippumatta seurata web-kameran lähettämää kuvaa ja käänellä kameraa eri suuntiin. Lisäksi sovellukseen suunnitellaan kodin erilaisten sähkölaitteiden valvontamahdollisuus niin, että niitä voidaan tarvittaessa käynnistää ja sammuttaa samaisen internet-sivun kautta.

Laitteeseen on helposti liitettävissä erilaisia varashälyttimiä, esimerkiksi liiketunnistin, joka hälytyksen sattuessa lähettää tiedon tekstiviestinä tai sähköpostina käyttäjälleen. Laitteella voidaan myös lukea helposti erilaisia lämpötilatietoja. Lisäksi askelmoottorin tilalle tai rinnalle voidaan liittää tasavirta- tai servomootoreita, joilla voidaan ohjata kodin mekaanista vääntöä tarvitsevia laitteita. Ulkoisten laitteiden ohjaukseen on myös mahdollista toteuttaa ajastimia. Näitä ideoita ei kuitenkaan toteutettu tässä työssä kireän aikataulun vuoksi.

Idea tämänkaltaiseen projektiin lähti siitä, että työn tuli olla haastava, koulutusta tukeva, hyödyllinen ja ennen kaikkea mielenkiintoinen. Piti siis miettiä sopiva elektroniikkaan liittyvä hanke, joka tulisi järkevän hintaiseksi toteuttaa kotikonstein ja josta voisi mahdollisesti olla vielä oikeata käytännön hyötyä.

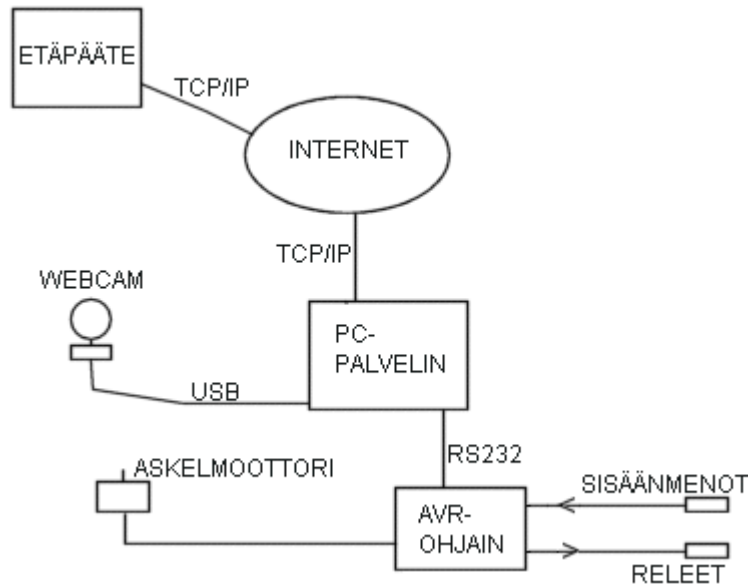
Tietoliikenneyhteyksien ja tietokoneiden yleistyessä tämänkaltaisen etäkäyttösovellus oli mahdollista toteuttaa edullisesti ja sille olisi tarvetta mahdollisesti muillakin käyttäjillä ja muissakin käyttökohteissa.

Koska internet-yhteys alkaa olla nykyään hyvin yleinen, on järkevää lähettää tiedot internet-sivulle, josta niitä voidaan lukea helposti esimerkiksi työpaikalta. Nykyisin alkaa olla myös mobiililaitteissa internet-yhteydet, joilla voi selata internetiä ja lukea sähköpostia. Mikäli kameraseurannalle ei ole tarvetta, markkinoilla on olemassa etäkäyttöisiä valvontalaitteita, jotka ovat ohjattavissa kännykällä lähetettävillä tekstiviesteillä, mutta nämä laitteet ovat hankalia käyttää, ja puhelinoperaattori yleensä veloittaa lähetetyistä tekstiviesteistä kertamaksun.

Internet-yhteys on sen sijaan kiinteästi hinnoiteltu ja kuuluu entistä usean kotitalouden perusvarustuksiin, joten tämä sovellus mahdollistaa oman internet-yhteyden vielä tehokkaamman hyötykäytön ilman erillisiä lisäkustannuksia. Tässä työssä toteutetusta sovelluksesta on kuitenkin vielä matkaa käyttäjäläheiseksi laitteeksi, mutta tarpeen vaatiessa se on kehitettävissä eteenpäin, sillä kukapa ei kotoa lähtiessään miettisi, jäikö kahvinkeitin tai hellan levy kytketyksi. Tämän voisi tarkistaa työpaikan tietokoneelta ja tarpeen vaatiessa sulkea käyntiin jääneen kahvinkeitin.

## 2 LAITTEISTON KUVAUS

Sovellus vaatii PC-tietokoneen, laajakaistaisen internet-yhteyden, erillisen PC-koneeseen liitettävän web-kameran, mikrokontrollerikortin, joka ohjaa askelmoottoria ja releitä sekä lukee ulkoisten laitteiden tiloja. Kuvassa 1 on esitetty toiminnasta periaatekuva lohkokaaviona.



**Kuva 1** Lohkokaaviokuva

Etäpäätte kuvaa mitä tahansa internettiin liitettyä tietokonetta tai laitetta, jolla voidaan selata HTML-pohjaisia sivuja. PC-palvelintietokone liitetään internetiin ja se alustetaan toimimaan WWW-palvelijana. PC-koneeseen liitetään web-kamera USB-liitännän kautta lähettämään kuvaa koneen kovalevylle. Mikrokontrollerikortti liitetään PC-koneeseen RS232-sarjaliitännän kautta.



## 2.1 PC-palvelin

Palvelinkoneena toimii PC-tietokone, jossa on Intel Pentium 2/450 MHz:n suoritin ja 64 Mb:n RAM-muisti. Käyttöjärjestelmäksi on asennettu Gentoo Linux-jakeluversio, joka on optimoitu PC-koneen laitteiston ominaisuuksille kääntämällä itse kaikki lähdekoodi.

Linux-käyttöjärjestelmä on hyvä valinta käytettäväksi tässä työssä, sillä se on täysin ilmainen, vakaa ja tehokas järjestelmä, johon palvelinohjelmistoja ja käyttöoppaita on saatavissa laajalti internetin välityksellä. Linux on alun perin suunniteltu monen käyttäjän käyttöjärjestelmäksi ja näin ollen etäkäyttö on pyritty luomaan mahdollisimman helpoksi käyttämällä virtuaalikonsoleita. Linux käsittelee Unix-tyyppisesti myös laitteistoa tiedostoina, siten esimerkiksi sarjaportin käsittely on yksinkertaista, sillä sarjaporttiin kirjoittaminen tapahtuu samaan tapaan kuin tiedostoon. /1/

Sovelluksen käyttöliittymä kirjoitetaan HTML-koodilla WWW-sivuksi palvelimelle, jonka Linux-koneessa oleva palvelinohjelmisto välittää eteenpäin internetin yli luettavaksi. WWW-palvelinohjelmistoksi valittiin ilmainen, laajalti levinyt Apache versio 2.0, jossa on tuki erilaisille internetissä käytetyille skriptikielille, kuten PHP ja PERL. Pelkällä HTML-koodilla ei saada aikaan WWW-sivulle toiminnallisuutta, joka lähettäisi sarjaportille tietoa ja lukisi sitä. Tähän tarkoitukseen valittiin PERL-sukuinen tulkattava komentosarjakieli PHP, josta Linux-koneelle asennettiin ilmainen PHP-tulkin versio 5.0.

## 2.2 Web-kamera

Web-kamerana toimii USB-liitäntäinen Logitechin valmistama Quickcam-peruskamera, joka toimii CMOS-sensoritekniikalla ja pystyy ottamaan 30 kuvaa sekunnissa 640 x 480 pikselin resoluutiolla. CMOS-sensoritekniikkaa käyttäen tuodaan jokainen pikseli omana signaalinaan, joten kuvasta voidaan rajata mielivaltainen alue tarkkailtavaksi. Näin on mahdollista tunnistaa liikettä ja rajata liikkeen tarkkailu tiettyyn alueeseen /2/. Laitteen valmistaja ei tue Linuxia, joten valmiit ajurit kameralle ovat vain Windows-käyttöjärjestelmille. Kameralle löytyy kuitenkin harrastelijoiden valmistamat Linux-ajurit (spca5xx), joilla kuvan suurin tarkkuus on 352 x 288 pikseliä, vaikka kamera vielä tarkempaan pystyisikin. Spca5xx-ajureiden valmistajan mukaan on mahdoton päästä parempaan suorituskäyttöön ilman laitteen valmistajan luovuttamia tarkkoja teknisiä dokumentteja /3/.



**Kuva 2** Logitech Quickcam web -kamera

### ASENNUS /4/

USB-liitäntäisen web-kameran asentamista varten kernelissä pitää olla tuki Video4Linux-rajapinnalle ja tietysti usb:lle. komennolla `lsusb` voidaan tarkistaa, onko kone tunnistanut kameran, ja kun kamera on löytynyt pitäisi tulostua:

```
Bus 001 Device 003: ID 046d:092c Logitech, Inc.
```

ID-tunnus kertoo heksadesimaalimuodossa kameran valmistajan ja tarkan merkin.

ID-tunnuksen perusteella voidaan tarkistaa mitkä ajurit kameralle tulee asentaa.

