

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Henri Vuorela

Opinnäytetyö

ZigBee-sensoriverkon käyttöliittymä

Työn ohjaaja
Työn tilaaja
Tampere 4/2010

Lehtori Jari Mikkolainen
Tampereen ammattikorkeakoulu

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma, Ohjelmistotekniikka

Tekijä: Henri Vuorela
Työn nimi: ZigBee-sensoriverkon käyttöliittymä
Sivumäärä: 30 sivua
Valmistumisaika: 5/2010
Työn ohjaaja: Lehtori Jari Mikkolainen
Työn tilaaja: Tampereen ammattikorkeakoulu

TIIVISTELMÄ

Tässä työssä tutustuttiin ZigBee-sensoriverkkotekniikkaan, tiedonkeruuseen ja käyttöliittymän toteuttamiseen. Työn tarkoitus oli tutustua MeshWorks Wireless Oy:n ja TappIT Oy:n Tampereen ammattikorkeakoululle tarjoamaan sensoriverkkoon ja sen laitteisiin.

Työ on tehty Tampereen ammattikorkeakoululle, projektityönä rakennetun sensoriverkon pohjalta. Työn tavoitteena oli saada tutkimustietoa verkosta ja sen käyttömahdollisuuksista. Sensoriverkon käyttömahdollisuuksien tutustumismenetelmäksi on valittu käyttöliittymän rakentaminen sensoriverkon palvelinrajanpinnalle.

Työn tuloksena on saatu aikaan mittaustietoja hyödyntävä käyttöliittymä, jossa tietoja näytetään usealla eri tavalla. Työn suunnittelu- ja toteutusvaiheiden inspiroimana on syntynyt muita projekteja, kuten mobiilikäyttöliittymä. Työn tuloksia ja sen mukana kertynyttä osaamista voidaan hyödyntää sensoriverkon jatkokehityksessä Tampereen ammattikorkeakoulussa.

TAMK University of Applied Sciences
Computer Science, Software Engineering

Writer: Henri Vuorela
Thesis: ZigBee sensor network user interface
Pages: 30
Graduation time: May 2010
Thesis Supervisor: Lecturer Jari Mikkolainen
Co-operating company: TAMK University of Applied Sciences

ABSTRACT

The purpose of this thesis was to develop user interface for wireless ZigBee sensor network. Interface is build to be one of the use cases for the network's sensor data. The thesis was commissioned by TAMK University of Applied Sciences.

The objective of this thesis was to get know the basics and the possibilities for this technique and offer one kind of way to view sensor network's data. Interface was build on TappIT Oy's provided server interface.

Keywords

ZigBee, sensor network, user interface

Sisällysluettelo

1 JOHDANTO	1
2 SENSORIVERKKO	2
2.1 ZIGBEE	2
2.2 MESHWORKS WIRELESS OY JA TAPPIT OY	3
2.3 SENSORIVERKON LAITTEET	4
2.3.1 Gateway	4
2.3.2 THB-anturi	5
2.3.3 THCLR-anturi	6
2.4 KÄYTETYT TEKNIIKAT	7
2.4.1 JavaScript	7
2.4.2 Anychart	7
2.4.3 XML	8
3 KÄYTTÖLIITTYMÄN SUUNNITTELU	10
3.1 TARPEIDEN KARTOITUS	10
3.2 KÄYTETTÄVYYS JA JOUSTAVUUS	10
4 KÄYTTÖLIITTYMÄN TOTEUTUS	12
4.1 PALVELIMEN TIEDONKERUU	12
4.1.1 Collect Template	13
4.1.2 Data Entity	13
4.1.3 Report Template	14
4.1.4 Report Entity	14
4.2 XML-KYSELY	15
4.2.1 MAP_BASE_XML	15
4.2.2 MAP_ROUTE_XML	16
4.2.3 FLASH_2D_XML	16
4.3 KÄYTTÖLIITTYMÄN RAKENNE	18
4.3.1 Mittaustietojen rakenne	18
4.3.2 Mittaustietojen tallentaminen	19
4.3.3 Mittaustietojen tulostaminen	20
5 KÄYTTÖLIITTYMÄN KÄYTTÖ	21
5.1 KIRJAUTUMINEN	21
5.2 PÄÄSIVU	21
5.3 JULKINEN NÄKYMÄ	22
5.4 LAITELISTA	23
5.5 LISÄTIEDOT	24
6 VERKON HYÖTYKÄYTTÖ TAMKISSA	25
6.1 GPS-PAIKANNUS	25
6.2 MOBIILISOVELLUS	26
6.3 HYÖTYKÄYTTÖ OPETUKSESSA	28
7. YHTEENVETO	29
LÄHTEET	30

LYHENTEET JA TERMISTÖ

ZigBee	IEEE 802.15.4 -standardin mukainen lyhyen kantaman tietoliikenneverkko (langaton likiverkko WPAN).
JavaScript	Web-ympäristössä käytettävä komentosarjakieli.
Anychart	Anychart on työkalu, jonka avulla voi helposti lisätä interaktiivisia animoituja flash-komponentteja Internet- ja työpöytäsovelluksiin.
Flash	Adobe Systemsin tuottama kehitysympäristö, jonka avulla voidaan luoda multimediaesityksiä esimerkiksi verkkosivustoille.
XML	Extensible Markup Language. Kuvauskieli tiedonvälitykseen järjestelmien välillä.
Gateway	Gateway on laite joka luo langattoman likiverkon, johon sen kantama-alueella sijaitsevat anturit ottavat yhteyttä.
THB-anturi	THB-anturi sisältää nimensä mukaan kolme sensoria, jotka mittaavat lämpötilaa, kosteutta ja patteriarvoa.
THCLR-anturi	Anturi, jolla tiedon välittämisen lisäksi voi mitata joko valinnaisesti lämpötilaa ja ilmankosteutta tai hiilidioksidipitoisuutta.

