KARELIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

Ari Hyttinen

ARDUINO-PROJEKTI: LÄMPÖTILOJEN MITTAUSPROJEKTI

Opinnäytetyö
Joulukuu 2016

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Joulukuu 2016** **Tietotekniikan koulutusohjelma** Tikkarinne 9 80200 JOENSUU

Tekijä(t)
Ari Hyttinen

Nimeke
Arduinoprojekti: Lämpötilojen mittausprojekti

Toimeksiantaja
Karelia-amk

Opinnäytetyö sisälsi Arduino mikrokontrollerin avulla tehtävän lämpötilojen mittausprojektin. Lämpötila-antureina olivat käytössä OneWire-lämpötilamittausanturit, joita voitiin liittää Arduinoon haluttu määrä. Arduinoon oli lisätty etäyhteyttä varten lisäosa Ethernet shield, joka mahdollisti tiedon lähetyksen Arduinoon ja Arduinosta eteenpäin palvelimelle.

Ohjausyksikkönä toimi paikalliselle palvelimelle tehty käyttöliittymä, johon tallennettiin sensoritiedot ja josta sensoritiedot vietiin Arduinoon.

Arduinoon tallennettujen sensoritietojen perusteella haettiin OneWire-lämpötilasensoreiden lämpötilatiedot ja ne muutettiin ohjelmallisesti celsiuksiksi. Sensoreiden lämpötilatiedot lähetettiin Arduinosta paikallisella palvelimella olevaan tietokantaan, josta tiedot poimittiin käyttöliittymän näytettäviksi.

Lämpötilasensoreihin oli mahdollista lisätä varoitusrajat, joiden ylittäminen pystyttiin näkemään käyttöliittymässä.

Opinnäytetyössä käydään läpi niin projektin suunnitteluvaiheet suunnittelusta tekemiseen kuin sen toiminta käytännössä.

Kieli suomi	Sivuja 26 Liitteet 22 Liitesivumäärä

Asiasanat
Arduino, OneWire, lämpötila, sensorit

Karelia
UNIVERSITY OF APPLIED SCIENCES

THESIS
December 2016

Tikkarinne 9
80200 JOENSUU
FINLAND

Author (s)
Ari Hyttinen

Title
Arduino project: Temperature measurement

Commissioned by
Karelia University of Applied Sciences

The thesis has details on how to make a temperature measuring project by using an Arduino microcontroller. The temperature sensors were of the model “OneWire”. The Arduino used an Ethernet shield for sending and receiving a data.

A web user interface was made for a local server where the sensors were added. After adding the sensors, the sensor's details could be sent to the Arduino by using the Ethernet connection. The Arduino used sensor's details for receiving the data from OneWire-sensors and changed the data to Celcius-degrees. Then Arduino sent the data to the local server's database and the web user interface showed it.

Warning limits could also be added for the temperature sensors so it was possible to see from the web user interface if any warnings happened. The thesis includes the project's planning process and how it worked for real.

Language
Finnish

Pages 26
Appendices 22
Pages of Appendices

Keywords
arduino, OneWire, temperature, sensors

Sisältö

1 Johdanto

Opinnäytetyössä käydään läpi Arduino-mikrokontrollerilla tehtävä lämpötilojen mittausprojekti suunnittelusta käytännön toteutukseen asti.

1.1 Opinnäytetyön toimeksianto

Opinnäytetyön aihe valikoitui Karelia-ammattikorkeakoulusta saadun toimeksiannon perusteella. Toimeksianto piti sisällään yleisselvityksen Arduinolla tehtävästä lämpötilojenmittausprojektista

1.2 Opinnäytetyön idea ja tavoitteet

Arduinon avulla oli tarkoitus suunnitella ja tehdä Karelian kone/metallipuolelle mittayksikkö, johon saisi lisättyä useita lämpötila-antureita, jotka mittaisivat eri toimipisteiden lämpötiloja ja välittäisivät tietoja Arduinolle. Arduino tallettaisi tiedot joko muistikortille tai johonkin muuhun paikkaan, että data olisi käytettävissä myöhempää käyttöä varten. Datasta olisi mahdollista myös erottaa virherajat, jolloin jos lämpötila menisi yli tai alle tietyn virherajan, niin se olisi helppoa huomata.

Mittayksikköä ei ollut tarkoitus rakentaa paikan päälle, vaan luoda siitä testattava versio. Tarvittavat välineet ja laitteet opinnäytetyötä varten olivat saatavilla koulun puolesta.

2 Arduino kehitysalusta, sensorit ja ohjelmistot

Arduino-kehitysalustana toimi Mega2560-piirilevy. Lisäosana piirilevyyn oli lisätty Ethernet-shield, joka mahdollisti nettiyhteyden käyttämisen osana Arduinoa. OneWire DS18B20-lämpötila-anturit oli kytketty 4,7 kΩ:n ylösvetovastuksen avulla kiinni Arduinoon.

Tietokantatallennusta ja käyttöliittymää varten oli käytössä XAMPP-ohjelmisto. Sen avulla

sai päälle palvelimen, jossa tietokanta ja käyttöliittymä sijaitsivat.

Arduinon ohjelmointi tapahtui C++:aan perustuvalla Arduinon ohjelmointikielellä ja kehitysympäristöllä. Käyttöliittymän PHP/HTML-ohjelmointi toteutettiin Notepad++:lla.

3 Projektiin tutustuminen ja sen toteutus

Ensimmäinen vaihe oli tutustua siihen, mitä osia mm. antureita oli käytettävissä ja miten ne toimisivat ja kuinka ne saisi yhdistettyä Arduinoon.

Anturin mallista ja muista tiedoista ei ollut mitään tarkempaa tietoa projektia aloitaessa, joten ensimmäisenä tulikin haettua Google-hakukoneella tietoja anturissa olevista numerosarjoista ja siitä mikä anturi mahtaisi olla kyseessä. Melko nopeasti selvisi kyseessä olevan lämpötila-anturi OneWire Digital Temperature Sensor – DS18B20.

Opinnäytetyön ohjaajalta oli tullut lyhyt tieto siitä, että anturin kanssa kannattaisi käyttää 4,7 kΩ:n ylösvetovastusta.

Tarkempi tietojen etsiminen osoittikin, että ylösvetovastuksen idea oli estää väärien signaalitietojen lukeminen. Sen avulla voitiin varmistaa signaalin olevan varmasti lähtöisin anturilta tietoa lähettäessä [1]. Nopeasti löytyi myös kytkentäesimerkki [2], jossa oli kuva siitä kuinka ylösvetovastuksen kytkeminen yhdessä Arduinon DS18B20-sensorin kanssa onnistuisi. Lisäksi piti ladata Arduinoa varten käytettävä anturin oma kirjasto ”OneWire” [3], että anturi tunnistuisi oikein. Samoin piti luoda ohjelma, jolla lukeminen ja tietojen muuttaminen tapahtuisi oikeaoppisesti. Malliesimerkki tietojen lukemisesta löytyi ohjelmakirjaston mukana.

Näiden tietojen pohjalta onnistuikin saada lämpötilasensori DS18B20 näyttämään oikeita lämpötilalukemia.

3.1 Sensor-luokan tekeminen

Anturia varten luotu valmis kirjastoluokka ”OneWire” ei vaikuttanut riittävän ja sen muuttaminen alkuperäisestä muodostaan tuntui huonolta idealta, joten parasta olikin luoda oma Sensor-luokka, joka laajentaisi OneWireä tarjoamalla mahdollisuuden lisätä omia muuttujia ja metodeja.

Sensor-luokkaan tarvittavat muuttujat tuli mietittyä aiheesta saadun yleisselvityksen ja yleisen tiedon perusteella. Perustiedot id (sensorin numerollinen tunnus), name (nimi), sensordata (anturin datatieto), lowerlimit/upperlimit (ala- ja ylähälytysrajat), type (tyyppi), model (malli), info (lisätieto) vaikuttivat tekohetkellä hyviltä ideoilta.

Tarkoitus oli myös ajatella mahdollisuutta, että ohjelmistoa voisi käyttää tai soveltaa muissakin yhteyksissä esim. erilaisia antureita pystyisi lisäämään muitakin kuin pelkästään lämpötilaan liittyviä. Anturit käytäisiin läpi tyyppi- ja mallitiedon perusteella mm. tunnistamiseen ja datatiedon lukemiseen liittyen.

3.2 Vector-luokka Arduinoon

Vektori-luokan käyttäminen apuna ohjelman tekemisessä tuli ajankohtaiseksi, kun piti miettiä, kuinka saisi tallennettua lukuisia antureita/sensoreita ohjelmaan. Varsinkin aluksi ideointi keskittyi ainoastaan siihen, kuinka Arduinossa saisi pyöritettyä antureita ja niitä saisi käytyä läpi kätevästi datatiedon saamiseen ja käsittelemiseen. Ohjelmaan lisättävien antureiden määrä ei ollut myöskään millään tavalla tiedossa, vaan ajatus oli tehdä siitä ainoastaan mahdollisimman helppoa.

C++:n STL-vektori oli tullut esiin opintojen yhteydessä ja sen käyttäminen tuntui hyvältä idealta, kun antureiden määrä ei ollut tiedossa ja haluttiin luoda dynaaminen muistinvaraus ohjelmaa varten.

Arduinon kohdalla oli vain huomioitava, että kyseessä on paljolti avoimen lähdekoodin mikrokontrolleri. Sen tarjoamat kokonaisominaisuudet olivat kaukana siitä, minkälaisia resursseja nykytietokoneet ja isot ohjelmistot käyttävät. Tämän takia saatavat lisäkirjastot ja ohjelmistot olivat usein käyttäjien tai yksittäisten yhtiöiden itse luomia ja tekemiä.

Sopiva vektorikirjasto löytyikin netistä hakemalla [4], mutta myöhemmin siinä osoittautui olevan puutteita yleisen toimivuuden kannalta. Vektorin ominaisuuksia ei ollut suunniteltu tarpeeksi hyvin, joten vektoriluokka ei toiminut enää oikealla tavalla, jos tietoa yritti poistaa ja myöhemmin lisätä uudestaan. Ainoa keino saada vektori toimimaan oikealla tavalla oli tyhjentää Arduinon väliaikaismuisti. Tämä tapahtui esimerkiksi ottamalla virtajohto pois Arduinosta.
Tätä projektia varten toimivuus oli silti riittävä, joten ei ollut mitään syytä vaihtaa sitä. Jos vektori-luokan toiminnan haluaa saada toimimaan täysin oikeaoppisesti, niin joutuu lataamaan virallisen luokan tai korjaamaan vektoriluokan tuhoamis- ja tiedonlisäysfunktiot.

3.3 Tallennus tietokantaan, XAMPP-ohjelmisto

Alkuperäisessä suunnitelmassa oli ajatus siitä, että tietoa tallennettaisiin ainoastaan Arduinon muistikortille ja mahdollisesti myöhemmin jollekin palvelimelle. Tämä ei vaikuttanut kovin järkevältä idealta, koska tieto olisi joka tapauksessa jouduttu hakemaan Arduinosta jollakin tavalla.

Tietojen tallettaminen tietokantaan tuntui tässä tapauksessa luonnolliselta valinnalta. Ainoa ongelma olikin siinä, millä tavalla tuo onnistuisi ja tapahtuisi.

Arduinoon piti lisätä ensiksi ”Ethernet shield”-lisäosa, jonka avulla yhteys nettiin onnistui ongelmitta.

Tämän jälkeen täytyi miettiä kuinka onnistuisi paikallisen palvelimen laittaminen päälle ja siihen tietokantaohjelmiston lisääminen. Avuksi tulikin XAMPP-ohjelmisto [5], joka tarjosi mahdollisuuden pystyttää Apache-serveri ja Mysql-tietokanta tietokoneelle.

Serveri ja tietokantaohjelmisto asennettiin kannettavaan tietokoneeseen, johon tietoa lähetettiin Arduinosta.

Arduino toimi omana lähetys/vastaanotinyksikkönä, joka oli yhdistetty RJ-45-kaapelilla reitittimeen.

3.4 Tietokannan ja taulujen luominen tietokantaan

Koska alusta lähtien ei ollut täyttä varmuutta siitä, mitä tietoja tietokantaan kannattaisi tallettaa, niin tietokantataulujen tekeminen tapahtui projektin edetessä tarpeen mukaan.

Tietokannan nimeksi tuli ”arduino” ja tietokantatauluiksi valikoituivat projektin edetessä: sensors, sensors_data, sensors_address.

Tietokantataulujen kuvaus:

- sensors-taulu sisältää sensoreiden tiedot.
- sensors_data-taulu sisältää sensoreiden lähettämät datatiedot eli sensorin lähettämän lämpötila-arvon celsiusasteena (float muodossa).
- sensors_address-taulu sisältää lämpötilasensoreiden osoitetiedot heksalukuina (tekstimuodossa)

3.5 Mysql-connector 1.1.1 kirjaston lisääminen Arduinoon

Tiedon lähettämiseen Arduinosta Mysql-tietokantaan tarvittiin sopiva ohjelmistokirjasto Arduinoon. Etsimällä hakukoneilla löytyi viittaus Mysql-connector-kirjastoon. Seuraava vaihe olikin kokeilla kirjaston toimintaa käytännössä.

Kirjaston käyttöön ottaminen ei tapahtunut hetkessä, vaan vaikka Arduinon IP- ja serverin asetukset tietokantoineen oli laitettu oikein, niin siitä huolimatta ohjelmisto ei vaikuttanut toimivan.

Ongelman takia otinkin yhteyttä Mysql-kirjaston tekijään Charles Belliin ja pyysin lisätietoja siitä, mikä mahtaisi olla ongelman nimi. Sähköpostien kautta selvisi, että Mysql-serverin käyttöoikeudet eivät olleet riittävät Mysql-connectorin käyttämiseen etäkäytössä ja sen takia ohjelmisto ei toiminut oikealla tavalla.

Tämä vaatikin tarkempaa XAMPP-ohjelmiston tutkimista, millä tavalla käyttöoikeuksien lisääminen onnistuisi oikealla tavalla. Ohjelmisto oli toiminut oikein, kun tietoa lisättiin koneelta suoraan, mutta etäkäyttämiseen paikalliselta koneelta eivät oikeudet riittäneet.

Aluksi tuli kokeiltua keinoa, jossa lisätään uusi käyttäjä ohjelmistoon ja annetaan käyttöoikeudet IP-osoitteelle. Syystä tai toisesta tämä ei onnistunut oikealla tavalla, joten lopulta tuli päivitettyä pääkäyttäjän käyttöoikeudet ja muutettua XAMPP:ssa MariaSQL:n asetuksia, jotka auttoivat ongelman ratkaisemisessa. Mysql-connectorin kirjaston kansiosta löytyi dokumentaatio-ohje tätä varten.

Kun tiedon lähettämisessä ei ollut enää ongelmia, niin tuli testattua Mysql-connector-kirjastoa käytännössä. Esimerkit toimivat oikeaoppisesti, mutta vielä oli muutettava ohjelmisto toimimaan halutulla tavalla.

3.6 HTML/PHP-WEB-käyttöliittymä

Tiedon lisääminen ainostaan tietokantaan ei olisi ollut kovin järkevää projektin kannalta, joten parasta oli luoda myös käyttöliittymä kokonaisuuden hallitsemiseen.

Käyttöliittymän suunnittelun ideana oli luoda monipuolinen käyttöliittymä erilaisille sensoreille, että sitä kautta pystyttäisiin lisäämään, muokkaamaan, poistamaan sensoreita ja niiden tietoja.

Ideana oli luoda käyttöliittymä, joka oli helposti muokattavissa, jos sellaista tarvittaisiin. Ajatus ei koskenut siis ainoastaan tämän projektin käyttötarpeita, vaikka ne olivatkin tietenkin pääroolissa.

