

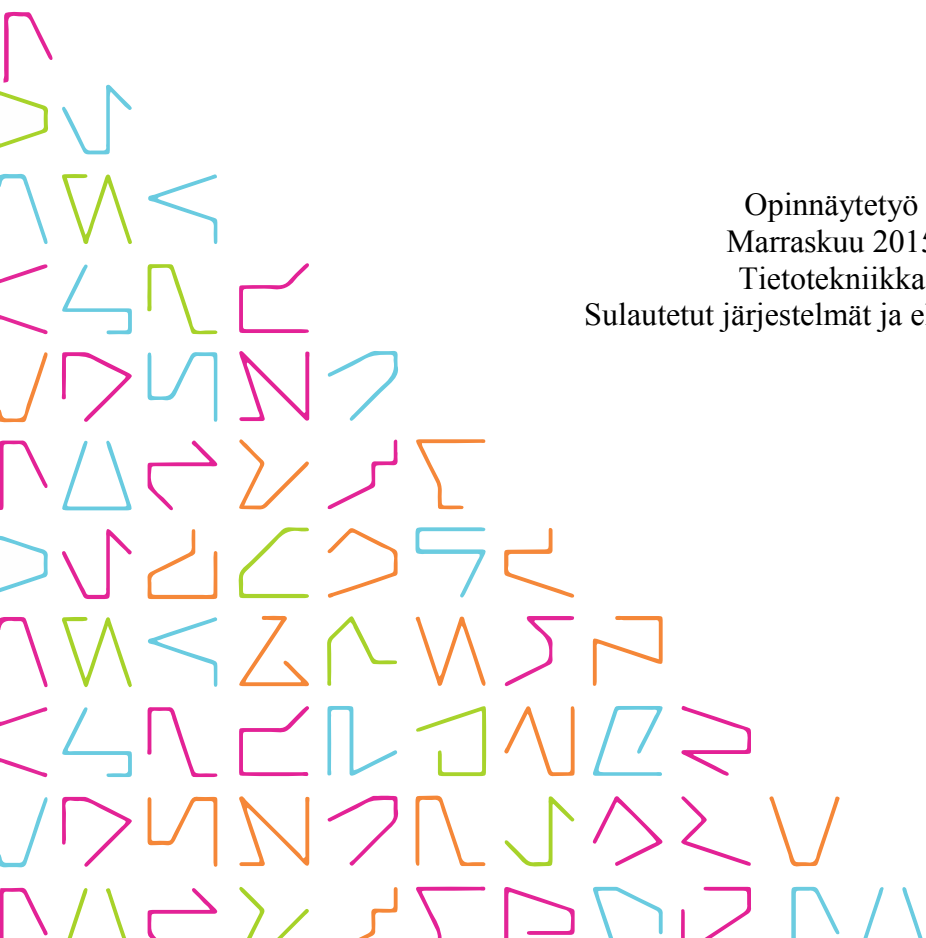


TAMPEREEN
AMMATTIKORKEAKOULU

Lämmityksen langaton ohjaus

Eero Koivulahti

Opinnäytetyö
Marraskuu 2015
Tietotekniikka
Sulautetut järjestelmät ja elektroniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Sulautetut järjestelmät ja elektroniikka

KOIVULAHTI EERO
Lämmityksen langaton ohjaus

Opinnäytetyö 30 sivua, joista liitteitä 4 sivua
Marraskuu 2015

Tässä opinnäytetyössä on tarkoituksena tutustua älykodin suunnitteluun ja toteutukseen lämmityksen ohjauksen kautta. Käytännön toteutuksessa tutustuttiin Bluetoothin ja lämpötila-anturin käyttöön mikroprosessorin kanssa. Työssä esitellään Bluetooth-tiedonsiirron ja lämpötila-anturin toimintaa ja käytännön ratkaisuja. Työssä tehtyä laitetta ei toistaiseksi asenneta mihinkään vaan se jätetään odottamaan sopivaa asennuskohdetta, jonka tarpeen mukaan siihen lisätään toimintoja.

Laite on mahdollista kehittää suuremman kokonaisuuden ohjauslaitteeksi. Huoneiston lämmityksen rinnalla onnistuisi esimerkiksi auton talvilämmitys. Laitteen liittäminen verkkoon tarjoaisi mahdollisuuden saada asunnon tilasta hälytyksiä sähköpostilla käyttämällä liike- ja kosteusantureita. Laitteen käytettävyyttä voisi parantaa graafisen käyttöliittymän kehittämisellä.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information Technology
Embedded Systems

KOIVULAHTI EERO
Wireless Heating System

Bachelor's thesis 30 pages, appendices 4 pages
November 2015

The purpose of this thesis was designing and developing of smart home application for heating control system. In practical implementation Bluetooth and sensors were used with microprocessor. The theoretical section explores Bluetooth data transmission and temperature sensors both specifications and practical solutions. Developed application was not installed for lack of appropriate housing.

Further developing is required for building comprehensive solution. Besides heating of apartment it could also control vehicle heating at winter. Attaching internet connection to developed application it would enable email alarms from motion and moisture sensors. Improving user experience would happen by developing graphical user interface.

Key words: Arduino, smart home, sensors, Bluetooth

SISÄLLYS

1	JOHDANTO.....	6
2	SÄHKÖLÄMMITYS.....	7
2.1	Lämmitys.....	7
2.2	Lämmityksen ohjaus.....	8
3	ARDUINO.....	9
3.1	Kehitysympäristö.....	9
3.2	Käytetty Arduino ja lisälaitteet.....	11
3.2.1	Laitteiston valinta.....	11
3.2.2	Arduino Mega 2560.....	11
3.2.3	Bluetooth-moduuli HC-06.....	12
3.2.4	LCD keypad shield.....	12
4	LÄMPÖTILA-ANTURI.....	14
4.1	Anturi.....	14
4.2	Lämpötilan mittaus.....	15
5	BLUETOOTH.....	16
5.1	Bluetooth-arkkitehtuuri.....	16
5.1.1	Topologia.....	16
5.1.2	Bluetooth-protokollapino.....	17
5.1.3	Bluetooth-arkkitehtuurin fyysinen kerros.....	17
5.1.4	Kanavanvaraus.....	18
5.1.5	Taajuushyppely.....	19
5.2	Yhteystyypit.....	19
5.2.1	Synkroninen yhteys.....	19
5.2.2	Asynkroninen yhteys.....	19
5.3	Sanoma.....	20
6	OHJAUSYKSIKKÖ.....	21
6.1	Langaton ohjaus.....	21
6.2	Paikallinen ohjaus.....	21
6.3	Ohjelmakoodi.....	22
7	POHDINTA.....	25
	LÄHTEET.....	26
	LIITTEET.....	27
	Liite 1. Lähdekoodi.....	27

LYHENTEET JA TERMIT

AD-muunnos	Analogisen datan muuttaminen digitaaliseksi.
AVR LibC	C-kielen kirjasto mikrokontrollereille
CRC	Cyclic redundancy check, algoritmi, joka luo virheentarkistuskoodin
EAGLE	Piirilevyn suunnitteluohjelmisto
I/O-pinni	Liitin, jota voi käyttää sekä sisään- että ulostulona
ISM	Industrial, Scientific and Medical, lyhyen kantaman tietoliikennekäyttöön tarkoitettu lupavapaa taajuusalue
Bluetooth	Avoin standardi lyhyen kantaman langattomaan tietoliikenteeseen

