

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietokonetekniikka

Insinööri työ

Pete Mähönen

KIINTEISTÖN LÄMPÖTILAN VALVONTAJÄRJESTELMÄ

Työn valvoja

Yliopettaja Kai Poutanen

Työn teettäjä

Torpan kurssikeskus, Risto Itämetsä

Tampere 2006

Tekijä:	Pete Mähönen
Työn nimi:	Kiinteistön lämpötilan valvontajärjestelmä
Päivämäärä:	23.11.2006
Sivumäärä:	21 sivua ja 41 liitesivua
Hakusanat:	Lämpötilan valvonta, Etävalvonta, GSM-etävaroitus
Koulutusohjelma:	Tietotekniikka
Suuntautumisvaihtoehto:	Tietokonetekniikka

Työn valvoja: Kai Poutanen (yliopettaja)

Työn ohjaaja: Johtaja Risto Itämetsä, Torpan Kurssikeskus

Nykyaikaisessa kiinteistöhuollossa ja -hallinnassa on tärkeää voida valvoa erilaisia suureita hyvin tarkasti. Tällaisia suureita ovat esimerkiksi lämpötila ja kulunvalvonta. Lämpötilan tarkka säätö voi alentaa kiinteistön käyttämättömien tilojen lämmityskustannuksia suuresti. Henkilökunnan tehtävien monipuolistuessa ja työtaakan suuretessa on tarpeen myös havainnoida järjestelmän toimintaa etäkäytöllä. Vikatilanteessa on myös tärkeää saada tieto ajoissa vastuuhenkilölle, jotta ylimääräiset vahingot saadaan ainakin rajoitettua ellei estettyä kokonaan.

Käytännönläheisissä elektroniikkasuunnitteluprojekteissa kuvataan laitteiston suunnitteluprosessi vaiheittain aina kytkentäkaavion laatimisesta piirilevy-suunnitteluun ja siitä lopulliseen rakentamiseen asti. Suunnittelussa painotetaan laitteiston edullista hankintahintaa, luotettavuutta sekä modulaarisuutta laitteistossa. Yleensä tiloissa, joihin laitteistoa päivitetään, käytetään valmiina olevaa infrastruktuuria hyväksi niin paljon kuin mahdollista, kuten esimerkiksi yleiskaapelointeja.

Hyväksi todetut, yleisesti käytössä olevat tavat ohjaavat suunnitteluprosesseja ja luovat niille suuripiirteisiä toimintatapoja. Kokemus ja suunnittelijan tietämys heijastuu niin materiaalivalinnoissa kuin komponenttivalinnoissakin sekä itse laitteen rakenteen suunnittelussa ja kokoamisessa. Laadukkaiden komponenttien valinta helpottaa myös huoltohenkilöstön toimintaa toimintavarmuutensa ansiosta. Modulaariseksi rakennettu järjestelmä helpottaa jatkokehitysmahdollisuuksia ja mahdollistaa peruslaitteiston käytön osana suurempaa järjestelmää pienien muutostöiden avulla.

Työvälineinä tässä projektissa käytettiin Internetistä saatavia ilmaisversioita ohjelmistoista kuten Eagle-elektroniikan suunnitteluohjelmisto ja Codevision-ohjelmointiympäristö.

Author: Pete Mähönen
Title: Temperature monitoring system Computer Engineering
Date: Product Development
Number of pages: 21 pages, 41 appendix pages
Keywords: Temperature, Remote monitoring, Remote alarming
Program: Computer Systems Engineering
Specialisation: Computer Engineering

Thesis Supervisor: Kai Poutanen (Senior Lecturer)

Instructor: Manager Risto Itämetsä, Torpan Kurssikeskus

In modern accommodation service companies, energy efficiency is important. So it is important to develop systems to control the temperature of the used and unused rooms periodically to lower the heating expenses as low as possible. It is also important to be able to monitor and control temperature in one location. In the case of the malfunction of the heating system, it is vital to get the information to the maintenance personnel as soon as possible to prevent further damage. This can be done, for example by using the public GSM short messaging system, sending a text message automatically or by making an automated call to the responsible personnel.

In modern product development projects the main idea is to develop a reliable low-cost and easy-to-use product. Currently the product was temperature monitoring system for smaller accommodation companies such as hostels. The development methods were mainly to use simple, much used in industry, reliable and low-cost components and utilize as much as possible the infrastructure already existing in the planned location. In this case, the basic telecommunication-cabling was used. This design was built to be modular, so that later it can be developed further and can be used as a component in larger systems.

ALKUSANAT

Tämä insinööri työ on kirjoitettu Tampereen ammattikorkeakoulussa insinööri työksi, valvojana toimi Kai Poutanen. Tosi elämän tarpeista lähteneen täysimittaisen suunnitteluprosessin haasteet ja mahdollisuudet ovat olleet todella opettavaisia. Järjestelmän suunnittelu on opettanut käytännön elektroniikkasuunnittelua, loogista prosessin etenemistä ja aikataulun suunnittelua kouriintuntuvalla tavalla. Erittäin tärkeässä osassa ovat olleet myös insinöörille tärkeät varmuuskopiot, joita on sovellettu useaan kertaan tämän insinööri työn valmistumisen aikana.

Haluan kiittää kaikkia tukijoitani, erityisesti vaimoani Niinaa ja Torpan kurssikeskuksen johtajaa Risto Itämetsää heidän tuestaan niin opiskelun kuin työharjoittelunkin aikana.

Tampereella 23. marraskuuta 2006

Pete Mähönen

KÄYTETYT TERMIT JA LYHENTEET

RISC	Reduced Instruction Set Computer, rajatulla käskykannalla oleva tietokoneprosessoriarkkitehtuuri
Atmel	Prossessorivalmistaja
AVR	Advanced Virtual RISC, 8- tai 16-bittinen Atmelin valmistama mikrokontrolleriperhe
Atmega	Atmelin valmistama AVR-tuoteperheen prosessiperhe
DallasSemi	Dallas Semiconductors, mikropiirien valmistaja
TTL	Transistor-Transistor-Logic, mikropiirityyppi
I/O	Input/Output, sisäänmeno/ulostuloliitännät, esimerkiksi mikroprosessorissa
RS-232	Yleisesti käytössä oleva sarjaliikenneväylän standardi
C-kieli	Korkean tason kieliksi luokiteltava ohjelmointikieli
ISO C99	C-kielen standardointi vuodelta 1999
Modulaarisuus	Isomman kokonaisuuden jako pienemmiksi osakokonaisuuksiksi ja niiden ongelmien ratkaisu
ISP	In-System-Programming, ohjelmointi joka tapahtuu mikropiirin ollessa laitteessa kiinni
STK-500	Atmelin valmistama AVR-kehitysalusta
Codevision	AVR-ohjelmointiympäristö Windows-käyttöjärjestelmälle
GSM	Groupe Spécial Mobile tai Global System for Mobile Communications, matkapuhelinverkkotyyppi
Nokia 6130	Nokia Oyj:n valmistama GSM-standardoitu matkapuhelin
UPS	Universal Power Source, yleiskäyttöinen varavirtalähde

SISÄLLYSLUETTELO

TIIVISTELMÄ.....	I
ABSTRACT OF THE THESIS	II
ALKUSANAT.....	III
KÄYTETYT TERMIT JA LYHENTEET.....	IV
SISÄLLYSLUETTELO.....	V
1 JOHDANTO.....	1
2 TOIMINNALLINEN MÄÄRITTELY.....	2
2.1 Lähtökohdat	2
2.2 Toimintaperiaate.....	2
2.3 Elektroniikka	3
2.4 Ohjelmisto.....	4
3 TOTEUTUS JA KOMPONENTIT.....	5
3.1 Atmel Atmega 16.....	5
3.2 Dallas Semiconductor DS18S20.....	6
3.3 Hitachi HD44780.....	7
3.4 Dallas Semiconductor MAX233CPP.....	8
3.5 Hälytysmekanismi ja GSM-rajapinta.....	8
3.6 Jännitteensyöttö.....	9
3.7 Kotelointi.....	10
3.8 Piirilevyn suunnittelu ja kytkentä.....	11
3.9 Ohjelmisto.....	16
4 ASENNUS JA TESTAUS.....	17
4.1 Pääyksikkö.....	17
4.2 Lämpötilanmittausyksikkö.....	17
4.3 Siirtolinjat.....	17
4.4 Käyttötestit.....	18
5 YHTEENVETO.....	19
LÄHTEET	
LIITTEET	

1 JOHDANTO

Merkittävä osa nykyaikaisten kiinteistöjen lämpötilan hallinnasta on toteutettu tietotekniikan keinoin. Tästä alkoi innostus suunnitella riittävän kattava, luotettava ja helppokäyttöinen järjestelmä mahdollisimman edullisesti kiinteistöihin, joissa sellaista ei ole.

Torpan kurssikeskus on yksi esimerkki tällaisesta kiinteistöstä. Kurssikeskuksen rakennukset on rakennettu 70-luvulla ja ne vaativat täsmällistä valvontaa. Kurssikeskus käsittää seitsemän taloa, joista kuuden tarkkailu on tärkeää kustannusten vuoksi. Valvottaessa lämpötilaa reaaliaikaisesti liiallista lämpenemistä tai jäähtymistä ei pääse tapahtumaan. Jatkuvan seurannan ansiosta lämmitystä voidaan säätää alaspäin niiden rakennusten osalta, joissa ei ole toimintaa. Näin saadaan aikaan huomattavia kustannussäästöjä lämmityksen osalta.

Tämä insinööri työ on suunniteltu ja toteutettu edellä mainittuun esimerkikiinteistöön kurssikeskuksen johtajan pyynnöstä. Valmista tuotetta voidaan myöhemmässä vaiheessa soveltaa esimerkiksi lämpötilan säätelyyn tai kulunvalvontaan hyvin pienillä muokkaustöillä.

Tässä insinööri työssä toteutetaan laitesuunnitteluprojekti täydellisenä komponenttien valinnasta lähtien lopulliseen asennukseen ja testauksiin. Tilattu laite testataan modulaarisesti suunnittelun aikana ja koekäytetään valmiiksi asennettuna kiinteistöön. Suunnitteluprosessi, testaus ja asennus kuvataan yksityiskohtaisesti luvuissa yksi, kaksi ja kolme.

2 TOIMINNALLINEN MÄÄRITTELY

2.1 Lähtökohdat

Kuten aiemmin mainittiin, puhelinjärjestelmä Torpalla oli jo olemassa, joten laitteistoksi tarvitaan valvonta-anturipaketti jokaiseen rakennukseen, valvontapäätelaite näkyvälle paikalle päärakennukseen ja mahdollisesti väyläsovittimet puhelinlinjojen hyödyntämiseen, jos se on tarpeellista. Ensimmäisessä testausvaiheessa peruslaitteisto rakennetaan ja kytketään toimintaan, toisessa vaiheessa lisätään anturit ja GSM-varmennus.

Rakennuksissa lämpötilaa säätelee Honeywellin Aquatrol 200 -kiertovesipumppujärjestelmä. Tarpeen mukaan myöhempänä ajankohtana lämpötilanvalvontalaitteiston anturipäähän voidaan liittää lisäprosessori käyttämään lämpötila-anturia ja säätämään lämpötilaa. Honeywellin alkuperäinen vedenkierron rajoitin, joka oli potentiometri, korvataan digitaalisella potentiometrillä, jota voidaan ohjata esimerkiksi Atmelin atTiny-prosessorilla.

2.2 Toimintaperiaate

Lämpötilanmittauksen periaatteet

Lämpötilan perusyksikkö on yksi Kelvin, josta on johdettu yleisesti käytetyt celsiusasteet ja fahrenheitasteet. Elektroninen lämpötilanmittaus perustuu lämpösähköiseen eli termosähköiseen ilmiöön. Tätä ilmiötä kutsutaan myös Seebeckin ilmiöksi.

Lämpötilanmittauskytkentä perustuu kahden eri materiaalista valmistetun ”metallilangan” välisen potentiaalieron havainnointiin. Lämpötilan muuttuessa materiaalien välille muodostuu potentiaaliero, joka on suoraan verrannollinen lämpötilan muutokseen. Kokonaisuutta, johon on liitetty nämä lämpöä mittaavat materiaalipalat, kutsutaan termoelementiksi. Termoelementti voi olla esimerkiksi rauta-konstantaani-pari. Termoelementin tuottama potentiaaliero on jännitearvoltaan muutaman mikrovoltin luokkaa, kun lämpötila muuttuu yhdellä Kelvinillä.

On olemassa myös metallista valmistettuja lämpötila-antureita. Näiden antureiden lämmönmittausperiaate perustuu tietyn metallin resistanssin muutokseen lämpötilan muuttuessa. Metalleja, joita on käytetty vuosien kuluessa antureihin, ovat esimerkiksi platina, nikkeli ja kupari. Yleisimpiä metalliantureita ovat malliltaan PT-100-platinapohjaiset anturit.

Puolijohdeanturi-tyyppiset lämpötilamittaimet ovat hyvin herkkiä lämpötilan vaihteluille. Termistoriksi kutsutun anturin vastusarvon eli resistanssin muutos voi olla yhden Kelvinin lämpötilanmuutoksen tapahtuessa useita prosentteja vastusarvon maksimista. Termistoreja on olemassa negatiivisella ja positiivisella lämpötilakertoimella varustettuna, jolloin niiden nimitykset ovat NTC ja PTC. Termistorien mitta-alue on hyvin laaja, -60 celsius-asteesta +300 celsiusasteeseen. Termistorien käyttöä hankaloittaa se, että ne joudutaan kalibroimaan tietyin väliajoin anturi kerrallaan.

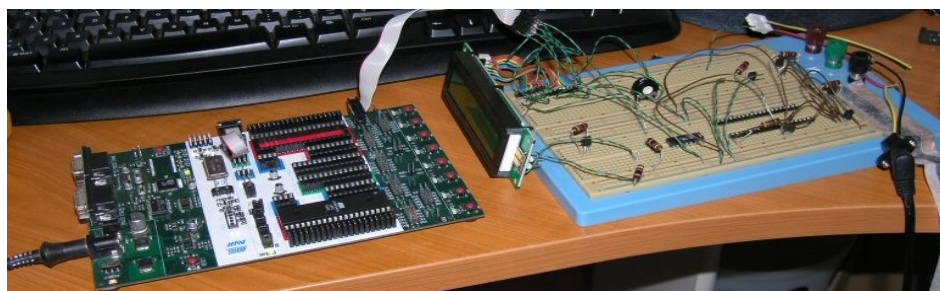
Tässä insinööriyössä käytetyn lämpötilamittainpaketin tyyppi on metallipohjainen. Laitteen tulee kyetä mittaamaan kuudesta erillisestä kohteesta lämpötilat ja valvomaan niiden pysymistä annetuissa rajoissa. Laitteen tulee päivittää myös lukemat reaaliaikaisesti LCD-näytölleen. Laitteen tulee sisältää myös käyttökytkimet hälytyksen poiskytkemistä ja rajojen asettamista varten. Tilaaja esitti toivomuksen, että rajojen ylityessä järjestelmä ilmoittaa GSM-rajapinnan yli vastuuhenkilölle virhetilanteesta sekä paikallisesti pietsosummerilla ja led-valolla.

2.3 Elektroniikka

Projektin aluksi laitteiston perustaksi oli ehdolla kaksi eri kontrolleria: Atmelin Atmega16 ja PIC16F84. Projektiin valittiin kuitenkin Atmega16, koska sitä oli jo aikaisemmin sovellettu samantyyppisiin projekteihin. Tälle prosessorille oli myös valmis kehitysympäristö lainattavissa Tampereen sähköopiskelijoilta, Taso Ry:ltä. Atmega16 oli myös helppokäyttöisempi runsaampien I/O-porttiansa takia.

