

Saimaan ammattikorkekoulu
Tekniikka Imatra
Sähkötekniikan koulutusohjelma

Pekko Jaatinen

VAPAA-AJAN ASUNNON HÄLYTYSJÄRJESTELMÄN TOTEUTUS MIKRO- KONTROLLERIN AVULLA

Opinnäytetyö 2010

TIIVISTELMÄ

Pekko Jaatinen

Vapaa-ajan asunnon hälytysjärjestelmän toteutus mikrorollerin avulla,

32 sivua, 6 liitettä

Saimaan ammattikorkeakoulu, Imatra

Tekniikka, Sähkötekniikan koulutusohjelma

Sähkövoimatekniikan suuntautumisvaihtoehto

Ohjaaja: Sähkötekniikan lehtori Timo Loukiala

Työn tarkoituksena on suunnitella ja kehittää hälytysjärjestelmä vapaa-ajan-asuntoon. Hälytysjärjestelmän tärkein ominaisuus on lähettää tekstiviesti hälytyksen tapahtuessa. Järjestelmän helppokäyttöisyyteen kiinnitettiin myös huomiota.

Hälytysjärjestelmän keskusyksikkö koostuu kolmesta osiosta, jotka ovat tehonlähde, ohjausyksikkö ja mikrokontrollerin ohjelmointi. Teholähde kytkennällä toteutetaan järjestelmän vaatimat käyttöjännitteet sekä akkukäyttöä varten akun ylläpitolaturi.

Ohjausyksikkö toteutetaan mikrokontrollerin avulla. Yhdessä oheiskomponenttien kanssa mikrokontrolleri tutkii antureiden tilaa ja tekee hälytyksen antureiden tilan perusteella.

Jotta mikrokontrolleri saataisiin toimimaan halutulla tavalla, pitää siihen kehittää sopiva ohjelmakoodi. Ohjelma toteutetaan c-kielellä käyttämällä Avr Studio 4 -ohjelmaa apuna koodin kirjoittamiseen ja testaamiseen. Ohjelmaa testattiin käytännössä ensin STK200-kehitysympäristössä, jonka avulla kokeiltiin ohjelman toimivuutta.

Ennen järjestelmän kokoonpanoa eri komponentteja testattiin ensin koekytkentäalustalla. Testausten jälkeen aloitettiin järjestelmän kokoonpano piirilevyille. Järjestelmää koottaessa tehtiin samalla järjestelmällisiä mittauksia, joiden avulla löydettiin virheitä. Virheet kokoonpanossa olisivat voineet aiheuttaa komponenttien tuhoutumisen. Järjestelmän viimeinen testaus suoritettiin ohjelman avulla.

Asiasanat: hälytysjärjestelmä, mikrokontrolleri

ABSTRACT

Pekko Jaatinen

Implementation of Leisure Home Alarm System through Microcontroller,

32 pages, 6 appendices

Saimaa University of Applied Sciences, Imatra

Technology, Electric engineering

Electrical power engineering

Tutor: Mr Timo Loukiala, Msc, Lecturer, SCP

The idea of this thesis is to design and develop an alarm system for a leisure home. The main function of this system is to send an SMS when alarm occurs. To the system's ease of use was also paid attention.

The system's central unit consists of three parts, which are the power source, control unit and microcontroller programming. With power source circuit was implemented the required operating voltage, and with battery charger the back up battery.

Control unit was carried out through the microcontroller. Together with peripheral components the microcontroller observed the state of the sensors and made alarm if the sensor was triggered.

The microcontroller needs a proper program code to work as designed. Program is written with c-language. AVR Studio 4 program is used to write and test the code. Program's functionality was tested in practice with the help of STK200 development board.

Before the system's assembly various components were first tested in the experimental circuit board. After testing started system's assembly to circuit board. Systematic measuring along assembly enabled finding coupling errors which could have caused serious damage to components. The last test was carried out successfully with the program code.

Keywords: alarm system, microcontroller

SISÄLTÖ

1	JOHDANTO	5
2	LÄHTÖKOHDAT	6
3	SUUNNITTELU.....	7
3.1	Mikrokontrolleri	7
3.2	Anturit	8
3.3	Led-valot ja sireeni.....	9
3.4	Tehonlähde	9
3.5	Akku.....	9
3.6	Akun kapasiteetti	9
3.7	Akkukäyttö	11
3.8	Akkulaturi	12
3.9	Käyttöjännitteiden regulointi	14
3.10	Regulaattoreiden jäähdytys	15
3.11	Sisääntulojen puskurointi.....	16
3.12	Lähtöjen puskurointi	18
3.13	GSM-modeemi	19
3.14	RS-232-muunninpiiri.....	21
4	ATTINY2313-MIKROKONTROLLERI	21
4.1	Ominaisuudet.....	22
4.2	Käytetyt ohjelmat	23
5	OHJELMA.....	24
5.1	Pääohjelma.....	24
5.2	Laskuri	24
5.3	Sarjaliikenne	25
5.4	Tekstiviestin lähetys.....	25
5.5	Ohjelman testaus.....	26
6	PIIRILEVYN VALMISTUS JA KOTELOINTI	26
6.1	Piirilevyn mitat.....	26
6.2	Komponenttien asettelu	27
6.3	Laitteen kotelointi.....	28
7	LAITTEISTON TESTAUS	28
7.1	Teholähteen testaus	28
7.2	Ohjauk kortin testaus	28
7.3	Kuormitustestaus	29
7.4	Testauksen yhteenveto.....	29
8	LOPPULUKU	30
	KUVAT	31
	TAULUKOT	31
	LÄHTEET.....	32

1 JOHDANTO

Työn tavoitteena on suunnitella hälytysjärjestelmä vapaa-ajan asuntoon kutsu-mattomien vieraiden varalta. Työn idean sain isältäni, joka ehdotti sitä muutama vuosi sitten. Noin vuosi sitten löysin internetistä sivun, jossa Nokian puhelin oli liitetty mikrokontrolleriin. Siitä sain ajatuksen, että samantyyppistä ratkaisua voisi kokeilla tähän projektiin. Toteutuksen lähtökohtana on järjestelmän helppokäyttöisyys ja toimintavarma sähkökatkon aikana. Helppokäyttöisyyden tarkoituksena on, että kukaan käyttäjä ei voi epäonnistua järjestelmän kytkemisessä päälle tai pois.

Työ koostuu suunnittelu, ohjelmointi, piirilevyn valmistus ja testausosuuksista. Suunnitteluosuudessa käsitellään, mitä komponentteja järjestelmään vaaditaan ja miten ne saadaan toimimaan oikein. Ohjelmointiosuudessa käsitellään mikrokontrollerin vaatimaa c-kielistä ohjelmaa. Ohjelmoinnin apuna käytetään Kandan valmistamaa STK200-kehitysympäristöä mikrokontrollerille, jonka avulla voidaan testata järjestelmän toiminta liittämällä siihen oheiskomponentteja. Piirilevyn valmistus osuudessa käsitellään eri vaiheita, joita vaaditaan piirilevyn koonpanoon.

Työssä käytettävää materiaalia olen etsinyt kirjoista ja internetistä.

2 LÄHTÖKOHDAT

Hälytysjärjestelmän suunnittelun lähtökohtana on suojata Saimaan rannalla sijaitseva vapaa-ajan asunto. Alueeseen kuuluu päärakennus, halkovarasto, käymälä ja varasto (Liite 1). Käymälä ja varasto sijaitsevat samassa rakennuksessa. Alueelle rakennetaan tulevaisuudessa myös erillinen rantsauna.

Valvottava alue voidaan jakaa ulko- ja sisätiloihin. Ulkotiloja voidaan valvoa esimerkiksi seuraavilla tavoilla.

- Tontille tulevan tien kulunvalvonta.
- Tontin rajan valvonta.
- Rakennuksien lähialueelle rajattu valvonta.
- Edellisiin yhdistetty videovalvonta.

Rakennuksien sisätilojen valvonta voidaan toteuttaa esimerkiksi seuraavilla tavoilla.

- Ovien valvonta magneettikytkimillä.
- Sisätilojen valvonta liiketunnistimilla.
- Lasinrikkoilmaisimella.
- Palonilmaisimella.
- Lämpötilan valvonta.

Kompromissiratkaisuna järjestelmä on rajattu valvomaan vain rakennusten sisätiloja. Ulkotilojen valvonta suurentaisi järjestelmän kokoa kohtuuttoman suureksi. Päärakennuksen valvontaan varattiin kolme hälytyssilmukkaa sekä tuloavainkytkimelle. Sekä varastolle että tulevalle saunalle varattiin yksi hälytyssilmukka. Lähtöjä varattiin hälytyssireenille sekä merkkivaloille yhteensä kolme kappaletta. Yhteensä tuloja ja lähtöjä on käytössä yksitoista, sisältäen kaksi sarjaliikenteen vaatimaa porttia. Valitussa mikrokontrollerissa on viisitoista kappaletta vapaita I/O-portteja, joten jäljelle jää vielä neljä kappaletta. Jäljelle jääneitä portteja voi käyttää myöhemmin lisävalvontaan tai kaukokäyttösovelluksiin.

Hälytyksen tapahtuessa lähetään tieto tekstiviestillä GSM-modeemin välityksellä. Tekstiviesti ilmoittaa, mikä anturi on aiheuttanut hälytyksen. Järjestelmän toimivuuden takaamiseksi sähkökatkosten aikana se varustetaan varmistusakulla.

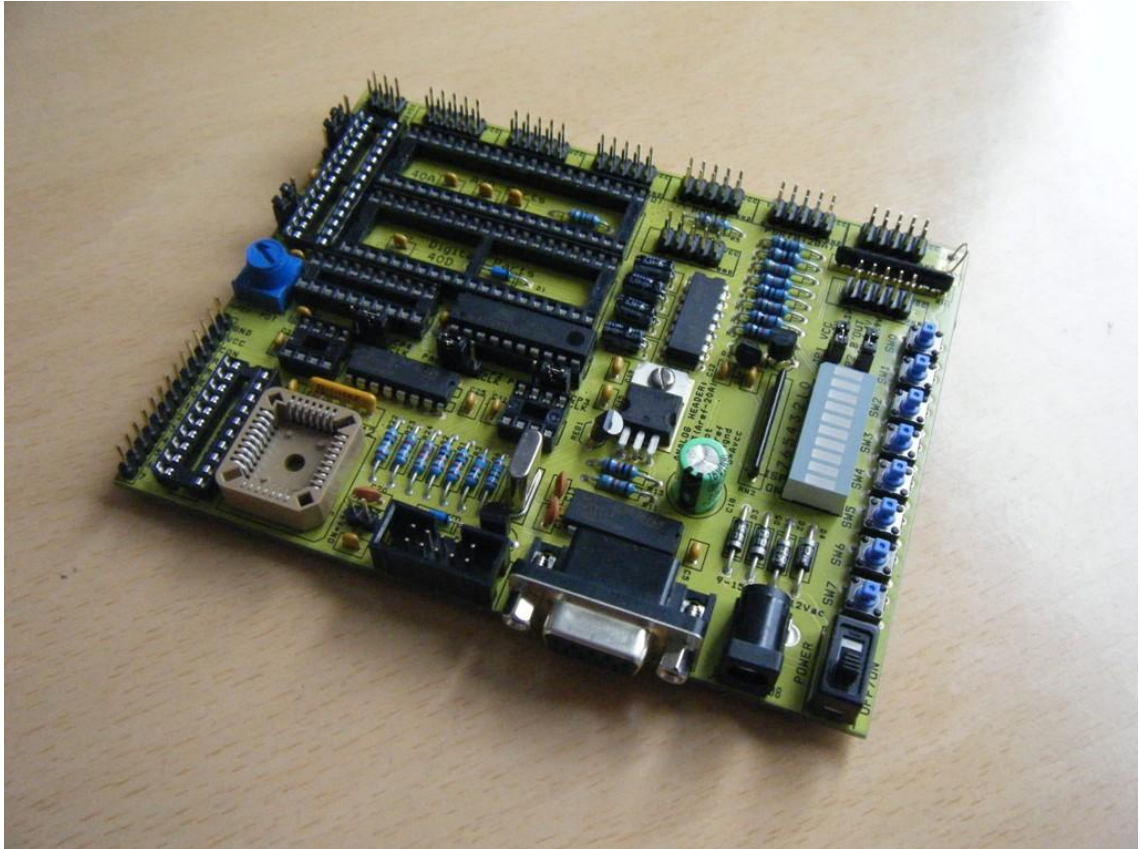
3 SUUNNITTELU

Järjestelmän rajauksen jälkeen aloitettiin suunnittelu keskeisten komponenttien ja osien valinnalla. Piirikaavion piirtämiseen käytettiin EAGLE Layout Editor ohjelmaa. Ohjelmasta on ladattavissa ilmaisversio, jonka ominaisuudet ovat varsin riittävät piirikaavion piirtämiseen. Ohjelma sisältää kattavan komponenttikirjaston, josta löytyy useiden valmistajien yleisimmät komponentit.