1 JOHDANTO

Yhä useampia asuntoja rakennettaessa ja remontoitaessa pohditaan älylaitteiden yhdistämistä kodin toimintojen ohjaamiseen sekä tarkkailuun. Älykodin ohjauskeskusten tarjotessa langattoman ohjausmahdollisuuden puhelimen tai tabletin välityksellä saadaan sisustukseen sopimattomat ohjauspaneelit piiloon. Älykäs asunnon ohjaus mahdollistaa taloudelliset säästöt ohjailemalla toimintoja asukkaan paikallaolon mukaan. Kodin valvonta antaa huonokuntoisille ihmisille turvaa tarjoamalla helpon yhteydenoton läheisiin sekä hätänumeroon ja valvomalla mahdollisia paloa aiheuttavia kodinkoneita.

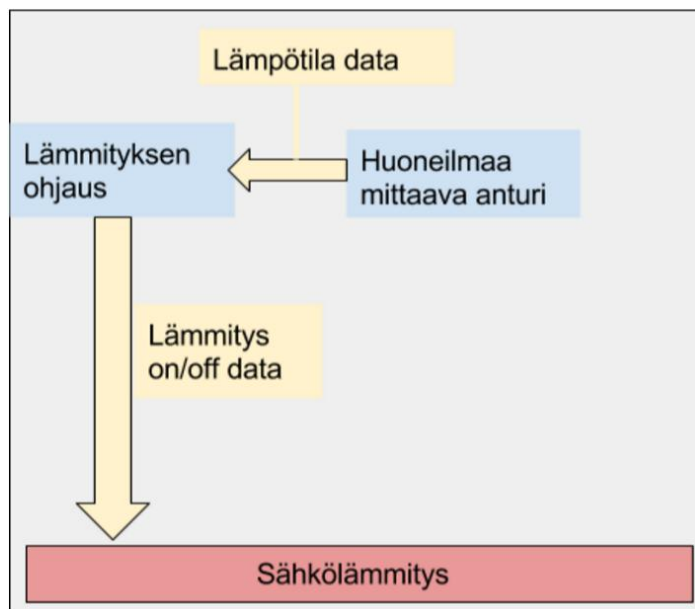
Tässä työssä on keskitytty kehittämään älykodeista tuttu tapa ohjata lämmitystä älylaitteella. Ohjaus tapahtuu antamalla älylaitteella komentoja mikroprosessorikortille, joka ohjaa lämmitysjärjestelmää Bluetooth-yhteyden välityksellä. Laitteen kehityksessä keskityttiin kehittämään halpa ohjauskeskus, jota olisi mahdollista laajentaa kattamaan useita toimintoja tulevaisuuden tarpeiden mukaan.

2 SÄHKÖLÄMMITYS

2.1 Lämmitys

Suomen ilmastossa kiinteistöjä tarvitsee lämmittää suurimman osan vuodesta, josta aiheutuu merkittävät kustannukset. Käytettyä energiamäärää pystytään vähentämään parantamalla asuntojen energiatehokkuutta sekä oikeanlaisella lämpötilan säätämällä. (Sähkölämmityksen peruskurssi)

Huonekohtaiseen sähkölämmitykseen voidaan käyttää joko sähköpattereita tai lattiamateriaalin alla sijaitsevia lämmitysvastuksia. Yksinkertaisimmillaan järjestelmässä on yksi lämpötila-anturi mittaamassa huoneen lämpötilaa, ja sen ollessa alhaisempi kuin haluttu lämpötila ovat sähkövastukset päällä lämmittämässä. Tällaisessa järjestelmässä on myös syytä käyttää hystereesiä, jolla vältetään turhalta päälle/pois kytkemiseltä. Huoneen lämpötilaa lämmitetään esimerkiksi 1°C korkeammalle kuin halutun lämpötilan asetus ennen lämmityksen pois kytkemistä. Lämmitys käynnistyy taas lämpötilan laskettua alle halutun.



Kuva 1. Yksianturisen sähkölämmityksen toimintaperiaate

2.2 Lämmityksen ohjaus

Sähkölämmityksen ohjaus hoidetaan kytkemällä se päälle lämpöä tarvittaessa hystereesi huomioon otettuna. Termostaatti hoitaa tämän käyttäjän valitessa halutun lämpötilan tilanteen mukaan. Mikroprosessorilla saadaan luotua älykäs termostaatti, jolla pystyy parantamaan energiatehokkuutta ja vähentämään käyttäjältä vaadittuja säätötoimenpiteitä ottamalla huomioon tilan käyttö ja vuorokaudenaika sekä ulkolämpötila.

3 ARDUINO

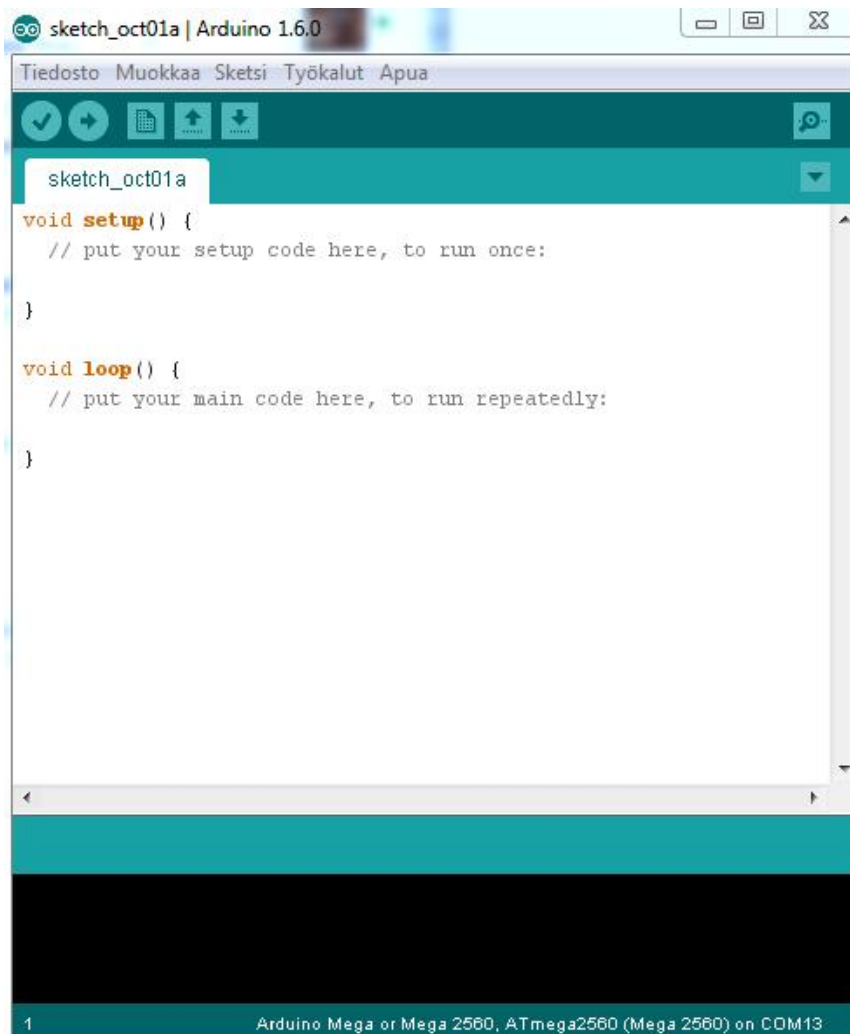
Arduino on kehitetty Ivrea Interaction Design Instituutissa helppokäyttöiseksi kehitysalustaksi nopeasti tehtäviin elektroniikan käytännön sovelluksiin ja prototyyppien rakenteluun. Arduinosta on useita eri malleja vaihteleviin käyttötarkoituksiin. Kehitysalusta valitaan käyttötarkoituksen mukaan, kuten vaadittujen I/O-pinnien määrän, muistin ja suoritustehon, fyysisen koon tai erityisten suunnittelukohteiden perusteella, kuten puettava elektroniikka tai esineiden Internet. (Arduino introduction)

