

Markus Liljestrand

Sovellus lämpötilan mittaamiseen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

22.4.2015

Tekijä Otsikko	Markus Liljestrand Sovellus lämpötilan mittaamiseen
Sivumäärä Aika	42 sivua + 1 liite 22.4.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaaja	Yliopettaja Petri Vesikivi
<p>Insinööriyön tarkoituksena oli suunnitella ja kehittää lämpötilan mittaamiseen liittyvä sovellus. Tavoitteena oli perehtyä työhön valitun mikrokontrollerin eri osa-alueisiin käytännön toteutuksen kautta, mikä tehtiin osana opinnäytetyötä.</p> <p>Insinööriyössä perehdyttiin mikrokontrollerin tarjoamiin mahdollisuuksiin ideasta lopulliseen versioon asti ja siihen miten sen tuottama lämpötilatieto esitetään dynaamisesti mobiilissa päätelaitteessa. Sovellus jaettiin kahteen osaan, mikrokontrollerin sketsi-ohjelmaan ja mobiilioptimoituun lämpötilaa mittaavaan verkkosovellukseen. Sketsi-ohjelma ohjelmoitiin mikrokontrollerin omalla ohjelmointikielellä, joka perustuu C++-ohjelmointikieleen. Mobiilioptimoidussa lämpötilaa mittaavassa verkkosovelluksessa hyödynnettiin PHP-ohjelmointikieltä. PHP-ohjelmointikielen lisäksi verkkosovelluksessa käytettiin CSS3-tyyliohjeita, HTML5-merkintäkieltä ja jQuery-kirjastoa, joilla pystyttiin luomaan verkkosovellukseen dynaamisuutta. MySQL-tietokantaa käytettiin sovelluksen tietokantana.</p> <p>Sovelluksen tekeminen aloitettiin opinnäytetyön tekijän omasta harrastuneisuudesta ja kiinnostuksesta mikrokontrolleihin ja niihin perustuviin kotiautomaatoratkaisuihin. Insinööriyön tuloksena valmistui mobiilioptimoitu lämpötilaa mittaava verkkosovellus, joka lukee tietokantaan tallennettua lämpötilatietoa. Lämpötilatieto tuotetaan mikrokontrolleriin kytketyllä lämpötila-anturilla ja tallennetaan tietokantaan. Insinööriyö osoitti, että mikrokontrollerilla ja siihen liitettävillä moduuleilla voidaan rakentaa mielenkiintoisia kotiautomaatoratkaisuja. Niillä voidaan säästää energiaa, lisätä mukavuutta ja turvallisuutta kotona ja loma-asunnoissa.</p>	
Avainsanat	Arduino, lämpötila, anturi, mobiili, sovellus

Author Title	Markus Liljestrand Application to measure the air temperature
Number of Pages Date	42 pages + 1 appendix 22 April 2015
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Petri Vesikivi, Principal Lecturer
<p>The goal of the study was to design and develop an application for measuring air temperature. The aim was to become more familiar with all the various functionalities of the microcontroller.</p> <p>The study presents the development steps of the application using the microcontroller from the first idea to the final release and how the temperature value created by microcontroller is presented dynamically in a mobile device. The application was divided into two parts, a sketch program and a mobile optimized web application to measure the air temperature. The sketch program was programmed with the microcontrollers own programming language which is based on the C++ programming language. The optimized mobile web application to measure the air temperature was programmed with PHP. CSS3 style sheets, the HTML5 markup language and the jQuery library were also used in the web application which made it possible to create dynamic functionality into the web application. The MySQL database was used as the database of the application.</p> <p>The development of the application began as a hobby and general interest in microcontrollers and home automation solutions based on them. As a result of the study, a mobile optimized application to measure the air temperature was designed. The application reads temperature values stored in its database. The temperature values is read from the temperature sensor connected to the microcontroller and stored into a database. The study showed that the microcontroller and its different modules can be used to build interesting home automation solutions. These solutions can save energy, increase comfort and safety at home and holiday houses.</p>	
Keywords	Arduino, temperature, mobile, sensor, application

Sisällys

Lyhenteet

1	Johdanto	1
2	Käytetyt tekniikat ja teknologiat	2
2.1	Arduino-mikrokontrolleri	2
2.2	Arduino WiFi Shield -moduuli	11
2.3	Arduinon käyttöönotto	16
2.4	Arduino WiFi Shield -moduulin käyttöönotto	17
2.5	Lämpötila-anturi	18
2.6	Mobiilioptimoitu lämpötilaa mittaava verkkosovellus	21
3	Käytetyt sovelluskehitystyökalut	27
3.1	Arduino-editori	27
3.2	Notepad++-tekstieditori	28
4	Sovellus lämpötilan mittaamiseen	29
4.1	Sovelluksen arkkitehtuuri	29
4.2	Tietokanta	30
4.3	Mobiilioptimoitu lämpötilaa mittaava verkkosovellus	31
5	Lähdekoodin piirteitä	34
5.1	Sovelluksessa suoritettu kutsut	36
5.2	Arduinon sketsi-ohjelma	37
6	Yhteenveto	39
	Lähteet	41

Liitteet

Liite 1. Arduino WiFi Shield -moduulin firmware-ohjelman päivitys

Lyhenteet

EEPROM	Electronically Erasable Programmable Read-Only Memory. Haihtumaton puolijohdemuisti, jota voidaan uudelleen kirjoittaa noin 10 000 – 100.000 kertaa. EEPROM-muistia käytetään pääasiassa asetustietojen tallentamiseen mikroprosessorin tai mikrokontrollerin sisältävissä laitteissa.
SRAM	Static Random Access Memory. Staattinen RAM-muisti on puolijohdetekniikalla toteutettu muistityyppi.
USB	Universal Serial Bus. Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.
FTDI	Future Technology Devices International. Skotlantilainen yksityisomistuksessa oleva puolijohdevalmistaja, joka on erikoistunut erityisesti USB-tekniologiaan.
LED	Light-Emitting Diode. Puolijohdekomponentti, joka säteilee valoa, kun sen läpi johdetaan sähkövirtaa.
IEEE	Institute of Electrical and Electronics Engineers. Kansainvälinen tekniikan alan järjestö.
SPI	Serial Peripheral Interface. Motorolan kehittämä tiedonsiirtostandardi.
SSID	Service Set Identifier. Langattoman lähiverkon verkkotunnus.
UDP	User Data Protocol. Yhteydetön protokolla, joka ei vaadi yhteyttä laitteiden välillä, mutta mahdollistaa tiedostojen siirron.
IDE	Integrated Development Environment. Integroitu ohjelmointiympäristö, joka sisältää yksinkertaisimmillaan tekstieditorin ja ohjelmointikielen kääntäjän.

1 Johdanto

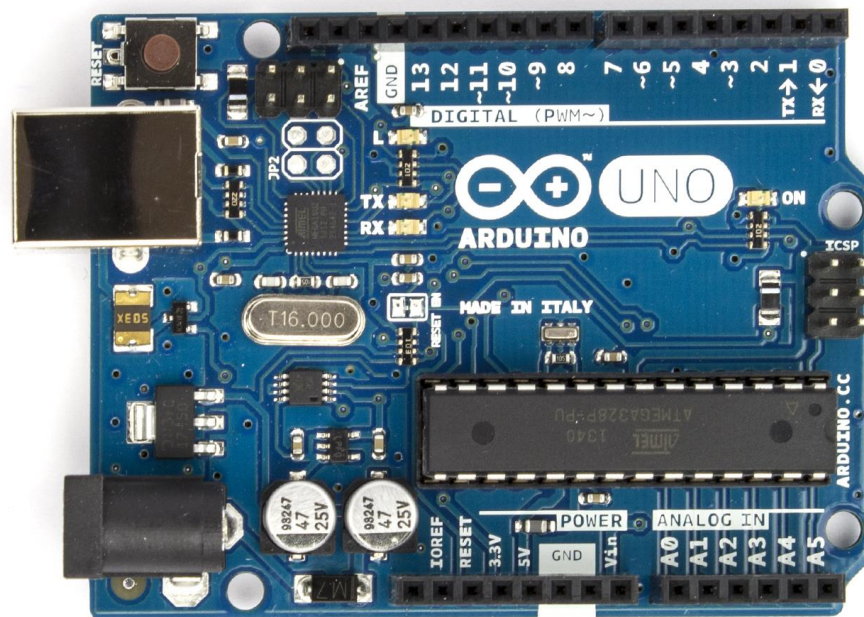
Internet of Things (IoT) eli esineiden Internet on herättänyt tavallisissa kuluttajissa halun kehittää erilaisia elämää helpottavia sovelluksia. Itse esineiden Internet -konseptista on virallisesti alettu puhua jo vuodesta 1999, kun Kevin Ashton ehdotti termiä pitämässään esityksessä. Tämän suuntaisia ajatuksia on tosin esitetty jo vuodesta 1991 lähtien. Teollisuudessa teollinen Internet tarkoittaa Internetin ja perinteisten teollisuustuotteiden yhdistämistä siten, että esineille annetaan tunnistettava identiteetti ja ne viestivät keskenään. Tämä perusajatus on myös toiminut lähtökohtana kuluttajien halussa kehittää erilaisia sovelluksia ja sovellutuksia omiin tarpeisiinsa. Puhutaan niin sanotusta DIY-ilmiöstä eli Do It Yourself -ilmiöstä. Tätä ilmiötä tukemaan on markkinoille tuotettu erilaisia pieniä tietokoneita ja tietokoneisiin kytkettyjä mikrokontrollereita. Niitä käytetään kehitysalustana, ja niihin voidaan kytkeä esimerkiksi erilaisia antureita ja liikeilmaisimia. Ne voivat toimia myös palvelimina, joihin otetaan yhteyttä erilaisilla sovelluksilla. [1.]

Sain ajatuksen tehdä opinnäytetyönä sovelluksen, joka mittaa lämpötilaa mikrokontrolleria hyödyntämällä. Ajatus syntyi saunaa lämmittäessä; sovellus hälyttäisi, milloin saunaan voisi mennä. Aloin tutkia eri mahdollisuuksia. Päädyin harkitsemaan kahta vaihtoehtoa: Raspberry Pi -tietokonetta tai Arduino-mikrokontrolleria. Lopulta valitsin Arduino-mikrokontrollerin siihen saatavien moduulien ja erilaisten antureiden helpon saatavuuden vuoksi. Tässä insinööriyössä käytetään Arduino WiFi Shield -moduulia, jolla Arduino voidaan yhdistää langattomaan lähiverkkoon. Näin Arduinoon kytkettyä lämpötilaa mittaavaa anturia hyödyntämällä lämpötilatieto välitetään Internetin välityksellä tietokantaan. Tietokannasta se sitten välitetään mobiilioptimoituun verkkosovellukseen.

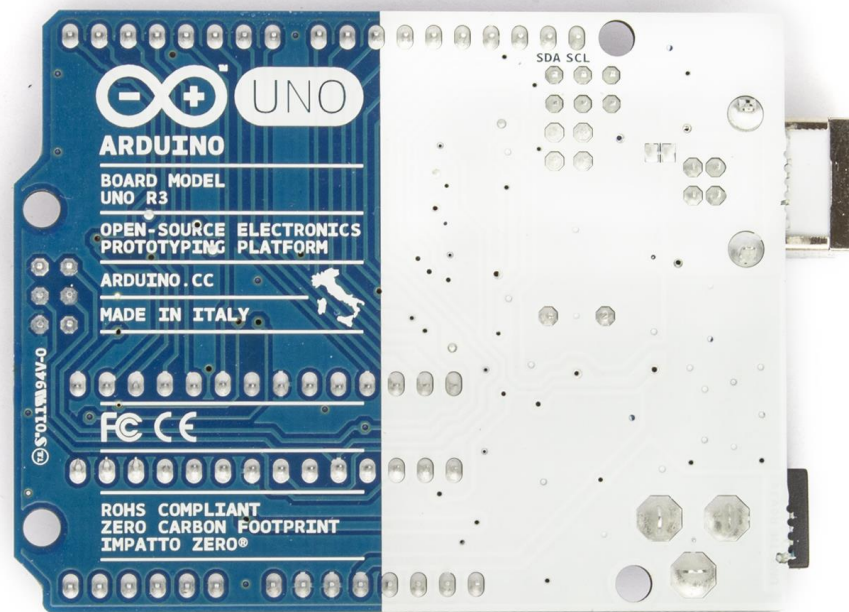
2 Käytetyt tekniikat ja teknologiat

2.1 Arduino-mikrokontrolleri

Oleellinen osa lämpötilamittausta on Arduino-mikrokontrolleri. Arduino on avoimeen laitteistoon perustuva mikrokontrolleri-elektroniikka-alusta ja ohjelmointiympäristö. Arduino perustuu 8-bittiseen Atmel AVR -mikrokontrolleriin, jonka pinneihin voi kytkeä erilaisia antureita, moottoreita, LED-valoja ja muita komponentteja. [2.] Kuvissa 1 ja 2 esitetään Arduino Uno R3 -mikrokontrolleri edestä ja takaa.



