

SATAKUNNAN AMMATTIKORKEAKOULU

Anna Haapala
OHEISLAITTEEN JA TIETOKANNAN
RAJAPINTAOHJELMOINTI

Liiketalous, matkailu, tietojenkäsittely ja viestintä, Pori
Tietojenkäsittelyn koulutusohjelma
2006

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SYMBOLI- JA TERMILUETTELO

1	JOHDANTO	7
1.1	Ohjelmiston käyttötarkoitus	7
2	OHEISLAITTEEN ESITTELY	8
2.1	Adapteri	9
2.2	Tuetut TMEX adapterit	11
3	SARJALIIKENNE	13
3.1	Liityntäteknikat	13
3.1.1	RS-232	13
3.1.2	RS485	14
3.1.3	USB	15
3.1.4	FireWire	16
3.2	Sarjaliikenteen ohjelmointi Windowsissa	17
3.3	Oheislaitteen ohjelmointi TMEX:llä	17
3.3.1	Sessiot	18
3.3.2	Tiedostokäsittely	18
3.3.3	Tiedonsiirto	18
3.3.4	Tietoverkko	19
3.3.5	Laitteistoriippuvuus	19
4	TIETOKANTA ACCESS	20
4.1	Accessin käyttöliittymä	21
5	RAPORTOINTI	22
5.1	Graafinen raportti	22
6	VISUAL BASIC	23
6.1	Visual Basicin käyttöliittymä	24
7	OHJELMAN RAKENNE	26
7.1	Piirin tunnistaminen	26
7.2	Tiedon tallentaminen	27
7.3	Graafinen käyttöliittymä	28
7.4	Raportointi ohjelmassa	32

8	TESTAUS	33
8.1	Ohjelmisto	33
8.2	Mahdollisten ongelmien selvittäminen ja niiden ratkaiseminen	34
9	LOPPUPÄÄTELMÄT JA TUTKIMUSTULOKSET	35
9.1	Tavoitteiden toteutuminen	35
9.2	Tavoitteiden onnistuminen	35
9.3	Kehitysehdotuksia	36
	LÄHDELUETTELO	37
	LIITE1	

TIIVISTELMÄ

Opinnäytetyön nimi:	Oheislaitteen ja tietokannan rajapintaohjelmointi
Tekijä:	Haapala, Anna Elina
Koulu:	Satakunnan AMK
yksikkö:	Liiketalous, matkailu, tietojenkäsittely ja viestintä, Pori
Aika:	Lokakuu, 2006
Laajuus:	40
Ohjaaja:	Juha Stenfors
Aihesanat:	Tietokanta, ohjelmointi, Visual Basic, DS18B20
UDK:	004.42

Tiivistelmä:

Opinnäytetyössä toteutettiin omaan käyttöön ohjelma, joka näyttää lämpöanturin avulla tietokoneiden lämpötilaa ja antaa käyttäjälle hälytyksen, jos lämpötila on ylittänyt tai alittanut hälytysrajat. Projekti eteni pääasiassa itsenäisenä työnä.

Ohjelma toteutettiin tekemällä koodi Visual Basic-ohjelmaa Tmex-ympäristössä hyväksikäyttäen ja tiedot tallennettiin Access-tietokantaohjelmaan.

ABSTRACT

Title of the thesis: Interface programming of peripheral device and database
Author: Haapala, Anna Elina
Institute: Satakunta University of Applied Sciences
Unit: School of Business, Tourism, Business Information Systems,
Media and Communication, Pori
Time: October, 2006
Extetion: 40
Tutor: Juha Stenfors
Keywords: Database, programming, Visual Basic, DS18B20
UDK: 004.42

Abstract:

In project I created to my personal use a program that shows server rooms temperature with thermo sensor. Program gives to user an alarm, if the thermo is above or below the alarm limits. The project progressed mainly as an independent work.

The program was created by making the code with Visual Basic-program in Tmex-environment and all the data was saved in Access-database.

SYMBOLI- JA TERMILUETTELO

1-Wire: Dallas Semicondustorin kehittämä väylä. Tähän voidaan liittää useita oheislaitteita esimerkiksi lämpömittareita.

Multi-drop: Moniajoyhteys.

Tmex: Dallas Semiconductorin 1- Wire ajuri.

Access: Microsoftin kehittämä tietokantaohjelmisto.

Visual Basic: Microsoftin kehittämä ohjelmointikieli/ohjelmointiympäristö.

1 JOHDANTO

Tässä opinnäytetyössä tehdään, tutkitaan ja testataan oheislaitteen ja tietokannan ohjelmointia. Työssä käytetään tietokoneen oheislaitteena Dallas semiconductor DS18B20-lämpöanturia. Kyseisellä anturilla tutkitaan tietokoneen(konesalin) lämpötilaa. Ainut kustannus, joka projektissa tuli oli lämpöanturin hankkimisesta.

Opinnäytetyön teoreettisena viitekehyksenä ohjelman ja tietokannan suunnittelu sekä niiden toteutus. Tällä opinnäytetyöllä ei ole asiakastilaa. Työ tulee omaan käyttöön. Anturilta tulevat tiedot tallennetaan Access-tietokantaan, josta tiedot haetaan ja näytetään Visual Basicilla tehdyllä ohjelmalla. Ohjelma toimii palveluna Windows-ympäristössä.

Kirjallisessa osassa olen kertonut teoriaa välineistä, joilla olen tehnyt ohjelman ja tietokannan.

1.1 Ohjelmiston käyttötarkoitus

Ohjelmisto suunniteltiin toimimaan palvelinhuoneen lämpötilan seurantaan. Ohjelma on suunniteltu tallentamaan tiedot tietokantaan, josta on raportointi tehty mahdollisimman yksinkertaisesti. Ohjelmassa on sisäänrakennettu hälytysjärjestelmä, joka ennelta määrättyjen raja-arvojen perusteella antaa hälytyksen tarvittaessa järjestelmä-asiantuntijalle. Ohjelman avulla voidaan seurata serverihuoneen lämpötilaa pitkällä aikavälillä sekä esimerkiksi konesalin jäähdytysjärjestelmän vikaantuessa antaa välittömästi hälytyksen järjestelmäasiantuntijalle.

2 OHEISLAITTEEN ESITTELY

Kuten jo aikaisemmin mainitsin, työssä käytetään PC:n oheislaitteena Dallas Semiconductor DS18B20-lämpöanturia. Tällä anturilla mitataan tietokoneiden lämpötilaa. Laite toimii lämpömittarina, jolla voidaan lukea lämpötiloista tietoja tietokoneelle sarjaportin kautta. Tiedot DS18B20-laitteesta menevät Access-tietokantaan.

DS18B20-lämpöanturi on kiinnitetty adapteriin, joka on ostettu jo valmiiksi koottuna. Kyseisen adapterin nimi on DS9097E. Adapteri muuttaa sarjaportin jännitteen DS18B20-lämpöanturille sopivaksi.

Kaikilla DS18B20-antureilla on oma uniikki 64-bittinen sarjanumero. Tätä sarjanumeroa sanotaan myös ROMid:ksi. Tämä sarjanumero luetaan ohjelmallisesti, jonka avulla käyttäjä tietää mistä anturista luettu tieto on peräisin. Tämän ansiosta voidaan sijoittaa useita antureita samaan 1-Wire väylään. Jos antureita on monia, niin silloin ne liitetään rinnakkain. (<http://pdfserv.maxim-ic.com/en/ds/2812.pdf>)

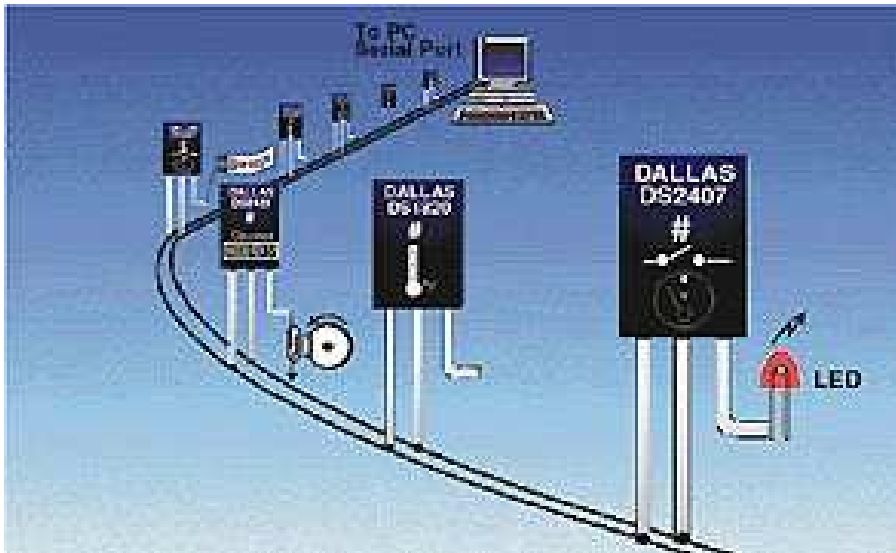
DS18B20 on 1-Wire(Dallas semiconductorin oma standardi) eli yksijohdinanturi, mutta kuitenkin siinä on kaksi johdinta. Nämä kaksi johdinta ovat maa ja datajohdin.

