



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Jani Koivula

MURTOHÄLYTIN

Tekniikka ja liikenne
2013

TIIVISTELMÄ

Tekijä	Jani Koivula
Opinnäytetyön nimi	Murtohälytin
Vuosi	2013
Kieli	suomi
Sivumäärä	28 + 1 liite
Ohjaaja	Kalevi Ylinen

Tämän opinnäytetyön tarkoituksena oli tehdä uusittu versio aiemmin Jukka Matilan tekemästä murtohälyttimestä, joka oli toteutettu tietotekniikan osastolle opetuskäyttöön. Murtohälyttimeen kuului tekstiviestin lähetys kosketussensorin kahden liittimen erotessa. Opinnäytetyössä käytettyä mikrokontrolleria ohjelmoitiin C-kielillä käyttäen AVR Studio 4-ohjelmaa.

Opinnäytetyössä tutustutaan SMS:n lähetykseen sekä AVR Universal boardiin ja sen lisäosiin. Opinnäytetyössä käytettiin AVR Universal boardia, Atmel ATmega32-mikrokontrolleria, LCD-näyttöä, RS-232 -sarjaliikennekorttia, kosketussensoria ja Nokia 6310i -matkapuhelinta.

Murtohälytintä rakentaessa on hyvä miettiä, mitä ominaisuuksia se tarvitsee ja miten ne kannattaa toteuttaa. Vaihtoehtoja on useita ja eri järjestelmien yhteensopivuuksiin kannattaa kiinnittää huomiota.

ABSTRACT

Author	Jani Koivula
Title	Burglar Alarm
Year	2013
Language	Finnish
Pages	28 + 1 Appendix
Name of Supervisor	Kalevi Ylinen

The purpose of this thesis work was to make a renewed version of an earlier made burglar alarm by Jukka Matila which was done for the department of Information Technology. The burglar alarm sends an SMS when two connectors of a touch sensor are pulled apart. The microcontroller used in this thesis work was programmed with AVR Studio 4 using the C-language.

This thesis work introduces sending of an SMS, an AVR Universal board and its expansion components. The components used in this thesis work were the AVR Universal board, an Atmel ATmega32-microcontroller, an LCD-display, an RS-232 -serial communications card, a touch sensor and a Nokia 6310i mobile phone.

When building a burglar alarm it is good to consider which properties are needed and how to execute them. There are multiple options and compatibility of different systems should be considered.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	8
2	LAITTEISTO	9
2.1	AVR Universal board ja sen ominaisuudet.....	9
2.2	AVR Universal boardin liitännäiset.....	11
2.2.1	LCD-näyttölevy.....	11
2.2.2	RS-232 -liitäntälevy	12
2.3	ATmega32-mikrokontrolleri.....	13
2.4	Matkapuhelin	14
2.5	Kosketussensori	15
2.6	AVR Studio 4.....	16
2.7	AVR Toolchain.....	16
3	TIETOLIIKENNE.....	17
3.1	Tekstiviestipalvelu	17
3.2	RS-232	17
4	TYÖN SUORITUS	19
4.1	Komponenttien liittäminen	19
4.1.1	RS-232 -levyn liittäminen	19
4.1.2	LCD-näyttölevyn liittäminen	20
4.1.3	Kosketussensorin liittäminen	21
4.2	AVR Studio 4 käynnistäminen	21
4.3	Ohjelmointi	24
5	LABORATORIOTYÖOHJE	26
6	YHTEENVETO	27
	LÄHTEET.....	28
	LIITTEET	

KUVA- JA TAULUKKOLUETTELO

Kuva 1.	AVR Universal board	s. 10
Kuva 2.	Mikrokontrollerien kytkentäkaavio	s. 10
Kuva 3.	PORTA-PORTD pinnien kytkentä	s. 11
Kuva 4.	LCD-näyttölevy	s. 12
Kuva 5.	LCD-näyttölevyn kytkentäkaavio	s. 12
Kuva 6.	RS-232 -liitäntälevy	s. 13
Kuva 7.	RS-232 -liitäntälevyn kytkentäkaavio	s. 13
Kuva 8.	ATmega32 pinnien järjestys	s. 14
Kuva 9.	Nokia 6310i	s. 15
Kuva 10.	Rele	s. 15
Kuva 11.	RS-232 -portin pinnijärjestys	s. 18
Kuva 12.	Muutokset RS-232 -levylle.	s. 20
Kuva 13.	LCD-näyttö toiminnassa	s. 21
Kuva 14.	New Project	s. 22
Kuva 15.	Projektin nimi ja tallennuspaikka	s. 22
Kuva 16.	Alustan ja laitteen valinta	s. 23
Kuva 17.	AVR Studio 4	s. 24
Kuva 18.	Murtohälytin	s. 25

LYHENNELUETTELO

3GPP	Third Generation Partnership Project 3G-järjestelmien standardijärjestöjen yhteistyöorganisaatio
A/D	Analog to Digital Analogisesta digitaaliseksi muuntava piiri
DCE	Data Communication Equipment Kommunikaatiolaite
DTE	Data Terminal Equipment Päätelaite
EEPROM	Electrically Erasable Programmable Read-Only Memory Sähköisesti purettava haihtumaton puolijohdemuisti
ETSI	European Telecommunications Standards Institute Eurooppalainen telealan standardisoimisjärjestö
GCC	GNU Compiler Collection GNU-kääntäjän kokoelma
GSM	Global System for Mobile Communications Maailmanlaajuinen matkapuhelinjärjestelmä
IDC	Insulation-Displacement Connector Pikaliitin
JTAG	Joint Test Action Group Standardi IEEE 1149.1

LCD	Liquid-crystal display Nestekidenäyttö
LED	Light-Emitting Diode Ledi
MIPS	Million Instructions Per Second Tietokoneiden suorituskyvyn mittayksikkö
RISC	Reduced Instruction Set Computing Tietokoneiden suunnitteluarkkitehtuurien suunnittelufilosofia
RS-232	Regular Standard 232 RS-232 -tietoliikenne standardi
SMS	Short Message Service Tekstiviesti
SRAM	Static Random-Access Memory Staattinen keskusmuisti
WAP	Wireless Application Protocol Langattomien sovellusten protokolla

1 JOHDANTO

Murtohälytyn valittiin tämän opinnäytetyön aiheeksi, koska aikaisemmin Jukka Matilan tekemä versio ei toiminut oikein. Tämä pieni hälytysjärjestelmä sopii mökille tai varastoon ja se valvoo yhtä sisäänkäyntiä. Laite sisältää LCD-näytön, kosketussensorin ja Nokia 6310i-matkapuhelimen. Hälytintä rakentaessa on hyvä miettiä, mitä ominaisuuksia se tarvitsee ja miten ne toteuttaa. Vaihtoehtoja löytyy useita ja eri järjestelmien yhteensopivuuksia kannattaa miettiä.

Kyseiset laitteet liitettiin AVR Universal boardiin ja ATmega32-mikrokontrolleria ohjelmoitiin AVR Studio 4-ohjelmalla käyttäen C-ohjelmointikieltä. Ohjelman kääntäjänä käytettiin AVR Toolchainia, joka sisältää yleisimmin käytetyt ohjelmointikirjastot. Kosketussensorin kahden liittimen erotessa tai liittyessä, mikrokontrollerille ohjelmoitu ohjelma hyppää keskeytysfunktioon. Ohjelma lähettää SMS-viestiin tarvittavat AT-komennot RS-232 -sarjaliikennejohtoa pitkin matkapuhelimelle, jolloin matkapuhelin lähettää tekstiviestin annettuun numeroon ja ilmoittaa onko liittimet auki vai kiinni. Kosketussensorina toimi tavallinen rele, joka laitettiin muuttamaan PORTD yhden pinnin tilaa joko päälle tai pois.