Gentoo-järjestelmässä ajureiden asennus tapahtuu lataamalla ja kääntämällä ensin paketti komennolla: `emerge spca5xx`. Tämän jälkeen ladataan ajurimoduuli

komennolla: `modprobe spca5xx` ja latauksen onnistuminen voidaan tarkistaa komennolla `lsmod`. Aiemmin todettiin, että Linux käsittelee laitteita tiedostoina.

Kun kamera on asennettu onnistuneesti, on laite käytössä tiedostossa

`/dev/video0`, jota lukemalla voidaan tulostaa kuva näytölle tai tiedostoon käyttäen sopivaa ohjelmaa. Kuvan lukeminen kamerasta ei vaadi erillistä graafista ohjelmistoa, sillä kameran lähettämä data voidaan lukea ja kirjoittaa suoraan kovalevylle haluttuun tiedostoon. Ainoastaan kuvan katseleminen vaatii grafiikkatilan, jotta kuva voidaan tulkita tietokoneen näytölle.

## OHJELMISTO

Kuvankaappausohjelmaksi asennettiin kevyt W3Cam-ohjelma, joka toimii komentorivipohjaisesti eikä graafista käyttöliittymää tarvita. Ohjelman voidaan käskellä hakemaan kamerasta joko yksittäinen kuva jpg/png-muodossa tai lähettämään jatkuva-aikaista videokuvaa (streamia) mpeg-muodossa. Stream lähetetään käyttäjän valitsemaan porttiin, johon vieraskone voi liittyä verkon yli katselemaan videokuvaa. Jos käsketään kameran hakea hyvälaatuinen kuva jpg-muodossa tiedostoon, se tehdään käyttämällä vidcat-skriptiä komennolla:

```
vidcat -d /dev/video0 -f jpeg -q 100 -o kuva.jpg
```

Mikäli kameraa halutaan käyttää erityisesti valvontaan, voidaan käyttää ohjelmallista liikkeentunnistusta. Liikkeentunnistusohjelma pyrkii vertailemaan pikseleiden muutoksia kahdesta peräkkäisestä kuvasta ja jos kuvat eroavat riittävästi toisistaan, liike on tunnistettu. Esimerkiksi Linuxille on olemassa ilmainen Motion-niminen liikkeentunnistusohjelmisto ja se voidaan asettaa lähettämään kuva palvelimelle aina, kun liikettä havaitaan. Jos käytössä on graafinen käyttöliittymä, voidaan sen avulla rajata kuvasta tiettyjä alueita, joissa liikettä tarkkaillaan, esimerkkinä oven kahvan liike. /2/

## 3 MIKROKONTROLLERIKORTTI

Mikrokontrollerikortille päätettiin suunnitella ja toteuttaa piirilevy itse. Komponenteiksi valittiin reikiin asennettavat mallit, jotta käsinjuotto olisi mahdollisimman helppo toteuttaa. Samalla suosittiin mikropiirejä, jotka eivät vaatineet juurikaan ulkoisia komponentteja, jotta kortin fyysinen koko pysyy järkevän pienenä ja hinta pysyy järkevän edullisena. Työssä tarvittava moottorin ja releiden ohjaus suunniteltiin erillisenä kytkentänä oheiskortiksi, jonka voi liittää emokortin laajennusliitimeen. Näin on mahdollista liittää mikrokontrollerikortille myös muita oheiskortteja tarpeen ja sovelluksen mukaan. Lisäksi itse emokortille ei tarvitse tuoda kuin 5 V käyttöjännite, sillä releet ja moottorit tarvitsevat usein 12 V jännitteen.

### 3.1 Komponenttivalinnat

Ensimmäinen vaihe suunnittelussa oli valita käytettävä mikrokontrolleri ja lähteä kehittämään korttia sen ympärille. Markkinoilla on lukuisia erilaisia mikrokontrollereita, mutta lopulta päädyttiin Atmelin valmistamaan 8-bittiseen ATmega88-kontrolleriin, joka on hintaansa nähden hyvin suorituskykyinen eikä tarvitse ulkoista kellokidettä, tästä mallista löytyy myös reikiin asennettava pakkaus. Ohjaimen valintaa puoltavat myös ilmaiset kehitysohjelmistot, kääntäjät ja simulaattorit sekä internetissä oleva suuri tietomäärä. ATmega88-mikrokontrollerin ominaisuuksia:

- Flash muistia 8 kB, 10 000 kertaa uudelleen kirjoitettavissa
- EEPROM muistia 512 B
- SRAM käyttömuistia 1024 B
- 32 kpl 8-bittistä yleisrekisteriä
- Kellotaajuus 0-20 MHz, sisäistä oskillaattoria käytettäessä max. 8 MHz
- Käyttöjännite 2,7 – 5,5 V
- 23 I/O-nastaa, jotka ohjelmoitavissa tuloiksi tai lähdöiksi
- 8-kanavainen 10-bitin AD-muunnin
- USART-sarjaliikenne (RS-232)
- ISP-ohjelmointiliitäntä /5/

AVR-mikrokontrolleri ei tarvitse ympärilleen juurikaan oheiskomponentteja. Käyttöjännite kannattaa kuitenkin suodattaa jänniteregulaattorilla ja muutamalla kondensaattorilla. Regulaattoriksi valittiin 7805-sarjan TS7805CZ-komponentti TO220-kotelossa, ja sen virtakestoisuus on riittävä 1 A. Valmistaja suosittelee regulaattorin tuloon ja lähtöön kondensaattoreita suodattamaan virran muutoksia ja estämään regulaattorin värähtely. Virransyötön rinnalle tulopuolelle valittiin 220 µF elektrolyyttikondensaattori. Lisäksi regulaattorin tuloon ja lähtöön valittiin kaksi 150 nF muovikondensaattoria. /6/

Lisäksi kortti tarvitsee RS232-sarjaliikennöintiä varten PC-tietokonekytkentää varten. Piirin tarkoitus on muuntaa mikrokontrollerin 5 V jännitte sarjaliikenteen vaatimalle 12 V tasolle ja päinvastoin. /7, s.15./ Dallasin valmistama DS275 ei tarvitse lainkaan ulkoisia komponentteja, joten se on hyvä valinta kytkentään /8/.

Kortti tarvitsee myös ohjelmointiliittimen, jonka avulla mikrokontrollerin flash-muisti voidaan ohjelmoida PC-tietokoneella kontrollerin ollessa kytkettynä kortilla. ATmega88 voidaan ohjelmoida ISP-liitännän kautta PC-tietokoneen rinnakkaisporttiin liitetyllä kaapelilla. Liitännään tulee neljä kytkentäjohdinta ja maajohdin. Tiedonsiirto on kaksisuuntaista ja tarvitsee toimiakseen kellopulssin, joka tuodaan PC-koneelta nastaan SCK. Yksi johdin kytketään RESET-nastaan vetämään RESET-linja ohjelmoinnin ajaksi maahan. MOSI-nastaan tuodaan PC-koneelta ohjelmointitieto, paluutieto ja kättely tapahtuu MISO-nastan kautta kontrollerilta PC-koneelle. Lisäksi tulee laitteiden maat yhdistää. Johtimet suojataan  $390 \Omega$  erotusvastuksilla niin, ettei PC-tietokoneen rinnakkaisportti häiritse mikrokontrollerin toimintaa ajon aikana, mikäli ohjelmointijohto on kytkettynä. Tarkka johdon kytkentä on esitetty myöhemmin. /7, s.13./

Kortille lisätään vielä kaksi opto-erotettua tuloa ulkoisten laitteiden tilojen seuraamista varten. Opto-erotus erottaa ja samalla suojaa mikrokontrolleria ulkoisten laitteiden signaaleilta, näin mahdollisten jännitepiikkien vuoksi ei rikkoudu mikrokontrolleri vaan pelkästään opto-erotin. Komponentiksi valittiin CNY17-3Z. Opto-erottimen ledin vaatima virta on tyypillisesti 2-20 mA. Piirin datalehti ilmoittaa ledin kynnysjännitteen olevan tyypillisesti 1,35 V, joskin tämä on mitattu 60 mA virralla, mutta datalehdeltä löytyvän kuvaajan mukaisesti kynnysjännite ei merkittävästi muutu käyttövirran muuttuessa, joten käytetään tyypillistä arvoa. Kun tulo valitaan 5 V jännitetasolle ja ledin virraksi valitaan 5 mA, voidaan etuvastukset mitoittaa kaavalla:

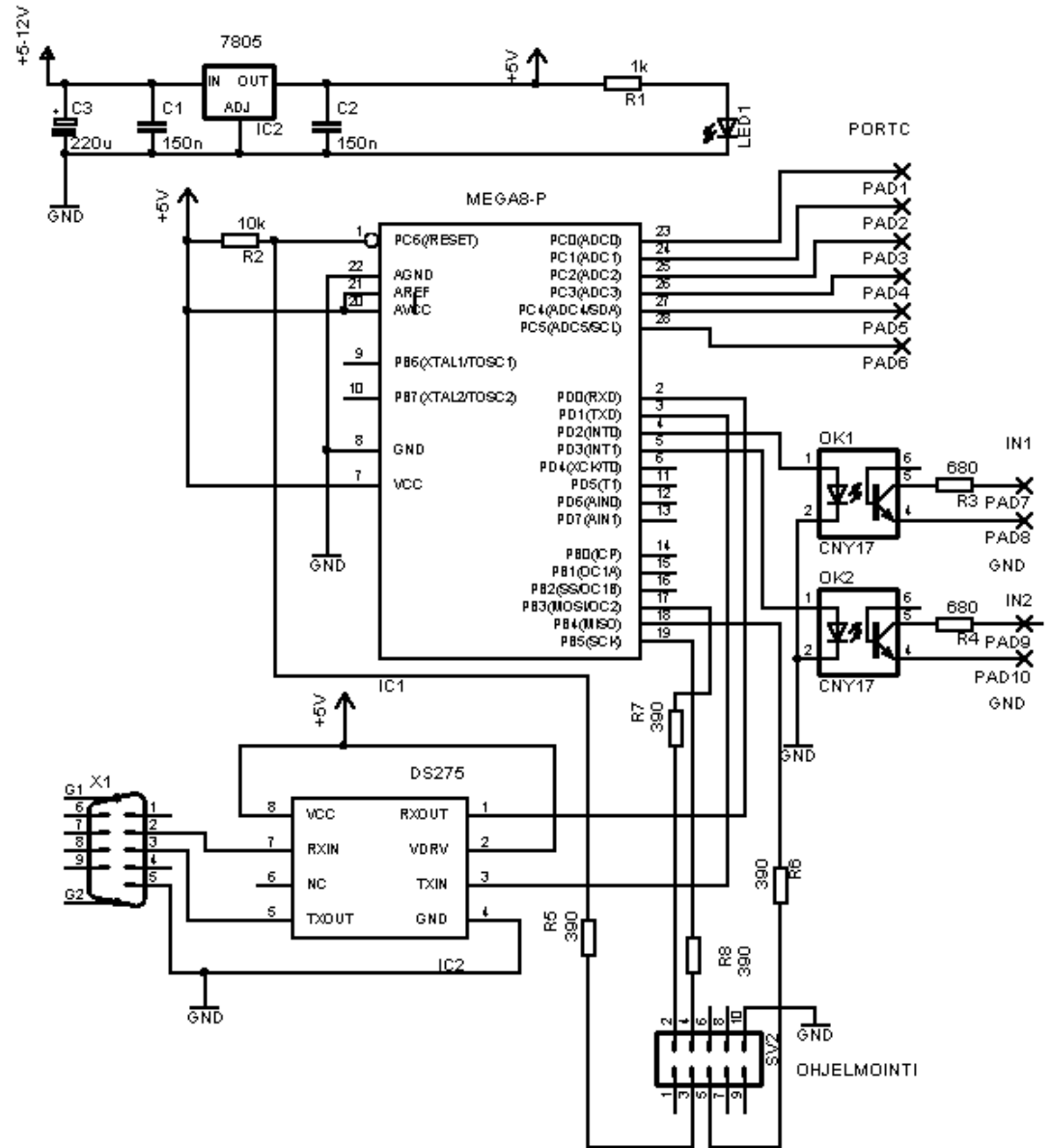
$$R = \frac{5,00 \text{ V} - 1,35 \text{ V}}{5 \text{ mA}} = 750 \Omega \quad (1)$$

Valitaan vastukseksi E-sarjasta löytyvä arvo  $680 \Omega$ . /7, s.10; 9/

### 3.2 Kytkenän piirtäminen

Kytkenän piirtämiseen ja suunnitteluun käytettiin ilmaisversiota Eagle Layout Editor 4.16r1-piirilevysuunnitteluohjelmistosta. Ohjelmalla voidaan piirtää kytkenä schematic-toiminnolla, joka sitten siirretään ohjelman piirilevyn suunnittelupuolelle. Piirilevy voidaan tarvittaessa suunnitella myös ilman alustavaa kytkenäpiirrosta suoraan board-toiminnolla. Ohjelmasta löytyy valmiina laaja kirjasto erilaisille komponenteille ja näiden lisäksi Eaglen omilta internet-sivuilta on ladattavissa suuri määrä harrastelijoiden tekemiä kirjastoja. Komponenttimalleja on myös mahdollista itse tehdä ohjelman Library-editorilla. Eaglen ilmaisversiolla voi suunnitella vain yksisivuisia kytkenäkaavioita ja suurin alue, jolle komponentteja voi asetella, on 80 mm x 100 mm (1/2-E1). Lisäksi ilmaisversiosta on rajattu pois monikerrospiirilevyjen tekeminen, joten levystä voidaan suunnitella vain yksi- tai kaksipuolinen. Nämä rajoitukset eivät kuitenkaan haittaa tämän työn levyn suunnittelua.

Kytkenään lisättiin merkkiledi 1 k $\Omega$  etuvastuksella osoittamaan käyttösjähk $\ddot{o}$ . Etuvastus mitoitettiin kaavan 1 mukaisesti. AVR:n RESET-linja on toiminnaltaan k $\ddot{a}$ anteinen, eli se reagoi loogiseen nollaan. RESET-linja on vedetty 10 k $\Omega$  vastuksen kautta käyttöj $\ddot{a}$ nnitteeseen, joten se pysyy vakaasti tilassa yksi, n $\ddot{a}$ in estet $\ddot{a}$ n ep $\ddot{a}$ m $\ddot{a}$ ar $\ddot{a}$ iset tilat ajon ja ohjelmoinnin aikana. Nastat AREF ja AVCC tulee kytke $\ddot{a}$  kontrollerin datalehden mukaan käyttöj $\ddot{a}$ nnitteeseen, vaikka ad-muunninta ei k $\ddot{a}$ ytet $\ddot{t}$ aisik $\ddot{a}$ n. /5/ Eagle-ohjelmasta ei l $\ddot{o}$ ytynyt valmiiksi kirjastoa DS275-piirille, joten se jouduttiin luomaan itse k $\ddot{a}$ ytt $\ddot{a}$ m $\ddot{a}$ ll $\ddot{a}$  library-editoria. Sarjaliit $\ddot{a}$ nt $\ddot{a}$  varten valittiin 9-napainen D-liitin. Laajennusliittimeen on vedetty kuusi I/O-ohjausta kontrollerin portista C. Askelmoottorille riitt $\ddot{a}$  yleens $\ddot{a}$  nelj $\ddot{a}$  ohjaussignaalia, joten kaksi ohjausta j $\ddot{a}$ e viela releille. Lopullinen kytken $\ddot{a}$ kaavio on esitetty kuvassa 3.

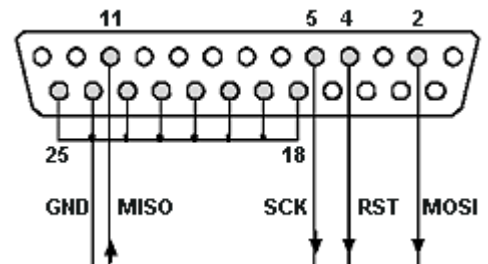


Kuva 3 Mikrokontrollerikortin kytkentä

### 3.3 Kytkennän käytännön testaus

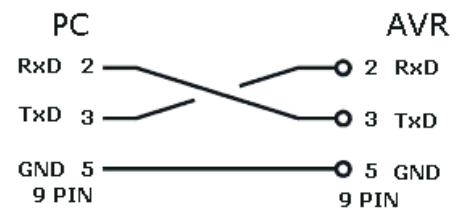
Ennen piirilevyn suunnittelua ja komponenttien lopullista juottamista päätettiin kasata kytkentä koekytkentälevylle ja testata kytkentä käytännössä mahdollisten yllätysten varalta. Kytkentään lisättiin yhteen kontrollerin I/O-nastaan ledi ja etuvastus, jotta voidaan todeta kytkennän toimivuus yksinkertaisella ledivilkuttelijalla.

Mikrokontrollerin ohjelmointia varten valmistettiin Dontronics DT006 -tyyppinen ohjelmointijohto. Kuvassa 4 on esitetty 25-napainen urospuolinen D-liitin PC-tietokoneen rinnakkaisporttiin ja ohjelmointijohdon kytkentänastat mikrokontrolleriin. /7, s.7./



Kuva 4 D-liitin juotospuolelta katsottuna

Sarjaliikennekaapeliksi käy tyypillinen nollamodeemikaapeli, jossa lähetys- ja vastaanotonastat kytketään ristiin kuvan 5 mukaisesti. /7, s.11./



Kuva 5 Sarjaliitännäkytkentä

Kehitysohjelmistoksi PC-tietokoneeseen valittiin CodeVision AVR C-kääntäjän ilmaisversio Windows-ympäristöön. Ohjelmassa on mukana myös piirin ohjelmointimahdollisuus ja tuki DT006-tyyppiselle ohjelmointijohdolle. Lisäksi ohjelmassa on terminaaliohjelma sarjaliikenteen testausta varten.

Mikrokontrollerin kelloaajuus ja käytettävä oskillaattori on valittavissa sisäänrakennetuilla fuse-biteillä. Mikrokontrolleri toimii tehdasasetuksilla piirin sisäisellä 1 MHz:n RC-oskillaattorilla. Fuse-biteillä on säädettävissä myös käyttöjännitteen ilmaisin (BurnDown), EEPROM-muisti, resetoinnin toiminta ja käynnistysohjelman lataustoiminnon käyttöönotto (Bootloader). Ennen flash-muistin ohjelmointia tarkistetaan, että fuse-bitit ovat oikein.