Tietokoneen ja oheislaitteen välinen liikenne alkaa siten, että tietokone lähettää pyynnön laitteelle, joka sitten vastaa tähän tietokoneen lähettämään pyyntöön. Mitään erillistä jännitelähdettä ei tarvita, koska tarvittava jännite tulee tietokoneelta lähetetyn pyynnön mukana. (<http://www.nic.fi/~skarna/etusivu.html>)

DS18B20-lämpöanturin ominaisuuksia ovat mm:

- $\pm 0.5^{\circ}\text{C}$ tarkkuus -10°C ja $+85^{\circ}\text{C}$ välillä.
- Mittaa lämpötiloja -55°C ja $+125^{\circ}\text{C}$ välillä.
- Käyttäjän muokattavissa oleva 9-12-bitin tarkkuus.
- 1-Wire käyttöliittymä tarvitsee kommunikointiin yhden portti
- 64-bittinen uniikki sarjanumero.

- Kääntää lämpötilan 12-bittiseen digitaalisanaan enintään 750ms.
 - Ei tarvita ulkoisia komponentteja lämpötilan mittaamiseen.
- (Dallas Semiconductor/Maxim 2005)

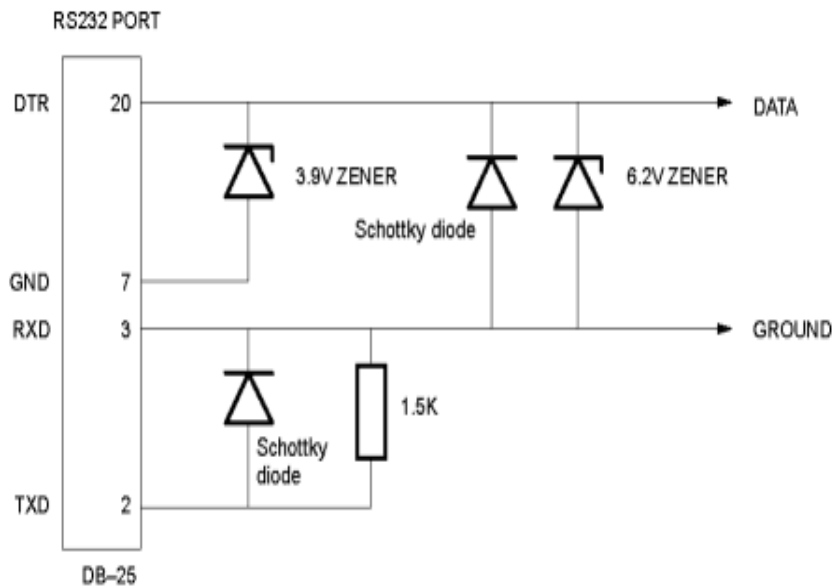


Kuva 1 1-Wire tietoverkko(<http://www.nic.fi/~skarna/etusivu.html>)

2.1 Adapteri

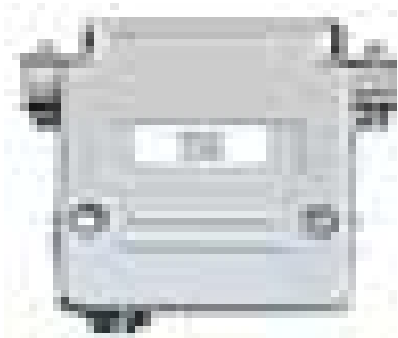
Jotta lämpötiloja voidaan tietokoneelta lukea, tarvittiin siihen adapteri, jonka avulla muunnetaan tietokoneen sarjaportin jännite anturille sopivaksi. Adapteri koostuu isäntäpiiristä, johtimista ja itse anturista. Lämpöanturi on pieni musta piiri, joka on liitetty johtimeen. Johdin puolestaan on kiinni adapterissa.

Web-sivustolla kirjoitetaan adapterin kokoamisesta näin, että ” Adapterin voi rakentaa vaikka D-liittimen sisälle: se koostuu kahdesta zener-diodista (3.9V ja 6.2V, 0.4W), kahdesta schottky-diodista (esim. SD103) ja vastuksesta (1.5kohm).” Kuva 2 nähdään adapterin kytkentäkaavio. (<http://www.nic.fi/~skarna/etusivu.html>)



Kuva 2 Adapterin kytkentäkaavio (<http://www.nic.fi/~skarna/etusivu.html>)

Työssä käytettiin valmiiksi koottua adapteria, joka liitettiin tietokoneen sarjaporttiin. Tämän adapterin nimi on DS9097E. Adapteri hankittiin suomalaisen verkkokaupan probyten kautta.



Kuva 3 DS9097E-adapteri (Dallas semiconductor/Maxim DS9097, DS9097E)

1-Wire tietoverkko pitää olla kytkettynä tietokoneeseen tuetulla adapterilla. Adaptereita voi kytkeä USB-, sarja- ja rinnakkaisporteille. Adapterin toinen pää tulee olla laitettu tietokoneeseen ja toinen pää tulee 1-Wire tietoverkkoon.

2.2 Tuetut TMEX adapterit

Adapttereita on monia ja niillä on erilaisia ominaisuuksia. Alla olevassa listassa näkyy eri adapttereita ja niiden ominaisuuksia sekä mihin windows-käyttöjärjestelmään kukin adapttereista sopii. Listalta löytyy myös työssä käytetty adapteri.

Adapter	Port	Port Type	Features	Win 95	Win 98	Win ME	Win NT	Win 2000	Win XP
DS1490F (2-in-1 Fob)	USB	6	power delivery, overdrive, led indicator, single iButton holder		X	X		X	X
DS1490X (future product)	USB	6	power delivery, overdrive, RJ-11 or iButton holder, optional EPROM write		X	X		X	X
DS1410E	Parallel	2	power delivery, overdrive, dual iButton holder, DS2401 ID	X	X	X	X	X	X
DS1410D	Parallel	2	dual iButton holder, DS2401 ID	X	X	X	X	X	X
DS9097U-009	Serial	5	power delivery, overdrive, RJ-11 connector, DS2502 ID	X	X	X	X	X	X
DS9097U-S09	Serial	5	power delivery, overdrive, RJ-11 connector	X	X	X	X	X	X
DS9097U-E25	Serial	5	power delivery, overdrive, RJ-11 connector, EPROM write	X	X	X	X	X	X
DS1411	Serial	5	power delivery, overdrive, single iButton holder	X	X	X	X	X	X
<u>DS9097E</u>	Serial	1	RJ-11 connector, EPROM write	X	X	X	X	X	X

DS9097	Serial	1	RJ-11 connector	X	X	X	X	X	X
DS1413	Serial	1	single iButton holder	X	X	X	X	X	X

Kuva 4 Adapterit

(http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/tmex8lrr.html)

3 SARJALIIKENNE

Sarjaliikenne on suosittu tapa siirtää dataa tietokoneen ja oheislaitteen välillä tai tietokoneiden välillä. Sarjaliikenne käyttää lähetintä tiedon siirtämiseen yhden bitin kerrallaan, yhden kommunikointilinjan yli vastaanottimelle. Suurin sallittu pituus riippuu käytetystä siirtonopeudesta ja tekniikasta. Useimmissa tietokoneissa on vakiona kaksi sarjaporttia(RS-232), joten ylimääräisen laitteiston hankkiminen on turhaa.(Gregory Elizabeth 2004)

3.1 Liityntäteknikat

Sarjaliikenteessä on erilaisia liitännästandardeja. Näitä ovat muun muassa RS232, RS485, Usb ja FireWire. Sarjaportti on hiljalleen poistumassa, koska tilalle ovat tulleet esimerkiksi USB ja firewire-liitännät. Niiden ohjelmointi on kuitenkin huomattavasti monimutkaisempaa.