Opinnäytetyössä tutustutaan tarkemmin RS-232 -sarjaliikenteeseen, sekä SMS-viestiin ja sen ominaisuuksiin. Lisäksi työssä tutustutaan AVR Universal boardiin ja sen sisältämiin liitännäisiin. Työ sisältää myös Vaasan ammattikorkeakoululle tarkoitetun laboratoriotyöohjeen ja opiskelijat voivat itse tutkia, sekä opiskella sulautetun järjestelmän toimintaa. Työn lopussa olevassa liitteessä on ATmega32-mikrokontrollerille ohjelmoitu ohjelma kommentteineen ja opiskelijat voivat näin tutustua myös ohjelmointiin.

2 LAITTEISTO

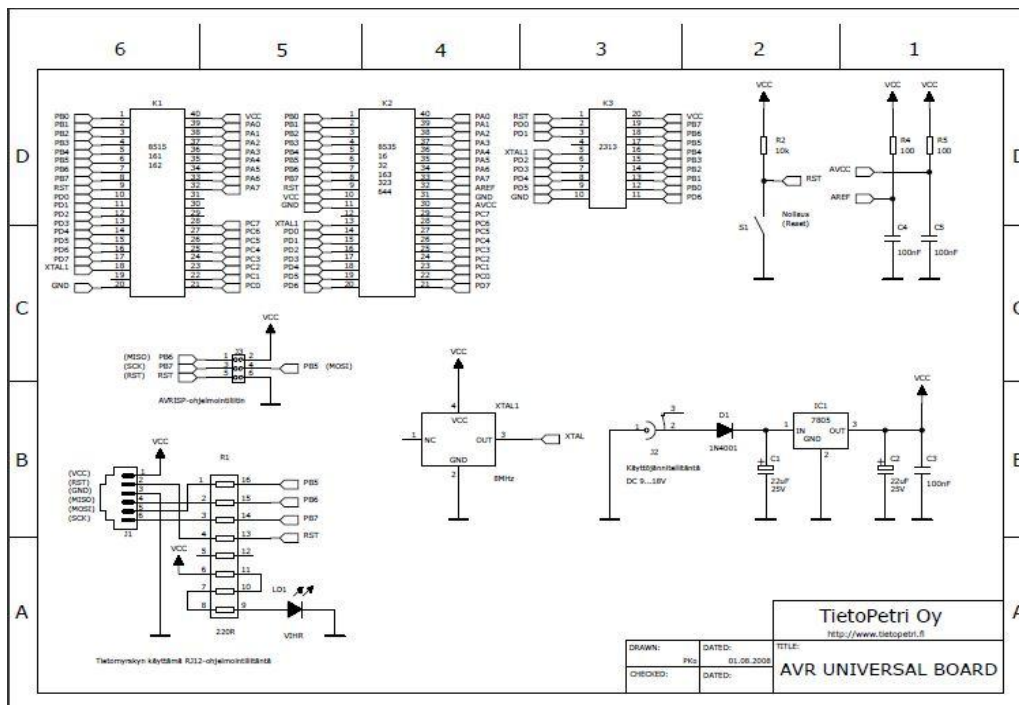
Tässä kappaleessa tutustutaan murtohälyttimessä käytettyihin komponentteihin. Laitteisto koostuu AVR Universal boardista ja sen liitännäisistä, Nokia 6310i-matkapuhelimesta, kosketussensorista ja Atmel ATmega32-mikrokontrollerista.

2.1 AVR Universal board ja sen ominaisuudet

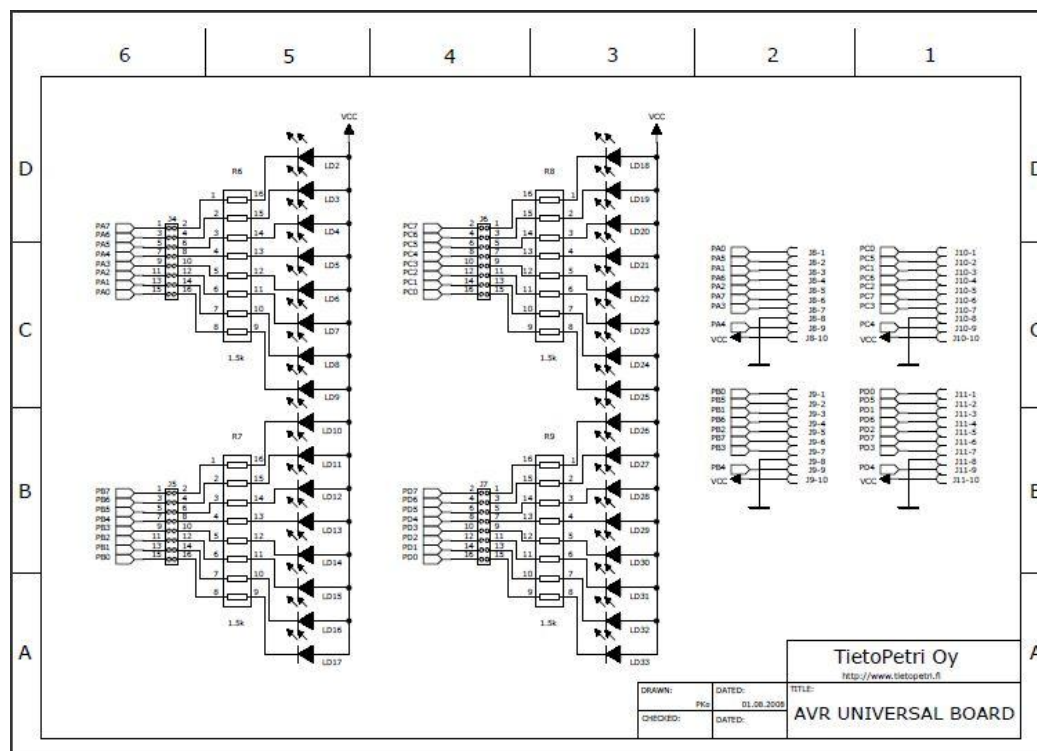
AVR Universal board (**Kuva 1.**) voidaan ohjelmoida eri ohjelmointiympäristöissä, kuten CodeVision tai WinAVR, jota käytettiin tässä oppinnäytetyössä. Kortissa on kolme eri paikkaa mikrokontrollereille, joita voidaan testata ja ohjelmoida. Tässä työssä ohjelmoitiin ATmega32-mikrokontrolleria C-ohjelmointikielellä. AVR Universal board sisältää ohjelmointiliitännät Atmel AVRISPille ja Tietomyrskyn käyttämälle RJ12:lle sekä vapaasti käytettävissä olevat mikro-ohjaimen portit PORTA, PORTB, PORTC ja PORTD. AVR Universal boardin käyttöjännite on DC 9 – 18 V. Kortti sisältää myös 32 kpl jumppereilla irtikytkettävää LEDiä (Light-Emitting Diode), jotka ilmoittavat portin bittien tilan, 8 MHz kideoskillaattorin ja reset-kytkimen. Mikrokontrollerille tarkoitettujen liittimien kytkentäkaavio (**Kuva 2.**) ilmoittaa miten mikrokontrolleri tulee liittää levyille. PORTA-PORTD liittimien kytkentäkaavio (**Kuva 3.**) ilmoittaa liittimien pinnien järjestyksen. (AVR Universal board, 21.1.2013)



Kuva 1. AVR Universal board. (AVR Universal board, 21.1.2013)



Kuva 2. Mikrokontrollerien kytkentäkaavio. (AVR Universal boardin kytkentäkaavio, 21.1.2013)