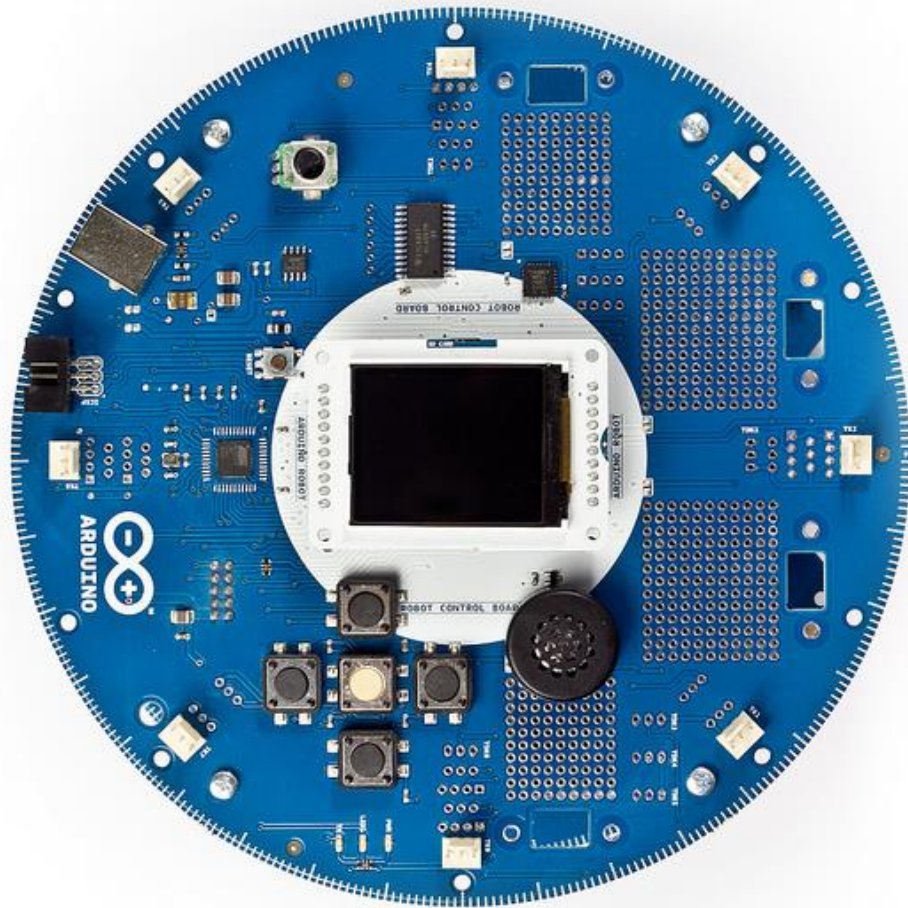
Kuva 1. Arduino Uno R3 -mikrokontrolleri edestä [2].



Kuva 2. Arduino Uno R3 -mikrokontrolleri takaa [2].

Ensimmäisen Arduino-prototyypin kehitti vuonna 2005 Massimo Banzi, joka toimi apulaisprofessorina Interactive Design Institute -instituutissa Ivreassa, Italiassa. Myöhemmin kehitystiimiin liittyivät David Cuartielles, Gianluca Martino, Tom Igoe, ja David Milles. Nimensä mikrokontrolleri sai Ivreassa sijaitsevan ”Bar Di Re Arduino” -pubin mukaan. Alusta asti oli selvää, että Arduino olisi nimenomaan avoimen lähdekoodin tuote. Arduino on lisenssoitu Creative Common -lisenssillä, jota tyypillisesti käytetään kirjallisuuden ja musiikin alalla. Arduinon kehitykseen vaikutti myös voimakkaasti ajatus, että sen tulisi olla mahdollisimman edullinen. Näin se olisi jokaisen kiinnostuneen opiskelijan ja kuluttajan ulottuvilla. [3.]

Varsinaisia Arduino-mikrokontrolleja on useita eri mallia, jotka eroavat toisistaan muun muassa keskusmuistin ja Flash-muistin sekä analogisten ja digitaalisten pinnien määrässä. Näitä malleja ovat muun muassa Uno, Leonardo ja Due. Arduinosta on myös erilaisia variantteja, joissa mikrokontrolleriin on valmiiksi asennettu esimerkiksi pyörät, kuten Arduino Robot -mikrokontrollerissa (kuvat 3 ja 4). [4.]

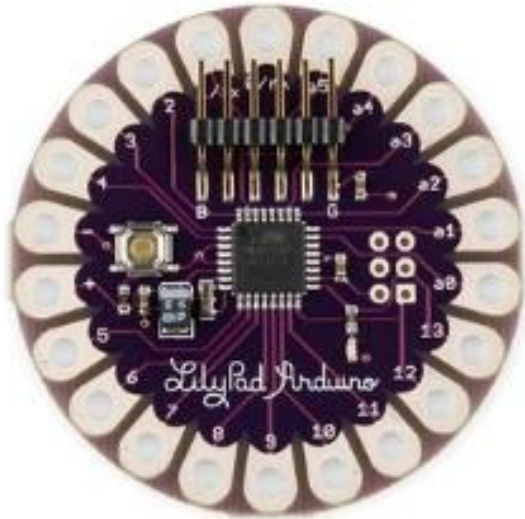


Kuva 3. Arduino Robot -mikrokontrolleri päältä.



Kuva 4. Arduino Robot -mikrokontrolleri alta.

Arduino-mikrokontrollereista on myös tehty hyvin pieniä versioita, kuten esimerkiksi Lily-Pad Arduino (kuva 5), joka voidaan ommella tekstiiliin ja näin luoda niin sanottuja puettavia e-vaatteita. [4.]



Kuva 5. LilyPad Arduino -mikrokontrolleri.

Arduino-mikrokontrollereihin on mahdollista kytkeä erilaisia moduuleita (shields), joilla voidaan lisätä Arduinon toiminallisuutta. Niitä ovat muun muassa GSM Shield, Motor Shield ja USB Host Shield. [4.]

Tässä insinööriyössä käytetään Arduino Uno R3 (Revision 3) -mallia . Tämän mallin nimi ”Uno” viittaa Arduino 1.0 -versioon. Sen tekniset tiedot ovat taulukon 1 mukaiset.

Taulukko 1. Arduino Uno R3 (Revision 3) [2].

Mikrokontrolleri	ATmega328P
Käyttöjännite	5 V
Sisääntuleva jännite (suositeltu)	7–12 V
Sisääntuleva jännite (rajoitettu)	6–20 V
Flash-muistia (ATmega328)	32 kB
Kellotaajuus	16 MHz
EEPROMia (ATmega328)	1 kB
SRAMia (KiB)	2
Digitaalisia siirtopinnejä	14
Analogisia pinnejä	6

USB	Atmega16U2
Mitat (mm)	68,6 mm x 53,3 mm
Paino	25 g

Arduino saa käyttöjännitteensä joko tietokoneesta USB-kaapelin avulla tai erillisellä virtalähteellä. Virtalähteenä voidaan käyttää joko AC/DC-adapteria tai yhdeksän voltin paristoa. Huomioitavaa on, että Arduino voi tulla epävakaa, jos jännite on vähemmän kuin seitsemän voltia. Mikäli jännite on enemmän kuin 12 voltia, Arduino voi ylikuumentua ja näin vahingoittua. Arduino Uno R3 -mikrokontrolleri eroaa aiemmista malleista USB-portin osalta. Se ei hyödynnä FTDI USB-to-serial -ajuria vaan käyttää ohjelmoitua Atmega16U2 USB-to-serial -konvertteria. [2; 21.]

Syötteet ja tulosteet

Arduino Unon jokaista 14 digitaalista siirtopinniä voidaan käyttää joko syötteenä tai tulosteena käyttämällä `pinMode()`-, `digitalWrite()`- ja `digitalRead()`-funktioita. Nämä siirtopinnit toimivat viiden voltin jännitteellä. Arduino Unon kuutta analogista siirtopinniä voidaan ohjata `analogReference()`-funktioilla.

Digitaaliseen siirtopinniin voidaan syöttää tai siirtopinnistä voi lukea vain kahta arvoa: high ja low. High-arvon merkitys eroaa riippuen siitä, käytetäänkö digitaalista siirtopinniä INPUT-moodissa vai OUTPUT-moodissa. Kun siirtopinni on asetettu INPUT-moodiin `pinMode()`-funktioilla tai OUTPUT-moodiin `digitalRead()`-funktioilla, Arduino tulkitsee siirtopinnin arvon high-arvoksi seuraavasti:

- Viiden voltin kortilla siirtopinnin arvo on enemmän kuin kolme voltia.
- 3,3 voltin kortilla siirtopinnin arvo on enemmän kuin kaksi voltia.

Jos siirtopinni on asetettu `pinMode()`-funktioilla INPUT-moodiin, voidaan siirtopinni saada antamaan high-arvo käyttämällä `digitalWrite()`-funktioita.

Low-arvolla on myös kaksi eri merkitystä riippuen siitä, onko siirtopinni asetettu INPUT- vai OUTPUT-moodiin. Kun siirtopinni on asetettu INPUT-moodiin `pinMode()`-funktioilla ja sitä luetaan `digitalRead()`-funktioilla, Arduino tulkitsee siirtopinnin arvon low-arvoksi seuraavasti:

- Viiden voltin kortilla siirtopinnin arvo on vähemmän kuin kolme voltia.
- 3,3 voltin kortilla siirtopinnin arvo on vähemmän kuin kaksi voltia.

Kun pinni on asetettu OUTPUT-moodiin `pinMode()`-funktiolla ja low-arvoksi `digitalWrite()`-funktiolla, pinni antaa arvon nolla voltia sekä viiden voltin että 3,3 voltin kortilla.

Arduino Unon (kuva 6) siirtopinnestä käytetään oletusarvoisesti syötteenä, jolloin siirtopinniä ei erikseen tarvitse määritellä syötteenä `pinMode()`-funktiolla. Kun siirtopinniä käytetään syötteenä, sen sanotaan olevan korkea impedanssi -tilassa (high-impedance state). Impedanssi kuvaa virtapiirin vaihtovirralla aiheuttamaa vastusta. Kun siirtopinniä käytetään tulosteena, sen sanotaan olevan matala impedanssi -tilassa (low-impedance state). [2.]

`pinMode()`-funktio

Siirtopinnestä käytetään oletusarvoisesti digitaaliseen syötteeseen (INPUT), mutta siirtopinnin käyttöä voi muuttaa `pinMode()`-funktiolla. Tällä funktiolla siirtopinniä voidaan käyttää joko digitaaliseen syötteeseen tai tulostamiseen. Funktion syntaksi on `pinMode(pin, mode)`, jossa `pin` on siirtopinnin numero, jota halutaan käsitellä, ja `mode` määrittää sen, halutaanko siirtopinniä käyttää digitaalisena syötteenä (INPUT) vai digitaalisena tulosteena (OUTPUT) vai halutaanko siirtopinniä käyttää digitaalisena syötteenä ja kytkeä se sisäiseen resistoriin eli vastukseen (INPUT_PULLUP). Resistori on elektroniikkakomponentti, joka vastustaa tasa- ja vaihtovirran kulkua. [2.]

`digitalWrite()`-funktio

Siirtopinnille voidaan kirjoittaa joko high- tai low-arvo käyttämällä `digitalWrite()`-funktiota. Funktion syntaksi on `digitalWrite(pin, value)`, jossa `pin` on siirtopinnin numero, jota halutaan käsitellä, ja `value` määrittää, kirjoitetaanko siirtopinnille high- vai low-arvo. [2.]

`digitalRead()`-funktio

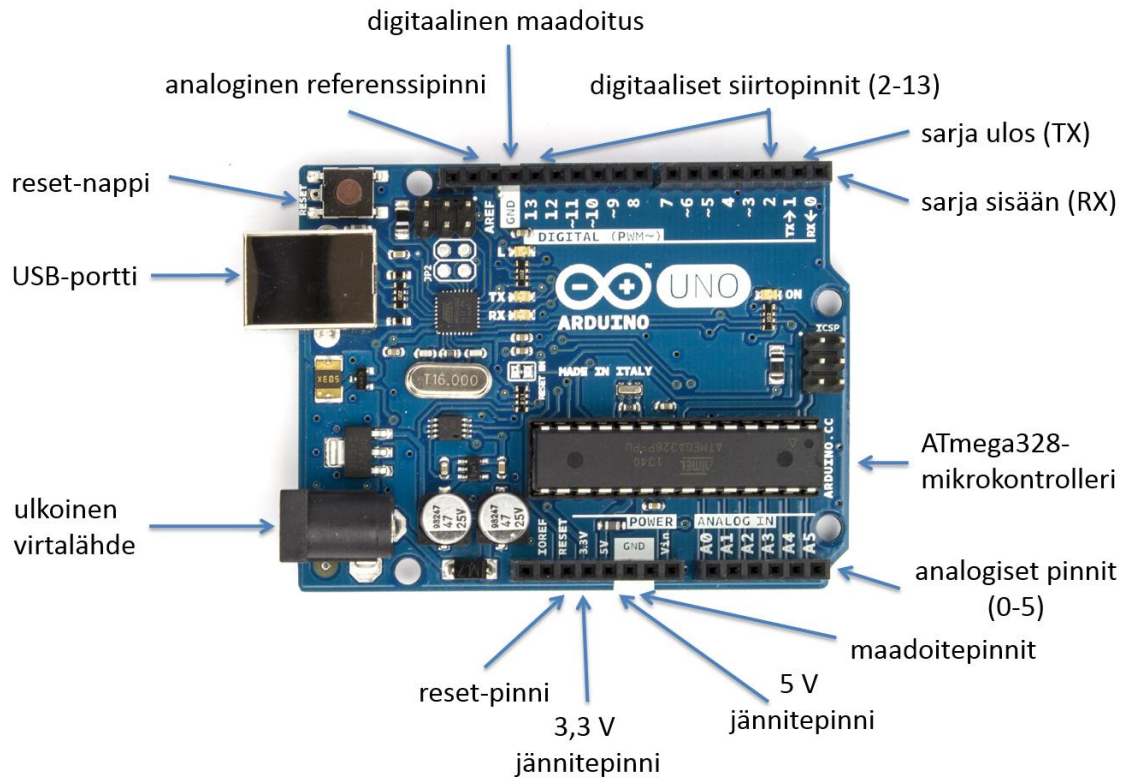
Siirtopinniltä välitetyt arvoja voi lukea käyttämällä `digitalRead()`-funktiota. Funktion syntaksi on `digitalRead(pin)`, jossa `pin` on siirtopinnin numero, jota halutaan käsitellä. Luettavat arvot voivat olla joko high- tai low-arvo.