Arduino-kehitysympäristö ja fyysinen laite ovat avointa lähdekoodia. Kehitysympäristö on julkaistu GPL-lisenssillä, joka sallii oikeudet käyttää, jakaa ja muuttaa lähdekoodia. Muutettu lähdekoodi on mahdollisen julkaisun tapauksessa jaettava samalla lisenssillä eteenpäin kuin alkuperäinen ohjelma. Kehitysalustasta on julkaistu piirilevyn tuottamiseen tarvittavat tiedostot Creative Commons Attribution Share-Alike -lisenssillä, joka mahdollistaa niin yksityisen kuin kaupallisen käytön, joiden ehtona on julkaista tehty työ samalla lisenssillä ja mainita alkuperäisen työn tekijä eli tässä tapauksessa Arduino. Arduino antaa ilmaiseksi kehitysalustojen EAGLE-tiedostot sekä kytkentäkaavion tuotantoa ja muutoksia varten.

Arduinosta on tullut erittäin suosittu kehitysympäristö harrastelijoiden keskuudessa halvan hinnan, helpon lähestyttävyyden ja hyvän saatavuuden vuoksi, mikä on helpottanut muodostumaan sen ympärille laajan yhteisön. Laajan yhteisön avulla aloittelijoille on saataville paljon apua ja harjoituksia helpottamaan harrastuksen aloittamista.

3.1 Kehitysympäristö

Arduinojen ohjelmointiin käytetään Arduinon omaa ohjelmointikieltä, joka perustuu C/C++ -kieleen. Kääntäjä linkittää käännöksen AVR:n LibC:tä vasten ja näin mahdollistaa kaikki sen käyttämät funktiot. (Arduino language reference)



Kuva 2. Arduinon kehitysympäristö

Arduinolta on saatavana ilmaiseksi helppokäyttöinen kehitysympäristö, jolla on mahdollista ohjelmoinnin lisäksi syöttää ja lukea sarjamuotoista dataa USB-väylän välityksellä. Kehitysympäristössä kirjoitettuja ohjelmia kutsutaan nimellä sketsi (sketch), jotka kirjoitetaan tekstieditorilla, ja niiden tiedostopäätteenä toimii .ino. Ohjelman alalaidassa olevaan mustaan alueeseen tulee ilmoitukset ohjelmakoodin kääntämisestä ja kääntämisen aikana mahdollisesti ilmenneet virheilmoitukset. Ennen sketsin lataamista kehitysalustalle valitaan työkaluvalikon kautta käytetyn levyn malli ja COM-portti, jossa se on kiinnitetty. (Arduino Environment)

Tiedosto-valikon kautta löytyy esimerkit-kohta, josta löytyy huomattava määrä aloittelijalle hyödyllisiä esimerkkejä lähtien yksinkertaisesta LED:in vilkutuksesta monipuolisiin robottien ohjauksiin.

3.2 Käytetty Arduino ja lisälaitteet

3.2.1 Laitteiston valinta

Arduino valikoitui tämän työn kehitysalustaksi helpon käytön ja halvan hankintahinnan vuoksi. Kaikki laitteet hankittiin Ebay:n kautta hintaa silmällä pitäen. Tästä johtuen osat ovat kiinalaisia kopioita. Säästetty summa on huomattava Arduino Megan maksaessa Suomesta tilattuna jo yksinään 50 euroa.

TAULUKKO 1. Laitteiston hankintahinnat

Laite	Hinta €
Arduino Mega 2560	16,75
LCD Keypad Shield	5,08
Bluetooth-moduuli HC-06	3,92
Yhteensä	25,75

3.2.2 Arduino Mega 2560

Arduino Mega 2560 on ATmega2560 mikrokontrollerin ympärille rakennettu kehitysalusta. Sen tärkeimpiä ominaisuuksia ovat 54 digitaalista I/O -pinniä, 16 analogista sisääntuloa ja USB-liitin ohjelmointia ja käyttöjännitettä varten. Piirillä on ohjelmistoa varten 256 kB flash-muistia.



Kuva 3. Arduino Mega 2560 -kehitysalusta (store.arduino.cc)

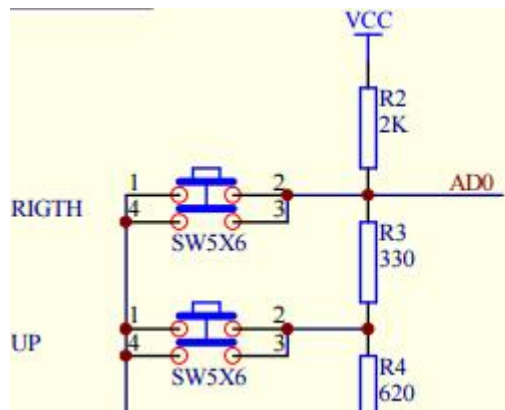
3.2.3 Bluetooth-moduuli HC-06

HC-06 on Bluetooth-yksikkö, joka lähettää ja vastaanottaa sarjamuotoista dataa. Laite käyttää Bluetooth-protokollaa v2.0+EDR. HC-06 on luokan 2 Bluetooth-laite, joka määrittelee sen maksimilähetystehoksi 2,5 mW (4 dBm) ja nimelliskantama on 10 metriä.

3.2.4 LCD keypad shield

Shieldit ovat lisälaitteita Arduinolle, jotka on suunniteltu liitettäväksi kehitysalustan päälle tarjoten samalla käytettävyyden sen alle jääviin käyttämättömiin pinneihin sekä virta- ja maapinneihin.

LCD keypad shield sisältää helposti liitettävän kaksi merkkiä korkean ja 16 merkkiä leveän näytön ja viisi sovelluksessa käytettävää painiketta sekä reset-painikkeen. Viisi ohjelmoitavaa painiketta ovat liitettynä analogiseen sisääntulopinniin A0 kehitysalustalla. Eri painikkeet tunnustetaan toisistaan jännitearvolla, joka saadaan yksinkertaisella jännitejaolla vastuksien avulla.