3.1 Mikrokontrolleri

Järjestelmään valittiin käytettäväksi Atmelin piiriperheeseen kuuluva mikrokontrolleri. Atmelin mikrokontrollerit ovat minulle tuttuja ammattikoulusta ja omista pienistä projekteista. Atmelin mikrokontrollerit ovat maailmalla suosittuja, joten internetistä löytyy useita sivustoja, joista on helppo etsiä tietoja.

Mikrokontrollerin kehitysympäristönä toimii STK200. Kehitysympäristö tarkoittaa piirilevyä, johon on integroitu erilaisia komponentteja, joita voi käyttää ohjelman testaamisessa. Esimerkiksi STK200 sisältää LED-valoja, painonappeja, sarjaliitännän, lcd-näyttöliitännän sekä mahdollisuuden ohjelmoida erikokoisia piirejä (Kuva 1).



Kuva 1. Stk200-kehitysympäristö.

Kehitysympäristöä käytettiin ohjelman esitestauksen apuna.

3.2 Anturit

Antureiksi valittiin ovimagneettikytkimet ja liiketunnistin. Molemmat ovat NC-tyyppisiä antureita. NC tarkoittaa normal close eli normaali tilassa anturi on kiinni. Päärakennukseen asennetaan kaksi ovikytkintä ja liiketunnistin. Saunalle ja varastolle varattiin omat hälytinsilmukat, joihin asennetaan ovikytkimet.

Järjestelmän kytkemiseksi valittiin sähköinen avainkytkin, jolla saadaan helposti kytkettyä järjestelmä päälle tai pois.

3.3 Led-valot ja sireeni

Mikrokontrollerin lähtöihin kytketään LED-valot ilmaisemaan hälyttimen tilaa. Hälytyssireeni kytketään omaan lähtöön. Valittu sireeni on ulkokäyttöön soveltuva äänenvoimakkuudeltaan 20 W 110 dB/1m. Sireenin käyttöjännitealue on 6-12 V ja virrankulutus 350 mA.

3.4 Tehonlähde

Teholähteen suunnittelun lähtökohtana on käyttää ulkoista tasajännitelähdettä, joka syöttää regulaattoreita (Liite 5). Regulaattoreiden ulostuloista saadaan järjestelmän vaatimat jännitetasot. Ulkoiseksi tasajännitelähteeksi valittiin vanhan kannettavan tietokoneen DC-hakkuri arvoiltaan 19V 4,7 A.

3.5 Akku

Normaalitilanteessa järjestelmää syötetään ulkoisella virtalähteellä. Tilanteessa, jossa sähkötkytkeytyvät syystä tai toisesta pois, pitää hälyttimen toimia. Tätä varten järjestelmään valittiin akkuvarmennus. Akuksi valittiin lyijyhyytelöakku, koska se on huoltovapaa ja toimii laajalla lämpötila-alueella (Yuasa 2008, 1).

3.6 Akun kapasiteetti

Akun kapasiteetin määrittämiseen vaikuttaa haluttu varmennusaika. Toisaalta iso kapasiteettinen akku on fyysisiltä mitoiltaankin suuri, joten näiden väliltä täytyy löytää sopiva kompromissiratkaisu.

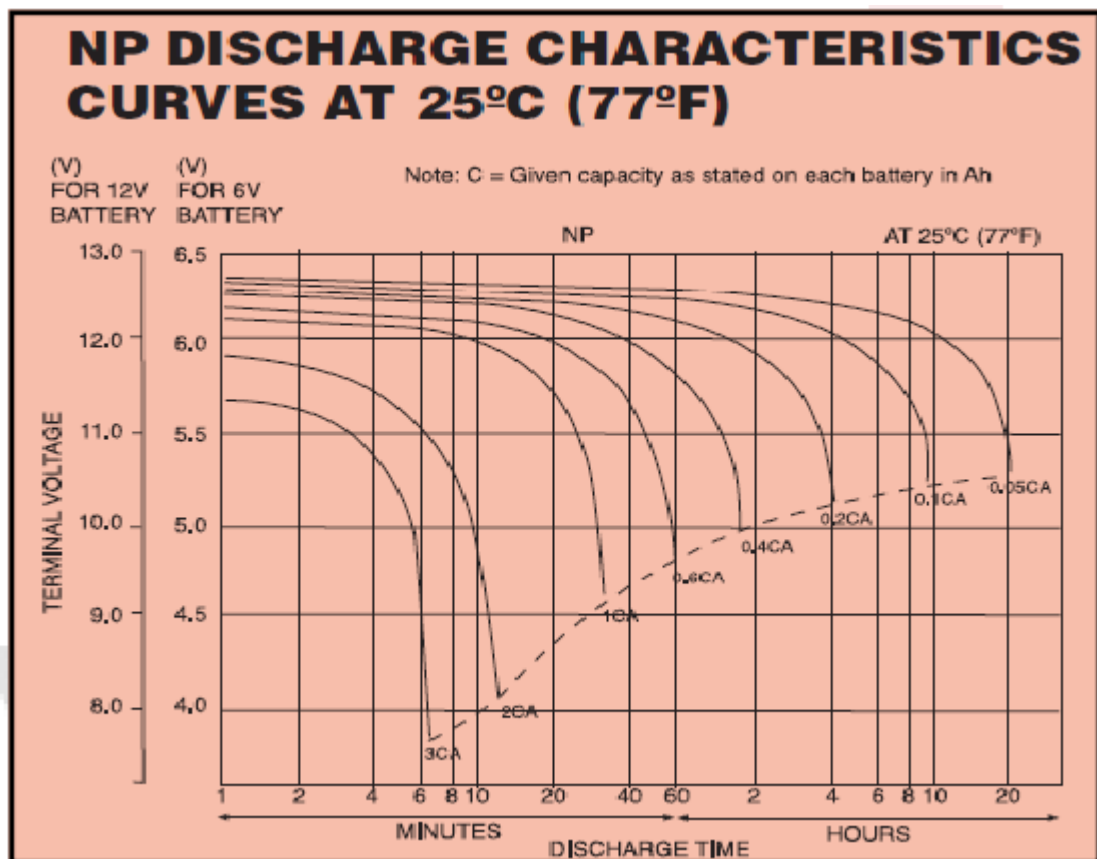
Taulukossa 1 on lueteltu kuluttavien kohteiden yhteisvirrankulutus, joka on 0,83 A. Haluttu varmistusaika sähkökatkon varalle on vähintään kaksi tuntia. Akuksi valittiin Yuasan valmistama 12 V ja 4 Ah lyijyhyytelöakku. Fyysiseltä kooltaan akku on 90x70x100 mm.

Taulukko 1. Kokonaisvirrankulutus järjestelmässä.

Jännitetaso	5V
ATTINY2313	200mA
MAX202CPE	20mA
LEDit	40mA
Anturit	90mA
Yhteensä	350mA

Jännitetaso	9V
Sireeni	350mA
Liiketunnistin	10mA
Tuuletin	120mA
Yhteensä	480mA

Akun datalehdessä löytyvällä kuvaajalla (kuva 2) voidaan arvioida akun tyhjenemisaika.



Kuva 2. Akun purkautumiskäyrä (Yuasa. 2008, 2).

Virrankulutuksella 0,83 A voidaan kuvasta arvioida akun kesto sähkökatkon aikana. Akun napajännite ei saa laskea alle 10,6 voltin, muuten 9 voltin regulaat-

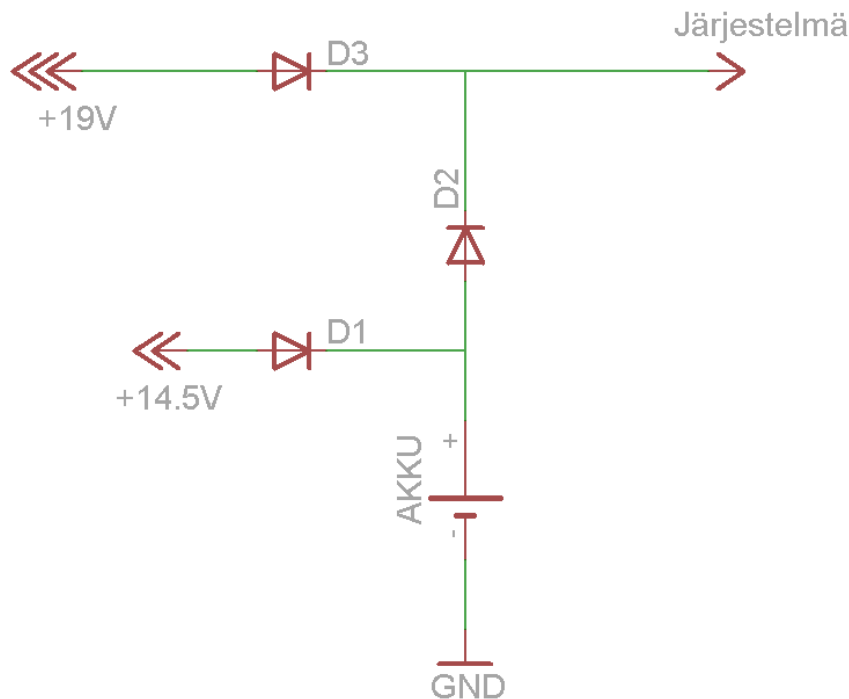
torin ei toimi. Koska järjestelmän jäähtytys ylimitoitettiin, ei tarvita järjestelmään suunniteltua tuuletinta. Virran kulutuksesta voidaan siis vähentää 120 mA. Kokonaisvirrankulutus on tällöin 0,71 A. Kuvasta 2 voidaan määrittää akun purkautumisaika kaavalla 1, jossa 0,2 on valmistajan määrittämä kerroin ja C on akun kapasiteetti ampeeritunteina eli tässä tapauksessa 4 Ah.

$$0,2 \cdot CA (1)$$

Tulokseksi saadaan 0,8 A, joka on hieman suurempi kuin kokonaiskulutuksen 0,71 A. Seuraamalla käyrää 0,2 CA ja etsimällä y-akselilta 10,6 V voidaan katsoa x-akselilta vastaava akun purkautumisaika. Ajaksi voidaan arvioida noin 3,9 h. Tämä aika on laskettu hälytyksen ollessa päällä. Toiminta-aika valmiustilassa on pidempi, koska sireeni ei ole kuluttamassa virtaa.

3.7 Akkukäyttö

Järjestelmän kytkeytyminen akun syötettäväksi toteutetaan yksinkertaisella diodikytkennällä (kuva 3).

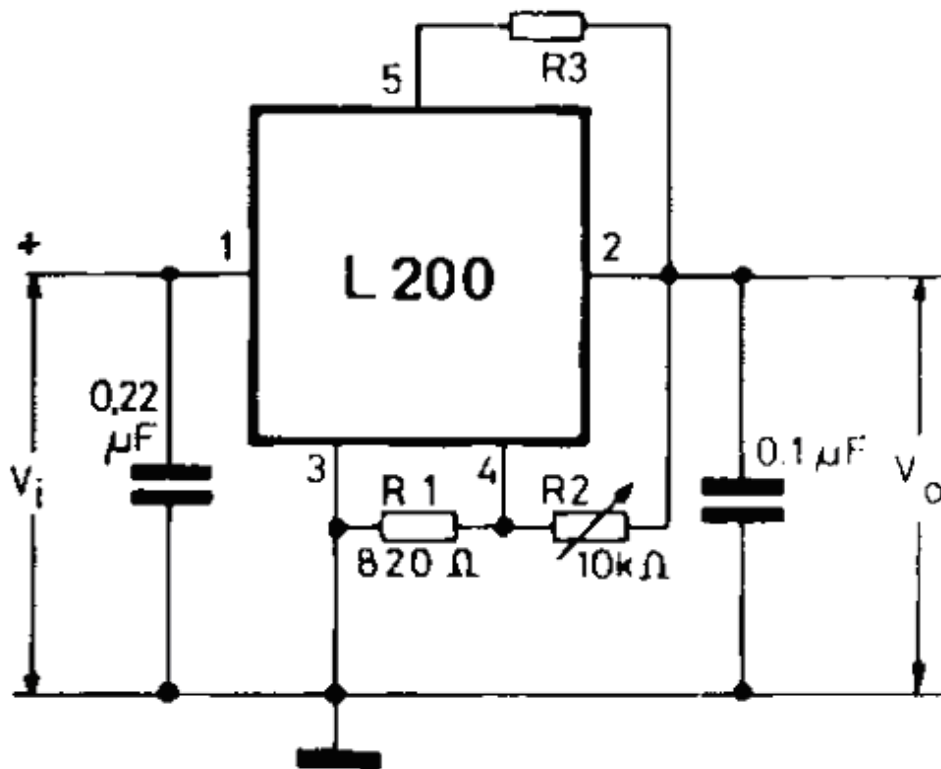


Kuva 3. Akkukäytön yksinkertaistettu kytkentäkuva.