Ohjelmointiesimerkissä 1 hyödynnetään edellä mainittuja funktioita. Siinä LED on kytkettynä siirtopinniin numero 13. Siirtopinniin numero 7 on taas asetettu kytkin, jolla voidaan kytkeä LED päälle tai sammuttaa se. Siirtopinni 13 asetetaan tulosteeksi antamalla pinMode()-funktion mode-parametriksi OUTPUT-arvo. Siirtopinni 7 asetetaan syötteenä. LED-valo voidaan sytyttää ja sammuttaa lukemalla siirtopinnin 7 arvoja. Nämä arvot kirjoitetaan siirtopinnin 13 arvoksi. [2.]

```
int ledPin = 13;
int butPin = 7;
int onoff;
void setup()
{
    pinMode(ledPin, OUTPUT);
    pinMode(butPin, INPUT);
}
void loop()
{
    onoff = digitalRead(butPin);
    digitalWrite(ledPin, onoff);
}
```

Esimerkki 1. LED-valon sytyttäminen ja sammuttaminen.

Kuvassa 6 on havainnekuva Arduino Uno R3 -mikrokontrollerin eri komponenteista.



Kuva 6. Arduino Uno R3 -mikrokontrolleri.

Arduinon ohjelmointi

Arduinon kehitysympäristö IDE on ohjelmoitu kokonaan Javalla. Kehitysympäristö sisältää ohjelmointieditorin, jossa on muun muassa syntaksitarkistus. Arduino-ohjelmat voidaan kirjoittaa joko C- tai C++-ohjelmointikielellä. Arduino IDE sisältää ohjelmointikirjaston nimeltään "Wiring", joka on peräisin alkuperäisestä Wiring-projektista. Arduino-ohjelmat ovat yksinkertaisia: ne sisältävät vain kaksi funktiota, `setup()`- ja `loop()`-funktiot. `Setup()`-funktio ajetaan aina kun Arduino käynnistetään. Funktio käytännössä alustaa Arduinon halutuilla parametreilla. `Loop()`-funktiota suoritetaan niin kauan, kunnes Arduino sammutetaan. Arduinon ohjelmointi itsessään jakautuu kolmeen osaan: struktuuriin, arvoihin (muuttujat ja vakiot) ja funktioihin. Struktuuri sisältää edellä mainittujen `setup()`- ja `loop()`-funktioiden lisäksi ohjelmoinnissa yleisesti käytettäviä tyypillisiä valinta-, toisto- ja hyppylauseita, kuten `if/else`-, `do/while`- ja `break/continue`-lauseet. Niiden lisäksi struktuuri sisältää muun muassa tunnettuja matemaattisia, yhdistely- ja vertailuoperaattoreita. Muuttujat sisältävät erilaisia vakioita, tietotyyppisiä ja konvertointiin tarvittavia funktioita. Funktiot sisältävät digitaalisia, analogisia ja kehittyneempiä I/O-funktioita. Digitaalisia ja analogisia funktioita käsiteltiin aiemmin tässä luvussa. Kehittyneemmillä I/O-funktioilla

voidaan luoda esimerkiksi ääntä, kuten esimerkiksi `tone()`-funktiolla. Funktioihin kuuluvat myös erilaiset ajan käsittelyyn tarkoitetut funktiot, kuten esimerkiksi `millis()`- ja `delay()`-funktioita. Näiden lisäksi funktiot sisältävät lukuisia matemaattisia ja trigonometrisia funktioita, kuten esimerkiksi `min()`-, `max()`- ja `sin()`-funktioita. [2.]

Serial- ja Stream-kirjastot

Sarjaporttien ohjelmoimiseen on käytössä Serial- ja Stream-luokat. Serial-luokan käytetyimpiä funktioita ovat

- `Serial.available()`-funktio, joka palauttaa sarjaportista luettavien merkkien määrän
- `Serial.begin()`-funktio, joka asettaa tiedonsiirron nopeuden
- `Serial.end()`-funktio, joka lopettaa tiedonsiirron
- `Serial.print()`-funktio, joka tulostaa dataa haluttuun sarjaporttiin
- `Serial.read()`-funktio, joka lukee dataa halutusta sarjaportista
- `Serial.write()`-funktio, jolla kirjoitetaan sarjaporttiin.

Stream-perusluokkaa käytetään merkki- ja binääripohjaiseen siirtoon. Stream-luokkaan perustuvat kirjastot ovat Serial-, Wire-, Ethernet Client-, Ethernet Server- ja SD-kirjastot. [2.]

2.2 Arduino WiFi Shield -moduuli

Arduino voidaan kytkeä Internetiin joko Ethernet Shield-, WiFi Shield- tai GSM Shield -moduulilla. Tässä insinööriyössä käytetään Arduino WiFi Shield -moduulia (kuvat 7 ja 8). Arduino WiFi Shield käyttää 802.11-tekniikkaa, joka on IEEE:n standardi langattomille lähiverkoille. Arduino liitetään Internetiin WiFi-kirjaston avulla, joka tukee sekä WEP- että WPA2 Personal -salausta. Arduino keskustelee WiFi Shieldin kanssa käyttämällä SPI-väylää. SPI-väylä sijaitsee siirtopinneissä 11, 12 ja 13. Siirtopinni 7 on varattu Arduinon ja WiFi Shieldin kättelyyn. Kättely kertoo vastaanottavalle laitteelle, minkälaista dataa laite lähettää ja millä nopeudella. WiFi-kirjasto on hyvin samantyyppinen kuin Ethernet-kirjasto, ja monet komennot ovat molemmilla samannimisiä. WiFi-kirjasto sisältää WiFi-, IPAddress-, Server-, Client- ja UDP-luokat. [4.]

WiFi-luokka sisältää seuraavat funktiot:

- WiFi.begin()-funktio, joka alustaa WiFi-kirjaston verkkoasetukset
- WiFi.disconnect()-funktio, joka kytkee Arduinon verkosta
- WiFi.config()-funktio, jolla voidaan määrittää muun muassa kiinteä IP-osoite
- WiFi.setDNS()-funktio, jolla voidaan määrittää DNS-palvelin
- WiFi.SSID()-funktio, joka palauttaa käytetyn verkon SSID:n
- WiFi.BSSID()-funktio, joka palauttaa reitittimen MAC-osoitteen, johon Arduino on yhdistetty
- WiFi.RSSI()-funktio, joka kertoo langattoman lähiverkon signaalin voimakkuuden
- WiFi.encryptionType()-funktio, joka kertoo langattomassa lähiverkossa käytettävän salaustyyppin
- WiFi.scanNetworks()-funktio, joka kertoo tavoitettavissa olevat langattomat lähiverkot
- WiFi.getSocket()-funktio, joka palauttaa ensimmäisen vapaan soketin
- WiFi.macAddress()-funktio, joka palauttaa käytettävän WiFi Shieldin MAC-osoitteen.

IPAddress-luokka sisältää seuraavat funktiot:

- WiFi.localIP()-funktio, joka palauttaa WiFi Shieldin IP-osoitteen
- WiFi.subnetMask()-funktio, joka palauttaa WiFi Shieldin aliverkkopeitteen (subnet mask)
- WiFi.gatewayIP()-funktio, joka palauttaa WiFi Shieldin yhdyskäytävän IP-osoitteen.

Server-luokka sisältää seuraavat funktiot:

- WiFiServer()-funktio luo palvelimen, joka kuuntelee sisään tulevia yhteyskutsuja tietystä portista
- Server.begin()-funktio, joka alustaa palvelimen kuuntelemaan sisääntulevia kutsuja
- Server.available()-funktio, joka palauttaa palvelimeen kytketyt clientit
- Server.write()-funktio, joka kirjoittaa kaikkiin palvelimeen kytkettyihin clienteihin
- Server.print()-funktio, joka tulostaa kaikkiin palvelimeen kytkettyihin clienteihin

- `Server.println()`-funktio, joka tulostaa kaikkiin palvelimeen kytkettyihin clienteihin ja lisää uuden rivin.

Client-luokka sisältää seuraavat funktiot:

- `WiFiClient()`-funktio luo clientin, jonka voi yhdistää tiettyyn IP-osoitteeseen ja porttiin.
- `Client.connected()`-funktioilla voi tarkistaa, onko client yhdistetty.
- `Client.connect()`-funktioilla client voidaan yhdistää haluttuun IP-osoitteeseen ja porttiin.
- `Client.write()`-funktioilla client voi kirjoittaa siihen palvelimeen, johon se on yhdistetty.
- `Client.print()`-funktioilla client voi tulostaa siihen palvelimeen, johon se on yhdistetty.
- `Client.println()`-funktio tulostaa kaikkiin palvelimeen kytkettyihin clienteihin ja lisää uuden rivin.
- Muita funktioita ovat `available()`-, `read()`-, `flush()`- ja `stop()`-funktioita.

Edellä kuvattujen luokkien lisäksi WiFi-kirjasto sisältää UDP-luokan UDP-viestien käsittelyyn.

Ohjelmointiesimerkissä 2 hyödynnetään WiFi-kirjaston eri luokkien funktioita sekä SPI-kirjaston Serial-luokan funktioita. SPI-kirjaston luokkia ja funktioita käytetään Arduinin ja sen lisälaitteiden keskinäiseen viestinvälitykseen. `WiFiServer`-komennolla alustetaan palvelinyhteys käyttämään porttia 80. `Serial.begin()`-funktioilla alustetaan lisälaitteyhteys käyttämään nopeutta 9600 bittiä sekunnissa. `WiFi.begin()`-funktioilla luodaan yhteys langattomaan lähiverkkoon annetuilla parametreilla. `Server.available()`- ja `Client.connected()`-funktioilla todetaan, onko laite yhdistetty verkkoon ja palvelimeen. [4.]

```
#include <SPI.h>
#include <WiFi.h>

char ssid[] = "TESTWLAN";
char pwd[] = "PASSWORD";

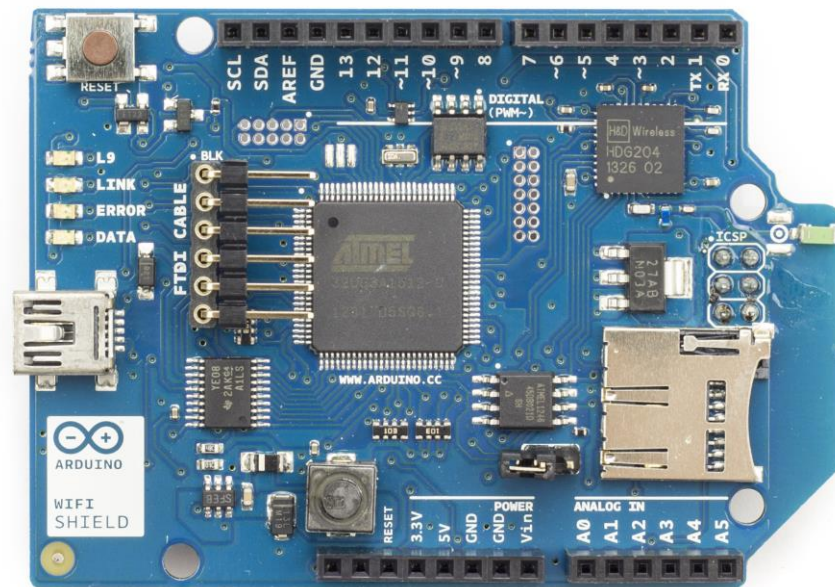
WiFiServer server(80);

Void setup()
{
    Serial.begin(9600);
```

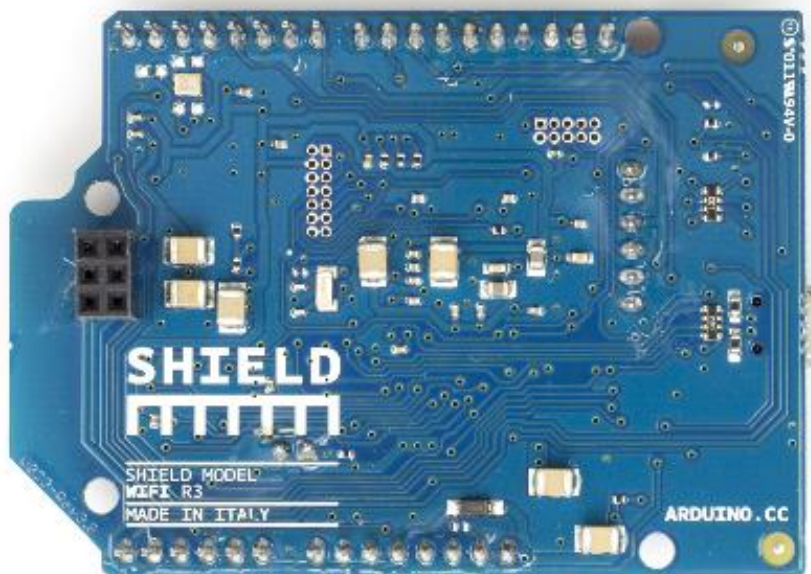
```
Status = WiFi.begin(ssid, pwd);
If (status = WL_CONNECTED)
{
    server.begin();
}
}
void loop()
{
    WiFiClient client = server.available();
    If (client)
    {
        if (client.connected)
        {
            Serial.println("Yhdistetty clienttiin");
        }
    }
}
}
```

Esimerkki 2. Arduinon yhdistäminen Internetiin.

Kuvissa 7 ja 8 esitetään Arduino WiFi Shield -moduuli edestä ja takaa.