1 JOHDANTO

ZigBee on monilla eri aloilla yleistynyt lyhyen kantaman tietoliikenneverkko. ZigBee-verkkoja on kehitetty jo vuodesta 1998. Tekniikka antaa mahdollisuuden luoda pienitehoisia, suuria määriä laitteita sisältäviä, nopeita verkkoja.

MeshWorks Wireless Oy ja TappIT Oy lähtivät yhteistyössä suunnittelemaan langatonta tiedonkeruujärjestelmää vuonna 2008. Ominaisuuksien luomisessa ja teknisessä toteutuksessa käytettiin ZigBee-tietoliikenneverkkoa.

Tämän työn tarkoituksena on perehtyä sensoriverkon tekniikkaan ja toteuttaa käyttöliittymä MeshWorks Wireless Oy:n ja TappIT Oy:n tarjoamalle sensoriverkolle.

Käyttöliittymä toimii näkymänä Tampereen ammattikorkeakoulun tiloihin sijoitetuista laitteista kerätyille mittaustiedoille.

Työssä käytetyt eri tyyppiset laitteet mittaavat pääasiassa lämpötilaa, kosteutta ja hiilidioksidia. Käyttöliittymä toimii yhtenä esimerkkinä MeshWorks Wireless Oy:n langattomien tiedonkeruulaitteiden käyttömahdollisuuksista.

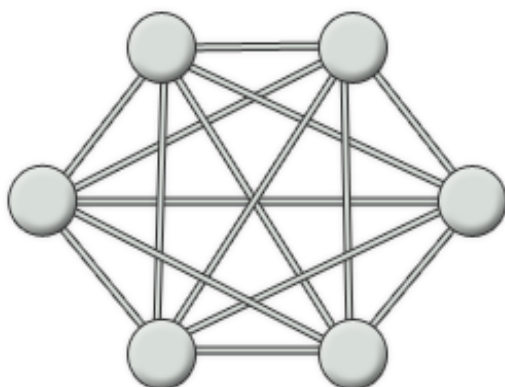
Käyttöliittymä on tehty TappIT Oy:n hallinnoimalle, Tampereen ammattikorkeakoululle konfiguroidulle palvelimelle. Työssä käytetään ohjelmointitekniikoina pääasiassa JavaScriptiä, HTML:ää ja CSS:ää.

Opinnäytetyö sisältää paljon tutkimustyötä ja testausta MeshWorks Wireless Oy:n ZigBee-laitteilla ja palvelinrajapinnan konfigurointia. Työ painottuu suurimmalta osin käyttöliittymän eri komponenttien rakentamiseen, palvelimen tiedonkeruun ohjaukseen ja erilaisiin tekniikoihin perehtymiseen.

Lopussa käydään läpi järjestelmän käyttömahdollisuuksia ja esitetään muutama Tampereen ammattikorkeakoulussa toteutunut projekti, jossa on hyödynnetty tämän sensoriverkon laitteita.

2 SENSORIVERKKO

Tässä työssä käytetty sensoriverkko on sijoitettu Tampereen ammattikorkeakoulun A-siipeen, kolmeen eri kerrokseen. Siiven jokaisessa kerroksessa on tukiasema (Gateway), johon sen kantama-alueella olevat laitteet ottavat yhteyttä ja lähettävät mittaustietoa. Verkon langaton viestintä tapahtuu ZigBee-tekniikan avulla. Verkon topologiana toimii langattomissa verkoissa yleisesti käytetty mesh-topologia, jossa kaikki asemat ovat suorassa yhteydessä toisiinsa (kuvio 1). Tukiasema lähettää kerätyt mittaustiedot TappIT Oy:n hallinnoimalle palvelimelle, johon tässä työssä toteutettu käyttöliittymä on tehty.



Kuvio 1: Mesh-topologia

2.1 ZigBee

ZigBee on IEEE 802.15.4-standardin mukainen lyhyen kantaman tietoliikenneverkko (langaton likiverkko WPAN). ZigBeen tapaisia verkkoja on suunniteltu jo vuodesta 1998, jolloin insinöörit arvioivat muiden vastaavien verkkojen kuten WiFin ja Bluetoothin olevan sopimattomia kaikkiin käyttötarkoituksiin. IEEE 802.15.4 -standardi valmistui toukokuussa 2003. (*Wikipedia: ZigBee. Viitattu 17.2.2010.*)

ZigBee-standardin kehittämisestä vastaa ZigBee-allianssi. ZigBee on avoin kaikille yrityksille, jotka haluavat olla osallisina ja myötävaikuttaa ZigBee-teknologian kehitykseen. Allianssiin kuuluu monia suuria yrityksiä, kuten Intel, HP ja Phillips. Allianssin tekemä standardin määrittäminen valmistui vuonna 2004. Allianssilla on jäsenmaksu, jonka maksamalla saa oikeuden käyttää allianssin standardia kaupallisiin tarkoituksiin. ZigBee-tunnuksen käyttäminen edellyttää laitemallin testauttamista ZigBee-yhteensopivaksi. (*Wikipedia: ZigBee. Viitattu 17.2.2010.*)