Näin ollen tuotteen tulee perustua Atmel Atmega16:n kahdeksanbittiseen RISC-tyyppiseen mikrokontrolleriin. Lämpötilamittainelementteinä käytetään kuutta Dallas Semiconductor DS18S20-lämpötilamittainpakettia. Datalinjoina rakennuksiin käytetään valmiiksi vedettyjä puhelinjärjestelmän parikaapeloituja kuparijohtimia. Parikaapeleita on vedetty kolme jokaiseen majoitusrakennukseen. LCD-moduuli sekä muut tarpeelliset komponentit ja toimintalogiikka on suunnittelijan valittavissa parhaiten projektin tarpeisiin soveltuvilla ratkaisuilla.

Etähälytysjärjestelmän välityslinkkinä käytetään Nokian 6130-GSM-puhelinta ja sen viestiverkon lyhytviestiominaisuutta.



Kuva 1 Prototyypin koekytkentä

2.4 Ohjelmisto

Ohjelmointiympäristöä ja ohjelmointikieltä ei ole erikseen määrätty, joten tässä projektissa valitaan tarkoitukseen sopivin ympäristö. Peruseriaatteen mukaisesti taloudellisuus on suurin ohjenuora, joten ohjelmistoja ei ole tarkoitus ostaa, vaan käyttää ilmaisia ohjelmistoja.

Ohjelmistoista oli myös ehdolla käytettäväksi kahta eri ohjelmointiympäristöä. Codevision, jota saa Internetistä ilmaiseksi opiskelukäyttöön, oli suunnittelijan mielestä helppokäyttöisempi kuin STK-500-kehityspaketin mukana tullut AVR-Studio 4. AVR-studio 4 oli maksullinen ohjelmisto, joten se oli yksi syistä, miksi kyseessä oleva ohjelmisto ei tullut valituksi projektiin.

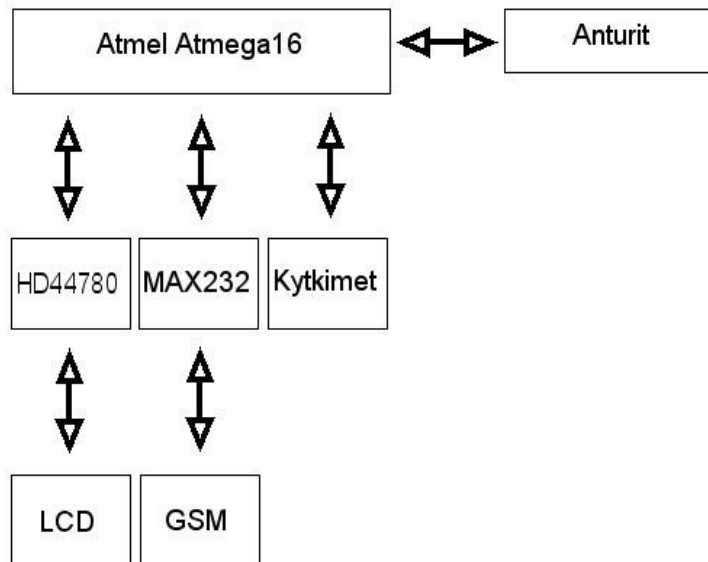
Mikrokontrollerin ohjelmointi tehdään vertailutestin voittanutta Codevision-ohjelmointiympäristöä käyttäen. Ohjelmointikielenä pääasiallisesti C-kieli, sen ISO-standardoitu versio C99. Tarpeen vaatiessa muokataan myös assembler-koodia. Codevision tekee muunnoksen assembleriin ja siitä kääntämisen konekielille automaattisesti. Ohjelmiston toimintalogiikka ja ohjelmointi kuvataan tarkemmin luvussa kolme.



Kuva 2 Erilaisia lämpötila-antureita eli termistoreita /6/

3 TOTEUTUS JA KOMPONENTIT

Toteutuksessa on syytä ottaa huomioon rajoitukset tiedonsiirtolinjojen määrässä. Muita rajoituksia ei oikeastaan ole. Suunnittelussa painotetaan yksinkertaisuutta, luotettavuutta, helppokäyttöisyyttä ja taloudellisuutta.



Kuva 3 Valvontajärjestelmän lohkokaaavio

3.1 Atmel Atmega 16

Kuten kappaleessa 2.3 todettiin, laitteen perustaksi valittiin Atmel Atmega 16, joka on 8-bittinen risc-prosessori. Valintaperusteina käytettiin helppoa saatavuutta, laajoja käyttömahdollisuuksia ja kohtuullisen edullista hintaa. Lisäksi Atmega 16 on ominaisuuksiltaan riittävän hyvä ja energiankulutukseltaan kohtuullinen. Järjestelmä rakennetaan siten, että sitä voidaan ajaa myös sähkökatkon sattuessa akuilla. Jatkuva jännite tuodaan järjestelmään UPSin läpi, joten erilliselektronikan tarvetta laitteistoon ei ole. Esimerkiksi 12 voltin, seitsemän ampeeritunnin lyijyhyytelökalla laitteisto toimii useita vuorokausia.

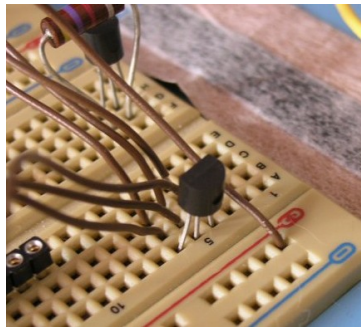
I/O-liitäntöjä on riittävästi tämän projektin tarpeisiin. Tämän lisäksi yhtä porttia voidaan käyttää suorana RX/TX-liitäntänä, jolla saadaan käyttöön In-System-Programming-ominaisuus. Samalla liitännällä kytketään prosessori GSM-puhelimeen, jotta saadaan viestintärajapinta käyttöön. Piirilevynsuunnittelua ajatellen prototyyppi on helppo rakentaa, koska projektiin valittua prosessorityyppiä saadaan normaalissa DIL-kotelossa.

Ensimmäinen ohjelmointi tehtiin Tampereen Sähköpiskelijät TASO Ry:n lainaamalla Atmelin System-Toolkitillä mallia 500 (STK500).

3.2 Dallas Semiconductor DS18S20

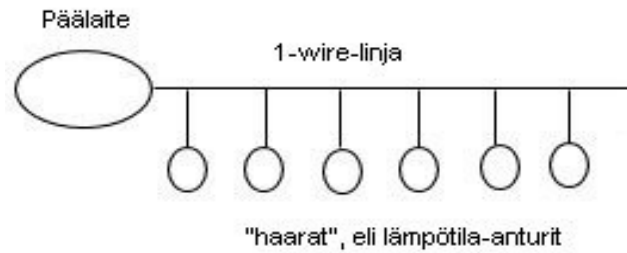
Lämpötila-anturiksi oli ehdolla Dallas Semiconductorin valmistamat DS1631 sekä DS18S20. DS1631:tä olisi ollut saatavilla edullisesti, kuten myös toista vaihtoehtoa. Piirit ovat toiminnaltaan samansuuntaisia, ainoastaan liikennöinnissä on eroa. DS18S20 kykenee one-wire-liikennöintiin, kun taas DS1631 ei. Jos anturiksi olisi valittu DS1631, olisi jouduttu liikennöintijohtimien määrää lisäämään ja se olisi tuonut lisäkustannuksia projektiin. DS1631 on myös poistuva malli valmistajan luettelossa, joten olisi ollut erittäin hankalaa saada varaosia järjestelmään jos joku antureista sattuisi rikkoutumaan. Näin ollen valittiin DIL-koteloinen DS1820.

DS18S20 on digitaalinen lämpötilamittainpiiri. Kyseessä oleva piiri valittiin tähän projektiin, koska sitä oli saatavilla halpaan hintaan. Tätä piiriä on käytetty runsaasti erilaisissa sovelluksissa, joten materiaaleja ja esimerkkejä käytöstä ja käytännön ongelmista on runsaasti. Piiriä on helppo soveltaa erilaisiin tarkoituksiin ja sen mittaustulokset ovat tarpeeksi tarkkoja. Sen käyttölämpötila-alue on -10 celsiusasteesta $+85$ celsiusasteeseen ja tarkkuus $\pm 0,5$ astetta ovat sopivat tähän järjestelmään. Tämän lisäksi se voi ottaa käyttöjännitteensä datalinjasta, joten se ei välttämättä tarvitse erillistä jännitelähdettä. Parhaan liikennöinnin varmistamiseksi on kuitenkin järkevää sijoittaa jokaiselle anturille oma jännitelähteensä ja jättää datalinjaan ainoastaan dataliikenne.



Kuva 4 DS18S20 koekytkenässä

Kytchentätopologia, johon päädyttiin järjestelmän kytkentätapaa valittaessa, on valmistajan standardoinnin mukainen haaroitettu väylä. Toisessa vaihtoehdossa, tähtitopologiassa, impedanssien epäsopivuus siirtolinjojen pituuserojen vuoksi aiheuttaisi suuria häiriöitä liikennöintiin. Kytchentätapaa valittaessa ajateltiin myös käytäntöä; valittu kytkentätapa helpottaa myös laitteiston asennusta pyydettyyn kohteeseen.



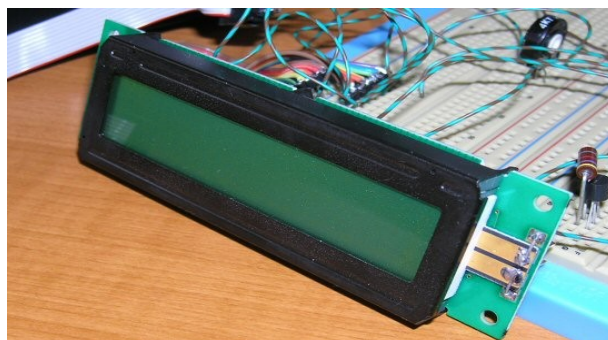
Kuva 5 Haaroitettu väylä

Master-laite eli pääyksikkö sijaitsee A-talossa ja väylän pää on A-talon lämpötila-anturi. Haaroitukset tapahtuvat puhelinjakamossa, jossa sijaitsevat kaikki tarpeelliset johdotukset muihin rakennuksiin meneville antureille.

Samaa datalinjaa, jolla DS18S20 liikennöi, voidaan käyttää tulevaisuudessa muuhunkin dataliikenteeseen.

3.3 Hitachi HD44780

LCD-moduulina käytettiin valmista Displaytekin moduulia, jonka tähän työhön tarjosi jäsenetuna Tampereen Sähköpiskelijät ry. LCD-moduulissa tyypiltään GB1494V-0 S on ohjaimena Hitachin HD44780A-mikrokontrolleri. HD44780A on yleisesti käytössä oleva LCD-ohjainpiiri, jota on ominaisuuksiensa puolesta helppo hyödyntää sovelletuissa kytkennöissä, tässä tapauksessa Atmega16:n yhteydessä. Tätä ohjainpiiriä on myös erittäin helppo ohjata, mikä puolestaan helpottaa myös firmwaren ohjelmointia. HD44780A:lle on olemassa valmis ohjelmistokirjasto esimerkiksi Codevision-AVR-ohjelmointiympäristössä.



Kuva 6 LCD-moduuli GB-1494V-0 S

3.4 Dallas Semiconductor MAX233CPP

Sarjaliikennepiirinä käytettiin perussarjaliikennepiiriä MAX233CPP. Tässä kytkennässä ei olisi ollut välttämätöntä tehdä RS232-tasomuunnosta, kuitenkin tulevaisuuden kannalta se kannattaa tehdä. Esimerkiksi, jos piiriä tarvitsee ohjelmoida uudelleen paikalleen kytkettynä, on se tällä tavalla mahdollista. Tähän voidaan käyttää ISP-liitäntää 9-pin-liittimen kautta. RS232-muunnettu sarjamuotoinen liikenne tarjoaa myös mahdollisuuden käyttää mitä tahansa AT-komentoja ymmärtävää GSM-puhelinta, jos nykyinen datakaapeli tai puhelin sattuisi esimerkiksi rikkoutumaan.

3.5 Hälytysmekanismi ja GSM-rajapinta

Laite ilmoittaa virhetilanteesta pietso-summerin ja led-valon avulla. Paikallinen varoituskonetti on mahdollista kytkeä pois päältä korjaustoimenpiteiden ajaksi. Lisäksi laite ilmoittaa vastuuhenkilön päivystyspuhelimeen tekstiviestillä virhetilanteesta. GSM-rajapintana käytettiin Nokian 6310-GSM-puhelinta, jonka valintaperusteina olivat edullisuus ja puhelintekniikan yhteensopivuus nykyisten puhelinverkkojärjestelmien kanssa.

Lisäksi tärkeä kriteeri olivat puhelimen tukemat standardit AT-komennot, joilla tekstiviesti voidaan lähettää yksinkertaista käskyä käyttäen. Yhteystapana käytettiin tarvikedatakaapelia, Nokian lisensoimaa tyyppiä DLR-3P. Puhelimen tarkka tuoteseloste sekä AT-komennot ja niiden syntaksit selvitettiin Nokian Euroopan-palvelimelta, 6310-tukisivustolta. /7/

Puhelimen virtanäppäimeen juotettiin johtimet ja käynnistymistä ja sammumista ohjataan prosessorilla. Puhelimen virtakytkin on maadoittava, joten tähän tarkoitukseen sopii viiden voltin optoerotin. Käytetyn erottimen valmistaja oli Avago Technologies ja malli oli HCPL-2611. Näin saadaan puhelimen akunkäyttö taloudellisemmaksi.

Optoerotin tilan vaihtava sisäänmenovirta on datalehden mukaan 5 mA. /8/ $R = U/I$, joten vähentämällä syöttäjännitteestä ledin kynnysjännitteen 1,7 voltia saadaan 3,3 voltia vastukseen hukattavaksi jännitteeksi. Tästä saadaan laskettua 3,4 voltia / 0,005 ampeeria, ja vastaukseksi vastuksen arvo 680 Ohmia.

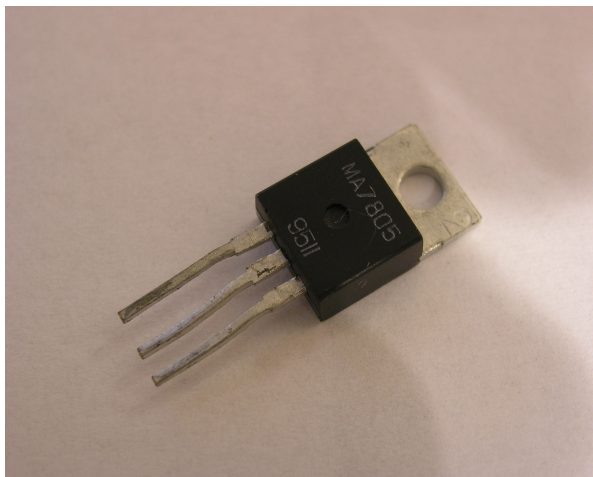


Kuva 7 Nokian 6310-mallinen GSM-puhelin ja DLR-3P-datakaapeli

3.6 Jännitteensyöttö

Jännitteensyöttö toteutetaan tasajännitettä antavalla verkkolaitteella verkkovirrasta. Vaikka kohteessa on mahdollista käyttää suoraan verkkovirrasta tulevaa jännitettä, on kuitenkin järkevintä asentaa varavirtalähde laitteiston ja yleisen sähköverkon väliin. Näin saadaan aikaan jatkuva toiminta myös sähkökatkon aikana. Torpalla on jo olemassa UPS, jota käytetään muun muassa puhelinjärjestelmän varmistamiseen. UPSissa on kuitenkin liitäntäpaikkoja enemmän kuin normaalin toiminnan aikana on käytössä, joten sitä voidaan hyödyntää tässä projektissa.