Järjestelmää syötetään jännitelähteellä. Kun syöttöjännite laskee alle 11,3 volttin, kytkeytyy syöttö akun kautta. Syötön ollessa yli 11,3 V ei akku voi kytkeytyä syöttämään järjestelmää, koska diodin D2 katodilla on positiivisempi jännite. (Aaltonen & Kousa & Stor-Pellinen 2004, 61)

3.8 Akkulaturi

Järjestelmään valittiin 12 V akku, koska suuremmalla jännitteellä reguloinnissa syntyvä hukateho olisi liian suuri. Akku on tyypiltään lyijyhyytelöakku, joka on huoltovapaa suljetun rakenteen ansiosta. Akku tarvitsee ylläpitolatauksen ollakseen täydessä varauksessa tarvittaessa. Laturi toteutetaan L200-regulaattoria käyttäen (Kuva 4). Lyijyhyytelöakku täytyy ladata vakiojännitteellä, joten L200 regulaattori soveltuu tähän mainiosti.

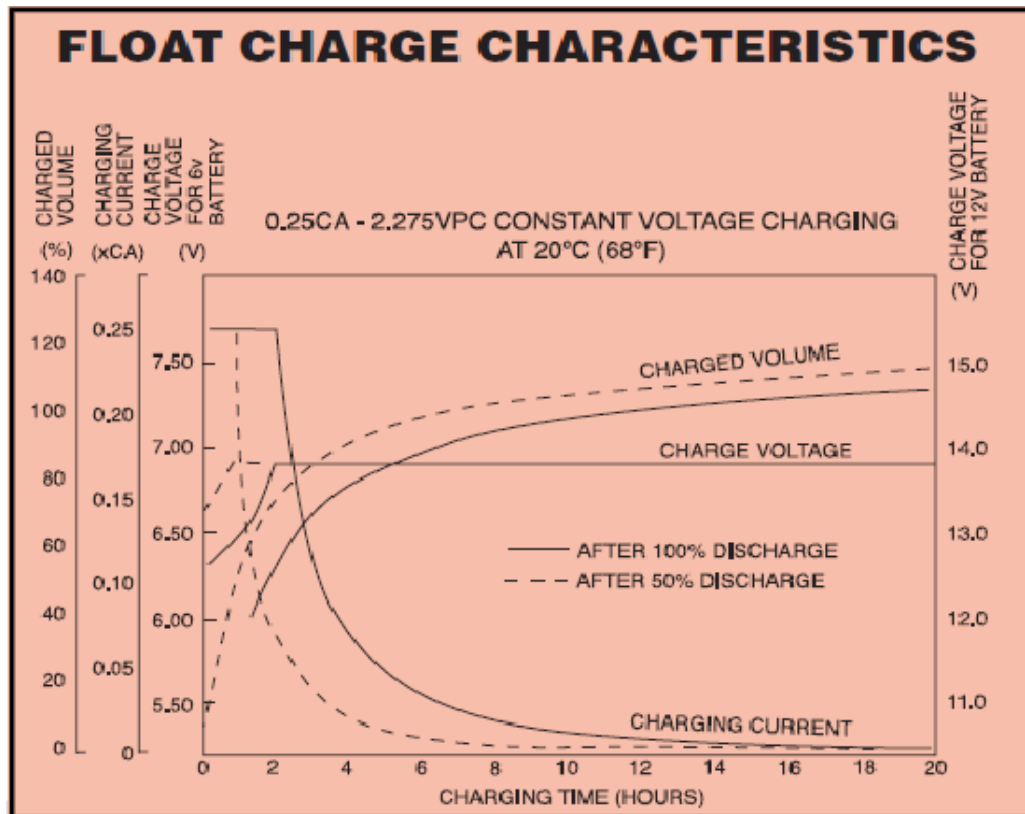


Kuva 4. L200 regulaattorin kytkentä (STMicroelektronics. 2000, 2).

Akun datalehddestä saadaan latausjännite 13,8 V ja latausvirta 0,175 A (kuva 5). Kaavasta 2 saadaan johdettua vastuksen R3 suuruus, jolla saadaan virta sovitettua oikean suuruiseksi. Kaavassa 3 V5-2 tarkoittaa jalkojen 5 ja 2 välistä jännite-eroa, joka on tyypillisesti 0,45 V (Liite 2).

$$I_{out} = \frac{V5-2}{R3} \quad (2)$$

$$R3 = \frac{V5-2}{I_{out}} = \frac{0,45V}{0,175A} = 2,57\Omega \quad (3)$$



Kuva 5. Akun latausjännitteen ja virran kuvaaja (Yuasa. 2008, 2).

Vastuksen R3 suuruudeksi saatiin 2,57 Ω kaavalla (3). Vastusarvo pyöristetään arvoon 2,5 Ω , jotta löydetään sopiva vastus E12-sarjasta. Tällöin virran arvoksi muodostuu kaavalla (4) 180 mA.

$$I_{out} = \frac{V5 - 2}{R3} = \frac{0,45V}{2,5\Omega} = 180mA \quad (4)$$

Latausjännitteen tiedetään olevan 13,8 V. Laturin jälkeen kytketään diodi myötäsuntaan, jolla saadaan estettyä akkukäytön kytkeytyessä virran syöttö laturiin päin. Vastusarvoja laskettaessa latausjännitteeseen täytyy lisätä diodin kynnyksjännite 0,7 V. Kaavassa 5 vastuksen arvon R1 tiedetään olevan 820 Ω ja vastuksen R2 tiedetään olevan 10 kΩ potentiometri. Arvo Vref on regulaattorin datalehdessä katsottuna 2,77 V (Liite 2). Kaavasta (5) johtamalla saadaan vastuksen R2 tarkka arvo.

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right) \quad (5)$$

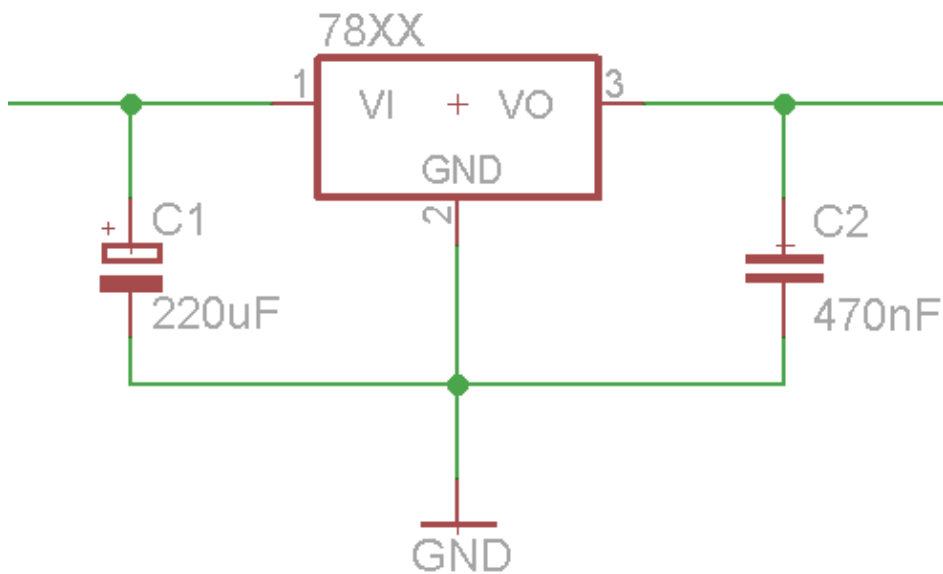
$$R2 = \left(\frac{U_{out}}{V_{ref}} - 1 \right) \cdot R1 = \left(\frac{14,5V}{2,77V} - 1 \right) \cdot 820\Omega = 3472\Omega \quad (6)$$

Potentiometrin R2 tarkka arvo laskettuna kaavassa (6).

3.9 Käyttöjännitteiden regulointi

Järjestelmä vaatii kolmen eri tason jännitteitä, jotka toteutetaan jänniteregulaattoreilla. Akkukäytössä syöttöjännite on suurimmillaan 11,3 V. Reguloinnilla voidaan toteuttaa suurimmillaan vain 9 voltin jännitetaso regulaattorissa syntyvän noin 2 voltin jännitehäviön vuoksi. Liiketunnistinta valittaessa täytyi ottaa huomioon saatavana oleva 9 voltin jännitetaso.

Regulaattoreiksi on valittu 7805 ja 7809, jotka ovat 5 ja 9 voltin jänniteregulaattoreita (kuva 6). Regulaattorilla 7809 syötetään liiketunnistinta, sireeniä sekä tuuletinta. Regulaattorilla 7805 syötetään muita järjestelmän komponentteja.



Kuva 6. Regulaattorin kytkentä.

Regulaattorin eteen ja taakse on kytketty suotokondensaattorit, jotka tasaavat jänniteheilahteluita.

3.10 Regulaattoreiden jäähdytys

Regulaattorit tarvitsevat jäähdytyslementtejä, koska ne muodostavat paljon lämpöä. Häviöteho syntyy jännite-erosta reguloidun ja sisääntulojännitteen välillä. Lasketaan 7805-regulaattorille jäähdytyksen tarve, kun regulaattori kehittää suurimman häviötehon.

$$\theta_{sa} = \frac{T_j - T_a}{P_D} - (\theta_{jc} + \theta_{cs}) \quad (7)$$

Kaavalla 7 saadaan laskettua jäähdytyslementin termisen resistanssin maksimiarvo θ_{sa} (sink to air). T_j tarkoittaa kaavassa komponentin liitoksen (junction) maksimiarvoa. Tämä arvo löytyy komponentin datalehdessä (STMicroelectronics 2000, 1). T_a on ympäröivän ilman lämpötila, joka tässä tapauksessa on arvioitu maksimilämpötila. P_D on komponentin hukateho (power dissipation), joka saadaan laskettua jännitehäviön ja virran avulla (kaava 8). θ_{jc} on komponentin ja rungon (junction to case) välinen termisen resistanssi. Tämä arvo löytyy kom-

ponentin datalehdessä (STMicroelektronics 2000, 1). θ_{cs} on kotelon ja jäähdytys-elementin (case to sink) välinen terminen resistanssi. (Aaltonen ym. 2004, 83).

Häviötehon laskennassa käytetään kaksinkertaista virran arvoa varmuuskertoimena.

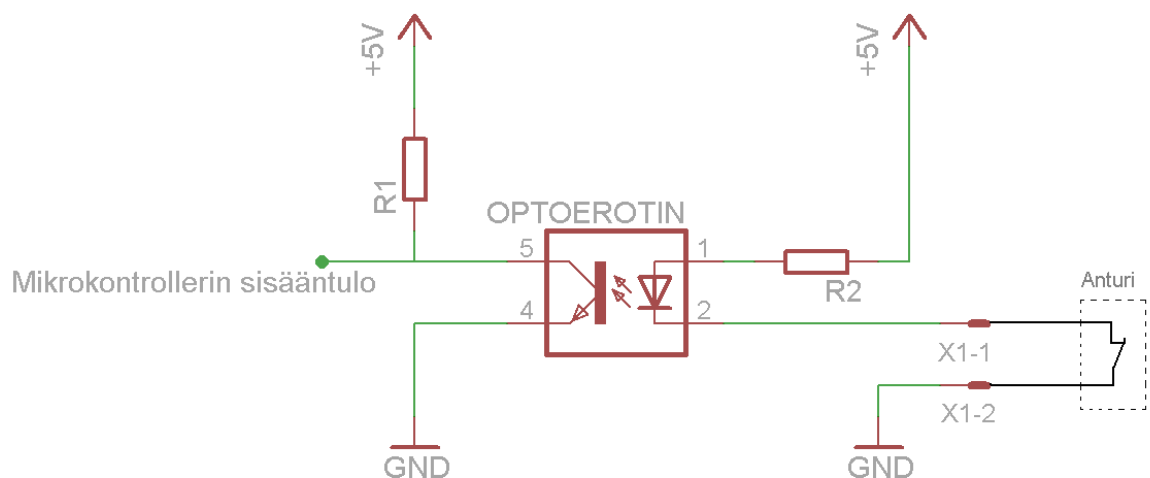
$$P_D = (18,3V - 5V) \cdot 0,7A = 9,31W \quad (8)$$

$$\theta_{sa(max)} = \frac{150^\circ C - 35^\circ C}{9,31W} - \left(5 \frac{^\circ C}{W} + 0,3 \frac{^\circ C}{W} \right) = 7,05 \frac{^\circ C}{W} \quad (9)$$

Kaavalla 9 saatiin laskettua jäähdytys-elementin maksimi terminen resistanssi. Käyttöön valittiin jäähdytys-elementti, jonka terminen resistanssi on $6,5^\circ C/W$.