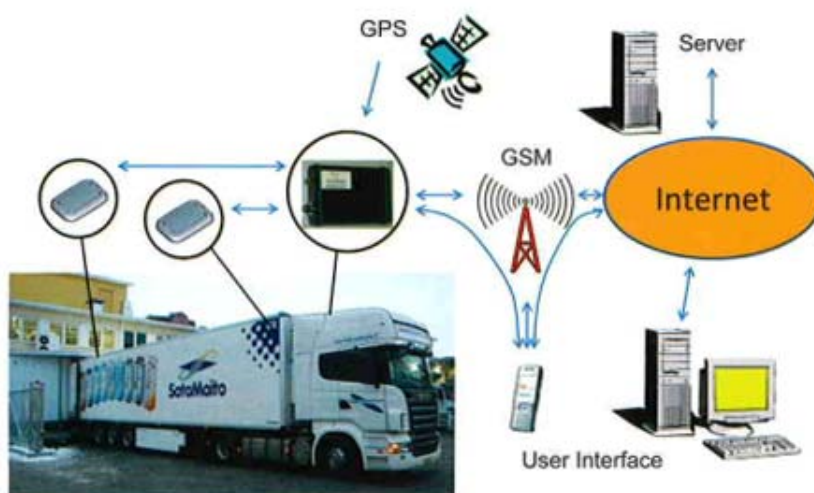
ZigBee-tekniikan suurimpia käyttötarkoituksia on lyhyen matkan viestintä ja tiedonkeruu eri teollisuuden aloilla. Tekniikalla on kuitenkin paljon sovellusmahdollisuuksia. ZigBeen käyttötarkoituksena ei ole haastaa Bluetoothia, vaan tarkoituksena on saavuttaa laitteille äärimmäisen vähäinen virrankulutus, jolloin paristonkesto voi olla jopa useita vuosia. (*Wikipedia: ZigBee. Viitattu 17.2.2010.*)

Standardi määrittelee kolme eri tietoturvasoaa. Taso 1 ei tarjoa minkäänlaista tietoturvaa, taso 2 käyttää hyväkseen pääsilystoja (ACL) ja tasossa kolme laite käyttää symmetristä salausta siirrettävän tiedon salaamiseen. Verkon tietoturvaa voidaan myös parantaa sovellustasolla. Tietoturvan monitasoisuus mahdollistaa monia erilaisia toteutuksia. (*Wikipedia: ZigBee. Viitattu 17.2.2010.*)

2.2 MeshWorks Wireless Oy ja TappIT Oy

MeshWorks Wireless Oy ja TappIT Oy lähtivät yhteistyössä suunnittelemaan langatonta tiedonkeruujärjestelmää vuonna 2008. Järjestelmän oli tarkoitus seurata lämpötilaa sekä sijaintia reaaliajassa. Mittausinformaatiota hyödyntämällä pystyttiin aktivoimaan sen käyttö erityyppisissä automaattisissa toiminnoissa, kuten paikkahälytyksissä, raja-arvojen ylityksissä ja esimerkiksi aikataulupoikkeamissa. Ajasta tai paikasta riippumattomasti järjestelmää kehitettiin käyttäjien toimintaa ja hyötyä ajatellen.

Ominaisuuksien luomisessa ja teknisessä toteutuksessa hyödynnettiin mm. langatonta lyhyen kantaman tietoliikenneverkkoa (ZigBee). Tekniikkaa voidaan hyödyntää esimerkiksi kiinnittämällä ajoneuvon lastitilaan langattomat anturit. Niiden läheisyyteen on asennettu tukiasema, joka kerää ja lähettää anturien mittaaman tiedon sekä kuljetusyksikön paikkatiedon palvelimelle (kuvio 2). Selainpohjaisilla sovelluksilla loppukäyttäjälle voi reaaliaikaisesti näyttää esimerkiksi kuljetuksessa tapahtuvia lämmön vaihteluita ja mittaustulosten historiatietoja.



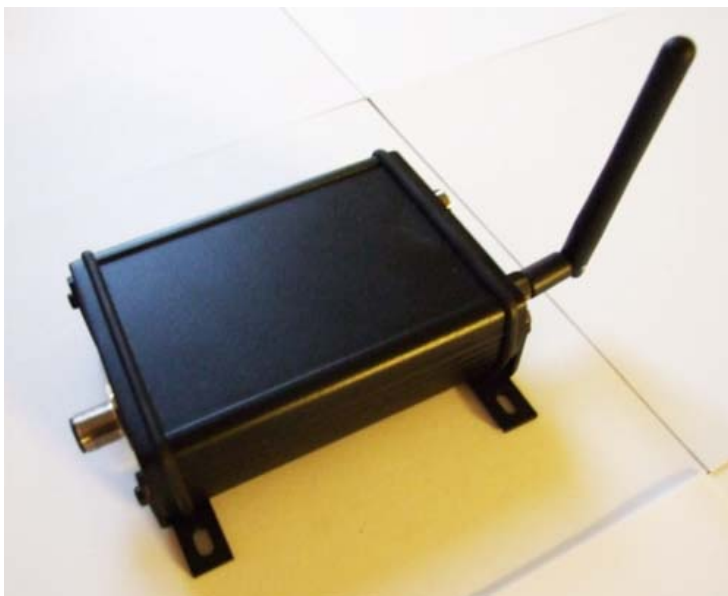
Kuvio 2: Langattoman tiedonkeruualustan toimintaperiaate

2.3 Sensoriverkon laitteet

Työssä käytetyt langattomat anturit keräävät sensoritietoa, joka lähetetään ZigBee-verkkoa hyödyntäen tukiasemalle. Tämä tieto voidaan lähettää tukiasemalle joko suoraan tai reitittimenä toimivan anturin kautta. Tukiasema kerää anturien lähettämän tiedon ja lähettää sen TappIT Oy:n ylläpitämälle palvelimelle käyttäen laitteessa olevaa GPRS-yhteyttä.

2.3.1 Gateway

Gateway on laite, joka luo langattoman likiverkon, johon sen kantama-alueella sijaitsevat anturit ottavat yhteyttä (kuvio 3). Gateway-laitteessa on myös GPRS-yhteys, jolla tieto välitetään eteenpäin palvelimelle. GPS:n avulla voidaan haluttaessa paikantaa esimerkiksi ajoneuvoihin asennettuja verkon tukiasemia.