3.1.1 RS-232

RS-232(com-portti) on yksi yleisimmistä tietokoneissa käytetyistä liitännätavoista. Tätä porttia on käytetty tietokoneissa kauan. Numerosarjan edessä oleva RS tulee sanoista Recommended Standard. RS-232-standardi on muuttunut vuosien saatossa. Standardeja ovat muun muassa RS-232A, RS-232B ja RS-232C. Olennainen merkitys standardien kehityksessä on ollut jännitteen muutos pienemmäksi.(Mustalahti 2004)

RS-232 on esimerkiksi tietokoneen ja oheislaitteen kuten tässä tapauksessa DS18B20-lämpöanturin väliseen tietoliikenteeseen tarkoitettu tietoliikenneportti. RS232 soveltuu kahden pisteen väliseen liikennöintiin matalilla nopeuksilla. Näin yhtä porttia vastaa yksi laite. RS232:ssa kaapelin pituus on enintään 30-60 m. Tässä standardissa data siirtyy yksi bitti kerrallaan sarjamuotoisena peräkkäin. Kun 8-bittinen rinnakkaismuotoinen data käännetään sarjamuotoiseksi, hävitään nopeudessa ainakin 8:1. Laitteet, jotka käyt-

tävät RS-232-porttia voittavat yksinkertaisuudessaan laitteet, jotka käyttävät muita portteja. (Wikipedia-vapaa tietosanakirja RS232)

3.1.2 RS485

RS485 on EIA:n eli Electronics Industries Associationin laatima sarjaliikennestandardi. RS485-sarjaliitäntä ei ole niin tunnettu kuin aikaisemmin mainittu RS232-sarjaliikennestandardi. RS485 suunniteltiin alunperin teollisuuden ohjausjärjestelmien datansiirtoon. RS485:ttä käytetään useamman laitteen väliseen liikennöintiin. Eli monia laitteita voidaan liittää yhteen signaalijohtimeen. RS485-järjestelmät käyttävät useinmielten isäntä/orja-suhdetta. Tässä jokaisella orjalla on oma osoite, josta sen tunnistaa. Orja vastaa vain omaan osoitteeseensa osoitetuihin paketteihin. Näitä paketteja isäntä lähettää. Isäntänä voi olla esimerkiksi tietokone. Isäntä aloittaa aina kommunikoinnin laitteiden välillä. (Laamanen 2000)

Suurin sallittu tiedonsiirtonopeus RS485-standardin lyhyillä etäisyyksillä on 10 megabittiä sekunnissa. Kuitenkin käytännössä voidaan päästä jopa 30 megabittinä sekunnissa. RS485-väylä ei säteile liiemmin häiriökohinaa ympäristöön, koska väylässä käytetään matalia signaalitasoja. Tietokoneympäristössä RS485-standardia käytetään muun muassa SCSI-liitännöissä.

RS485:llä on monia sovellusmahdollisuuksia. Yleisempiä ovat erilaiset tiedonkeruujärjestelmät. Tällöin RS485-verkossa toimii isäntänä tietokone, joka hakee dataa anturilta ja ohjaa järjestelmää saamansa vastauksen mukaan. Alla olevassa kuva 5 on vertailtu RS232- ja RS485-standardia. (Laamanen 2000)

	RS232	RS422	RS485
Toimintatapa	Yksipäinen signaali, balansoimaton	Differentiaalinen signaali, balansoitu	
Laitteiden lukumäärä	1 lähetin, 1 vastaanotin	1 lähetin, 10 vastaanotinta	32 lähettä, 32 vastaanotinta

Tiedonsiirtotapa	Full duplex	Full duplex, Half duplex	Half duplex
Maks. etäisyys	15 m nopeudella 19,2 kbps	1200 m nopeudella 100 kbps	
Maks. tiedonsiirtonopeus (matka 15 m)	19.2 kbps	10 Mbps	
Space (data 0)	5 V min. 15 V max.	2 V min. (A>B) 6 V max. (A>B)	1.5 V min. (A>B) 5 V max. (A>B)
Mark (data 1)	-5 V min. -15 V max.	2 V min. (B>A) 6 V max. (B>A)	1.5 V min. (B>A) 5 V max. (B>A)
Tulovastus (kilo-ohmia)	3-7	4	12
Jännitetaso min.	± 3 V	0,2 V ero	
Protokolla	Ei määritelty		

Kuva 5 RS232 ja RS485 vertailu(Laamanen 2000)

3.1.3 USB

USB muodostuu sanoista Universal Serial Bus. USB-portteja on melkein kaikissa pc- ja Macintosh-koneissa. USB on sarjaväylästandardi, jonka avulla voidaan liittää oheislaitteita tietokoneeseen. Laitteet kommunikoivat keskenään. USB:n tarkoituksena on helpottaa oheislaitteiden liittämistä. USB-tyyppejä ovat USB-A(isäntä), USB-B(orja), USB-OTG ja WUSB. (Wikipedia-vapaa tietosanakirja USB , Vähimaa 2006, s. 58)

USB:ssä on luokiteltu kolme nopeusluokkaa. Nämä ovat Low Speed, jonka tiedonsiirtonopeus on 1.5 megabittia sekunnissa; Full Speed, jonka tiedonsiirtonopeus on 12 megabittia sekunnissa sekä Hi-Speed, joka tiedonsiirtonopeus on 480 megabittia sekunnissa.(Afterdawn.com)

USB-laitteille on yksi yhteinen liittintyyppi. Oheislaitteen voi asentaa tietokoneelle koska tahansa. Koneita ei tarvitse uudelleenkäynnistää. USB:n maximi nopeus 2.0:ssa on 480 megabittiä sekunnissa ja 1.1:ssa nopeus on 12 megabittiä sekunnissa. Datasignaalit muodostavat USB-kaapelissa pareja. Toisella näistä on vastakkainen suunta. Käänteisvaiheinen signaali käännetään takaisin vastaanottopäässä ja signaali summataan. Näin matkalla syntynyt häiriö kumotaan. (Wikipedia-vapaa tietosanakirja USB)

3.1.4 FireWire

Tietokonevalmistaja Apple suosii koneissaan FireWireä. FireWire tunnetaan myös nimillä IEEE 1394 ja i.Link. FireWire on sarjamuotoinen väylä. Tämän kautta laitteet kommunikoivat keskenään. FireWiren perusversio on nopeudeltaan enimmillään 400 megabittiä. On myös olemassa nopeampi versio nimeltä IEEE1394b, jonka tiedonsiirtonopeus on 800 megabittiä sekunnissa. (Vähimaa 2006, s. 58. Afterdawn.com)

FireWiressä käytetään differentiaalitekniikkaa. Kaksi johdinta kierretään yhteen ja näiden johtimien jännitteiden erotuksella data tulkitaan. Tämän ansiosta päästään eroon tasajännitevirheestä. FireWiressä on pareja kaksi. Toinen on lähtevälle datalle ja toinen saapuvalla datalle. (Kangas 2002)

	usb 1.1	usb 2.0	IEEE-1394 (FireWire)	IEEE-1394b
Maksiminopeus	12 Mb/s	480 Mb/s	400 Mb/s	3200 Mb/s
Laitteiden maksimimäärä	127		63	
Kaapelien maksimipituudet	3 - 5 m nopeudesta riippuen		4,5 - 11 m nopeudesta riippuen, 70 m ja yli valokuidulla tai CAT-5:llä	
Jännite- ja virta-annot	5 v, 100 mA tai 500 mA		8 - 30 V, 1,5 A	

Kuva 6 USB:n ja FireWire eroja (Kangas 2002)

3.2 Sarjaliikenteen ohjelmointi Windowsissa

Windows 9x, NT, 2000, XP työskentely laitteiston kanssa toimii erilalla kuin DOS:ssa. Windowsissa laitteiston kanssa toimiminen on monimutkaisempaa kuin DOS:ssa.

DOS-ajuri voidaan luoda asm-ohjelmointikielen avulla ja sillä on suora yhteys portteihin. Johtuen Windowssin moniajojärjestelmästä on mahdotonta sallia jokaiselle ohjelmalle lupa muuttaa laitteiston asetuksia, koska jos yksi ohjelma on muuttanut laitteistoasetuksia, niin toinen ohjelma ei välttämättä ymmärrä vaihdettuja asetuksia.

Ohjelmoitsijoita varten on luotu API(application programming interface). Tämä käyttöliittymä sallii ohjelmoijan käyttää ohjelmissa Windows-järjestelmäpalveluita.

Windows Driver Developer Kit (DDK) käytetään ajureiden luomiseen. Kaikille Windowsille on olemassa oma versionsa DDK:sta. DDK on Windowsin ajureiden kehityspaketti. (fCoder Group International 1998-2002)

3.3 Oheislaitteen ohjelmointi TMEX:llä

SDK(Software Developer's Kit) on kehityspaketti. Sen avulla kommunikoidaan esimerkiksi ohjelman ja oheislaitteen välillä. Windowsille soveltuva SDK on jakautunut kahteen sovellukseen. Näiden nimet ovat OWCOM API ja TMEX API. Työssä on käytetty TMEX APIa, koska se soveltui tähän työhön paremmin. Jotta TMEX toimii, täytyy tietokoneeseen asentaa 1-Wire-ajurit. (Dallas Semiconductor Tmex API Overview)

TMEX on ohjelmointipaketti, joka kykenee kommunikoimaan useamman 1-Wire-laitteiden kanssa. Lämpötila-anturi tunnistetaan sen sarjanumeron perusteella.