Käyttöliittymään oli lisätty valmiina komponentteina ainoastaan kaksi kohtaa: CSS-menu [6], jota kautta eri kohdat löytyvät helposti ja kalenterifunktio [7], jonka avulla pystytään näyttämään halutun päivämäärän tarkkuudella tietoja tietokannasta.

3.7 OneWire-lämpötilasensorin osoitetiedon käyttöönotto

Projektissa oli käytettävissä OneWire-lämpötila-antureita, joiden tietoja Arduinon avulla mitattiin ja muutettiin oikeanlaiseen lämpötilamuotoon (celsiuksiksi).

Perehtyminen OneWire-antureiden käyttöön jäi aluksi hyvin suppeaksi, koska projektin kuvaus oli hyvin yleisluonteinen. Tärkeintä olikin saada anturit välittämään tietoa Arduinolle, että tietoa pystyisi myöhemmin käyttämään hyödyksi.

Koska aiemmat kokemukset Arduinosta perustuivat yleensä yksittäisten sensoreiden käyttöön, niin projektia tulikin tehtyä tällä perusteella. Vasta uudemman näyttöpalaverin kautta selvisi, että useamman anturin rinnankytkentä oli täysin mahdollista yhteen Arduinon digitaalisisääntuloon ja anturitietojen hakeminen tapahtuisi niihin liitetyn yksilöllisen osoitetiedon kautta.

Projektia täytyikin muuttaa tämän ajattelun myötä toisenlaiseksi aiemmasta versiosta. Toisaalta pintietojen kautta sensorit tunnistettiin, mutta osoitetieto toimi id-tunnisteena yksittäisille antureille.

Osoitetieto oli ”byte array” taulukkomuodossa, joten sen ottaminen talteen ja käyttäminen olisi onnistunut monella tapaa. Tässä tapauksessa idea osoitetiedon muuttamisesta heksadesimaalijärjestelmään ”string” tiedostomuotoon vaikutti järkevimmältä, koska sillä tavalla osoite näyttäytyi loppukäyttäjälle järkevässä lyhyessä muodossa ja sitä oli helppoa käyttää itse ohjelmakoodissa.

Jokainen taulukkotieto pirstaloitiin yhdeksi heksadesimaaliluvuksi ja erotustietona toimi yhden merkin tyhjä väli (” ”).

Tällä tavalla mahdollistettiin tiedon muuttaminen myöhemmin samanlaiseen ”byte array” muotoon. Samalla osoitetieto näytti käyttäjäystävälliseltä tuossa muodossa (helposti luettavissa).

Ohjelma käytti osoitetietoja sensoreiden anturidatan saamiseen, joten ne täytyi lisätä sensoritietoihin.

3.8 OneWire-luokan muuttaminen

Antureille saatavaa OneWire-luokkaa piti muuttaa lisäämällä sinne erityinen muuttuja pintietoa varten, että sensorin pintieto olisi käytettävissä sensoreiden tunnistamiseen vektorissa. Pintietoa käytettiin apuna, kun osoitetiedot lähetettiin address-tietokantaan.

3.9 UDP-kutsulla sensoritietojen etäpäivitys

Arduinon sensoritietojen päivittäminen käsin kerta toisensa jälkeen paikan päällä olisi ollut työlästä. Tämän takia tuli idea, että se kannattaisi hoitaa etänä.

Tiedossa ei ollut valmista ratkaisua asian hoitamiseen, joten hakukonekäytön tuloksena tuli vastaan tietoa UDP-lähetysten käytöstä [8].

Arduinoon oli saatavilla UDP-esimerkki [9]. PHP:hen ohjelmoitavalla UDP-kutsulla onnistuikin rakentaa UDP-käyttöinen etäpäivitys Arduinoon.

Sensori-tietojen päivitys tapahtui kahdessa osassa. Osoitetietojen hakeminen onnistui ainoastaan, jos Arduinoon oli päivitetty sensori, josta oli pintieto saatavilla. Osoitetiedot toimivat sensoreiden tunnistamisessa id-tietoina.

1. Arduinoon pitikin ensin päivittää sensoritiedot käyttöliittymästä, joista saatiin pin-tiedot Arduinoon, joihin sensorit oli kytketty.
2. Tämän jälkeen täytyi päivittää osoitetiedot eli Arduino haki pintietojen perusteella sensoreiden osoitetiedot ja tallensi ne tietokantaan.
3. Osoitetiedot yhdistettiin sensoritietoihin ja sensoritiedot päivitettiin uudestaan Arduinoon.

Vektori-luokan ongelmien vuoksi täytyi Arduinosta katkaista virrat sensoritietojen uudelleen lähetystä varten. Tällä tavalla tyhjennettiin Arduinon muistista väliaikaistiedot (esim. muuttujien sisältö). Muuten oli ongelmana, että vektori-luokkaan olisi lisätty useita samanlaisia sensoreita, joita käytiin läpi ohjelmallisesti.

3.10 Arduinon kirjastot

Arduinon ohjelmointi oli toteutettu Arduinon omalla ohjelmointikielellä, joka perustuu C++:aan.

Include-toiminnoilla oli tuotu käyttöön eri kirjastoja, joilla onnistui niin OneWire-lämpötila-anturin kuin Ethernet-shieldin käyttö. Mysql-connectoriin olivat omat kirjastolisäykset, kuten myös UDP:n käyttämiseen.

Esim: Lisätyt kirjastot

```
#include <OneWire.h> // Sensor
#include <SPI.h> // Ethernet
#include <Ethernet.h> // Ethernet
#include <MySQL_Connection.h> // MYSQL
#include <MySQL_Cursor.h> // MYSQL
#include <EthernetUdp.h>
```

Define-määrittelyillä oli määritetty char-taulukoiden koot. Char-taulukoita käytettiin pääosin Mysql-kutsuissa yhdessä Mysql-connectorin kanssa.

```
#define NAMELEN 30
char namebuffer[NAMELEN];
```

Alussa oli määritelty muitakin käytön kannalta tärkeitä muuttujia mm. Ethernetiin liittyen. Siihen oli myös lisätty Arduinon ja serverin IP-osoitteet ja Arduinon MAC-osoite.

```
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 0, 13); // Arduino ip
EthernetClient client;
EthernetServer server(80);
IPAddress server_addr(192, 168, 0, 15); // IP of the MySQL *server* here
```

RJ45-kaapelilla kytketyn Arduinon IP-osoitteen sai selville esim. kytkemällä kaapelin ensin pöytäkoneeseen ja tarkistamalla komentoriviltä ipconfig-toiminnolla IP-osoitteen. Samalla tavalla sai tiedon serverin IP-osoitteesta.

Alkumuuttujat ja kirjastojen lisäykset oli määritelty ylhäällä. Setup-kohdassa oli määritelty Arduinon käytön kannalta olennaiset toiminnot eli Ethernetin käyttö, serverin päälle laittaminen ja UDP-portin käyttöönotto.

Arduinon setup-kohdassa oli määritelty seuraavat asetukset:

```
Ethernet.begin(mac, ip); // ETHERNET
server.begin(); // SERVER
Udp.begin(udpPort); // UDP
```

Alempana näkyi yhteyden luominen Mysql-palvelimelle. Jos yhteys ei ollut käytössä, niin sitä yritettiin tarvittaessa myöhemminkin. Ensiksi valittiin serverin IP-osoite, sen jälkeen portti, käyttäjänimi ja salasana.

```
conn.connect(server_addr, 3306, user, password); // MYSQL - connection to mysql server
```

3.11 Arduinon OneWire-Sensor-luokka

Sensor-luokkaa oli laajennettu OneWire-luokasta. Tämä mahdollisti OneWire-luokan toimintojen käyttämisen Sensor-luokassa.

OneWire-kirjastoa oli muutettu alkuperäisestä ainoastaan lisäämällä pintiedon saanti luokkaan.

ONEWIRE.H

```
    uint8_t spin;
```

ONEWIRE.CPP

```
OneWire::OneWire(uint8_t pin)
{
    spin = pin;
}
```

Sensor-luokan tiedot ovat näkyvillä alempana.

Muuttujat:

int id

String address;

String name;

float upperlimit, lowerlimit;

String sensordata;

Rakentaja:

Sensor(int _id, String _address, uint8_t _pin, String _name, float _lowerlimit, float _upperlimit): OneWire(_pin);

Metodit:

void datatoSensors(Sensor& ds);

- Sensoridatan hakeminen ja laittaminen sensorsdata-muuttujaan

void getAlladdress(Sensor& ds);

- Tallensi OneWire-address-tiedot pinnien perusteella Address-luokkaan.

String getSensordata();

- Palautti sensoridatan.

int getId();

- Palautti id-tiedon.

String getName();

- Palautti nimi-tiedon.

float getUpperlimit();

- Palautti ylärajatiedon.

float getLowerlimit();

- Palautti alarajatiedon.

~Sensor();

- Tuhoamisfunktio.

3.12 Arduinon Vector-luokan käyttäminen

Vektoriluokka oli lisätty Arduinon ohjelmaan kopioiden se netistä [3]. Vektoriluokasta ohjelmassa käytettiin ainoastaan ”push_back”-metodia, johon sensori- ja osoitetiedot tallennettiin luokkien alustamisen jälkeen.

Vector<Addresses*> addresses;

- Luotiin addresses-vektori, johon voitiin tallentaa ”Address”-luokkia eli antureiden osoitteita.

Vector<Sensor*> sensors;

- Luotiin sensors-vektori, johon voitiin tallentaa ”Sensor”-luokkia eli antureita.

3.13 Arduinon Address-luokan avulla OneWire-osoitetiedot käyttöliittymään

Address-luokkaa käytettiin ainoastaan OneWire-osoitetietojen tallentamiseen tietokantaan.

Alempana olivat perustiedot Address-luokasta.

Muuttujat:

String address;

int pin = 0;

Rakentaja:

Addresses(String _address, int _pin);

Metodit:

String getAddress();

int getPin();

3.14 Arduinon funktiot

Tässä on listattu Arduinon koodissa käytetyt metodit, joilla pyöritettiin ohjelmaa. Mukana on lyhyt kuvaus siitä, että mikä oli metodin käyttötarkoitus.

Metodit:

void getAlladdresses();

- Etsi sensoreiden pintietojen perusteella kaikki OneWire-address-tiedot (osoitetiedot) ja lähetti ne palvelimen tietokannan address-tauluun.

void updateSensors();

- Haki palvelimen sensors-tietokannasta antureiden tiedot ja talletti ne Arduinoon.

void myDelay(int x);

- Viive-funktio. Tehtiin synkronointia varten, jota ei ohjelmaan tullut. Käytössä mittausviiveessä, kun sensoridataa mitattiin anturilta.

void mysqlSavetodatabase();

- Tallensi sensori-datan sensoreihin ja lähetti tiedot palvelimen sensors_data-tietokantaan.

boolean checkSensors();

- Testasi onko sensoreita ollenkaan lisätty, jos oli niin palautti arvon true.

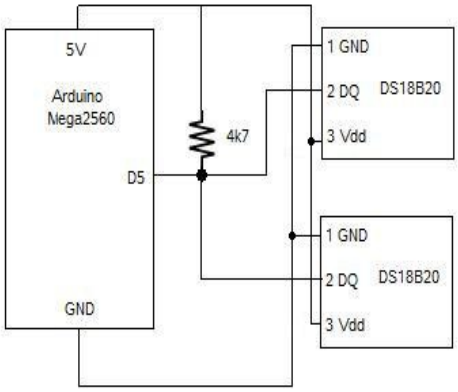
void loop(void);

- Pyöritti pääohjelmaa.
- Testasi onko Mysql-yhteyttä, jos ei ollut niin kokeili uudelleen yhdistämistä.
- UDP-kutsujen hallinnointi ja niiden perusteella getAlladdresses() ja updateSensors() funktioiden kutsuminen.
- Kutsui funktiota checkSensors() ja jos sensoreita oli lisätty niin kutsui mysqlSavetodatabase()-funktiota.

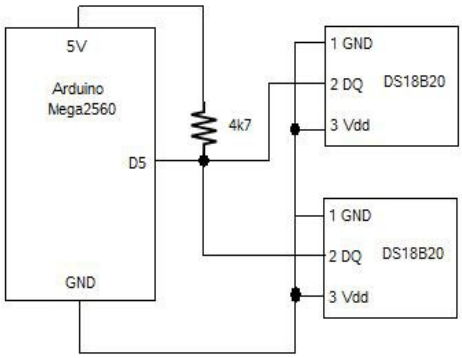
4 Kytkentäkaavio

Kytkentäkaavio (kuvio 1) näytti kuinka OneWire-lämpötila-anturit oli kytketty Arduinoon. D5 kuvaa Arduinon digitaalisisääntuloa 5. Ylösvetovastuksena on käytössä 4,7kΩ:a.

Kytkentäkaavio (kuvio 2) oli myös mahdollista toteuttaa testausta varten siten, että lämpötilasensoreiden + (Vdd) ja – (GND) navat oli yhdistetty maahan. Tämä toimi ainoastaan lyhyillä matkoilla.



Kuvio 1. Kytkentä oli kahdesta OneWire-anturista kytkettynä sarjakytkentänä. Arduinoon.



Kuvio 2. Kytkentä oli testausta varten, kun anturit olivat lähellä Arduinoa.

5 Tietokannan sisältö

Tietokantaa käytettiin projektissa, niin sensori- ja osoitetietojen tallentamiseen kuin myös mittausdatan säilyttämiseen.

Tietokannan nimi oli ”arduino”. Tietokanta-tauluja olivat:

- sensors: Sisälsi sensori-tiedot. Sensorit lisättiin tietokantaan käyttöliittymästä. Niitä voitiin muokata ja poistaa käyttöliittymän avulla.
- sensors_address: Sisälsi Arduinosta haetut osoitetiedot. Tiedot haettiin sensoreiden pintietojen avulla. Osoitetietoa käytettiin oikean datan saamiseen yksittäisiltä sensoreilta.
- sensors_data: Sisälsi sensoreiden datatiedot. Arduino lähetti tiedot sensoreiden pin- ja osoitetietojen perusteella.

Tarkempi kuvaus tietokantatauluista:

sensors:

sid int(11) primary key auto_increment: Sensorin id

address varchar(30): OneWire-address-tieto

pin int(11): Pintieto

name varchar(30): Nimi

lowerlimit float: Alempi varoitusraja

upperlimit float: Ylempi varoitusraja

type varchar(30): Tyyppi

model varchar(30): Malli

info varchar(200): Infoa sensorista

sensors_address:

id int(11) primary key auto_increment: Address id

address varchar(30): Address-tieto hexatekstimuodossa

pin int(11): Pintieto (tämän avulla liitettiin osoitetiedot sensoreihin)

time timestamp: Tallennusaika (näytti milloin tieto oli haettu)

sensors_data:

did int(11) primary key auto_increment: Datan id-tieto

address varchar(30): Sensorin osoitetieto

sid int(11): Sensorin id-tieto

sdata float: Lämpötila-datan tieto celsius-asteina

stime timestamp: Lähetysaika

wlowerlimit float: Varoitus-alaraja

wupperlimit float: Varoitus-yläraja

receipt tinyint(1): Kuittaustoiminto varoituksille (ei käytty mukana ohjelmassa)

6 Käyttöliittymän kuvaus

Käyttöliittymä toteutettiin pääosin PHP/HTML-kielten yhdistelmällä. Mysql-funktioissa käytettiin PHP:n PDO:ta [10]. Osa tyylitiedostoista muokattiin CSS:llä. Käyttöliittymän koodi on liitteenä.

Menussa käytettiin valmista CSS-menua. Hakufunktioissa käytettiin valmista kalenteria.