Kuvio 3: Gateway

2.3.2 THB-anturi

THB-anturi sisältää nimensä mukaan kolme sensoria, jotka mittaavat lämpötilaa, kosteutta ja patteriarvoa (kuvio 4). Tampereen ammattikorkeakoulun käyttämät THB-anturit ovat paristokäyttöisiä ja toimivat yhdellä tai kahdella kolmen voltin paristolla. Anturi yhdistää langattomasti tukiasemaan, kun se on tukiaseman kantama-alueella. Anturi on asetettu mittaamaan sensoritietoa viiden minuutin välein ja lähettämään tiedon tukiasemalle. Anturin ollessa verkon kantama-alueen ulkopuolella se kerää mittaustietoja muistiin. Seuraavan kerran yhdistyessään mihin tahansa tukiasemaan, se lähettää muistissa olevat sensorien mittaustiedot tukiasemalle.



Kuvio 4: THB-anturi

2.3.3 THCLR-anturi

THCLR-anturi sisältää samat sensorit kun THB-anturi, mutta sisältää lisäksi hiilidioksidipitoisuutta mittaavan sensorin (kuvio 5). Anturi toimii verkkovirralla ja se voi mitata valinnaisesti joko lämpötilaa ja ilmankosteutta tai hiilidioksidipitoisuutta, koska hiilidioksidisensori lämmittää ja kuivattaa ilmaa sen verran että se vääristäisi lämpötila- ja kosteussensoreiden mittaamia arvoja. THCLR-toimii myös tiedon välittäjänä muiden antureiden ollessa liian kaukana tukiasemasta. THCLR-anturin toimiessa tukiasemana, se vastaanottaa antureiden lähettämää sensoritietoa ja välittää sen eteenpäin oman yhteydenoton aikana Gateway-tukiasemalle.



Kuvio 5: THCLR-anturi

2.4 Käytetyt tekniikat

Työssä tehty käyttöliittymä on toteutettu TappIT Oy:n hallinnoimalle palvelimelle. Palvelin on valittu käyttöliittymän alustaksi sen vuoksi, että tieto on helposti kyseltävissä palvelimen XML-rajapinnan avulla, eikä omaa tietoturvaa käyttöliittymän tiedoille tarvitse tällöin rakentaa. Lisäksi palvelimelle on mahdollista toteuttaa Anychart-työkalun avulla flash-komponentteja.

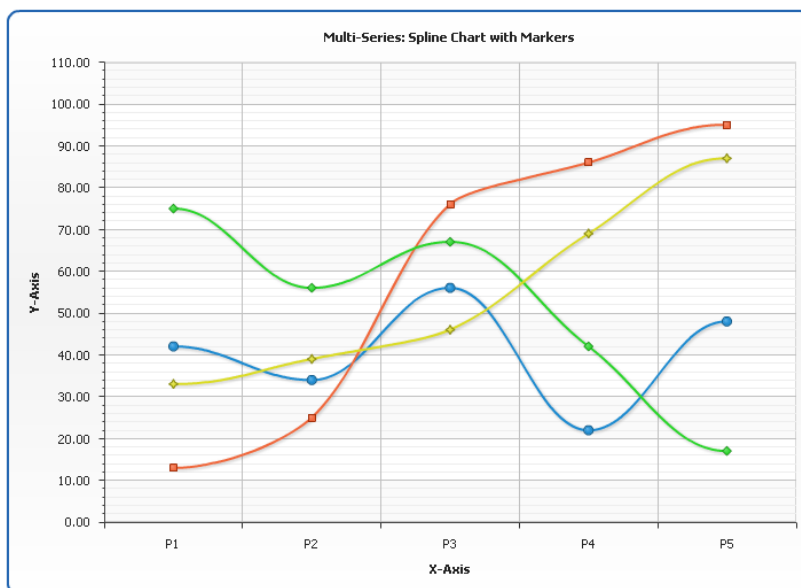
2.4.1 JavaScript

JavaScript on alkujaan Netscape Communications Corporationin kehittämä Web-ympäristössä käytettävä komentosarjakieli. JavaScriptin tärkein ominaisuus on mahdollisuus lisätä Web-sivuille dynaamista toiminnallisuutta. Se on yksinkertainen ja suoraan html-koodin sekaan kirjoitettava skriptikieli. Se mahdollistaa interaktiivisten toimintojen lisäämisen www-sivuille ilman, että sivu on jatkuvassa yhteydessä palvelimeen.

(Wikipedia: JavaScript. Viitattu 14.3.2010)

2.4.2 Anychart

Anychart on työkalu, jonka avulla voi helposti lisätä interaktiivisia animoituja flash-komponentteja Internet- ja työpöytäsovelluksiin (kuvio 6). Työkalun avulla voi luoda kaavioita ja näyttää niitä esimerkiksi Internet-selaimessa ilman, että palvelimelle täytyisi asentaa komponentteja. Anychart on alustariippumaton. Sillä voi luoda visuaalisen ilmeen erilaisille raporteille ja tilastoille.