Jänniteregulaattorina, niin anturissa kuin keskusyksikössäkkin, käytetään LM7805-regulaattoria. Kyseessä oleva regulaattori hyväksyy datalehden mukaan +7,5 voltin - +20 voltin tasajännitteen syöttöjännitteensä. +5 voltin syöttöjännitteellä regulaattorin ulostulojännite on yleismittarilla mitattuna 3,75 voltia, joka ei riitä laitteiston käyttöön. Tätä regulaattoria käyttämällä syöttöjännitteen antajana voidaan käyttää erilaisia käytöstä poistettujen laitteiden, esimerkiksi akkuporakoneiden verkkolaitteita. Tällä saadaan alennettua kokoamiskustannuksia. Jänniteregulaattoreihin asennettiin eriste ja kuparinen jäädytinripa.



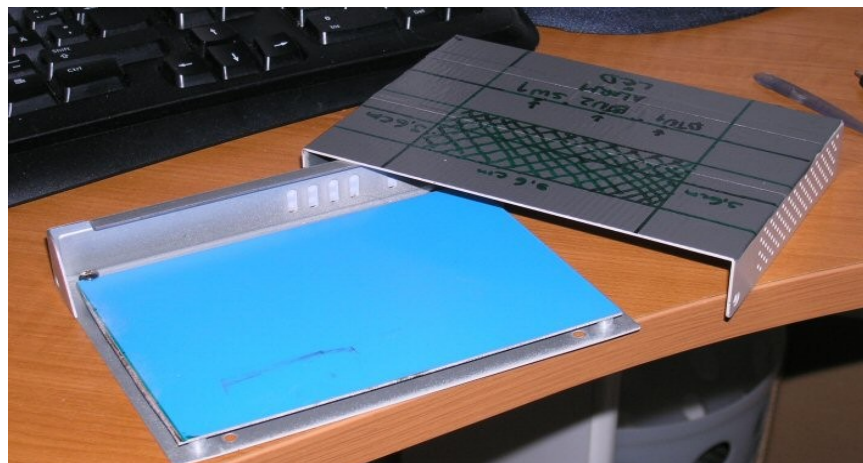
Kuva 8 Jänniteregulaattori LM7805.

Kontrollilaitteen päässä ja antureiden sijoituspaikassa on mahdollista käyttää erillisiä muuntajia. DS18S20 ei tarvitse erillistä jännitelähdettä, vaan se voi ottaa käyttöjännitteensä suoraan datalinjasta. Tässä ratkaisussa päädyttiin kuitenkin erilliseen jännitteensyöttöelektronikkaan jokaisessa anturissa siksi, että tulevaisuuden edelleenkehitysmahdollisuudet lämpötilasäädön lisäämiseksi olisivat helpompia.

3.7 Kotelointi

Koteloksi valittiin kierrätysmetallilaatikko, joka on ulkomitoiltaan 120mm x 160 mm x 27 mm. Kotelossa oli valmiina kierteet kiinnitysruuveille sekä kotelon ulkopinnassa jäähdytysrei'istö. Valmis yksikkö sijoitetaan tilaan, jossa on paljon sähköisiä laitteita, joten EMC-suojaus oli yksi valintaan vaikuttaneista perusteista. Kotelon kanteen jyrtsittiin täsmällinen aukko ja valmistettiin kiinnitykset LCD-piirilevylle sekä käyttökytkimille. Koteloon tehtiin läpiviennit myös jännitteensyötön, GSM-rajapinnan sekä antureiden liittimille.

LCD-moduulin luettavuuden ja sijoituksen vuoksi kotelon orientaatio muodostui vaakatasoon. Kotelon koko huomioon otettuna piirilevyn suunnittelun perustaksi otettiin 100 mm x 160 mm:n kokoinen yksipuolinen peruspiirilevy, joka saatiin kiinnitettyä kotelon omilla ruuveilla. Jyrtsinnän jälkeen kotelo pinnoitettiin mustaksi ja lakattiin kiiltäväksi. Anturipäässä mittainkomponentit kotelointiin kooltaan 35 mm x 50 mm x 15 mm:n muovikoteloilla, joihin tehtiin läpivienti jännitteensyöttöä ja dataliikennettä varten. Mahdollisimman tarkan mittaustuloksen saamiseksi lämpötila-anturi kiinnitettiin alumiiniseen jäähdytysprofiiliin. Se tasaa nopeita lämpötilanvaihteluita mittaustuloksissa.



Kuva 9 Mitoitettu metallikotelo piirilevy koteloon sovitettuna

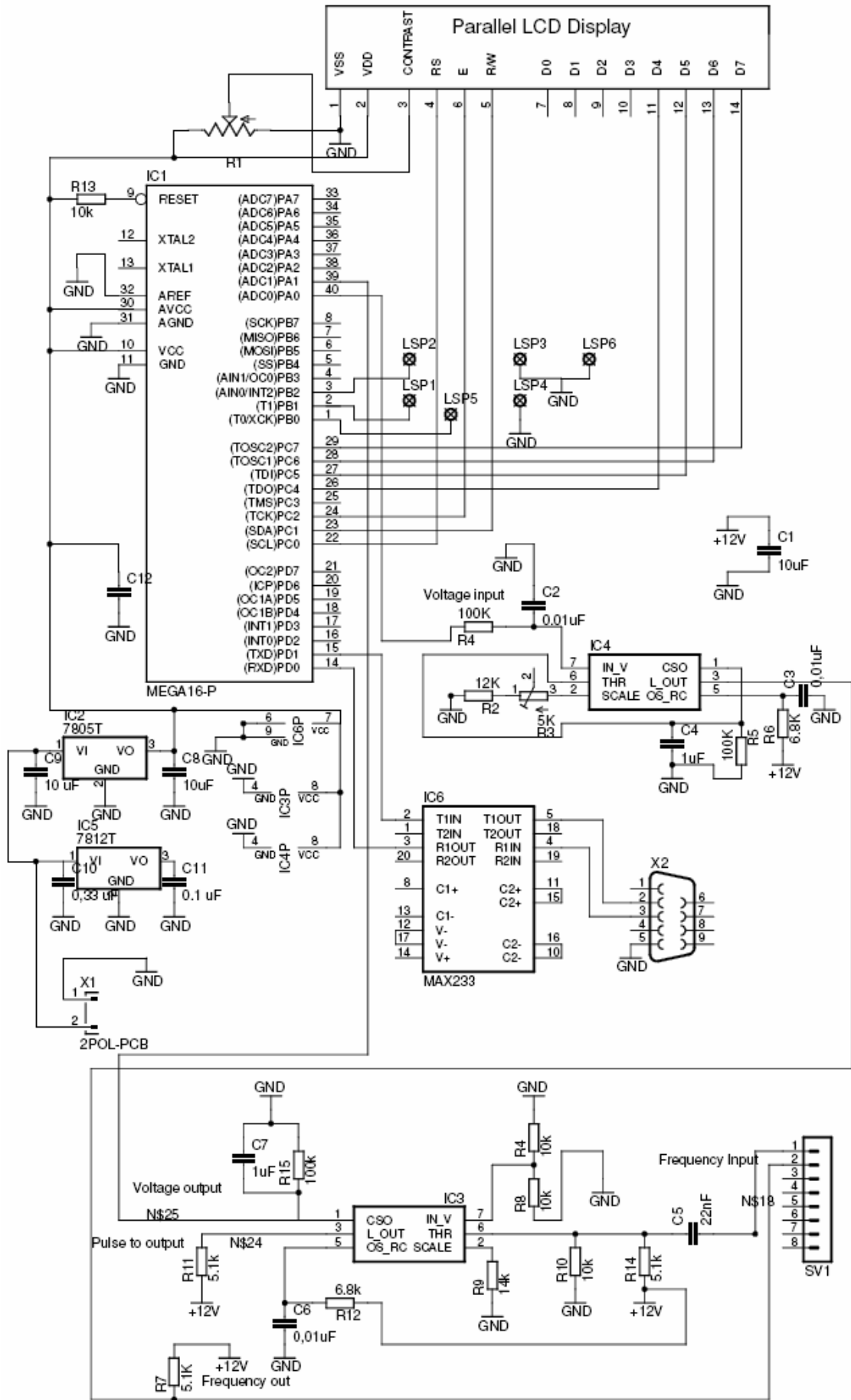
3.8 Piirilevyn suunnittelu ja kytkentä

Kuten useissa vaiheissa tässä projektissa, tämänkin vaiheen työkalut olivat vertailussa ennen suunnittelun varsinaista aloittamista. Vaihtoehtoina olivat Cadsoftin Eagle ja Labcenter Electronicsin Proteus. Molemmat ohjelmistot ovat monipuolisia piirilevysuunnittelu ympäristöjä, kumpikin omalla tavallaan toteutettuna. Suunnittelija päätyi kuitenkin Eagleen sen sujuvamman käytön takia. Kummastakin ohjelmistosta oli saatavissa ilmainen testiversio, ja niiden lisenssien nojalla voidaan suorittaa tämän projektin piirilevyjen suunnittelu.

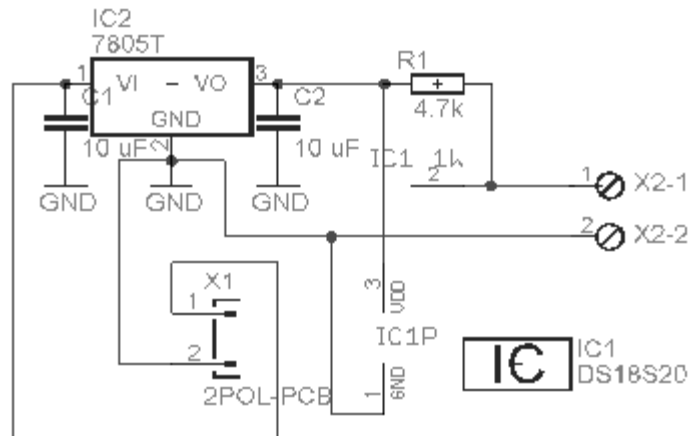
Lopullinen piirilevysuunnittelu tehtiin siis Eagle-piirilevysuunnittelu-ohjelmiston Lite-versiolla, jonka sai ilmaiseksi käyttöön Internetistä. /13/ Suunnittelussa noudatettiin yleisiä piirilevysuunnittelun periaatteita, ladottiin jännitelähteet piirilevyn yhteen reunaan sekä pidettiin erillään analogisten ja digitaalisten komponenttien kokonaisuudet.

Suunnitteluprosessissa tehtiin ensin yksi versio kytkentäkaaviosta, sitten testattiin kytkentää, ja tämän jälkeen tehtiin toinen kytkentäkaavio. Toisesta versiosta kytkentäkaaviota on jätetty osia pois, kuten erilliset taajuus- jännitemuuntimet ja 12 voltin jänniteregulaattori tarpeettomina ja kustannussäästön vuoksi.

LSP-merkinnällä kuvataan kytkentäkaaviossa erillisten ilmajohtimien juotosnastoja ja optoerottimen liitospinnejä. Muutoksen tehtiin siksi, että lopullinen kytkeminen ja komponenttien sijoittelu olisivat helpompia tehdä.



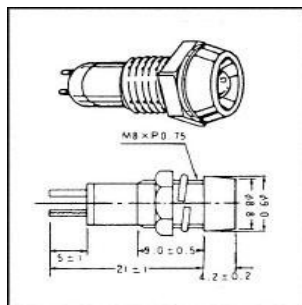
Kuva 10 Kytentäkaaviokuva 1



Kuva 11 Anturipään kytkentäkaaviokuva

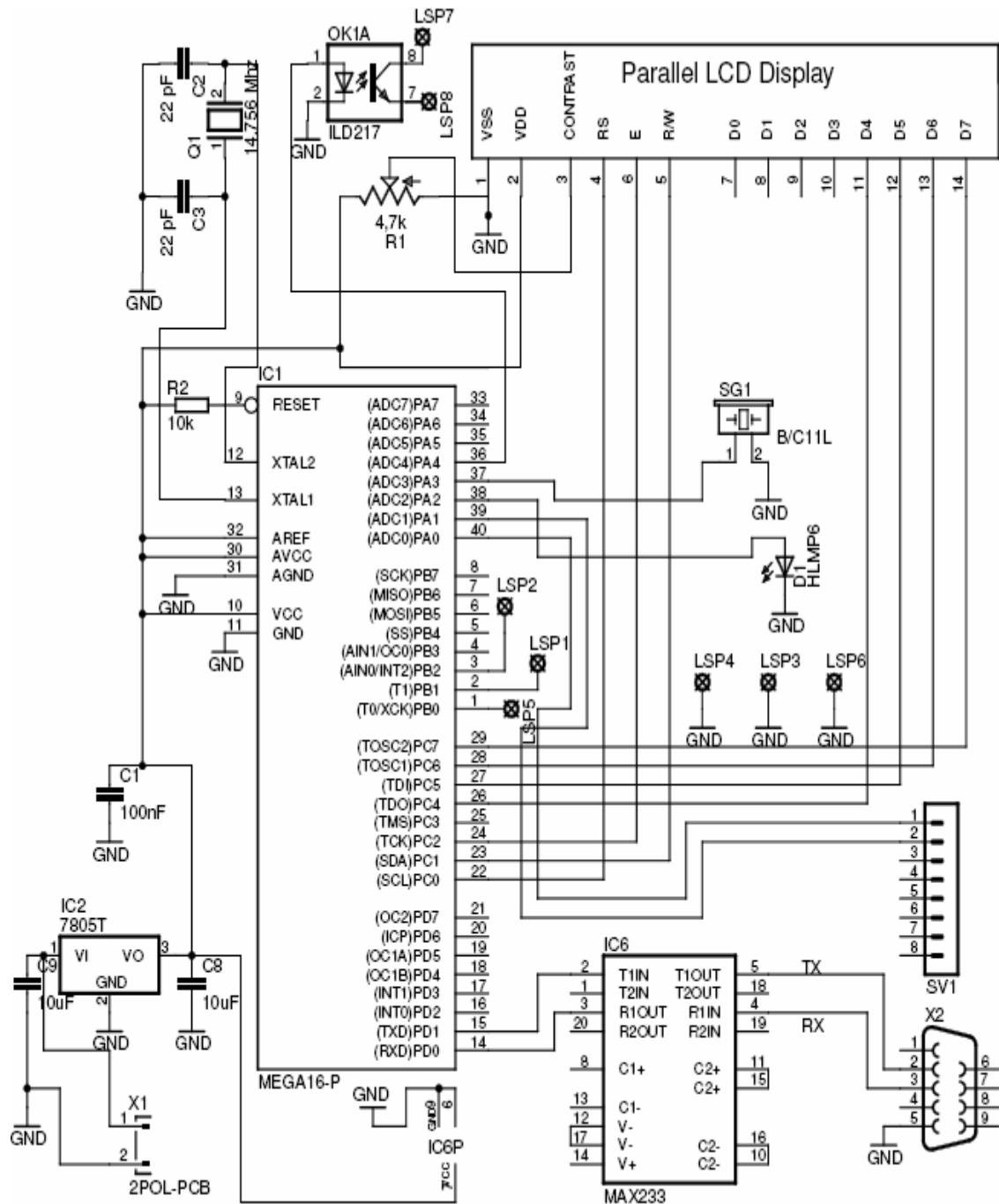
Kytkentään kaksi on myös lisätty pietsosummeri sekä led-valo ja yksi kontrollipainike. Kontrollipainikkeina käytetään värähtelemättömiä kytkimiä ohjelmiston suunnittelemisen helpottamiseksi. Kytkin jota käytettiin oli Multimec-merkinen ja tyypiltään RA3ETL9104 /12/, Farnellin tilausnumerolla 1132883. Tämän vuoksi voidaan hyödyntää suoraan kytkimen tilan muutosta, eikä tarvitse lisätä laskureita kytkinvärähtelyiden kompensoimiseksi.