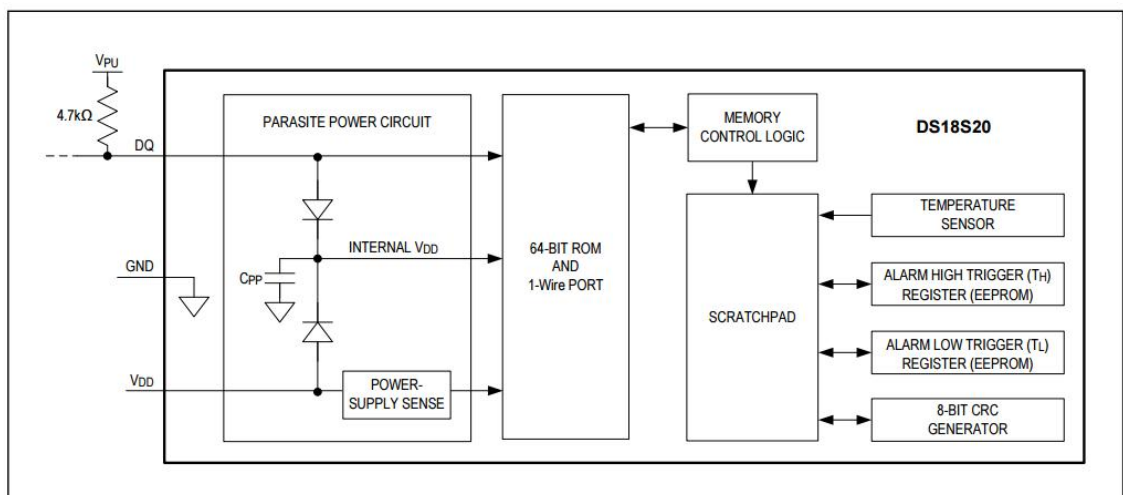
Kuva 4. Painikkeiden kytkentätapa (<http://www.dfrobot.com>)

4 LÄMPÖTILA-ANTURI

4.1 Anturi

Projektissa käytetyksi lämpötila-anturiksi valikoitui digitaalinen lämpötila-anturi Maxim DS18S20. Sillä on riittävät ominaisuudet ja tarkkuus kiinteistön lämpötilan mittaamiseen, ja kyseistä anturia sai Maximilta muutaman kappaleen ilmaisina näytteinä. Anturi antaa lämpötilan celsiusasteissa ja siinä on mahdollisuus hälytystoimintoon, johon käyttäjä voi ohjelmoida ylä- ja alarajat. Tiedonsiirtoon anturi käyttää 1-Wire väylää, jolle riittää yksi datalinja ja maa kommunikointiin mikroprosessorin kanssa. Kaikki väylässä liikkuva data ja komennot lähetetään sarjamuotoisena vähintään merkittävä bitti ensimmäisenä. (datalehti DS18S20)

Jokaisella anturilla on yksilöllinen 64-bittinen sarjanumero, jotta useita samaan datalinjaan kytkettyjä antureita voidaan erottaa toisistaan luotettavasti. Työssä käytetyn TO-92-paketoinnin lisäksi anturia on saatavilla kahdeksanpinnisenä pintaliitosversiona. Anturin lohkokaaviosta (kuva 5) näkyy työmuisti (scratchpad), joka on kahden tavun suuruinen alue väliaikaiseen lämpötiladatan säilöntään. Työmuistin avulla pääsee myös ylä- ja alarajahälytyksen muistiin, joka säilyttää tietonsa käyttöjännitteen kadotessa.



Kuva 5. DS18S20 lohkokaavio (datalehti DS18S20)

64-bittisessä sarjanumerossa kahdeksan vähiten merkittävää bittiä muodostavat 1-Wire tuoteperheen koodin: 10h. Seuraavat 48 bittiä muodostavat anturin sarjanumeron. Merkittävimmät kahdeksan bittiä muodostavat aiemmista biteistä laskemalla CRC-

tarkistussumman. Myös työmuistilla on CRC, joka lasketaan sen sisältämän väliaikaisen datan perusteella. CRC:n arvoa verrataan luettuun dataan ja näin tarkistetaan onko sisältöön tullut lukuvirheitä.

4.2 Lämpötilan mittaus

Mikroprosessori lähettää anturille "Convert T" -komennon, joka herättää anturin vähävirtaisesta lepotilasta ja käynnistää mittauksen sekä muutoksen analogisesta arvosta digitaalseksi. Lämpötila-arvo säilötään työmuistin lämpötilarekisteriin. Mittauksen jälkeen anturi palaa vähävirtaiseen lepotilaansa, jos käytössä ei ole ulkoista virtalähdettä. Mikroprosessori pyytää tietoa lämpötilan AD-muunnoksesta, joka voi enimmillään kestää 750 μ s, "read-time slots" -komennolla, johon anturi lähettää vastauksena 0-tilan muunnoksen ollessa kesken ja valmiin muunnoksen vastauksena lähetetään 1-tila. Tämä toiminta on käytettävissä ulkoisella virtalähteellä, jolloin anturi ei tarvitse virtaa datalinjasta muunnosta varten. Lämpötiladata säilötään rekisteriinsä 16-bitin mittaisena etumerkillisenä kahden komplementtilukuna.

TAULUKKO 2. Lämpötila-arvo suhteessa saatuun dataan

Lämpötila (°C)	Digitaalinen data (binääri)
+85,0	0000 0000 1010 1010
+25,0	0000 0000 0011 0010
0	0000 0000 0000 0000
-25,0	1111 1111 1100 1110

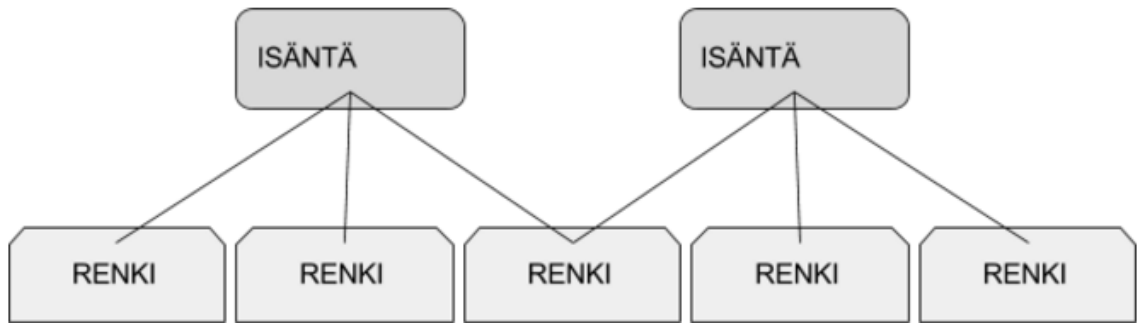
5 BLUETOOTH

Vuonna 1998 perustettiin työryhmä nimeltään Bluetooth Special Interest Group, johon kuuluivat teknologiayhtiöt Intel, IBM, Nokia, Toshiba ja Ericsson. Työryhmän tavoitteena oli kehittää avoin standardi lyhyen kantaman radiotaajuuksia käyttäville sovelluksille. Työryhmä julkaisi virallisen version 1.0 standardista kesäkuussa vuonna 1999, ja samana vuonna työryhmään liittyi yhtiötä kuten Microsoft ja Motorola. Bluetooth-tekniikan tavoitteena oli korvata kaapelit ja versio 1.1 keskittyi lähinnä sarjakaapelien korvaamiseen. Sovelluksista löytyi useita käytännön ongelmia, joista muutaman koettiin erittäin hyvin sopivan Bluetoothilla ratkaistaviksi. Tällaisia olivat esimerkiksi puhelimen yhdistäminen langattomasti sekä data- että lankaverkkoihin ennen matkapuhelimen yleistymistä, kaapelien korvaaminen puhelimiin, tietokoneiden lisälaitteisiin, digikameroihin sekä muihin laitteisiin, jotka tarvitsevat satunnaista tietoliikennettä. Myös kotiautomaatiolaitteiden säätöpaneelit voisivat toimia langattomasti, jolloin ne voisi piilottaa näkyvistä. (Langaton tiedonsiirto)