Kuvio 6: Anychart

2.4.3 XML

XML-kieltä käytetään kuvaamaan ja järjestämään tietoa. XML on lyhenne sanoista ”Ex-tensible Markup Language” ja sen kehittäjä on kansainvälinen yritysten ja yhteisöjen yhteenliittymä World Wide Web Consortium. XML-kieltä käytetään formaattina tiedonvälitykseen järjestelmien välillä ja tiedon tallentamiseen. XML-kieli on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomassoja selkeästi. (*Wikipedia: XML. Viitattu 14.3.2010.*)

XML-kielen tulkitseminen eli parsinta on helppoa. Sitä varten on tehty valmiita kirjastoja ja sovelluksia. Oman sovelluksen ei tarvitse koskaan nähdä XML-tietoa sellaisena kuin se on tallennettu, vaan se voidaan välittömästi antaa parsittavaksi. Tällöin tulkitseminen tehdään ohjelmoijan puolesta. Parserit eli sovellukset, jotka tulkitsevat XML-tietoa, toimivat niin, että ne lukevat XML-tagien välistä tietoa. Parseri tekee tehtävänsä riippumatta siitä, mitä tietoa tagit sisältävät. Se palauttaa arvon sellaisenaan ja ohjelmoijan tehtäväksi jää tiedon tyyppin valinta. XML:n parsimiseen yleisimmät tavat ovat DOM ja SAX. (*Wikipedia: XML. Viitattu 14.3.2010.*)

DOM (Document Object Model) on ohjelmointirajapinta, joka mahdollistaa XML-dokumenttien sisällön lukemisen ja muokkaamisen. Yhdessä Javascriptin kanssa sillä voidaan toteuttaa vuorovaikutteisia www-sivuja, jotka eivät vaadi jatkuvaa palvelinyh-

teyttä. DOM-parserointitavassa tiedostosta rakennetaan puumainen kuvaus muistiin. (*Wikipedia: XML. Viitattu 14.3.2010.*)

Ylimpänä elementtinä puussa on dokumentti-elementti. Tällä elementillä voi olla yksi tai useampi lapsielementti, jolla voi olla myös lapsielementtejä. Elementeillä voi olla myös tekstiosia. DOM-parserointi on käytännöllinen silloin kun tiedetään, että XML-tiedosto on pieni, koska DOM-parseroinnissa otetaan koko XML-tiedosto muistiin käsittelyn ajaksi. Suurilla XML-tiedostoilla parserointi olisi tehotonta. (*Wikipedia: XML. Viitattu 14.3.2010.*)

3 Käyttöliittymän suunnittelu

Käyttöliittymän suunnittelu lähti liikkeelle sovelluksen tarpeiden ja käyttötarkoitusten määrittelystä ja suunnittelusta sekä langattomiin ZigBee-laitteisiin ja niiden käyttömahdollisuuksiin tutustumisesta. Laitteet mittaavat lämpötilaa, kosteutta, patteriarvoa ja hiilidioksidia. Näitä tietoja on tarkoitus pystyä helposti ja vaivattomasti selaamaan palvelin rajapinnan käyttöliittymän avulla.

3.1 Tarpeiden kartoitus

Käyttöliittymän tarkoituksena on toimia yleisenä näkymänä MeshWorksin langattomien tiedonkeruulaitteiden tiedoille. Käyttöliittymästä on mahdollista selata Tampereen ammattikorkeakoulun tiloihin sijoitettujen antureiden mittaamia arvoja reaaliajassa sekä mittaustietojen historiaa.

Käyttäjällä ei tarvitse olla minkäänlaista kokemusta langattomasta tiedonkeruujärjestelmästä tai siinä käytetyistä tekniikoista. Käyttöliittymä on tarkoitus toteuttaa helposti omaksuttavaksi ja yksinkertaiseksi käyttää.

Käyttöliittymään on tehty mahdollisuus selata listaa kaikista koululle sijoitetuista laitteista ja niiden mittausarvoista ja historiatiedoista. Lisäksi sovelluksen julkinen näkymä pyörii itsekseen näyttäen käyttäjän valitseman tilan historiatietoja ja lämpötilan ja kosteuden mittausarvoja visuaalisina kuvina ja kaavioina.

Rajapinnan, sen hyödyntämisen tekniikan ja laitteiden käyttömahdollisuudet ovat monipuoliset, joten tämän työn tuloksena tehty käyttöliittymä on vain yksi tapa hyödyntää tekniikkaa.

3.2 Käytettävyys ja joustavuus

Käyttöliittymä suunnitellaan käytettävyydeltään helpoksi ja joustavaksi. Käyttäjän ei tarvitse tuntea sensoriverkon hyödyntämää tekniikkaa. Sovellus toimii selaimessa, joten se on myös alustariippumaton.

Sovellus listaa langattomat laitteet ilman, että se rajoittaa niiden määrää tai tyyppiä.

Julkinen näkymä on tehty visuaaliseksi itsestään tietoja päivittäväksi sovellukseksi. Näkymän taustalla sovellus hakee aina uusimmat mittaustiedot käyttäjän nähtäväksi ja samalla päivittää edelliset mittauksen historianäkymän kärkeen.

4 Käyttöliittymän toteutus

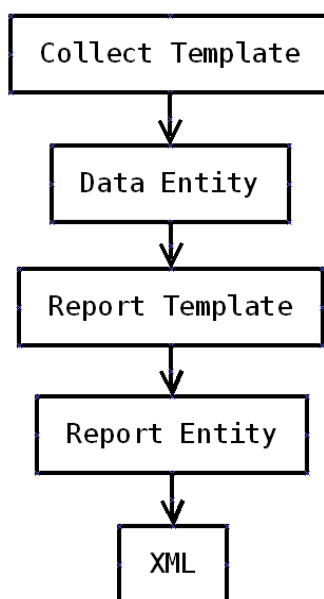
Käyttöliittymä on toteutettu TappIT Oy:n TAMK:lle tarjoamalle palvelinrajapinnalle. Sovellus toimii pääasiassa palvelimelle tallennettujen JavaScript-ohjelmakoodien avulla. Visuaalisessa ulkoasussa on käytetty Adobe Photoshop CS3 -sovellusta.

Käyttöliittymän ulkoasusta muodostui monia erilaisia versioita. Ulkoasun visuaalinen ilme muodostui lopulta mahdollisimman selkeäksi ja yksikertaiseksi omaksua. Käyttöliittymän ulkoasun kuvat ja tyylitiedosto sijaitsevat toisella palvelimella, koska tarjotulle palvelinrajapinnalle ei ollut vielä FTP-oikeuksia.