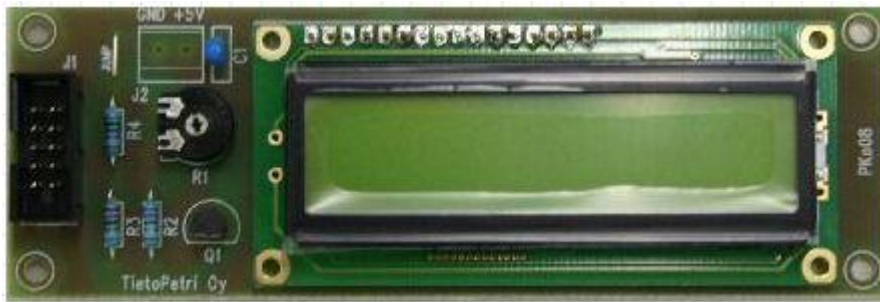
Kuva 3. PORTA-PORTD pinnien kytkentä. (AVR Universal boardin kytkentäkaavio, 21.1.2013)

2.2 AVR Universal boardin liitännäiset

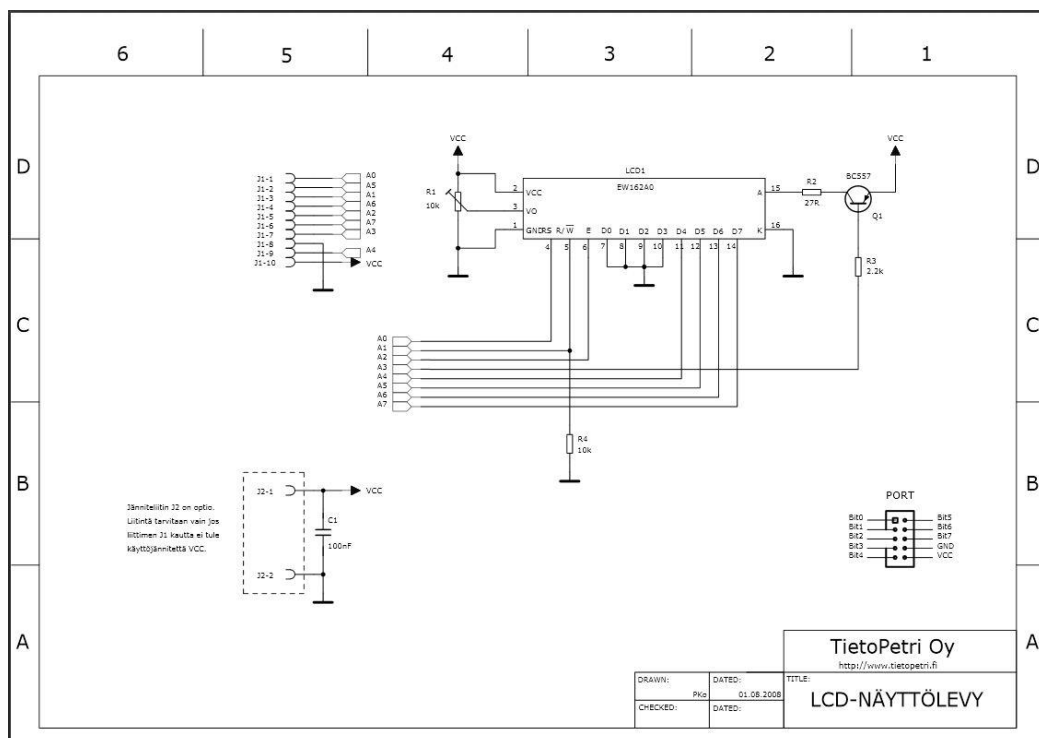
Tässä luvussa kerrotaan tarkemmin AVR Universal boardin liitännäisistä. Liitännäisiä on useita, mutta työssä keskitytään vain siinä käytettyihin liitännäisiin. Opinnäytetyössä käytettiin kahta liitännäistä, LCD-näyttölevyä (Liquid-crystal display) ja RS-232 -liitännälevyä (Regular Standard 232).

2.2.1 LCD-näyttölevy

LCD-näyttölevy (**Kuva 4.**) sisältää 2x16 -näyttömoduulin. Näytön kirkkautta voidaan säätää kääntämällä trimmeripotentiometriä R1 (**Kuva 4.**). Näyttö liitetään AVR Universal boardiin IDC-liittimellä (Insulation-Displacement Connector). Kytkentäkaavio (**Kuva 5.**) osoittaa kuinka näyttölevy on koottu ja miten se kytketään AVR Universal boardille.



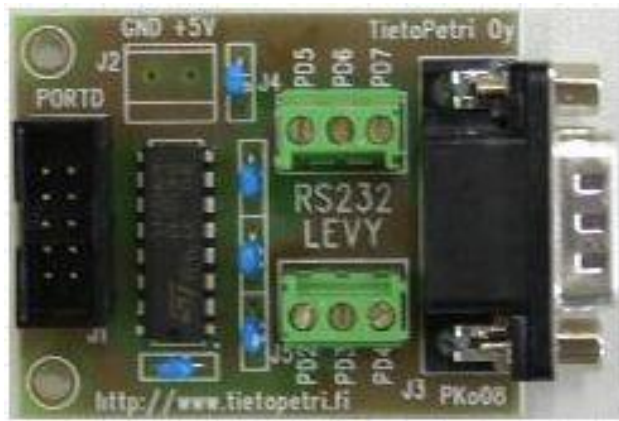
Kuva 4. LCD-näyttölevy. (LCD-näyttölevy, 1.4.2013)



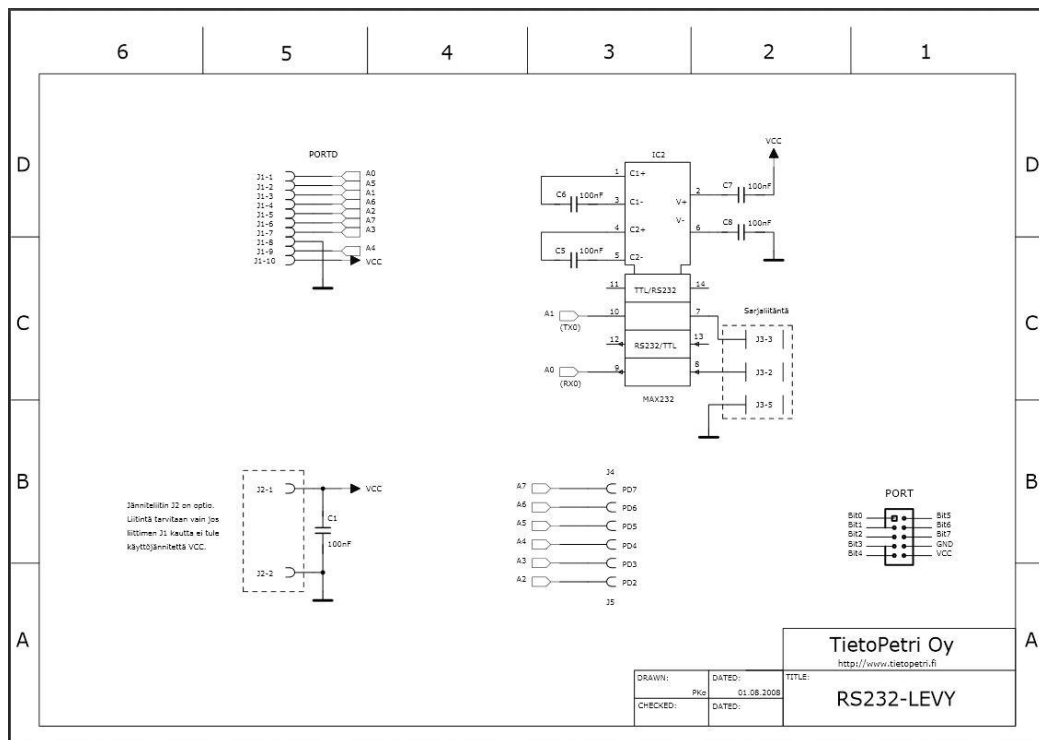
Kuva 5. LCD-näyttölevyn kytkentäkaavio. (LCD-näyttölevyn kytkentäkaavio, 1.4.2013)

2.2.2 RS-232 -liitäntälevy

RS-232 -liitäntälevy (**Kuva 6.**) mahdollistaa matkapuhelimen liittämisen AVR Universal boardille. Matkapuhelimen ja RS-232 -liitäntälevyn välinen sarjaliikenneliitin on 9-napainen uros. RS-232 -liitäntälevy liitetään AVR Universal boardiin IDC-liittimellä. Kytkeäkaavio (**Kuva 7.**) osoittaa kuinka sarjaliikennelevy on koottu ja miten se kytketään AVR Universal boardille.