Kun ohjelmoidaan fuse-bitit CKSEL0=0, CKSEL1=1, CKSEL2=0 ja CKSEL3=0 -asentoon, saadaan käyttöön sisäinen oskillaattori. Kun käytössä on sisäinen oskillaattori, voidaan määrittää bitillä CKDIV8, onko kellotaajuus 1 MHz vai 8 MHz. Kellotaajuudeksi valittiin 8 MHz, joten ohjelmoidaan bitti ykköseksi. /5/

Tehdään yksinkertainen ohjelmaesimerkki C-kielessä, joka vilkuttaa porttiin B kytkettyä lediä ja lähettää sarjaporttiin merkin, jotta voidaan todeta kytkennän toimivuus. Kommenteissa on selitetty tarkemmin koodin toiminta.

```
/* Lähettää sarjaporttiin merkin ja vaihtelee portin B lähtöjä
500ms:n välein. */

#include <mega88.h>
#include <delay.h>

void main(void)
{
    DDRB = 0xFF;          // alustetaan portti B lähdöksi
    PORTB = 0x00;        // portin bitit nolliksi

    /*
    USART:n alustus parametreilla 8-data, 1 stop, ei pariteettia,
    vastaanotto ja lähetys päälle, asynkroninen lähetys,
    nopeus 9600 bps
    */

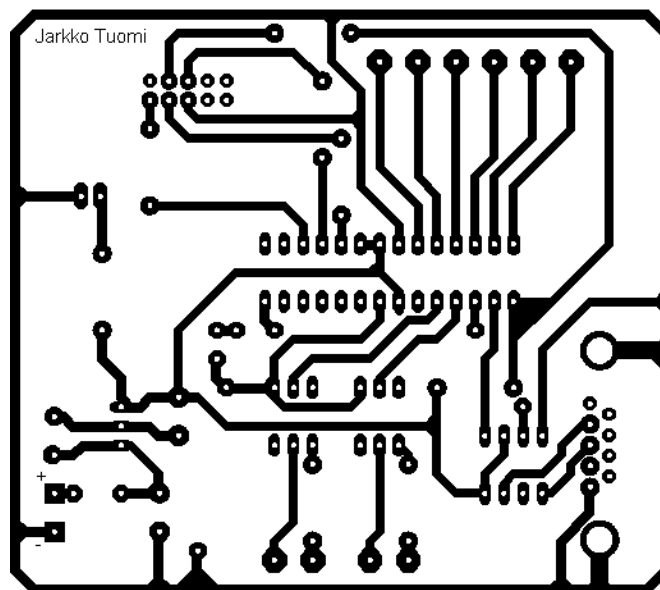
    UCSRA=0x00;
    UCSRB=0xD8;
    UCSRC=0x86;
    UBRR0H=0x00;
    UBRR0L=0x33; /* sarjaportin nopeus lasketaan kaavasta:
                  UBRR0L=(System clock)/16* baud rate)-1 */

    while (1)            // ikuinen silmukka
    {
        UDR0='A';        // lähetetään merkki A sarjarekisteriin
        PORTB = 0xFF;    // portin kaikki bitit ykkösiksi
        delay_ms(500);   // 500ms viive
        PORTB = 0x00;    // portin kaikki bitit takaisin nolliksi
    }
}
```

Ohjelma käännetään CodeVision-kääntäjällä heksadesimaalimuotoon, joka siirretään kontrollerin flash-muistiin. Ohjelma toimi odotetusti, joten seuraavaksi voidaan suunnitella kytkennälle piirilevy.

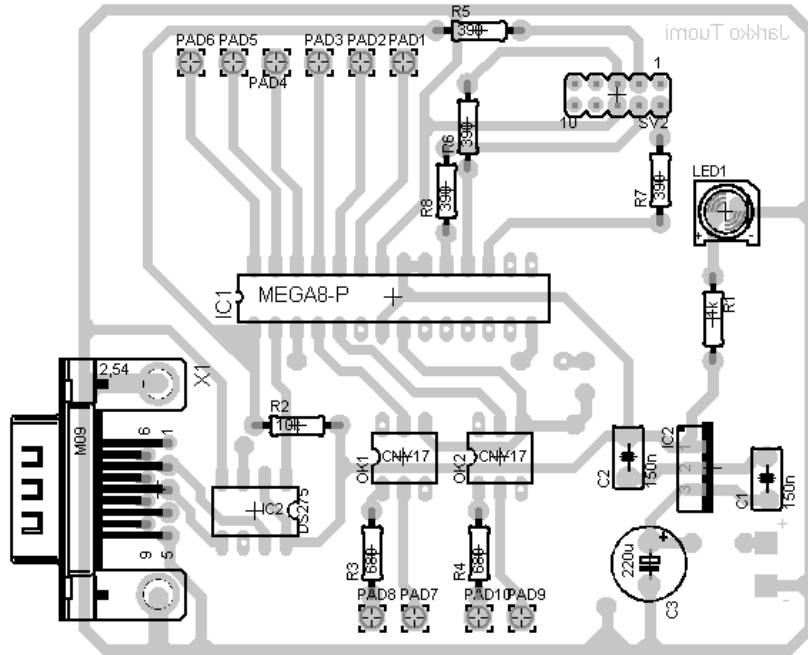
### 3.4 Piirilevyn suunnittelu

Valmiiksi piirretty kytkentä siirrettiin Eagle-ohjelman piirilevynsuunnitteluosioon. Ohjelmassa on autoroute-toiminto, joka koittaa hakea parhaat mahdolliset vedot levyille. Tätä ei kuitenkaan käytetty, koska vedoista tuli pääosin mielettömiä ja jotkin vedot jäivät kokonaan vetämättä. Näin johdotukset vedettiin käsin, jolloin päästiin paljon järkevämpään tulokseen. Vedoista pyrittiin tekemään mahdollisimman paksut ja padit (juotostäplät) pitämään mahdollisimman suurina, jotta käsinjuotto olisi helpompaa. Komponentit pyrittiin asettelemaan niin, että niillä on riittävä välimatka toisiinsa. Johdotuksissa pyrittiin välttämään 90 asteen kulmia. Piirilevyä suunniteltaessa lisättiin kytkentään vielä paikka jännitesyöttöön suojadiodille estämään vääränapainen kytkentä. Lisäksi levyille lisättiin portin B lähtöön mahdollisuus kytkeä ledi ja etuvastus. Kuvassa 6 on esitetty lopullinen piirilevyn foliokuva.



Kuva 6 Piirilevyn foliokuva

Kytkenälle tulostettiin vielä osien sijoittelukuva komponenttien sijoittelun helpottamiseksi. Kuvassa 7 on osien sijoittelu komponenttipuolelta katsottuna.



Kuva 7 Osien sijoittelukuva

## 4 OHJAUSKORTTI

Askelmoottorin ja releiden ohjaus päätettiin suunnitella ja toteuttaa juottamalla kytkentä reikälevylle. Kortille tuodaan erillinen reguloitu 12 V jännite. ATmega88-mikrokontrolleri kykenee syöttämään noin 20 mA virran jokaisesta I/O-portistaan /5/. Mikäli portista yritetään ottaa suurempia virtoja, on virrankestoisuutta kasvatettava puskuripiirillä tai yksittäisellä transistorilla.

## 4.1 Komponenttivalinnat

Askelmoottorin ohjausta varten valittiin 8-porttinen puskuripiiri ULN2803A, joka pystyy käsittelemään jokaisesta portistaan 500 mA virran. Piirin tuloja ohjataan 5 V jännitteellä, joten se sopii hyvin suoraan liitettäväksi mikrokontrollerin portteihin. Piiri pyrkii vetämään maihin aina sen portin lähdön, jonka tuloon on kytketty 5 V eli looginen ykkönen. Kun askelmoottorin käämin toinen pää on kytketty käyttöjännitteeseen ja toinen pää piirin lähtöporttiin, lähtee käämissä virta kulkemaan. /10/

Releen ohjausta varten valittiin n-tyyppinen Mosfet RFD14N05L, jota ohjataan logiikkatasoilla. Fetin virrankesto on 14 A ja jännitteenkesto 50 V. Fetin hilalle tuotu looginen ykkönen (5 V) ohjaa fetin johtavaan tilaan ja looginen nolla tekee päinvastoin. /11/

Releellä on tarkoitus ohjata kodin erilaisia sähkölaitteita, joten sen pitää soveltua sähköverkkoon 230 VAC jännitteeseen. Releeksi valittiin Texcellin valmistama piirilevylle juotettava malli, jota voidaan ohjata 12 V jännitteellä.

Tehonkestoisuudeksi ilmoitetaan vaihtovirtaan kytkettynä 2500 VA näennäisteho (250 VAC, 10 A) ja tasavirtaan 300 W pätöteho (30 V, 10 A). Lasketaan vielä releen käämin kuluttama virta käyttäen Ohmin lakia. Käämin resistanssiksi yleismittarilla saatiin 270 Ω, ja kun käämiin syötetään 12 V tasajännite, saadaan virta releen vetäessä laskettua kaavasta:

$$I = \frac{U}{R} = \frac{12 \text{ V}}{270 \Omega} = 0,0444 \text{ A} = 44 \text{ mA} \quad (2)$$

Askelmoottori (kuva 8) irrotettiin vanhasta nauha-  
asemasta. Moottoriin tuli kuusi eri väristä johdinta ja  
päällä luki MCM-1 4608 Epson. Näillä tiedoilla ei  
löydetty moottorista datatietoja, joten piti itse selvittää  
johtimien ja käämien järjestys. Seuraavassa  
kappaleessa on esitetty moottorin johtimien selvitys.