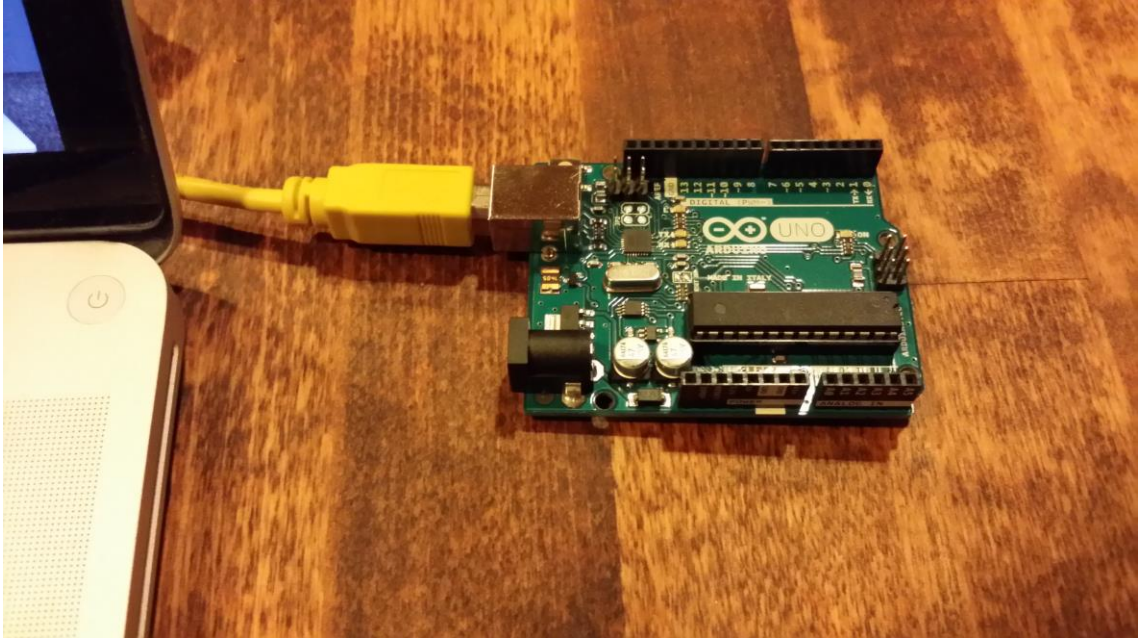
Kuva 7. Arduino WiFi Shield -moduuli edestä [7].



Kuva 8. Arduino WiFi Shield -moduuli takaa [7].

2.3 Arduinon käyttöönotto

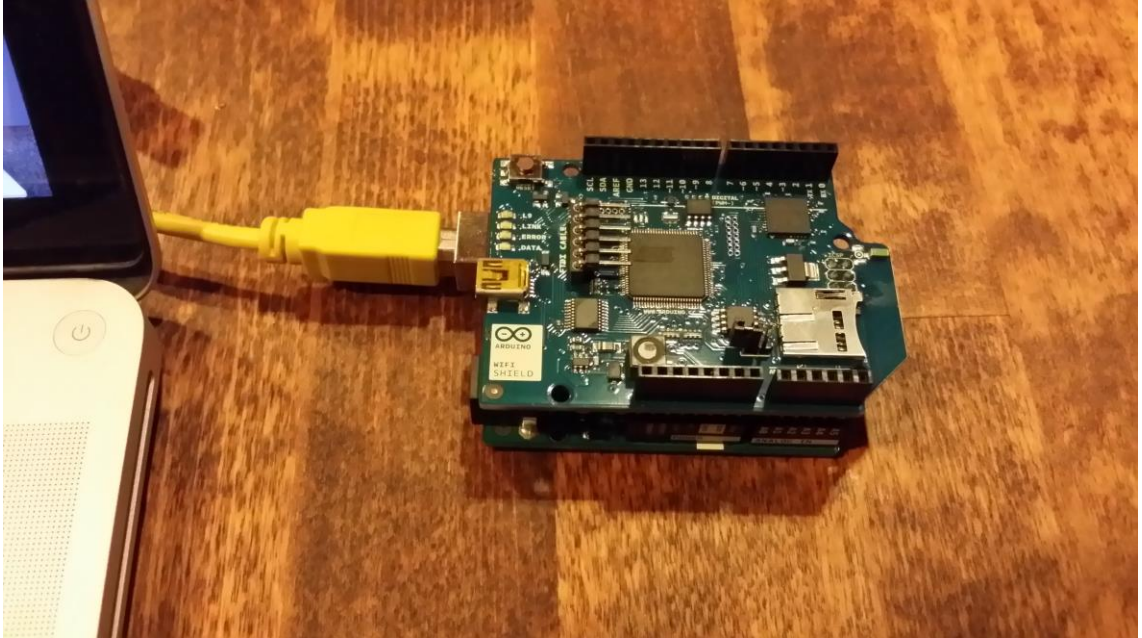
Arduinon käyttöönotto on suhteellisen yksinkertainen toimenpide. Arduinon lisäksi tarvitaan USB-kaapeli, jolla Arduino kytketään tietokoneeseen. Riippuen siitä, mille käyttöjärjestelmälle Arduinon kehitysympäristö asennetaan, saatetaan joutua asentamaan erillinen ajuri tietokoneeseen. Arduino Uno R3 on valmiiksi asetettu ottamaan käyttöjännitteensä tietokoneen USB-portin kautta, mutta esimerkiksi Arduinon Diecimila-versiossa on erikseen määritettävä, että käyttöjännite otetaan nimenomaan USB-portista. Kun valmis ohjelmakoodi on ladattu Arduinoon, USB-kaapeli voidaan irrottaa tietokoneesta ja käyttää virtalähteenä esimerkiksi 9 voltin paristoa. Arduinon vihreä LED-valo syttyy, kun Arduino saa virtaa virtalähteestä. Kun käyttöönotto on tehty, käynnistetään Arduinon IDE-kehitystyökalu. Samalla on syytä tarkistaa, että Arduino toimii moitteettomasti eikä se ole jostain syystä vioittunut. Tämä on helppo tarkistaa lataamalla kehitystyökalun valmis esimerkkiohjelma Arduinoon. Se tehdään siten, että Arduinon kehitystyökalun Työkalut-valikosta valitaan kohta Sarja. Siitä valitaan oikea portti, joka on useimmiten COM3-portti. Valinnan oikeellisuuden voi tarkistaa vielä sammuttamalla Arduino ja valitsemalla uudestaan Työkalut-valikon kohta Sarja, jolloin Arduinon tulisi olla poistunut listasta. Kun Arduino käynnistetään uudelleen, poistuneen portin pitäisi olla jälleen valittavissa. Tämän jälkeen avataan esimerkkiohjelma valitsemalla valikosta Tiedosto -> Esimerkit -> 1.Basics ja siitä ohjelma "Blink". Painamalla "Lähetä" ohjelma latautuu Arduinoon. Jos lataus onnistui, Arduinossa siirtopinnan numero 13 LED-valo alkaa vilkkua. Arduino on näin toimintavalmis (kuva 9). [5; 6.]



Kuva 9. Käyttövalmis Arduino Uno R3.

2.4 Arduino WiFi Shield -moduulin käyttöönotto

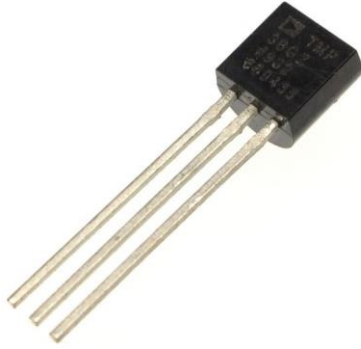
Arduino WiFi Shield -modulin käyttöönotto on helppoa. WiFi Shield asennetaan Arduinin päälle siten, että WiFi Shieldin pinnit kohdistetaan Arduinin vastaaviin pinneihin (kuva 10). Tämän jälkeen siihen voidaan ladata haluttu ohjelma kytkemällä tietokoneeseen liitetty USB-kaapeli Arduinoon. Kun ohjelma on ladattu, USB-kaapeli irrotetaan ja ulkopuolinen virtalähde voidaan kytkeä Arduinoon. WiFi Shield hakee avoimia langattomia verkkoja, mutta myös WEP- ja WPA2-salattuja langattomia verkkoja. Se, mihin langattomaan verkkoon WiFi Shield kytkeytyy, täytyy määritellä ohjelmallisesti. [7.]



Kuva 10. WiFi Shield kytkettynä Arduinoon.

2.5 Lämpötila-anturi

Valitsin lämpötilan mittaamiseen TMP36-anturin (kuva 11). Se on kestävä, luotettava ja halpa. Se on hyvin samantyyppinen anturi kuin LM35/TMP35 tai LM34/TMP34. TMP36-anturin arvoalue on kuitenkin laajempi kuin edellä mainituissa kahdessa muussa anturissa: se on -40 celsiusasteesta aina +150 celsiusasteeseen asti. Tässä mielessä anturi soveltuu hyvin esimerkiksi saunan korkeisiin lämpötilamittauksiin. Lämpötila-anturin perustoiminto on välittää lämpötilan vaihtelut perustuen anturin lämpötilaan, joka muuttaa anturin antamaa ulos menevää jännitettä. Anturin ulommat jalat kytketään virtalähteeseen ja maadoitukseen. Keskimmäisen jalan jännite vaihtelee anturin lämpötilavaihteluiden mukaan. Anturiin johdettavan jännitteen tulee olla 2,7 ja 5,5 voltin välillä, muuten anturi voi vaurioitua. [9.]

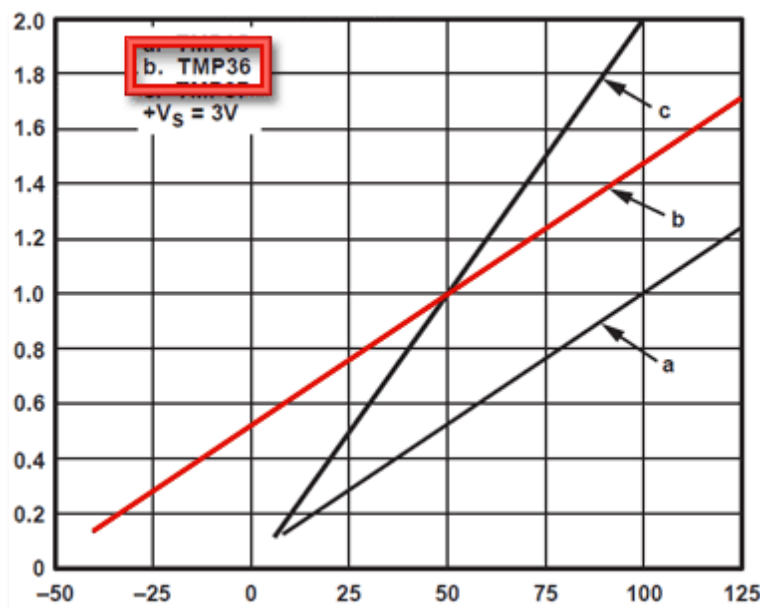


Kuva 11. TMP36-lämpötila-anturi.

TMP36-anturin lämpötilanvaihteluista (kuva 12) johtuva ulos tuleva jännite muutetaan celsiusasteiksi seuraavasti:

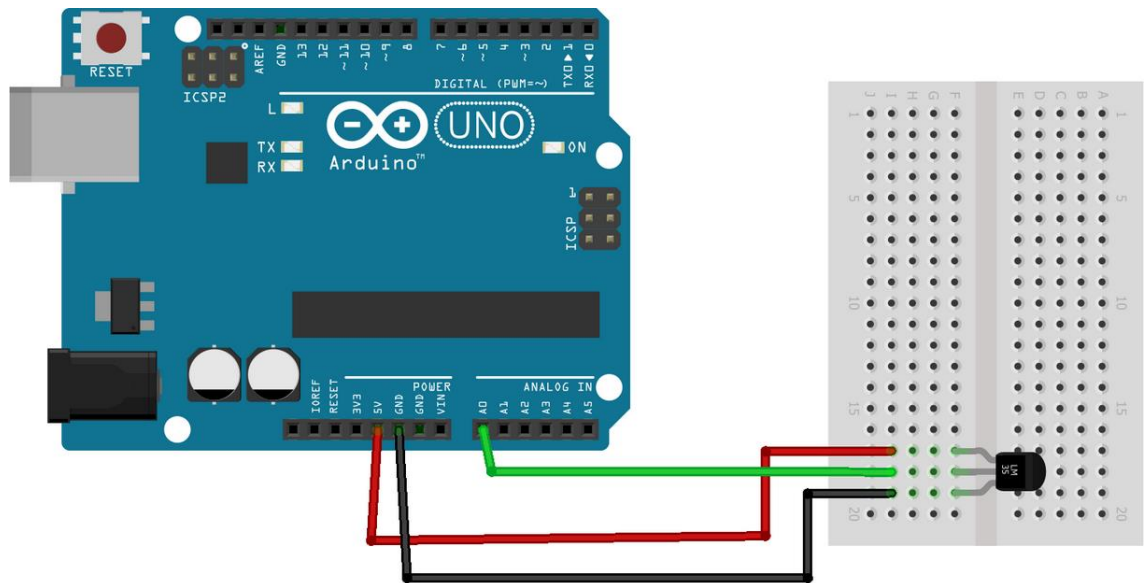
- Mikäli anturiin johdetaan 5 voltin jännite, anturin ulos tuleva jännite muutetaan millivolteiksi seuraavasti: (anturista luettu arvo) * (5000/1024).
- Mikäli anturiin johdetaan 3,3 voltin jännite, anturin ulos tulevan jännite muutetaan millivolteiksi seuraavasti: (anturista luettu arvo) * (3300/1024).

Millivoltit voidaan muuntaa celsiusasteiksi kaavalla (millivoltit-500)/10. [9.]