Formien tietoja siirrettiin HTML:n POST-metodilla, mutta joissakin tilanteissa käytettiin muuttujien tallennuksen/siirron apuna PHP:n SESSION-metodia (esim. jos oli valittu tietty määrä näytettäviä tietoja, jolloin ne eivät nollaudu sivulatausten yhteydessä). Sivusto laitettiin päivittymään automaattisesti 15 s:n välein, mikä mahdollisti livemonitoroinnin ilman selaimen uudelleen päivittämistä.

MYSQL-funktioissa käytettiin select/insert/update/delete-metodeja, joilla tietoja valittiin, lisättiin, päivitettiin ja poistettiin. Osassa tietokantojen nollausta käytettiin truncate-metodia, jolla voitiin poistaa taulun tiedot ja saada id-tiedot alkamaan nollasta. Describe-metodilla haettiin tietokantojen taulujen nimet.

PHP:n foreach-funktioilla käytettiin mm. tietokannan tietojen läpikäymiseen. Joitakin formien koon määritystietoja muokattiin käyttämällä PHP:n array/stack:ta muuttujien tallennukseen.

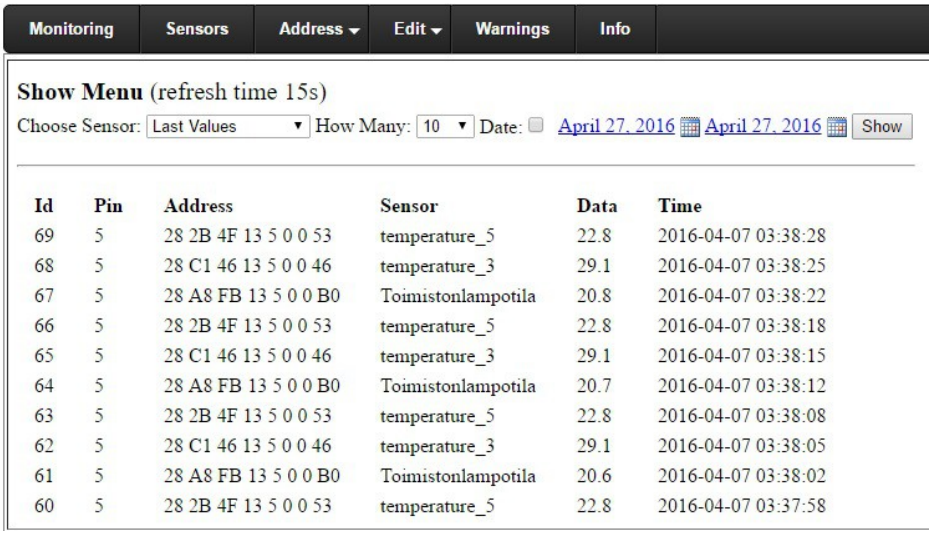
Käyttöliittymän sisältö lueteltu alempana. Jokainen sivu on mainittu erikseen ja niiden sisältämät tiedot.

Menu: Monitoring
index.php

- Toimi ensimmäisenä sivuna, joka aukesi.
- Näytti sensoridatan viimeiset 10 tietoa.
- Hakufunktiot sensoreista ohjautuvat automaattisesti monitoring.php-sivulle.

Menu: Monitoring
monitoring.php

- Näyttöalusta, jonka kautta pystyi hakemaan sensoreiden tietoja joko tietyltä aikaväliltä (jos date-näppäin oli alhaalla) tai sitten viimeiset 10-100 tietoa (kuvio 3).

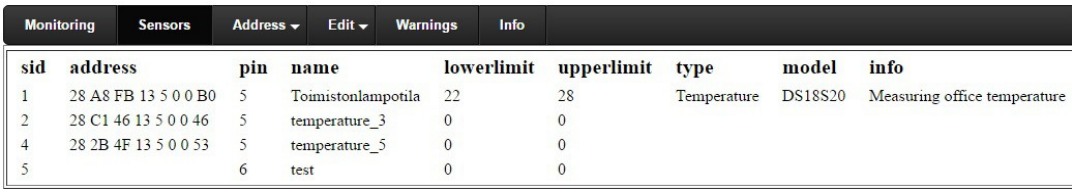


Id	Pin	Address	Sensor	Data	Time
69	5	28 2B 4F 13 5 0 0 53	temperature_5	22.8	2016-04-07 03:38:28
68	5	28 C1 46 13 5 0 0 46	temperature_3	29.1	2016-04-07 03:38:25
67	5	28 A8 FB 13 5 0 0 B0	Toimistonlampotila	20.8	2016-04-07 03:38:22
66	5	28 2B 4F 13 5 0 0 53	temperature_5	22.8	2016-04-07 03:38:18
65	5	28 C1 46 13 5 0 0 46	temperature_3	29.1	2016-04-07 03:38:15
64	5	28 A8 FB 13 5 0 0 B0	Toimistonlampotila	20.7	2016-04-07 03:38:12
63	5	28 2B 4F 13 5 0 0 53	temperature_5	22.8	2016-04-07 03:38:08
62	5	28 C1 46 13 5 0 0 46	temperature_3	29.1	2016-04-07 03:38:05
61	5	28 A8 FB 13 5 0 0 B0	Toimistonlampotila	20.6	2016-04-07 03:38:02
60	5	28 2B 4F 13 5 0 0 53	temperature_5	22.8	2016-04-07 03:37:58

Kuvio 3. Sisältää monitorointi-näkymän

Menu: Sensors
sensors.php

- Haki sensoreiden tiedot tietokannasta ja näytti ne (kuvio 4).



Kuvio 4. Sensoreiden tiedot.

Menu: Address → Show Address

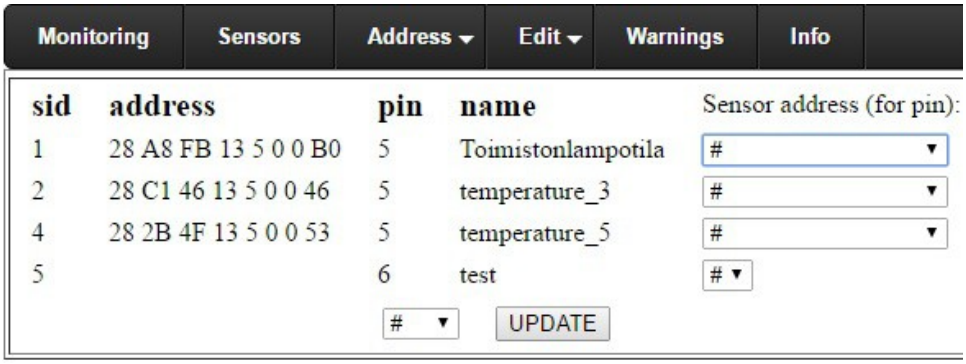
address.php

- Näytti address-tietokannan tiedot, jotka oli haettu Arduinosta.
- Poisti automaattisesti tietokannasta mahdolliset duplikaatti-address tiedot.

Menu: Address → Sensor Address

sensoraddress.php → editedaddress.php

- Mahdollisuus tallentaa address-tiedot sensoreihin (käytti apuna sensorin pintietoa). Update-näppäimen painaminen ohjasi editedaddress.php-sivulle, joka hoiti oikeat Mysql-funktiot (kuvio 5).

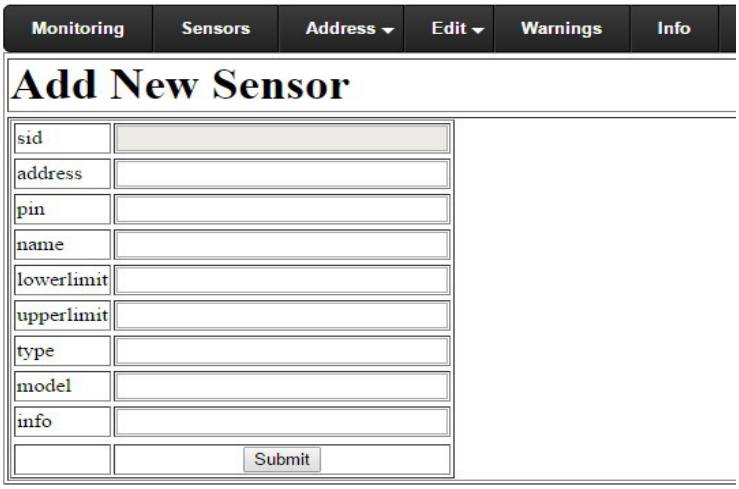


Kuvio 5. Osoitetiedot.

Menu: Edit → Add New Sensor

addnew.php

- Uuden sensoritiedon lisääminen tietokantaan. Javascript-viesti onnistuneesta lisäämisestä. Lähetti formin addnew.php sivulle, joka tallensi uuden sensorin tietokantaan (kuvio 6).

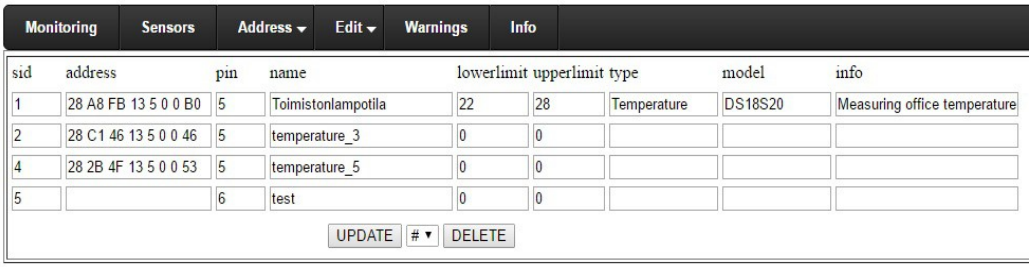


Kuvio 6. Uuden sensorin lisääminen.

Menu: Edit → Edit Sensors

editsensors.php → editedsensors.php

- Pystyi muokkaamaan sensoritietoja esim. tietojen päivittäminen tai poistaminen. Lähetti formin editedsensors.php-sivulle (kuvio 7)



Kuvio 7. Sensoreiden editoiminen.

Menu: Edit → Edit Address

editaddressdata.php → editedaddressdata.php

- Address-tietokannan tietojen poistaminen.

Menu: Edit → Edit Sensor Data

editsensordata.php → editedsensordata.php

- Pystyi poistamaan kaikki tai halutut sensoridata tiedot tietokannasta. Lähetti formin editedsensordata.php-sivulle.

Menu: Edit → Edit Warnings

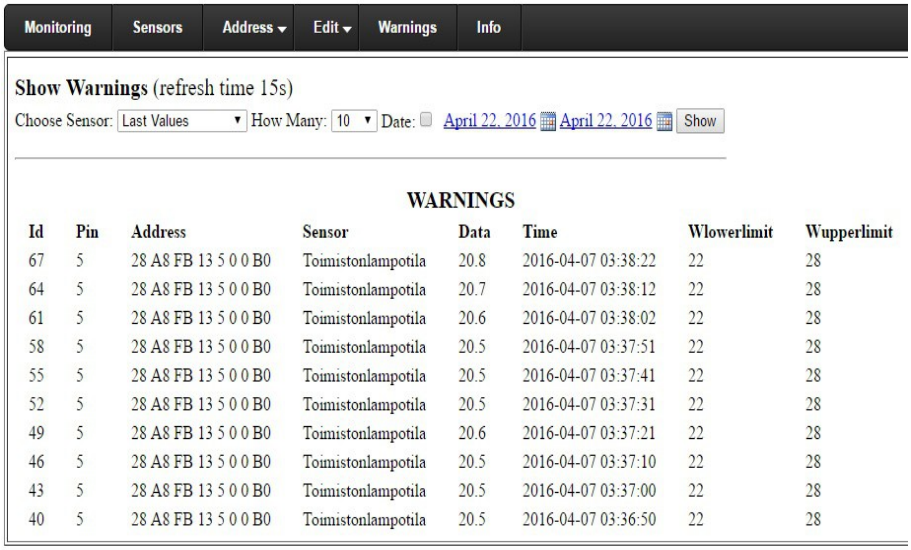
editwarningsdata.php → editedwarningsdata.php

- Pystyi poistamaan joko tietyn sensorin tai kaikki warning-datan sensoritiedostoista. Lähetti formin editedwarningsdata.php-sivulle.

Menu: Warnings

warnings.php

- Arduino tallensi sensoreihin liitetyt varoitusrajat sensors_data-tietokantatauluun, jos rajat ylittyivät. Tiedot näkyivät tältä sivulta (kuvio 8).



Kuvio 8. Sisältää varoitusrajat ylittäneet sensoreiden lämpötilatiedot.

Menu: Info

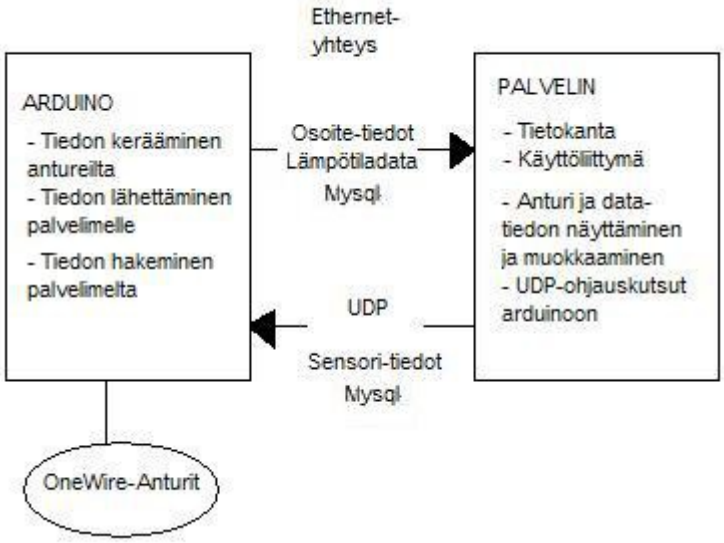
info.php (kuvio 9)

- Sisälsi kaksi toimintoa:
    1. Update Arduino sensors
        - Tallensi Arduinoon sensors-tietokannasta sensorit (tarvittiin myös osoitetietojen hakemiseen pintiedon avulla).
        - Metodina toimi UDP-kutsun lähetys Arduinoon.
    2. Update all address
        - Tallensi address-tietokantaan kaikki mahdolliset Onewire-sensoritietojen osoitetiedot hexamuodossa.
        - Metodina UDP-kutsun lähetys Arduinoon.

Kuvio 9: Info-menun sisältö.

7 Tulokset

Projektissa onnistuttiin tekemään testiversio lämpötilojen mittausprojektista. Sen avulla pystyttiin käymään läpi antureita ja saamaan niiltä oikeanlaista mittausdataa käyttöliittymän näytettäväksi ja muokattavaksi. Toimintaperiaate kuvattu alempana (kuvio 10).



Kuvio 10: Lämpötilojen mittausprojektin toimintaperiaate.

Käyttöliittymän tehtävänä oli toimia ohjausyksikkönä ja näyttöpäätteenä. Käyttöliittymästä voitiin päivittää ja muokata anturitietoja tietokantaan, samoin kuin poistaa anturidatan tietoja. Antureiden tietoihin voitiin määritellä halutut virherajat.

Käyttöliittymästä lähetettiin UDP-kutsuilla yksinkertainen viesti Arduinolle, jolloin Arduino tallensi joko ohjelmaansa anturitietoja Mysql-connector-kirjaston avulla palvelimen ”arduino”-tietokannasta ”sensors”-tietokantataulusta tai tarvittaessa lähetti osoitetiedot ”arduino”-tietokantaan ”address”-tietokantatauluun palvelimelle. Osoitetiedot piti lisätä antureihin, että yksilöllinen datan hakeminen ja tallentaminen tietokantaan jokaista anturia varten onnistuisi oikeaoppisesti.

Mysql-connectorin mysql-insert/select-komennot löytyvät Arduinon koodista (Liite 1) ja siihen oli oma dokumentaatio ohjelmakirjaston mukana.