Pietsosummerina käytettiin Farnellin tilausnumerolla 1193319, valmistajan osanumerolla MT-510 löytyvää, 3-20V:n käyttöjännitteellä toimivaa, merkitöntä summeria. /9/ Ledinä käytetään Imlecin valmistamaa koteloitua punaista lediä, tyypiltään LED-440R, Farnellin tilausnumerolla 1185299. /10/



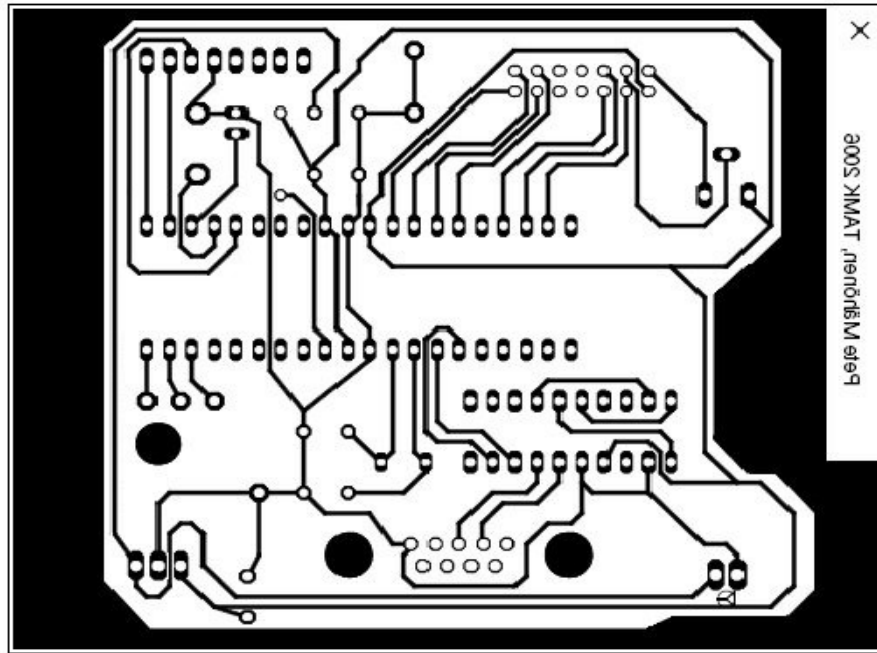
Kuvat 12 ja 13 Koteloitu led ja pietsosummeri /10/,/9/

Punaisen ledin kynnyksjännite on Farnellin sivuston mukaan 2 V ja maksimivirta 30 mA. /9/ Kuten edellä mainittiin, pietsosummerin virrankulutus on 10 mA. Atmegan jokaista sisäänmeno- tai ulostulonastaa voidaan kuormittaa datalehden mukaan 40 mA:n virralla, joten ja sekä pietsosummeri voidaan kytkeä suoraan prosessorin ulostulopinneihin.



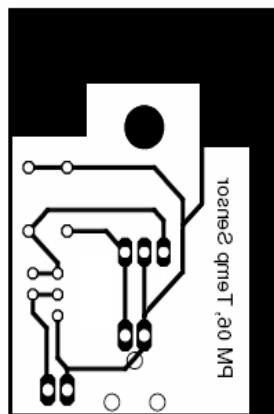
Kuva 14 Kytentäkaaviokuva 2.

Anturipäässä DS18S20 ja 7805-jänniteregulaattori sijoitettiin piirilevyn eri reunoihin ja koetettiin näin vähentää johtumalla aiheutuvia häiriöitä. Piirilevyihin lisättiin myös laaja maataso johtamalla siirtyvien häiriöiden vähentämiseksi. Liittimien sijoitus, niin anturikotelossa kuin päälaitteessakin, tapahtui huomioiden koteloiden valmiit ulostulot ja asennuksen mielekkyys. Näin ollen liitännät sijoitettiin lopulliseen orientaatioon verraten piirilevyn alareunaan. Kytännän mukaisia anturipäitä tehtiin kuusi kappaletta projektin vaatimalla tavalla.



Kuva 15 Päälaitteen piirilevy

Piirilevysuunnittelussa otettiin huomioon nykyaikaisen piirilevysuunnittelun periaatteet ja lopputuloksena piirilevyt päätelaitteelle sekä antureille tarvittavine liitännöineen.



Kuva 16 Anturipään piirilevy

Piirrettäessä piirilevyä otettiin huomioon myös se, että kontaktien vedoissa ei saisi tulla 90 asteen kulmia. Suorat kulmat aiheuttavat suuritaajuuksisissa signaaleissa vastuksen lisääntymistä ja voivat toimia jopa pieninä "antenneina", häiriösäteilyn lähteinä. Vaikka tässä laitteistossa ei olekaan suuritaajuuksisia signaaleita käytössä, silti jokainen vedon kaartoa tehtiin 45 asteen kulmaan.

Prossoria ei juoteta kiinni itse piirilevyyn, vaan se on kontaktissa piirikannan välityksellä. Samoin painonapit ja sarjaliikennepiiri, kuten myös näyttömoduuli tulevat kiinnittymään piirilevyille piikkirimojen välityksellä. Nämä toimenpiteet suoritettiin sen takia, että tulevaisuudessa voitaisiin hyödyntää samaa prossoria ja laitealustaa kehitettäessä kytkentää eteenpäin.

3.9 Ohjelmisto

Projektin alkuvaiheessa testattiin AVR-Studio versio 4:sta sekä Codevision-ohjelmointiympäristöä. Vaikka AVR-Studio tarjosi suuremman määrän valintoja, Codevision valittiin helppokäyttöisyytensä takia. Kummassakin kehitysympäristöissä oli kuitenkin tämän projektin tarvitsemat liitännät STK-500-evaluointikorttiin.

Ohjelmointi päätettiin siis suorittaa Internetistä saatavalla, Windows-käyttäjärjestelmään tarkoitettulla Codevision-AVR-ohjelmointiympäristöllä. Codevision on ilmainen ei-kaupalliseen käyttöön. Tätä kirjoitettaessa uusimman version Codevisionista oli 1.24.8e. /11/

Ohjelmisto tehtiin osissa eli modulaarisesti. Kokonaisuus jaettiin pienempiin osakokonaisuuksiin ja sitten ne ratkaistiin yksi kerrallaan. Ensimmäisenä ohjelmoitiin LCD-näytön alustukset ja toiminta. Codevisionissa tuli mukana kirjasto jota käyttämällä saatiinkin alustukset tehtyä ilman sen suurempia ongelmia.

Toisena osana viritettiin anturit toimimaan. Jokainen anturi tunnistetaan ja sitten sille varataan oma muuttuja lämpötilan lukua varten. Sitten while-loopilla luetaan lämpötilat muuttujiin. Tämän jälkeen muuttujat päivitetään LCD-näytölle tietyn edeltä määrätyn kaavion mukaisesti.

Kolmantena ongelmana oli käyttöliittymä ja sen kytkimet. Laitteessa on kolme kytkintä, led ja varoituspietsosummeri. Laitteen käynnistyessä pyydetään asetusarvoja jokaiselle valvottavalle kohteelle. Yksi painike lisää lämpötilaa puolella asteella, toinen vastaavasti vähentää puolella asteella. Kuittauspainike kuittaa lämpötilan ja siirtyy seuraavan kohteen asetusarvon määrittämiseen. Asetusarvojen syöttämisen jälkeen kuitataan kolmannelta napista kokonaisuus päättyneeksi. Tämän jälkeen laitteisto valvoo ja ilmoittaa asetusarvoista poikkeamista.

Laitteen käynnissä ollessa voidaan kuitenkin asetusarvoja muuttaa kuittauspainiketta painamalla, jolloin asetusarvo alkaa alusta ja voidaan syöttää kaikki arvot uudelleen. Lopetus edelleen kuittausnappia painamalla.

Ohjelmisto toimii periaatteella “valvo ja ilmoita häiriöstä”. Laitteiston käynnistyessä tunnistetaan siihen kytketyt anturit sekä näytetään LCD-näytöllä tekijänoikeustiedot.

Sitten siirrytään silmukkaan, jossa luetaan kaikkien antureiden antamat lukemat ja päivitetään ne näytölle. Lämpötiloja verrataan annettuihin rajoihin ja jos rajat ylittyvät tai alittuvat, ilmoitetaan siitä led-valolla sekä pietsosummerilla paikallisesti. Lisäksi lähetetään 30 min välein päivystyspuhelimeen tekstiviesti vikatilanteesta. Viestin lähetystä toistetaan niin kauan kuin on tarpeen tai niin kauan kuin hälytysmekanismi on kytketty päälle. Mekanismi voidaan kytkeä mekaanisesti pois päältä korjaustoimenpiteiden ajaksi.

4. ASENNUS JA TESTAUS

4.1 Pääyksikkö

Pääyksikön asennus tapahtui vaiheittain. Laitteistoa koekytettäessä testattiin niin LCD-moduulin samoin kuin kaikkien antureiden toimintakin. Piirilevyn valmistuessa tarpeelliset komponentit juotettiin piirilevylle ja käyttötestit toistettiin. Kolmannessa vaiheessa järjestelmä asennettiin paikalleen ja käyttötestit toistettiin siellä.

Päämoduuli asennettiin seinään, joka on laitteistokomeroa vastapäätä Torpan keittiöön johtavan käytävän varrella. Tästä on helppo tehdä kytkennät puhelinkeskuksen kaapelointeihin, jotka olivat asennusseinän takana. Tällä sijainnilla saamme myös aikaiseksi sen, että laitepaneeli huomataan helpommin.

4.2 Lämpötilanmittausyksikkö

Lämpötilanmittausyksiköt asennettiin kohteiden pääkäytäviin sulaketaulujen lähistöille, jotta saataisiin mahdollisimman hyvä sekä kohteen kattava sijainti lämpötila-anturille. Samalla saatiin aikaan myös helppo liitettävyyys sulaketaulun lähellä olevaan kiertovesipumpun tai huippuimurin säätimeen.

4.3 Siirtolinjat

Siirtolinjat kohteessa olivat valmiiksi asennettuna, ja koostuivat kohtalaisen iäkkäistä puhelinparikaapeloinneista. Haastavin osuus tässä vaiheessa oli etsiä dokumentointi 30 vuotta vanhoista puhelinjohtojen asennuksista. Itse kytkentä puhelinkeskuskaapissa oli asentajalle suuri haaste asennusten mittavuuden, dokumentoimattomuuden sekä iäkkyyden takia. Intensiivisten etsintöjen jälkeen oikeat johtimet löytyivät. Alustavien mittausten jälkeen kaapeloinnit osoittautuivat toimiviksi.

4.4 Käyttötестit

Testaus suoritettiin kolmessa osassa. Tämä sen takia että saatiin luotettavat testitulokset jokaisesta osasta kytkentöjä sekä erillisinä moduuleina että yhteenkoottuna kokonaisuutena. Kolmiosaisella testausjaksolla saatiin myös karsittua ympäristöstä johtuvat virhetekijät minimiin testien aikana.

4.4.1 Ensimmäinen testi

Testausten ensimmäisessä osassa selvitettiin koekytkentälevylle kasatun laitteistokokonaisuuden toiminta. Kokonaisuus kytkettiin työpöydällä toimintaan ja käyttöä testattiin satunnaisilla lämpötilan nostoilla tai laskuilla. Apuvälineinä käytettiin lämminilmapuhallinta ja peltier-elementtiä. Satunnaisten kymmenen otoksen ja viiden vuorokauden kestotestin jälkeen todettiin kytkentä toimivaksi.

4.4.2 Toinen testi

Toisessa osassa laitteisto koottiin valmiiksi asti koteloonsa ja lämpötila-anturit kytkettiin omiin koteloihinsa. Ensimmäisen testin kaltainen viiden vuorokauden kestotesti toistettiin ja testiotannat uusittiin. Tämäkin testi onnistui odotusten mukaisesti.

4.4.3 Kolmas testi

Kolmannessa ja viimeisessä testissä laitteisto asennettiin paikalleen kokonaisuudessaan ja varmistettiin kytkennät ulkoisille laitteistoille. Sen jälkeen täysin koottu lopullinen järjestelmä testattiin valmiina lopullisena kokoonpanollaan. Mittaustulosten perusteella laitteisto toimi kuten pitääkin ja se voidaan ottaa jatkuvaan käyttöön.

5. YHTEENVETO

Tässä insinööriyössä kuvataan Tampereen seurakuntayhtymän Torpan Kurssikeskukseen suunnitellun lämpötilanmittausjärjestelmän tuotekehitys- ja suunnitteluprosessia ja itse järjestelmää kuvien ja tekstin avulla. Insinööriyössä on myös selitetty syitä, miksi tämä järjestelmä on ollut tarpeellista rakentaa ja millaisia muutoksia se tuo kiinteistön kustannustehokkuuteen, millaisia muutoksia tarvitaan infrastruktuuriin sekä millaisia muutoksia se tuo vastuuhenkilöiden työpäivään.

Haasteellisinta tässä projektissa oli selvittää 30 vuotta vanhojen kytkentöjen toiminta ja rakentaa vakaasti toimiva järjestelmä, joka on yhteensopiva iäkkäiden laitteistojen kanssa. Myös yhteistyö sekä tiedonhankinta osoittautui hyvinkin konstikkaaksi, sillä vuosien varrella huoltohenkilökunta oli vaihtunut useasti ja jokainen heistä oli tuonut oman osansa infrastruktuuriin.

Järjestelmää olisi voinut parantaa siten, että yksinkertaisen anturitiedon liikuttamisen sijaan mittausyksikköön olisi voitu lisätä esimerkiksi ATTiny11-prosessori, joka valvoisi lämpötilaa keskusyksiköltä saatujen asetustietojen mukaan. Aliprosessori voisi toimia myös talokohtaisena lämpötilanjärjestelmän säätölaitteistona.

Keskusyksikköön voisi liittää flash-muistia, jotta saataisiin tallennettua lämpötilat pidemmältä ajalta ja laatia niiden pohjalta kulutusennusteita ja seurantajakson loputtua tarkistaa ovatko arvioidut laskelmat pitäneet paikkansa. Toteutuneista arvoista saataisiin laadittua tarkkoja kuvaajia esimerkiksi taulukkolaskentaohjelmistolla.

Kiinteistökohtaisesti voitaisiin pienentää lämmityskustannuksia laatimalla Just-On-Time -mallisia lämmityssuunnitelmia. Näin ollen tiettyjen rakennusten lämpötilaa voitaisiin pudottaa käyttämättömyyden ajaksi.

Liikennöinti voitaisiin muuttaa langattomaksi TCP/IP-liikenteeksi, jolla saataisiin sijoitettua useampia antureita vapaammin ympäri aluetta. Vapaa liikkuvuus mahdollistaisi järjestelmän muokkaamisen myös esimerkiksi kulunvalvonta- tai varashälytínjärjestelmäksi.

Vaikka nämä visiot pudotettiin ajanpuutteen ja resurssipulan takia pois projektista, eivät ne ole kuitenkaan täysin poissuljettuja jatkokehitystä ajatellen.

Tässä projektissa keskityttiin tilatuista ominaisuuksista oleellisimpaan asiaan, eli lämpötilanvalvontaan ja mahdollisten lisävahinkojen estämiseen informoimalla nopeasti vastuuhenkilöitä häiriötilasta.

Taloudellinen näkökulma on ollut läsnä koko projektin ajan. Kustannustehokkaan järjestelmän luominen tuo myös mukanaan potentiaalisen kaupallisen puolen esiin. Modulaarisena järjestelmänä sovellettavuus uusiin kokonaisuuksiin on helppo. Kaikki tämän projektin prosessiohjelmoinnit tehtiin käyttäen ilmaisia versioita kehitysympäristöistä. Järjestelmän lähdekoodi on GPL-lisenssin alainen ja on tulevien sovelluskehittäjien käytettävissä.