5.1 Bluetooth-arkkitehtuuri

5.1.1 Topologia

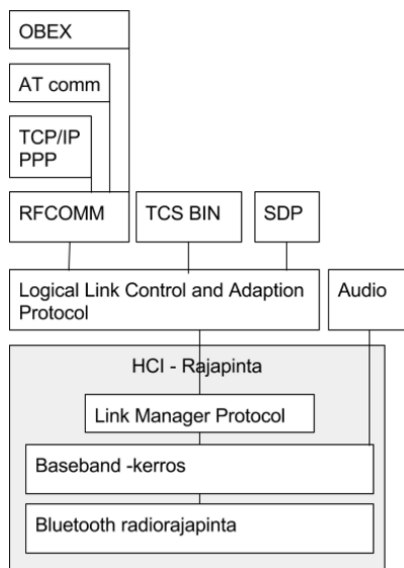
Bluetooth-tekniikassa on isäntälaitte (master), johon yksi tai useampi renkilaite (slave) yhdistyy. Tämän kaltaista verkkoa kutsutaan pikoverkoksi. Bluetoothin versiossa 1.1 aktiivisten renkien määrä on rajoitettu seitsemään, ja niiden lisäksi voi olla passiivisia renkejä, joilla ei ole tiedonsiirtotarpeita. Tämän kokoinen järjestelmä riittää hyvin kattamaan työpöydän tarpeet kuten hiiren, näppäimistön, tulostuksen ja kuulokkeet. Kotiautomaatiossa aktiivisina laitteina voisi toimia kulunvalvonta, huonekohtainen lämpötilan säätäminen ja kodinkoneiden toiminnan valvonta. Aktiivisten renkien määrän lisääntyessä on syytä käyttää scatternet-topologiaa, jotta saavutetaan haluttu määrä aktiivisia laitteita. Tällöin yksi renki yhdistää kahden isännän pikoverkot toisiinsa.



Kuva 6. Bluetooth scatternet

5.1.2 Bluetooth-protokollapino

Protokollapinin määrittäminen lukeutuu siirtoyhteyskerroksen ja fyysisen kerroksen toiminnot. Kuvassa 7 on Bluetooth-protokollapino, ja siitä näkee että kaikkia kerrosarkkitehtuurin peruseräitä ei noudateta. Bluetooth sallii tiettyjen toimintojen ohittamisen arkkitehtuurissa joitain palvelukerroksia. Tyypillinen esimerkki ohittamisesta on audiodatan siirto Bluetoothilla.



Kuva 7. Bluetooth-protokollapino

5.1.3 Bluetooth-arkkitehtuurin fyysinen kerros

Bluetooth käyttää 2,4 GHz:n ISM-taajuusalueita (industrial, science, medical; teollinen, lääketieteellinen ja tieteellinen käyttö), joka on vapaasti käytettävissä. Euroopassa käytetään

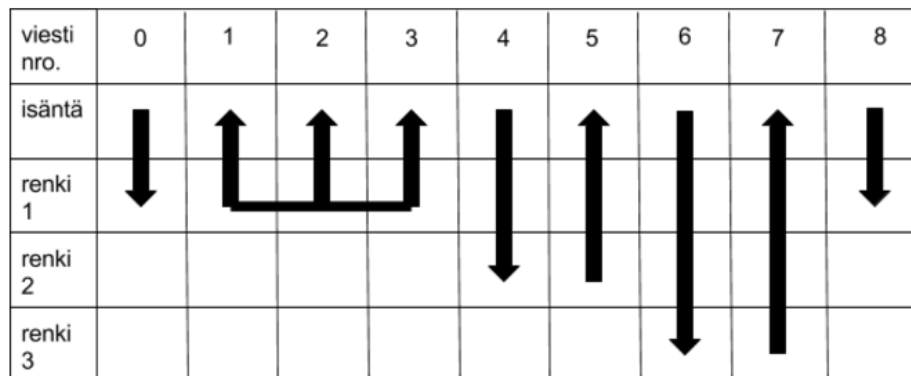
tetään taajuusaluetta 2400 MHz - 2483,5 MHz, joka sisältää 1 MHz välein 79 kanavaa ensimmäisen alkaessa taajuudesta 2402 MHz. Bluetoothin käyttämässä taajuusmodulaatiossa 1-bitti ilmaistaan taajuusalueella, joka sijaitsee keskitaajuudesta ylöspäin 140 - 175 kHz. 0-bitti ilmaistaan alueella, joka sijaitsee yhtä kaukana keskitaajuudesta kuin 1-bitti, mutta keskitaajuuden alapuolella. Spesifikaatio rajoittaa lähetystehot kolmeksi ryhmäksi taulukon (taulukko 1) mukaisiin luokkiin.

TAULUKKO 3. Bluetoothin lähetysteholuokat

Kantama	Teholuokka	Suurin teho	Pienin teho	Tehonsäätö
100 m	1	100 mW	1 mW	pakollinen
10 m	2	2,5 mW	1 mW	ei pakollinen
10 m	3	1 mW	1 mW	ei pakollinen

5.1.4 Kanavanvaraus

Bluetoothin kautta dataa siirretään 625 µs mittaisissa ajanjaksoissa. Siirrossa käytetään vuoro-suuntaista datasiirtoa, mikä tarkoittaa, että vain yksi osapuoli voi lähettää dataa kerrallaan. Lähetysvuoro vaihtuu osapuolelta toiselle siirtosuunnan mukaan. Menetelmää kutsutaan nimellä “time division duplex”. Lähetetyt sanomat numeroidaan väliltä 0 - $(2^{27} - 1)$. Pikoverkossa käytetään isännän ja renkien välillä vuorottelumenetelmää, jossa isäntä lähettää parillisella vuoronumerolla ja renki vastaa parittomalla vuoronumerolla. Käytetty vuorottelumenetelmä rajoittaa sanoman pituuden 1, 3 tai 5 aikavälin mittaiseksi.