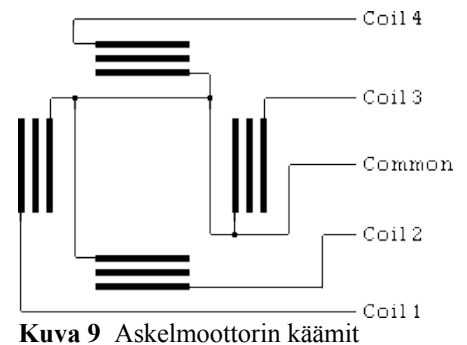


Kuva 8 Askelmoottori

## 4.2 Askelmoottorin johtimien selvittäminen /12/

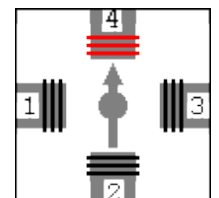
Koska moottoriin tulee kuusi johdinta, joista kaksi on samanväristä, voidaan päätellä, että moottori on nelivaiheinen unipolaarinen askelmoottori, jolla on kaksi käämien keskiulosottojohdinta. Keskiulosottojohtimet kytetään molemmat käyttöjännitteeseen. Kuvassa 9 on esitetty nelivaiheisen askelmoottorin käämien kytkentä. Varmistetaan vielä ohmi-mittarilla moottorin Common eli keskiulosottojohdin. Keskiulosoton ja käämin välisen resistanssin tulee olla kaksi kertaa pienempi kuin käämien välinen resistanssi. Saadaan mitattua kaksi eri resistanssia, 117 Ω ja 234 Ω, joten samanväriset ruskeat kaksi johdinta ovat keskiulosotot. Nyt voidaan laskea Ohmin lailla käämin kuluttama virta, kun se kytetään 12 V jännitteeseen:

$$I = \frac{12 \text{ V}}{117 \Omega} = 0,1026 \text{ A} \approx 103 \text{ mA} \quad (3)$$



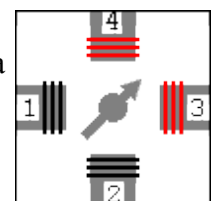
Kuva 9 Askelmoottorin käämit

Seuraavaksi selvitetään neljän muun johtimen käämijärjestys. Kun kytetään keskiulosottojohtimet 12 V jännitteeseen ja valitaan yksi käämijohtimista kytketyksi maihin kuvan 10 mukaisesti. Tätä käämijohdinta voidaan merkitä numerolla neljä.



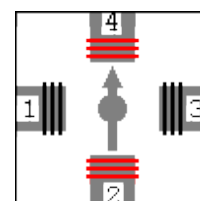
Kuva 10 Käämi 4 kytkettynä

Seuraavaksi kytetään jokin muukin kolmesta johtimesta maihin, ja kun moottorin akseli kääntyy kuvan 11 mukaisesti oikeaan yläviistoon, merkitään tätä johdinta numerolla kolme.

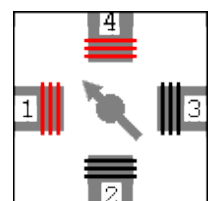


Kuva 11 Käämit 4 ja 3 kytkettynä

Se johdin, joka ei maahan kytkettynä pyöritä akselia, on kytketty käämiin numero 2 kuvan 12 mukaisesti. Viimeinen jäljelle jäänyt johdin merkitään käämiksi 1 (kuva 13).



Kuva 12 Käämit 4 ja 2 kytkettynä



Kuva 13 Käämit 4 ja 1 kytkettynä

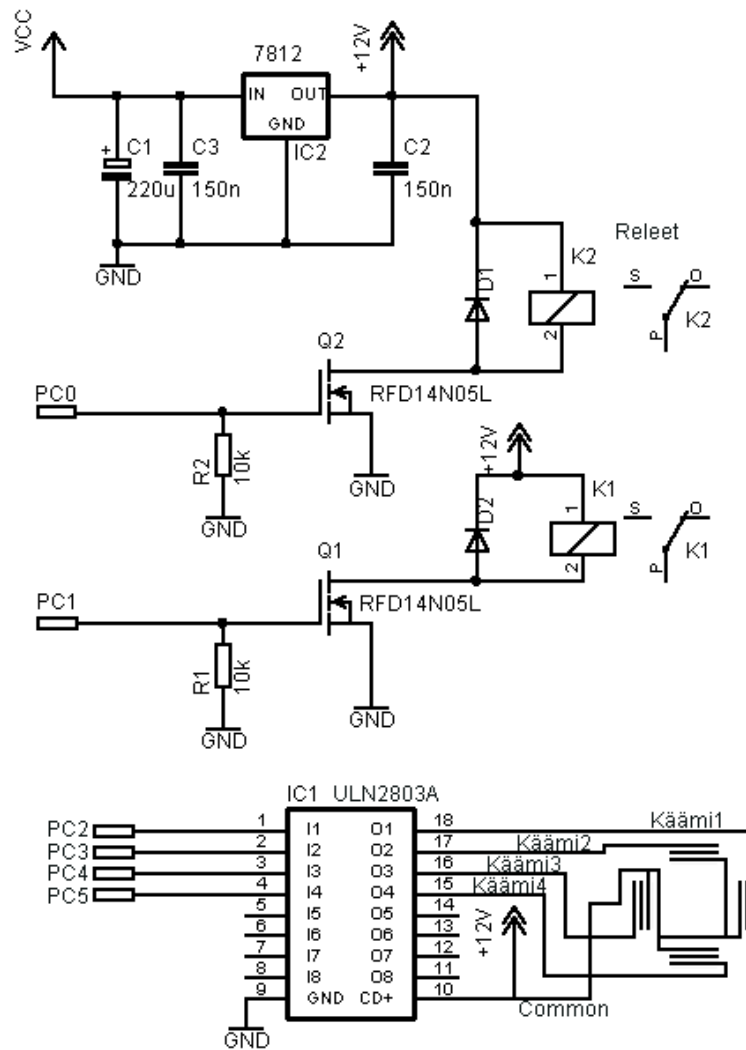
**Taulukko 1** Askelmoottorin johtimet

| JOHTIMEN VÄRI | SELITYS       |
|---------------|---------------|
| ruskea        | keskiulosotto |
| sininen       | käämi 1       |
| punainen      | käämi 2       |
| oranssi       | käämi 3       |
| valkoinen     | käämi 4       |

Lopulliset askelmoottorin johtimet ovat esitetty taulukossa 1. Kun johtimet vedetään käämien numerojärjestyksessä maihin, lähtee askelmoottori askeltamaan kytkentäjärjestyksen määräämään suuntaan.

### 4.3 Kytkennän piirtäminen

Kortin viemä virta edellä lasketun mukaisesti on molempien releiden vetäessä ja moottorin käydessä minimissään noin 200 mA. Virtalähteeksi kortille kannattaa kuitenkin valita vähintään 300 mA syöttävä lähde, jotta taataan riittävä virran saanti. Kytkentään sijoitettiin 12 V:n reguloitu syöttö 7812 regulaattorilla suodatuskondensaattoreineen. Releen rinnalle käyttöjännitteestä katsottuna estosuuntaan asetettiin suojadiodi suojaamaan fettiä, näin releen avausvaiheessa käämin virtapiikki purkautuu diodin kautta. Fettien hilasta maihin kytkettiin 10 k $\Omega$  vastukset rajoittamaan ottovirtaa kontrollerin portista ja purkamaan releen käämin virta katkaisun jälkeen /13, s.207/. Kuvassa 14 on esitetty lopullinen ohjauskortin kytkentä.



Kuva 14 Ohjauskortin kytkentä

## 5 OHJELMISTON KEHITYS

Laitteen ohjelmisto koostuu kolmesta eri ohjelmakoodista. Mikrokontrollerikortille on kirjoitettu laitteistonläheinen C-kielinen koodi, joka hoitaa sarjaliikenteen, moottorin ja releiden ohjauksen sekä ulkoisten tilojen ilmaisun. PC-koneella on sarjaporttia haasteleva shell-skripti, joka kirjoittaa kortilta tulleet tiedot talteen. Lisäksi PC-koneella toimii PHP- ja HTML-kieltä käyttäen kirjoitettu hallintaohjelma, jolla annetaan sarjaportin kautta ohjauskomennot mikrokontrollerikortille.

## 5.1 Mikrokontrollerikortin ohjelma /5/

Mikrokontrollerikortin ohjelma toteutettiin täysin keskeytyspohjaisesti. Keskeytyksiä käytettäessä kontrollerin ei tarvitse käyttää resursseja jatkuvaan tilojen kyselyyn vaan tapahtumaa vastaava palvelu suoritetaan välittömästi, kun sille esiintyy tarvetta. ATmega88-mikrokontrollerissa on yhteensä 26 erilaista yksikköä, joiden perustella voidaan toteuttaa keskeytys. Lisäksi kontrollerin mistä tahansa I/O-portin tilanmuutoksesta voidaan aiheuttaa ulkoinen keskeytys. Porttien ulkoiset keskeytyksille on varattu kolme eri keskeytysvektoria ja portit on jaettu niin, että porteissa 0-7 tapahtuvat tilanmuutokset suoritetaan keskeytysvektorissa PCINT0, 8-14 vektorissa PCINT1 ja 15-23 vektorissa PCINT2. Ohjelma on toteutettu niin, että kun portin D nastoissa 2 (PCINT18) tai 3 (PCINT19) muuttuu tila, tapahtuu ulkoinen keskeytys. Näihin portteihin voidaan keskeytys tarvittaessa kytkeä myös reagoimaan pelkästään tilaan 1 tai 0. Jotta ulkoiset keskeytykset saadaan käyttöön, tulee alustaa rekisterit PCICR, PCMSK2 ja PCIFR. Tarkastellaan näitä rekistereitä tarkemmin. Taulukossa 2 on PCICR-rekisterin bittien merkitykset lueteltuna, kun bitti 2 (PCIE2) asetetaan, kytkeytyy käyttöön ulkoisten porttien 16-23 keskeytykset, joten rekisteriin kirjoitetaan bitit 0000 0100, joka on heksadesimaalilukuna 04 h.