Antureiden tiedot tallennettiin Arduinon muistiin ”sensors”-vektoriin. Tällä tavalla pystyttiin käymään jokainen yksittäinen anturi läpi tiedon prosessointia ja lähetystä varten palvelimen tietokantaan.

Vector<Sensor*> sensors;

– Antureiden tallennuspaikka niiden läpikäyntiä varten.

Osoitetiedoilla anturit yksilöitiin, mutta ne täytyi hakea ja lisätä jokaiseen anturiin erikseen, että tiedonsiirto toimi oikeaoppisesti.

Vector<Addresses*> addresses;

– Väliaikainen muistipaikka osoitetietoja varten, jotka lähetettiin palvelimen tietokantaan ja lisättiin käyttöliittymässä jokaiseen anturiin.

Datatieto haettiin ja muutettiin jokaiselle anturille erikseen ja se lähettiin samanaikaisesti palvelimen tietokantaan. Tarkemmat tiedot funktioista löytyvät mukana tulevasta Arduinon ohjelmakoodista.

void mysqlSavetodatabase():

– Käy yksitellen läpi kaikki vektoriin tallennetut sensorit.
– Metodi kutsuu tiedon hakufunktiota ”datatoSensors(Sensor& ds)” ja suorittaa datan tallennuksen tietokantaan Mysql-connector-kirjaston avulla.
– Metodi huomioi myös virherajat tallettamalla tietokantaan virherajojen arvot, jos ne ylittyvät.

void datatoSensors(Sensor& ds);

– Hoitaa lämpötilasensoreiden datatietojen hakemisen ja muuttamisen jokaiselle OneWire-lämpötilasensorille osoitetiedon perusteella (osoitetieto muutetaan heksadesimaalijärjestelmästä takaisin ”byte array”-muotoon anturia haettaessa). Tallettaa tiedon Sensor-luokan ”sensorsdata” muistipaikkaan, josta se on käytettävissä tiedon lähetystä varten.

8 Projektin yleisarviointi

Projektin tuloksena onnistuttiin saamaan lämpötilatiedot OneWire-antureilta oikeaan celsius-lämpötilamuotoon ja lämpötilatietojen lähetys paikallisen palvelimen tietokantaan huomioiden virherajat ja tietokannassa olevien tietojen näyttäminen omalla käyttöliittymällä. Käyttöliittymän kautta voitiin myös muokata Arduinon sensoritietoja. Projektille asetetut alkuperäistavoitteet täyttyivät suunnitellusti.

Monet toiminnot on tehty vaiheittain, joten projektin aikana toimintojen tekeminen vaikkapa toisella tavalla olisi voinut olla hankalaa. Ratkaisuja ei ollut valmiiksi tiedossa, vaan ne piti tutkia tapaus kerrallaan.

Ratkaisuista esimerkkeinä voi mainita mm. käyttöliittymän tekemisen ja tietokantatallennuksen. Alkuperäisessä selonteossa suunnitelma oli hyvin vapaamuotoinen projektin toteuttamiseksi, joten tekotapa valikoituikin vasta pikkuhiljaa projektin edetessä.

Arduinon ohjelmasta olisi voinut tehdä yksinkertaisemman, jolloin se olisi toiminut ainoastaan OneWire-lämpötilasensoridatan lähetyskanavana tietokantaan. Toisaalta tässä tapauksessa olisi edelleen pitänyt ratkaista pin-ongelma eli sensoriketjut olisi täytynyt lisätä Arduinoon pintiedon perusteella ja tiedon lähetys olisi tapahtunut yksilöidyn OneWire-osoitetiedon avulla.

Projektia varten suunniteltu toimintaperiaate osoittautui hieman monimutkaiseksi, koska ensin piti tallettaa Arduinoon sensoreihin pintiedot, joiden avulla haettiin osoitetiedot ja ne täytyi lisätä sensoreihin, että saatiin mitattua yksilölliset lämpötilatiedot kaikille OneWire-sensoreille ja lähetettyä ne eteenpäin palvelimelle.

Projektin edetessä tuli vastaan useita ongelmia, joiden arvioiminen etukäteen oli vaikeaa. Ainoa keino olikin testata sopivia ratkaisuita ja tarvittaessa kokeilla uusia.

Yksi ongelma liittyi mm. ohjelmistokieli C++:n ja toisaalta Arduinon sisäisiin kirjastofunktioihin. Tiedon dynaaminen muuttujien tallennus käyttökelpoiseksi projektia varten oli oma ongelmansa.

C++:ssa vektori-luokka on yleensä tärkeä, koska sillä tavalla voidaan tallentaa helposti isoja tietokokonaisuuksia, mikä ei muulla tavalla onnistuisi tai olisi vaikeampaa toteuttaa.

Arduinoon tehty valmis vektori-luokka ei toiminut odotetulla tavalla. Tämä tuli aluksi ilmi mm. sen kautta, ettei tuhoamisfunktio toiminut oikealla tavalla, mutta myös myöhemmin siinä, ettei tiedon tallettaminen onnistunut siihen kaikissa muodoissa.

C++:ssa (joka on pääasiallinen ohjelmointikieli Arduinolle) string tiedostojen sijasta käytetään yleensä dynaamisia char-taulukoita. Vektori-luokan puutteet tulivat esiin tässäkin yhteydessä, kun aiemmin muutetut char*-muodossa olevat dynaamiset taulukot eivät toimineet oikealla tavalla vektori-luokassa, vaan ne täytyi muuttaa string-muotoon.

Samanlaisia ongelmia tuli esiin mm. Mysql-connectorin käytössä. Esimerkkien kanssa ohjelmisto toimi oikealla tavalla, mutta myöhemmin ilmeni ongelmia siinä, jos yritti poistaa muuttujia (delete-toiminnolla). Muuttujien poistamisen takia ohjelmassa saattoi tapahtua virheitä, ettei se toiminutkaan oikealla tavalla. Tästä syystä väliaikaisten muuttujien poistamiset on otettu pois ohjelmistosta.

Ongelmat eivät saata olla tämän projektin kannalta isoja, mutta niiden vaikutusta isommissa kokonaisuuksissa on vaikeaa arvioida. Varsinkin vektori-luokka tarvitsisi korjaamista. Samoin on vaikeaa arvioida, että millä tavalla muistinvaraus toimii nykyisessä ohjelmassa. Tämän vaikutus voi näkyä esim. jos tallennetaan satoja sensoreita Arduinoon, jotka vievät tietoa tietokantaan. Toimintaa on testattu ainoastaan muutamilla sensoritiedoilla.

8.1 Kehitysideat jatkoa varten

Aiemmassa luvussa on mainittu joitakin projektiin liittyviä ongelmia, jotka kaipaisivat korjaamista/tutkimista esim. vektori-luokan toimivuus ja Arduinoon liittyvä muistinvaraus eli tuleeko toiminnan kannalta minkälaisia ongelmia, jos muuttujia ei poisteta ohjelman muistista tietyn väliajoin.

Kehitysideoina jatkoa ajatellen olisi hyvä tutkia mm. sensoritietojen synkronoitua lukemista. Tämä mahdollistaisi lyhyet viiveet sensoritietojen hakemisessa ja nopeamman tiedon-

käsittelyn. Tästä projektista ne on jätetty pois paljolti ajankäytön takia eli ominaisuuksia oli jo nykyisellään projektia varten paljon ja lisäominaisuuksien ottaminen mukaan ei olisi ollut järkevää.

Muita pienempien ominaisuuksien lisäämisiä olisivat mm. Arduinon konfigurointi (esim. ip-osoitteet, serveritiedot, pin-tiedot yms.) muistikortin kautta ja data-tiedon tallennus muistikortille, jos tietokantayhteyttä ei olisi saatavilla. Mustikortilla olevan data-tiedon liittäminen käytettävissä olevaan tietokantaan käyttöliittymän kautta voisi olla hyvä idea.

Yksi tärkeä kohta olisi sensoreiden läpikäyminen yksinkertaisesti. Silloin pitäisi ratkaista ainoastaan kuinka sensoriketjujen pintieto tallennetaan Arduinoon sensoreiden läpikäyntiä varten. Arduino keskittyisi ainoastaan datan lähettämiseen eteenpäin ja tietokanta käyttöliittymän avulla hoitaisi tiedon tallentamisen ja virherajojen huomioimisen. Sensorit tunnistettaisiin edelleen lähetetyn osoitetiedon avulla automaattisesti ja niiden tietoja voisi muokata käyttöliittymässä.

UDP-kutsun käyttäminen ohjelmassa voitaisiin korvata myös paremmalla menetelmällä.

Koodi on tehty ennen kaikkea tätä projektia varten vaihe kerrallaan ja sitä on testattu ainoastaan toiminnan kannalta välttämättömiä toimintoja. Tästä syystä siinä voi olla sellaisia kohtia, joita pystyisi lyhentämään tai toteuttamaan toisella tavalla yksinkertaisemmin ja järkevämmin.

Varsinkin koodia pitäisi testata useita kertoja vaihe kerrallaan, mikä mahdollistaisi parempien ratkaisujen löytämisen ja koodin toimimisen paremmin käytännössä.

Projektia voi joka tapauksessa käyttää apuna, jos tehdään lopullista versiota lämpötilojen mittausyksiköstä alkuperäiseen käyttökohteeseensa. Siitä voi ottaa mahdollisia vinkkejä toteuttamiseen, ohjelmistolliseen puoleen ja toimintojen testaamiseen.

8.2 Kaupalliseen ja kotikäyttöön liittyvät mahdollisuudet sekä opinnäytetyön hyödyntäminen jatkossa

Tämän opinnäytetyön avulla pystytään ratkaisemaan mahdollisia teknisiä ja ohjelmallisia ongelmia, jos mietitään Arduinoon liittyviä kaupallisia tai kotikäyttöön soveltuvia ratkaisuja.

Arduinoon on helppoa liittää erilaisia sensoreita, joita voidaan käyttää apuna vaikkapa erilaisten taloon liittyvien mittaustietojen saamiseen.

Tietokantojen ja palvelimen yhteiskäyttö mahdollistaa erilaisten ohjaus- ja mittauspaneelien tekemisen, jonka avulla kaikki tiedot ovat löydettävissä yhdestä paikasta. Hyötyä siitä voisi löytyä mm. talonrakennus/käyttöpuolella, jolloin erilaiset sähkönkulutus, huoneistojen lämpötila yms. tiedot löytyisivät helposti.

On vaikeaa silti nähdä tarkkaan, että minkälaisia mahdollisia kaupallisuuteen liittyviä sovelluksia tämän projektin avulla pystyttäisiin rakentamaan. Suurin ongelma koska on siinä, että kaikki nuo projektit täytyisi miettiä hyvin yksilöllisesti ja tapauskohtaisesti. Usein kysymys olisi myös uusista tuotteista, joiden markkinointiin täytyisi panostaa rahaa. Toisaalta tuotteen pitäisi olla myös loppukäyttäjän kannalta niin valmis, että se olisi helppoa markkinoida ja myydä eteenpäin.

Kotikäytössä projektista saattaa olla apua, jos haluaa rakentaa jotakin vastaavanlaista lämpötilan mittaus- ja näyttöjärjestelmää tai muita Arduinoon liittyviä sensorijärjestelmiä.

Lähteet

1. Wikipedia. Pull up resistor. 2016. https://en.wikipedia.org/wiki/Pull-up_resistor 12.10.2016
2. Hobbytronics.co.uk. OneWire-DS18B20-sensorin kytkentäesimerkki. 2016. http://www.hobbytronics.co.uk/ds18b20-arduino 12.10.2016
3. Playground.arduino.cc. OneWire-DS18B20-sensorin ohjelmakirjasto. 2016. http://playground.arduino.cc/Learning/OneWire 12.10.2016
4. Github.com. Vektori-kirjasto. 2016. https://github.com/canalplus/tvduino/blob/master/vector.h 12.10.2016
5. Wikipedia. XAMPP. 2016. https://fi.wikipedia.org/wiki/XAMPP 12.10.2016
6. Cssmenumaker.com. CSS-pudotusvalikko. 2016. http://cssmenumaker.com/menu/css3-drop-down-menu 12.10.2016
7. Triconsole.com. PHP-Kalenteri. 2016. http://www.triconsole.com/php/calendar_datepicker.php 12.10.2016
8. Arduino.cc. Arduinoon UDP-esimerkki. 2016. http://www.arduino.cc/en/Tutorial/UDPSendReceiveString 12.10.2016
9. Wikipedia. UDP. 2016 https://fi.wikipedia.org/wiki/UDP 12.10.2016
10. PHP.net. PDO-manual. 2016. PDO-käyttöopas http://php.net/manual/en/book.pdo.php 12.10.2016