TMEX API on jaettu viiteen eri kerrokseen. Nämä kerrokset ovat Sessiot(Session layer), tiedostonkäsittely(File Operations layer), tiedonsiirto(Transport layer), tietoverkko(Network layer) ja laitteistoriippuvainen(Hardware Specific).
(Dallas Semiconductor Tmex API Overview)

Laitteistoriippuvainen-kerros sisältää ainoastaan API kutsuja, jotka ovat laitteistoriippuvaisia. Sessio-kerros antaa 1-Wire tietoverkon sovittautua moniajo-ympäristöön. Tietoverkko-kerros esittelee API:n toiminnot, jotka lukevat ja varmistavat oheislaitteen uniikin sarjanumeron 1-Wire-tietoverkosta. Tiedonsiirto-kerroksella on tietotoiminnot kuten data pakettien lukeminen ja kirjoittaminen. Tiedonkäsittely-kerros sisältää API-kutsut, jotka ovat tiedostojen luku ja kirjoittaminen sekä kansioden käyttö.

(Dallas Semiconductor Tmex API Overview)

3.3.1 Sessiot

TMEX Session API kutsut kontrolloivat pääsyä 1-Wire tietoverkkoon. Jos tietty 1-Wire tietoverkko on saatavilla, niin TMExtendedStartSession palauttaa arvon **session handle**, joka on muu kuin nolla. Tämä **session_handle** on käytössä kaikissa TMEX API kutsuissa. Jotta API kutsut toimivat, pitää TMExtendedStartSession palauttama **session handle** olla oikea. (Dallas Semiconductor Tmex Session API Overview)

3.3.2 Tiedostokäsittely

Tiedostokäsittely-kerroksen toiminnot suorittavat tiedostojen ja kansioden käsittelyä. 8-bittinen puskuri halutulta 1-Wire laitteelta pitää olla varattuna muistista ennenkuin mitään tiedostokäsittelyä voidaan suorittaa. Tämä voidaan tehdä kirjoittamalla suoraan puskuuriin käyttämällä TMROM API-kutsua tai käyttämällä yhtä näistä seuraavista tietoverkko toiminnoista: TMFirst, TMNext, TMFirstAlarm tai TMNextAlarm. Tämä toiminto sallii kaikkien näiden API-kutsujen olla monipisteyhteyteen (Multi-drop) sopiva. Jos 1-Wire tietoverkkoon on liitetty monta laitetta, niin joka tapauksessa tämä monipisteyhteys yhteensopivuus viittaa onnistuneeseen toimintaan.

(Dallas Semiconductor Tmex File Operations API Overview)

3.3.3 Tiedonsiirto

TMEX Transport API kutsuilla on mahdollisuus sallia ryhmätaso tiedon siirtämiseen 1-Wire-laitteeseen. Tämä taso sisältää **TMReadPacket** ja **TMWritePacket**. Nämä luke-

vat ja kirjoittavat CRC-16 varmistettuja tietoryhmiä. Jos 1-Wire-laitteella tehdään tavallinen tiedosto toiminto, niin silloin ei ole mitään tarvetta kutsua Transport API toimintojä. (Dallas Semiconductor Tmex Transport API Overview)

3.3.4 Tietoverkko

TMEX Network API kutsut tukevat 1-Wire-protokollan monipisteyhteyden pystyvyyttä. Kutsut sallivat kutsuvan ohjelman paikantaa ja kommunikoida yksitellen monista 1-Wire-laitteiden kanssa, jotka on liitetty rinnakkain. (Dallas Semiconductor Tmex Network API Overview)

3.3.5 Laitteistoriippuvuus

TMEX Hardware Specific API-kerroksen toiminnot sallivat alimman tason kommunikoinnin 1-Wire-laitteille. Kutsuista **TMsetup** ja **TMClose** tekevät poikkeuksen. Näitä ei tarvitse kutsua, kun käytetään tiedostosuuntautunutta TMEX-toimintoja. TMEXin monimutkaisemmat tasot kutsuvat tätä kerrosta kaikkiin operaatioihin. (Dallas Semiconductor Tmex Hardware Specific API Overview)

4 TIETOKANTA ACCESS

Relaatiotietokantaohjelmaa (relational database program) käytetään relaatiotietokannan toteutukseen ja sen toimintaan. Ohjelmaa sanotaan myös relaatiotietokannan hallintajärjestelmäksi (relational database management system). Relatiotietokanta tarkoittaa, että siinä on kokoelma toisiinsa liittyviä tietoja. Tämä kokoelma on järjestetty. Relatiotietokanta muodostuu useista tauluista, joihin tiedot tallennetaan. Tietokannassa yksi tieto tallennetaan yhteen paikkaan.

Relaatiotietokannan positiivisina ominaisuuksia voidaan pitää muun muassa tietojen helppoa saatavuutta ja muutosjoustavuutta. Muutosjoustavuudella tarkoitetaan arvojen syöttämistä valmiiseen tietokantaan sitä liioja häiritsemättä. (Sainio, 2001, s. 6)

Relaatiomalliksi sanotaan sääntöjä, jotka koskettavat esimerkiksi Access-tietokantaa. Nämä säännöt ovat: tietokantaan tallennetaan tietoja sekä näitä tietoja hyväksikäytetään. (Sainio 2002, s. 6)

Microsoft Accessia kutsutaan relaatiotietokantaohjelmistoksi. Tätä ohjelmaa kutustaan myös tietokantojen hallintajärjestelmäksi. Koska Access kuuluu Microsoft Office pakettiin, niin Access soveltuu käytettäväksi myös muiden Microsoft Officen ohjelmien kanssa. Accessilla voidaan esimerkiksi siirtää dataa muihin Officen ohjelmiin kuten esimerkiksi Exceliin. Excelillä voidaan lukea esimerkiksi tietokannan tauluja. (Sainio 2002, s. 6-8)

Accessin sanotaan olevan helppokäyttöinen. Ohjelmoinnin kannalta Access on hyväksi huomioitu tietokantasovelluksen työväline. Accessissa on mahdollisuus käyttää erilaisia ohjattuja toimintoja. Näitä kutsutaan velhoiksi eli Wizard-toimintoja. Näillä ohjatuilla toiminnoilla voi esimerkiksi tehdä lomakkeita tai raportteja. Access-ohjelmassa on selkeät ohjeet kuinka sitä voi käyttää. (Callahan 2000. s. 13)

4.1 Accessin käyttöliittymä

Tietokannan luonti Accesissa ei ole vaikeaa. Kun Access aukaistaan, tulee esiin tyhjä sovellus. Kun ruvetaan luomaan tietokantaa, täytyy tietokanta ensin nimetä ja tallennetaan haluamaansa paikkaan. Access sisältää monia eri näkymiä, joilla voidaan tutkia tehtyä tietokantaa. Näitä näkymiä ovat muun muassa rakenne- ja taulukkonäkymä.

Tärkein asia tietokannassa on tiedot. Tietokannassa tiedot talletetaan tauluihin. Tauluja voidaan sanoa myös taulukoiksi. Taulussa on yhteen aiheeseen liittyviä tietoja. Taulussa olevaa yhtä riviä sanotaan tietueksi, jossa on yhden kohteen tiedot. Taulussa on myös sarakkeita, joita kutsutaan kentiksi. Soluksi sanotaan sarakkeen ja rivin risteyskohtaa. Tähän soluun tallennetaan yksi tiedon arvo. Solun arvoksi voi tulla tyhjä-arvo eli Null-arvo. (Sainio 2002. s. 6 ja 7)

Jokaiseen tauluun, jotka relaatiotietokanta sisältää, tulisi määritellä perusavain(Primary Key). Perusavaimen sisältämä tieto yksilöi rivin. Viiteavain(Foreign Key) viittaa toisen taulun perusavaimeen. Viiteavain on yksi sarake. Tämä viittaa toisen taulun perusavaimeen. Periaatteessa viiteavain toimii ns. linkkinä toiseen tauluun. (Sainio 2002. s. 6 ja 7)

Yhteyden eli Relationship avulla yhdistetään taulujen tiedot toisiinsa. Yhteys muodostetaan kahden eri taulun kenttien välille. Nämä kentät sisältävät samaa tietoa. Yhteyksiä on kolme erilaista. Nämä ovat yksi-moneen-yhteys(1:n), yksi-yhteen-yhteys(1:1) ja monta-moneen-yhteys(N:N). (Sainio 2002. s. 6 ja 7)

Raportit ovat tarkoitettu tiedon tulostamista paperille. Raportille voidaan laittaa haluamiansa tietoja. Raporttiin voidaan laittaa tietoja yhdestä tai useammasta taulusta. Ohjattujen toimintojen avulla voidaan tehdä raportteja ja lomakkeita. Ohjattujen toimintojen vuoksi ohjelmaa voisi sanoa helppokäyttöiseksi.