Kuva 6. RS-232 -liitântälevy. (RS-232 -liitântälevy, 1.4.2013)

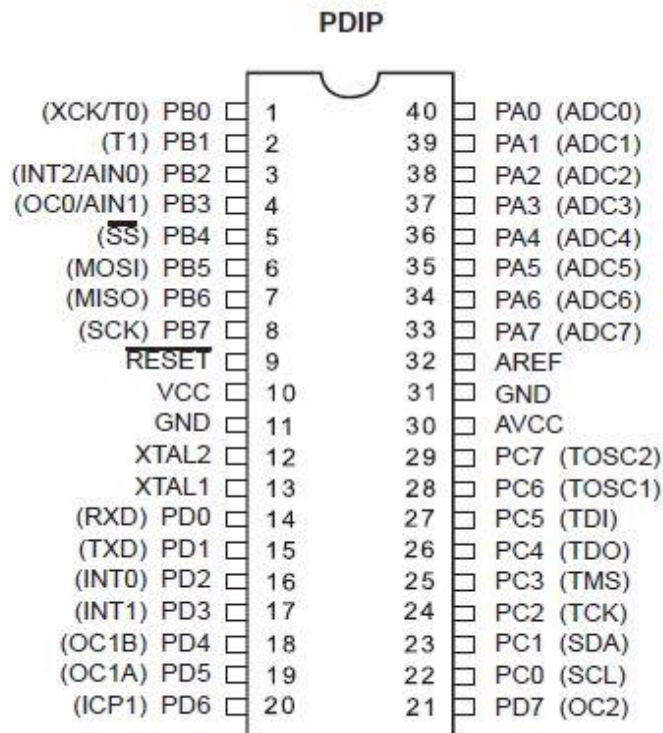


Kuva 7. RS-232 -liitântälevyn kytkentäkaavio. (RS-232 -liitântälevyn kytkentäkaavio, 1.4.2013)

2.3 ATmega32-mikrokontrolleri

Atmelin 8-bittinen AVR RISC -teknologiaan (Reduced Instruction Set Computing) perustuva mikrokontrolleri sisältää 32 KB ohjelmoitavaa flash-muistia, 2 ktavua SRAM (Static Random-Access Memory), 1 ktavua EEPROM (Electrically Erasable Programmable Read-Only Memory), 8-kanavainen 10-

bittinen A/D muunnin (Analog to Digital) ja JTAG (Joint Test Action Group) -rajapinta debuggausta varten. ATmega32 tukee 16 MIPS (Million Instructions Per Second) suoritustehoa 16 MHz taajuudella ja se toimii 2,7 – 5,5 V välillä. ATmega32 pinnien järjestys on esitetty **Kuvassa 8**. Opinnäytetyön kannalta tärkeimmät pinnit ovat PD1, joka on lähtevään sarjaliikenteeseen varattu pinni ja PD2, joka on keskeytyksiä varten varattu pinni. (ATmega32, 23.1.2013)



Kuva 8. ATmega32 pinnien järjestys. (ATmega32/L Datasheet, 23.1.2013)

2.4 Matkapuhelin

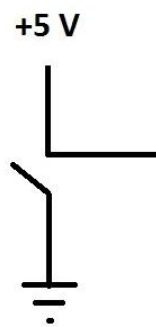
Matkapuhelimenä käytettiin Nokia 6310i-mallia (**Kuva 9**). Matkapuhelin valittiin työhön, koska siinä oli sarjaliikenneliitäntä. Matkapuhelin sisältää myös monia yleisimpiä sovelluksia, kuten bluetooth, GPRS, Java ja sähköposti. (Käyttöohje Nokia 6310i, 9.3.2013)



Kuva 9. Nokia 6310i. (Käyttöohje Nokia 6310i, 9.3.2013)

2.5 Kosketussensori

Kosketussensorina toimii tavallinen rele (**Kuva 10.**). Rele sisältää metalliliuskoja, joita ohjataan magneetin avulla. Kun magneetti irroitetaan vastakappaleesta metalliliuska avautuu ja aiheuttaa oikosulun. (The electromechanical relay of Joseph Henry, 15.4.2013)



Kuva 10. Rele.

2.6 AVR Studio 4

AVR Studio on suunniteltu mikrokontrollereiden ohjelmointia varten. Tässä työssä käytettiin AVR Studio 4.19. AVR Studiolla voi ohjelmoida ja debuggata C/C++ -ohjelmointikielillä. Ohjelman uusin versio tällä hetkellä on AVR Studio 6. (AVR Studio 4, 1.4.2013)

2.7 AVR Toolchain

Tässä työssä käytettiin AVR Toolchain 3.3.0, joka on avoimen lähdekoodin GCC-kääntäjä (GNU Compiler Collection). AVR Toolchainia tarvittiin kääntämään AVR Studio 4.19:lla luotua ohjelmaa ja se sisältää kaikki tarvittavat ohjelmointikirjastot. Ohjelman C-kieli käännettiin mikrokontrollerin ymmärtämiksi biteiksi. (AVR 32-bit GNU Toolchain 3.3.0, 1.4.2013)

3 TIETOLIIKENNE

3.1 Tekstiviestipalvelu

Short Message Service (SMS) tarkoittaa suomeksi tekstiviestiä. SMS-viestintää on käytetty Euroopassa vuodesta 1992 lähtien ja siitä tuli heti Global System for Mobile Communications (GSM) standardi. European Telecommunications Standards Institute (ETSI) kehitti standardit alunperin ja nykyään Third Generation Partnership Project (3GPP) on vastuussa GSM- ja SMS -standardien kehityksestä ja huollosta. Yksi SMS-viesti sisältää enintään 140 tavua (1120 bittiä) dataa, joten yksi viesti voi sisältää 160 7-bittistä kirjainta, eli latinalaisia aakkosia tai 70 kirjainta 16-bittisiä, kuten kiinalaisia kirjaimia. SMS toimii kaikilla Unicoden tukemilla kielillä ja lisäksi arabian, kiinan, japanin ja korean kielillä. Tekstin lisäksi SMS voi sisältää binääristä dataa, kuten soittoääniä, kuvia, animaatioita, käyntikortteja ja Wireless Application Protocol (WAP) -asetuksia. Yksi SMS:n suurin etu on, että 100% GSM-puhelimista tukee sitä toisin kuin WAP ja Java, jotka eivät ole tuettuina monissa vanhoissa matkapuhelimissa. (Introduction to SMS messaging, 9.3.2013)

3.2 RS-232

RS-232 (Regular Standard 232) on kahden tietokonelaitteen väliseen kommunikointiin suunniteltu tietoliikenneliitäntä. RS-232 siirtää dataa yksi bitti kerrallaan asynkronisesti, eli data tarvitsee aloitus- ja lopetusbitin. Laitteet, jotka kommunikoivat keskenään voivat olla joko päätelaite DTE (Data Terminal Equipment) tai kommunikaatiolaite DCE (Data Communication Equipment). RS-232 -porttia käyttävät laitteet ovat yksinkertaisia ja näin ollen päihittävät monia muita portteja käyttäviä laitteita. Sarjaliikenneporttia on käytetty tietokoneissa pisimpään ja standardia on muutettu ajan kuluessa. Uudempia standardeja ovat RS-232A, RS-232B ja RS-232C. Ainoa tietokoneen osa, joka tarvitsee -12 V jännitteen, on RS-232 minkä takia tietokoneiden valmistajat haluavat päästä eroon RS-232 -portista, koska se lisää turhia kustannuksia ja monimutkaisuutta virtalähteissä. Sarjaliikenneportteja on saatavilla eri tarkoituksiin eri kokoisina,

esimerkiksi 4-, 9- tai 25-pinnisinä. Tässä työssä käytettiin 9-pinnistä urosta (**Kuva 11.**) (What is RS232 and serial communications?, 15.4.2013)

9 Pin Connector on a DTE device (PC connection)



Male RS232
DB9

Pin Number	Direction of signal:
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

Kuva 11. RS-232 -portin pinnijärjestys. (What is RS232 and serial communications?, 15.4.2013)

Pinnin numero ja selitys suomeksi.