Sovellus saa käyttöönsä langattomien laitteiden mittaustiedot palvelimella suoritettavalla XML-kyselyllä. XML-kyselyitä on mahdollista tehdä reaaliaikaisista mittaustiedoista sekä historiatiedoista. Reaaliaikaiset mittaustiedot ovat anturien viimeisimpänä lähettämiä tietoja. Historiakyselyn voi tehdä palvelimelta esimerkiksi haluamaltaan aikaväliltä.

4.1 Palvelimen tiedonkeruu

Sovellus saa esitettävät tiedot käyttöönsä XML-kyselyiden avulla. XML-kyselyt määntyvät TappIT Oy:n hallinnoimalla palvelimella muodostettujen raporttien mukaan (kuvio 7). Tässä osiossa käsitellään sensoritiedon muodostumista XML-kyselyksi.



Kuvio 7: Palvelimen tiedonkeruun rakenne

4.1.1 Collect Template

Collect Templaten avulla määritellään, mitä tietoa halutaan kyselyssä esittää ja jokaiselle tiedolle määritellään mitattavan tyyppin mukaan erilaisia ryhmiä. TAMKin laitteille on luotu Collect Template, joka sisältää kaikkien laitteiden kerättävät mittausarvot. Alla olevassa kuviossa (kuvio 8) on määritelty yhdelle laitteelle kerättäviksi tiedoiksi seuraavat: patteriarvo, ilmankosteus, nimi, sekvenssi, lämpötila ja signaalin voimakkuus. Rivit on nimetty numeroilla, jotta niitä voidaan helposti selata lajittelemalla ne nimen mukaiseen järjestykseen. XML-kyselyn kannalta merkittävin kenttä on ”Xml-title”, jossa määritellään XML-kyselyssä näkyvän tiedon kentän nimi.

Collect Template Items Detail													
<input type="button" value="Delete"/> <input type="button" value="Apply Changes"/> <input type="button" value="Add Row"/>													
<input type="checkbox"/>	Name ▲	Title	Short Title	Xml Title	If Filed Name	Ed Item Type Id	Remarks	Item Value Type	Fol	Mon	Act	Alert	Sync
<input type="checkbox"/>	01_BATTERY	Battery 01	Ba01	batt1	Battery01	BATTERY		NUMBER1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	01_HUMIDITY	Humidity 01	Hu01	humidity1	Humidity01	HUMIDITY		NUMBER1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	01_NAME	Name 01	N1	name1	Name01			STRING	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	01_SEQ	Seq 01	Se01	seq1	Seq01			SEQ	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	01_TEMP	Temp 01	Te01	temp1	Temp01	TEMP		NUMBER1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	01_XRR	Xrr 01	Xrr01	xrr1	Xrr01			NUMBER1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Kuvio 8: Collect Template

4.1.2 Data Entity

Kun tietojen tyypit on määritelty, tulee laitteet yhdistää Data Entityyn, jossa käytettävät anturit liitetään kyselyyn ja anturien sensoritiedot yhdistetään oikeille riveille. Kuviossa 9 määritellään esimerkiksi ”01_BATTERY”-rivin sensoriksi laitteen ”0050c2ffffeacdaf0” sensori ”xba”. Tämä tarkoittaa sitä, että laitteen patteritieto kohdistuu ensimmäiselle riville.

Data Entityssä on myös nimetty laitteet kenttään ”01_NAME”, jotta ne olisi helpompi tunnistaa käyttäjän sovelluksen listatessa antureita. Kun laitteet oli sijoitettu tiloihin, ne nimettiin tilan mukaan. Nimen perään lisättiin myös laitteen MAC-osoitteen loppuosa (esimerkiksi [daf0]).

COLLECT TEMPLATE DATA											Submit
Name ▲	Title	Value	Number Value	Monitor	Mon	Act	Al	Sync	Sensor Field	Sensor	
01_BATTERY	Battery 01	6.2	6.2	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0050c2ffeacdaf0 - xba	0050c2ffeacdaf0	
01_HUMIDITY	Humidity 01	24.2	24.2	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0050c2ffeacdaf0 - xhu	0050c2ffeacdaf0	
01_NAME	Name 01	A1-18 [daf0]		0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
01_SEQ	Seq 01	3560	3560	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0050c2ffeacdaf0 - xsq	0050c2ffeacdaf0	
01_TEMP	Temp 01	20.8	20.8	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0050c2ffeacdaf0 - xte	0050c2ffeacdaf0	
01_XRR	Xrr 01	-9.5	-9.5	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0050c2ffeacdaf1 - xrr	0050c2ffeacdaf1	

Kuvio 9: Data Entity

4.1.3 Report Template

Kun tieto on yhdistetty oikeille riveille, yhdistetään luodut tietokentät raportin pohjaan (Report Template). Tässä vaiheessa valitaan, mitä tietoja raportissa halutaan näytettävän (kuvio 10). Haluttaessa voidaan osa tiedoista jättää raportin ulkopuolelle tai nimetä XML-kyselyn kenttiä uudelleen.

Report Template Items							Delete	Apply
<input type="checkbox"/>	Name ▲	Xml Status	Xml Analyze	Col Xml Title	Col name	Item Type		
<input type="checkbox"/>	01_BATTERY	batt1	batt1	batt1	01_BATTERY	BATTERY		
<input type="checkbox"/>	01_HUMIDITY	humidity1	humidity1	humidity1	01_HUMIDITY	HUMIDITY		
<input type="checkbox"/>	01_NAME	name1	name1	name1	01_NAME			
<input type="checkbox"/>	01_SEQ	seq1	seq1	seq1	01_SEQ			
<input type="checkbox"/>	01_TEMP	temp1	temp1	temp1	01_TEMP	TEMP		
<input type="checkbox"/>	01_XRR	xrr1	xrr1	xrr1	01_XRR			
<input type="checkbox"/>	02_BATTERY	batt2	batt2	batt2	02_BATTERY	BATTERY		

Kuvio 10: Report Template

4.1.4 Report Entity

Lopuksi luodaan Report Entity, johon yhdistetään Report Templatessa luotu tietorakenne (kuvio 11). Tätä raporttia voidaan kysyä XML-muodossa Data Entityn nimen avulla. Tämä raportti on nimetty nimellä ”TAMK_ALL”.