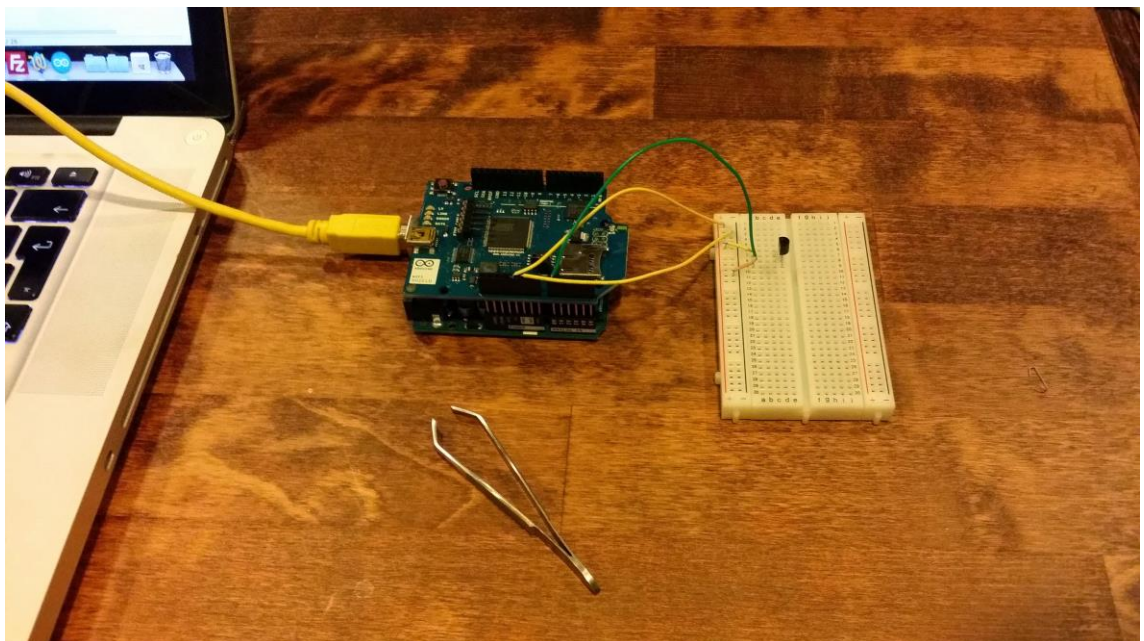


Kuva 12. TMP36-anturin (b) ulos lähtevän jännitteen suhde lämpötilaan.

Lämpötila-anturi voidaan kytkeä Arduinoon esimerkiksi käyttämällä erillistä kytchentäalustaa (kuva 13), jota tässä insinööriyössä käytetään. Lämpötila-anturi kytketään siis kytchentäalustaan ja kytchentäalusta taas kytketään Arduinoon hyppykaapeleilla (kuva 14). [13.]



Kuva 13. Esimerkkikaavio lämpötila-anturin kytkemisestä Arduinoon.



Kuva 14. TMP36-lämpötila-anturi kytkettynä Arduinoon.

2.6 Mobiilioptimoitu lämpötilaa mittaava verkkosovellus

Mobiilioptimoitun lämpötilaa mittaavan verkkosovelluksen toteutuksessa käytettiin useita Internet-ympäristössä yleisimmin käytettyjä tekniikoita ja teknologioita. Itse verkkosovelluksen sivut ohjelmoitiin PHP-ohjelmointikielellä kuitenkin siten, että sivut muodostetaan verkkoselaimelle HTML5-merkintäkielellä ja CSS3-tyylitiedostolla. Verkkosivujen dynaamisuus toteutettiin JavaScript-ohjelmointikielellä ja jQuery-kirjastolla.

PHP-ohjelmointikieli

PHP-ohjelmointikieli on Perlin kaltainen ohjelmointikieli, jota käytetään varsin yleisesti Internet-ohjelmoinnissa. PHP on komentosarjakieli, jossa kirjoitettu ohjelmakoodi tulka-
taan vasta, kun ohjelmaa kutsutaan palvelimelta.

Vuonna 1995 tanskalainen ohjelmistokehittäjä Rasmus Lerdorf kehitti Perl/CGI-skriptin, jolla hän pystyi tallentamaan Internet-sivustonsa kävijätietoja ja näyttämään sivustolla käyneiden määrän. Toteutus herätti laajalti kiinnostusta. Rasmus Lerdorf päätti jakaa kehittämänsä työkalua ja nimesi sen PHP:ksi. PHP tulee sanoista Personal Home Page. Rasmus Lerdorf jatkoi PHP:n kehittämistä ja vaihtoi toteutuksen ohjelmointikieltä Perlillä C-kieleen. Vuonna 1997 Rasmus Lerdorf julkaisi PHP:stä version 2.0. PHP alkoi saada yhä enemmän suosiota. Myöhemmin Rasmus Lerdorf sai tuekseen joukon ohjelmistokehittäjiä.

Vuoden 1998 kesällä julkaistiin versio 3.0. Kehittäjätiimin jäsenet, Zeev Suraski ja Andi Gutmans, kirjoittivat PHP:n ytimen kokonaan uudelleen. Tavoitteena oli tuottaa versio, joka tukisi paremmin kolmansien osapuolten ohjelmointirajapintoja. Versio 4 julkaistiin toukokuussa vuonna 2000. Netcraftin tilastojen mukaan sitä oli tuolloin asennettu yli 3,6 miljoonaan domainpalvelimeen. PHP 4 -version julkaisun jälkeen monet pitivät sitä ohjelmointikielen virallisena julkistuksena, sillä siinä oli monia merkittäviä parannuksia, kuten tuki olio-ohjelmoinnille, natiivi tuki istuntojen käsittelyyn ja natiivi Java-tuki.

Vuonna 2005 julkaistiin versio 5. Kuten edeltäjässään, tässä versiossa oli lukuisia uusia parannuksia kuten try/catch-virrehallinta, SQLiten natiivi tuki ja parannettu XML- ja www-palvelutuki. Tällä hetkellä viimeisin stabiili versio on 5.6.6, joka julkaistiin helmikuussa vuonna 2015. [14, s. 1–8)

PHP-ohjelmointikieltä käytetään usein upotettuna HTML-merkintäkieleen. PHP-ohjelmointikieli kirjoitetaan lähdekoodiin aloitus- ja lopetustagien väliin. Nämä aloitus- ja lopetustagit ovat `<?p` ja `?>`.

Ohjelmointiesimerkissä 3 PHP-ohjelmakoodi on upotettu HTML-merkintäkielellä tehtyyn HTML-sivuun.

```
<!DOCTYPE html>
<html>
  <head>Testi</head>
  <body>
    <?php echo '<H1>Testisivu</H1>'; ?>
    <?php echo '<p>Tämä on tekstiä</p>'; ?>
  </body>
</html>
```

Esimerkki 3. PHP-ohjelmakoodi upotettuna HTML-sivuun.

HTML5-merkintäkieli

HTML5-kieli julkaistiin vuonna 2014. Se on uusin versio yleisesti käytetystä HTML-merkintäkielestä. HTML-merkintäkieli on avoimesti standardoitu merkintäkieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä. HTML-merkintäkieli on rakenteellista tekstiä, joka muodostuu sisäkkäisistä ja peräkkäisistä elementeistä. Elementit eli tágit kuvataan kulmasulkein. Elementit jaetaan rakenteellisiin, esityksellisiin ja hypertekstuaalisiin elementteihin. Rakenteellinen elementti määrittää elementin tarkoituksen, esityksellinen elementti määrittää elementin ulkoasun, ja hypertekstuaalinen elementti linkittää HTML-dokumentin osan toiseen Internetissä julkaistuun dokumenttiin.

Ohjelmointiesimerkissä 4 `<!DOCTYPE>`-elementti kertoo dokumentin tyyppin. Kyseessä on siis HTML-merkintäkielellä ohjelmoitu dokumentti. Seuraavissa elementeissä `<html>` kuvaa HTML-dokumentin juuren, `<head>`-elementissä kuvataan dokumentin tietoa käyttäjille, ja `<body>`-elementti on dokumentin runko. Mainitut elementit on tyypillisesti esitetty kaikissa HTML-merkintäkielellä toteutetuissa dokumenteissa, ja niiden jälkeen voidaan käyttää haluttuja elementtejä. Tässä esimerkissä seuraavana on otsikko esitetty

<H1>-elementillä ja <p>-elementillä kappale, jossa esitettävä teksti on vahvennettu -elementillä. Kappaleen jälkeen esitetään hyperlinkki <a>-elementtiä käyttäen.

```
<!DOCTYPE html>
<html>
  <head>Testi</head>
  <body>
    <H1>Testisivu</H1>
    <p><b>Tämä on tekstiä</b></p>
    <a href="http://www.w3c.org">Linkki W3C-
sivuille</a>
  </body>
</html>
```

Esimerkki 4. HTML-merkintäkielellä ohjelmoitu dokumentti.

HTML5-versiossa tuotiin lukuisia uudistuksia aiempiin HTML-versioihin verrattuna. Näitä uudistuksia ovat muun muassa <video>- ja <audio>-elementit, jotka helpottavat HTML-dokumentteihin upotettujen video- ja audiotiedostojen käsittelyä. Uudistusten myötä HTML-merkintäkielellä pystytään tekemään helpommin itsenäisiä sovelluksia, jotka ovat varsin kilpailukykyisiä eri kannettavien päätelaitteiden natiivisovelluksiin verrattuna. [16.]

CSS-tyyliohjeet

Cascading Style Sheets (CSS) eli tyyliohjeet on erityisesti Internet-dokumenteille kehitetty ohjeisto. CSS:ssä voidaan määrittellä useita eri tyyliohjeita, miten HTML-elementit esitetään Internet-dokumentissa. CSS-tiedostoon, jonka tiedostopääte on .css, viitataan HTML-dokumentissa <link>-elementillä ja siinä määritetyillä parametreilla. Tyyliohjeet voidaan upottaa myös suoraan HTML-dokumenttiin joko <style>-elementtien sisään tai tyyliohjeeseen voidaan viitata suoraan HTML-elementissä style-attribuutilla. CSS:ssä esitetyt säännöt voidaan kiertää niin haluttaessa. CSS:ää käytetään tyypillisesti HTML-dokumenteissa, mutta sitä voidaan käyttää myös muissa rakenteellisissa dokumenteissa, kuten esimerkiksi erityyppisissä XML-dokumenteissa.

Ensimmäinen versio CSS:stä (CSS1) julkaistiin vuonna 1996. Sen kehitti World Wide Consortium eli W3C. CSS1-version tyypillisiä ominaisuuksia olivat muun muassa yleinen

muotoilu, fonttien ominaisuudet ja asemointi. Versio CSS2 julkaistiin vuonna 1998 ja CSS3 vuonna 1999. CSS2-versiossa asemoinnin uudistuminen, elementtien asettaminen päällekkäin ja ylivuodon käsittely olivat uusia ominaisuuksia. CSS3-version uusia ominaisuuksia olivat muun muassa animointi, puheominaisuudet ja X/Y/Z-suuntainen kääntäminen. Alkuaikoina CSS ei ollut kovinkaan tuettu eri Internet-selainohjelmissa, mutta nykyään CSS-tyyliohjeista on tullut standardi Internet-ohjelmoinnissa.

CSS-tyyliohjeiden syntaksi on yksinkertainen. Tyyliohjeissa voidaan viitata suoraan esimerkiksi HTML-merkintäkielen elementtiin tai HTML-elementin attribuuttiin (id tai class). Tyyliohjeessa voidaan viitata myös elementin sijaintiin.

Esimerkissä 5 <p>-elementin "style"-attribuutilla määritetään <p>-elementin tekstin väri mustaksi.

```
<p style="color:black"> This is black. </p>
```

Esimerkki 5. "style"-attribuutin käyttö.

Esimerkissä 6 viitataan erilliseen CSS-tiedostoon, ja siihen viitataan <link>-elementillä.

```
<link href="path/css/style.css" rel="stylesheet">
```

Esimerkki 6. <link>-elementin käyttö.

Esimerkissä 7 tyyliohje on sijoitettu <style>-elementin sisälle. <style>-elementti sijoitetaan <head>-elementin sisään ennen <body>-elementtiä.

```
<style>p { color : black }</style>
```

Esimerkki 7. <style>-elementin käyttö.

Yllä oleva esimerkki on ote CSS-tiedostosta, jossa viitataan <p>-elementtiin ja sen tyyliohjeisiin. [17.]

```
p {  
    color: #000;  
    font-weight: bold;  
}
```

Esimerkki 8. Ote CSS-tiedostosta.

JavaScript-ohjelmointikieli

JavaScript-ohjelmointikieli on vuonna 1995 kehitetty dynaaminen komentosarjakieli. JavaScriptillä voidaan lisätä HTML-merkintäkielellä toteutettujen Internet-sivujen dynaamista toiminallisuutta. JavaScriptiä hyödynnetään myös muun muassa palvelinympäristön Node.js-ympäristössä, työpöytä- ja mobiilisovellusten luomisessa.

JavaScriptin kehitti alun perin Brendan Eich, joka työskenteli tuolloin Netscape Communicationsissa. JavaScript tuli ensimmäisen kerran käyttöön Netscape Navigator 2.0 -selaimessa vuonna 1996. JavaScript eroaa Javasta, ja sen syntaksi perustuu osin C-ohjelmointikieleen. JavaScriptin suuren suosion myötä Microsoft kehitti siitä oman version, jota kutsutaan nimellä JScript. JScript julkaistiin vuonna 1996 Internet Explorer 3.0 -version mukana.

JavaScriptin oliomallin perusyksikköjä ovat object- ja function-oliot. JavaScriptin oliomalli ja periytyvyys pohjautuu prototyyppeihin eikä luokkiin kuten useimmissa olio-ohjelmointikielissä. JavaScriptiä voidaan upottaa HTML-merkintäkielillä toteutettuihin HTML-dokumentteihin, tai se voidaan eriyttää omaksi tiedostokseen, jonka tiedostopääte on js. [18.]

jQuery-kirjasto

jQuery on avoimen lähdekoodin JavaScript-kirjasto, joka sopii muun muassa animaatioiden tekemiseen ja AJAX-sovelluksien tekemiseen. jQuery pyrkii yksinkertaistamaan JavaScriptin käyttöä. Tämä on tehnyt siitä maailman suosituimman JavaScript-kirjaston. jQuery toimii tällä hetkellä kaikilla verkkoselaimilla. jQuery-kirjasto sisältää seuraavat ominaisuudet:

- HTML-dokumenttioliomallin muokkauksen
- tyyliohjeiden muokkauksen

- HTML-tapahtumien metodit
- efektit ja animaatiot
- AJAX-toiminnallisuuden.

jQuery-kirjastoa varten on tehty tuhansia erilaisia lisäosia, joiden toiminnallisuutta ohjelmistokehittäjät voivat hyödyntää omissa ohjelmissaan.