Liite 1. Arduinon koodi

```
#define NAMELEN 30 // name char array length
#define ADDRESSLEN 30 // address char array length
#define TLEN 10 // temperature and warnings char array lentgh
#define QUERYLEN 200 // mysql query length for selet and insert (need to be enough big for data)
#define AQUERYLEN 128 // mysql query length for address (need to be enough big for data)

#include <OneWire.h> // Sensor
///====SERVER-CLIENT-DETAILS====///
#include <SPI.h> // Ethernet
#include <Ethernet.h> // Ethernet
#include <MySQL_Connection.h> // MYSQL
#include <MySQL_Cursor.h> // MYSQL
#include <EthernetUdp.h> // UDP
unsigned int udpPort = 8888; // UDP PORT
// buffers for receiving and sending data
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; // UDP - buffer to hold incoming packet
// An EthernetUDP instance to let us send and receive packets over UDP
EthernetUDP Udp;

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 0, 13); // Arduino ip
// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetClient client;
EthernetServer server(80);
```

```
///====SERVER-CLIENT-DETAILS====///
///====SERVER-DETAILS====///
IPAddress server_addr(192, 168, 0, 15); // IP of the MySQL *server* here
char user[] = "root";              // MySQL user login username
char password[] = "test";        // MySQL user login password
//EthernetClient client2;
MySQL_Connection conn((Client *)&client);
boolean mysqlconnection = false; // test for Mysql-connection (setup)
// Mysql-query
char INSERT_DATA[] = "INSERT INTO arduino.sensors_data (sid, address, sdata) VALUES
(%d,'%s',%s)";
char INSERT_DATA_WARNING[] = "INSERT INTO arduino.sensors_data (sid, address, sdata,
wlowerlimit, wupperlimit) VALUES (%d,'%s',%s,%s,%s)";
char INSERT_ADDRESS[] = "INSERT INTO arduino.sensors_address (address,pin) VALUES
('%s',%d)";
const char SELECT_SENSORS[] = "SELECT sid, address, pin, name, lowerlimit, upperlimit
FROM arduino.sensors";
char query[QUERYLEN]; // for mysql-insert data, check that its enough big for query
char aquery[AQUERYLEN]; // for mysql-insert address
char squery[QUERYLEN]; // for mysql-select
char temperature[TLEN]; // myslq-insert data
char wlowerlimit[TLEN]; // mysql-insert warnings
char wupperlimit[TLEN];
char namebuffer[NAMELEN];
char addressbuffer[ADDRESSLEN];
char alladdressbuffer[ADDRESSLEN];
///====SERVER-DETAILS====///

#define SENSORDELAY 1000 // 1s sensor delay

volatile int state = LOW; // for Attachinterrupt, ei vielä lisätty

template<typename Data>
```

```
class Vector {
  size_t d_size; // Stores no. of actually stored objects
  size_t d_capacity; // Stores allocated capacity
  Data *d_data; // Stores data
 public:
  Vector() : d_size(0), d_capacity(0), d_data(0) {}; // Default constructor
  Vector(Vector const &other) : d_size(other.d_size), d_capacity(other.d_capacity), d_data(0) {
    d_data = (Data *)malloc(d_capacity * sizeof(Data));
    memcpy(d_data, other.d_data, d_size * sizeof(Data));
  }; // Copy constuctor
  ~Vector() {
    free(d_data);
  }; // Destructor
  Vector &operator=(Vector const &other) {
    free(d_data);
    d_size = other.d_size;
    d_capacity = other.d_capacity;
    d_data = (Data *)malloc(d_capacity * sizeof(Data));
    memcpy(d_data, other.d_data, d_size * sizeof(Data));
    return *this;
  }; // Needed for memory management
  void push_back(Data const &x) {
    if (d_capacity == d_size) resize();
    d_data[d_size++] = x;
  }; // Adds new value. If needed, allocates more space
  void clearv() //here
  {
    memset(d_data, 0, d_size);
    d_capacity = 0;
    d_size = 0;
    free(d_data);
  }
  size_t size() const {
```

```
    return d_size;
  }; // Size getter
  Data const &operator[](size_t idx) const {
    return d_data[idx];
  }; // Const getter
  Data &operator[](size_t idx) {
    return d_data[idx];
  }; // Changeable getter

 private:
  void resize() {
    d_capacity = d_capacity ? d_capacity * 2 : 1;
    Data *newdata = (Data *)malloc(d_capacity * sizeof(Data));
    memcpy(newdata, d_data, d_size * sizeof(Data));
    free(d_data);
    d_data = newdata;
  };// Allocates double the old space
};

class Addresses {
 public:
  String address = "";
  int pin = 0;
  Addresses(String _address, int _pin) {
    address = _address;
    pin = _pin;
  }
  String getAddress() {
    return address;
  }
  int getPin(){
    return pin;
  }
```

```
};

Vector<Addresses*> addresses; // alustaa addresses vektorin, jota käytetään sensor-class

// DS18S20 Temperature chip i/o
class Sensor : public OneWire
{
  public:
    int id;
    String address;
    String name;
    float upperlimit, lowerlimit; // limits
    /*String type = ""; // Sensor's type ex: Temperature  // DATABASE VALUES - NOT FOR
ARDUINO
    String model = ""; // Sensor's model ex: DS18S20
    String info = ""; // extra information about Sensor*/
    String sensordata = ""; // data
    Sensor(int _id, String _address, uint8_t _pin, String _name, float _lowerlimit, float _upperlimit):
OneWire(_pin) {
      id = _id;
      address = _address;
      name = _name;
      lowerlimit = _lowerlimit;
      upperlimit = _upperlimit;
    }

    void datatoSensors(Sensor& ds) { // ok. measure a data to the sensordata variable
      sensordata = "";
      int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;

      byte i;
      byte present;
      byte data[12];
```

```
byte addr[8];

char copyaddress[ADDRESSLEN]; // move *address to addressbuffer

char buffer1[8][5]; // defaults to all 0's
char *ptr;
int count = 0;

// the address is in char* so this method change it to byte-array
address.toCharArray(copyaddress,30);
ptr = strtok(copyaddress, " ");
while (*ptr) {                // Parse into substrings
  strcpy(buffer1[count], ptr);          // Save the substring
  addr[count] = (byte) strtoul(buffer1[count], 0, 16);   // Convert to unsigned byte
  count++;
  ptr = strtok(NULL, " ");
}

ds.reset();
ds.select(addr);
ds.write(0x44, 1);        // start conversion, with parasite power on at the end

myDelay(SENSORDELAY);     // maybe 750ms is enough, maybe not - calling own function so attachinterrupt can be added to code
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset(); // need for right temperature
ds.select(addr);
ds.write(0xBE);         // Read Scratchpad

for (i = 0; i < 9; i++) {           // we need 9 bytes
  data[i] = ds.read();
}
```

```
    LowByte = data[0];
    HighByte = data[1];
    TReading = (HighByte << 8) + LowByte;
    SignBit = TReading & 0x8000;  // test most sig bit
    if (SignBit) // negative
    {
      TReading = (TReading ^ 0xffff) + 1; // 2's comp
    }
    Tc_100 = (6 * TReading) + TReading / 4;    // multiply by (100 * 0.0625) or 6.25

    Whole = Tc_100 / 100;  // separate off the whole and fractional portions
    Fract = Tc_100 % 100;

    sensordata = sensordata + String(Whole);
    sensordata = sensordata + ".";
    if (Fract < 10)
    {
      sensordata = sensordata + "0";
    }
    sensordata = sensordata + String(Fract); // put data to sensordata variable

    if (SignBit) // If its negative change data to negative
    {
      float miinusdata = sensordata.toFloat() * -1.0;
      sensordata = String(miinusdata);
    }

  }
  void getAlladdress(Sensor& ds) { // ok. measure a data to the sensordata variable
/*https://www.pjrc.com/teensy/td_libs_OneWire.html*/
    boolean getall = true;
    while (getall) {
      byte addr[8];
```

```
   String mystring = "";
    char s[8] = {};
    if ( !ds.search(addr)) {
      ds.reset_search();
      delay(250);
      getall = false;
      return;
    }

    for (byte i = 0; i < 8; i++) {
      snprintf(s, 8, "%X", addr[i]); // %d int, %X hex. add address to mystring
      mystring += s;
      if (i < 7) {
        mystring += " ";
      }
    }
    addresses.push_back(new Addresses(mystring,spin));
  }
}
String getSensordata() {
  return sensordata;
}
int getId() {
  return id;
}
String getName() {
  return name;
}
String getAddress() {
  return address;
}
float getUpperlimit() {
  return upperlimit;
```

```
    }
    float getLowerlimit() {
      return lowerlimit;
    }
    int getPin(){ // return OneWire spin
      return spin;
    }
    ~Sensor(); // destruction
};

Sensor::~Sensor() {
}

Vector<Sensor*> sensors; // luo vektori sen jälkeen kun on vektor & sensor alustettu

void setup(void) {

  // sensors for testing
  //sensors.push_back(new Sensor (1, "28 A8 FB 13 5 0 0 B0", 5, "Temperature_1", 23, 30));

  Serial.begin(9600); // SERIAL

  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip); // ETHERNET
  server.begin(); // SERVER
  Udp.begin(udpPort); // UDP

  conn.connect(server_addr, 3306, user, password); // MYSQL - connection to mysql server

}

int packetSize = 0;
```

```
void loop(void) {

  if (!conn.connected()) { // make mysql connection if its not connected
    conn.connect(server_addr, 3306, user, password);
  }
  // UDP
  packetSize = Udp.parsePacket();
  if (packetSize > 0)
  {
    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
    if(!strcmp(packetBuffer,"updatesensors")){ //
        updateSensors();
    }else if(!strcmp(packetBuffer,"updateaddress")){
        getAlladdresses();
    }
    // remove all contens from packetbuffer
    for( int i = 0; i < sizeof(packetBuffer);  ++i ){
      packetBuffer[i] = (char)0;
    }
  }

  if(checkSensors()){
    mysqlSavetodatabase();
  }
  myDelay(1000);
}

boolean checkSensors(){ // if there is any sensors added test
  if(sensors.size()>0){
    return true;
  }
}
```

```
// find all onewire sensor addresses (need atleast one sensor) and save them to the database
void getAlladdresses() {
  for (int i = 0; i < sensors.size(); i++) { // ok
    sensors[i]->getAlladdress(*sensors[i]);
  }
  EthernetClient client = server.available();
  if (conn.connected()) {
    for (int i = 0; i < addresses.size(); i++) {
      // Initiate the query class instance
      MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);

      addresses[i]->getAddress().toCharArray(alladdressbuffer, ADDRESSLEN);
      sprintf(aquery, INSERT_ADDRESS, alladdressbuffer, addresses[i]->getPin());
      cur_mem->execute(aquery);
      //delete cur_mem; not work if delete
    }
  }
}

void updateSensors() { // save sensors from mysql database to vector

  int intid; // variables for changing & saving char* to mysql-database
  int intpin;
  float floatlow;
  float floatupp;
  char *ptr;

  EthernetClient client = server.available(); // tarvitsee tämän

  if (conn.connected()) { //

    // Initiate the query class instance
    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
```

```
    sprintf(squery, SELECT_SENSORS);
    cur_mem->execute(squery);
    column_names *cols = cur_mem->get_columns();
    row_values *row = NULL;
    do {
      row = cur_mem->get_next_row();
      if (row != NULL) {
        intid = strtod(row->values[0], &ptr); // char* to int,float
        strcpy(addressbuffer, row->values[1]); // copy address to char addressbuffer[30]
        intpin = strtod(row->values[2], &ptr);
        strcpy(namebuffer, row->values[3]); // copy name to char namebuffer[30] // STRCPY EI
TOIMI OIKEIN(!!!)
        floatlow = strtod(row->values[4], &ptr);
        floatupp = strtod(row->values[5], &ptr);

        //Sensor(int _id, char* const _address, uint8_t _pin, char* const _name, float _lowerlimit, float
_upperlimit)
        sensors.push_back(new Sensor (intid, addressbuffer, intpin, namebuffer, floatlow, floatupp)); //
laittaa saman tiedon aina
      }
    } while (row != NULL);
  }
}


void myDelay(int x) { // own function for delay, becose attachinterrupt causes that delay()-function
it not work normally, myDelay(1000) = 1s delay
  for (unsigned int i = 0; i <= x; i++) {
    delayMicroseconds(1000);
  }
}

void mysqlSavetodatabase() { // save sensor data to database (toimiiko negatiiviset luvut?)
```

```
  EthernetClient client = server.available(); // tarvitsee tämän

  int vsid = 0;
  float vsdata = 0;
  boolean warning = false; // test boolean if warning happen (data value is under or over limits)

  for (int i = 0; i < sensors.size(); i++) {
    sensors[i]->datatoSensors(*sensors[i]); // data to sensorsdata variable
    vsid = sensors[i]->getId();
    sensors[i]->getAddress().toCharArray(addressbuffer,30); // copy address to char
addressbuffer[30]
    vsdata = sensors[i]->getSensordata().toFloat(); // try-catch?
    if (vsdata < sensors[i]->getLowerlimit() || sensors[i]->getUpperlimit() < vsdata) { // test limits
      warning = true;
      dtostrf(sensors[i]->getLowerlimit(), 1, 1, wlowerlimit); // put values to char tables
      dtostrf(sensors[i]->getUpperlimit(), 1, 1, wupperlimit);
    }

    if (conn.connected()) {
      // Initiate the query class instance
      MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
      // Save
      dtostrf(vsdata, 1, 1, temperature); // put data to char table for mysql sending

      if (warning) { // if warning happen send warning query
        sprintf(query, INSERT_DATA_WARNING, vsid, addressbuffer, temperature, wlowerlimit,
wupperlimit);
      } else {
        sprintf(query, INSERT_DATA, vsid, addressbuffer, temperature);
      }
      // Execute the query
      cur_mem->execute(query);
      delete cur_mem;
```

```
      warning = false;
      myDelay(1000);
    }
  }
  //conn.close(); working faster if not close connection-reconnect
}
```

Liite 2. PHP-käyttöliittymä: index.php

```
<?php session_start();?>
<?php $refresh=15; ?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
                <td><table><td>
                <tr></tr>
                <td style="font-size:20px" colspan="5"><strong>Show Menu</strong> (refresh time <?php echo $refresh."s";?>)</td><tr></tr><td>Choose
Sensor:</td><td>
                <FORM METHOD="POST" ACTION="monitoring.php">
                <select NAME="ALLSENSORS">
                <option value="lastvalues">Last Values</option>
<?php

        include 'settings/mysqldetails.php'; // user-password
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stm = $dbh->prepare("SELECT name FROM sensors");
                                $stm->execute();
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value) {
                                                ?><?php echo "<option value='".$value."'>".$value."</option>";?>
                                        <?php }
                                }
                                $dbh = null;
                        }catch (PDOException $e) {
                        print "Error!: " . $e->getMessage() . "<br/>";
                        die();
                        }
?><?php $optionvalues = array("10", "20", "50", "100", "ALL"); // option values ?>
        </select></td><td>How Many:</td><td>
                <select NAME="AMOUNT">
                        <?php
                        foreach ($optionvalues as $value) {
                                echo "<option value=".$value.">".$value."</option>";
                        }
                        ?>
                </select>
        </td>   <td>Date:<input type="checkbox" name="CHECKDATE" value="date"></td>
        <td>
```

```
<?php

                $date3_default = date("Y-m-d");
                $date4_default = date("Y-m-d");


                $myCalendar = new tc_calendar("date3", true, false);
                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                $myCalendar->setDate(date('d', strtotime($date3_default))
        , date('m', strtotime($date3_default))
        , date('Y', strtotime($date3_default)));
                $myCalendar->setPath("calendar/");
                $myCalendar->setYearInterval(2016, 2050);
                $myCalendar->setAlignment('left', 'bottom');
                $myCalendar->setDatePair('date3', 'date4', $date4_default);
                $myCalendar->writeScript();

                $myCalendar = new tc_calendar("date4", true, false);
                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                $myCalendar->setDate(date('d', strtotime($date4_default))
        , date('m', strtotime($date4_default))
        , date('Y', strtotime($date4_default)));
                $myCalendar->setPath("calendar/");
                $myCalendar->setYearInterval(2016, 2050);
                $myCalendar->setAlignment('left', 'bottom');
                $myCalendar->setDatePair('date3', 'date4', $date3_default);
                $myCalendar->writeScript();
                ?>
              </td>
              <td><INPUT TYPE="SUBMIT" VALUE="Show" NAME="SHOW"></td><tr></tr><td colspan=5></td>
              <tr></tr><td colspan="8"><hr></hr></td><tr></tr>
              </form>
              </table></td><tr></tr>
              <td><table class="data"><td>

<strong>Id</strong></td><td><strong>Pin</strong></td><td><strong>Address</strong></td><td><strong>Sensor</strong></td><td><strong>Data</strong></td><td><strong>Time
</strong></td><tr></tr>
              <?php
              include 'settings/mysqldetails.php'; // user-password
                                        try {
                                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                                $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime FROM sensors_data t1, sensors t2
WHERE t1.sid = t2.sid ORDER BY did DESC,stime desc LIMIT 10");
                                                $stm->execute();
                                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                                        foreach($tsensors as $row => $value) {
                                                                ?><td><?php echo $value;?></td>
                                                        <?php } ?><tr></tr><?php
                                                }
                                                $dbh = null;
                                        }catch (PDOException $e) {
                                        print "Error!: " . $e->getMessage() . "<br/>";
                                        die();
                                        }
              ?>
              </td></table></td>
</table>
</div>
</body>
</html>
```

Liite 3. PHP-käyttöliittymä: addnew.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
if(isset($_POST['addnew'])){

        $sid = "LAST_INSERT_ID()";
        $address = $_POST['address'];
        $pin = $_POST['pin'];
        $name = $_POST['name'];
        $lowerlimit = $_POST['lowerlimit'];
        $upperlimit = $_POST['upperlimit'];
        $type = $_POST['type'];
        $model = $_POST['model'];
        $info = $_POST['info'];

        include 'settings/mysqldetails.php'; // user-password

                try {
                        $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                        $stmt = $dbh->prepare("INSERT INTO sensors (sid, address, pin, name, lowerlimit, upperlimit, type, model, info) VALUES
(?, ?, ?, ?, ?, ?, ?, ?, ?)");
                        $stmt->execute(array($sid,$address,$pin,$name,$lowerlimit,$upperlimit,$type,$model,$info));