(Wikipedia-vapaa tietosanakirja 2006 Microsoft Access)

5 RAPORTTOINTI

Raporttijärjestelmä luotiin sitä varten, että ohjelman avulla voitaisiin katsella tai tulostaa eri hakutekijoilla tehtyjä raportteja. Visual Basicin raportointi tehdään data report-objektilla. (Rahmel 1999. s. 164)

Data reportin oletusnäkyvä on esikatselu ikkuna, jossa on tulostustoiminto eli tulostaa varsinaisen raportin paperille. (Rahmel 1999. s. 164)

Data report-objekti on tehokas esimerkiksi graafisen raportin rakentamisessa, automaattisessa esikatselussa, liittynässä VB-koodiin raportin tekovaiheessa ja HTML-raportin luomisessa. Data report-objekti luodaan helpokäyttöisesti lomakkeen tapaisessa Visual Basic-ympäristössä. (Rahmel 1999. s. 164)

5.1 Graafinen raportti

Graafisessa muodossa oleva raportti on helppo tulkita/analysoida. MSChart-ohjaimen avulla voidaan helposti tehdä/näyttää tietoja graafisessa muodossa. Liityntä tietokantaan on helppo tehdä, koska MSChart käyttää data-ohjainta. Ohjaimen avulla voidaan tietoja näyttää useilla eri tavoilla. Näistä monet voidaan valita kaksi- tai kolmiulotteisina. Kaaviotyyppejä, jotka kuuluvat ohjaimen, on useita. Näitä ovat muun muassa pylväs-kaaviot ja viivakaaviot. (Rahmel 1999. s. 179-180)

Kun kaavio laitetaan suoraan kiinni Data Environment-objektin command-objektiin, saadaan näin helpoiten suoritettua kaavion täyttäminen tiedoilla. Command-objekti on asennettu näyttämään vain tarpeelliset tiedot. (Rahmel 1999. s. 179-180)

6 VISUAL BASIC

Microsoft on kehittänyt helppokäyttöisen Visual Basic-ohjelmointikielen. Visual Basic on kehitetty 15 vuotta sitten. Tänä päivänä Visual Basic kuuluu .NET-perheeseen Visual Basic.Net. Visual Basic soveltuu vallan mainiosti tietokantaohjelmointiin.

(Wikipedia-vapaa tietosanakirja 2006 Visual Basic)

Aiemmin ohjelmat, jotka ovat ohjelmoitu Visual Basicilla olivat laitteistoriippuvaisia. Tästä johtuen sillä tehdyt ohjelmat toimivat ainoastaan Windows-käyttöjärjestelmissä. Tänä päivänä Visual Basicilla tehtyjä ohjelmia voidaan ajaa esimerkiksi Linuxissa. Tällöin käytetään jotain Windows-emulaattoria. Emulaattori mahdollistaa ohjelman toisenlaisen käytön. (Wikipedia-vapaa tietosanakirja 2006 Visual Basic)

Visual Basicin syntaksi on hyvin lähellä Basic-kieltä. Visual Basic on laajennus Basic-kieleen. Tämä tarkoittaa sitä, että pystytään rakentamaan graafisen käyttöliittymän visuaalisilla välineillä. Visual Basicissa on käytössä monesta ohjelmointikielestä tutut if-, for- ja while-lauseet, mutta lauseet eivät vaadi mitään ylimääräisiä merkkejä kuten C#:ssa puolipiste. (Wikipedia-vapaa tietosanakirja 2006)

Visual Basic versio 6 ei ole varsinaisesti oliokieli, mutta siinä voidaan simuloida olioita. Ohjelmointikieltä on yritetty pitää helppona, josta johtuen siitä puuttuu korkeiden tason ohjelmointikielten ominaisuuksia. Korkean tason ohjelmointikieliä on muun muassa C-kieli. (Wikipedia-vapaa tietosanakirja 2006)

Visual Basicissa on laaja työkaluvalikko, joka tulee kääntäjän mukana. Tämän ansiosta ohjelmaa voidaan tulkita, jotta koodin mahdolliset virheet löydetään ja voidaan niiden löydyttyä korjata. (Wikipedia-vapaa tietosanakirja 2006)

Ohjelmointi esimerkkejä:

Visual Basicilla tehdään kommentteja ' -merkillä. Esimerkiksi:

'Tällainen on kommentti (Heinonen 2005)

Muuttujia, esimerkiksi Byte, Integer ja Date määritellään näin:

```
' Asetetaan muuttujat
Dim Bitti As Byte
Dim Paiva As Date (Heinonen 2005)
```

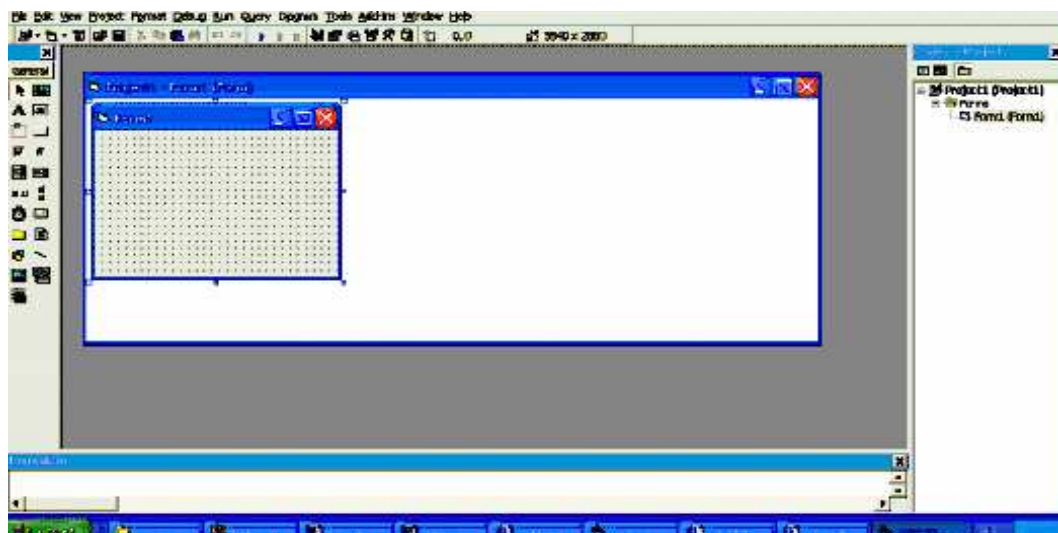
Funktioita, joilla kommunikoidaan käyttäjän kanssa:

```
'Kysytään käyttäjältä onko kaikki ok
MsgBox (Onko kaikki OK?) (Heinonen 2005)
```

```
'Kysytään käyttäjältä tietoja
Sub syotto()
Dim syote As String
Syote=InputBox("Anna nimesi")
End Sub (Heinonen 2005)
```

6.1 Visual Basicin käyttöliittymä

Ohjelma on projekti. Siihen voi kuulua monia tiedostoja. Lomake on yleisin tiedosto projektissa. Lomake vastaa yhtä ikkunaa valmiissa ohjelmassa. Alla oleva kuva 7 esittää Visual Basic-projektin aloitusnäkyä. Lomakkeeseen kuuluu myös lomaketta koskeva lähdekoodi, joka reagoi käyttäjän toimenpiteisiin. (Laaksonen Antti, Eronen Lassi 2002.)



Kuva 7 Visual Basicin aloitusnäkyä

Lomakkeeseen voidaan sijoittaa erilaisia objekteja kuten esimerkiksi painike(command button) ja tekstikenttä(textbox). Lomake kerää ja näyttää tietoja käyttäjän ja tietokoneen välillä(käyttöliittymä). (Laaksonen Antti, Eronen Lassi 2002)

Koodeissa voidaan käyttää muuttujia, joihin voidaan tallettaa tietoja. Jokaiselle muuttujalle määritellään tietotyyppi. Tästä ilmenee minkälaista tietoa muuttujaan talletetaan. (Laaksonen Antti, Eronen Lassi 2002. Tietotekniikka liitto 2006)

Visual Basicin tiedostopäätteitä ovat vbp, frm, frx, bas ja cls. Seuraavassa taulukossa(taulukko1) selitetään tarkemmin päätteistä. (Laaksonen Antti, Eronen Lassi 2002)

Taulukko 1. Visual Basicin tiedostopäätteitä
(http://www.ohjelmointiputka.net/opas.php?tunnus=vbo_1)

pääte	selitys
vbp	projektitiedosto, joka sisältää kaikkien muiden projektiin kuuluvien tiedostojen nimet
frm	ikkunaa vastaava lomake, jossa on ohjaimet ja niihin liittyvät ominaisuudet sekä koodi
frx	sisältää lomakkeeseen liitetyn grafiikan (tästä ei tarvitse huolehtia itse)
bas	moduuli, jossa on yleisiä ohjelman osia; voi myös olla itsenäinen ohjelma
cls	luokka, joka vastaa pääosin moduulia, mutta on olio-ohjelmointia

7 OHJELMAN RAKENNE

Lähdekoodi kirjoitettiin Visual Basic-ohjelmalla. Ohjelma ei pysty hakemaan oheislaitteelta tietoja, jos TMEX-ajureita ei ole asennettu.

Ohjelman tarkoitus on hakea ja näyttää tietokoneen lämpötila kyseisellä hetkellä. Ohjelma sisältää myös hälytysrajat, jotka hälyttävät, jos lämpötila menee yli tai ali hälytysrajojen. Ohjelmassa on myös raportointiosuus, josta onnistuu myös tulostus paperille.