- 1 kanta-aaltotunnistus
- 2 tuleva data
- 3 lähtevä data
- 4 päätelaite valmiina
- 5 signaalimaa
- 6 siirtolaite valmiina
- 7 lähetyspyyntö
- 8 lähetysvalmius
- 9 soiton osoitus.

4 TYÖN SUORITUS

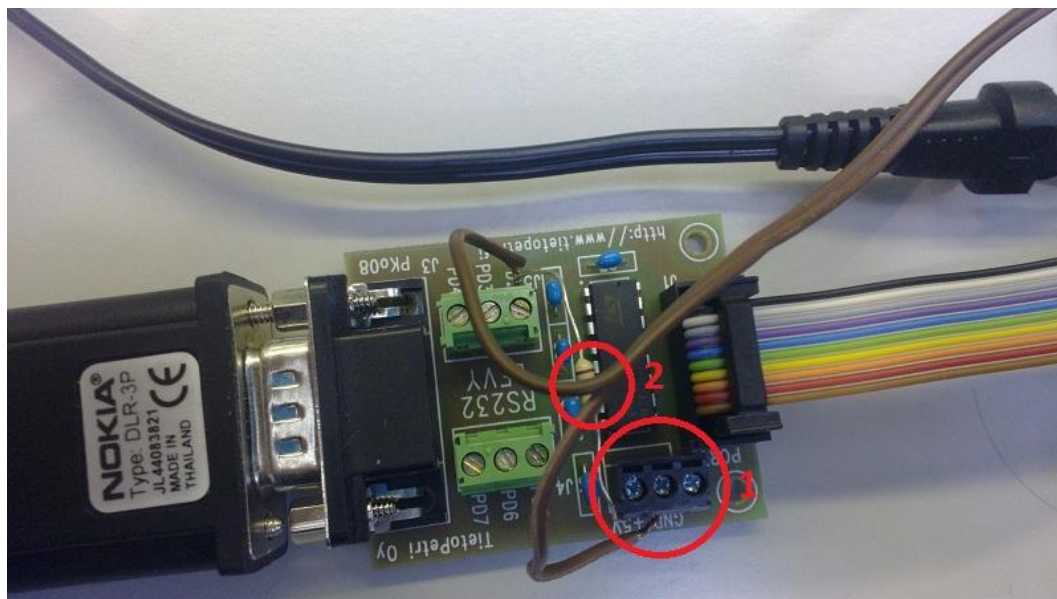
Opinnäytetyö aloitettiin liittämällä tarvittavat komponentit paikoilleen AVR Universal boardille. RS-232 -levylle jouduttiin kolvaamaan yksi riviliitin lisää, jotta saatiin +5 V jännite ja maataso käyttöön kyseiselle levyllä kosketussensoria varten. Lopuksi AVR Studio 4-ohjelmalla aloitettiin uusi projekti ja kirjoitettiin ohjelmakoodi.

4.1 Komponenttien liittäminen

Komponentteja ei voi liittää AVR Universal boardille mielivaltaisesti. ATmega32 pinnijärjestyksestä täytyy katsoa mikä pinni ohjaa mitään porttia AVR Universal boardilla ja komponentit liitetään sen mukaan.

4.1.1 RS-232 -levyn liittäminen

RS-232 -levy liitettiin AVR Universal boardin PORTD-liittimeen (**Kuva 1.**). ATmega32-mikrokontrollerin D pinnit ohjaavat sarjaliikennettä ja keskeytyksiä (**Kuva 8.**). Koska ATmega32 ohjaa keskeytyksiä ja sarjaliikennettä samasta portista, jouduttiin kolvaamaan RS-232 -levylle vielä yksi riviliitin lisää. Kolvauksella saatiin +5 V jännite ja maataso tuotua samalle levyllä kosketussensoria varten (**Kuva 12.**). +5 V jännite otettiin 10 k Ω vastuksen läpi, jottei mikrokontrolleri vahingoittuisi liian suuresta virrasta, kun kosketussensori tekee oikosulun kahden liittimensä erotessa.



Kuva 12. Muutokset RS-232 -levylle. 1. Riviliitin 2. 10 kΩ vastus.

4.1.2 LCD-näyttölevyn liittäminen

LCD-näyttölevy liitettiin AVR Universal boardin paikkaan PORTA (**Kuva 1**). Kyseinen portti valittiin, koska näytölle oli jo ennestään valmis ohjelma, missä se oli määritelty kyseiseen porttiin toimivaksi. Näytölle voidaan kirjoittaa kahdelle eri riville haluttu teksti. Kyseisessä työssä kirjoitettiin ”SYSTEM ARMED”, kun virrat laitettiin päälle (**Kuva 13**).



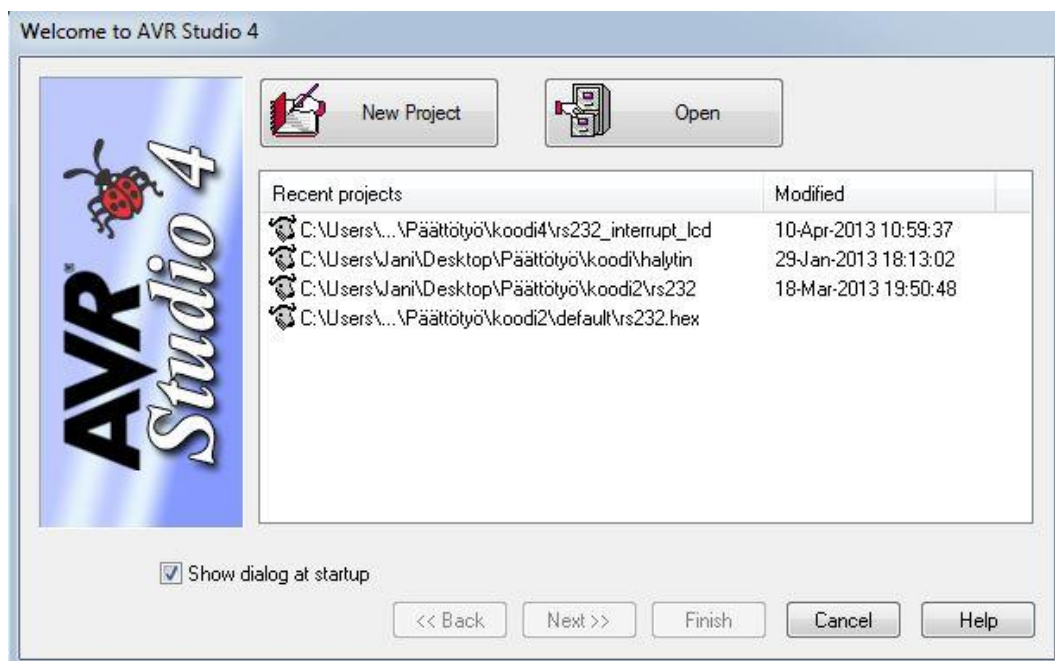
Kuva 13. LCD-näyttö toiminnassa.