Kuva 8. Isännän ja renkien välinen kiertokysely pikoverkossa

5.1.5 Taajuushyppely

Tiedonsiirto perustuu 625 μ s mittaisiin sanomiin, joista jokaisen sanoman jälkeen vaihdetaan kanavaa eli 1600 kertaa sekunnissa. Aikavälistä kuluu 220 μ s laitteen asettamiseksi oikealle taajuudelle, joka on noin kolmasosa aikavälistä. Taajuushyppelyssä on poikkeus sanomia varten, jotka eivät mahdu siirtymään yhden ajanjakson aikana. Tällaisessa tapauksessa sanoma siirtyy kokonaisuudessaan samalla taajuudella ja tämän jälkeen taajuushyppely palaa järjestyksen mukaiselle taajuudelle. Kolmen aikavälin mittaisessa kanavassa säästetään taajuuden asettamiseen kulunut aika kahdesti eli 440 μ s, joka voidaan hyödyntää tiedonsiirtoon. Jokaisella pikoverkolla on oma satunnaishyppelyjärjestyksensä, joka muodostetaan isäntälaitteen osoitteesta. Isäntälaitte vastaa verkon tahdistuksesta ja renkit korjaavat ajastustaan tarpeen mukaan isännän 625 μ s syklien mukaan. Isäntälaitteen lähettämät sanomat sisällyttävät sanomaansa synkronointikentän tätä tarkoitusta varten.

5.2 Yhteystyypit

5.2.1 Synkroninen yhteys

Bluetoothia käytetään myös sovelluksissa, joissa datan pitää kulkea jatkuvasti tasaisella siirtonopeudella. Yleisin tällaisista sovelluksista on äänen siirto. Synkronista yhteyttä käytettäessä molemmat osapuolet tarvitsevat säännöllisesti lähetysvuoron, jonka väliaika asetetaan yhteydenmuodostuksen alussa. Säännöllisestä lähetysvuorosta huolimatta renkin tarvitsee odottaa isännältä tulevaa pyyntöä ennen datan lähettämistä. Topologian vuoksi synkroninen yhteys on kaksisuuntainen isännän lähettäessä dataa samalla sanomalla kuin se pyytää renkiä lähettämään oman datansa.

5.2.2 Asynkroninen yhteys

Yhteysmuodon ollessa asynkroninen isäntä lähettää verkkonsa reneille sanomia, joihin renkien on mahdollista vastata datasanomalla. Jos renki ei tunnista vastaanottamansa

sanoman osoitetta, sillä ei ole oikeutta lähettää verkkoon mitään. Tässä suhteessa synkroninen kanava toimii erilaisella periaatteella, koska reneilläkin on tiedossa lähetysvuoronsa. Asynkroninen kanava tarjoaa mahdollisuuden lähettää kaikille vastaanottajille kuuluvaa viestiä kuten mainoksia ja tiedotteita, jotka lähetään isännän osoitteella.

5.3 Sanoma

Verkossa lähetetyillä sanomilla on perusmuoto, joka sisältää tahdistuskoodin, otsikon ja hyötykuorman, joka on kooltaan 0 - 2745 bittiä. Tahdistuskoodi (access code) sisältää alkutahdistuksen ja synkronointikentän. Sanoman otsikkokenttä sisältää vastaanottajan tarvitsemat tiedot vastaanottajan tunnistamiseen ja viestin tyyppiin. Otsikkoon on myös liitetty ohjaukseen, kuittaukseen ja järjestysnumeroon liittyvät tiedot. Hyötykuorma sisältää nimensä mukaisesti siirrettävän datan.

6 OHJAUSYKSIKKÖ

6.1 Langaton ohjaus

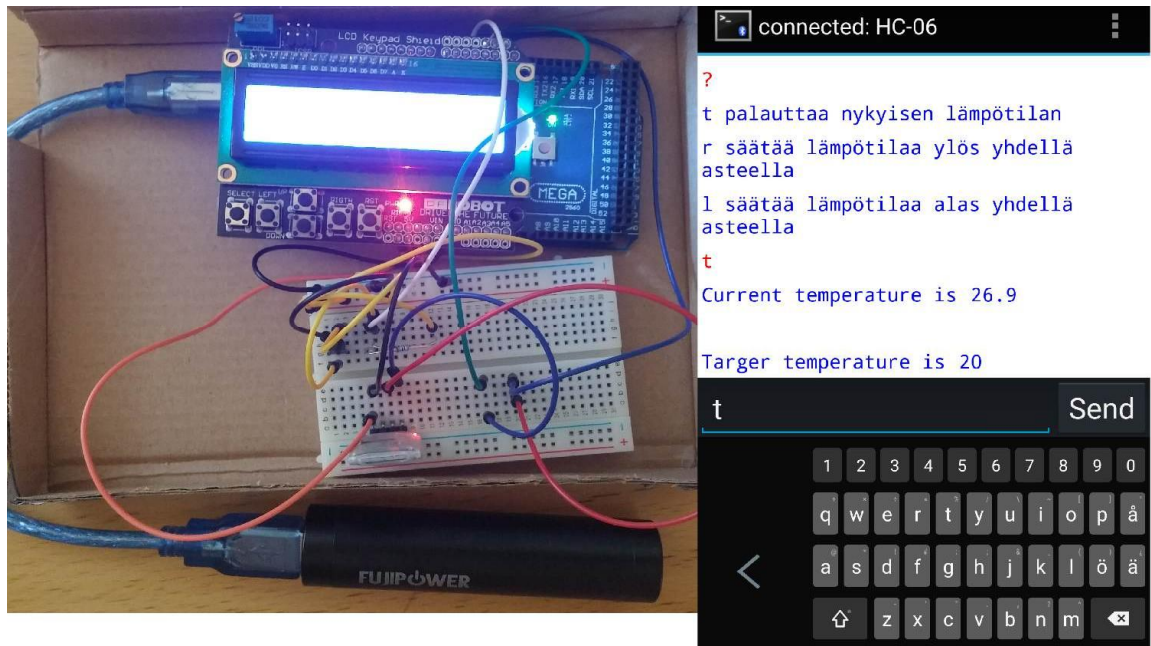
Laitetta etäkäytetään Android-käyttöjärjestelmään pohjautuvalla laitteella Bluetooth Terminal -ohjelmalla, joka yhdistää puhelimen ja Arduinon Bluetooth-moduulin kautta. Ohjelman käyttöliittymä puhelimesta on yksinkertainen komentorivi ja lähetettävänä komentoina käytetään yhden merkin viestejä, jotka Arduino lukee char -tyyppiseen muuttuun. Yhden merkin mittaiset komennot valittiin helpon muistettavuuden vuoksi sekä vähentämään kirjoitusvirheitä, joita pitkät merkkijonot helposti aiheuttaisivat.

TAULUKKO 4. Komennot laitteen ohjaukseen puhelimesta

t	Lähetää Arduinolta halutun ja nykyisen lämpötilan puhelimen komentoriville.
r	Nostaa haluttua lämpötilaa 1°C.
l	Laskee haluttua lämpötilaa 1°C.
?	Tulostaa komentoriville käyttöohjeet.