**Taulukko 2** PCICR-rekisteri

| bitti    | 7 | 6 | 5 | 4 | 3 | 2     | 1     | 0     |
|----------|---|---|---|---|---|-------|-------|-------|
| merkitys | - | - | - | - | - | PCIE2 | PCIE1 | PCIE0 |

Taulukossa 3 on PCIFR-rekisteri, jolla PCICR-rekisterin tapaan sallitaan keskeytyspyynnöt vielä erikseen. Samaan tapaan rekisteriin kirjoitetaan luku 04 h.

**Taulukko 3** PCIFR-rekisteri

| bitti    | 7 | 6 | 5 | 4 | 3 | 2     | 1     | 0     |
|----------|---|---|---|---|---|-------|-------|-------|
| merkitys | - | - | - | - | - | PCIF2 | PCIF1 | PCIF0 |

Rekisterillä PCMSK2 (taulukko 4) otetaan vielä käyttöön ne portit, joista keskeytykset sallitaan. Opto-erotetut sisäänmenot on kytketty kontrollerin nastoihin PCINT18 ja PCINT19, joten ne bitit asetetaan.



**Taulukko 4** PCMSK2-rekisteri

| bitti    | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| merkitys | PCINT23 | PCINT22 | PCINT21 | PCINT20 | PCINT19 | PCINT18 | PCINT17 | PCINT16 |

Lisäksi pitää alustaa sarjaportti ja ottaa käyttöön sarjaliikennekeskeytys. Tämä tapahtuu rekistereillä UCSR0B ja UCSR0C. Laitteiden väliset ohjaukset hoidetaan lähettämällä sarjaportin kautta yksi merkki (tavu). Jokaiselle merkille on standardoidussa ASCII-kooditaulukossa koodattu tietty bittimuodostelma, minkä perusteella vastaanottajapuoli osaa kääntää bitit takaisin merkiksi. Tiedonsiirtotavaksi valitaan 8 databittiä, yksi stop-bitti ja pariteettitarkistus pois päältä. Tiedonsiirtonopeudeksi valitaan 9600 bps, tästä voidaan laskea, kuinka monta tavua lähetetään sekunnissa. Koska yhden tavun lähettäminen vaatii todellisuudessa 10 bittiä, aloitusbitti, 8 databittiä ja yksi lopetusbitti, siirtonopeudeksi saadaan:

$$\frac{9600 \text{ bps}}{10} = 960 \text{ t/s} \quad (4)$$

Rekistereihin UBRR0L ja UBRR0H kirjoitetaan haluttu tiedonsiirtonopeus. Rekisterin arvo määritellään kaavalla:

$$UBRR = \frac{f_{ck}}{16 \cdot BAUD} - 1 \quad (5)$$

jossa  $f_{ck}$  = kontrollerin kellotaajuus ja  $BAUD$  = haluttu tiedonsiirtonopeus. Kun kontrollerin kellotaajuus on 8 MHz ja haluttu tiedonsiirtonopeus 9600 bps, saadaan laskettua kaavasta desimaaliluku 51 ja se muunnettuna heksadesimaaliluvuksi on 33 h. Tämä kirjoitetaan rekisteriin UBRR0L ja ylempään rekisteriin UBRR0H kirjoitetaan nolla. Taulukoissa 5 ja 6 on esitetty sarjaliikennerekistereiden merkitykset tarkemmin.

**Taulukko 5** UCSR0B-rekisteri

| bitti    | 7      | 6      | 5      | 4     | 3     | 2      | 1     | 0     |
|----------|--------|--------|--------|-------|-------|--------|-------|-------|
| merkitys | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 |

RXCIE0 = Merkin vastaanoton keskeytyksen sallinta  
TXCIE0 = Merkin lähetyksen keskeytyksen sallinta  
UDRIE0 = Lähetyspuskuri tyhjä-keskeytyksen sallinta  
RXEN0 = Merkkien vastaanoton sallinta  
TXEN0 = Merkkien lähetyksen sallinta  
UCSZ02 = Merkin pituus, jatkuu rekisterissä UCR0C, eniten merkitsevä bitti  
RXB80 = 9-bittisissä merkeissä vastaanotettu 9-bitti  
TXB80 = 9-bittisissä merkeissä lähetettävä 9-bitti

**Taulukko 6** UCR0C-rekisteri

| bitti    | 7       | 6       | 5     | 4     | 3     | 2      | 1      | 0      |
|----------|---------|---------|-------|-------|-------|--------|--------|--------|
| merkitys | UMSEL01 | UMSEL00 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 | UCPOLO |

UMSEL01 = Lähetystavan valinta (synkroninen tai asynkroninen), ylempi bitti  
UMSEL00 = Lähetystavan valinta (synkroninen tai asynkroninen), alempi bitti  
UPM01 = Pariteetin valinta, ylempi bitti  
UPM00 = Pariteetin valinta, alempi bitti  
USBS0 = Stop-bittien lukumäärä, 0 = 1 bitti ja 1 = 2 bittiä  
UCSZ01 = Merkin pituus, keskimmäinen bitti  
UCSZ00 = Merkin pituus, vähiten merkitsevä bitti  
UCPOLO = Synkronisessa siirrossa käytössä, tahdistus nousevaan tai laskevaan reunaan

Rekisteriin UCSR0B kirjoitetaan luku 1101 1000 (D8 h) eli otetaan merkkien lähetyksen ja vastaanoton sekä keskeytykset käyttöön. Rekisteriin UCR0C kirjoitetaan 0000 0110 (06 h).

Kontrollerin I/O-portit alustetaan rekistereiden DDRB, DDRC ja DDRD avulla niin, että 1 asettaa portin bitin lähdeksi ja 0 tuloksi. Sarjaliikennekeskeytyksen palveluohjelma toimii niin, että ensin luetaan rekisteristä UDR0 saapunut merkki. Switch-case-rakenteella seulotaan lähetetyn merkin perusteella, kytetäänkö releet vai pyöritetäänkö askelmootoria. Ulkoisten keskeytysten palveluohjelma toimii niin, että sarjaporttiin lähetetään tuloporttien tilat ”,”-merkillä erotettuna, lopuksi lähetetään vielä rivinvaihtomerkki, joka toimii merkinä lähetyksen päättymisestä. Merkin lähetykselle on tehty oma aliohjelmansa, joka odottaa sarjaportin tyhjenemistä ennen kuin lähettää uuden merkin linjalle. Rekisteri UCR0A on sarjaportin tilarekisteri, jossa viides bitti UDRE0 asettuu, kun lähetyspuskuri on tyhjä. Lopullinen ohjelma kommentteineen on esitetty liitteessä 1.

## 5.2 Sarjaportin tarkkailu-skripti

PC-koneelle kirjoitettiin shell-skripti tarkkailemaan sarjaporttia. PC-koneen sarjaportti pitää alustaa samoilla arvoilla kuin mikrokontrollerikortin sarjaportti.

Linuxissa sarjaportti alustetaan käyttämällä stty-skriptiä:

```
stty -F /dev/ttyS0 9600 cs8 -parenb -cstopb -icanon
```

Skriptille annetaan parametreina sarjaportin laitetiedosto, nopeus (9600 bps), databitit (8), pariteettitarkistus (pois), stop-bitit (1), ei-kanoninen tulostus eli jokainen merkki tulostetaan. Koneelle on asennettu Bash komentotulkki versio 3.0, joka on oletuksena valtaosassa Linux-jakeluversioita. Skripti toimii niin, että se lukee sarjaportilta merkkejä niin kauan kuin tulee rivinvaihtomerkki, jonka jälkeen se käy kirjoittamassa tulleet merkit tiedostoon. Tiedostoon kirjoitetaan vielä date-komentoa käyttäen aikaleima tiedostoon, jotta tiedetään koska tilanmuutoksia on tapahtunut. Tiedostoa ei välillä tyhjennetä, vaan kaikki tiedot kirjoitetaan aina tiedoston loppuun. Samalla tiedosto toimii tapahtumien loki-tiedostona /14/.