Valitsin työssä käytettäväksi Dallas semiconductor DS18B20-anturin lukuun kyseisen anturin valmistajan tekemää ajuripakettia TMEX, joka toimii Windows-ympäristössä. TMEX-ajurit täytyy asentaa ennen kuin kehitysympäristö toimii.

7.1 Piirin tunnistaminen

Tietokoneeseen asennettu oheislaitte tunnistetaan kuten aikaisemmin olen maininnut, oman sarjanumeronsa mukaan. Ohjelmassa tarkistetaan oheislaitteen sarjanumero(ROMid). Alla oleva koodi tarkistaa oheislaitteen tyyppin.

```
//tarkistetaan onko piiri oikeaa tyyppiä
Function FindFirstFamily(family, Session_Handle)
Dim romstr As String
Dim rom(9) As Integer
romstr = " "
If ((family And &H7F) = (rom(0) And &H7F)) Then

T = 7
Do While T > 1
romstr = romstr + Hex(rom(T))
T = T - 1
Loop

Form1.Label6 = "ROMid: " & romstr
End If
```

7.2 Tiedon tallentaminen

Ohjelma hakee DS18B20-lämpöanturilta tiedot, jotka tallennetaan tietokantaan. Tietokannan toteutuksessa olen käyttänyt Microsoft Access-tietokantaohjelmaa. Tässä työssä Access on yhdistetty Visual Basic-ohjelmaan. Oli luonnollista valita Access-tietokantaohjelma ja Visual Basic, koska nämä ovat jo ennestään tuttuja välineitä.

Tauluja suunniteltaessa oli mielessä, että tietokantaan tulee sijoittaa vain kaikkein tähdellisimmät tiedot. Tietokantaa suunnitellessa mietin mikä on sen tarkoitus ja millainen tietokannan sisältö tulee olla.

Tauluja luotiin kaksi kappaletta. Taulut ovat lt ja tunnus. Lt-taulu sisältää kentät lampotila, yläraja, alaraja, pvm ja tunnus. Lampotila on perusavain. Toisen taulun eli tunnus-taulun kentät ovat tunnus ja lampotila. Perusavain on tunnus-kenttä.

Tiedot tallennetaan tietokantaan käyttäen ActiveX Data Objekteja(ADO). Ado toimii liittymänä Visual Basicin ja Access-tietokannan välillä.

Alla olevassa koodissa näkyy kuinka ohjelman tiedot tallentuvat tietokantaan. Aluksi on määritelty yhteysasetukset eli tietokantaan on otettu yhteyttä Jet-moottorin avulla. Tämän jälkeen avataan taulu. Ensin aukaistaan lt- eli lämpötila-taululle, johon sijoitetaan lampotila-kentän, alaraja-kentän, yläraja-kentän, pvm-kentän ja tunnus-kentän tiedot. Pvm-kenttään tulee koko päivämäärä(pp.kk.yyyy hh.mm). Päivittämällä lt-taulu tiedot tallentuvat tietokantaan. Tämän jälkeen yhteys tietokantaan suljetaan. Samankaltainen operaatio tehdään myös tunnus-taululle.

```
'otetaan yhteys tietokantaan
Dim myRS As New ADODB.Recordset
'asetetaan muuttujaan nyt päiväys ja kellon aika
Dim nyt As Date

nyt = Now
'asetetaan yhteysasetukset
Const ConnectStr = "provider=Microsoft.Jet.OLEDB.3.51; Data Source = C:\tk\TK1.mdb;"
Set myRS = New ADODB.Recordset
myRS.ActiveConnection = ConnectStr
```

```

myRS.CursorType = adOpenDynamic
myRS.LockType = adLockOptimistic

'avataan It-taulu
myRS.Open "It", ConnectStr, , , adCmdTable
With myRS
'tallennetaan tiedot lämpötila-,ylaraja-,alaraja-,pvm- ja tunnus-kenttiin
    .AddNew
    ![lampotila] = tmp
    ![ylaraja] = ylaraja
    ![alaraja] = alaraja
    ![pvm] = nyt
    ![tunnus] = romid
'päivitetään tietokanta, jolloin tiedot tallentuvat
    .Update
End With
myRS.Close

myRS.Open "tunnus", ConnectStr, , , adCmdTable
With myRS
    .AddNew
    ![tunnus] = romid
    ![lampotila] = tmp
    .Update
End With
myRS.Close

End Sub

```

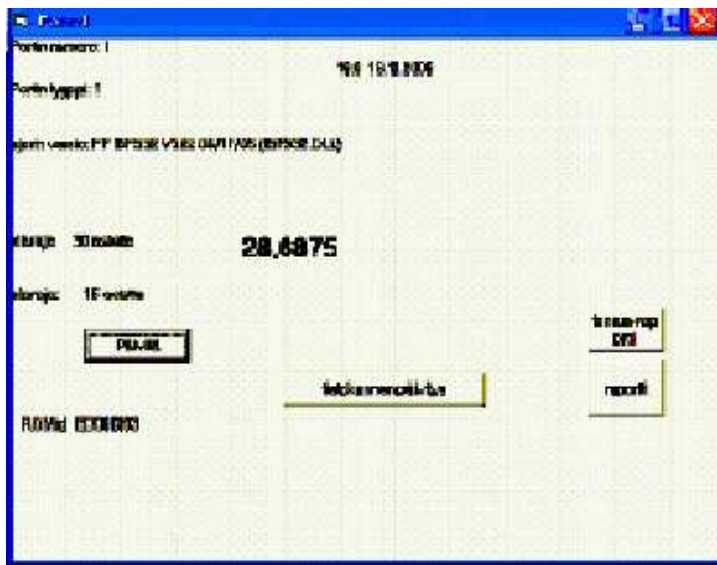
Tiedot kulkeutuvat tietokantaan lomakkeella olevan painikkeen kautta. Kun käyttäjä on asettanut lämpötilan hälytysrajat, täytyy käyttäjän painaa kysesitä painiketta, jotta tiedot tallettavat tietokantaan.

7.3 Graafinen käyttöliittymä

Ohjelman käyttöliittymä on yksinkertainen ja helppo käyttää. Käyttöliittymän ulkonäköasioihin ei ole panostettu paljoakaan, koska ohjelma tuli omaan käyttöön. Lähdekoodin toimintaan saaminen on ollut tärkein asia projektissa.

Ohjelmassa luodaan lomake näyttämään lämpötila, päivämäärä, ajurin versio, aika, portin numeron, portin tyyppin, ROMid(laitteen oma sarjanumero) ja lämpötilan hälytysrajat. Lomakkeeseen on myös sijoitettu painikkeita raporteja varten.

Kuvassa 8 näkyy miten käyttöliittymään on sijoitettu painikkeet, päivämäärä ja aika sekä oheislaitteelta haettavat tiedot.



Kuva 8 Käyttöliittymä

Alla näkyvä koodi kertoo kuinka lomakeella näytetään tietoja.

```
// näytä form1
Form1.Show
Form1.Label4 = tunnit & ":" & minuutit & " " & paiva & "." & kuukausi & "." & vuosi

Form1.Label9 = ylaraja & " astetta"
Form1.Label10 = alaraja & " astetta"

// kirjoitetaan portin tiedot formille
Form1.Label11 = "Portin numero: " & PortNum
Form1.Label12 = "Portin tyyppi: " & PortType

//haetaan ajurin versio
Get_Version (tstr)
Form1.Label13 = "ajurin versio: " & tstr

//tarkastetaan saatiinko ajurin versio
If (TMGetTypeVersion(PortType, tstr) < 0) Then
MsgBox "ei ajuria"

End If
```

```
Form1.Label6 = "ROMid: " & romstr
End If
```

Ohjelman lähdekoodiin on laitettu mukaan lämpötilan ylä- ja alahälytysrajat. Kun ohjelman aukaisee, pyydetään käyttäjää antamaan lämpötilan ylä- ja alarajat. Käyttäjä kirjoittaa pyydettyt tiedot, jonka jälkeen ohjelma aukaisee ”formin” eli lomakkeen, johon tulee aikaisemmin mainitut tiedot näkyviin. Käyttäjän antamat arvot asetetaan muuttujiin yläraja ja alaraja. Näin ei tarvitse joka kerta antaa uusia arvoja.