3.11 Sisääntulojen puskurointi

Mikrokontrollerin tulot erotetaan galvaanisesti optoerottimen avulla tulosignaalista (Koskinen 2004, 208). Vaikka jännitetaso on sopiva suoraan kytkettäväksi mikrokontrolleriin, päädyttiin tulojen galvaaniseen erottamiseen, jotta mahdolliset häiriöt eivät vaikuttaisi mikrokontrollerin toimintaan (Kuva 7).



Kuva 7. Optoerottimen kytkentä.

Anturi kytketään X1-liitäntään kuvan mukaisesti. Anturit on valittu tyypiltään normaalisti kiinni oleviksi, joten normaalitilassa optoerottimen transistori johtaa ja menee alatilaa. Anturin avautuessa transistori ei enää johda, mikä aiheuttaa sisääntulon menevän ylätilaan.

Mikrokontrollerin sisääntulovirraksi valittiin 5 mA, jonka avulla mitoitettiin ylösve-tovastus R1 (Kaava 10).

$$R1 = \frac{5V}{5mA} = 1000\Omega \quad (10)$$

Optoerottimen LEDin virraksi valittiin 15 mA, joka on lähellä tyypillistä valitussa TLP620-optoerottimessa (Kaava 11).

$$R = \frac{5V}{15mA} = 333,3\Omega \quad (11)$$

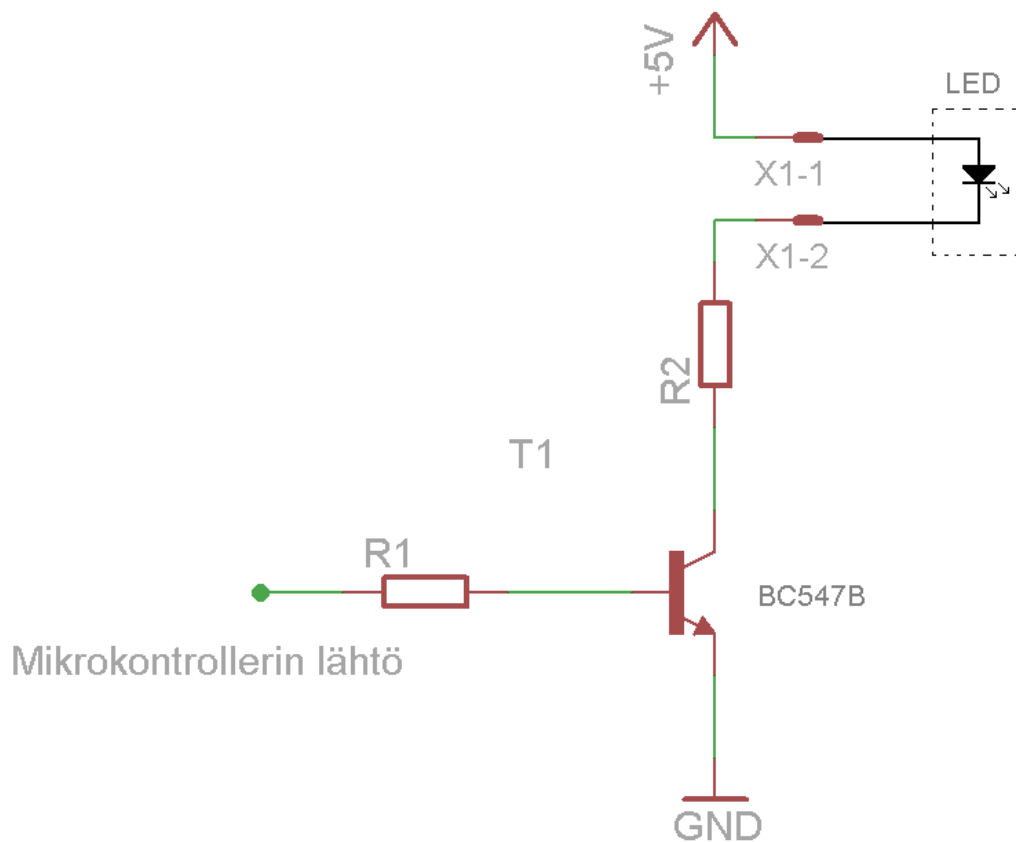
Lähin vastusarvo E12-sarjassa on 330 Ω . Koska pisin matka anturille on 75 m, täytyy ottaa huomioon johdon aiheuttama resistanssi. Kaapeli tyypiksi valittiin KLMA 2x0,8 (Kaava 12).

$$R = \frac{0,0168 \cdot 10^{-6} \frac{\Omega mm^2}{m} \cdot 150m}{0,8mm^2 \cdot 10^{-6}} = 3,15\Omega \quad (12)$$

Kaapelin aiheuttaman resistanssin ja mitoitettun resistanssin (330 Ω) suuruus on yhteensä 333,15 Ω . Tällä resistanssilla päästään hyvin lähelle haluttua mitoitus-virtaa 15 mA.

3.12 Lähtöjen puskurointi

Järjestelmässä on kolme lähtöä, jotka puskuroidaan transistorien avulla (Koskinen, J. 2004, 206). Merkkivaloina toimivat LEDit eivät käytännössä tarvitse puskurointia, mutta turvallisuuden takia niille ei oteta käyttöjännitettä suoraan mikrokontrollerin lähdöstä (kuva 8). Hälytyssireenin käyttöjännite on 9V. Sen ohjaamiseen tarvitaan kytkintransistoria.



Kuva 8. Lähtöjen puskurointikytkentä.

LEDeille valittiin transistoriksi BC547B ja sille lasketaan sopiva kantavastus R1 kantavirran avulla (Kaava 13).

$$I_b = \frac{I_c}{h_{FE}} = \frac{20mA}{200} = 0,100mA \quad (13)$$

Kantavirraksi saadaan 0,1 mA, joka kerrotaan varmuuskertoimella viisinkertaiseksi (Kaava 14).

$$R_b = \frac{U - U_{be}}{I_b} = \frac{5V - 0,7V}{0,500mA} = 8600\Omega \quad (14)$$

Valitaan E12 vastussarjasta 8200 Ω kantavastukseksi R1. Kantavirraksi saadaan tällöin 0,524 mA.

Sireenille valittiin transistori BC639 sen yhden ampeerin virrankeston takia. Kantavastuksen arvo lasketaan kantavirran avulla (Kaava 15).

$$I_b = \frac{I_c}{h_{FE}} = \frac{350mA}{40} = 8,75mA \quad (15)$$

Koska mikrokontrolleri voi antaa ulostuloon vain 40 mA, varmuuskerroin valitaan vain kolminkertaiseksi (Kaava 16).

$$R_b = \frac{U - U_{be}}{I_b} = \frac{5V - 0,7V}{26,25mA} = 163,8\Omega \quad (16)$$

Kantavastukseksi valitaan E12 sarjasta 150 Ω , joten kantavirraksi saadaan 28,7 mA.

3.13 GSM-modeemi

Kun mikrokontrolleri havaitsee antureissa normaalista poikkeavan tilan, se aiheuttaa hälytyksen, josta pitää saada tieto matkapuhelimeen. Eri vaihtoehtoja selvitettiin ja löytyi kolme erityyppistä ratkaisua. Ensimmäinen on erillinen gsm-modeemiyksikkö. Tämä suljettiin pois sen korkean hinnan takia.

Toinen tapa on käyttää Nokian puhelimen omaa Fbus-väylää tiedonsiirtoon ja sitä kautta lähettää tekstiviesti. Tämä oli varteenotettava vaihtoehto. Ongelmak-

si syntyi Fbus-väylän toimintakuntoon saaminen, koska se vaatii huomattavan paljon koodaamista verrattuna muihin vaihtoehtoihin (Embedtronics).

Kolmas vaihtoehto on käyttää sellaista matkapuhelinta, jossa on sisäänrakennettu modeemi. Modeemilla varustettua puhelinta voi ohjata lähettämään ja vastaanottamaan tekstiviestejä käyttämällä AT-komentoja (Vahtera, P. 2003, 292). AT-komennot ovat Hayes-yhtiön kehittämä epästandardi modeemien komento-kieli.

Modeemiksi valittiin Nokian 6310i-puhelin, koska sellainen löytyi käyttämättömänä ja sitä voidaan ohjata AT-komennoilla (Kuva 9). Puhelin liitetään mikrokontrollerin Rx- ja Tx-nastoihin rs-232-muunninpiirin välityksellä.

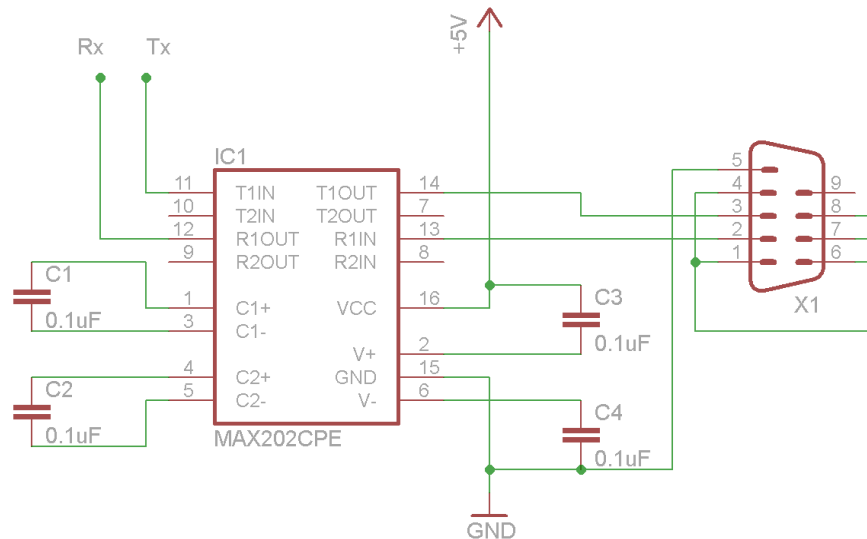


Kuva 9. Nokia 6310i ja datakaapeli.

Puhelimen takapuolella on liitäntä ulkoiselle antennille. Ulkoista antenniliitäntää voidaan tarvita, jos järjestelmä sijoitetaan metallikoteloon.

3.14 RS-232-muunninpiiri

Sarjaliikenteen muunninpiiriksi valittiin MAX202CPE (Kuva 10).



Kuva 10. MAX202CPE-piirin kytkentä.

Kuvasta 10 nähdään piirin kytkentä. Rx ja Tx kytketään mikrokontrollerin vastaaviin liitäntöihin. Piirin lähtö T1OUT ja tulo R1IN kytketään yhdeksän pinniseen urospuoleiseen d-liitimeen, johon saadaan puhelin kytkettyä liitäntäkaapelilla. Piirin ylimääräiset jalat voidaan jättää kytkemättä, koska piirissä on sisäänrakennetut 400 k Ω ylösvetovastukset (Liite 3). D-liittimen kättelyjohtimet kytketään niin sanotusti itse kätteleviksi (Huhtama).

4 ATTINY2313-MIKROKONTROLLERI

Atmel on yhdysvaltalainen puolijohdevalmistaja, joka valmistaa muun muassa mikrokontrollereita. Atmelin mikrokontrolleriperhe jakautuu tiny-, mega-, xmega- ja UC3-sarjoihin. Tiny sarja on pienikokoisin sisältäen vähiten integroituja ominaisuuksia. Mentäessä sarjoissa eteenpäin piirin jalkojen määrä kasvaa, sekä ominaisuudet lisääntyvät.

Kaikki ovat RISC-tyyppisiä prosessoreita. Tiny- ja megasarjat ovat 8-bittisiä. Xmega-sarjasta löytyy 8/16-bittisiä mikrokontrollereita. UC3-sarjan mikrokontrollerit ovat puhtaasti 32-bittisiä. (Atmel).

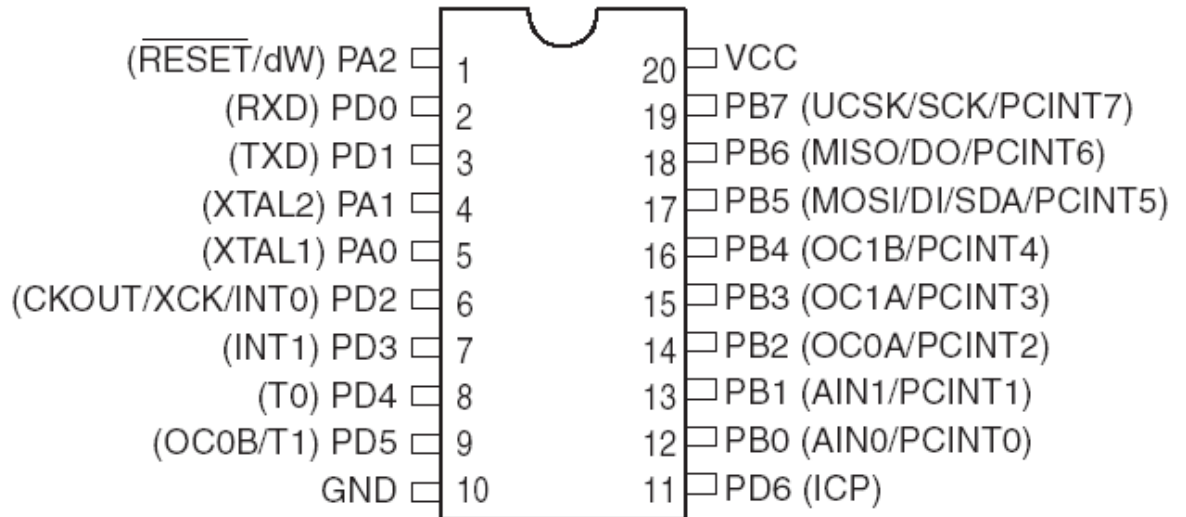
RISC tarkoittaa Reduced Instruction Set Computer eli käskykanta on supistettu. Käskyt suoritetaan liukuhihnaperiaatteella peräjälkeen (Koskinen 2004, 89).