6.2 Paikallinen ohjaus

Arduinon on liitetty näyttö ja näppäimistö vaihtoehtoisesti käyttötapaa varten. Näyttö on korkeudeltaan kaksi merkkiä ja leveydeltään 16 merkkiä, jolloin siihen mahtuu tyypistettynä oleellinen informaatio. Paikallista ohjausta varten laitteesta käytetään neljää näppäintä. Ylös/alas -näppäimillä vaihdetaan eri valikoiden välillä ja vasen/oikea -näppäimillä muutetaan lukuarvoja. Kun näytöllä lukee Current Temp, siinä näkyy nykyinen lämpötila anturilta luettuna. Kun näytöllä lukee Wanted Temp -teksti, näytöltä on luettavissa haluttu lämpötila sekä tieto onko lämmitys päällä vai ei. Halutun lämmön näyttötilassa vasemmalla osoittavalla näppäimellä vähennetään kohdelämpötilaa ja oikealla osoittavalla näppäimellä nostetaan sitä yhden asteen muutoksilla painallusta kohden. Ohjelmaan on asetettu pieni viive näppäinten painamisen jälkeen, jotta yhtä painallusta ei tulkittaisi useammaksi painallukseksi.



Kuva 9. Valmis laite sekä käytetty Bluetooth Terminal -ohjelman käyttöliittymä älypuhelimien näytössä

6.3 Ohjelmakoodi

Bluetooth-funktiossa alustetaan muuttuja, johon tallennetaan väliaikaisesti Bluetooth-moduulilta saatu muuttujan arvo. Alustuksen jälkeen funktio tarkistaa onko sarjaliikennedataa luettavissa, ja sijoittaa sen muuttujaan. Lopuksi muuttujan arvoa verrataan esimääritettyihin komentoihin ja suoritetaan mahdollinen komento.

```
void bluetooth(){
char BT_arvo = '0';
if (Serial.available()){BT_arvo = Serial.read();}
if (BT_arvo == 't'){
  Serial.println("Current temperature is ");
  Serial.println(temp, 1);
  Serial.println("Targer temperature is ");
  Serial.println(wtemp, 1);
}
else if (BT_arvo == 'r'){wtemp = wtemp + 1;}
else if (BT_arvo == 'l'){wtemp = wtemp - 1;}
else if (BT_arvo == "?") {
  Serial.println("t palauttaa nykyisen lämpötilan ");
  Serial.println("r säätää lämpötilaa ylös yhdellä asteella");
  Serial.println("l säätää lämpötilaa alas yhdellä asteella");
}
}
```

Valikko-funktiossa tarkistetaan globaalista menu-muuttujasta, mikä valikko on valittuna, ja päivitetään näytölle oikeat valikkotekstit ja lämpötilatiedot.

```

void valikko(){
if (menu == 1){
  lcd.setCursor(0, 0);
  lcd.print("Wanted temp ");
  lcd.setCursor(0, 1);
  lcd.print(wtemp);
  if (wtemp >= temp){lcd.print(" C    ON ");}
  else{lcd.print(" C    OFF ");}
}
else if (menu == 2){
  lcd.setCursor(0, 0);
  lcd.print("Current temp ");
  lcd.setCursor(0, 1);
  lcd.print(temp, 1);
  lcd.print(" C    ");
}
if(wtemp >= temp){digitalWrite(Relay, HIGH);}
else{digitalWrite(Relay, LOW);}
}

```

Funktiossa napit luetaan muuttujaan nappien_luku funktion palautus, jonka avulla tunnistetaan painettu painike. Sen jälkeen switch-lausekkeessa suoritetaan painetun napin toiminnot. Painikkeita käytetään valikkojen välillä liikkumiseen ja kohdelämpötilan asettamiseen.

```

void napit(){
int painettu_nappi = nappien_luku();
switch (painettu_nappi)
{
  case btnDOWN: {
    menu = menu - 1;
    if (menu < 1){menu = 2;}
    delay(200);
    break;
  }
  case btnUP: {
    menu = menu + 1;
    if (menu > 2){menu = 1;}
    delay(200);
    break;
  }
  case btnLEFT: {
    if (menu == 1){wtemp = wtemp - 1;}
    delay(200);
    break;
  }
  case btnRIGHT: {
    if (menu == 1){wtemp = wtemp + 1;}
    delay(200);
    break;
  }
}
}
}

```

Ikuisesti toistuvassa loop-funktiossa luetaan lämpötila-arvo anturilta ja sen jälkeen kutsutaan funktiot bluetooth, napit ja valikko.

```
void loop(){
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  bluetooth();
  napit();
  valikko();
}
```


7 POHDINTA

Työn tavoitteena oli tutustua älykodin järjestelmän ohjauksen kehittämiseen toimivaksi laitteeksi. Lopputuloksena on älylaitteella ohjattava lämmitysjärjestelmän ohjausyksikkö. Työtä toteutettaessa ja suunniteltaessa ilmeni Arduino-kehitysalustan olevan kykenevä laite monipuolisen älykodin ohjaukseen ja monitorointiin. Toteutetussa laitteessa Bluetoothin kantama rajoittaa laitteen ohjausetäisyyttä noin kymmeneen metriin, mikä haittaa laitteen käytettävyyttä.

Työssä tehty laite on koottu koekytkentälevylle mahdollista jatkokehitystä varten. Koekytkentälevy osoittautui toimivaksi kokeiluihin, mutta lopulliseen käyttöön menevän tuotteen ratkaisuksi se on epäkelpo. Lisälaitteita sekä johtoja joutui useaan kertaan siirtämään, jotta ne saivat kunnollisen kontaktin.

LÄHTEET

Arduino. Introduction. Luettu 16.8.2015. <https://www.arduino.cc/en/Guide/Introduction>

Arduino. Enviroment. Luettu 4.9.2015. <https://www.arduino.cc/en/Guide/Environment>

Arduino. Language Reference . Luettu 25.8.2015.
<https://www.arduino.cc/en/Reference/HomePage>

Maxim integrated products inc. 2015. DS18S20 High-Precision 1-Wire Digital Thermometer. Tallennettu 15.7.2015.

<http://pdfserv.maximintegrated.com/en/ds/DS18S20.pdf>

Granlund, K. 2001. Langaton tiedonsiirto. Langattoman tiedonsiirron perusteet. 1. painos. Jyväskylä: Docendo Finland Oy.

Pirkko Harsia. 2005. Sähkölämmityksen peruskurssi. Päivitetty 3.12.2009. Luettu 2.7.2015.