                        echo "<script type='text/javascript'>alert('Sensor added successfully!')</script>";
                }catch (PDOException $e) {
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
                }
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td><strong style="font-size: 35px;">Add New Sensor</strong></td>
        <tr></tr>
        <td>
        <table border=1>
        <FORM METHOD="POST" ACTION="addnew.php">
<?php
        include 'settings/mysqldetails.php'; // user-password

                try {
                        $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                        $stm = $dbh->prepare("DESCRIBE sensors");
                        $stm->execute();
                        while($tfields = $stm->fetchAll(PDO::FETCH_COLUMN, 0)){
                                foreach($tfields as $row) {
                                        ?><td><?php echo $row ?></td><td><INPUT TYPE="<?php echo 'TEXT';?>" NAME="<?php echo
$row; ?>" SIZE="30" <?php if((string)$row=='sid'){echo 'DISABLED';}else{}?>></td><tr></tr>
```

```
                                                    <?php }?>
                                                    <?php
                                            }
                                    }catch (PDOException $e) {
                                    print "Error!: " . $e->getMessage() . "<br/>";
                                    die();
                                    }
?>
                        <tr></tr>
                        <td></td><td align=center ><INPUT TYPE="SUBMIT" VALUE="Submit" NAME="addnew"></td>
                </form>
                </table>
                </td>
</table>
</div>
</body>
</html>
```

Liite 4. PHP-käyttöliittymä: address.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table class="sensors">
<?php
		include 'settings/mysqldetails.php'; // user-password

		$dtable = 'sensors_address';
					// remove always dublicates from address-database when refresh page
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("DELETE db1 FROM sensors_address db1, sensors_address db2 WHERE db1.id > db2.id AND
db1.address = db2.address");
						$stm->execute();
					}catch (PDOException $e) {
						print "Error!: " . $e->getMessage() . "<br/>";
						die();
					}
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("DESCRIBE " . $dtable);
						$stm->execute();
						while($tfields = $stm->fetchAll(PDO::FETCH_COLUMN)){
							foreach($tfields as $row) {
								?><td class="sensorcolumns"><?php echo $row ?></td>
							<?php }?>
							<tr></tr>
							<?php
						}
					}catch (PDOException $e) {
					print "Error!: " . $e->getMessage() . "<br/>";
					die();
					}
?>
			<tr></tr>
<?php
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("SELECT * FROM sensors_address");
						$stm->execute();
						while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
							foreach($tsensors as $row => $value) {
								?><td><?php echo $value ?></td>
							<?php }?>
							<tr></tr>
							<?php
						}
```

```
                    $dbh = null;
                }catch (PDOException $e) {
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
                }
?>
        </table>
        </td>
</table>
</div>
</body>
<html>
```

Liite 5. PHP-käyttöliittymä: sensoraddress.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table class="sensors">
<?php
		include 'settings/mysqldetails.php'; // user-password

		$dbtable = 'sensors';
		$dbname = 'arduino';

					// remove always dublicates from address-database when refresh page
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("DELETE db1 FROM sensors_address db1, sensors_address db2 WHERE db1.id > db2.id AND
db1.address = db2.address");
						$stm->execute();
					}catch (PDOException $e) {
						print "Error!: " . $e->getMessage() . "<br/>";
						die();
					}
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("SELECT column_name FROM information_schema.columns WHERE table_name = '".$dbtable."' AND
table_schema = '".$dbname."' AND (COLUMN_NAME LIKE 'sid' OR COLUMN_NAME LIKE 'address' OR COLUMN_NAME LIKE 'pin' OR COLUMN_NAME LIKE 'name')");
						$stm->execute();
						while($tfields = $stm->fetchAll(PDO::FETCH_COLUMN)){
							foreach($tfields as $row) {
								?><td class="sensorcolumns"><?php echo $row ?></td>
							<?php }?>

							<?php
						}
					}catch (PDOException $e) {
						print "Error!: " . $e->getMessage() . "<br/>";
						die();
					}
?>
			<td>Sensor address (for pin):</td>
			<tr></tr><FORM METHOD="POST" ACTION="editedaddress.php">
<?php
			$pin = ""; // for right pin-address
			$asid = ""; // right sid-id


					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
```

```
                                        $stm = $dbh->prepare("SELECT sid, address, pin, name FROM sensors");
                                        $stm->execute();
                                        while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                                foreach($tsensors as $row => $value) {
                                                        ?><td><?php echo $value ?></td>
                                                <?php if((string)$row=="pin"){$pin = $value;}elseif((string)$row=="sid"){$asid = $value;}}?>

                                                <?php // other query using pin - address for pin
                                                try {
                                                        $stm2 = $dbh->prepare("SELECT address FROM sensors_address WHERE pin='".$pin."'");
                                                        $stm2->execute();
                                                ?>
                                                        <td><select NAME="ALLADDRESS<?php echo $asid;?>">
                                                        <option value="#">#</option>
                                                        <?php while($tsensors = $stm2->fetch(PDO::FETCH_ASSOC)){
                                                                foreach($tsensors as $row => $value) {?>
                                                                <option value="<?php echo $value;?>"><?php echo $value;?></option>
                                                                <?php }}?>
                                                        </select></td>
                                                <?php
                                                }catch (PDOException $e) {
                                                        print "Error!: " . $e->getMessage() . "<br/>";
                                                        die();
                                                }?>
                                                <tr></tr>
                                                <?php
                                                $pin = "";
                                                $asid = "";
                                        }
                                        $dbh = null;
                                }catch (PDOException $e) {
                                print "Error!: " . $e->getMessage() . "<br/>";
                                die();
                                }
?>              <td colspan="5">
                <table align="center">
                        <td>
                        <select NAME="SENSORADDRESSUPDATE">
                                <option value="#">#</option>
                                <option value="ALL">ALL</option>
                                <?php
                                try {
                                        $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                        $stm = $dbh->prepare("SELECT sid FROM sensors");
                                        $stm->execute();
                                        while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                                foreach($tsensors as $row => $value) {
                                                        ?><option value="<?php echo $value ?>"><?php echo $value ?></option>
                                                <?php }
                                        }
                                        $dbh = null;
                                }catch (PDOException $e) {
                                print "Error!: " . $e->getMessage() . "<br/>";
                                die();
                                }
                                ?>
                        </select>
                        </td>
                        <td><INPUT value="UPDATE" type="submit" NAME="UPDATE"></td></FORM>
                </table>
                </td>
</table>
</div>
</body>
<html>
```

Liite 6. PHP-käyttöliittymä: sensors.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table class="sensors">
<?php
		include 'settings/mysqldetails.php'; // user-password

		$dtable = 'sensors';

					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("DESCRIBE " . $dtable);
						$stm->execute();
						while($tfields = $stm->fetchAll(PDO::FETCH_COLUMN)){
							foreach($tfields as $row) {
								?><td class="sensorcolumns"><?php echo $row ?></td>
							<?php }?>
							<tr></tr>
							<?php
						}
					}catch (PDOException $e) {
					print "Error!: " . $e->getMessage() . "<br/>";
					die();
					}
?>
			<tr></tr>
<?php
					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("SELECT * FROM sensors");
						$stm->execute();
						while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
							foreach($tsensors as $row => $value) {
								?><td><?php echo $value ?></td>
							<?php }?>
							<tr></tr>
							<?php
						}
						$dbh = null;
					}catch (PDOException $e) {
					print "Error!: " . $e->getMessage() . "<br/>";
					die();
					}?>
		</table>
		</td>
</table>
</div></body><html>
```

Liite 7. PHP-käyttöliittymä: monitoring.php

```
<?php session_start();
?>
<?php $refresh=15; ?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
                <td><table><td>
                <tr></tr>
                <td style="font-size:20px" colspan="6"><strong>Show Menu</strong> (refresh time <?php echo $refresh."s";?>)</td><tr></tr><td>Choose
Sensor:</td><td>
                <FORM METHOD="POST" ACTION="monitoring.php">
                <select NAME="ALLSENSORS">
                <option value="lastvalues">Last Values</option>
<?php

        $date1 = isset($_REQUEST["date3"]) ? $_REQUEST["date3"] : "";
        $date2 = isset($_REQUEST["date4"]) ? $_REQUEST["date4"] : "";

        if(!empty($date1)){ // save date values to session variables
                $_SESSION["date3"] = $_REQUEST["date3"];
                $_SESSION["date4"] = $_REQUEST["date4"];
        }else{ // for mysql
                $date1 = $_SESSION["date3"];
                $date2 = $_SESSION["date4"];
        }

        if(isset($_POST['CHECKDATE'])){
                $checkdate = true;
                                                                        }else{ $checkdate = false; }

                                                                $amount = $_POST['AMOUNT'];
                                                        $allsensors = $_POST['ALLSENSORS'];

                                                if(!empty($amount)){ // save details to $_SESSION for freshfresh page if used $_POST method
                                                                $_SESSION["sessionamount"] = $amount;
                                                        $_SESSION["sessionallsensors"] = $allsensors;
                                                                $_SESSION["checkdate"] = $checkdate;
                                                                                        }else{
                                                                $amount = $_SESSION["sessionamount"];
```

```
                $allsensors = $_SESSION["sessionallsensors"];
                $checkdate = $_SESSION["checkdate"];
        }

        include 'settings/mysqldetails.php'; // user-password
                try {
                        $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                        $stm = $dbh->prepare("SELECT name FROM sensors");
                        $stm->execute();
                        while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                foreach($tsensors as $row => $value){
                                        if($value==$allsensors){
                                                echo "<option value='".$value."' selected>".$value."</option>";
                                        }else{
                                                echo "<option value='".$value."'>".$value."</option>";
                                        }
                                }
                        }
                        $dbh = null;
                }catch (PDOException $e) {
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
                }
?><?php $optionvalues = array("10", "20", "50", "100", "ALL"); // option values
?>
        </select></td><td>How Many:</td><td>
                <select NAME="AMOUNT">
                        <?php
                        foreach ($optionvalues as $value) {
                                if($value==$amount){
                                        echo "<option value=".$value." selected>".$value."</option>";
                                }else{
                                        echo "<option value=".$value.">".$value."</option>";
                                }
                        }
                        ?>
                </select>
        </td>
        <td>Date:<input type="checkbox" name="CHECKDATE" value="date" <?php if($checkdate){echo "checked";}?>></td>
        <td>
<?php
        if(empty($date1)){ // set right days
          $date3_default = date("Y-m-d");
        }else{
          $date3_default = $date1;
        }

        if(empty($date2)){
          $date4_default = date("Y-m-d");
        }else{
          $date4_default = $date2;
        }

          $myCalendar = new tc_calendar("date3", true, false);
          $myCalendar->setIcon("calendar/images/iconCalendar.gif");
          $myCalendar->setDate(date('d', strtotime($date3_default))
      , date('m', strtotime($date3_default))
      , date('Y', strtotime($date3_default)));
          $myCalendar->setPath("calendar/");
          $myCalendar->setYearInterval(2016, 2050);
          $myCalendar->setAlignment('left', 'bottom');
          $myCalendar->setDatePair('date3', 'date4', $date4_default);
          $myCalendar->writeScript();

          $myCalendar = new tc_calendar("date4", true, false);
          $myCalendar->setIcon("calendar/images/iconCalendar.gif");
```

```
            $myCalendar->setDate(date('d', strtotime($date4_default))
        , date('m', strtotime($date4_default))
        , date('Y', strtotime($date4_default)));
            $myCalendar->setPath("calendar/");
            $myCalendar->setYearInterval(2016, 2050);
            $myCalendar->setAlignment('left', 'bottom');
            $myCalendar->setDatePair('date3', 'date4', $date3_default);
            $myCalendar->writeScript();
            ?>
            </td>
            <td><INPUT TYPE="SUBMIT" VALUE="Show" NAME="SHOW"></td><tr></tr><td colspan=5></td>
            <tr></tr><td colspan="8"><hr></hr></td><tr></tr>
            </form>
            </table></td><tr></tr>
            <td><table class="data"><td>

<strong>Id</strong></td><td><strong>Pin</strong></td><td><strong>Address</strong></td><td><strong>Sensor</strong></td><td><strong>Data</strong></td><td><strong>Time
</strong></td><tr></tr>
            <?php

            if((string)$amount=='ALL'){$amount = "1844674407370955161"; } // all mysql rows

            include 'settings/mysqldetails.php'; // user-password
                        try {
                                    $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                    if($allsensors=="lastvalues"){
                                                if($checkdate){
                                                            $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime FROM
sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND stime >='".$date1."' AND stime <= '".$date2." 23:59:59' ORDER BY did DESC,stime desc LIMIT ".$amount);

                                                }else{
                                                            $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime FROM
sensors_data t1, sensors t2 WHERE t1.sid = t2.sid ORDER BY did DESC,stime desc LIMIT ".$amount);
                                                }
                                    }else{
                                                if($checkdate){
                                                            $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime FROM
sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND t2.name='".$allsensors."' AND stime >='".$date1."' AND stime <= '".$date2." 23:59:59'  ORDER BY did DESC,stime desc
LIMIT ".$amount);
                                                }else{
                                                            $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime FROM
sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND t2.name='".$allsensors."' ORDER BY did DESC,stime desc LIMIT ".$amount);
                                                }
                                    }
                                    $stm->execute();
                                    while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                                foreach($tsensors as $row => $value) {
                                                            ?><td><?php echo $value;?></td>
                                                <?php } ?><tr></tr><?php
                                    }
                                    $dbh = null; // remove $_POST
                                    unset($_POST);
                                    $_POST = array();
                        }catch (PDOException $e) {
                        print "Error!: " . $e->getMessage() . "<br/>";
                        die();
                        }
            ?>
            </td></table></td>
</table>
</div>
</body>
</html>
```