4.1 Ominaisuudet

Mikrokontrolleri sisältää 2 kt Flash-ohjelmamuistia, jota voidaan kirjoittaa tai pyyhkiä 10 000 kertaa, 128 tavua ohjelmitavaa EEPROM-muistia, jota voidaan kirjoittaa tai pyyhkiä 100 000 kertaa. Mikrokontrolleri sisältää myös 128 tavua SRAM-muistia. Piirissä on 18 ohjelmitavaa I/O-jalkaa (Kuva 11).

Piiri sisältää yhden kappaleen 8- ja 16-bittistä ajastinta tai laskuria, neljä pulsinleveysmodulaatio kanavaa, analogisen komparaattorin sekä Full Duplex kaksisuuntaisen USART-piirin.

Työssä käytetään 16-bittistä laskuria tarvittavien ajastimien toteuttamiseen. USARTia (Universal Synchronous/Asynchronous Receiver-Transmitter) tarvitaan toteuttamaan sarjaliikenne puhelimen kanssa, jotta voidaan lähettää hälytysviesti.



Kuva 11. Attiny2313-mikrokontrollerin jalkajärjestys (Attiny2313, s. 2).

Piirissä on sisällä RC-oskillaattori, jota voidaan käyttää piirin kellopulssina. RC-oskillaattori on herkkä lämpötilan vaihtelulle, jonka takia se ei pidä aivan tarkkaa kellopulssia yllä. Tästä on haittaa ajastimien kanssa, koska aika saattaa muuttua ratkaisevasti lämpötilan takia. Ulkoisella kiteellä saadaan tarkka kellopulssi suuremmalla lämpötila-alueella.

Kiteeksi on valittu 3,6864 MHz. Tämän kiinteän taajuuden avulla saadaan tiedonsiirtonopeus sopivaksi puhelimen ja mikrokontrollerin välille ilman virheitä. Valittuun mikrokontrolleriin voidaan kytkeä maksimissaan 20 MHz kide. (Attiny2313).

4.2 Käytetyt ohjelmat

Ohjelman kirjoittamiseen käytettiin Atmelin AVR Studio 4 -ohjelmaa. Ohjelman saa ladattua ilmaiseksi Atmelin internetsivuilta. Ohjelmakoodin kääntämiseen käytetään AVR-GCC kääntäjää, jonka saa integroitua Avr Studion kanssa. Kääntäjä on avoimen lähdekoodin ohjelma, ja sen saa ladattua ilmaiseksi internetistä. Käännetty ohjelmakoodi siirretään AVRISP-U-ohjelmaan, jonka avulla koodi voidaan siirtää mikrokontrollerin muistiin. AVRISP-U on AVRISP USB-ohjelmointikaapelin mukana tuleva ohjelma.

5 OHJELMA

Mikrokontrollerin ohjelmointi toteutetaan c-kielellä, jota käytetään yleisesti mikrokontrollerien ohjelmoimiseen (Koskinen 2004, 169).

5.1 Pääohjelma

Pääohjelma tutkii avainkytkimen asentoa (Liite 6). Kun kytkin asetetaan 1-asentoon, kytketty hälytysjärjestelmä 60 sekunnin viiveellä päälle. Tämän jälkeen anturien avautuessa laukeaa hälytys, jolloin sireeni alkaa soimaan ja lähettää tapahtumasta tekstiviestin. Päärakennuksen toisen oven toiminnassa on 60 sekunnin viive, jotta järjestelmä saadaan kytkettyä pois päältä ilman hälytystä. Hälytyksen jälkeen voidaan järjestelmä nollata palauttamalla kytkin asentoon 0.

5.2 Laskuri

Järjestelmän toimintaan vaaditaan eripituisia viiveitä, jotka on paras toteuttaa esijaettua laskuria käyttäen. Esijaettua 16-bittistä laskuria käytetään, jotta voidaan saavuttaa vaaditut viiveet. Tavallisella laskurilla muisti vuotaisi yli, koska laskuri pystyy käsittelemään vain lukuja 0–65535 väliltä. Kaavalla (17) saadaan lasketuksi arvo, joka syötetään laskurin rekisteriin TCNT1. Laskuri laskee niin kauan, että rekisterin arvo on saavutettu. Tässä tapauksessa tavoitellaan yhden sekunnin aikaa, jota on helppo kasvattaa, kun halutaan suurempi viive. Yksi sekunti saavutetaan taajuudella 1 Hz, koska 1 Hz vastaa sekunnin mittaista kello sykliä. Kellotaajuus on 3,6864 MHz.

$$LaskurinArvo = \frac{\frac{KelloTaajuus}{Esijakaja}}{HaluttuTaajuus} - 1 \quad (17)$$

Taulukko 2. Esijakajaa vastaava laskurin arvo.

Esijakajan arvo	Laskurin arvo
1	3684399
8	460799
64	57599
256	14399
1024	3599

Taulukossa 2 lasketuista arvoista voidaan jättää pois kaksi ensimmäistä arvoa, koska ne menevät yli 65535:n. Valitaan seuraava arvo käytettäväksi eli esijakajan arvolla 64 laskurin arvo 57599. Ohjelmoinnissa määritellään esijakajan arvo rekisteriin TCCR1B. (Camera 2007).

5.3 Sarjaliikenne

Tiedonsiirtoon puhelimen kanssa tarvitaan sarjaliikennettä. Mikrokontrolleriin on siisäänrakennettu USART-piiri (Universal Synchronous/Asynchronous Receiver-Transmitter), jonka avulla yhteys voidaan toteuttaa. Haluttu tiedonsiirtonopeus tulee määrittää UBRR-rekisteriin (Baud Rate Register). Rekisterin arvo saadaan laskettua kaavalla (18), missä kellotaajuus on 3,6864 MHz ja siirtonopeus 9600 bps.

$$UBRR = \frac{Kellotaajuus}{USART_Siirtonopeus \cdot 16} - 1 \quad (18)$$

Kaavalla (18) saadaan UBRR-arvoksi 23. Jos luku ei olisi mennyt tasan, se aiheuttaisi tiedonsiirrossa viheitä. Jotta virheitä ei syntyisi, kellotaajuudeksi valittiin kyseinen 3,6864 MHz. (Vahtera 2003, 255-260).

5.4 Tekstiviestin lähetys

Tekstiviestin lähetys tapahtuu mikrokontrollerista puhelimeen AT-komentoja käyttäen (Taulukko 3). Puhelin asetetaan ensin tekstimoodiin lähettämällä ensimmäinen komento (Taulukko 3). Komennon perään syötetään enter-merkki,

jotta modeemi ottaa komennon vastaan ja valmistautuu seuraavaan komentoon. Toisella komenolla lähetetään tekstiviesti (Taulukko 3). Komennon heitto-merkkien sisään asetetaan vastaanottavan puhelimen numero. Perään syötetään enter-merkki, jolla modeemi asetaan vastaanottamaan viestiä. Viimeisenä lähetetään haluttu viesti. Viestin perään syötetään lopetuskomento, jolloin viesti lähetetään määriteltyyn numeroon. (Vahtera 2003, 292-296).

Taulukko 3. Puhelimelle lähetettävät komennot.

1. at+cmgf=1
2. at+cmgf="050xxxxxx"
3. Viesti
4. Lopetuskomento

5.5 Ohjelman testaus

Ohjelmaa testattiin stk-200-kehitysympäristön (Kuva 1) avulla. Kehitysympäristössä olevat painonapit kuvasivat käytettäviä antureita ja LEDit suunniteltuja lähtöjä. Gsm-modeemi kytkettiin kortin rs-232 liitántään, joten ohjelmaa voitiin testata täysivaltaisesti.

6 PIIRILEVYN VALMISTUS JA KOTELOINTI

Jäjestelmän prototyypin kokoaminen päätettiin tehdä kytkentäpiirilevyille. Vaikka kytkennän toteutus on aikaavievää, mahdollisten virheiden korjaus onnistuu kuitenkin helposti.

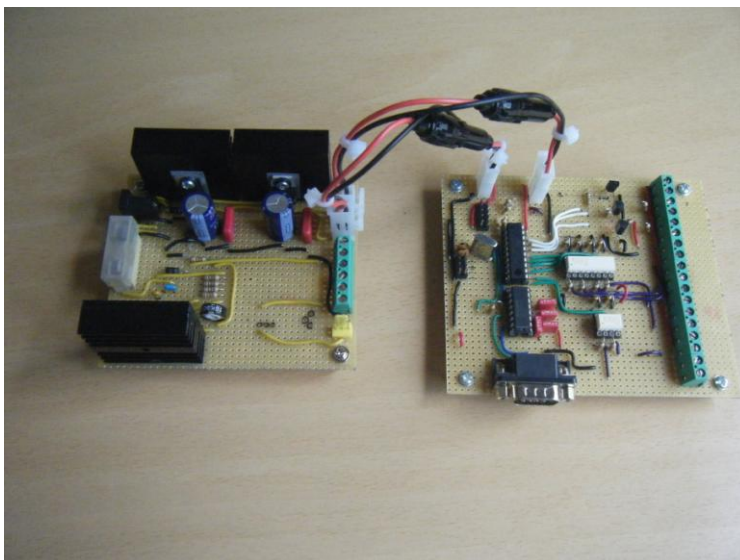
6.1 Piirilevyn mitat

Teholähde ja ohjausjärjestelmä tehtiin eri piirilevyille, koska piirilevyn koon takia ne eivät olisi mahtuneet samalle levyille. Toinen syy jakaa koko järjestelmä kahdelle piirilevyille on saada kytkennöistä selkeämmät (Liite 4 ja 5). Kytkentöjen selkeydestä on hyötyä mahdollisten virheiden etsimisessä.

Piirilevyn koon mitoituksen aloitin sommittelemalla komponentteja kytkentäpiirilevylle. Teholähteen ulkoasun suunnittelemisessa täytyi ottaa huomioon jäähdytyselementtien viemä tila (Kuva 12). Teholähteen piirilevyn mitat ovat 87x100 mm ja ohjausjärjestelmän piirilevyn mitat ovat 100x117 mm.

6.2 Komponenttien asettelu

Komponenttien asettelussa pyrittiin siihen, että kytkennästä tulisi mahdollisimman yksinkertainen (Kuva 12). Komponentit aseteltiin niin, että välttyttäisiin pitkiltä johdotuksilta, jotka ovat alttiita häiriöille (Reitmaa & Gustafsson 2004, 11-14). Induktiivisten häiriöiden minimoimiseksi virtajohtimet pyrittiin asettamaan mahdollisimman kauaksi tiedosiirtojohtimista (Aaltonen 317-318). Tärkeintä oli sijoittaa mikrokontrolleri ja sarjaliikennemuunninpiiri sekä RS-232-liitin niin, että tiedosiirtojohtimien pituus olisi mahdollisimman lyhyt. Tiedonsiirtojohtimet sijoitettiin myös kulkemaan eri suuntaan kuin muut johtimet. Tällä pyritään estämään ylikuulumista (Reitmaa & Gustafsson 2004, 44-45). Ruuviliittimet sijoitettiin kaikki samalle sivulle piirilevyä.



Kuva 12. Järjestelmä koottuna.

6.3 Laitteen kotelointi

Koteloinnin suunnittelussa täytyi ottaa huomioon kotelon koko ja liitäntöjen suunta. Kotelon sisälle sijoitetaan kiinteästi kaksi pistorasiaa, joista järjestelmä ja GSM-modeemi saavat syöttöjännitteen. Hälytysjärjestelmän lisäksi koteloon sijoitetaan akku, DC-jännitelähde ja GSM-modeemi. Koteloon sijoitetaan paljon tilaa vieviä elementtejä, joten sen tulee olla sisämitoiltaan vähintään 400x300x100 mm. Kotelon tulee olla rakenteeltaan sellainen, että siihen on mahdollista asentaa ulostuloja joka sivulle.