6. LÄHTEET

Lähteet tarkistettu 22.11.2006

1. Atmel Atmega16 datalehti
http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf
[21.4.2006]
2. Dallas Semiconductor, DS18S20 datalehti
<http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf> , [8.5.2006]
3. Hitachi HD44780-LCD-ohjaimen datalehti
www.web.media.mit.edu/~ayah/documents/hd44780u.pdf
[7.5.2006]
4. DS18S20 Application notes
www.maxim.ic.com/appnotes.cfm/appnote_number/148
[16.5.2006]
5. Lämpötilamittaimet-artikkeli,
<http://www.edu.fi/pageLast.asp?path=498,529,7666,41737,41743,41767>
[24.7.2006]
6. Lämpömittainten kuva,
www.wordsun.com/cr02e.htm [8.8.2006]
7. Nokia 6310:n oppaat ja AT-komennot
<http://europe.nokia.com/A4143050>, [20.10.2006]
8. Optoerotin Avago technologies HCPL-2611:n datalehti
<http://www.farnell.com/datasheets/41313.pdf> [6.11.2006]
9. Merkittömän pietsosummerin datalehti
<http://www.farnell.com/datasheets/82689.pdf> [22.11.2006]
10. Koteloidun ledin IMLEC LED-440R datalehti
<http://www.imlec.com/indicators/model440.pdf> [22.11.2006]
11. Codevision-ohjelmointiympäristö
<http://www.codevision.be> [22.11.2006]

12. Multimec-kytkimen datalehti

<http://http://www.farnell.com/datasheets/74113.pdf> [22.11.2006]

13. Eagle-piirilevynsuunnitteluohjelmisto

<http://www.cadsoft.de> [22.11.2006]

7. LIITTEET

1. Lähdekoodi järjestelmään, 23 sivua

2. Atmega16:n datalehdessä prosessorin sähköiset ominaisuudet (s.294-300) ja oskillaattorin mitoitus (s.26),

http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf, 8 sivua

3. DS18S20:n datalehdessä sivut 1-2, sekä 19-20, liitettävyys sekä sähköiset ominaisuudet,

<http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf>, 4 sivua

4. MAX233CPP:n datalehdessä liitettävyys(s.21), 1 sivu

<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>, 1 sivu

5. 7805-jänniteregulaattorin datalehdessä sivut 1 ja 3, liitettävyys ja sähköiset ominaisuudet, <http://www.national.com/opf/LM/LM7805C.html>, 2 sivua

6. 6310-GSM-puhelimen tärkeimmät AT-komennot(s. 2,3)

<http://europe.nokia.com/A4143050>, 2 sivua

Yhteensä 41 sivua.

```
/*  
Insinööriyö Pete Mähönen, TAMK 2006
```

Lämpötilan seurantajärjestelmä GSM-varmennuksella.

```
Project : TempMon  
Version : 1.0  
Date    : 2.11.2006  
Author  : Pete Mähönen  
Company : PMData/Torpan kurssikeskus  
Comments: Lopputyön koodi
```

```
Chip type      : ATmega16  
Program type   : Application  
Clock frequency : 14,759500 MHz  
Memory model   : Small  
External SRAM size : 0  
Data Stack size : 256  
*****/
```

```
#asm  
    .equ __w1_port=0x1b //anturiportti  
    .equ __w1_bit=6     //nasta jossa anturi kiinni  
    .equ __lcd_port=0x15 //lcd-portti  
#endasm  
  
/* oheiskirjastot codevisionista, prosessorille, lcd:lle, antureille ja  
käyttöliittämälle */  
  
#include <mega16.h>  
#include <lcd.h>  
#include <ds1820.h>  
#include <delay.h>  
#include <stdio.h>  
#include <string.h>  
  
char nayttopuskuri[33];  
  
/* määritellään napit, ledi ja pietso sekä antureitten määrä */  
#define laitemaksimi 8  
#define pluspainike PORTB.1  
#define miinuspainike PORTB.2  
#define asetuspainike PORTB.3  
#define ledivalo PORTA.2  
#define pietso PORTA.3  
#define kannypaalle PORTA.4  
  
#define RXB8 1 //määritellään datasiirron asetukset  
#define TXB8 0  
#define OVR 3  
#define FE 4  
#define UDRE 5  
#define RXC 7
```

```

#define FRAMING_ERROR (1<<FE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

flash static unsigned char alusta[]= "at+cmgf=1"; // 9+1
flash static unsigned char termis[]="at+cnmi=1,2,0,0"; // 15+1
// vastaanotto : rxready[]= "at+csmp=17,167,0,0"; // 18+1 ... muutoin
tähän asti sama
flash static unsigned char lahetys[]="AT+CMGS=";
flash static unsigned char kenelle[]="\"+358445845675\""; //kenelle
lähetetään
flash static unsigned char viesti[]="Lampotilahalytys";
flash static unsigned int klkm=24; //montako kirjainta ylläolevassa
viestissä
flash static unsigned char soitto[]="atd=";
flash static unsigned char haly[]="\"+358445845675\""; //numero on
suunnittelijan mutta muutetaan tarpeen mukaan
int i;

//uart vastaanotin
//-----

#pragma savereg-
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
#asm
    push r26
    push r27
    push r30
    push r31
    in r26,sreg
    push r26
#endasm
status=USR;
data=UDR;
if ((status & (FRAMING_ERROR | DATA_OVERRUN))==0)
{
    rx_buffer[rx_wr_index]=data;
    if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
    if (++rx_counter == RX_BUFFER_SIZE)
    {
        rx_counter=0;
        rx_buffer_overflow=1;
    };
};
#asm
    pop r26
    out sreg,r26
    pop r31
    pop r30
    pop r27
    pop r26
#endasm
}
#pragma savereg+

#ifdef _DEBUG_TERMINAL_IO_

```

```
// USARTIN vastaanottopuskurista palautuva merkki
#define _ALTERNATE_GETCHAR_
#pragma used+

// lähetyspuskuri
#define TX_BUFFER_SIZE 6
char tx_buffer[TX_BUFFER_SIZE];
unsigned char tx_wr_index,tx_rd_index,tx_counter;

// USART-lähetysrutiini
#pragma savereg-
interrupt [USART_TXC] void usart_tx_isr(void)
{
#asm
    push r26
    push r27
    push r30
    push r31
    in r26,sreg
    push r26
#endasm
if (tx_counter)
{
    --tx_counter;
    UDR=tx_buffer[tx_rd_index];
    if (++tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
};
#asm
    pop r26
    out sreg,r26
    pop r31
    pop r30
    pop r27
    pop r26
#endasm
}
#pragma savereg+

#ifdef _DEBUG_TERMINAL_IO_

// Kirjoitetaan merkki USART-lähetyspuskuriin
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((USR & DATA_REGISTER_EMPTY)==0))
{
    tx_buffer[tx_wr_index]=c;
    if (++tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
    ++tx_counter;
}
else UDR=c;
#asm("sei") //sallitaan keskeytykset
}
```

```
#pragma used-
#endif

void krediitit(void)
{
    lcd_init(20);
    lcd_gotoxy(0,0);
    lcd_putsf("Torpan Kurssikeskus ");
    lcd_gotoxy(0,1);
    lcd_putsf("TempMonitor, PM 2006");
    delay_ms(2000);
    lcd_clear();
};

void anturihaku(unsigned char)
{
    unsigned char devices;
    devices=w1_search(0xf0,rom_code);
// ajetaan antureiden rom-koodit taulukkoon
    sprintf(nayttopuskuri,"%u DS1820-anturia\nKytketty ja
luvussa",devices); // annetaan statustieto montako anturia
messissä
    lcd_puts(nayttopuskuri);
    delay_ms(2000);
};

void lammonmittaus(void)
{
    temp1=ds1820_temperature_10(&rom_code[1][0]);
    temp2=ds1820_temperature_10(&rom_code[2][0]);
    temp3=ds1820_temperature_10(&rom_code[3][0]);
    temp4=ds1820_temperature_10(&rom_code[4][0]);
    temp5=ds1820_temperature_10(&rom_code[5][0]);
    temp6=ds1820_temperature_10(&rom_code[6][0]);

    cd_gotoxy(0,0);
    lcd_putsf("A:";temp1;"B:";temp2;"C";temp3);
    lcd_gotoxy(0,1);
    lcd_putsf("D:";temp4;"E:";temp5;"F";temp6);
    delay_ms(2000);
};
```

```
if (asetusnappi ==0) //asetusnappi painettuna
luetaan raja-arvot
{
    lueA;
    int AA=0;AY=0;
    AA=(A-5);AY=(A+5);
    lueB;
    int BA=0;BY=0;
    BA=(B-5);BY=(B+5);
    lueC;
    int CA=0;CY=0;
    CA=(C-5);CY=(C+5);
    lueD;
    int DA=0;DY=0;
    DA=(D-5);DY=(D+5);
    lueE;
    int EA=0;EY=0;
    EA=(E-5);EY=(E+5);
    lueF;
    int FA=0;FY=0;
    FA=(F-5);FY=(F+5);
}

if (temp1 <AA && temp1>AY) //vertaillaan ja hälytetään
//jos tarvis
{ halyta (); };
if (temp2 <BA && temp2>BY)
{ halyta (); };
if (temp3 <CA && temp3>CY)
{ halyta (); };
if (temp4 <DA && temp4>DY)
{ halyta (); };
if (temp5 <EA && temp5>EY)
{ halyta (); };
if (temp6 <FA && temp6>FY)
{ halyta (); };

};
```



```
void lueA(int)
{
  int tempA=20;
  eeprom int A;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("A-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempA);
    if (plusnappi == 0)
    {
      tempA++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempA);
      tempA=A;
    }
    if (miinusnappi ==0)
    {
      tempA--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempA);
      tempA=A;
    }
  }

  return A;
}

void lueB(int)
{
  int tempB=20;
  eeprom int B;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("B-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempB);
    if (plusnappi == 0)
    {
      tempB++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempB);
      tempB=B;
    }
    if (miinusnappi ==0)
    {
      tempB--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempB);
      tempB=B;
    }
  }

  return B;
}
```

```
void lueC(int)
{
  int tempC=20;
  eeprom int C;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("C-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempC);
    if (plusnappi == 0)
    {
      tempC++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempC);
      tempC=C;
    }
    if (miinusnappi ==0)
    {
      tempC--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempC);
      tempC=C;
    }
  }
  return C;
}

void lueD(int)
{
  int tempD=20;
  eeprom int D;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("D-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempD);
    if (plusnappi == 0)
    {
      tempD++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempD);
      tempD=D;
    }
    if (miinusnappi ==0)
    {
      tempD--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempD);
      tempD=D;
    }
  }

  return D;
};
```

```
void lueE(int)
{
  int tempE=20;
  eeprom int E;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("E-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempE);
    if (plusnappi == 0)
    {
      tempE++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempE);
      tempE=E;
    }
    if (miinusnappi ==0)
    {
      tempE--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempE);
      tempE=E;
    }
  }
  return E;
};

void lueF(int)
{
  int tempF=20;
  eeprom int F;
  while (asetusnappi == 1)
  {
    lcd_gotoxy(0,0);
    lcd_putsf("F-talo, asetustalpo ?");
    lcd_gotoxy(0,1);
    lcd_putsf(tempF);
    if (plusnappi == 0)
    {
      tempF++;
      lcd_gotoxy(0,1);
      lcd_putsf(tempF);
      tempF=F;
    }
    if (miinusnappi ==0)
    {
      tempF--;
      lcd_gotoxy(0,1);
      lcd_putsf(tempF);
      tempF=F;
    }
  }
  return F;
};
```

```
void halyta (char paikka)

{
    lcd_gotoxy(0,0);
    lcd_putsf("HALYYTYS");
    lcd_gotoxy(0,1)
    lcd_putsf("Talo";paikka);
    ledivalo=1;
    pietso=1;
    delay_ms(1000);
    asetus();
};

int toiminto;

// Ulkopuolinen keskeytysrutiini

interrupt [EXT_INT0] void ext_int0_isr(void)
{
    i=1;
    delay_ms(300);
    /*

    putchar('A'); //testataan at-komennolla onko puhelin päällä?
    putchar('T'); //jos ei sytytä ledejä puhelimissa puhelin aina
    putchar(0x0D); //käynnistää ja sammuttaa erikseen...
    putchar(0x0A);

    delay_ms(300);
    */
}

/* tehdään taulukko antureiden rom-koodeille */
unsigned char rom_code[laitemaksimi][9];

void asetus(void)
{
    int k=0;

    kannypaalle=1; //virrat päälle, oletusarvoisesti puhelin EI
ole päällä
    delay_ms(10000); //käynnistysviive
    kannypaalle=0;

    for(k=0; k<9; k++)
    {
        putchar(alusta[k]); //6310:n alustus
    }
    k=0;
    putchar(0x0D);
    putchar(0x0A);
    delay_ms(1000);
}
```

```

    for(k = 0; k<15; k++)          //määritellään terminaali
    {
        putchar(termis[k]);
    }
    k=0;
    putchar(0x0D);
    putchar(0x0A);
    delay_ms(1000);

/*
    tekstiviestikeskus, tarvitseeko määritellä ?
    strcpyf(keskus,"AT+CSCA=");
    for (k=0; k<8; k++)
    {putchar(keskus[k]);}

    strcpyf(keskus,"\"+358410400\"");
    for (k=0;k<15; k++){putchar(keskus[k]);}
    putchar(0x0D);
    putchar(0x0A);
*/

    for (k=0;k<8; k++)          //lähetyskomento
    {
        putchar(lahetys[k]);
    }

    for (k=0;k<9; k++)          //kenelle viesti lähtee
    {
        putchar(kenelle[k]);
    }
    putchar(0x0D);
    putchar(0x0A);
    delay_ms(1000);           /*kun on sekuntti viivettä, luuri ehtii
                                toimia*/

for (k=0;k<klkm; k++) //viestin lähetystä
    {
        putchar(viesti[k]);
    }
    putchar(0x1A);           //ctrl-z loppuun 0x1A
    k=0;
    delay_ms(8000);         //viive että viesti lähtee

/* myös soitto mahdollista, ei kuitenkaan soiteta tässä mallissa*/
for (k=0;k<4; k++) //halytetaan
    {
        putchar(soitto[k]);
    }
    for (k=0;k<17; k++) //halytetaan
    {
        putchar(haly[k]);
    }