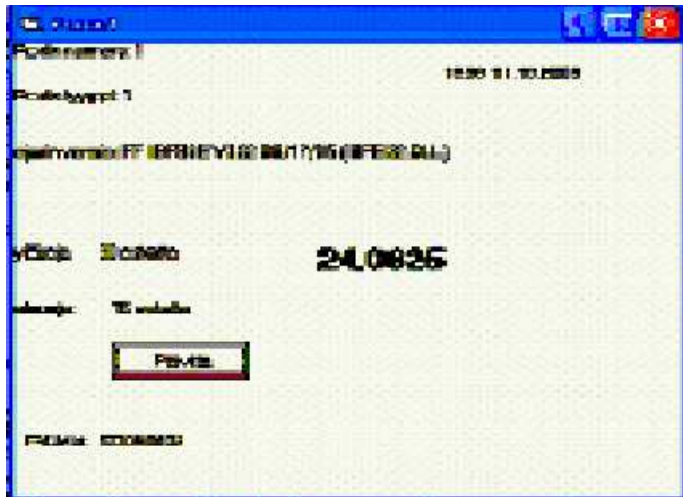
```
// näytä form1
Form1.Show
If ylaraja = 0 Then
ylaraja = InputBox("Kirjoita ylähälytysraja esim 30:", "Asetukset")
End If
If alaraja = 0 Then
alaraja = InputBox("Kirjoita alahälytysraja esim 15:", "Asetukset")
End If

If tmp < alaraja Then
  MsgBox "HÄLYTYYS ALARAJA"
Else

  If tmp > ylaraja Then
    MsgBox "HÄLY YLÄRAJA"

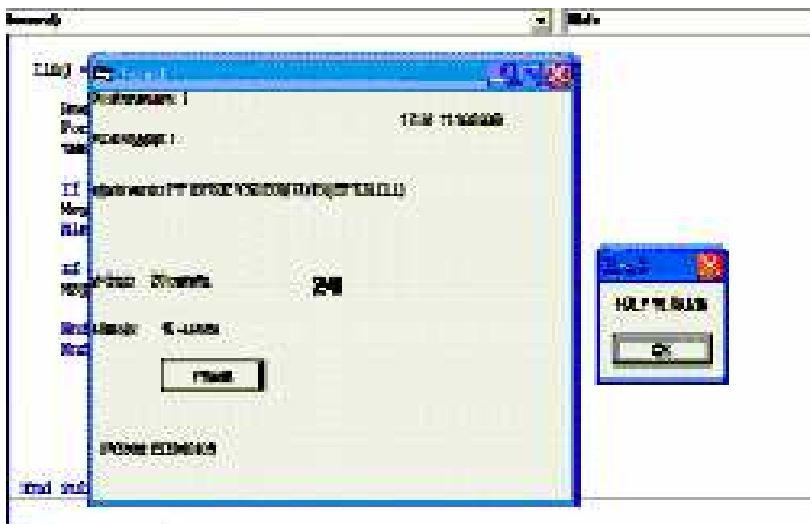
  End If
End If
```

Kuvassa 9 näkyy esimerkki, jos tietokoneen lämpötila on käyttäjän asettamien hälytysrajojen sisäpuolella. Kuvassa ylärajalle on annettu 30 ja alarajalle 15 ja tietokoneen lämpötila on 24,0625 astetta, niin tällöin ei tule mitään hälytyksiä käyttäjälle.



Kuva 9 Esimerkki ei hälytyksiä.

Mutta, jos, kuten alla olevassa kuvassa 10 näkyy, tietokoneen lämpötila ylittää asetetun ylähälytysraja, joten ohjelma ilmoittaa käyttäjälle ”messageboxilla” ”Hälytys yläraja”.



Kuva 10 Esimerkki lämpötilan ylärajan ylityksestä

Jos taas alaraja alitetaan, niin silloin ”messagebox” ilmoittaa käyttäjälle, että ”Hälytys alaraja”. Alla olevassa kuvassa 11 lämpötilan ylärajaksi asetettiin 30 ja alarajaksi 25. Koneen silloinen lämpötila oli 24,4 astetta, joten tuli hälytys alarajan alituksesta.



Kuva 11 Esimerkki lämpötilan alarajan alittamisesta

7.4 Raportointi ohjelmassa

Ohjelmaan lisättiin muutama painike, joita painamalla käyttäjä saa esiin eri tekijöillä tehtyjä raportteja. Näitä raportteja käyttäjä voi halutessaan vaikka tulostaa. Raportit ovat yksinkertaisia. Niitä voi halutessaan lisätä ja tehdä itse enemmän niin halutessaan.

loppo	alku	määrä	paino	laajuus
20				
20				
20				
20				
20		41		
20		33		
20	1	88		
20	20	183		
20	1	44		
20	20	28		
20	10	28		
20	10	36		
20	17	41	16.10.2008 15:01:00	ESCR0003
20	17	41	16.10.2008 15:41:01	ESCR0003
20	17	28	16.10.2008 16:41:07	ESCR0003
20	18	20	16.10.2008 17:21:10	ESCR0003
20	18	28	16.10.2008 17:21:30	ESCR0003
20	18	55	16.10.2008 17:21:33	ESCR0003
20	18	20	16.10.2008 17:21:43	ESCR0003
20	18	55	16.10.2008 17:21:50	ESCR0003
20	18	88	16.10.2008 17:21:50	ESCR0003
20	21	98	16.10.2008 17:21:01	ESCR0003
20	21	88	16.10.2008 17:21:03	ESCR0003
20	20	88	16.10.2008 17:21:00	ESCR0003
20	12	77	16.10.2008 17:21:06	ESCR0003
20	14	88	16.10.2008 17:21:10	ESCR0003

Kuva 12 Esimerkki kuva raportista

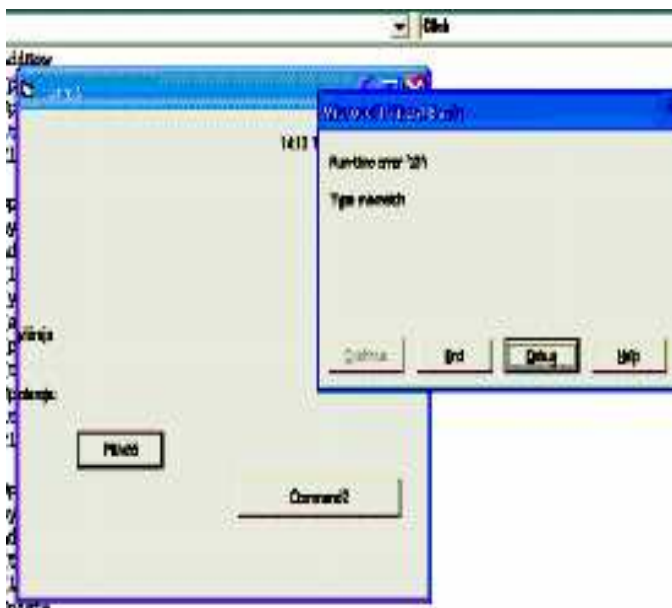
8 TESTAUS

Testaus on vaihe, jota toteutetaan koko projektin ajan. Testausta suoritettiin siksi, että nähtäisiin kuinka ohjelma toimii ja toimiiko oikein.

Anturia testattiin vertaamalla lämpötilalukemia sisälämpömittariin. Ohjelmistoa pidettiin päällä useiden tuntien ajan. Tuloksena oli, ettei ohjelma kaatunut.

8.1 Ohjelmisto

Ohjelmaa testattiin myös siten, että jos ohjelman hälytysrajoihin syöttää muuta kuin numeroita, niin ohjelmaan tulee toimintahäiriö. Kuvassa 13 on esimerkki testauksessa tehty testaus, jossa näkyy toimintahäiriö. Tämä tuli kun käyttäjältä pyydettiin hälytysrajoja ja käyttäjä laittoi numeroiden sijasta kirjaimia.



Kuva 13 Esimerkki toimintahäiriöstä

Ohjelmaa testattiin kahdessa eri tietokoneessa ja kahdella eri käyttöjärjestelmällä. Tietokoneissa käytetyt käyttöjärjestelmät olivat Windows XP ja Windows 2000. Mitään ongelmia ei tullut esiin, kun ohjelman toimivuutta testattiin.

Ohjelmiston hälytysrajoja testattiin vuoroin lämmittämällä ja viilentämällä lämpöanturia. Ohjelmisto hälytti lämpötilojen ylityksistä ja alituksista juuri niin kuin oli alunperin suunniteltu. Raportoinnin osa toimi suunniteltusti. Paperi tulosteet tulostuivat oikein.

8.2 Mahdollisten ongelmien selvittäminen ja niiden ratkaiseminen

Mahdollisia ongelmia saattaa tulla, jos käyttäjä antaa väärää tietoa hälytysrajoille, niin ohjelmaan tulee toimintahäiriö. Tämä olisi estettävissä tiedontarkistus-menetelmillä. Visual Basicissa on validate-event ominaisuus, jolla voidaan tarkistaa käyttäjän antaman syötteen oikeellisuus.

Havaittiin myös, että tietokannan kokoa ei ole rajoitettua/määritelty. Tuloksena saattaa olla, että pitkän ajan kuluessa tietokannasta voi tulla kohtuuttoman suuri. Tämä voidaan estää siten, että tietyn ajanjakson jälkeen tietoja ruvetaan tallentamaan uuteen tietokantaan ja vanhoja tietokantoja poistetaan käytöstä.

9 LOPPUPÄÄTELMÄT JA TUTKIMUSTULOKSET

9.1 Tavoitteiden toteutuminen

Tämän opinnäytetyön tavoitteena oli tehdä ja toteuttaa ohjelma, josta tiedot tallennetaan tietokantaan. Tiedot, jotka tietokoneelle menevät tulevat oheislaitteelta. Oheislaitte mittaa tietokoneen lämpötilaa. Työ on tehty omaan tarkoitukseen.