Liite 8. PHP-käyttöliittymä: warnings.php

```
<?php session_start();
?>
<?php $refresh=15; ?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table border=0>
			<td><table><td>
			<tr></tr>
			<td style="font-size:20px" colspan="6"><strong>Show Warnings</strong> (refresh time <?php echo $refresh."s";?>)</td><tr></tr><td>Choose
Sensor:</td><td>
			<FORM METHOD="POST" ACTION="warnings.php">
			<select NAME="ALLSENSORS">
			<option value="lastvalues">Last Values</option>
<?php

		$date1 = isset($_REQUEST["date3"]) ? $_REQUEST["date3"] : "";
		$date2 = isset($_REQUEST["date4"]) ? $_REQUEST["date4"] : "";

		if(!empty($date1)){ // save date values to session variables
			$_SESSION["date3"] = $_REQUEST["date3"];
			$_SESSION["date4"] = $_REQUEST["date4"];
		}else{ // for mysql
			$date1 = $_SESSION["date3"];
			$date2 = $_SESSION["date4"];
		}

		if(isset($_POST['CHECKDATE'])){
			$checkdate = true;
		}else{ $checkdate = false; }

		$amount = isset($_POST['AMOUNT']) ? $_POST['AMOUNT'] : "";
		$allsensors = isset($_POST['ALLSENSORS']) ? $_POST['ALLSENSORS'] : "";

		if(!empty($amount)){ // save details to $_SESSION for freshfresh page if used $_POST method
			$_SESSION["sessionamount"] = $amount;
			$_SESSION["sessionallsensors"] = $allsensors;
			$_SESSION["checkdate"] = $checkdate;
		}else{
			$amount = $_SESSION["sessionamount"];
```

```
                $allsensors = $_SESSION["sessionallsensors"];
                $checkdate = $_SESSION["checkdate"];
        }

        include 'settings/mysqldetails.php'; // user-password
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stm = $dbh->prepare("SELECT name FROM sensors");
                                $stm->execute();
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value){
                                                if($value==$allsensors){
                                                        echo "<option value='".$value."' selected>".$value."</option>";
                                                }else{
                                                        echo "<option value='".$value."'>".$value."</option>";
                                                }
                                        }
                                }
                                $dbh = null;
                        }catch (PDOException $e) {
                        print "Error!: " . $e->getMessage() . "<br/>";
                        die();
                        }
?><?php $optionvalues = array("10", "20", "50", "100", "ALL"); // option values
?>
        </select></td><td>How Many:</td><td>
                <select NAME="AMOUNT">
                        <?php
                        foreach ($optionvalues as $value) {
                                if($value==$amount){
                                        echo "<option value=".$value." selected>".$value."</option>";
                                }else{
                                        echo "<option value=".$value.">".$value."</option>";
                                }
                        }
                        ?>
                </select>
        </td>
        <td>Date:<input type="checkbox" name="CHECKDATE" value="date" <?php if($checkdate){echo "checked";}?>></td>
        <td>
<?php

        if(empty($date1)){ // set right days
          $date3_default = date("Y-m-d");
        }else{
          $date3_default = $date1;
                                                                                                                                }

                                                                                                                        if(empty($date2)){
                                                                                                                $date4_default = date("Y-m-d");
                                                                                                                        }else{
                                                                                                                $date4_default = $date2;
                                                                                                                        }

                                                                                                        $myCalendar = new tc_calendar("date3", true, false);
                                                                                                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                                                                                                $myCalendar->setDate(date('d', strtotime($date3_default))
                                                                                                                , date('m', strtotime($date3_default))
                                                                                                                , date('Y', strtotime($date3_default)));
                                                                                                        $myCalendar->setPath("calendar/");
                                                                                                $myCalendar->setYearInterval(2016, 2050);
                                                                                                        $myCalendar->setAlignment('left', 'bottom');
                                                                                                $myCalendar->setDatePair('date3', 'date4', $date4_default);
                                                                                                        $myCalendar->writeScript();
```

```
                $myCalendar = new tc_calendar("date4", true, false);
                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                $myCalendar->setDate(date('d', strtotime($date4_default))
        , date('m', strtotime($date4_default))
        , date('Y', strtotime($date4_default)));
                $myCalendar->setPath("calendar/");
                $myCalendar->setYearInterval(2016, 2050);
                $myCalendar->setAlignment('left', 'bottom');
                $myCalendar->setDatePair('date3', 'date4', $date3_default);
                $myCalendar->writeScript();
                ?>
                </td>
                <td><INPUT TYPE="SUBMIT" VALUE="Show" NAME="SHOW"></td><tr></tr><td colspan=5></td>
                <tr></tr><td colspan="8"><hr></hr></td><tr></tr>
                </form>
                </table></td><tr></tr>
                <td><table class="data">
                <td colspan="8" align="center" class="warningstitle">WARNINGS</td><tr></tr>

<td><strong>Id</strong></td><td><strong>Pin</strong></td><td><strong>Address</strong></td><td><strong>Sensor</strong></td><td><strong>Data</strong></td><td><strong>T
ime</strong></td><td><strong>Wlowerlimit</strong></td><td><strong>Wupperlimit</strong></td><tr></tr>
                <?php

                if((string)$amount=='ALL'){$amount = "1844674407370955161"; } // all mysql rows

                include 'settings/mysqldetails.php'; // user-password
                                        try {
                                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                                if($allsensors=="lastvalues"){
                                                        if($checkdate){
                                                                $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime,
t1.wlowerlimit, t1.wupperlimit FROM sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND stime >='".$date1."' AND stime <= '".$date2." 23:59:59' AND (wlowerlimit>0 OR
wupperlimit>0) ORDER BY did DESC,stime desc LIMIT ".$amount);
                                                        }else{
                                                                $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime,
t1.wlowerlimit, t1.wupperlimit FROM sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND (wlowerlimit>0 OR wupperlimit>0) ORDER BY did DESC,stime desc LIMIT ".$amount);

                                                        }
                                                }else{
                                                        if($checkdate){
                                                                $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime,
t1.wlowerlimit, t1.wupperlimit FROM sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND t2.name='".$allsensors."' AND stime >='".$date1."' AND stime <= '".$date2." 23:59:59'
AND (wlowerlimit>0 OR wupperlimit>0) ORDER BY did DESC,stime desc LIMIT ".$amount);
                                                        }else{
                                                                $stm = $dbh->prepare("SELECT t1.did, t2.pin, t2.address, t2.name, t1.sdata, t1.stime,
t1.wlowerlimit, t1.wupperlimit FROM sensors_data t1, sensors t2 WHERE t1.sid = t2.sid AND t2.name='".$allsensors."' AND (wlowerlimit>0 OR wupperlimit>0) ORDER BY did
DESC,stime desc LIMIT ".$amount);
                                                        }
                                                }
                                                $stm->execute();
                                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                                        foreach($tsensors as $row => $value) {
                                                                ?><td><?php echo $value;?></td>
                                                        <?php } ?><tr></tr><?php
                                                }
                                                $dbh = null; // remove $_POST
                                                unset($_POST);
                                                $_POST = array();
                                        }catch (PDOException $e) {
                                        print "Error!: " . $e->getMessage() . "<br/>";
                                        die();
                                        }
                ?>
                </td></table></td>
</table>
</div></body></html>
```

Liite 9. PHP-käyttöliittymä: editsensors.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table border=0>
<?php

		include 'settings/mysqldetails.php'; // user-password

					try {
						$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
						$stm = $dbh->prepare("DESCRIBE SENSORS");
						$stm->execute();
						while($tfields = $stm->fetchAll(PDO::FETCH_COLUMN)){
							foreach($tfields as $row) {
								?><td><?php echo $row ?></td>
							<?php }?>
							<tr></tr>
							<?php
						}
					}catch (PDOException $e) {
					print "Error!: " . $e->getMessage() . "<br/>";
					die();
					}
?>
			<tr></tr>
			<FORM METHOD="POST" ACTION="editedsensors.php">
<?php
			unset($_POST); // remove $_POST variables
			$_POST = array();

			$stack = array(); // stack for dropdown-menu
			$sizes = array("2","15","2","20","5","5","10","10","20","0"); // td sizes
				try {
					$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
					$stm = $dbh->prepare("SELECT * FROM sensors");
					$stm->execute();
					while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
						$i = 0;
						foreach($tsensors as $row => $value) {
							if((string)$row=='sid'){array_push($stack,$value); }
							?><td><INPUT TYPE="<?php echo 'TEXT';?>" NAME="<?php echo $row.end($stack); ?>"
VALUE="<?php echo $value;?>" <?php if((string)$row=='sid'){echo 'READONLY';}else{}?> SIZE="<?php echo $sizes[$i]?>"></td>
						<?php $i++; }?>
						<tr></tr>
						<?php
					}
					$dbh = null;
```

```
                        }catch (PDOException $e) {
                        print "Error!: " . $e->getMessage() . "<br/>";
                        die();
                        }
?>
                <tr></tr>
                <td colspan="8"><table width="100%">
                <td align="center">
                <INPUT TYPE="SUBMIT" VALUE="UPDATE" NAME="UPDATE">
                        <select NAME="SENSORID">
                        <option value="#">#</option>
                        <?php
                        foreach ($stack as $value) {
                        echo "<option value=".$value.">".$value."</option>";
                        }
                        ?>
                        </select>
                <INPUT TYPE="SUBMIT" VALUE="DELETE" NAME="DELETE">
                </td>
                <tr></tr>
                </form>
        </table>
        </td>
</table>
</div>
</body>
</html>
```

Liite 10. PHP-käyttöliittymä: info.php

```
<?php
                $dest_address = '192.168.0.13';
                $dest_port = 8888;
                $msg = "";
                $infomessage = "Update message";

if(isset($_POST['UPDATESENSORS'])){
        $infomessage = "Sensors updated to arduino";
                $sock = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);
                $msg = "updatesensors";
                $len = strlen($msg);
                socket_sendto($sock, $msg, $len, 0, $dest_address, $dest_port);
                socket_close($sock);
        }elseif(isset($_POST['UPDATEADDRESS'])){
                $infomessage = "Addresses updated to database";

                $sock = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);
                $msg = "updateaddress";
                $len = strlen($msg);
                socket_sendto($sock, $msg, $len, 0, $dest_address, $dest_port);
                socket_close($sock);
        }

?>

<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td align="center"><table><FORM METHOD="POST" ACTION="<?php echo $_SERVER['PHP_SELF'];?>">
                <tr></tr>
                <td align="center">Arduino Sensor Measurer</td>
                <tr></tr>
                <td align="center">Version 1.00</td>
                <tr></tr>
                <td align="center"><INPUT TYPE="SUBMIT" VALUE="UPDATE ARDUINO SENSORS" NAME="UPDATESENSORS"></td>
                <tr></tr>
                <td align="center"><INPUT TYPE="SUBMIT" VALUE="UPDATE ALL ADDRESS" NAME="UPDATEADDRESS"></td>
                <tr></tr>
                <td align="center"><?php echo $infomessage;?></td>
                <tr></tr>
                </FORM></table></td>
        <tr></tr>
</table>
</div>
</body>
<html>
```

Liite 11. PHP-käyttöliittymä: menu.php

```
<div id='cssmenu'>
<ul>
   <li><a href='index.php'><span>Monitoring</span></a></li>
   <li><a href='sensors.php'><span>Sensors</span></a></li>
   <li class='active has-sub'><a href='address.php'><span>Address</span></a>
      <ul>
         <li class='has-sub'><a href='address.php'><span>Show Address</span></a></li>
         <li class='has-sub'><a href='sensoraddress.php'><span>Sensor Address</span></a></li>
      </ul>
   </li>
   <li class='active has-sub'><a href='addnew.php'><span>Edit</span></a>
      <ul>
         <li class='has-sub'><a href='addnew.php'><span>Add New Sensor</span></a></li>
         <li class='has-sub'><a href='editsensors.php'><span>Edit Sensors</span></a></li>
         <li class='has-sub'><a href='editaddressdata.php'><span>Edit Address</span></a></li>
                                 <li class='has-sub'><a href='editsensordata.php'><span>Edit Sensor Data</span></a></li>
                                 <li class='has-sub'><a href='editwarningdata.php'><span>Edit Warnings</span></a></li>
      </ul>
   </li>
   <li><a href='warnings.php'><span>Warnings</span></a></li>
   <li class='last'><a href='info.php'><span>Info</span></a></li>
</ul>
</div>
```

http://cssmenumaker.com/menu/css3-drop-down-menu

Liite 12. PHP-käyttöliittymä: editaddressdata.php

```
<?php session_start();?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
                <td><table><td>
                <tr></tr>
                <td style="font-size:20px" colspan="5"><strong>Edit All Address</strong></td>
                <tr></tr><td colspan="8">DELETE ADDRESS DATA</td>
                <tr></tr><td>Delete all address: </td><td>
                <FORM METHOD="POST" ACTION="editedaddressdata.php">
                <td><INPUT TYPE="SUBMIT" VALUE="DELETEADDRESS" NAME="DELETEADDRESS"></td><tr></tr><td colspan=5></td>
                </FORM>
        </table></td><tr></tr>
</table>
</div>
</body>
</html>
```

Liite 13. PHP-käyttöliittymä: editedaddress.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
		$message = "";

if(isset($_POST['UPDATE'])){
		$sid = $_POST['SENSORADDRESSUPDATE'];
		if((string)$sid!='#' && (string)$sid!='ALL'){
			$address = $_POST['ALLADDRESS'.$sid];
		}else{
			$address = "#";
		}

		// check form and update databases
		include 'settings/mysqldetails.php'; // user-password
			if(((string)$sid!='#' && (string)$address!='#')&&((string)$sid!='ALL')){
				try {
					$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
					$stmt = $dbh->prepare("UPDATE sensors SET address=? WHERE sid=?");
					$stmt->execute(array($address,$sid));
					$message = "Sensor updated successfully!";

					unset($_POST);
					$_POST = array();

				}catch (PDOException $e) {
				print "Error!: " . $e->getMessage() . "<br/>";
				die();
				}
			}elseif(((string)$sid=='ALL')){
				try {
					$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
					$stm = $dbh->prepare("SELECT sid FROM sensors");
					$stm->execute();

					while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
						foreach($tsensors as $row => $value) {
							$address = $_POST['ALLADDRESS'.$value];
							if((string)$address!='#'){
								$stmt2 = $dbh->prepare("UPDATE sensors SET address=? WHERE sid=?");
								$stmt2->execute(array($address,$value));
							}
						}
					}
					$message = "All sensors updated successfully!";
					unset($_POST);
					$_POST = array();
				}catch (PDOException $e) {
				print "Error!: " . $e->getMessage() . "<br/>";
				die();
```

```
                        }
                        }else{
                                $message = "# Nothing happened";
                        }
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
        <tr></tr>
        <td><?php echo $message;?> <a href="sensoraddress.php">Return back here</a>.</td>
        <tr></tr>
        </table>
        </td>
</table>
</div>
</body>
</html>
```

Liite 14. PHP-käyttöliittymä: editedaddressdata.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
        $message = "";

if(isset($_POST['DELETEADDRESS'])){
        include 'settings/mysqldetails.php'; // user-password
                try {
                        $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                        $stmt = $dbh->prepare("TRUNCATE sensors_address");
                        $stmt->execute(array());

                        $message = "Address data deleted successfully!";
                        unset($_POST);
                        $_POST = array();
                }catch (PDOException $e) {
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
                }
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
        <tr></tr>
        <td><?php echo $message;?> <a href="editaddressdata.php">Return back here</a>.</td>
        <tr></tr>
        </table>
        </td>
</table>
</div>
</body>
</html>
```

Liite 15. PHP-käyttöliittymä: editedsensordata.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
		$message = "";

if(isset($_POST['DELETEDATA'])){

		$sid = $_POST['ALLSENSORDATA'];

		if(isset($_POST['CHECKDATE'])){
			$checkdate = true;
			$date1 = isset($_REQUEST["date3"]) ? $_REQUEST["date3"] : "";
			$date2 = isset($_REQUEST["date4"]) ? $_REQUEST["date4"] : "";
		}else{ $checkdate = false; }

		include 'settings/mysqldetails.php'; // user-password
			if((string)$sid!='#'){
				try {
					$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);

					if($checkdate){
						if($sid=="allvalues"){
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE stime >='".$date1."' AND stime
<= '".$date2." 23:59:59'");
							$stmt->execute(array());
						}else{
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE sid=? AND stime >='".$date1."'
AND stime <= '".$date2." 23:59:59'");
							$stmt->execute(array($sid));
						}
					}else{
						if($sid=="allvalues"){
							$stmt = $dbh->prepare("TRUNCATE SENSORS_DATA");
							$stmt->execute(array());
						}else{
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE sid=?");
							$stmt->execute(array($sid));
						}
					}

					$message = "Sensor data deleted successfully!";

					unset($_POST);
					$_POST = array();
				}catch (PDOException $e) {
				print "Error!: " . $e->getMessage() . "<br/>";
				die();
				}
			}else{
				$message = "# Nothing happened";
			}
```

```
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
		<tr></tr>
		<td>
		<table border=0>
		<tr></tr>
		<td><?php echo $message;?> <a href="editsensordata.php">Return back here</a>.</td>
		<tr></tr>
		</table>
		</td>
</table>
</div>
</body>
</html>
```

Liite 16. PHP-käyttöliittymä: editedsensors.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
        $message = "";

if(isset($_POST['UPDATE'])){
        $sid = $_POST['SENSORID'];

        if((string)$sid!='#'){
                $pin = $_POST['pin'.$sid];
                $address = $_POST['address'.$sid];
                $name = $_POST['name'.$sid];
                $lowerlimit = $_POST['lowerlimit'.$sid];
                $upperlimit = $_POST['upperlimit'.$sid];
                $type = $_POST['type'.$sid];
                $model = $_POST['model'.$sid];
                $info = $_POST['info'.$sid];
        }

        include 'settings/mysqldetails.php'; // user-password
                if((string)$sid!='#'){
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stmt = $dbh->prepare("UPDATE sensors SET address=?, pin=?, name=?, lowerlimit=?, upperlimit=?, type=?, model=?, info=?
WHERE sid=?");
                                $stmt->execute(array($address,$pin,$name,$lowerlimit,$upperlimit,$type,$model,$info,$sid));