```

```
        putchar(0x0D);
        putchar(0x0A);
        k=0;

        */
    }

void main(void)
{

// Porttien määrittelyt
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=In
Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=T
State0=T
PORTA=0x00;
DDRA=0xFC;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out
Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=T State1=0
State0=0
PORTD=0x00;
DDRD=0x03;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Global enable interrupts
#asm("sei")

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;

toiminto=0;
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x5F;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

toiminto=0;

// Global enable interrupts
#asm("sei")

unsigned char i,j,devices;
int temp;

krediitit;          // näytön alustus ja alkukrediitit

while 1
{
    anturihaku;          //huudellaan väylään onko siellä
                        //antureita ja näytetään montako jos on

    lammonmittaus;     // mitataan lammit antureista, pyydetään
                        //raja-arvot, vertaillaan
};
}

/* Seuraavilla sivuilla kirjastotiedot, ovat vakiokirjastoja
Codevisionissa. */
```



```
// STRING.H

// CodeVisionAVR C Compiler
// (C) 1998-2005 Pavel Haiduc, HP InfoTech S.R.L.
// Prototypes for string functions

#ifndef _STRING_INCLUDED_
#define _STRING_INCLUDED_

#ifndef NULL
#define NULL 0
#endif

#pragma used+

char *strcat(char *str1, char *str2);
char *strcatf(char *str1, char flash *str2);
char *strchr(char *str, char c);
signed char strcmp(char *str1, char *str2);
signed char strcmpf(char *str1, char flash *str2);
char *strcpy(char *dest, char *src);
char *strcpyf(char *dest, char flash *src);
unsigned char strcspn(char *str, char *set);
unsigned char strcspnf(char *str, char flash *set);
unsigned int strlenf(char flash *str);
char *strncat(char *str1, char *str2, unsigned char n);
char *strncatf(char *str1, char flash *str2, unsigned char n);
signed char strncmp(char *str1, char *str2, unsigned char n);
signed char strncmpf(char *str1, char flash *str2, unsigned char n);
char *strncpy(char *dest, char *src, unsigned char n);
char *strncpyf(char *dest, char flash *src, unsigned char n);
char *strpbrk(char *str, char *set);
char *strpbrkf(char *str, char flash *set);
signed char strpos(char *str, char c);
char *strrchr(char *str, char c);
char *strrpbrk(char *str, char *set);
char *strrpbrkf(char *str, char flash *set);
signed char strrpos(char *str, char c);
char *strstr(char *str1, char *str2);
char *strstrf(char *str1, char flash *str2);
unsigned char strspn(char *str, char *set);
unsigned char strspnf(char *str, char flash *set);
char *strtok(char *str1, char flash *str2);

#ifdef _MODEL_TINY_
unsigned char strlen(char *str);
void *memcpy(void *dest, void *src, char c, unsigned char n);
void *memchr(void *buf, unsigned char c, unsigned char n);
signed char memcmp(void *buf1, void *buf2, unsigned char n);
signed char memcmpf(void *buf1, void flash *buf2, unsigned char n);
void *memcpy(void *dest, void *src, unsigned char n);
void *memcpyf(void *dest, void flash *src, unsigned char n);
void *memmove(void *dest, void *src, unsigned char n);
void *memset(void *buf, unsigned char c, unsigned char n);
#endif

#if defined _MODEL_SMALL_ | defined _MODEL_MEDIUM_ | defined
_MODEL_LARGE_
unsigned int strlen(char *str);
```

```
void *memcpy(void *dest,void *src,char c,unsigned n);
void *memchr(void *buf,unsigned char c,unsigned n);
signed char memcmp(void *buf1,void *buf2,unsigned n);
signed char memcpcf(void *buf1,void flash *buf2,unsigned n);
void *memcpy(void *dest,void *src,unsigned n);
void *memcpyf(void *dest,void flash *src,unsigned n);
void *memmove(void *dest,void *src,unsigned n);
void *memset(void *buf,unsigned char c,unsigned n);
#endif

#pragma used-
#pragma library string.lib

#endif
```

```
// STDIO.H

// CodeVisionAVR C Compiler
// (C) 1998-2003 Pavel Haiduc, HP InfoTech S.R.L.

// Prototypes for standard I/O functions

#ifndef _STDIO_INCLUDED_
#define _STDIO_INCLUDED_

#include <stdarg.h>

#ifndef NULL
#define NULL 0
#endif

#ifndef EOF
#define EOF -1
#endif

#pragma used+

char getchar(void);
void putchar(char c);
void puts(char *str);
void putsf(char flash *str);

#ifdef _MODEL_TINY_
char *gets(char *str, unsigned char len);
#endif

#ifdef _MODEL_SMALL_
char *gets(char *str, unsigned int len);
#endif

void printf(char flash *fmtstr, ...);
void sprintf(char *str, char flash *fmtstr, ...);
void vprintf(char flash *fmtstr, va_list argptr);
void vsprintf(char *str, char flash *fmtstr, va_list argptr);
signed char scanf(char flash *fmtstr, ...);
signed char sscanf(char *str, char flash *fmtstr, ...);

#ifdef _DEBUG_TERMINAL_IO_
// Used for AVR Studio Terminal I/O
// DON'T CHANGE!

#ifndef _DEBUG_TERMINAL_IO_FN_DEFINED_
#define _DEBUG_TERMINAL_IO_FN_DEFINED_

#pragma warn-
void _c_getchar(void) {}

char getchar(void)
{
    #asm
        push    r16
        push    r17
    getchar0:
    #endasm
    _c_getchar();
    #asm

```

```
        cpi    r17,0xff
        breq   getchar0
        mov    r30,r16
        pop   r17
        pop   r16
#endasm
}

void _c_putchar(void) {}

void putchar(char c)
{
#asm
    mov    r30,r16
    ld     r16,y
#endasm
_c_putchar();
#asm
    mov    r16,r30
#endasm
}

#ifdef _WARNINGS_ON_
#pragma warn+
#endif

#endif

#endif
#pragma library stdio.lib

#endif
```

```
// DELAY.H

// CodeVisionAVR C Compiler
// (C) 1998-2000 Pavel Haiduc, HP InfoTech S.R.L.

#ifndef _DELAY_INCLUDED_
#define _DELAY_INCLUDED_

#pragma used+

void delay_us(unsigned int n);
void delay_ms(unsigned int n);

#pragma used-

#endif
```

```
// MEGA16.H

// CodeVisionAVR C Compiler
// (C) 1998-2001 Pavel Haiduc, HP InfoTech S.R.L.

// I/O registers definitions for the ATmega16

#ifndef _MEGA16_INCLUDED_
#define _MEGA16_INCLUDED_

#pragma used+
sfrb TWBR=0;
sfrb TWSR=1;
sfrb TWAR=2;
sfrb TWDR=3;
sfrb ADCL=4;
sfrb ADCH=5;
sfrw ADCW=4; // 16 bit access
sfrb ADCSRA=6;
sfrb ADMUX=7;
sfrb ACSR=8;
sfrb UBRRL=9;
sfrb UCSRB=0xa;
sfrb UCSRA=0xb;
sfrb UDR=0xc;
sfrb SPCR=0xd;
sfrb SPSR=0xe;
sfrb SPDR=0xf;
sfrb PIND=0x10;
sfrb DDRD=0x11;
sfrb PORTD=0x12;
sfrb PINC=0x13;
sfrb DDRC=0x14;
sfrb PORTC=0x15;
sfrb PINB=0x16;
sfrb DDRB=0x17;
sfrb PORTB=0x18;
sfrb PINA=0x19;
sfrb DDRA=0x1a;
sfrb PORTA=0x1b;
sfrb EECR=0x1c;
sfrb EEDR=0x1d;
sfrb EEARL=0x1e;
sfrb EEARH=0x1f;
sfrw EEAR=0x1e; // 16 bit access
sfrb UBRRH=0x20;
sfrb UCSRC=0x20;
sfrb WDTCSR=0x21;
sfrb ASSR=0x22;
sfrb OCR2=0x23;
sfrb TCNT2=0x24;
sfrb TCCR2=0x25;
sfrb ICR1L=0x26;
sfrb ICR1H=0x27;
sfrb OCR1BL=0x28;
sfrb OCR1BH=0x29;
sfrw OCR1B=0x28; // 16 bit access
sfrb OCR1AL=0x2a;
```

```
sfrb OCR1AH=0x2b;
sfrw OCR1A=0x2a; // 16 bit access
sfrb TCNT1L=0x2c;
sfrb TCNT1H=0x2d;
sfrw TCNT1=0x2c; // 16 bit access
sfrb TCCR1B=0x2e;
sfrb TCCR1A=0x2f;
sfrb SFIOR=0x30;
sfrb OSCCAL=0x31;
sfrb OCDR=0x31;
sfrb TCNT0=0x32;
sfrb TCCR0=0x33;
sfrb MCUCSR=0x34;
sfrb MCUCR=0x35;
sfrb TWCR=0x36;
sfrb SPMCR=0x37;
sfrb TIFR=0x38;
sfrb TIMSK=0x39;
sfrb GIFR=0x3a;
sfrb GICR=0x3b;
sfrb OCR0=0x3c;
sfrb SPL=0x3d;
sfrb SPH=0x3e;
sfrb SREG=0x3f;
#pragma used-
```

```
// Interrupt vectors definitions
```

```
#define EXT_INT0 2
#define EXT_INT1 3
#define TIM2_COMP 4
#define TIM2_OVF 5
#define TIM1_CAPT 6
#define TIM1_COMPA 7
#define TIM1_COMPB 8
#define TIM1_OVF 9
#define TIM0_OVF 10
#define SPI_STC 11
#define USART_RXC 12
#define USART_DRE 13
#define USART_TXC 14
#define ADC_INT 15
#define EE_RDY 16
#define ANA_COMP 17
#define TWI 18
#define EXT_INT2 19
#define TIM0_COMP 20
#define SPM_READY 21

#endif
```

```
// LCD.H

/* LCD driver routines

CodeVisionAVR C Compiler
(C) 1998-2003 Pavel Haiduc, HP InfoTech S.R.L.

BEFORE #include -ING THIS FILE YOU
MUST DECLARE THE I/O ADDRESS OF THE
DATA REGISTER OF THE PORT AT WHICH
THE LCD IS CONNECTED!

EXAMPLE FOR PORTB:

    #asm
        .equ __lcd_port=0x18
    #endasm
    #include <lcd.h>
*/

#ifndef _LCD_INCLUDED_
#define _LCD_INCLUDED_

#pragma used+

void _lcd_ready(void);
void _lcd_write_data(unsigned char data);
// write a byte to the LCD character generator or display RAM
void lcd_write_byte(unsigned char addr, unsigned char data);
// read a byte from the LCD character generator or display RAM
unsigned char lcd_read_byte(unsigned char addr);
// set the LCD display position  x=0..39 y=0..3
void lcd_gotoxy(unsigned char x, unsigned char y);
// clear the LCD
void lcd_clear(void);
void lcd_putchar(char c);
// write the string str located in SRAM to the LCD
void lcd_puts(char *str);
// write the string str located in FLASH to the LCD
void lcd_putsf(char flash *str);
// initialize the LCD controller
unsigned char lcd_init(unsigned char lcd_columns);

#pragma used-
#pragma library lcd.lib

#endif
```



```
//DS1820.H

/*
CodeVisionAVR C Compiler
(C) 1998-2005 Pavel Haiduc, HP InfoTech S.R.L.

Prototypes for Dallas Semiconductor
DS1820 1 Wire bus temperature sensor
functions

BEFORE #include -ING THIS FILE YOU
MUST DECLARE THE I/O ADDRESS OF THE
DATA REGISTER OF THE PORT AT WHICH
THE 1 WIRE BUS IS CONNECTED AND
THE DATA BIT USED

EXAMPLE FOR PORTB:

    #asm
        .equ __w1_port=0x18
        .equ __w1_bit=3
    #endasm
    #include <ds1820.h>
*/

#ifndef _DS1820_INCLUDED_
#define _DS1820_INCLUDED_

#include <1wire.h>

#define DS1820_FAMILY_CODE 0x10
#define DS1820_SEARCH_ROM_CMD 0xf0
#define DS1820_ALARM_SEARCH_CMD 0xec

#pragma used+
extern struct __ds1820_scratch_pad_struct
{
    unsigned char temp_lsb,temp_msb,
        temp_high,temp_low,
        res1,res2,
        cnt_rem,cnt_c,
        crc;
} __ds1820_scratch_pad;

unsigned char ds1820_select(unsigned char *addr);
unsigned char ds1820_read_spd(unsigned char *addr);
int ds1820_temperature_10(unsigned char *addr);
unsigned char ds1820_set_alarm(unsigned char *addr,signed char
temp_low,signed char temp_high);
#pragma used-

#pragma library ds1820.lib

#endif
```

```
// 1WIRE.H

/*
  CodeVisionAVR C Compiler
  (C) 1998-2000 Pavel Haiduc, HP InfoTech S.R.L.

  Prototypes for Dallas Semiconductor
  1 Wire protocol functions

  BEFORE #include -ING THIS FILE YOU
  MUST DECLARE THE I/O ADDRESS OF THE
  DATA REGISTER OF THE PORT AT WHICH
  THE 1 WIRE BUS IS CONNECTED AND
  THE DATA BIT USED

  EXAMPLE FOR PORTB:

      #asm
          .equ __w1_port=0x18
          .equ __w1_bit=3
      #endasm
      #include <1wire.h>
*/

#ifndef _W1_INCLUDED_
#define _W1_INCLUDED_

#pragma used+
unsigned char w1_init(void);
unsigned char w1_read(void);
unsigned char w1_write(unsigned char data);
unsigned char w1_search(unsigned char cmd,void *p);
unsigned char w1_dow_crc8(void *p,unsigned char n);
#pragma used-

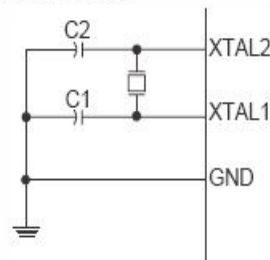
#endif
```



This mode has a limited frequency range and it can not be used to drive other clock buffers.

For resonators, the maximum frequency is 8 MHz with CKOPT unprogrammed and 16 MHz with CKOPT programmed. C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in Table 4. For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 12. Crystal Oscillator Connections



The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in Table 4.

Table 4. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 - 0.9	-
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

Note: 1. This option should not be used with crystals, only with ceramic resonators.



Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except RESET with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on RESET with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0mA PDIP and 400.0mA TQFP/MLF

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 2.7V$ to $5.5V$ (Unless Otherwise Noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage except XTAL1 and RESET pins	$V_{CC}=2.7 - 5.5$	-0.5		$0.2 V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage except XTAL1 and RESET pins	$V_{CC}=2.7 - 5.5$	$0.6 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage XTAL1 pin	$V_{CC}=2.7 - 5.5$	$0.7 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage XTAL1 pin	$V_{CC}=2.7 - 5.5$	-0.5		$0.1 V_{CC}^{(1)}$	V
V_{IH2}	Input High Voltage RESET pin	$V_{CC}=2.7 - 5.5$	$0.9 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IL2}	Input Low Voltage RESET pin	$V_{CC}=2.7 - 5.5$	-0.5		$0.2 V_{CC}$	V
V_{OL}	Output Low Voltage ⁽³⁾ (Ports A,B,C,D)	$I_{OL} = 20 \text{ mA}, V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}, V_{CC} = 3V$			0.7 0.5	V V
V_{OH}	Output High Voltage ⁽⁴⁾ (Ports A,B,C,D)	$I_{OH} = -20 \text{ mA}, V_{CC} = 5V$ $I_{OH} = -10 \text{ mA}, V_{CC} = 3V$	4.2 2.2			V V
I_{IL}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin low (absolute value)			1	μA
I_{IH}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin high (absolute value)			1	μA
R_{RST}	Reset Pull-up Resistor		30		60	$k\Omega$
R_{pu}	I/O Pin Pull-up Resistor		20		50	$k\Omega$

ATmega16(L)

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 5.5V (Unless Otherwise Noted) (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{CC}	Power Supply Current	Active 1 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)		1.1		mA
		Active 4 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)		3.8	5	mA
		Active 8 MHz, $V_{CC} = 5\text{V}$ (ATmega16)		12	15	mA
		Idle 1 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)		0.35		mA
		Idle 4 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)		1.2	2	mA
		Idle 8 MHz, $V_{CC} = 5\text{V}$ (ATmega16)		5.5	7	mA
	Power-down Mode ⁽⁶⁾	WDT enabled, $V_{CC} = 3\text{V}$		<8	15	μA
		WDT disabled, $V_{CC} = 3\text{V}$		< 1	4	μA
V_{ACIO}	Analog Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	$V_{CC} = 2.7\text{V}$ $V_{CC} = 4.0\text{V}$		750 500		ns

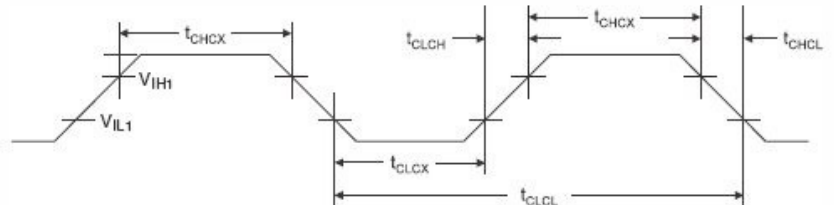
- Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low
 2. "Min" means the lowest value where the pin is guaranteed to be read as high
 3. Although each I/O port can sink more than the test conditions (20 mA at $V_{CC} = 5\text{V}$, 10 mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
PDIP Package:
1) The sum of all IOL, for all ports, should not exceed 200 mA.
2) The sum of all IOL, for port A0 - A7, should not exceed 100 mA.
3) The sum of all IOL, for ports B0 - B7, C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.
TQFP and QFN/MLF Package:
1) The sum of all IOL, for all ports, should not exceed 400 mA.
2) The sum of all IOL, for ports A0 - A7, should not exceed 100 mA.
3) The sum of all IOL, for ports B0 - B4, should not exceed 100 mA.
4) The sum of all IOL, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.
5) The sum of all IOL, for ports D3 - D7, should not exceed 100 mA.
6) The sum of all IOL, for ports C0 - C7, should not exceed 100 mA.
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
 4. Although each I/O port can source more than the test conditions (20 mA at $V_{CC} = 5\text{V}$, 10 mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
PDIP Package:
1) The sum of all IOH, for all ports, should not exceed 200 mA.
2) The sum of all IOH, for port A0 - A7, should not exceed 100 mA.
3) The sum of all IOH, for ports B0 - B7, C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.
TQFP and QFN/MLF Package:
1) The sum of all IOH, for all ports, should not exceed 400 mA.
2) The sum of all IOH, for ports A0 - A7, should not exceed 100 mA.
3) The sum of all IOH, for ports B0 - B4, should not exceed 100 mA.
4) The sum of all IOH, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.