Seuraavassa on shell-skripti kommentteineen:

```
#!/bin/bash
while true          # Ikuinen silmukka
do

# Bash-shellin sisäinen komento read lukee sarjaportista merkkejä
# muuttujaan RIVI, kunnes tulee rivinvaihtomerkki.

    read RIVI < /dev/ttyS0

# Tulostetaan muuttujan RIVI sisältö ilman rivinvaihtomerkkiä
# tiedostoon ja erotinmerkki ", " perään.

    echo -n $RIVI", " >> /home/www/tiedosto

# Date-komennolla tulostetaan aikaleima vielä tiedoston samalle
# riville. Date-komento tulostaa loppuun rivinvaihtomerkin.

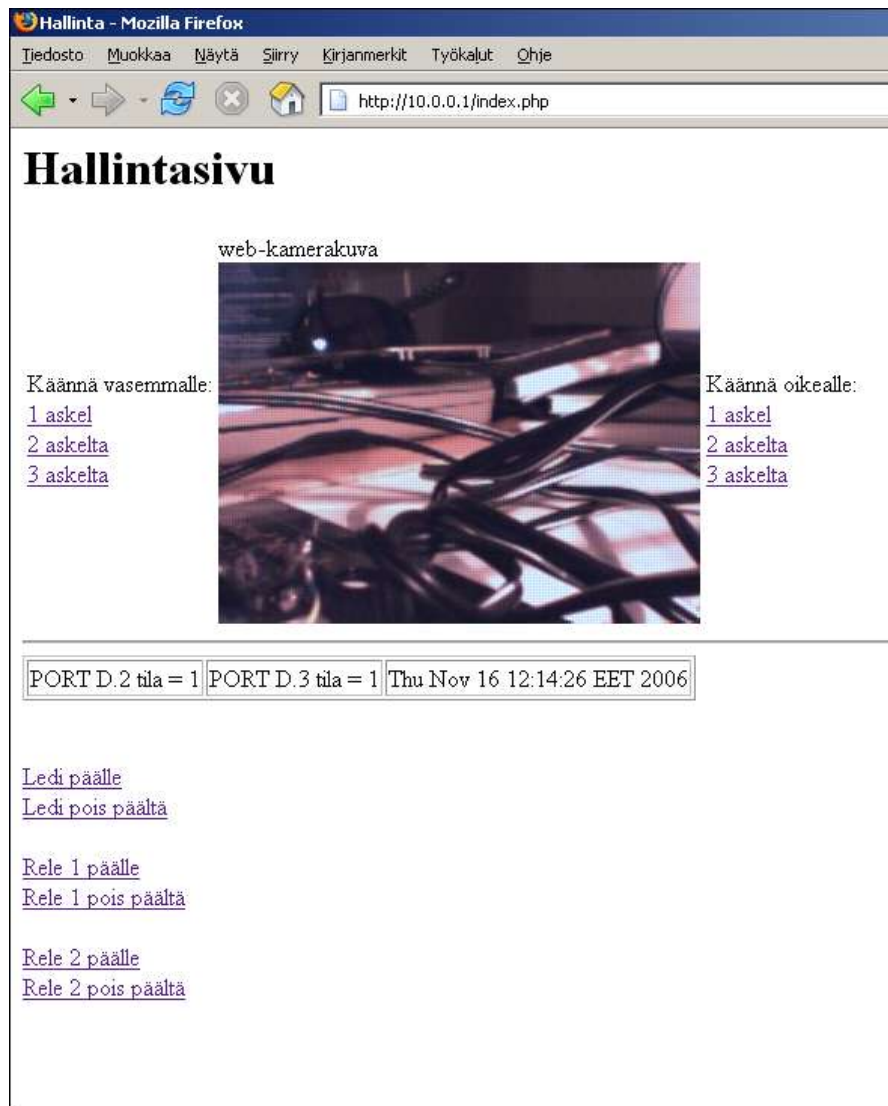
    date >> /home/www/tiedosto

done
```

Skripti käynnistetään Linux-käyttäjärjestelmässä tausta-ajoon prosessiksi kirjoittamalla skriptin perään ”&”-merkki.

### 5.3 Käyttöliittymä

Käyttöliittymä kirjoitettiin WEB-sivuksi HTML- ja PHP-kielillä. Käyttöliittymä on esitetty kuvassa 15. Askelmoottoria voi liikuttaa molempiin suuntiin kolme eri pituista askelta. Käytössä olevan askelmoottorin yhden askeleen kääntymä on  $360/24 = 15$  astetta. Sivulla näkyy kortin sisäänmenojen viimeisimmät tilat ja päiväys. Lisäksi hallintasivulta on mahdollisuus sytyttää ja sammuttaa kortilla oleva testiledi. PHP-ohjelma toimii niin, että se ottaa HTML-sivulta syötteenä halutun komennon ja lähettää sen PC-koneen sarjaporttiin kirjoittamalla laitetiedoston. Laitetiedoston oikeuksia pitää muuttaa niin, että PHP-ohjelma saa kirjoittaa siihen. Ohjelma lukee valmiiksi muotoiltua loki-tiedostoa, johon on kirjoitettu kortilta tulleet tiedot. Lopullinen HTML-koodi PHP-koodeineen on esitetty liitteessä 2 kommentteineen.



Kuva 15 Käyttöliittymä

## 6 YHTEENVETO

Työssä saavutettiin sille asetetut tavoitteet ja se toimi odotetulla tavalla. Työ osoittautui varsin haasteelliseksi, koska aikaisempaa kokemusta Atmelin mikrokontrollereista ei ollut. Lisäksi työtä tehdessä opeteltiin Eagle-ohjelmiston käyttö perin pohjaisesti. Mikrokontrollerikortin suunnittelun ja ohjelmiston kehityksen apuna käytettiin pääosin ATmega88-mikrokontrollerin datalehteä, joka osoittautui erittäin kattavaksi ja yksityiskohtaiseksi. Laitteen kokoaminen tuli erittäin edulliseksi, sillä kokonaiskustannukset olivat kaikkine komponentteineen muutaman kymmenen euroa. Liitteessä 3 on vielä esitetty kuva laitteen lopullisesti testauskytkennästä.

Askelmoottorin suuri kiihtyvyys ja askeltaminen vaatii moottorilta enemmän vääntömomenttia, tästä syystä moottoriksi pitäisi valita tehokkaampi askelmoottori tai vaihtoehtoisesti käyttää servomoottoria tai tavallista DC-moottoria, jonka kiihtyvyyttä ja pyörimisnopeutta pystyy helposti muuntelemaan mikrokontrollerin PWM-lähdöllä. Lisäksi ohjelmistoon voisi vielä lisätä ympäri pyörimisen eston, jotta kameran johto ei mene solmuun. Jatkossa kameralle voisi myös lisätä pystysuuntaisen kääntymisen mahdollisuuden toisen moottorin avulla.

Seuraava mahdollinen kehitysvaihe projektissa on suunnitella laite täysin itsenäiseksi valvontalaitteeksi niin, että laite käsittelee itse verkko- ja palvelintoiminnot. Tämä on täysin mahdollista toteuttaa ATmega-mikrokontrolleriperheellä. Kytkentään liitetään sopiva ethernet-piiri ja kontrollerille kirjoitetaan reaaliaikakäyttäjärjestelmä. Kontrollerille on olemassa myös valmiita harrastelijoiden valmistamia käyttäjärjestelmiä, jotka käyttävät tcp/ip-pinoa ja näin kontrolleri saadaan toimimaan WWW-palvelijana. Näin ei tarvita ulkoista PC-palvelinkonetta eikä RS232-muunninta, koska kaikki liikenne kulkee suoraan ethernetin kautta kortille. ATmega-kontrolleriin on ulkoisella piirillä toteutettavissa USB-väylä, joten sen kautta voidaan muodostaa langaton yhteys tukiasemaan käyttäen WLAN-tekniikkaa kytkeällä kontrollerin USB-porttiin WLAN-vastaanotin. Web-kameran liittämien tämänkaltaiseen laitteeseen on haastavaa, koska ajurit ohjelmistot pitää kirjoittaa täysin itse ja valmistajia on vaikea saada luovuttamaan kameroiden teknisiä dokumentteja ilman kaupallisia sopimuksia.

## LÄHTEET

- 1 Koski, Raimo – Tervo, Esa. Linux tehokäyttäjän opas, 2. painos. Suomen ATK-kustannus, Jyväskylä 1998. 552 s. Luku 2.
- 2 Virtanen, Mika. Fotogrammetrian erikoistyö. [Sähköinen dokumentti]. [Viitattu 21.11.2006] <http://www.foto.hut.fi/opetus/290/julkaisut/mvi-etyo.pdf> 47 s.
- 3 Michel Xhaard. [www-sivu]. [Viitattu 21.11.2006] <http://mxhaard.free.fr>
- 4 Wikipedia. [www-sivu]. [Viitattu 21.11.2006] [http://gentoo-wiki.com/HOWTO\\_logitech\\_quickcam\\_on\\_2.6.x\\_kernel](http://gentoo-wiki.com/HOWTO_logitech_quickcam_on_2.6.x_kernel)
- 5 Atmel, ATmega88 datakirja. [Sähköinen dokumentti]. [Viitattu 21.11.2006] [http://www.atmel.com/dyn/resources/prod\\_documents/doc2545.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf) 372 s.
- 6 Taiwan Semiconductor. 7805-regulaattorin datakirja. [Sähköinen dokumentti]. [Viitattu 21.11.2006] [http://www.taiwansemi.com/TSCbig5/ezcatfiles/cust/img/img/ts7800\\_0312f.pdf](http://www.taiwansemi.com/TSCbig5/ezcatfiles/cust/img/img/ts7800_0312f.pdf) 9 s.
- 7 Tietomyrsky Oy. PB8515-kortin ohje. [Sähköinen dokumentti] [Viitattu 21.11.2006] <http://www.tietomyrsky.fi/tiedostot/pb8515/pb8515.pdf> 25 s.
- 8 Dallas Semiconductor. DS275-piirin datakirja. [Sähköinen dokumentti]. [Viitattu 21.11.2006] <http://pdfserv.maxim-ic.com/en/ds/DS275.pdf> 8 s.
- 9 Fairchild Semiconductor. CNY17-3Z-opto-erottimen datakirja. [Sähköinen dokumentti]. [Viitattu 21.11.2006] <http://optimus.meleeisland.net/downloads/datasheets/cny17-4.pdf> 11 s.
- 10 Texas Instruments. ULN2803A-puskuripiirin datakirja. [Sähköinen dokumentti]. [Viitattu 21.11.2006] <http://focus.ti.com/lit/ds/symlink/uln2803a.pdf> 11 s.