John Resiq julkaisi jQueryn alun perin vuonna 2006 New Yorkissa pidetyssä BarCampissa. John Resiqin työhön vaikutti suuresti Dean Edwardsin aiemmin julkaisema cssQuery-kirjasto. jQuery-kirjasto on JavaScript-tiedosto, joka sisältää muun muassa yleisimmät efektit ja AJAX-toiminnallisuudet. Ohjelmakoodissa voidaan viitata joko jQuery-kirjaston paikalliseen kopioon tai jossakin päin maailmaa sijaitsevalla palvelimella olevaan jQuery-kirjaston kopioon.

Esimerkissä 9 viitataan paikalliseen kopioon.

```
<script src="jquery.js"></script>
```

Esimerkki 9. jQuery-kirjaston paikallinen kopio.

Esimerkissä 10 viitataan Googlen palvelimella sijaitsevaan jQuery-kirjastoon.

```
<script src="//ajax.googleapis.com/  
ajax/libs/jquery/1.8.3/jquery.min.js">  
</script>
```

Esimerkki 10. jQuery-kirjaston sijainti Googlen palvelimella.

Esimerkissä 11 kaikki HTML-merkintäkielellä ohjelmoidun verkkosivun <p>-elementit piilotetaan niitä klikattaessa. [18; 19]

```
$(document).ready(function ()  
{  
    $("p").click(function ()  
    {  
        $(this).hide();  
    }  
});
```

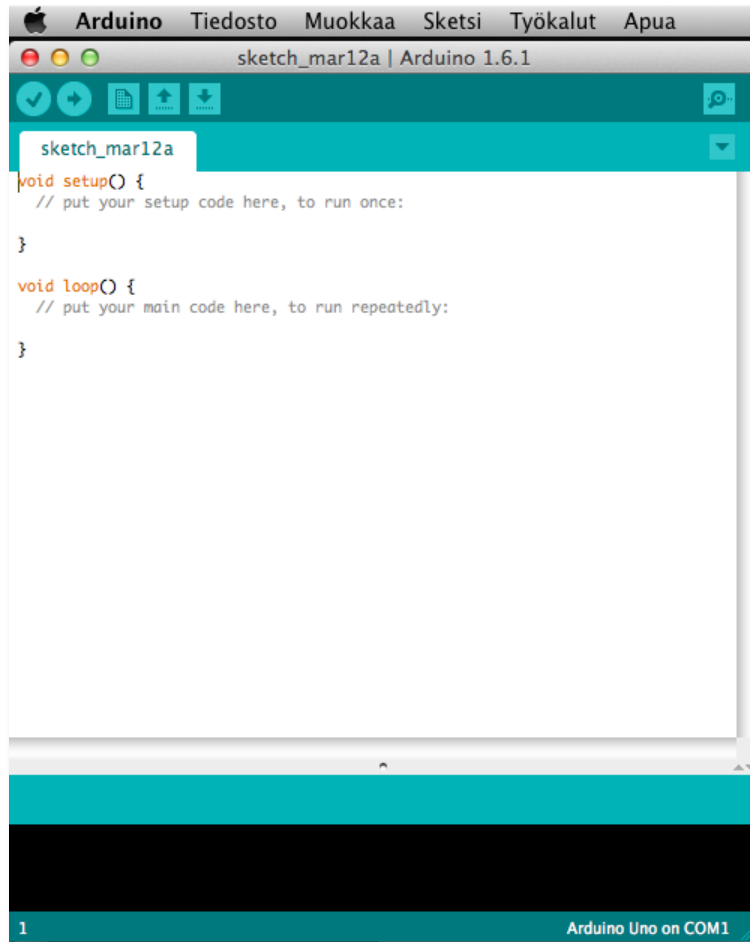
```
});  
});
```

Esimerkki 11. jQuery-kirjaston käyttö.

3 Käytetyt sovelluskehitystyökalut

3.1 Arduino-editori

Arduinon integroitu ohjelmointiympäristö, IDE (kuva 15), on ilmainen ja ladattavissa esimerkiksi Arduinon virallisilta kotisivuilta osoitteesta <http://arduino.cc/en/Main/Software>. Se toimii Windows-, Mac OS X- ja Linux-käyttöjärjestelmissä. Tässä opinnäytetyössä Arduinon IDE asennettiin laitteeseen, jossa käyttöjärjestelmänä on Mac OS X Mavericks. Arduinon IDE on toteutettu Java-ohjelmointikielellä. Se on pyritty tekemään mahdollisimman helppokäyttöiseksi. Arduino-ohjelmat ohjelmoidaan Arduino-ohjelmointikielellä, jossa on kirjastoja sekä C- että C++-ohjelmointikielestä. Arduino IDE:llä ohjelmoituja ohjelmia kutsutaan ”sketseiksi”. Arduinon IDE sisältää Wiring-ohjelmistokirjaston, joka on peräisin alkuperäisestä Wiring-projektista. Wiring-ohjelmistokirjasto tekee useimmat I/O-toiminnot helpommiksi. Kun sketsi on valmis, se voidaan ladata Arduinoon IDE:n ”Lähetä”-napin painalluksella. [6, s. 17–19]



Kuva 15. Arduinon IDE.

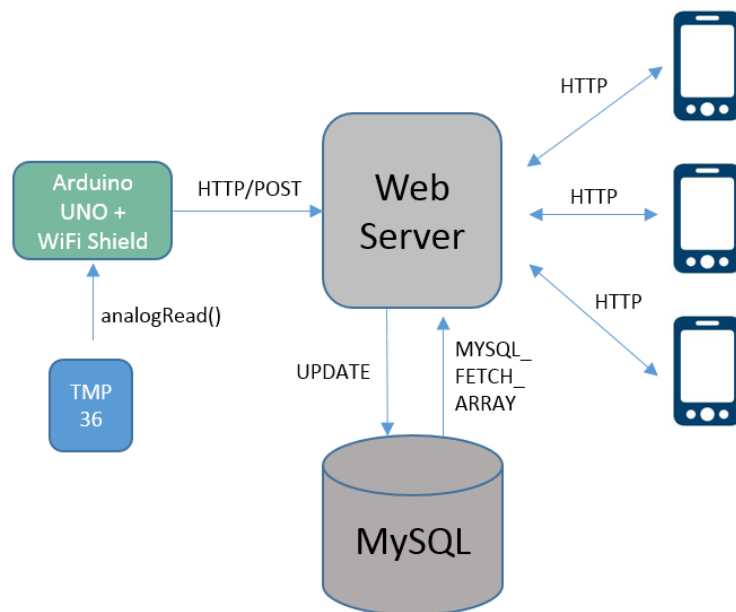
3.2 Notepad++-tekstieditori

Notepad++ on ilmainen tekstieditori, joka soveltuu usean eri ohjelmointikielen editointiin. Alun perin pariisilaisen Don Hon kehittämä tekstieditori perustuu avoimeen lähdekoodiin ja on asennettavissa vain Windows-käyttöjärjestelmään. Notepad++ perustuu tehokkaan C++-ohjelmointikielillä ohjelmoituun Scintilla-komponenttiin. Win32 API:n ja STL:n käyttö mahdollistaa tehokkaan suorituskyvyn ja pienemmän ohjelmakoon. Notepad++-tekstieditorissa on tuki makroille ja lisäosille, joita on julkaistu yli 140. Notepad++ tukee muun muassa seuraavia ohjelmointi-, merkintä- ja skriptikieliä: HTML, C++, PHP, Python ja JavaScript.

4 Sovellus lämpötilan mittaamiseen

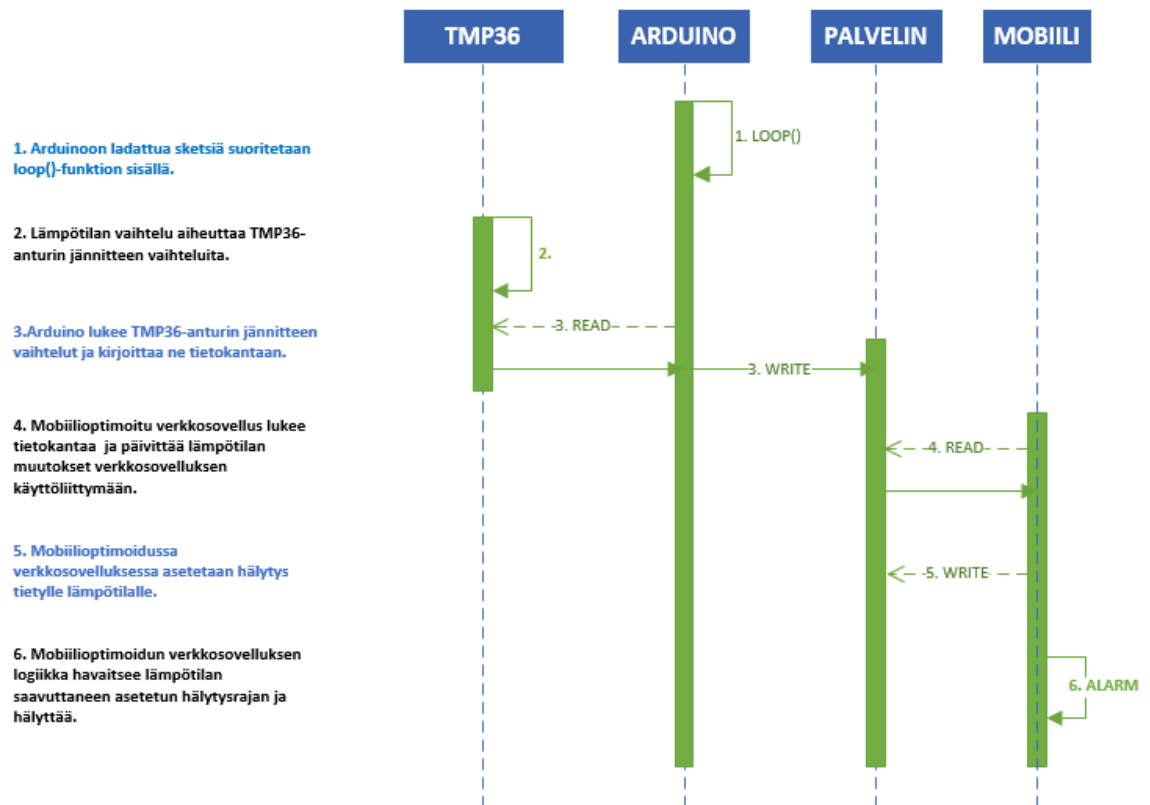
4.1 Sovelluksen arkkitehtuuri

Lämpötilaa mittavan sovelluksen arkkitehtuuri (kuva 16) on seuraavanlainen. Arduinoon on kytketty WiFi Shield -moduuli ja kytkentäalustaan on asennettu TMP36-lämpötila-anturi. Arduinoon ladattu ohjelma eli sketsi on ohjelmoitu Mac OS X Mavericks -käyttöjärjestelmään asennetulla Arduino IDE -editorilla. Valmis sketsi ladataan Arduinoon siihen kytketyllä USB-kaapelilla.



Kuva 16. Lämpötilaa mittaavan sovelluksen arkkitehtuurikaavio.

Kun lataus on tehty, USB-kaapeli irrotetaan Arduinoon. Kun Arduinoon kytketään virtalähde, se alkaa välittömästi tulkita TMP36-lämpötila-anturin välittämää jännitteen vaihtelua. Tämä jännitteen vaihtelu luetaan sketsissä ja muutetaan celsiusasteiksi. Celsiusasteet välitetään langattoman verkon välityksellä palvelimelle ja tallennetaan tietokantaan. Mobiilioptimoitu lämpötilaa mittaava verkkosovellus taas lukee tietokantaa ja esittää ajankohtaisen lämpötilatiedon. Mikäli mobiilioptimoidussa lämpötilaa mittaavassa verkkosovelluksessa on asetettu jokin lämpötilan hälytysraja, mobiilioptimoitu lämpötilaa mittaava verkkosovellus hälyttää, kun lämpötila on saavutettu (kuva 17).



Kuva 17. Lämpötilaa mittaavan sovelluksen sekvenssikaavio.

4.2 Tietokanta

Valitsin lämpötilaa mittaavan sovelluksen tietokannaksi MySQL-tietokannan. Se on ilmainen, avoimen lähdekoodin relaatiotietokantaohjelmisto, jota kehittää ruotsalainen MySQL AB. Yrityksen omistaa nykyään Oracle. MySQL on saatavissa vapaalla GNU GPL -lisenssillä tai kaupallisella lisenssillä, jos käyttäjä ei halua käyttää ohjelmistoa GPL-lisenssillä. [20.]

Päädyn MySQL-relaatiotietokantaan nimenomaan tulevaisuuden kehitystyötä varten. Ajatuksena on tilastoida lämpötilatietoa ja lisätä mobiilioptimoituidun sovelluksen toiminnallisuutta. Tällainen toiminnallisuus voisi olla esimerkiksi tilastoon perustuva arvio siitä, milloin käyttäjän asettama lämpötilan hälytysraja saavutetaan. Muita tallennusvaihtoehtoja olisi ollut esimerkiksi Arduinoon asennettava SD-muistikortti, johon lämpötilatieto

olisi voitu tallentaa tekstitiedostona, tai muut relaatiotietokannat, kuten esimerkiksi SQLite tai PostgreSQL, jotka molemmat ovat ilmaisia relaatiotietokantoja.