Report Entity - Collect entities (TAMK_ALL) (1389) (322)						
Name	Id	Rep.dat. ent id	Rep.dat. ent Serial No	Data Entity id	Data Entity name	
<input checked="" type="checkbox"/> TAMK_Sensor_allinfo	401	01250	NO_SERIAL	1187	TAMK_ALL	1 - 1

Kuvio 11: Report Entity

4.2 XML-kysely

Tietoa voidaan hakea palvelimelta XML-kyselyiden avulla. Näiden perusteella sovellus voi pyytää viimeisimpiä mittaus- ja historiatietoja tai molempia monessa eri muodossa. Tässä työssä on käytössä kolme eri XML-kyselyä: MAP_BASE_XML, MAP_ROUTE_XML ja FLASH_2D_XML. Kyselytyypin mukaan pakollisia tietoja ovat vähintään ”session_id” ja ”application_id”, jotka sovellus saa käyttöönsä palvelimen kautta. ”session_id” on palvelimen automaattisesti generoima numerosarja, jonka avulla voidaan lisätä palvelun tietoturva. Kun ”session_id” on saatu käyttöön, se on voimassa tietyn määräjän tai kunnes käyttäjä kirjautuu ulos järjestelmästä.

4.2.1 MAP_BASE_XML

MAP_BASE_XML luo XML-tiedoston antureiden sensorien viimeisimmistä mittaustuloksista. Tietoa voidaan käyttää esimerkiksi ajoneuvon sen hetkisen sijainnin tai lämpötilan viimeisimmän tiedon selvittämiseksi.

Kysely hakee luodun Report Entityn nimen (TAMK_ALL) perusteella kaikkien raportista löytyvien anturien viimeisimmät mittaustulokset. Työssä kyselyä käytetään viimeisimpien mittaustulosten hakemiseen (esimerkki 1).

Kysely tapahtuu osoitteella:

```
http://www.../MAP_BASE_XML?session_id=1136196346114499&rep_name=TAMK_ALL&application_id=200
```

Vastaus tulee muodossa:

```
<markers>
<marker id="1389" XMLtype="322" rep_name="TAMK_ALL" rep_levent=""
rep_type="TAMK_A" rep_page="JS_REPORT" remarks="" batt1="6.2" seq1="3560"
humidity1="23.2" temp2="21.5" name2="A3-14 [daf1]" .../>
</markers>
```

Esimerkki 1: MAP_BASE_XML-kysely

4.2.2 MAP_ROUTE_XML

MAP_ROUTE_XML luo XML-tiedoston antureiden sensorien mittaustulosten historia-tiedoista. Tietoa voidaan käyttää esimerkiksi erilaisten kaavioiden piirtämiseen tai ajo-neuvon kulkeman reitin selvittämiseen.

Kysely hakee Report Entityn id:n (1389) perusteella kaikkien raportista löytyvien anturien historiatietoja. Kyselyä voidaan rajata esimerkiksi päivämäärän ja merkkien perusteella. Työssä kyselyä käytetään viimeisimpien mittaustuloksen ja niiden aikaleimojen selvittämiseen (esimerkki 2). Tällä voidaan selvittää, onko anturi aktiivisessa tilassa, eli onko se lähettänyt tietoa lähiaikana.

Kysely tapahtuu osoitteella:

```
http://www.../MAP_ROUTE_XML?session_id=1136196346114499&application_id=200&p_rep_ent_id=1389&p_markers=100
```

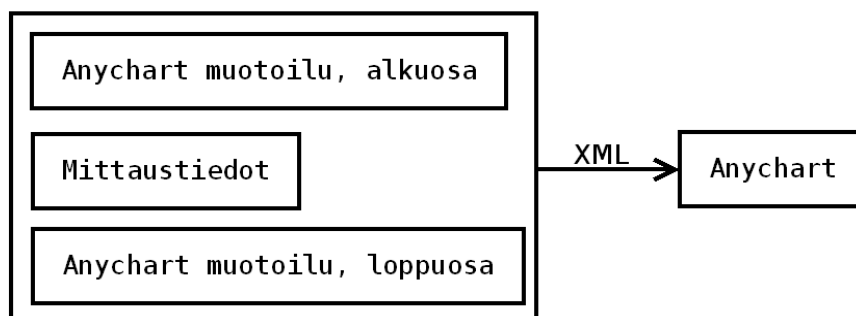
Vastaus tulee muodossa:

```
<markers>
<marker id="3504811" XMLtype="322" rep_name="TAMK_ALL" rep_id="1389"
con_lat="0" con_lng="0" distance="0" total_dis="0" remarks=""
xlog_humidity4="25.6" xlog_temp4="19.4" xlog_devdate="2010-03-29 14:41"
xlog_date="2010-03-29 17:41" />
<marker id="3504812" XMLtype="322" rep_name="TAMK_ALL" rep_id="1389"
con_lat="0" con_lng="0" distance="0" total_dis="0" remarks=""
xlog_temp11="21.2" xlog_humidity22="24" xlog_temp3="21.2"
xlog_humidity3="23.8" xlog_humidity11="21.8" xlog_humidity9="22.5"
xlog_temp9="21.3" xlog_devdate="2010-03-29 14:40" xlog_date="2010-03-29
17:40" />...
</markers>
```