- 5] The sum of all IOH, for ports D3 - D7, should not exceed 100 mA.
- 6] The sum of all IOH, for ports C0 - C7, should not exceed 100 mA. If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
- 5. Minimum V_{CC} for Power-down is 2.5V.

External Clock Drive Waveforms

Figure 145. External Clock Drive Waveforms



External Clock Drive

Table 118. External Clock Drive⁽¹⁾

Symbol	Parameter	$V_{CC} = 2.7V$ to $5.5V$		$V_{CC} = 4.5V$ to $5.5V$		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	8	0	16	MHz
t_{CLCL}	Clock Period	125		62.5		ns
t_{CHCX}	High Time	50		25		ns
t_{CLCX}	Low Time	50		25		ns
t_{CLCH}	Rise Time		1.6		0.5	μs
t_{CHCL}	Fall Time		1.6		0.5	μs
Δt_{CLCL}	Change in period from one clock cycle to the next		2		2	%

Note: 1. Refer to "External Clock" on page 31 for details.

Table 119. External RC Oscillator, Typical Frequencies ($V_{CC} = 5$)

R [k Ω] ⁽¹⁾	C [pF]	f ⁽²⁾
33	22	650 kHz
10	22	2.0 MHz

- Notes: 1. R should be in the range 3 k Ω - 100 k Ω , and C should be at least 20 pF.
- 2. The frequency will vary with package type and board layout.

ATmega16(L)

Two-wire Serial Interface Characteristics

Table 120 describes the requirements for devices connected to the Two-wire Serial Bus. The ATmega16 Two-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 146.

Table 120. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	$0.3 V_{CC}$	V
V_{IH}	Input High-voltage		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{HYS}^{(1)}$	Hysteresis of Schmitt Trigger Inputs		$0.05 V_{CC}^{(2)}$	–	V
$V_{OL}^{(1)}$	Output Low-voltage	3 mA sink current	0	0.4	V
$t_r^{(1)}$	Rise Time for both SDA and SCL		$20 + 0.1C_b^{(3)(2)}$	300	ns
$t_{cf}^{(1)}$	Output Fall Time from V_{Hmin} to V_{ILmax}	$10 \text{ pF} < C_b < 400 \text{ pF}^{(3)}$	$20 + 0.1C_b^{(3)(2)}$	250	ns
$t_{SP}^{(1)}$	Spikes Suppressed by Input Filter		0	$50^{(2)}$	ns
I_I	Input Current each I/O Pin	$0.1V_{CC} < V_I < 0.9V_{CC}$	-10	10	μA
$C_I^{(1)}$	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	$f_{CK}^{(4)} > \max(16f_{SCL}, 250\text{kHz})^{(5)}$	0	400	kHz
R_p	Value of Pull-up resistor	$f_{SCL} \leq 100 \text{ kHz}$	$\frac{V_{CC} - 0.4V}{3\text{mA}}$	$\frac{1000\text{ns}}{C_b}$	Ω
		$f_{SCL} > 100 \text{ kHz}$	$\frac{V_{CC} - 0.4V}{3\text{mA}}$	$\frac{300\text{ns}}{C_b}$	Ω
$t_{HD:STA}$	Hold Time (repeated) START Condition	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
t_{LOW}	Low Period of the SCL Clock	$f_{SCL} \leq 100 \text{ kHz}^{(6)}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}^{(7)}$	1.3	–	μs
t_{HIGH}	High period of the SCL clock	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
$t_{SU:STA}$	Set-up time for a repeated START condition	$f_{SCL} \leq 100 \text{ kHz}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
$t_{HD:DAT}$	Data hold time	$f_{SCL} \leq 100 \text{ kHz}$	0	3.45	μs
		$f_{SCL} > 100 \text{ kHz}$	0	0.9	μs
$t_{SU:DAT}$	Data setup time	$f_{SCL} \leq 100 \text{ kHz}$	250	–	ns
		$f_{SCL} > 100 \text{ kHz}$	100	–	ns
$t_{SU:STO}$	Setup time for STOP condition	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100 \text{ kHz}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}$	1.3	–	μs

Notes: 1. In ATmega16, this parameter is characterized and not 100% tested.

2. Required only for $f_{SCL} > 100 \text{ kHz}$.

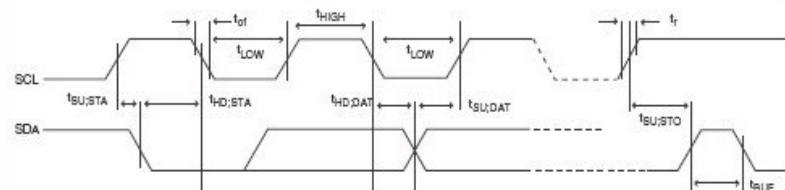
3. C_b = capacitance of one bus line in pF.

4. f_{CK} = CPU clock frequency



5. This requirement applies to all ATmega16 Two-wire Serial Interface operation. Other devices connected to the Two-wire Serial Bus need only obey the general f_{SCL} requirement.
6. The actual low period generated by the ATmega16 Two-wire Serial Interface is $(1/f_{SCL} - 2/f_{CK})$, thus f_{CK} must be greater than 6 MHz for the low time requirement to be strictly met at $f_{SCL} = 100$ kHz.
7. The actual low period generated by the ATmega16 Two-wire Serial Interface is $(1/f_{SCL} - 2/f_{CK})$, thus the low time requirement will not be strictly met for $f_{SCL} > 308$ kHz when $f_{CK} = 8$ MHz. Still, ATmega16 devices connected to the bus may communicate at full speed (400 kHz) with other ATmega16 devices, as well as any other device with a proper t_{LOW} acceptance margin.

Figure 146. Two-wire Serial Bus Timing



SPI Timing Characteristics

See Figure 147 and Figure 148 for details.

Table 121. SPI Timing Parameters

	Description	Mode	Min	Typ	Max	
1	SCK period	Master		See Table 58		ns
2	SCK high/low	Master		50% duty cycle		
3	Rise/Fall time	Master		3.6		
4	Setup	Master		10		
5	Hold	Master		10		
6	Out to SCK	Master		$0.5 \cdot t_{SCK}$		
7	SCK to out	Master		10		
8	SCK to out high	Master		10		
9	\overline{SS} low to out	Slave		15		
10	SCK period	Slave	$4 \cdot t_{SCK}$			
11	SCK high/low	Slave	$2 \cdot t_{SCK}$			
12	Rise/Fall time	Slave			1.6	μs
13	Setup	Slave	10			ns
14	Hold	Slave	10			
15	SCK to out	Slave		15		
16	SCK to \overline{SS} high	Slave	20			
17	\overline{SS} high to tri-state	Slave		10		
18	\overline{SS} low to SCK	Slave	$2 \cdot t_{SCK}$			

ATmega16(L)

Figure 147. SPI Interface Timing Requirements (Master Mode)

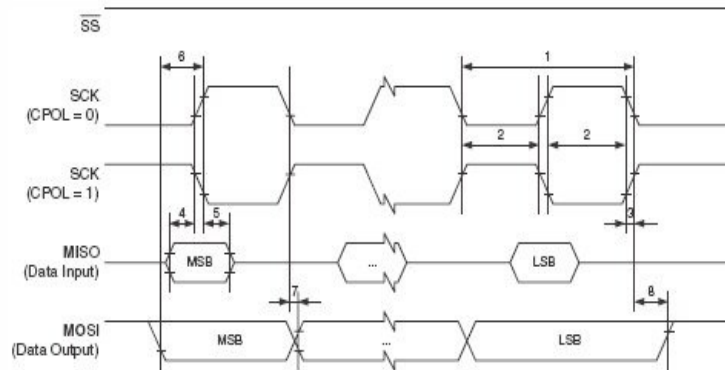
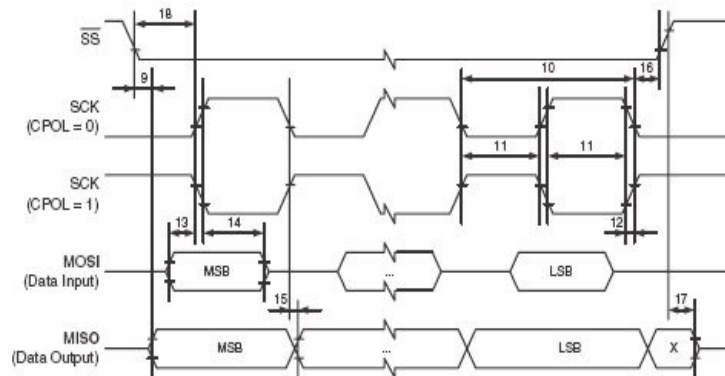


Figure 148. SPI Interface Timing Requirements (Slave Mode)





ADC Characteristics – Preliminary Data

Table 122. ADC Characteristics

Symbol	Parameter	Condition	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾	Units	
	Resolution	Single Ended Conversion		10		Bits	
		Differential Conversion Gain = 1x or 10x		8		Bits	
		Differential Conversion Gain = 200x		7		Bits	
	Absolute Accuracy (Including INL, DNL, Quantization Error, Gain, and Offset Error).	Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz		1.5	2.5	LSB	
		Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 1 MHz		3	4	LSB	
		Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz Noise Reduction mode		1.5		LSB	
		Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 1 MHz Noise Reduction mode		3		LSB	
	Integral Non-linearity (INL)	Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz		1		LSB	
	Differential Non-linearity (DNL)	Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz		0.5		LSB	
	Gain Error	Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz		1		LSB	
	Offset Error	Single Ended Conversion $V_{REF} = 4V, V_{CC} = 4V$ ADC clock = 200 kHz				LSB	
	Conversion Time	Free Running Conversion	13		260	μs	
	Clock Frequency		50		1000	kHz	
AVCC	Analog Supply Voltage		$V_{CC} - 0.3^{(2)}$		$V_{CC} + 0.3^{(3)}$	V	
V_{REF}	Reference Voltage	Single Ended Conversion	2.0		AVCC	V	
		Differential Conversion	2.0		AVCC - 0.2	V	
V_{IN}	Input voltage	Single ended channels	GND		V_{REF}	V	
		Differential channels	0		V_{REF}	V	
	Input bandwidth	Single ended channels		38.5			kHz
		Differential channels		4			kHz



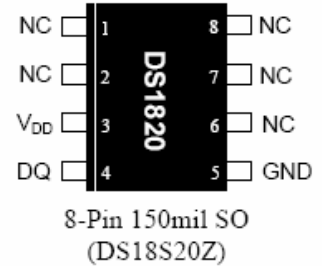
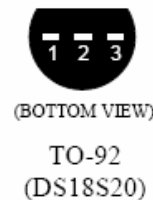
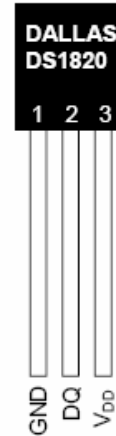
DS18S20 High-Precision 1-Wire Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire[®] interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an onboard ROM
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ accuracy from -10°C to $+85^{\circ}\text{C}$
- 9-bit thermometer resolution
- Converts temperature in 750ms (max.)
- User-definable nonvolatile (NV) alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
 V_{DD} - Power Supply Voltage
NC - No Connect

DESCRIPTION

The DS18S20 Digital Thermometer provides 9-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to $+125^{\circ}\text{C}$ and is accurate to $\pm 0.5^{\circ}\text{C}$ over the range of -10°C to $+85^{\circ}\text{C}$. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-Wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

DETAILED PIN DESCRIPTIONS Table 1

8-PIN SOIC*	TO-92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	Data Input/Output Pin. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	3	V _{DD}	Optional V_{DD} Pin. V _{DD} must be grounded for operation in parasite power mode.

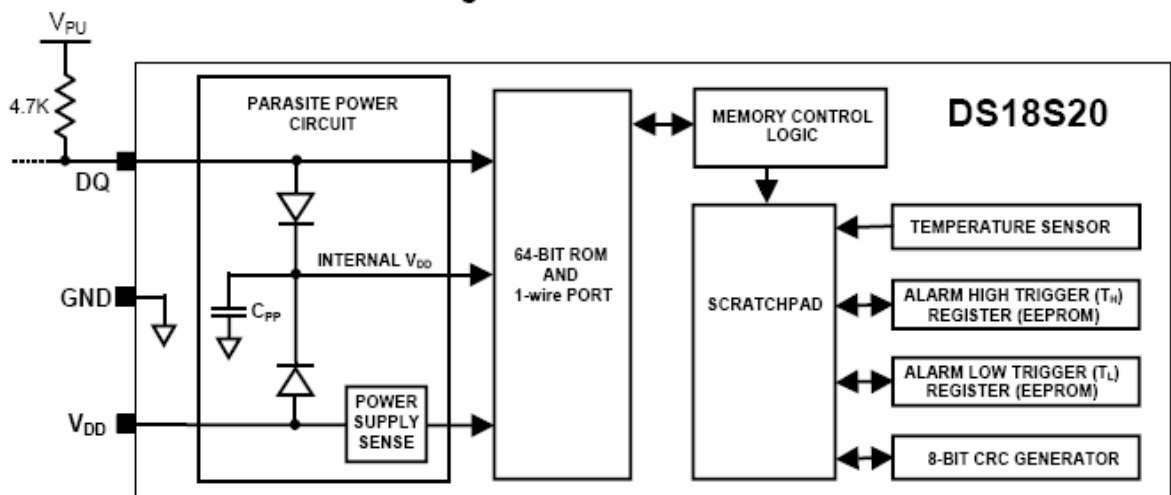
*All pins not specified in this table are "No Connect" pins.

OVERVIEW

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L). The T_H and T_L registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Dallas' exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-WIRE BUS SYSTEM* section of this datasheet.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V_{DD}.