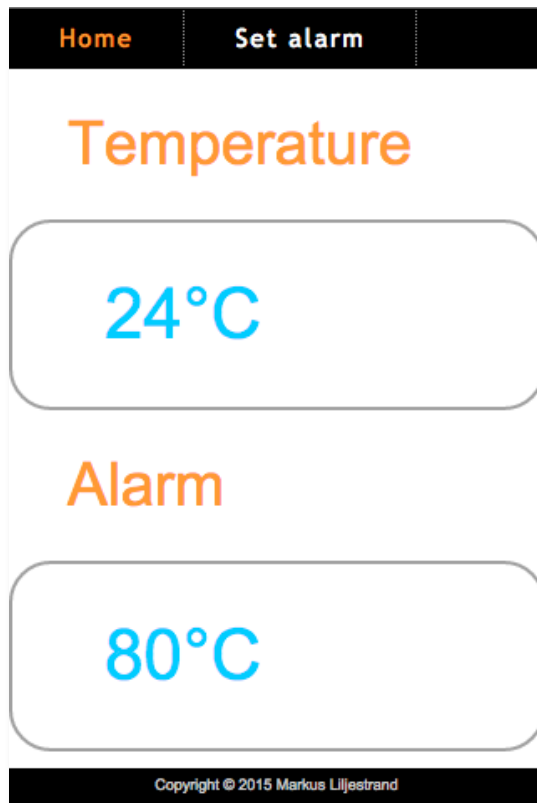
Tietokanta sisältää seuraavat taulut:

Lämpötila-taulu		
Kentän nimi	Tyyppi	Muuta
Temper_ID	INT(11)	AUTO_INCREMENT
Temper_degree	INT(11)	
Temper_timestamp	timestamp	CURRENT_TIMESTAMP

Lämpötilahälytykset-taulu		
Kentän nimi	Tyyppi	Muuta
Alarm_ID	INT(11)	AUTO_INCREMENT
Alarm1	INT(11)	
Alarm2	INT(11)	
Alarm3	INT(11)	
Alarm_timestamp	timestamp	CUR- RENT_TIMESTAMP

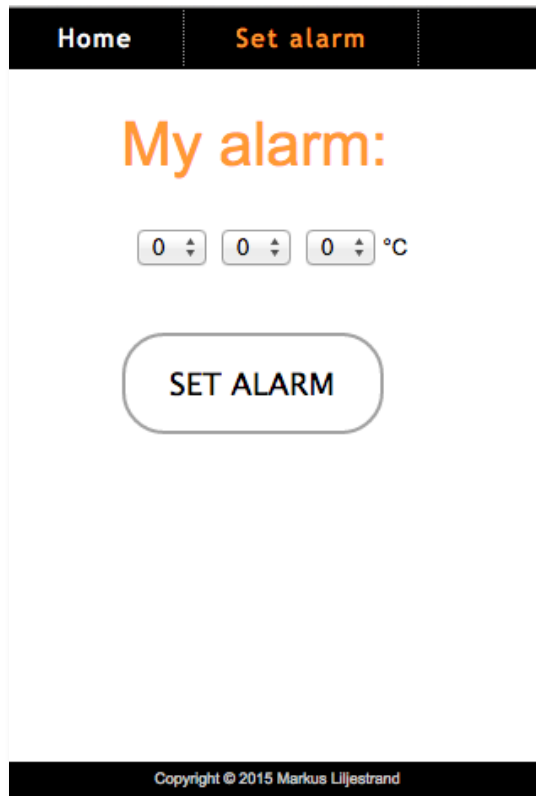
4.3 Mobiilioptimoitu lämpötilaa mittaava verkkosovellus

Mobiilioptimoitu lämpötilaa mittaava verkkosovellus toteutettiin selainpohjaisena ratkaisuna, joka on responsiivinen ja sovitettu eri mobiileille päätelaitteille. Mobiilioptimoidussa lämpötilaa mittaavassa verkkosovelluksessa on kaksi eri näyttöä, joissa ensimmäinen sivu toimii sovelluksen etusivuna. Etusivulla (kuva 18) näytetään senhetkinen lämpötila, joka päivittyy reaaliaikaisesti. Jos käyttäjä on asettanut hälytyksen, näytöllä näkyy asetettu hälytysraja.



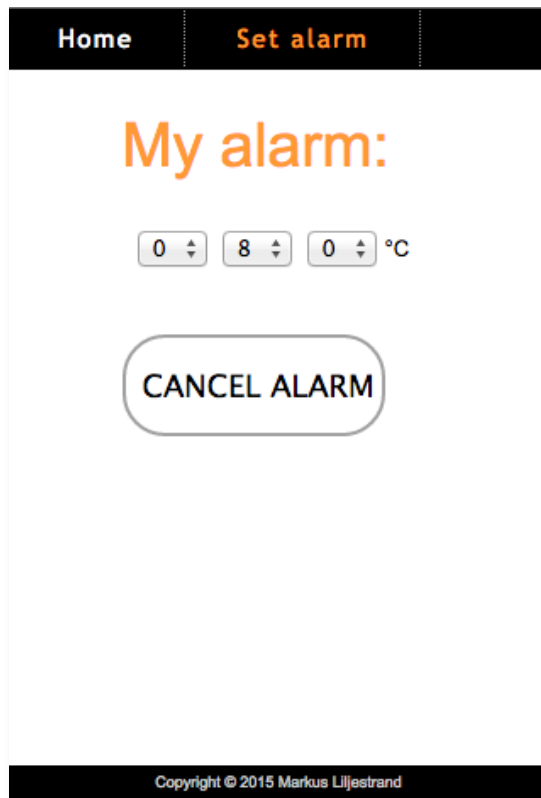
Kuva 18. Lämpötilasovelluksen etusivu.

Kun käyttäjä painaa valikon "Set alarm" -valintaa, tulostetaan sivu, jossa käyttäjä voi asettaa lämpötilahälytyksen (kuva 19).



Kuva 19. Lämpötilahälytyksen asetus.

Kun lämpötilahälytys on asetettu, palataan takaisin etusivulle. Mikäli lämpötilahälytys halutaan perua, painetaan valikon "Set alarm" -valintaa, jolloin tulostetaan "Cancel alarm" -sivu (kuva 20).



Kuva 20. Lämpötilahälytyksen peruutus.

5 Lähdekoodin piirteitä

Mobiilioptimoidun lämpötilaa mittaavan verkkosovelluksen lähdekoodi perustuu PHP-skriptiin ja sen sisälle ohjelmoituun HTML5-merkitäkieleen ja jQuery-kirjastoon. jQuery-kirjaston funktioita käytetään päivittämään mobiilioptimoidun lämpötilaa mittaavan verkkosovelluksen lämpötilatietoa dynaamisesti. Tämä toteutetaan siten, että mobiilioptimoitu lämpötilaa mittaava verkkosovellus toteuttaa AJAX-pohjaisen tietokantakyselyn ja päivittää hakutuloksen dynaamisesti verkkosovelluksen sivun osalle. Tietokantakysely tehdään assosiativisena hakuna eli kysely palauttaa taulukon, jonka solut on nimetty tietokantataulun kenttien mukaisesti (esimerkki 12).

```
while($row = mysql_fetch_array($result, MYSQL_ASSOC))
{
    /* näytetään tulos */
    try {
```

```

echo "<div id='myTemperDiv'><h1 id='myTemper'>" .
$row['temper_degree'] . "°C</h1><input type='hidden'
id='tmp_myTemper' value='" . $row['temper_degree'] .
"'></input></div>";
} catch (Exception $e) {
    echo 'Error :'. $e->getMessage(). '<br />';
    echo 'Code :'. $e->getCode(). '<br />';
    echo 'File :'. $e->getFile(). '<br />';
    echo 'Line :'. $e->getLine(). '<br />';
    exit();
}
}

```

Esimerkki 12. Assosiatiivinen tietokantakysely.

Jokaista tietokantaan kirjoitusta edeltää tietokantayhteyden avaus ja kirjoituksen jälkeinen tietokantayhteyden sulkeminen (esimerkki 13).

```

include_once ('openDB.php');
try {
    $query = "INSERT INTO temper_alarms (alarm1,
alarm2, alarm3) VALUES (". $_POST['alarm1'] .", ".
$_POST['alarm2'] .", ". $_POST['alarm3'] .")";
    echo $query;
    $result = mysql_query($query) or die ("Error in
query: $query " . mysql_error());
} catch (Exception $e){
    echo 'Error :'. $e->getMessage(). '<br />';
    echo 'Code :'. $e->getCode(). '<br />';
    echo 'File :'. $e->getFile(). '<br />';
    echo 'Line :'. $e->getLine(). '<br />';
    exit();
}
include_once ("closeDB.php");

```

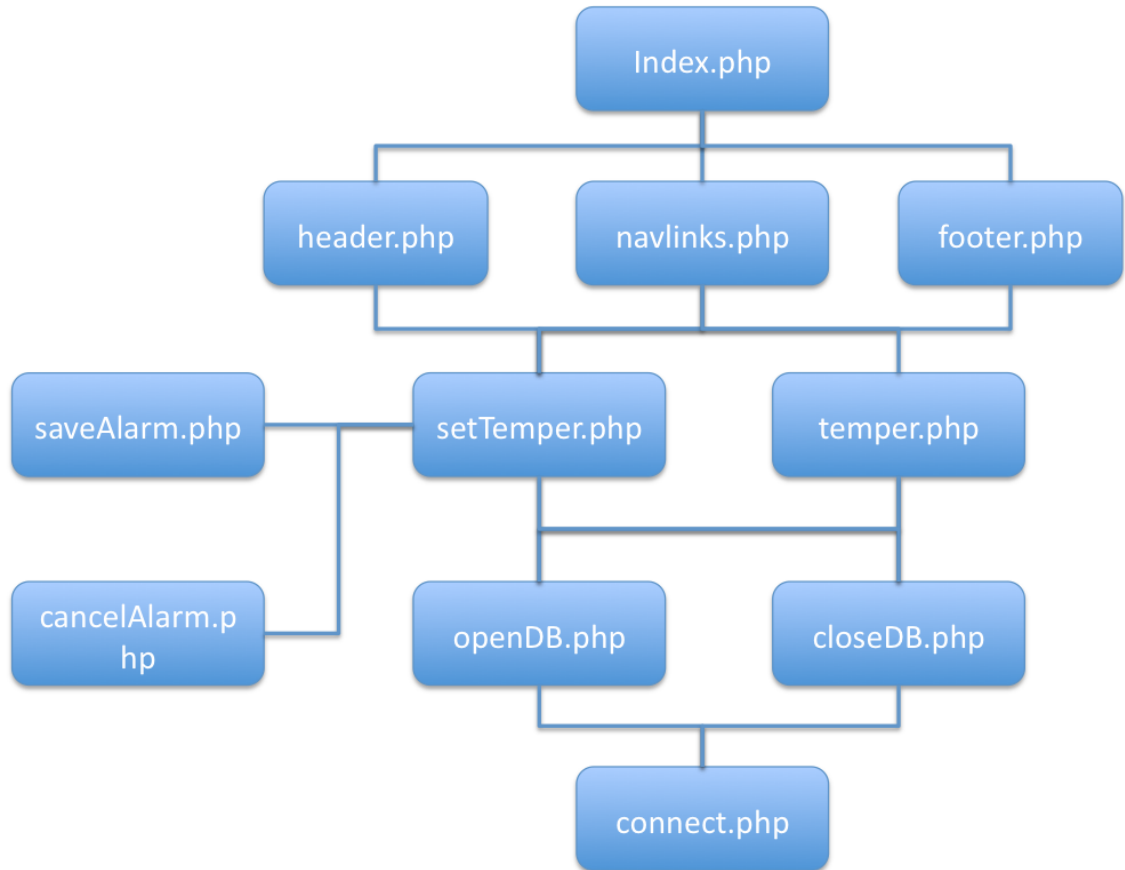
Esimerkki 13. Tietokantaan kirjoitus.

5.1 Sovelluksessa suoritettu kutsut

Mobiilioptimoidussa lämpötilaa mittaavassa verkkosovelluksessa usein käytetyt toiminnot on eriytetty erillisiksi tiedostoiksi. Niitä ovat

- header.php
- navlinks.php
- footer.php
- openDB.php
- closeDB.php.

Header.php-tiedostossa huolehditaan mobiililaitteen tunnistus ja käyttöliittymän optimointi tunnistetun laitteen mukaan. Header.php-tiedostossa huolehditaan myös jQuery-kirjaston viittauksesta. Navlinks.php-tiedostoon on toteutettu verkkosovelluksen navigaatio. Footer.php-tiedostossa esitetään muun muassa verkkosovelluksen tekijänoikeustiedot. openDB.php-tiedostossa toteutetaan tietokantayhteyden avaaminen ja closeDB.php-tiedostossa toteutetaan taas tietokantayhteyden sulkeminen. Lämpötilahälytyksen tallentaminen tehdään saveAlarm.php-tiedostossa ja hälytyksen peruutus cancelAlarm.php-tiedostossa. Ajankohtainen lämpötila ja asetettu hälytys esitetään temper.php-tiedostossa (kuva 21).



Kuva 21. Mobiilioptimoidun lämpötilaa mittaavan verkkosovelluksen rakenne.

5.2 Arduinon sketsi-ohjelma

Arduinossa suoritettava ohjelma eli sketsi on lyhyt ohjelma, joka on jaettu neljään funktioon:

- Setup()
- Loop()
- PrintWifiStatus()
- HttpRequest().