7 LAITTEISTON TESTAUS

Järjestelmän mittaukset suoritettiin Amprobe 5XP-A-yleismittarilla. Tärkeintä mittauksissa oli löytää mahdolliset kokoamisvaiheessa tapahtuneet virhekytkennät sekä tarkastaa järjestelmän jännitetasot. Ennen järjestelmän juottamista piirilevyyn testattiin eri komponentteja koekytkentäalustalla.

7.1 Teholähteen testaus

Teholähteen kokoaminen suoritettiin regulaattori kerrallaan. Tällä tavalla voitiin helposti mitata kytkentä virheiden varalta. Kytkennän varmistamisen jälkeen mitattiin regulaattorin antama ulostulojännite. Viiden voltin jänniteregulaattorin ulostulojännitteeksi mitattiin 5,05 V. Yhdeksän voltin jänniteregulaattorin ulostulojännitteeksi mitattiin 8,99 V. Viimeisenä mitattiin akkulaturin jännite, jonka ulostulojännite on säädettävissä potentiometrin avulla. Ulostulojännitteen arvoksi säädettiin 13,79 V.

7.2 Ohjauskortin testaus

Ohjauskortin kasaamisen yhteydessä tehtiin jatkuvasti mittauksia virhekytkentöjen välttämiseksi. Järjestelmällisten mittausten ansiosta löydettiin juotosvirheitä,

jotka olisivat voineet aiheuttaa komponenttien tuhoutumisen. Virheiden korjaamisen jälkeen kytkettiin korttiin jännite ja mitattiin ic-piirien käyttöjännitteet piirien ollessa pois paikoiltaan. Kun kytkentä todettiin mittauksilla virheettömäksi, aloitettiin testaukset ohjelmallisesti. Ohjelmallisessa testauksessa tarkastettiin tulojen, lähtöjen ja viestin lähetyksen toiminta. Akkukäyttöä testattiin irrottamalla teholähteen syöttö ja testaamalla järjestelmän toiminta tässä tilanteessa.

7.3 Kuormitustestaus

Kuormitustestauksen tarkoituksena oli selvittää, kuinka paljon komponentit lämpenevät vakiokuormituksella. Järjestelmää kuormitettiin taulukon 1 (s. 10) mukaisilla virroilla. Mittaukset suoritettiin huoneen lämpötilan ollessa 22 °C. Järjestelmää kuormitettiin yhden tunnin ajan samanaikaisesti seuraamalla komponenttien lämpötilaa. Lämpötiloja mitattiin alussa aistinvaraisesti kokeilemalla komponenttien pintalämpötilaa ja lopulta siirryttiin digitaaliseen lämmön mittaukseen. Ainoastaan 5 ja 9 voltin jänniteregulaattorien todettiin lämpenevän yli kädenlämpöiseksi. Tunnin aikana regulaattorien lämpötilaksi vakiintui noin 41 °C.

7.4 Testauksen yhteenveto

Järjestelmällisten mittausten ansiosta löydettiin inhimillisistä virheistä johtuvat kytkentävirheet sekä varmistettiin jännitetasojen oikeellisuus. Ohjelmallisella testauksella saatiin testattua järjestelmä kokonaisuudessaan ja varmistettua, että suunnitellut ominaisuudet toimivat. Jos halutaan, että regulaattorit pysyvät viileämpinä, täytyy tuuletin ottaa käyttöön. Tuulettimen käyttöönotto laskee hie-man akkukäytön valmiusaikaa.

8 LOPPULUKU

Opinnäytetyössä saavutettiin se, mitä lähtökohdissa määriteltiin. Työn tekemistä helpotti selvät rajaukset, joten työ ei päässyt paisumaan liian suureksi. Aikataulun ansiosta pysyttiin hyvin suunnitellussa tavoitteessa.

Oppimisen kannalta opinnäytetyön tekeminen oli erittäin antoisaa. Opin, mitä vaikeuksia voi tulla vastaan elektronisen laitteen suunnittelussa ja toteutuksessa. Tämän kaltaisen laitteen tekemistä voisi verrata talon rakentamiseen. Pohjatöiden on oltava kunnossa, jos halutaan rakennuksen kestävän.

Projekti etenee tämän jälkeen järjestelmän asentamisella suunniteltuun kohteeseen, jossa suoritetaan viimeinen testaus ja tarvittaessa säädetään ohjelmaa. Jatkokehityksen kannalta olen suunnitellut parantavani laitteen energiatehokkuutta. Energiatehokkuuden parantaminen pienentäisi laitteiston kuluttamaa virtaa, joten se toimisi akun varassa kauemmin.

KUVAT

Kuva 1. Stk200-kehitysympäristö, s. 8

Kuva 2. Akun purkautumiskäyrä (Yuasa. 2008, 2), s. 10

Kuva 3. Akkukäytön yksinkertaistettu kytkentäkuva, s. 12

Kuva 4. L200 regulaattorin kytkentä (STMicroelectronics. 2000, 2), s. 13

Kuva 5. Akun latausjännitteen ja virran kuvaaja (Yuasa. 2008, 2), s. 14

Kuva 6. Regulaattorin kytkentä, s. 15

Kuva 7. Optoerottimen kytkentä, s. 17

Kuva 8. Lähtöjen puskurointikytkentä, 19

Kuva 9. Nokia 6310i ja datakaapeli, s. 21

Kuva 10. MAX202CPE-piirin kytkentä, s. 22

Kuva 11. Attiny2313-mikrokontrollerin jalkajärjestys (Attiny2313, 2), s. 23

Kuva 12. Järjestelmä koottuna, s. 28

TAULUKOT

Taulukko 1. Kokonaisvirrankulutus järjestelmässä, s. 10

Taulukko 2. Esijakajaa vastaava laskurin arvo, s. 25

Taulukko 3. Puhelimelle lähetettävät komennot. s. 26

LÄHTEET

Aaltonen, J. & Kousa, S. & Stor-Pellinen, J. 2004. Elektroniikan perusteet. Helsinki: Limes Ry

Attiny2313. Avr mikrokontroller.

http://www.atmel.com/dyn/resources/prod_documents/DOC2543.PDF

(Luettu 20.1.2010)

Atmel. Avr solutions.

http://www.atmel.com/products/avr/default.asp?family_id=607&source=main_nav

(Luettu 28.1.2010)

Camera, D. 2007. Avrfreaks. Guide to AVR Timers.

<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=50106>

(Luettu 14.1.2010)

Embedtronics. Nokia f-bus protocol.

<http://www.embedtronics.com/nokia/fbus.html> (Luettu 11.1.2010)

Huhtama, Kari. Pc:n sarjaportin nastajärjestys ja nollamodeemikaapeli.

<http://koti.mbnet.fi/~huhtama/ele/index.php?si=ml03.sis> (Luettu 19.4.2010)

Koskinen, J. 2004. Mikrotietokonetekniikka sulautetutjärjestelmät. Helsinki: Ota-va

Reitmaa, I. & Gustafsson, J. 1998. Varma digitaalelektroniikka. Helsinki: Otatie-to Oy

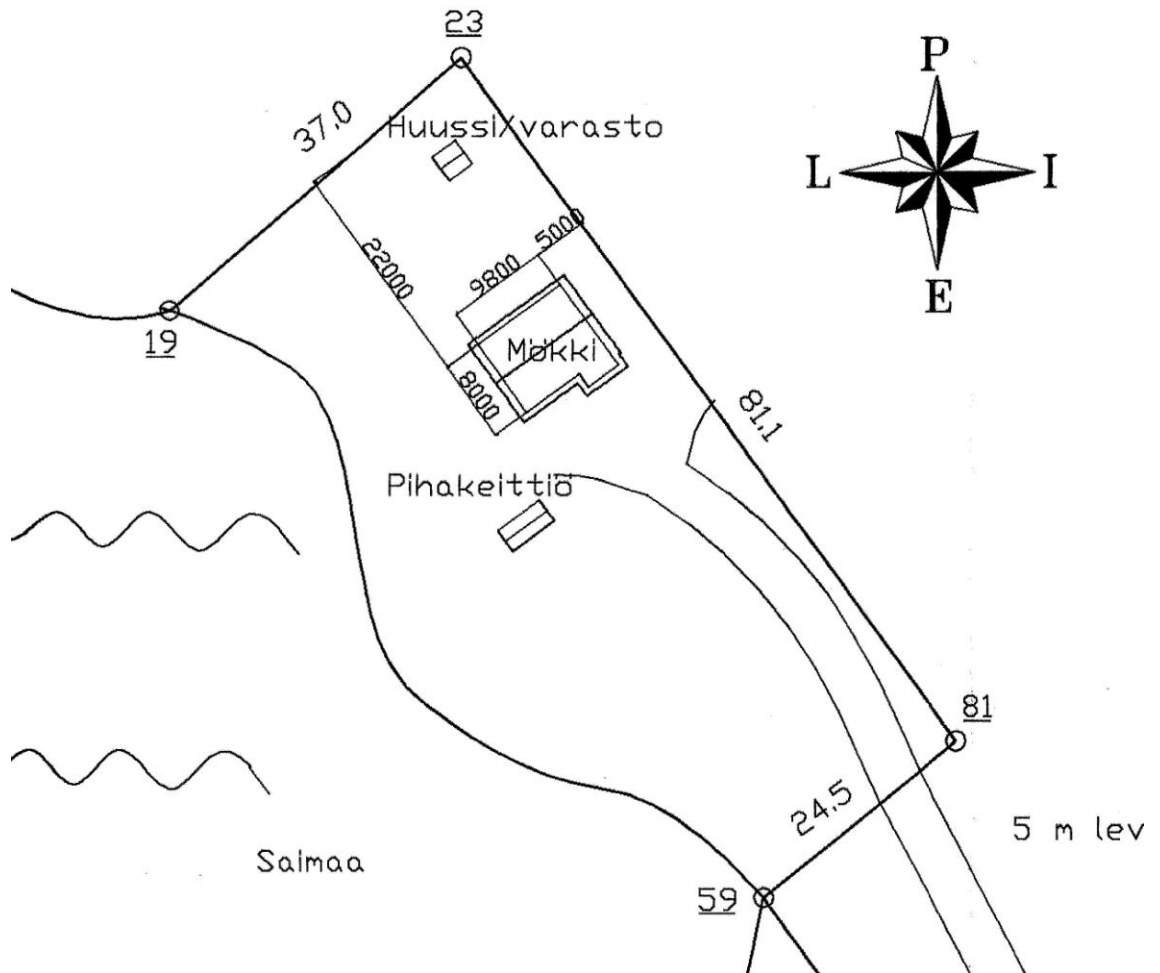
STMicroelektronics. 2000. L200 Adjustable voltage and current regulator.

<http://www.st.com/stonline/books/pdf/docs/1318.pdf> (Luettu 1.2.2010)

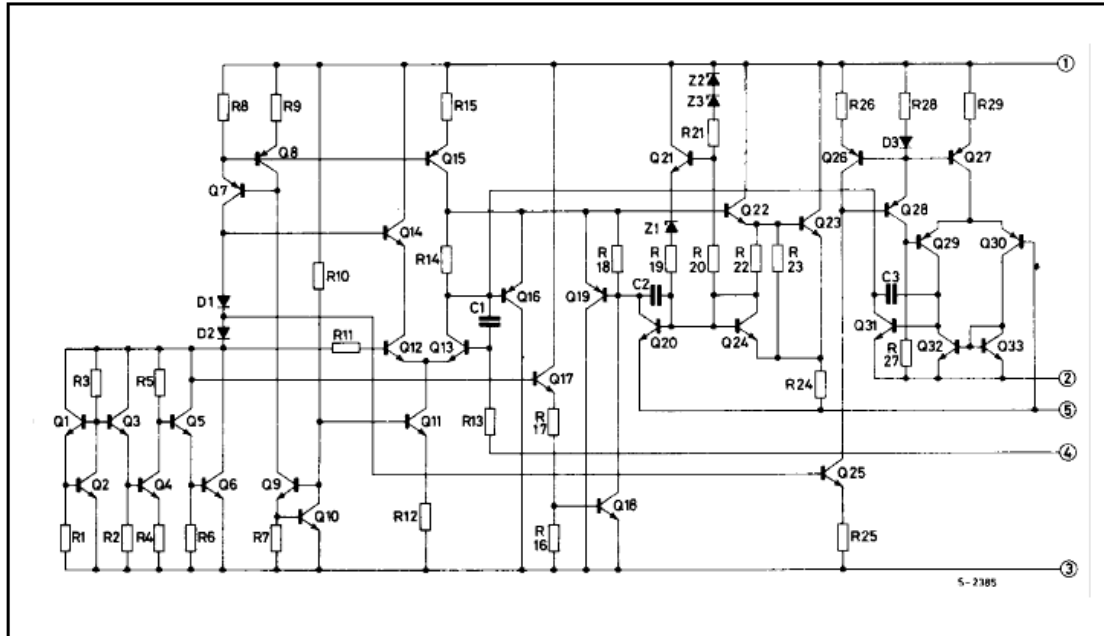
Vahtera, P. 2003. Mikro-ohjaimen ohjelmointi c-kielellä. Helsinki: Wsoy