4.1.3 Kosketussensorin liittäminen

Kosketussensori liitettiin RS-232 -levylle, koska PORTD tarvittiin myös keskeytyksiä varten. Jännite otettiin 10 k Ω vastuksen (**Kuva 12.**) läpi, jotta välttyttiin mikrokontrollerin tuhoutumiselta liian suuren virran takia kosketussensorin aiheuttaessa oikosulun. Sensorin toinen johto liitettiin pinniin PD2, joka sisältää keskeytykset mikrokontrollerissa ja toinen johto liitettiin maatasoon.

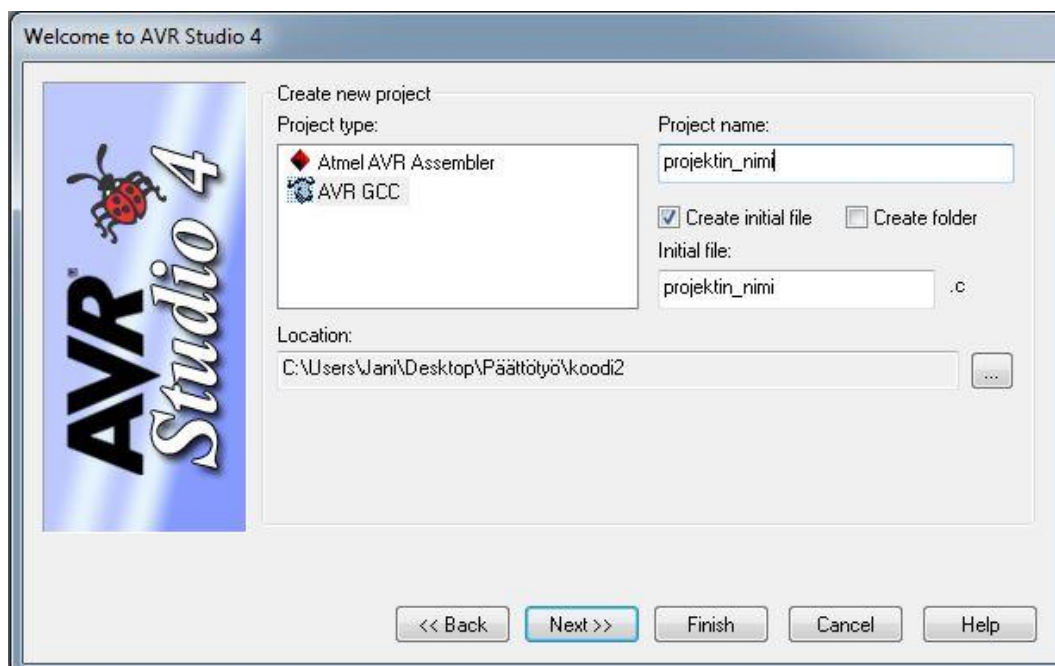
4.2 AVR Studio 4 käynnistäminen

AVR Studio 4-ohjelma käynnistettiin ja valittiin "New Project" (**Kuva 14.**).



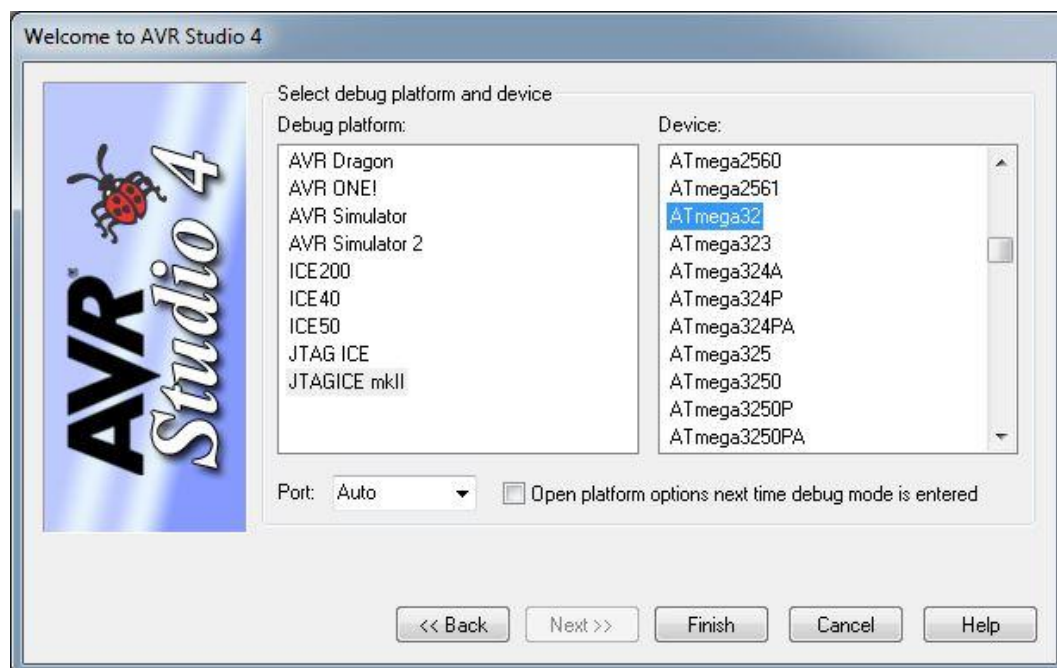
Kuva 14. New Project.

Seuraavaksi valittiin AVR GCC ja projektille kirjoitettiin nimi sekä valittiin haluttu tallennuspaikka ja painettiin ”Next >>” (**Kuva 15.**).



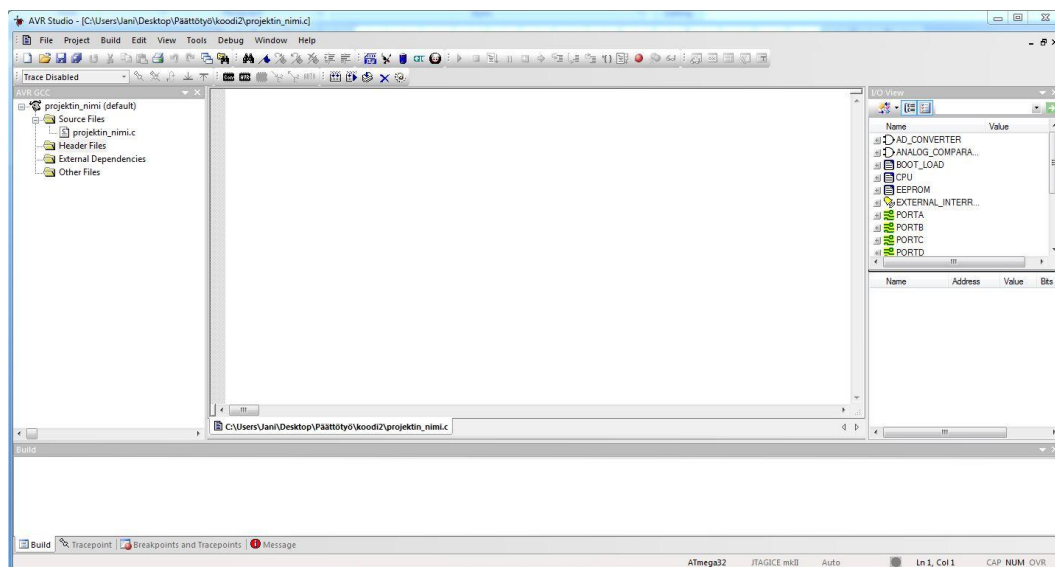
Kuva 15. Projektin nimi ja tallennuspaikka.

Seuraavaksi valittiin debuggusalustaksi JTAGICE mkII ja laitteeksi ATmega32 ja painettiin ”Finish” (**Kuva 16.**).



Kuva 16. Alustan ja laitteen valinta.