Tavoitteet onnistuivat, mutta ohjelmasta olisi voinut saada kehittyneemmän/laajemman. Ohjelmaa ja tietokantaa voisi parantaa tai kehittää paremmaksi henkilö, joka hallitsee tai on enemmän kokemusta näistä asioista.

9.2 Tavoitteiden onnistuminen

Työn alkuun saattaminen oli aluksi hankalaa, johtuen erinnäköisistä syistä. Tämä koostautui myöhemmin ja aikataulu ns. petti. Lähdekoodin tekemiseen meni enemmän aikaa kuin alunperin oli suunniteltu, koska koodin tekeminen oli odotettua vaikeampaa. Lopputa kohti tuli kiire saada työ tehtyä. Otin opikseni sen, että täytyy yrittää pitää mahdollisimman hyvin aikataulusta kiinni.

Jouduin hakemaan paljon taustatietoa, koska Tmex oli minulle uusi tuttavuus. Muutenkin koska ohjelmointi ei ole minulle vahva alue, niin jouduin tekemään paljon töitä ohjelman toimintaan saattamiseksi.

Omat tavoitteet olivat kehittää ja ymmärtää ohjelmointia. Tavoitteena oli myös saada ohjelma toimimaan halutulla tavalla.

Työtä tehdessäni sain käytännön kokemusta tietokannasta ja ohjelmoinnista. Lisäksi sain vähän lisätietoa jo koulussa opittuihin asioihin ja koska jouduin opettelemaan paljon uutta tietoa, niin tuli sitten opittua paljon uusia asioita.

9.3 Kehitysehdotuksia

Ohjelman kannalta kehityskelpoinen idea voisi olla ohjelmaan liittyen se, että koska nyt päivittämisen joutuu käyttäjä itse hoitamaan, niin kehittyneempi ohjelmoija voisi lisätä lähdekoodiin päivityksen esimerkiksi siten, että päivitys tapahtuisi automaattisesti vaikka 5 minuutin välein.

Hälytysjärjestelmää voisi muokata siten, että hälytysjärjestelmä hälyttäisi tekstiviestillä. Tällä hetkellä käyttäjä saa tietokoneen ruudulle ilmoituksen lämpötilan ylityksestä tai alituksesta. Näin voisi helpottaa järjestelmäasiantuntijan roolia.

Raportointia voisi kehittää siten, että tiedot tallennettaisiin esimerkiksi HTML-tiedostoon, jolloin tietojen esittäminen olisi mahdollista intranetissä/internetissä.

LÄHDELUETTELO

Afterdawn.com. Ei ajankohtaa. USB.[verkkodokumentti] Saatavissa:
<http://fin.afterdawn.com/sanasto/termit/usb.cfm> (Haettu 06.11.2006)

Anonyymi. 02.12.2001. One wire/1-Wire. <http://www.nic.fi/~skarna/etusivu.html> (Haettu 12.09.2006)

Callahan Evan. 2000. Microsoft Access 2000-Visual Basic for Applications.Oy Edita Ab. ISBN 951-826-092-3.

Dallas Semiconductor/Maxim. 30.04.2001. DS18B20. [verkkodokumentti] Saatavissa:
http://www.maxim-ic.com/quick_view2.cfm?qv_pk=2812. (Haettu 20.3.2006.) Päivitetty 15.06.2005

Dallas Semiconductor/Maxim. Ajankohta ei tiedossa. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. [verkkodokumentti] Saatavissa:
<http://pdfserv.maxim-ic.com/en/ds/2812.pdf>. (Haettu 25.05.2006)

Dallas Semiconductor/Maxim. 20.10.1999. DS9097, DS9097E.[kuva] Saatavissa:
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2982 (24.09.2006)

Gregory Elizabeth. 16.07.2004. Serial port communication.[verkkodokumentti] Saatavissa: <http://cnx.org/content/m12293/latest/>. (24.09.2006)

fCoder Group International. 1998-2002. About programming. [verkkodokumentti] Saatavissa: http://www.lookrs232.com/com_port_programming/(Haettu 14.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex API Overview. [verkkodokumentti] Saatavissa:
http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/tmex8lrr.html(Haettu 14.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex Session API Overview. [verkkodokumentti] Saatavissa:
http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/sess1ut3.html. (Haettu 20.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex File Operations API Overview. Saatavissa:
http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/fiop6apj.html. (Haettu 20.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex Transport API Overview. [verkkodokumentti] Saatavissa:
http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/tran5ozb.html. (Haettu 20.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex Network API Overview. [verkkodokumentti] Saatavissa: http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/netw70vr.html. (Haettu 20.09.2006)

Dallas Semiconductor. Ajankohta ei tiedossa. Tmex Hardware Specific API Overview. [verkkodokumentti] Saatavissa: http://files.dalsemi.com/auto_id/softdev/owdocs/Docs/TMEX/hso5i2f.html. (Haettu 20.09.2006)

Heinonen Petri. 03.02.2005. VBA-perusteet erittäin lyhyesti. Jyväskylän yliopisto. [verkkodokumentti] Saatavissa: <http://appro.mit.jyu.fi/doc/tiedonhallinta/vba/>. (Haettu 01.10.2006)

Kangas Antti. 2002. Mikrobitti - Suomen suurin tietokonelehti. Nopeat sarjaväylät esitetyvät. [verkkodokumentti] Saatavissa: <http://www.mikrobitti.fi/nettijatkot/2002/02/vaylat/> (Haettu 06.11.2006)

Laaksonen Antti, Eronen Lassi. 2002. Visual Basic-opas. [verkkodokumentti]. Saatavissa: http://www.ohjelmointiputka.net/opas.php?tunnus=vbo_1. (Haettu 05.10.2006)

Laamanen Tomi. 20.11.2000. RS485. Jyväskylän yliopisto. [verkkodokumentti]. Saatavissa: http://72.14.221.104/search?q=cache:CZCT3uNOoUcJ:www.cc.jyu.fi/~laamanen/datasiirto/+rs485&hl=fi&gl=fi&ct=clnk&cd=1&lr=lang_fi (Haettu 04.11.2006)

Mustalahti Pasi T. 04.12.2002. Sarjaväylä eli RS232 eli com-portti. Turun yliopisto. [verkkodokumentti] Saatavissa: <http://users.utu.fi/ptmusta/rs232.shtml> (Päivitetty 17.12.2004) (Haettu 05.11.2006)

Rahmel Dan. 1999. Visual Basic 6-tietokantaohjelmointi. IT Press. ISBN 951-826-078-8.

Sainio Arto. 2002. Access 2002 – visual-sarja. Docendo Finland Oy, SanomaWSOY-konserni. ISBN 951-846-123-6.

Tietotekniikka liitto. 28.8.2006. Visuaalinen ohjelmointi - visual programming [verkkouutinen]. Saatavissa: http://www.ttlry.fi/osaamisyhteisot/atksanasto/viikon_sana/?x20547=1341302 (Haettu 10.10.2006)

Vähimaa Aleks. Mikrobitti - Suomen suurin tietokonelehti, Marras 11/2006. Gigatavut kämmenelle s. 58, Sanoma Magazines Finland Oy (haettu 04.11.2006)

Wikipedia-vapaa tietosanakirja. Microsoft Access. [verkkodokumentti] Saatavissa: http://fi.wikipedia.org/wiki/Microsoft_Access päivitetty: 29.08.2006 kello 15.27.

Wikipedia-vapaa tietosanakirja. RS232. [verkkodokumentti] Saatavissa:
<http://fi.wikipedia.org/wiki/RS232> Päivitetty: 30.08.2006 kello 23.01

Wikipedia-vapaa tietosanakirja. USB. [verkkodokumentti] Saatavissa:
<http://fi.wikipedia.org/wiki/USB> Päivitetty: 06.11.2006 kello 14.55

Wikipedia-vapaa tietosanakirja. Visual Basic. [verkkodokumentti] Saatavissa:
http://fi.wikipedia.org/wiki/Visual_Basic Päivitetty 20.08 2006 kello 04.59.

PROJEKTIN SUUNNITTELU

Työn rajaus:

Opinnäytetyö on rajattu siten, että työssä tehdään ohjelma, jonka avulla voidaan seurata tietokoneiden lämpötilaa. Ohjelma myös ilmoittaa käyttäjälle, jos tietokoneen lämpötila on noussut liian korkealle tai liian alas.

Tiedot tallennetaan Access-tietokantaan.

Tietokannasta raportointi on tehtävä helppokäyttöiseksi.

Vastuu: Anna Haapala (tekijä)

Projektin aikataulu:

Opinnäytetyön aiheanalyysin hyväksyttäminen

09.01.06

Materiaalin hankinta

01.02-01.03.06

Ohjelmointi, Tietokanta ja Testaus

01.03-01.10.06