                                $message = "Sensor updated successfully!";

                                unset($_POST);
                                $_POST = array();

                        }catch (PDOException $e) {
                        print "Error!: " . $e->getMessage() . "<br/>";
                        die();
                        }
                }else{
                        $message = "# Nothing happened";
                }
}elseif(isset($_POST['DELETE'])){

        $sid = $_POST['SENSORID'];

        include 'settings/mysqldetails.php'; // user-password
                if((string)$sid!='#'){
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stmt = $dbh->prepare("DELETE FROM SENSORS WHERE sid=?");
                                $stmt->execute(array($sid));
                                $stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE sid=?");
```

```
                                        $stmt->execute(array($sid));
                                                $message = "Sensor deleted successfully!";

                                                unset($_POST);
                                                $_POST = array();
                                        }catch (PDOException $e) {
                                        print "Error!: " . $e->getMessage() . "<br/>";
                                        die();
                                        }
                                }else{
                                        $message = "# Nothing happened";
                                }
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
        <tr></tr>
        <td><?php echo $message;?> <a href="editsensors.php">Return back here</a>.</td>
        <tr></tr>
        </table>
        </td>
</table>
</div>
</body>
</html>
```

Liite 17. PHP-käyttöliittymä: editedwarningdata.php

```
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php
		$message = "";

if(isset($_POST['DELETEDATA'])){

		$sid = $_POST['ALLWARNINGDATA'];

		if(isset($_POST['CHECKDATE'])){
			$checkdate = true;
			$date1 = isset($_REQUEST["date3"]) ? $_REQUEST["date3"] : "";
			$date2 = isset($_REQUEST["date4"]) ? $_REQUEST["date4"] : "";
		}else{ $checkdate = false; }

		include 'settings/mysqldetails.php'; // user-password
			if((string)$sid!='#'){
				try {
					$dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);

					if($checkdate){
						if($sid=="allvalues"){
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE stime >='".$date1."' AND stime
<= '".$date2." 23:59:59' AND (wlowerlimit>0 OR wupperlimit>0)");
							$stmt->execute(array());
						}else{
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE sid=? AND stime >='".$date1."'
AND stime <= '".$date2." 23:59:59' AND (wlowerlimit>0 OR wupperlimit>0)");
							$stmt->execute(array($sid));
						}
					}else{
						if($sid=="allvalues"){
							$stmt = $dbh->prepare("DELETE * FROM SENSORS_DATA WHERE (wlowerlimit>0 OR
wupperlimit>0)");
							$stmt->execute(array());
						}else{
							$stmt = $dbh->prepare("DELETE FROM SENSORS_DATA WHERE sid=? AND (wlowerlimit>0 OR
wupperlimit>0)");
							$stmt->execute(array($sid));
						}
					}

					$message = "Sensor warning data deleted successfully!";

					unset($_POST);
					$_POST = array();
				}catch (PDOException $e) {
				print "Error!: " . $e->getMessage() . "<br/>";
				die();
				}
```

```
                        }else{
                                $message = "# Nothing happened";
                        }
}
?>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
        <tr></tr>
        <td><?php echo $message;?> <a href="editsensordata.php">Return back here</a>.</td>
        <tr></tr>
        </table>
        </td>
</table>
</div>
</body>
</html>
```

Liite 18. PHP-käyttöliittymä: editsensordata.php

```
<?php session_start();?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
                <td><table><td>
                <tr></tr>
                <td style="font-size:20px" colspan="5"><strong>Edit Sensor Data</strong></td>
                <tr></tr><td colspan="8">DELETE SENSOR DATA</td>
                <tr></tr><td>Choose Sensor:</td><td>
                <FORM METHOD="POST" ACTION="editedsensordata.php">
                <select NAME="ALLSENSORDATA">
                <option value="#">#</option>
                <option value="allvalues">All Values</option>
<?php

        $stack = array();

        include 'settings/mysqldetails.php'; // user-password
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stm = $dbh->prepare("SELECT sid FROM sensors");
                                $stm->execute();
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value) {
                                                array_push($stack,$value);

                                        }
                                }
                                $stm = $dbh->prepare("SELECT name FROM sensors");
                                $stm->execute();
                                $i = 0;
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value) {
                                                if(count($stack)>$i){
                                                ?><?php echo "<option value='".$stack[$i]."'>".$value."</option>";?>
                                                <?php $i++;}
                                                }
                                }
                                $dbh = null;
                        }catch (PDOException $e) {
```

```
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
                }

?><td>Date:<input type="checkbox" name="CHECKDATE" value="date"></td>
        <td>
<?php
        $date3_default = date("Y-m-d");
        $date4_default = date("Y-m-d");

        $myCalendar = new tc_calendar("date3", true, false);
        $myCalendar->setIcon("calendar/images/iconCalendar.gif");
        $myCalendar->setDate(date('d', strtotime($date3_default))
    , date('m', strtotime($date3_default))
    , date('Y', strtotime($date3_default)));
        $myCalendar->setPath("calendar/");
        $myCalendar->setYearInterval(2016, 2050);
        $myCalendar->setAlignment('left', 'bottom');
        $myCalendar->setDatePair('date3', 'date4', $date4_default);
        $myCalendar->writeScript();

        $myCalendar = new tc_calendar("date4", true, false);
        $myCalendar->setIcon("calendar/images/iconCalendar.gif");
        $myCalendar->setDate(date('d', strtotime($date4_default))
    , date('m', strtotime($date4_default))
    , date('Y', strtotime($date4_default)));
        $myCalendar->setPath("calendar/");
        $myCalendar->setYearInterval(2016, 2050);
        $myCalendar->setAlignment('left', 'bottom');
        $myCalendar->setDatePair('date3', 'date4', $date3_default);
        $myCalendar->writeScript();
        ?>
        </td>
        <td><INPUT TYPE="SUBMIT" VALUE="DELETE" NAME="DELETEDATA"></td><tr></tr><td colspan=5></td>
        <tr></tr><td colspan="8"><hr></hr></td><tr></tr>
        </form>
        </table></td><tr></tr>
</table>
</div></body></html>
```

Liite 19. PHP-käyttöliittymä: editwarningdata.php

```
<?php session_start();?>
<!doctype html>
<html lang="">
<head>
  <meta charset='utf-8'>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta http-equiv="refresh" content="<?php echo $refresh;?>" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">
  <script src="http://code.jquery.com/jquery-latest.mwin.js" type="text/javascript"></script>
  <script src="script.js"></script>
  <link href="calendar/css/default/calendar.css" rel="stylesheet" type="text/css" />
<?php
// Load the calendar class
require('calendar/tc_calendar.php');
?>
  <title>Arduino Sensor Measurer</title>
</head>
<body>
<?php include "menu.php"; ?>
<div id="dataa">
<table border=1 width="100%">
        <tr></tr>
        <td>
        <table border=0>
                <td><table><td>
                <tr></tr>
                <td style="font-size:20px" colspan="5"><strong>Edit Warning Data</strong></td>
                <tr></tr><td colspan="8">DELETE WARNING DATA</td>
                <tr></tr><td>Choose Sensor:</td><td>
                <FORM METHOD="POST" ACTION="editedwarningdata.php">
                <select NAME="ALLWARNINGDATA">
                <option value="#">#</option>
                <option value="allvalues">All Values</option>
<?php

        $stack = array();

        include 'settings/mysqldetails.php'; // user-password
                        try {
                                $dbh = new PDO('mysql:host=localhost;dbname=arduino', $user, $pass);
                                $stm = $dbh->prepare("SELECT sid FROM sensors");
                                $stm->execute();
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value) {
                                                array_push($stack,$value);

                                        }
                                }
                                $stm = $dbh->prepare("SELECT name FROM sensors");
                                $stm->execute();
                                $i = 0;
                                while($tsensors = $stm->fetch(PDO::FETCH_ASSOC)){
                                        foreach($tsensors as $row => $value) {
                                                if(count($stack)>$i){
                                                ?><?php echo "<option value='".$stack[$i]."'>".$value."</option>";?>
                                                <?php $i++;}
                                                }
                                }
                                $dbh = null;
```

```
                                        }catch (PDOException $e) {
                                        print "Error!: " . $e->getMessage() . "<br/>";
                                        die();
                                        }

?><td>Date:<input type="checkbox" name="CHECKDATE" value="date"></td>
                <td>
<?php
                 $date3_default = date("Y-m-d");
                 $date4_default = date("Y-m-d");

                 $myCalendar = new tc_calendar("date3", true, false);
                 $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                 $myCalendar->setDate(date('d', strtotime($date3_default))
        , date('m', strtotime($date3_default))
        , date('Y', strtotime($date3_default)));
                 $myCalendar->setPath("calendar/");
                 $myCalendar->setYearInterval(2016, 2050);
                 $myCalendar->setAlignment('left', 'bottom');
                 $myCalendar->setDatePair('date3', 'date4', $date4_default);
                 $myCalendar->writeScript();

                 $myCalendar = new tc_calendar("date4", true, false);
                 $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                 $myCalendar->setDate(date('d', strtotime($date4_default))
       , date('m', strtotime($date4_default))
       , date('Y', strtotime($date4_default)));
                 $myCalendar->setPath("calendar/");
                 $myCalendar->setYearInterval(2016, 2050);
                 $myCalendar->setAlignment('left', 'bottom');
                 $myCalendar->setDatePair('date3', 'date4', $date3_default);
                 $myCalendar->writeScript();
                 ?>
                </td>
                <td><INPUT TYPE="SUBMIT" VALUE="DELETE" NAME="DELETEDATA"></td><tr></tr><td colspan=5></td>
                <tr></tr><td colspan="8"><hr></hr></td><tr></tr>
                </form>
                </table></td><tr></tr>
</table></div></body></html>
```

Liite 20. PHP-käyttöliittymä: styles.css

```
#cssmenu,
#cssmenu ul,
#cssmenu li,
#cssmenu a {
  border: none;
  line-height: 1;
  margin: 0;
  padding: 0;
}
#cssmenu {
  height: 37px;
  display: block;
  border: 1px solid;
  border-radius: 5px;
  width: auto;
  border-color: #080808;
  margin: 0;
  padding: 0;
}
#cssmenu > ul {
  list-style: inside none;
  margin: 0;
  padding: 0;
}
#cssmenu > ul > li {
  list-style: inside none;
  float: left;
  display: inline-block;
  position: relative;
  margin: 0;
  padding: 0;
}
#cssmenu.align-center > ul {
  text-align: center;
}
#cssmenu.align-center > ul > li {
  float: none;
  margin-left: -3px;
}
#cssmenu.align-center ul ul {
  text-align: left;
}
#cssmenu.align-center > ul > li:first-child > a {
  border-radius: 0;
}
#cssmenu > ul > li > a {
  outline: none;
  display: block;
  position: relative;
  text-align: center;
  text-decoration: none;
  text-shadow: 1px 1px 0 rgba(0, 0, 0, 0.4);
  font-weight: 700;
  font-size: 13px;
  font-family: Arial, Helvetica, sans-serif;
  border-right: 1px solid #080808;
  color: #ffffff;
  padding: 12px 20px;
}
#cssmenu > ul > li:first-child > a {
```

```
border-radius: 5px 0 0 5px;
}
#cssmenu > ul > li > a:after {
  content: "";
  position: absolute;
  border-right: 1px solid;
  top: -1px;
  bottom: -1px;
  right: -2px;
  z-index: 99;
  border-color: #3c3c3c;
}
#cssmenu ul li.has-sub:hover > a:after {
  top: 0;
  bottom: 0;
}
#cssmenu > ul > li.has-sub > a:before {
  content: "";
  position: absolute;
  top: 18px;
  right: 6px;
  border: 5px solid transparent;
  border-top: 5px solid #ffffff;
}
#cssmenu > ul > li.has-sub:hover > a:before {
  top: 19px;
}
#cssmenu > ul > li.has-sub:hover > a {
  padding-bottom: 14px;
  z-index: 999;
  border-color: #3f3f3f;
}
#cssmenu ul li.has-sub:hover > ul,
#cssmenu ul li.has-sub:hover > div {
  display: block;
}
#cssmenu > ul > li.has-sub > a:hover,
#cssmenu > ul > li.has-sub:hover > a {
  background: #3f3f3f;
  border-color: #3f3f3f;
}
#cssmenu ul li > ul,
#cssmenu ul li > div {
  display: none;
  width: auto;
  position: absolute;
  top: 38px;
  background: #3f3f3f;
  border-radius: 0 0 5px 5px;
  z-index: 999;
  padding: 10px 0;
}
#cssmenu ul li > ul {
  width: 200px;
}
#cssmenu ul ul ul {
  position: absolute;
}
#cssmenu ul ul li:hover > ul {
  left: 100%;
  top: -10px;
  border-radius: 5px;
}
#cssmenu ul li > ul li {
```

```
  display: block;
```



```
list-style: inside none;
  position: relative;
  margin: 0;
  padding: 0;
}
#cssmenu ul li > ul li a {
  outline: none;
  display: block;
  position: relative;
  font: 10pt Arial, Helvetica, sans-serif;
  color: #ffffff;
  text-decoration: none;
  text-shadow: 1px 1px 0 rgba(0, 0, 0, 0.5);
  margin: 0;
  padding: 8px 20px;
}
#cssmenu,
#cssmenu ul ul > li:hover > a,
#cssmenu ul ul li a:hover {
  background: #3c3c3c;
  background: -moz-linear-gradient(top, #3c3c3c 0%, #222222 100%);
  background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #3c3c3c), color-stop(100%, #222222));
  background: -webkit-linear-gradient(top, #3c3c3c 0%, #222222 100%);
  background: -o-linear-gradient(top, #3c3c3c 0%, #222222 100%);
  background: -ms-linear-gradient(top, #3c3c3c 0%, #222222 100%);
  background: linear-gradient(top, #3c3c3c 0%, #222222 100%);
}
#cssmenu > ul > li > a:hover {
  background: #080808;
  color: #ffffff;
}
#cssmenu ul ul a:hover {
  color: #ffffff;
}
#cssmenu > ul > li.has-sub > a:hover:before {
  border-top: 5px solid #ffffff;
}
table.data td {
    padding-top: 1px;
    padding-right: 15px;
    padding-bottom: 1px;
    padding-left: 15px;
}
table.sensors td {
                border-px: 1px;
    padding-top: 1px;
    padding-right: 10px;
    padding-bottom: 1px;
    padding-left: 10px;
}
table.sensors td.sensorcolumns {
                font-weight: bold;
                font-size: 20px;
}
td.warningstitle{
                font-weight: bold;
                font-size: 20px;
}
td.infotitle{
                font-weight: bold;
                font-size: 20px;
    padding-left: 10px;
    padding-top: 10px;
```

```
}
```

Liite 21. PHP-käyttöliittymä: /settings/mysqldetails.php

```
<?php
        $user='root'; // user name
        $pass='test';  // password
?>
```

Liite 22. Arduinoon kirjaston lisääminen

Mitä ovat arduinon kirjastot?

- Arduinon kirjastot ovat yleensä c++:lla kirjoitettuja apuohjelmia, joiden avulla pystyy käyttämään monipuolisesti erilaisia toimintoja, antureita tai muita laitteita.
- Kirjaston lisääminen arduinoon:
    - Kirjaston voi lisätä arduinoon (uudemmissa arduino-ohjelmisto versioissa) ainakin kolmella eri tavalla:
        - I.) Tallentamalla sen oikeaan arduinon libraries-kansioon (usein löytyy käyttäjän documents/arduino-kansiosta),
        - II.) Sketch-Include library-Add Zip-library.
        - III.) Sketch-Include library-Manage libraries (voit etsiä arduinon omasta palvelusta sopivia kirjastoja arduinoon)