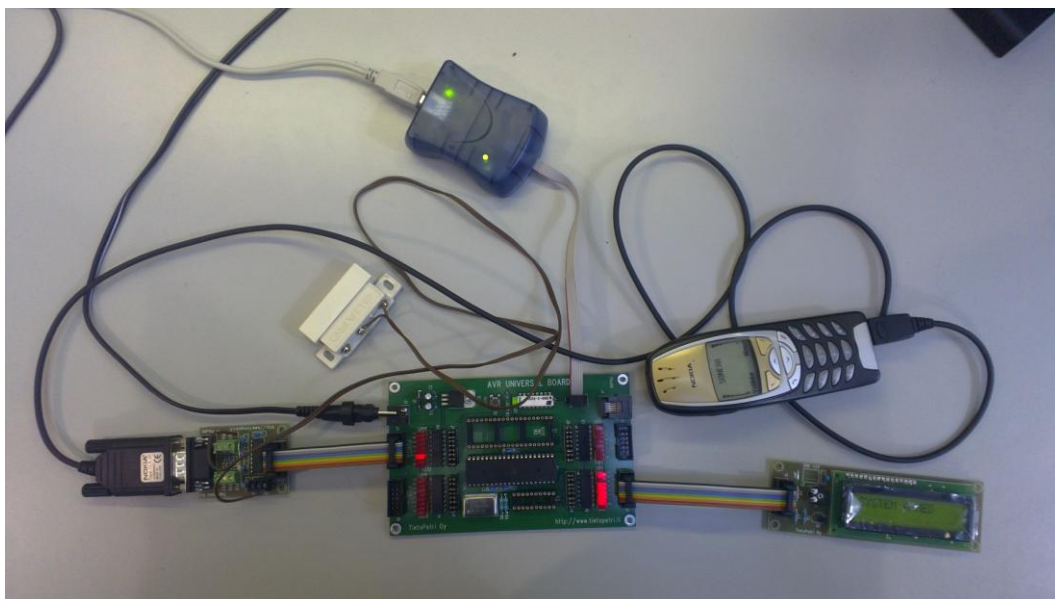
Tämän jälkeen lisättiin vielä kaksi valmiiksi kirjoitettua tiedostoa painamalla ”Source Files” ja ”Header Files” kansioiden päällä hiiren oikeaa nappia ja valittiin ”Add Existing Source File(s)”. ”Source Files”-kansioon lisättiin ”lcd_tat.c” ja ”Header Files”-kansioon lisättiin ”lcd_tat.h”, jotka löytyivät koulun koneelta. Nämä kaksi tiedostoa sisälsivät LCD-näytön määrittelyt PORTA-porttiin. Näiden valintojen jälkeen voitiin aloittaa ohjelman kirjoitus C-kielellä.



Kuva 17. AVR Studio 4.

4.3 Ohjelmointi

Mikrokontrolleria ohjelmoitiin C-kielellä. Ohjelmassa määriteltiin kaikki tarvittavat funktiot, jotta kokonaisuus saatiin toimimaan oikein. Aluksi kaikki lisälaitteet ohjelmoitiin toimimaan erikseen ja samalla tutustuttiin tarvittavaan ohjelmaan sekä ohjelmointikirjastoihin. Sarjaliikenne tarvitsi omat määrittelyt ja funktiot toimiakseen oikein. Keskeytyksiä varten tarvittiin omat määrittelyt, jotka ohjasivat mikä viesti lähtee kun sensori avattiin tai kiinnitettiin. AT-komennot tarvitsivat omat määrittelyt, jotta ne lähtivät oikeaan aikaan ja oikeassa järjestyksessä sarjaportille. ATmega32-mikrokontrollerille määritettiin toimintataajuus 8 MHz. AVR Universal boardin PORTD pinnit ohjelmoitiin toimimaan lähettäjinä ja PORTA pinnit ohjelmoitiin toimimaan vastaanottajina. Kun kaikki kitin lisälaitteet toimivat oikein, lisättiin kaikki ohjelmakoodit yhdeksi isoksi kokonaisuudeksi. Toimiva murtohälytyn näkyy kuvassa 18.



Kuva 18. Murtohälytin.

5 LABORATORIOTYÖOHJE

Kiinnittäkää LCD-näyttö PORTA ja RS-232 -levy PORTD paikkoihin ARV Universal boardilla. Liittäkää RS-232 -levy USB-to-Serial liittimen toiseen porttiin ja Nokian puhelin toiselle puolelle. Käynnistäkää ”Free Serial Port Monitor” -ohjelma ja valitkaa oikea COM-portti analysoitavaksi. Nyt voitte käynnistää ”Hyper Terminal” -ohjelman, valitkaa oikea COM-portti ja BAUD 9600.

Jos kitin LCD-näytöllä ei lue ”SYSTEM ARMED” painakaa reset kytkintä niin, että teksti tulee näkyviin. Nyt voitte irroittaa liittimen osat toisistaan ja seurata liikennettä ”Free Serial Port Monitor” -ohjelmassa. Liittäkää liittimet takaisin yhteen ja seuratkaa liikennettä. Miten tämä liikenne eroaa normaalista tekstiviestiliikenteestä?

Jos kitillä oleva ohjelma on eri kuin tässä työssä vaadittava, niin löydätte sen tämän dokumentin lopusta LIITE 1 kohdasta. Kiinnittäkää sininen debuggeri AVR Universal boardille ohjelmointia varten. Avatkaa AVR Studio 4 ja valitkaa uusi projekti. Valitkaa AVR GCC ja antakaa projektille nimi sekä tallennuspaikka ja painakaa ”Next”. Valitkaa debuggaus alustaksi JTAGICE mkII ja laitteeksi ATmega32, jonka jälkeen voitte painaa ”Finish”. Kopioikaa koodi liitteestä ja lisääkää haluamanne puhelinnumero `"AT+CMGS=\"NUMERO\"";` kohtaan.

Lisätkää tarvittava ”lcd_tat.c” painamalla hiiren oikeaa nappia kansion ”Source Files” päällä ja valitkaa ”Add Existing Source File(s)”. Kyseinen tiedosto löytyy koulun y: asemalta ”Staff” → ”MM” → ”oppilaan kirja” → ”LCD_koodit”. Tehkää sama myös kansiolle ”Header Files” ja lisääkää samasta paikasta ”lcd_tat.h” kyseiseen kansioon. Nyt voitte kääntää ohjelman painamalla ”Build” tai näppäimistöä ”F7”.

Valitkaa ”Tools” valikosta kohta ”Program AVR” ja ”connect”. Seuraavasta listasta valitkaa ”AVRISP mkII” ja portiksi USB. Valitkaa ”Program” välilehdeltä ”Flash” ja hakekaa oikea HEX-tiedosto projektikansioista, minkä jälkeen voitte painaa ”Program”. Nyt kitti on ohjelmoitu toimimaan kurssin vaatimalla tavalla.

6 YHTEENVETO

Työn tarkoituksena oli rakentaa toimiva murtohälytin Vaasan ammattikorkeakoululle. Murtohälyttimen kittinä käytettiin AVR Universal boardia, johon liitettiin Atmel ATmega32-mikrokontrolleri, LCD-näyttö, RS-232-sarjaliikennekortti, kosketussensori ja Nokia 6310i-matkapuhelin. Työtä olisi voinut vielä jatkaa ja lisätä enemmän ja parempia ominaisuuksia, mutta ajan puutteen takia lopetimme.

Olin jo kursseilla opiskellut sulautetun järjestelmän toimintaa, mutta työ syvensi ja paransi oppimaani. Tiedon etsiminen oli suhteellisen helppoa koska aiheena SMS:n lähettäminen mikrokontrollerin avulla ei ollut uusi. Sarjaliikenteestä löytyi tietoa todella paljon, ja opin ohjelmoimaan viestin lähettämiseen tarvittavia AT-komentoja. Lisäksi opin ohjelmoimaan keskeytyksiä, joita tarvittiin seuraamaan releen tilaa.