<http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0505015/1119948180490/1119952690711/1119954346468/1119954387009.html>

LIITTEET

Liite 1. Lähdekoodi

1(4)

Lähdekoodi

```

#include <LiquidCrystal.h>           // LCD-näytön header
#include <OneWire.h>                 // 1-wire väylän header
#include <DallasTemperature.h>       // dallasin lämpötilanmittaus header

//määritellään näppäistön käyttämät muuttujat
#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5

OneWire oneWire(2);                 // käynnistää one wire protokollan (käyttää pinniä 2)
DallasTemperature sensors(&oneWire); // antaa 1-wire väylän referenssit dallastemp kirjastolle
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); // LCD-näytön käyttämät pinnit
int Relay = 13;                     //relettä ohjaavan pinnin määrittely

// Globaalit muuttujat
unsigned int menu = 1;               // muuttujan arvo kertoo näytettävän valikon
unsigned int wtemp = 20;             // Lämmityksen kohdelämpötilan muuttuja
float temp;                          // Tähän muuttujaan sijoitetaan anturilta luettu lämpötila

void setup()
{
  lcd.begin(16, 2);                  // määritetään LCD-näyttö
  lcd.setCursor(0, 0);               // asetetaan kursori ensimmäisen rivin ensimmäiseen merkkiin
  lcd.print("Temp control");         // kirjoittaa ohjelman nimen ensimmäiselle riville laitteen käynnistyessä
  sensors.begin();                   // käynnistää lämpötilan mittauksen, oletuksena 9 bitin resoluutiolla.
  Serial.begin(9600);                // käynnistää sarjaliikenteen ja asettaa sen nopeuden
  pinMode(Relay, OUTPUT);            //Lediä käytetään releen puutteessa testaukseen
}

void loop()
{
  sensors.requestTemperatures();     // komento pyytää anturilta lämpötilan
  temp = sensors.getTempCByIndex(0); // asetetaan muuttujaan temp lämpötila-arvo anturilta, joka on
                                     // indexinumroltaan 0 eli ensimmäisenä linjassa*/

  bluetooth();                       // bluetooth yhteys
  napit();                            // nappien toiminta
  valikko();                          // pidetään tekstivalikot ajan tasalla ja lämmityksen kytkeä
}

/*hoidaan laitteen ja puhelimen välinen bluetooth kommunikointi*/
void bluetooth()
{
  char BT_arvo = 'o';                 // muuttuja bluetooth-moduulilta luettua arvoa varten
  if (Serial.available())              //lue mahdollinen sarjaliikennedata (bluetoothilta)
  {
    BT_arvo = Serial.read();          //luetaan char-tyyppiseen muuttujaan bluetoothin kautta annettu komento.
  }
}

```

```

if(BT_arvo == 't')      //lähettää puhelimen komentoriville lämpötila-arvon
{
  Serial.println("Current temperature is "); // nykyinen lämpötila
  Serial.println(temp, 1);
  Serial.println("Target temperature is "); // haluttu lämpötila
  Serial.println(wtemp, 1);
}
else if (BT_arvo == 'r')      //nostaa haluttua lämpötilaa
{
  wtemp = wtemp + 1;
}
else if (BT_arvo == 'l')      //laskee haluttua lämpötilaa
{
  wtemp = wtemp - 1;
}
else if (BT_arvo == '?')      //tulostaa puhelimen näytölle käyttöohjeet
{
  Serial.println("t palauttaa nykyisen lämpötilan ");
  Serial.println("r säättää lämpötilaa ylös yhdellä asteella");
  Serial.println("l säättää lämpötilaa alas yhdellä asteella");
}
}

/*näytön tekstit*/
void valikko()
{
  if (menu == 1)
  {
    /* Valikko tilassa 1 näytöllä näytetään haluttu lämpötila
    ja ilmoitetaan oikealla alhaalla onko lämmitys päällä vai pois*/
    //tulostetaan kohdelämpötila näytölle
    lcd.setCursor(0, 0);
    lcd.print("Wanted temp ");
    lcd.setCursor(0, 1);
    lcd.print(wtemp);
    //verrataan haluttua lämpötilaa nykyiseen ja näytetään sen mukaan on/off
    if (wtemp >= temp)
    {
      lcd.print(" C   ON  ");
    }
    else
    {
      lcd.print(" C   OFF ");
    }
  }
  else if (menu == 2)
  {
    /*valikkotilassa 2 näytöllä näytetään nykyinen lämpötila,
    joka päivittyy jokaisella ohjelmakierrolla*/
    lcd.setCursor(0, 0);
    lcd.print("Current temp ");
    lcd.setCursor(0, 1);
    lcd.print(temp, 1);
    lcd.print(" C      ");
  }
  if(wtemp >= temp)
  {
    /*lämmitystarpeen tarkistus ja lämmityksen kytkeminen päälle/pois sen mukaan*/
    digitalWrite(Relay, HIGH);
  }
}

```

```

        else
        {
            digitalWrite(Relay, LOW);
        }
    }
    /*suoritetaan näppäinpainallusta vastaavat toiminnot*/
    void napit()
    {
        //napin numeerinen arvo
        int painettu_nappi = nappien_luku(); // painettu_nappi muuttuun painettua näppäintä vastaava numero 0-5 alun määritysten mukaan
        switch (painettu_nappi) // valikkoissa liikkuminen ja nappien toiminnot eri valikoissa
        {
            case btnDOWN: // ylös ja alas näppäimet kelaavat eri valikko vaihtoehtoja. Päästyään loppuun aloittaa valikkolistan alusta.
            {
                menu = menu - 1; //Siirrytään yhden valikon verran alaspäin
                if (menu < 1) //Jos mennään muuttujan arvoon, joka ei vastaa valikkoa siirrytään listan toiseen päähän
                {
                    menu = 2;
                }

                delay(200); //lisätään ohjelmaan viive napin painamisen jälkeen, jotta välttyttäisiin tulkitsemasta yhtä painallusta useaksi.*/

                break;
            }
            case btnUP: //toimii vastaavasti kuin alaspäin näppin, mutta päinvastaiseen suuntaan.
            {
                menu = menu + 1;
                if (menu > 2)
                {
                    menu = 1;
                }

                delay(200);
                break;
            }
            case btnLEFT:
            {
                if (menu == 1) //valikon ollessa tilassa 1 (wanted temp) napin painaminen vasemmalla vähentää halutun lämpötilan arvoa yhdellä.*/
                {
                    wtemp = wtemp - 1;
                }
                delay(200);
                break;
            }
            case btnRIGHT:
            {
                if (menu == 1) //valikon ollessa tilassa 1 (wanted temp) napin painaminen oikealla korottaa halutun lämpötilan arvoa yhdellä.*/
                {
                    wtemp = wtemp + 1;
                }

                delay(200);
                break;
            }
        }
    }
}

```

```
/*palauttaa painettua näppäintä vastaavan lukuarvon 0-5.
Jos mikään näppäin ei ole painettuna ohjelma palauttaa arvon 5 eli mikään näppäin ei ole painettuna.*/
int nappien_luku()
{
    //muuttuja napin AD-muunnetun arvon lukemiseen
    int napin_arvo = analogRead(0);          // Luetaan näppäinten arvo muuttujaan

    if (napin_arvo > 1000) { return btnNONE; } //Tulkitaan luetun arvon avulla, mikä näppäin on painettuna. btnNONE on ensimmäisenä nopeussyistä sen ollessa useimmiten oikeassa.
    else if (napin_arvo < 50) { return btnRIGHT; }
    else if (napin_arvo < 250) { return btnUP; }
    else if (napin_arvo < 350) { return btnDOWN; }
    else if (napin_arvo < 450) { return btnLEFT; }
    else if (napin_arvo < 700) { return btnSELECT; }

    return btnNONE; //Jos näppäimiä lukiessa tapahtuu virhe ohjelma palauttaa arvon 5, joka tarkoittaa ei mikään näppäin painettuna.
}
```