Yuasa. 2008. NP4-12 battery.

http://www.yuasabatteries.com/pdfs/NP_4_12_DataSheet.pdf (Luettu 2.2.2010)



SCHEMATIC DIAGRAM



ELECTRICAL CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
--------	-----------	-----------------	------	------	------	------

VOLTAGE REGULATION LOOP

I_d	Quiescent drain Current (pin 3)	$V_i = 20\text{ V}$		4.2	9.2	mA
e_N	Output Noise Voltage	$V_o = V_{ref}$ $B = 1\text{ MHz}$ $I_o = 10\text{ mA}$		80		μV
V_o	Output Voltage Range	$I_o = 10\text{ mA}$	2.85		36	V
$\frac{\Delta V_o}{V_o}$	Voltage Load Regulation (note 1)	$\Delta I_o = 2\text{ A}$ $\Delta I_o = 1.5\text{ A}$		0.15 0.1	1 0.9	% %
$\frac{\Delta V_i}{\Delta V_o}$	Line Regulation	$V_o = 5\text{ V}$ $V_i = 8\text{ to }18\text{ V}$	48	60		dB
SVR	Supply Voltage Rejection	$V_o = 5\text{ V}$ $\Delta V_i = 10\text{ V}_{pp}$ $f = 100\text{ Hz}$ (note 2) $I_o = 500\text{ mA}$	48	60		dB
ΔV_{i-o}	Droupout Voltage between Pins 1 and 5	$I_o = 1.5\text{ A}$ $\Delta V_o \leq 2\%$		2	2.5	V
V_{ref}	Reference Voltage (pin 4)	$V_i = 20\text{ V}$ $I_o = 10\text{ mA}$	2.64	2.77	2.86	V

L200

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
ΔV_{ref}	Average Temperature Coefficient of Reference Voltage	$V_i = 20\text{ V}$ $I_o = 10\text{ mA}$ for $T_j = -25\text{ to }125\text{ }^\circ\text{C}$ for $T_j = 125\text{ to }150\text{ }^\circ\text{C}$		-0.25 -1.5		mV/ $^\circ\text{C}$ mV/ $^\circ\text{C}$
I_4	Bias Current and Pin 4			3	10	μA
$\frac{\Delta I_4}{\Delta T \cdot I_4}$	Average Temperature Coefficient (pin 4)			-0.5		%/ $^\circ\text{C}$
Z_o	Output Impedance	$V_i = 10\text{ V}$ $I_o = 0.5\text{ A}$ $V_o = V_{ref}$ $f = 100\text{ Hz}$		1.5		m Ω

CURRENT REGULATION LOOP

V_{SC}	Current Limit Sense Voltage between Pins 5 and 2	$V_i = 10\text{ V}$ $I_5 = 100\text{ mA}$ $V_o = V_{ref}$	0.38	0.45	0.52	V
$\frac{\Delta V_{SC}}{\Delta T \cdot V_{SC}}$	Average Temperature Coefficient of V_{SC}			0.03		%/ $^\circ\text{C}$
$\frac{\Delta I_o}{I_o}$	Current Load Regulation	$V_i = 10\text{ V}$ $I_o = 0.5\text{ A}$ $I_o = 1\text{ A}$ $I_o = 1.5\text{ A}$ $\Delta V_o = 3\text{ V}$		1.4 1 0.9		% % %
I_{SC}	Peak Short Circuit Current	$V_i - V_o = 14\text{ V}$ (pins 2 and 5 short circuited)			3.6	A

Note 1: A load step of 2 A can be applied provided that input-output differential voltage is lower than 20 V (see Figure 3).

Note 2: The same performance can be maintained at higher output levels if a bypassing capacitor is provided between pins 2 and 4.

Figure 3. Typical Safe Operating Area Protection.

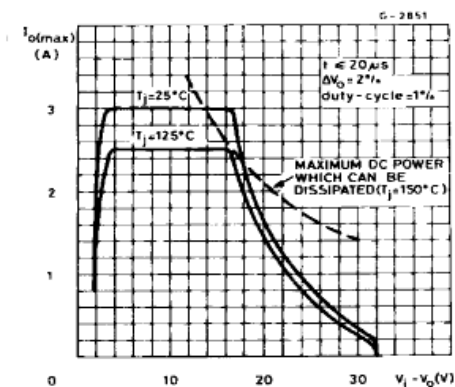
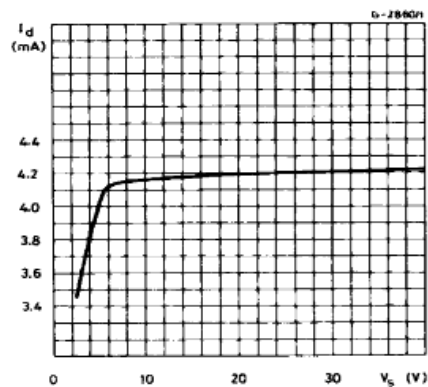


Figure 4. Quiescent Current vs. Supply Voltage.



+5V, RS-232 Transceivers with 0.1µF External Capacitors

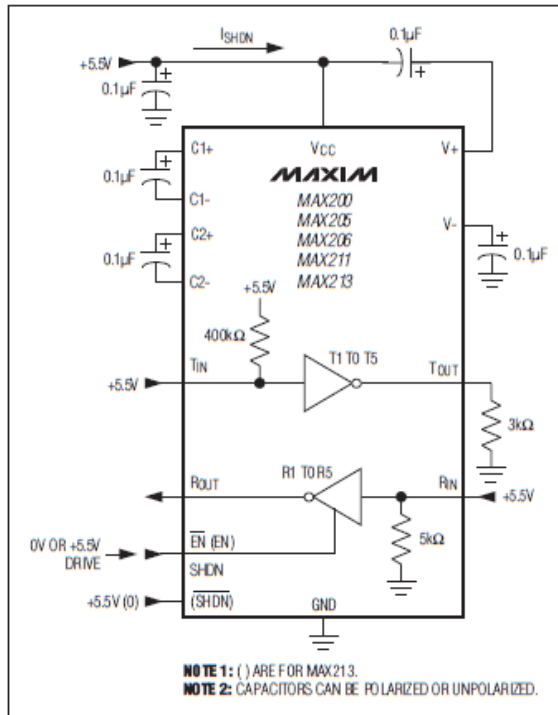


Figure 1. Shutdown Current Test Circuit

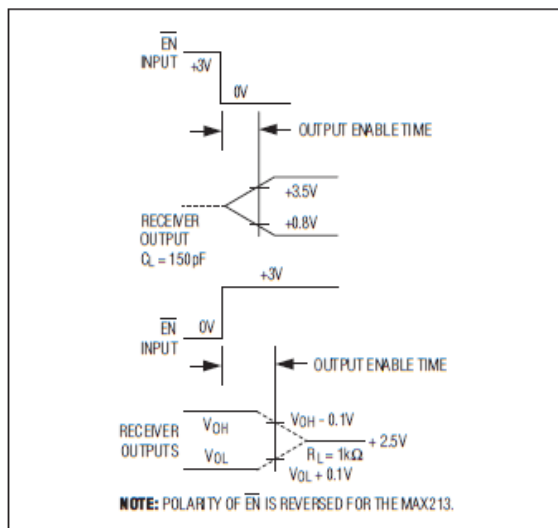


Figure 2. Receiver Output Enable and Disable Timing

Detailed Description

The MAX200–MAX209/MAX211/MAX213 consist of three sections: charge-pump voltage converters, drivers (transmitters), and receivers. Each section is described in detail.

+5V to ±10V Dual Charge-Pump Voltage Converter

The +5V to ±10V conversion is performed by two charge-pump voltage converters (Figure 4). The first uses capacitor C1 to double +5V to +10V, storing +10V on the V+ output filter capacitor, C3. The second charge-pump voltage converter uses capacitor C2 to invert +10V to -10V, storing -10V on the V- output filter capacitor, C4.

The MAX201 and MAX209 include only the V+ to V- charge pump, and are intended for applications that have a VCC = +5V supply and a V+ supply in the +9V to +13.2V range.

In shutdown mode, V+ is internally connected to VCC by a 1kΩ pulldown resistor and V- is internally connected to ground by a 1kΩ pullup resistor.

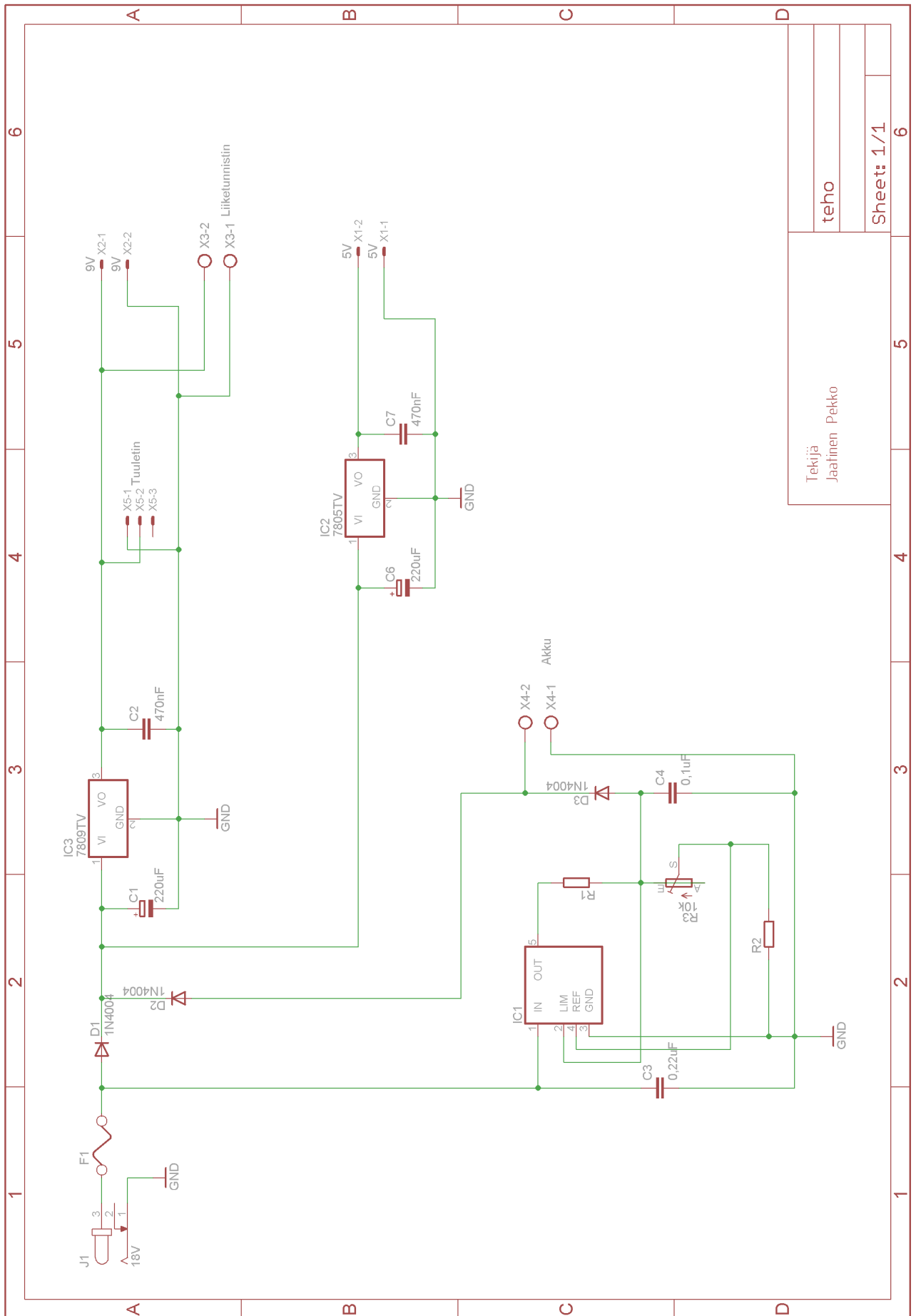
RS-232 Drivers

When VCC = +5V, the typical driver output-voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver. The output swing is guaranteed to meet the EIA/TIA-232E and V.28 specifications, which call for ±5V minimum output levels under worst-case conditions. These include a minimum 3kΩ load, VCC = +4.5V, and the maximum operating temperature. The open-circuit output-voltage swing ranges from (V+ - 0.6V) to V-.