Ongelmaksi tässä työssä osoittautui ATmega32 toiminta. Saman portin pinnit sisälsivät sarjaliikenteen sekä keskeytykset, joten pieni improvisointi oli tarpeen. Toiseksi ongelmaksi osoittautui myöhemmin AT-komentojen lähetys sarjaliikennekortille ja sarjaliikenteen konfigurointi. Lopulta esimerkkejä sekä kirjallisuutta hyödyntämällä ohjelma saatiin toimimaan haluamalla tavalla. Lopputulokseen olen tyytyväinen vaikka mahdollisuuksia olisi ollut vielä parannella ominaisuuksia.

LÄHTEET

ATmega32. Atmel Corporation. Viitattu 23.1.2013.

<http://www.atmel.com/devices/atmega32.aspx?tab=overview>

ATmega32/L Datasheet. Atmel Corporation. Viitattu 23.1.2013.

<http://www.atmel.com/Images/doc2503.pdf>

AVR 32-bit GNU Toolchain 3.3.0. Atmel Corporation. Viitattu 1.4.2013.

<http://www.atmel.com/Images/avr32-gnu-toolchain-3.3.0.275-readme.pdf>

AVR Studio 4. Atmel Corporation. Viitattu 1.4.2013.

<http://www.atmel.com/tools/avrstudio4.aspx>

AVR Universal board. Tietopetri OY. Viitattu 21.1.2013.

<http://www.tietopetri.fi/tuotteet.html#uni>

AVR Universal boardin kytkentäkaavio. Tietopetri OY. Viitattu 21.1.2013.

<http://www.tietopetri.fi/data/univer.pdf>

Introduction to SMS messaging. Developershome. Viitattu 9.3.2013.

<http://www.developershome.com/sms/smsIntro.asp>

Käyttöohje Nokia 6310i. Nokia OYJ. Viitattu 9.3.2013.

http://nds1.nokia.com/phones/files/guides/6310i_usersguide_fi.pdf

LCD-näyttölevy. Tietopetri OY. Viitattu 1.4.2013.

<http://www.tietopetri.fi/tuotteet.html#lcd>

LCD-näyttölevyn kytkentäkaavio. Tietopetri OY. Viitattu 1.4.2013.

<http://www.tietopetri.fi/data/lcd.pdf>

RS-232 -liitäntälevy. Tietopetri OY. Viitattu 1.4.2013.

<http://www.tietopetri.fi/tuotteet.html#rs>

RS-232 -liitäntälevyn kytkentäkaavio. Tietopetri OY. Viitattu 1.4.2013.

<http://www.tietopetri.fi/data/rs.pdf>

The electromechanical relay of Joseph Henry. History-Computer. Viitattu 15.4.2013. <http://history-computer.com/ModernComputer/Basis/relay.html>

What is RS232 and serial communications?. TAL Technologies. Viitattu 15.4.2013. http://www.taltech.com/support/entry/serial_intro

OHJELMAKOODI

```
/*
*****
/* VARASHÄLYTIN joka lähettää SMS- */
/* viestin keskeytyksen sattuessa. */
/* Jani Koivula */
*****
*****
**** KIRJASTOT JA MUUT MÄÄRITTELYT ****
*****
#include<avr/io.h>
#include<avr/interrupt.h>
#include<util/delay.h>
#include "lcd_tat.h"
#define WAIT(time) for(uint16_t i=0;i<2000;i++)_delay_loop_2(time);
#define XTAL 8000000UL // prosessorin kellotaajuus 8 MHz
#define BAUD 9600UL // määritellään BAUD, eli 9600
*****
**** MÄÄRITELLÄÄN PROTOTYYPIT ****
*****
void USART_Transmit(uint8_t data);
void USART_Init(void);
void MSG_Send(char s[]);
ISR (INT1_vect);
volatile int i;
uint8_t data1[] = "ATE0\r\n"; // echo off, vähentää sarjaliikennettä
```

```
uint8_t data2[] = "AT+CMGF=1\r\n";    // asetetaan puhelin modeemitilaan
uint8_t data3[] = "AT+CMGS=\"NUMERO\""; // annetaan puhelinnumero

/*****/

/**** LÄHETÄ MERKKI DATAREKISTERIIN ****/

/*****/

void USART_Send_String(uint16_t a)
{
    char buffer[50];
    uint8_t i=0;
    sprintf(buffer, "%s", a);
    while( buffer[i]!= '\0')
        {
            USART_Transmit(buffer[i]);
            i++;
        }
}

void USART_Transmit(uint8_t data)
{
    // Wait for empty transmit buffer
    while(!(UCSRA & (1<<UDRE))); // odota kunnes UDRE == 1
    // UDRE on 1, jos puskuri on tyhjä,
    // eli ed. merkki on lähetetty
    UDR = data; // lähtevä merkki USARTin datarekisteriin
}

/*****/

/**** SARJAPORTIN MÄÄRITYS *****/
```

```
/******  
  
void USART_Init(void)  
  
    {  
  
        UBRRH = 0x00;    // write high register first  
  
        UBRL = (XTAL/16/BAUD)-1;  
  
        UCSRB |= (1<<RXEN) | (1<<TXEN);    // 0001 1000  
  
        // Bit 4 – RXEN: Receiver Enable  
  
        // Bit 3 – TXEN: Transmitter Enable  
  
        UCSRB |= 1<<RXCIE;    // rx interrupt enable, 1xx1 1xxx  
  
        // sarjavastanottokeskeytyks sallittu  
  
        UCSRC |= (1<<URSEL) | (1<<UCSZ1) | (1<<UCSZ0);    // 8 Data, 1  
        Stop, No Parity 1xxx x11x  
  
    }  
  
/******  
  
/**** KESKEYTYSFUNKTIO ****/  
  
/******  
  
ISR (INT0_vect)  
  
    {  
  
        cli(); //estetään lisäkeskeytykset  
  
        if(PIND & (1<<PD2)) //PIN2 shorts to ground  
  
            {  
  
                LCD_Clear();  
  
                printf("Auki");  
  
                WAIT(1500);  
  
                LCD_Clear();  
  
                MSG_Send("Auki");  
  
            }  
  
    }  
  
/******
```

```
        i = 1;
    }
else
    {
        LCD_Clear();
        printf("Kiinni");
        WAIT(1500);
        LCD_Clear();
        MSG_Send("Kiinni");
        i = 1;
    }
}

/*****
**** VIESTIN LÄHETYS ****
*****/

void MSG_Send(char s[])
{
    WAIT(1500); //odotetaan
    USART_Send_String(data1); // lähetetään ATE0
    WAIT(1500)
    USART_Send_String(data2); // lähetetään AT+CMGF=1
    WAIT(1500);
    USART_Send_String(data3); // lähetetään AT+CMGS="puhnum"
    USART_Transmit(0x0D); // CR eli enter
    WAIT(1500);
    USART_Send_String(s); // lähetetään ALARM
}
```



```
        USART_Transmit(0x1A);    // ctrl Z
    }

/*****/

/**** PÄÄOHJELMA ****/

/****/

int main(void)
{
    cli();

    i=1;

    DDRD=1<<PD1;

    LCD_init(1, 0, 0);

    fdevopen((void *) LCD_WriteChar, 0);

    USART_Init();

    GICR |= (1<<INT0);    // Enable INT0

    MCUCR |= (1<<ISC00); // Any logical change on INT0 generates an in-
interrupt request

    sei(); // enable global interrupt

    WAIT(1500);

    if(i==1)
    {
        while(1)
        {
            PORTD = 0x00;

            printf("SYSTEM ARMED");

            WAIT(150);

            LCD_Clear();
        }
    }
}
```

}
}
}