Sketsin alussa liitetään vaadittavat kirjastot, kuten esimerkiksi "WiFi.h". Tämän jälkeen suoritetaan muuttujien kuvaus ja alustus halutuilla arvoilla. Yksi oleellisista muuttujista on lämpötila-anturin ja siihen kytketyn analogisen pinnin A0 esittely. Tämän jälkeen alustetaan WiFiClient-kirjasto. Setup()-funktiossa alustetaan sarjaportti ja varmistetaan WiFi

Shield -moduuliin yhteys. Kun WiFi Shield -moduulin yhteys on varmistettu, yritetään luoda yhteys haluttuun langattomaan verkkoon, jonka parametrit (SSID ja verkon salasana) on kuvattu ja alustettu sketsin alussa. Kun yhteys langattomaan verkkoon on luotu, kutsutaan PrintWiFiStatus()-funktiota, joka tulostaa verkon tietoja Arduino IDE:n sarjamonitoriin. Tämän jälkeen siirrytään suorittamaan loop()-funktiota. Komennolla client.connect(server, 80) tarkistetaan yhteys haluttuun palvelimeen, jonne lämpötilatietoja lähetetään. Mikäli yhteyttä haluttuun palvelimeen ei saada, syy saattaa olla vanhentuneessa WiFi Shield -moduulin firmware-ohjelmassa. Sen tulisi olla vähintään versio 1.1.0 (kuva 22). Mikäli firmware-ohjelma on vanhentunut, se on syytä päivittää (liite 1).

```

ly (Arduino WiFi Shield upgrade is 0.5.4 or later).
=====
Instructions:
To access to the USB devices correctly, the dfu-programmer needs to have the root permissions. de. Then connect the USB cable to the
You can upgrade the firmware of the antenna together with the shield firmware or only the shield firmware
if there aren't changes on the antenna firmware.
Use the '-h' parameter for help
=====
Terminal window and move to the path were you saved the script and type as following to get the help:
Arduino Path not setted. Retry...
Markus-Liljestrand-MacBook-Pro:scripts markusliljestrand$ sudo sh ArduinoWifiShield_upgrade_mac.sh -a /Users/markusliljestrand/Desktop/Arduino.app/Contents/Reso
urces/Java/ -f shield_upgrade_mac.sh -h
Password:
sudo /Users/markusliljestrand/Desktop/Arduino.app/Contents/Reso
urces/Java/ -f shield_upgrade_mac.sh -h
Arduino WiFi Shield upgrade
=====
Disclaimer: to access to the USB devices correctly, the dfu-programmer needs to be used as root. Run this script as root.
****Upgrade WiFi Shield firmware****
=====
Parameter you li pass to the script the current Arduino installation path, for example: /home/user/Coding
Validating...
261594 bytes used (100.01%)
Done. Remove the J3 jumper and press the RESET button on the shield.
Thank you!
Markus-Liljestrand-MacBook-Pro:scripts markusliljestrand$

```

Kuva 22. WiFi Shield -moduulin firmware-ohjelman päivittäminen versioon 1.1.0.

Seuraavassa vaiheessa luetaan lämpötila-anturin jännitettä ja muunnetaan se celsiusasteiksi (esimerkki 14).

```

int anturi = analogRead(tmpPin);
float voltit = (anturi/1024.0) * 5.0;
float temp = (voltit-.5)*100;

```

Esimerkki 14. Lämpötila-anturin jännitteen muuttaminen Celsius-asteiksi.

Tämän jälkeen kutsutaan addTemp.php -ohjelmaa, joka tallentaa tiedon tietokantaan (esimerkki 15).

```

client.println("POST          /temper/includes/addTemp.php
HTTP/1.1");
client.print( "Host: " );
client.println(server);
client.println( "Content-Type: application/x-www-form-
urlencoded" );
client.print("Content-Length: ");
client.println(data.length());
client.println();
client.print(data);
client.println( "User-Agent: Arduino-Box" );
client.println( "Connection: close\r\n" );

```

Esimerkki 15. Lämpötilatieto tallennetaan tietokantaan.

Kun Arduinoon kytketään virtalähde, sketsi-ohjelman suoritus alkaa (kuva 23). Mikäli Arduinosta kytketään virtalähde irti, sketsi-ohjelman suoritus päättyy.

```

Yritetaan yhteyttä langattomaan verkkoon (SSID): HUAWEI-B593-C192
MAC: 78:C4:E:2:9:28
SSID: HUAWEI-B593-C192
IP osoite: 192.168.1.12
signaalin voimakkuus (RSSI):-65 dBm

Luodaan yhteyttä palvelimeen...
Yhteys palvelimeen luotu
Yhteys luotu...
anturin: 147, jännite: 0.72, lampotila: 21.78
Yhteys luotu...
anturin: 151, jännite: 0.74, lampotila: 23.73
Yhteys luotu...
anturin: 151, jännite: 0.74, lampotila: 23.73
Yhteys luotu...
anturin: 153, jännite: 0.75, lampotila: 24.71

```

Kuva 23. Arduinon sketsin suoritus IDE:n sarjamonitorista tarkasteltuna.

6 Yhteenveto

Insinööriyössä perehdyttiin Arduino-mikrokontrolleriin ja siihen, miten sitä voidaan hyödyntää lämpötilatiedon mittaamisessa ja tallentamisessa. Työssä perehdyttiin myös Arduinon lisälaitteeseen, WiFi Shield -moduuliin, jolla Arduino voidaan liittää langattomaan

verkkoon ja sen avulla välittää tietoa Internet-yhteyden avulla palvelimella sijaitsevaan tietokantaan. Työssä toteutettiin sovellus lämpötilan mittaamiseen Arduinon avulla. Sovellus onnistui mainiosti. Sovelluksen toteuttaminen oli erittäin mielenkiintoista ja opettavaista, sillä olen jo pitempään ollut kiinnostunut kotiautomaatioon liittyvistä itse tehdyistä laitteista ja sovelluksista.

Aloittaessani insinööriä minun piti valita sovelluksen toteutukseen joko Raspberry Pi -tietokone tai Arduino-mikrokontrolleri. Varsinkin kotiautomaatioon liittyvät sovellukset ja ratkaisut on tyypillisesti tehty jommallakummalla laitteella. Niiden vertailustakin saisi mielenkiintoisen insinööriä aiheen. Itse päädyin Arduinon sen moduulien ja erilaisten antureiden helpon saatavuuden vuoksi. Pelkästään Suomessa on verkkokauppoja, joista osia voi tilata edulliseen hintaan, mutta useimmin harrastajat tilaavat osat kiinalaisista verkkokaupoista. Työskentelyssä ei ollut mainittavia ongelmakohtia lukuun ottamatta kahta hieman enemmän aikaa vievää ongelmaa. Valitsemani Arduino Uno R3 toimi moitteettomasti, mutta WiFi Shield -moduulin osalta jouduin päivittämään sen firmware-ohjelmiston. Syy tähän oli, ettei Arduinon sketsi-ohjelmaan ohjelmoimani yhteydenotto palvelimeen onnistunut. Firmware-ohjelmiston päivityksen jälkeen yhteydenotto palvelimeen onnistui. Toinen ongelma oli lämpötilatiedon palvelimen GET-metodin parametrin oikean koodauksen ratkaiseminen. Etsiessäni tietoa alan harrastajien eri foorumeista havaitsin, että se oli usein kohdattu ongelma. Päädyin muuttamaan GET-metodin POST-metodiksi. Näin sain ongelman ratkaistua.

Sovelluksen jatkokehityksen osalta olen miettinyt lämpötilatiedon tilastoimista ja sen hyödyntämistä, kuten luvussa 4.2 Tietokannat kuvasin. Ajatuksena olisi siis hyödyntää lämpötilavaihtelun tilastoja, jolloin sovellus voisi laskea arvion käyttäjän asettaman hälytysrajan saavuttamiseksi. Toinen hyödyllinen toiminto voisi olla Arduinon asennetun virtalähteen varauksen näyttäminen sovelluksessa, kun Arduinon virtalähteeksi on kytketty 9 voltin paristo.

Työn edetessä havaitsin, että Arduinoa voidaan ohjelmoida monipuolisesti. Käytännössä mielikuvitus on rajana, millaisia eri laitteita ja sovelluksia voidaan rakentaa. Itseäni ilahdutti Arduinon alati laajentuva mikrokontrolleri- ja moduulivalikoima, jotka mahdollistavat hyvinkin kehittyneitä sovelluksia. Yhteenvetona voin todeta työskentelyn opettaneen ja innostaneen minua. Uskon, että tulevaisuudessa tutustun tarkemmin eri moduuleihin ja mikrokontrolleihin.

Lähteet

- 1 That "Internet of Things" Thing. 2002–2015. Verkkodokumentti. RFID Journal LLC. <<http://www.rfidjournal.com/articles/view?4986>>. Luettu 14.2.2015.
- 2 Arduino Uno. 2015. Verkkodokumentti. Arduino. <<http://www.arduino.cc/en/Main/ArduinoBoardUno>>. Luettu 14.2.2015.
- 3 The Making of Arduino. 2015. Verkkodokumentti. IEEE Spectrum. <<http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>>. Luettu 15.2.2015.
- 4 Products. 2015. Verkkodokumentti. Arduino. <<http://www.arduino.cc/en/Main/Products>>. Luettu 14.2.2015.
- 5 Getting started with Arduino on Mac OS X. 2015. Verkkodokumentti. Arduino. <<http://arduino.cc/en/Guide/MacOSX>>. Luettu 25.2.2015.
- 6 Fitzgerald, Scott & Shiloh, Michael. 2013. The Arduino Projects Book, Second reprint. Arduino.
- 7 Arduino WiFi Shield. 2015. Verkkodokumentti. Arduino. <<http://www.arduino.cc/en/Main/ArduinoWiFiShield>>. Luettu 25.2.2015.
- 8 Monitor your home remotely using the Arduino WiFi Shield. 2014. Verkkodokumentti. Open Home Automation. <<https://www.openhomeautomation.net/monitor-your-home-remotely-using-the-arduino-wifi-shield/>>. Luettu 20.2.2015.
- 9 TMP 36 Temperature Sensor. 2015. Verkkodokumentti. Adafruit. <<https://learn.adafruit.com/tmp36-temperature-sensor>>. Luettu 25.2.2015.
- 10 8-bit Avrel Microcontroller with 8/16/32K Bytes of ISP Flash and USB Controller. 2010. Verkkodokumentti. Atmel Corporation <<http://www.atmel.com/Images/doc7799.pdf>>. Luettu 13.2.2015.
- 11 Makezine. 2004–2015. Verkkodokumentti. Maker Media. <<http://www.makezine.com>>. Luettu 15.2.2015.
- 12 Instructables. 2015. Verkkodokumentti. Autodesk. <<http://www.instructables.com>>. Luettu 20.2.2015.
- 13 Arduino Project: Arduino and LM35 Sensor. 2015. Verkkodokumentti. Xathrya Sa<bertooth. <http://xathrya.web.id/blog/2014/04/02/arduino-project-arduino-and-lm35-sensor/>>. Luettu 26.2.2015.
- 14 Gilmore, W. Jason. 2010. Beginning PHP and MySQL: From Novice to Professional. Fourth Edition. Apress.

- 15 News Archive – 2015. 2001–2015. Verkkodokumentti. The PHP Group. <<http://php.net/archive/2015.php#id2015-02-19-2>>. Luettu 6.3.2015.
- 16 HTML5. 2014. Verkkodokumentti. W3C. <<http://www.w3.org/TR/html5/>>. Luettu 7.3.2015.
- 17 Cascading Style Sheets. 2015. Verkkodokumentti. W3C. <<http://www.w3.org/Style/CSS/>>. Luettu 6.3.2015.
- 18 jQuery Tutorial. 1999-2015. Verkkodokumentti. Refsnes Data. <<http://www.w3schools.com/jquery/default.asp>>. Luettu 8.3.2015.
- 19 What is jQuery. 2015. Verkkodokumentti. The jQuery Foundation. <<http://jquery.com/>>. Luettu 8.3.2015.
- 20 MySQL. 2015. Verkkodokumentti. Oracle Corporation. <<http://www.mysql.com/>>. Luettu 21.2.2015.
- 21 Bridging Technologies. 2015. Verkkodokumentti. Future Technology Devices International Ltd. <<http://www.ftdichip.com>>. Luettu 7.2.2015.
- 23 Karvinen, Kimmo & Karvinen, Tero. 2014. Make: Getting Started with Sensors: Measure the World with Electronics, Arduino and Raspberry Pi. Maker Media. Kindle-lukulaitteella luettava e-kirja.
- 24 Karvinen, Kimmo & Karvinen, Tero. 2011. Make: Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software (Learning by Discovery). Maker Media. Kindle-lukulaitteella luettava e-kirja.

Arduino WiFi Shield -moduulin firmware-ohjelman päivitys

Seuraavassa esitetään WiFi Shield -moduulin firmware-ohjelmiston päivitys tietokoneella, johon on asennettu Mac OS X -käyttöjärjestelmä.

1. Asenna MacPorts-ohjelma ohjeiden mukaan osoitteesta <http://www.macports.com>.
2. Varmista, että sinulla on viimeisin versio Arduino IDE -sovelluskehitystyökalusta asennettuna.
3. Suorita DFU-skripti, joka löytyy alihakemistosta /Applications/Arduino.app/Contents/Resources/Java/Hardware/Arduino/Firmwares/wifi-shield. Tämä tehdään Terminaalissa suorittamalla seuraava komento: `sudo port install dfu-programmer`.
4. Jos Arduino on yhdistetty tietokoneeseen USB-kaapelilla, irrota Arduino tietokoneesta. Mikäli Arduinoon on kytketty erillinen virtalähde, kytke se irti.
5. Irrota WiFi Shield -moduuli Arduinosta.
6. Ennen kuin yhdistät WiFi Shieldin tietokoneeseen, siirrä J3-kytkintä niin, että se yhdistää molemmat pinnit.
7. Kytke WiFi Shield -moduuli tietokoneeseen mini-USB-kaapelilla.
8. Suorita seuraava komento Terminaalissa: `cd sudo sh /Applications/Arduino.app/Contents/Resources/Java/Hardware/Arduino/Firmwares/wifi-shield/scripts`.
9. Jatka suorittamalla seuraava komento: `sudo sh ArduinoWiFiShield_upgrade_mac.sh -a /Applications/Arduino.app/Contents/Resources/Java -f shield`.

10. Mikäli päivitys onnistui, komennon suorittaminen antaa seuraavan viestin:
Done. Remove J3 jumper and press RESET button on the shield. Thank you!

11. Irrota mini-USB-kaapeli tietokoneesta ja WiFi Shield -moduulista. Kytke WiFi Shield -moduuli takaisin Arduinoon.

12. WiFi Shield -moduuli on toimintavalmis.