DS18S20 BLOCK DIAGRAM Figure 1

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +6.0V
Operating Temperature Range	-55°C to +125°C
Storage Temperature Range	-55°C to +125°C
Solder Temperature	See IPC/JEDEC J-STD-020A
Reflow Oven Temperature	+220°C

*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD} = 3.0V$ to $5.5V$)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1, 2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic Low	V_{IL}		-0.3		+0.8	V	1, 4, 5
Input Logic High	V_{IH}	Local Power	+2.2		The lower of 5.5 or $V_{DD} + 0.3$	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	$V_{IO}=0.4V$	4.0			mA	1
Standby Current	I_{DDs}			750	1000	nA	7, 8
Active Current	I_{DD}	$V_{DD}=5V$		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		μA	10
Drift				±0.2		°C	11

NOTES:

- 1) All voltages are referenced to ground.
- 2) The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU} . In order to meet the V_{IH} spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
- 3) See typical performance curve in Figure 16
- 4) Logic low voltages are specified at a sink current of 4mA.
- 5) To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
- 6) Logic high voltages are specified at a source current of 1mA.
- 7) Standby current specified up to 70°C. Standby current typically is 3μA at 125°C.
- 8) To minimize I_{DDs} , DQ should be within the following ranges: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
- 9) Active current refers to supply current during active temperature conversions or EEPROM writes.
- 10) DQ line is high ("hi-Z" state).
- 11) Drift data is based on a 1000 hour stress test at 125°C with $V_{DD} = 5.5V$.

AC ELECTRICAL CHARACTERISTICS: NV MEMORY(-55°C to +100°C; $V_{DD} = 3.0V$ to 5.5V)

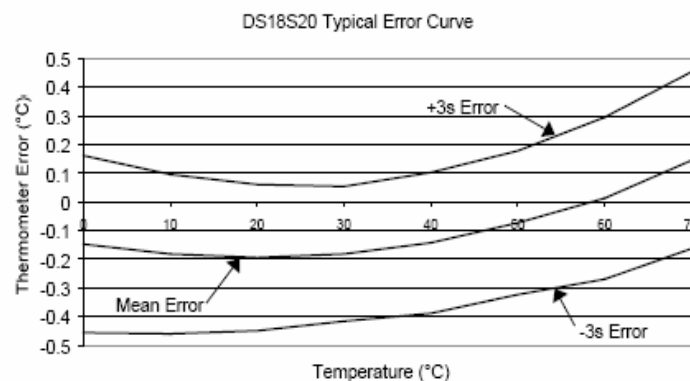
PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{wr}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}				750	ms	1
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	μs	
Time Slot	t_{SLOT}		60		120	μs	1
Recovery Time	t_{REC}		1			μs	1
Write 0 Low Time	t_{LOW0}		60		120	μs	1
Write 1 Low Time	t_{LOW1}		1		15	μs	1
Read Data Valid	t_{RDV}				15	μs	1
Reset Time High	t_{RSTH}		480			μs	1
Reset Time Low	t_{RSTL}		480			μs	1, 2
Presence Detect High	t_{PDHIGH}		15		60	μs	1
Presence Detect Low	t_{PDLOW}		60		240	μs	1
Capacitance	$C_{IN/OUT}$				25	pF	

NOTES:

- 1) Refer to timing diagrams in Figure 17.
- 2) Under parasite power, if $t_{RSTL} > 960\mu s$, a power on reset may occur.

TYPICAL PERFORMANCE CURVE Figure 16

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

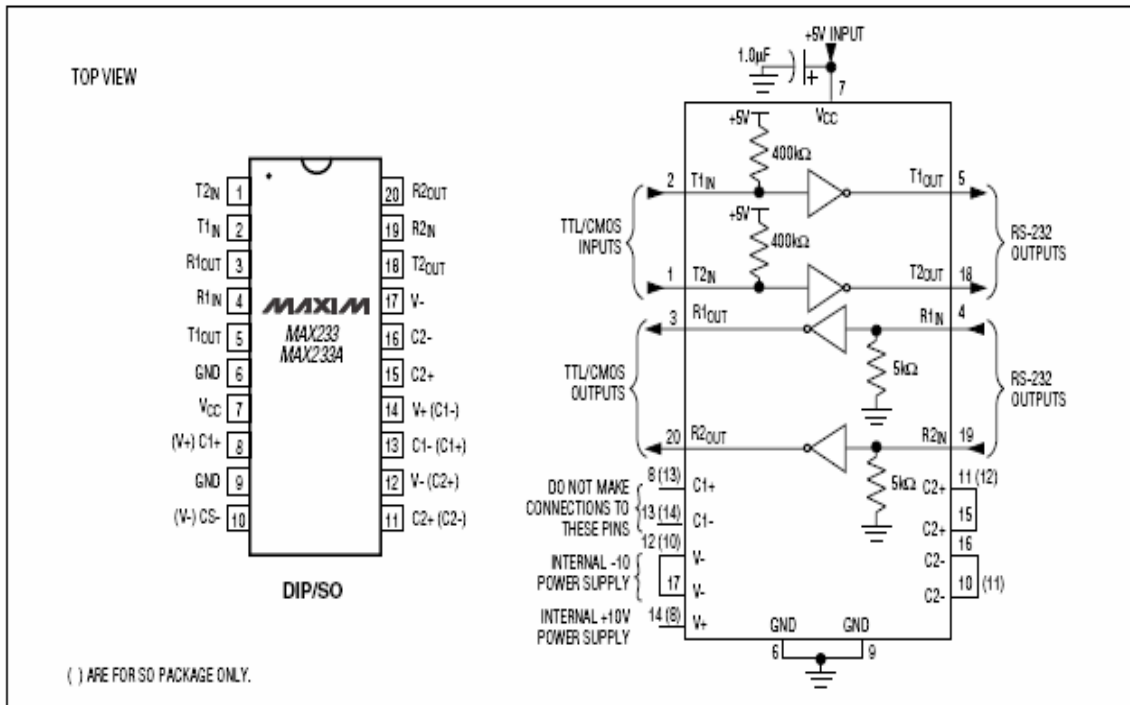


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

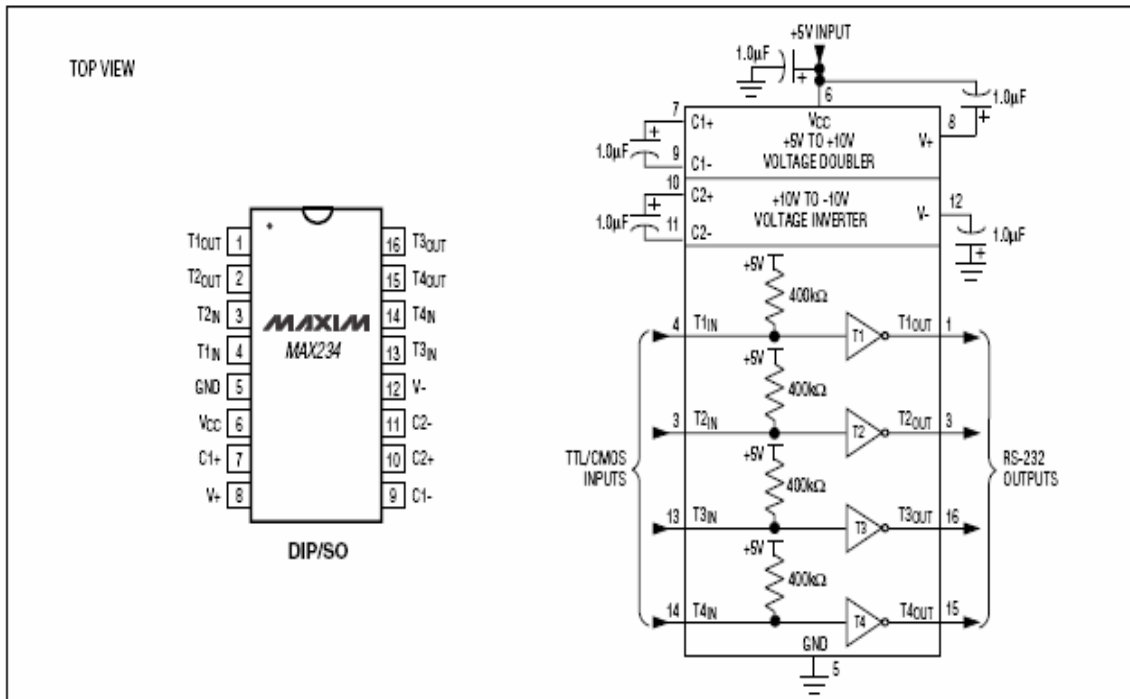
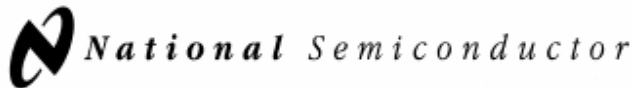


Figure 12. MAX234 Pin Configuration and Typical Operating Circuit



May 2000

LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

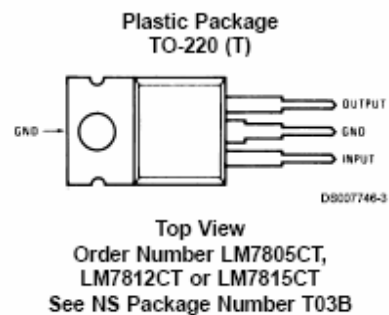
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

Connection Diagrams



Output Voltage		5V			12V			15V			Units			
Input Voltage (unless otherwise noted)		10V			19V			23V						
Symbol	Parameter	Conditions			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_O	Output Voltage	$T_J = 25^\circ\text{C}, 5\text{ mA} \leq I_O \leq 1\text{ A}$			4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		$P_D \leq 15\text{ W}, 5\text{ mA} \leq I_O \leq 1\text{ A}$			4.75		5.25	11.4		12.6	14.25		15.75	V
		$V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$			(7.5 $\leq V_{\text{IN}} \leq 20$)			(14.5 $\leq V_{\text{IN}} \leq 27$)			(17.5 $\leq V_{\text{IN}} \leq 30$)			V
ΔV_O	Line Regulation	$I_O = 500\text{ mA}$	$T_J = 25^\circ\text{C}$			3	50	4	120	4	150			mV
			ΔV_{IN}			(7 $\leq V_{\text{IN}} \leq 25$)			14.5 $\leq V_{\text{IN}} \leq 30$			(17.5 $\leq V_{\text{IN}} \leq 30$)		
		$0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$			50			120			150			mV
		ΔV_{IN}			(8 $\leq V_{\text{IN}} \leq 20$)			(15 $\leq V_{\text{IN}} \leq 27$)			(18.5 $\leq V_{\text{IN}} \leq 30$)			V
ΔV_O	Load Regulation	$T_J = 25^\circ\text{C}$	$5\text{ mA} \leq I_O \leq 1.5\text{ A}$			10	50	12	120	12	150			mV
			$250\text{ mA} \leq I_O \leq 750\text{ mA}$			25			60			75		
		$5\text{ mA} \leq I_O \leq 1\text{ A}, 0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$			50			120			150			mV
		I_Q	Quiescent Current	$I_O \leq 1\text{ A}$	$T_J = 25^\circ\text{C}$			8		8		8		
$0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$					8.5			8.5			8.5			mA
ΔI_Q	Quiescent Current Change	$5\text{ mA} \leq I_O \leq 1\text{ A}$			0.5			0.5			0.5			mA
		$T_J = 25^\circ\text{C}, I_O \leq 1\text{ A}$			1.0			1.0			1.0			mA
		$V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$			(7.5 $\leq V_{\text{IN}} \leq 20$)			(14.8 $\leq V_{\text{IN}} \leq 27$)			(17.9 $\leq V_{\text{IN}} \leq 30$)			V
V_N	Output Noise Voltage	$T_A = 25^\circ\text{C}, 10\text{ Hz} \leq f \leq 100\text{ kHz}$			40			75			90			μV
		$\frac{\Delta V_{\text{IN}}}{\Delta V_{\text{OUT}}}$	Ripple Rejection	$I_O \leq 1\text{ A}, T_J = 25^\circ\text{C}$ or $I_O \leq 500\text{ mA}$ $0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$			62	80	55	72	54	70		
$f = 120\text{ Hz}$				62			55			54			dB	
$V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$				(8 $\leq V_{\text{IN}} \leq 18$)			(15 $\leq V_{\text{IN}} \leq 25$)			(18.5 $\leq V_{\text{IN}} \leq 28.5$)			V	
R_O	Dropout Voltage	$T_J = 25^\circ\text{C}, I_{\text{OUT}} = 1\text{ A}$			2.0			2.0			2.0			V
	Output Resistance	$f = 1\text{ kHz}$			8			18			19			$\text{m}\Omega$

LM78XX

Absolute Maximum Ratings (Note 3)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage

 $(V_O = 5\text{ V}, 12\text{ V and } 15\text{ V})$

35V

Internal Power Dissipation (Note 1)

Internally Limited

Operating Temperature Range (T_A) $0^\circ\text{C to } +70^\circ\text{C}$

Maximum Junction Temperature

(K Package)

150°C

(T Package)

150°C

Storage Temperature Range

 $-65^\circ\text{C to } +150^\circ\text{C}$

Lead Temperature (Soldering, 10 sec.)

TO-3 Package K

300°C

TO-220 Package T

230°C

Electrical Characteristics LM78XX (Note 2) $0^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$ unless otherwise noted.



1. INTRODUCTION

How to read/write SMS, start a modem, HSCSD or GPRS connection with Nokia phones by using AT-commands.

2. BASIC AT-COMMANDS

Before making a dial-up connection it is recommended to reset the modem in phone and also set the flow control. If you have PIN-code request on it can be typed in also with AT-command (I e Nokia Card Phone may need this when used in non Windows environment).

Below some useful AT-commands:

Check that the connection between the phone and device works	AT
<i>- should give OK for response</i>	
Reset the modem and set the factory defaults	ATZ&F
No flow control	AT&K[0]
Hardware flow control (CTS/RTS)	AT&K3
Software flow control (XON/XOFF)	AT&K4
Type in PIN-code in terminal	AT+CPIN="pppp"
<i>- where pppp is your pin code. Yes, the double quotes are needed</i>	
Check signal strength	AT+CSQ
<i>- value close to 0 means no field. Value 31.99 is maximum.</i>	
Start a data call	ATDT nnnnnnn
<i>- where nnnnnnn is the phone number.</i>	
Start a voice call	ATDT nnnnnnn;
<i>- where nnnnnnn is the phone number. Note the semicolon (;) is needed!</i>	
Hang up the call	ATH
Answer incoming call	ATA

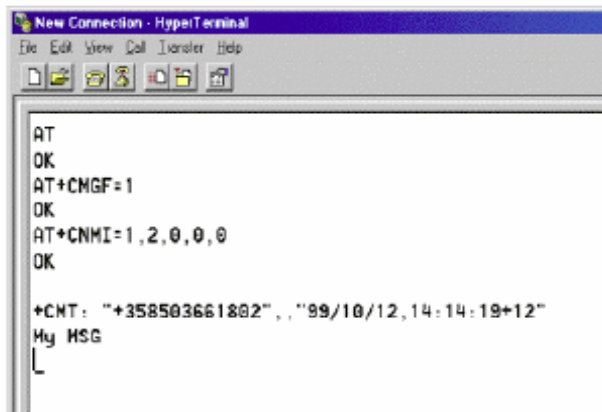
During a call you can return to AT command interpreter with command +++ (three times plus). Then you can give AT commands during a call. This is useful when calling to I e banking service which requires you to enter information with keypad (especially when using Nokia Card Phone).





RECEIVING AN SMS

1. Define the message format by typing AT+CMGF=1 (See Picture 1.).
2. Type AT+CNMI=1,2,0,0,0 in order to see all the received messages (See Picture 2.).



```
New Connection - HyperTerminal
File Edit View Call Transfer Help
[Icons]
AT
OK
AT+CMGF=1
OK
AT+CNMI=1,2,0,0,0
OK
+CMNT: "+358503661802", , "99/10/12,14:14:19+12"
My MSG
L
```

Picture 2. Receiving an SMS.