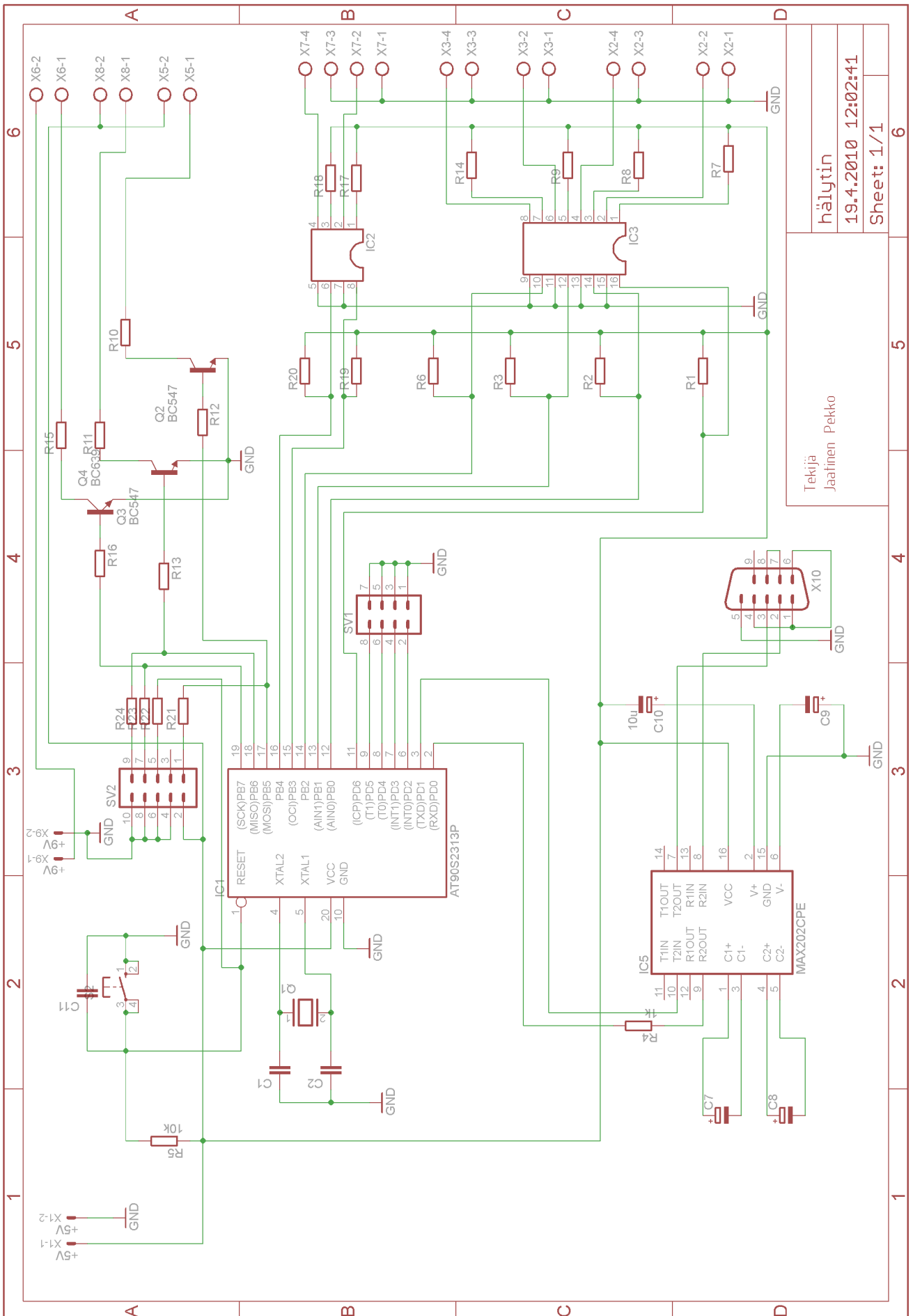
Input thresholds are both CMOS and TTL compatible. The inputs of unused drivers can be left unconnected since 400kΩ pullup resistors to VCC are included on-chip. Since all drivers invert, the pullup resistors force the outputs of unused drivers low. The input pullup resistors typically source 15µA; therefore, the driver inputs should be driven high or open circuited to minimize power-supply current in shutdown mode.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1mA, even if the transmitter output is backdriven between 0V and (VCC + 6V). Below -0.5V, the transmitter output is diode clamped to ground with a 1kΩ series impedance. The transmitter output is also zener clamped to approximately (VCC + 6V), with a 1kΩ series impedance.

MAX200–MAX209/MAX211/MAX213



Tekijä		teho	
Jaatinen Pekko		Sheet: 1/1	
1	2	3	4
5	6	5	6



Tekija
Jaatinen Pekko

hälyt'in

19.4.2010 12:02:41

Sheet: 1/1

```
/*  
  Nimi: Hälytysjärjestelmä  
  Tekijä: Pekko Jaatinen  
*/  
  
#ifndef F_CPU  
  #define F_CPU 3686400UL  
#endif  
  
#define BAUD 9600  
#include <avr/interrupt.h>  
#include <avr/io.h>  
#include <util/delay.h>  
#include <util/setbaud.h>  
#include <inttypes.h>  
  
// Lähtöjen määrittäminen  
#define PUN 5  
#define VIH 6  
#define SIR 7  
  
// Tulojenmäärittäminen  
#define AVAIN 0  
#define OVI1 6  
#define OVI2 1  
#define LIIKE 2  
#define VSTO 3  
#define SAUNA 4  
  
/*  
  Prototyyppien esittely  
*/  
void Init_USART(void);  
void USART_SendByte(uint8_t data);  
void text(void);  
void numero(void);  
void viesti(void);  
void ovi1(void);  
void ovi2(void);  
void liike(void);  
void vsto(void);  
void sauna(void);
```

```

/*
Pääohjelma
*/
int main(void)
{
    unsigned char KulunutAika = 0;
    int a; //apumuuttujat
    int b;
    int c;
    int d;
    int e;
    int f;
    int nappi;

    DDRB = 0x70; // 0b11100000
    PORTB = 0xFF; // 0b11111111

    DDRD = 0x00; //sisääntulo
    PORTD = 0xFC; //0b11111100

    TCCR1B |= ((1 << CS10) | (1 << CS11));

    a = 0;
    b = 0;
    c = 0;
    d = 0;
    e = 0;
    f = 0;
    nappi = 1;

    while(1)
    {

        if(bit_is_clear(PINB, AVAIN)) // tutkitaan onko hälytin kytketty päälle
        {
            if (TCNT1 >= 57599)
            {
                TCNT1 = 0; // Resetoidaan ajastimen arvo
                KulunutAika++;

                if (KulunutAika == 60) // Tarkistetaan onko 60s kulunut
                {
                    KulunutAika = 0; // Resetoidaan ajastimen muuttuja
                    PORTB &= ~(_BV(VIH)); //led pois päältä
                    PORTB |= _BV(PUN); //led päälle
                    a = 1;
                }
            }
        }
    }
}

```



```
}  
  
if (a==1)  
{  
  
    if(bit_is_clear(PIND, ovi1) && nappi == 1)  
    {  
  
        if (TCNT1 >= 57599)  
        {  
            TCNT1 = 0; // Resetoidaan ajastimen arvo  
            KulunutAika++;  
  
            if (KulunutAika == 60)    // Tarkistetaan onko 60s kulunut  
            {  
                KulunutAika = 0;    // Resetoidaan ajastimen muuttuja  
  
                if (b == 0)  
                {  
                    Init_USART();  
  
                    text();  
                    _delay_ms(200);  
  
                    viesti();  
                    _delay_ms(200);  
  
                    ovi1();  
                    _delay_ms(200);  
  
                    PORTB |= _BV(SIR);  
  
                    b = 1;  
                }  
            }  
        }  
    }  
  
    else if(bit_is_set(PINB, OVI2))    //Tutkitaan anturin 2 asento  
    {  
        if (c==0)  
        {
```

```
    Init_USART();

    text();
    _delay_ms(200);

    viesti();
    _delay_ms(200);

    ovi2();
    _delay_ms(200);

    PORTB |= _BV(SIR);

    c = 1;
    }
}

else if(bit_is_set(PINB, LIIKE)) //Tutkitaan anturin 3 asento
{
if (d==0)
    {
    Init_USART();

    text();
    _delay_ms(200);

    viesti();
    _delay_ms(200);

    ovi2();
    _delay_ms(200);

    PORTB |= _BV(SIR);

    d = 1;
    }
}

else if(bit_is_set(PINB, VSTO)) //Tutkitaan anturin 4 asento
{
if (e==0)
    {
    Init_USART();

    text();
```

```
        _delay_ms(200);

        viesti();
        _delay_ms(200);

        vsto();
        _delay_ms(200);

        PORTB |= _BV(SIR);

        e = 1;
    }
}

else if(bit_is_set(PINB, SAUNA)) //Tutkitaan anturin 5 asento
{
    if (f==0)
    {
        Init_USART();

        text();
        _delay_ms(200);

        viesti();
        _delay_ms(200);

        sauna();
        _delay_ms(200);

        PORTB |= _BV(SIR);

        f = 1;
    }
}

}

}

else
{

PORTB |= _BV(VIH); // led päälle
PORTB &= ~(_BV(PUN)); // led pois päältä
PORTB &= ~(_BV(SIR)); // sireeni pois päältä
```

```

a = 0;
b = 0;
c = 0;
d = 0;
e = 0;
f = 0;
}

}

return 0;
}

/*
Sarjaliikenteen alustusfunktio
*/
void Init_USART(void)
{
    UBRRH = UBRRH_VALUE;
    UBRRL = UBRRL_VALUE;
    UCSRA = 0x00;
    UCSRB |= (1 << TXEN); // Aktivoidaan lähetys
    UCSRC = 0x86; // 8 Data bitti, 1 Stop bittto, Ei pariteettia
}

/*
Sarjaliikenteen lähetysfunktio
*/
void USART_SendByte(uint8_t data)
{
    loop_until_bit_is_set(UCSRA, UDRE);
    UDR = data;
}

/*
Puhelin tekstimoodiin
*/
void text(void)
{
    unsigned char p;
    // text = { a, t, +, c, m, g, f, =, 1, enter }
    unsigned char text[10] = { 0x61, 0x74, 0x2B, 0x63, 0x6D, 0x67, 0x66, 0x3D,
    0x31, 0x0D
    };

    for(p = 0; p <= 9; p++)
    {

```

```

        USART_SendByte(text[p]);
        _delay_ms(4);
    }
}

/*
   Viestin numerotiedot
*/
void numero(void)
{
    unsigned char i;
    //viesti = { a, t, +, c, m, g, s, =, ", 0, 5, 0, x, x, x, x, x, x, x, x, ", enter }
    unsigned char viesti[21] = { 0x61, 0x74, 0x2B, 0x63, 0x6D, 0x67, 0x73, 0x3D,
    0x22,
    0x30, 0x35, 0x30, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x78, 0x0D};

    for(i = 0; i <= 20; i++)
    {

        USART_SendByte(viesti[i]);
        _delay_ms(4);
    }
}

/*
   Viestin sisältö
*/
void ovi1(void)
{
    unsigned char a;
    //sana = { O, v, i, enter, substitute }
    unsigned char sana[5] = { 0x4F, 0x76, 0x69, 0x0D, 0x1A
    };

    for(a = 0; a <= 4; a++)
    {
        USART_SendByte(sana[a]);
        _delay_ms(4);
    }
}

```

```

/*
  Viestin sisältö
*/
void ovi2(void)
{
  unsigned char a;
  //sana = { O, v, i, 2, enter, substitute }
  unsigned char sana[6] = { 0x4F, 0x76, 0x69, 0x32, 0x0D, 0x1A
};

for(a = 0; a <= 5; a++)
  {
    USART_SendByte(sana[a]);
    _delay_ms(4);
  }

}

/*
  Viestin sisältö
*/
void liike(void)
{
  unsigned char a;
  //sana = { L, i, i, k, e, t, u, n, ., enter, substitute }
  unsigned char sana[11] = { 0x4C, 0x69, 0x69, 0x6B, 0x65, 0x74, 0x75, 0x6E,
0x2E, 0x0D, 0x1A
};

for(a = 0; a <= 10; a++)
  {
    USART_SendByte(sana[a]);
    _delay_ms(4);
  }

}

/*
  Viestin sisältö
*/
void vsto(void)
{
  unsigned char a;
  //sana = { V, a, r, a, s, t, o, enter, substitute }
  unsigned char sana[9] = { 0x56, 0x61, 0x72, 0x61, 0x73, 0x74, 0x6F, 0x0D,
0x1A
};

```

```
for(a = 0; a <= 8; a++)
    {
        USART_SendByte(sana[a]);
        _delay_ms(4);
    }
}

/*
  Viestin sisältö
*/
void sauna(void)
{

    unsigned char a;
    //sana = { S, a, u, n, a, enter, substitute }
    unsigned char sana[7] = { 0x53, 0x61, 0x75, 0x6E, 0x61, 0x0D, 0x1A
};

    for(a = 0; a <= 6; a++)
        {
            USART_SendByte(sana[a]);
            _delay_ms(4);
        }

}
```