- 11 Fairchild Semiconductor. RFD14N05L-fetin datakirja.  
[Sähköinen dokumentti]. [Viitattu 21.11.2006]  
<http://www.fairchildsemi.com/ds/RF/RFD14N05LSM.pdf> 8 s.
- 12 Harries, Ian. Stepper Motors. [www-sivu]. [Viitattu 21.11.2006]  
<http://www.doc.ic.ac.uk/~ih/doc/stepper/others/>
- 13 Koskinen, Jari. Mikrotietokonetekniikka sulautetut järjestelmät, 2. painos.  
Otavan kirjapaino Oy, Keuruu 2006. 294 s.
- 14 Kuivanen, Ilpo. Johdatus komentoriviohjelmointiin.[www-sivu].  
[Viitattu 21.11.2006] <http://cs.stadia.fi/~kuivanen/linux/kom.php>

**/\* ATmega88-mikrokontrollerin ohjelmakoodi \*/**

```
#include <mega88.h>
#include <delay.h>

char merkki;           // Alustetaan ohjelmassa tarvittavat muuttujat char-tyyppiseksi
char askel;
char suunta;
char q;

void putchar(char lahmerk) { // Aliohjelma, joka lähettää merkin sarjaportille
    while(!(UCSR0A & 0x20)); // Odotetaan kunnes lähetyspuskuri tyhjä. Silmukka lopetetaan,
                            // kun bittioperaatio and luvun 0010 0000 (20h) ja UCR0A-
                            // rekisterin kanssa on tosi

    UDR0 = lahmerk;       // Asetetaan merkki sarjaliikennerekisteriin
}

interrupt [PCINT2] void pin_change_isr2(void) // Ulkoisen keskeytyksen palveluohjelma
{
    char tila;
    if(PIND.2)           // Tarkastetaan portin D nastan 2 tila ja asetetaan se muuttujaan tila
        tila='1';
    else
        tila='0';

    putchar(tila);      // Lähetetään nastan tila sarjaporttiin

    if(PIND.3)
        tila='1';
    else
        tila='0';

    putchar(',');       // Lähetetään sarjaporttiin merkki "," porttien tilojen erotusmerkkinä
    putchar(tila);     // Lähetetään sarjaporttiin portin D nastan 3 tila
    putchar(0x0A);     // Lähetetään sarjaporttiin rivinvaihtomerkki, jotta
                    // vastaanottopää tietää lähetyksen päättyneen
}

interrupt [USART_RXC] void usart_rxd_isr(void) // Sarjaliikenteen keskeytyksen palveluohjelma,
                                                // kun dataa vastaanotetaan
{
    suunta=0;
    merkki=UDR0;       // Luetaan sarjarekisteristä tieto muuttujaan merkki
}
```



```
switch(merkki) // Switch-case-rakenteella valitaan ohjaus
{
case 'z':      // ----- Askelmoottorin ohjaus -----
    suunta=1; // Parametreina suunta ja askeltamisen pituus
    askel=1;
    break;
case 'x':
    suunta=1;
    askel=2;
    break;
case 'c':
    suunta=1;
    askel=3;
    break;
case 'a':
    suunta=2;
    askel=1;
    break;
case 's':
    suunta=2;
    askel=2;
    break;
case 'd':
    suunta=2;
    askel=3;
    break; // -----

case 'R':     // ----- Releiden ohjaus -----
    PORTC.4 = 1;
    break;
case 'r':
    PORTC.4 = 0;
    break;

case 'E':
    PORTC.5 = 1;
    break;
case 'e':
    PORTC.5 = 0;
    break; // -----

case 'L':     // ----- Ledin ohjaus -----
    PORTB.1 = 1;
    break;
case 'l':
    PORTB.1 = 0;
    break; // -----

}

if(suunta==1)
{
    for(q=0;q<askel;q++) // Montako kertaa toistetaan askellus
    {
        PORTC = (PORTC & 0xF0) | 0x01; delay_ms(50); // Kirjoitetaan portin C neljään alimpaan bittiin
        // askelmoottorin ohjausbitit
        PORTC = (PORTC & 0xF0) | 0x02; delay_ms(50); // Ensin nollataan bittioperaatiolla and PORTC-
        // rekisterin neljä alinta bittiä
        PORTC = (PORTC & 0xF0) | 0x04; delay_ms(50); // tämän jälkeen asetetaan ne bittioperaatiolla or
        PORTC = (PORTC & 0xF0) | 0x08; delay_ms(50); // Jokaisen askelluksen välillä pidetään
        // 50 ms:n tauko
    }
}
```

```
if(suunta==2)                // Askellaan toiseen suuntaan
{
    for(q=0;q<askel;q++)
    {
        PORTC = (PORTC & 0xF0) | 0x08; delay_ms(50);
        PORTC = (PORTC & 0xF0) | 0x04; delay_ms(50);
        PORTC = (PORTC & 0xF0) | 0x02; delay_ms(50);
        PORTC = (PORTC & 0xF0) | 0x01; delay_ms(50);
    }
}

PORTC &= 0xF0;                // Käämit lopuksi jännitteettömiksi
}

void main(void)
{
    DDRB = 0xFF;                // Alustetaan portti B lähdöksi
    DDRC = 0xFF;                // Alustetaan portti C lähdöksi
    DDRD = 0x00;                // Alustetaan portti D tuloksi

    PORTB = 0x00;                // Asetetaan portin B bitit nolliksi
    PORTC = 0x00;                // Asetetaan portin C bitit nolliksi
    PORTD = 0xFF;                // Asetetaan portin D bitit ykkösiksi

    #asm("sei")                // Globaalien keskeytysten sallinta käyttämällä assembleria

    /* Sarjaportin alustus
    Asynkroninen, 8 databittiä, 1 stop-bitti, ei pariteettitarkistusta
    Nopeus 9600 bps
    Lähetys ja vastaanotto päälle
    Lähetys- ja vastaanottokeskeytysten sallinta
    */
    UCSR0A=0x00;
    UCSR0B=0xD8;
    UCSR0C=0x06;
    UBRR0H=0x00;
    UBRR0L=0x33;

    /* Ulkoisten keskeytysten alustus
    INT0: pois
    INT1: pois
    Keskeytys nastoissa 0-7: pois
    Keskeytys nastoissa 8-14: pois
    Keskeytys nastoissa 16-23: päällä
    */
    PCICR=0x04;
    PCMSK2=0x0C;
    PCIFR=0x04;

    while (1) // Ikuinen silmukka
    {
    }
}
```

## // Käyttöliittymäkoodi

```
<?php
$syotto= $_GET['syotto']; // Luetaan html-sivulta tullut syotto-parametrin tieto muuttujaan syotto

if (isset($syotto)) // Jos syotto on määritelty
{
    $kahva = fopen("/dev/ttyS0", "w"); // Avataan sarjaportin laitetiedosto kirjoitettavaksi
    fwrite($kahva, "$syotto"); // Kirjoitetaan laitetiedostoon syöte
    fclose($kahva); // Suljetaan tiedostokahva
}

$kahva = fopen("tiedosto", "r"); // Avataan tiedosto luettavaksi
while(!feof($kahva)) // Luetaan rivi kerrallaan kunnes törmätään tiedoston
// lopetusmerkkiin
{
    $rivi = fgets($kahva, filesize ("tiedosto"));
    $rivi=trim($rivi); // Poistetaan riviltä rivinvaihtomerkki
    if (!$rivi) // Jos rivi on tyhjä, lopetetaan while-silmukka
        break;
    $taulukko = split ("", $rivi); // Paloitellaan rivin tiedon taulukkoon käyttämällä ","
    // erotusmerkinä
}
fclose($kahva);

?>

<html>
<head>
    <META HTTP-EQUIV="Pragma" CONTENT="no-cache">

    <title>Hallinta</title>
</head>

<body>

<h1>Hallintasivu</h1>

<table><tr>

<td>
Käännä vasemmalle:<br>
<a href="index.php?syotto=a">1 askel</a><br>
<a href="index.php?syotto=s">2 askelta</a> <br>
<a href="index.php?syotto=d">3 askelta</a>
</td>

<td>
web-kamerakuva<br>
</td><td>Käännä oikealle:<br>
<a href="index.php?syotto=z">1 askel</a><br>
<a href="index.php?syotto=x">2 askelta</a><br>
<a href="index.php?syotto=c">3 askelta</a><br>
</td>
</tr></table>

<hr>
```

```
<table border=1><tr>
```

```
<?php
```

```
echo "<td>PORT D.2 tila = $taulukko[0]</td> <td>PORT D.3 tila = $taulukko[1] </td>  
      <td> $taulukko[2]</td>" ; // Tulostetaan taulukosta tiedot
```

```
?>
```

```
</tr></table>
```

```
<br><br>
```

```
<a href="index.php?syotto=L">Ledi päälle</a><br>
```

```
<a href="index.php?syotto=l">Ledi pois päältä</a><br><br>
```

```
<a href="index.php?syotto=E">Rele 1 päälle</a><br>
```

```
<a href="index.php?syotto=e">Rele 1 pois päältä</a><br><br>
```

```
<a href="index.php?syotto=R">Rele 2 päälle</a><br>
```

```
<a href="index.php?syotto=r">Rele 2 pois päältä</a>
```

```
<br><br>
```

```
</body>
```

```
</html>
```

**Kuva lopullisesta testauskytkennästä**