Esimerkki 2: MAP_ROUTE_XML-kysely

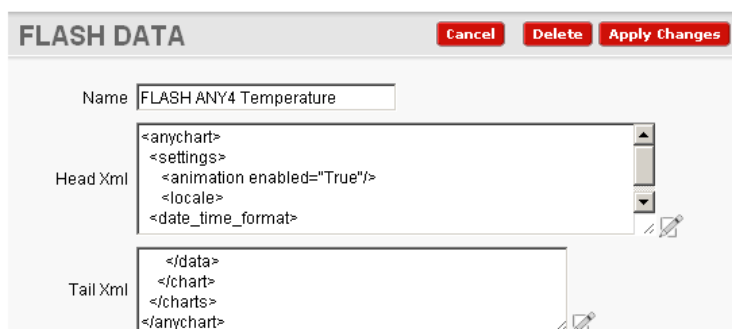
4.2.3 FLASH_2D_XML

FLASH_2D_XML luo XML-tiedoston, johon on yhdistetty alkuun ja loppuun Anychart-työkalun flash-muotoiluun tarvittavat määrittäykset (kuvio 12).



Kuvio 12: Anychart-työkalun käyttämän XML:n tiedonrakenne

Kyselyä käytetään erilaisten Anychartin flash-kaavioiden luonnissa. Muotoilu määritellään palvelimella ja yhdistetään FLASH DATA ID:n (264) avulla kyselyyn. Kuviossa 13 voidaan nähdä, että ”Head Xml” -osiossa määritellään XML-tiedoston alkuosaan lisättävät muotoilutiedot ja ”Tail Xml” -osiossa loppuosa. Näiden välille sijoitetaan mittaustieto Anychartin ymmärtämässä muodossa data-kenttien sisään (esimerkki 3). Työssä kyselyä käytetään mittaustietojen historiatiedon kaavioiden esittämiseen Anychart-työkalun avulla.



Kuvio 13: Flash-kysely

Kysely tapahtuu osoitteella:

http://www.../FLASH_2D_XML?TEMPLATE_ID=264&DATA_TYPE=TEMP&REP_ID=1389&CHART=ANY2DV4&SESSION_ID=1136196346114499&APPLICATION_ID=200&START_DATE=2010-03-21&END_DATE=2010-03-21&INC_LINKS=NO&step_time=15

Vastaus tulee muodossa:

```
<anychart>
  <settings>
    <animation enabled="True" />
    <locale>
    <date_time_format>
      <format>%yyyy/%MM/%dd %HH:%mm</format>
    </date_time_format>
    </locale>
  </settings>
  ...
  <data>
  <series type="Line" name="Temp 01" id="temp1" color="0x3366FF" border_color="0x3333FF" dots_radius="3">
  <point id="212355" x="2010/03/21 23:55" y="18.9">
  </point>
  <point id="212339" x="2010/03/21 23:39" y="18.83">
  </point>
  <point id="212324" x="2010/03/21 23:24" y="18.8">
  </point>
  <point id="212309" x="2010/03/21 23:09" y="18.8">
  </point>
  <point id="212250" x="2010/03/21 22:50" y="18.75">
```

```

</point>
...
</series>
...
</anychart>

```

Esimerkki 3: FLASH_2D_XML-kysely

4.3 Käyttöliittymän rakenne

Käyttöliittymä on tehty TappIT Oy:n tarjoamalle palvelinrajapinnalle. Se on toteutettu käyttäen JavaScriptiä, HTML:ää, CSS-tyylitiedostoja, XML-kyselyjä ja Anychart-työkalua.

Käyttöliittymän pääasialliset lähdekoodit on tässä työssä jaettu kolmeen osaan: HTML, DATA ja ENGINES (kuvio 13). Lisäksi tyyli-tiedostot ja kuvat sijaitsevat toisella palvelimella. HTML-osiossa esitellään käyttöliittymän käyttämät kirjastot ja muotoillaan sivujen yleinen ulkoasu. DATA-osiossa määritellään sivujen toiminnallisuus ja tulostetaan sivuille mittaustietoa eri muodoissa. ENGINES-osio sisältää kaikki XML-kyselyt.

DEV-I Report Template Layouts Create									
Page Type <input type="text" value="ALL"/>									
Js Code	Name	Code	Js Version	Page Type	Report template	Remarks	Sort Seq ▲	Js Ready	
	HTML	-	0	STATUS	TAMK_A_Report_template2010	-	01	No	
	DATA	-	0	STATUS	TAMK_A_Report_template2010	-	02	No	
	ENGINES	-	0	STATUS	TAMK_A_Report_template2010	-	99	No	

Kuvio 13: Käyttöliittymän lähdekoodi jaettuna kolmeen osaan.

4.3.1 Mittaustietojen rakenne

Antureiden mittaustiedot tallennetaan anturin tyyppin mukaisesti taulukoihin (esimerkki 4). Riippuen siitä minkälaista mittaustietoa anturi lähettää, sille annetaan myös tyyppimäärittäminen, joka tallennetaan taulukkoon ”type”. Tässä työssä käytettyjä tyyppimäärittämiä on kaksi: THB-anturille ja THCLR-anturille. Lämpötilalle, kosteudelle ja hiilidiok-

sidille tallennetaan myös aikaleimat taulukoihin, jotka haetaan MAP_ROUTE_XML-kyselyn avulla.

```
// anturin tyyppi
var type = new Array();
// mittauksetiedot
var name = new Array();
var temp = new Array();
var humidity = new Array();
var batt = new Array();
var co2 = new Array();
// aikaleimat
var tempT = new Array();
var humidityT = new Array();
var co2T = new Array();
```

Esimerkki 4: Mittaustietojen tallennus

4.3.2 Mittaustietojen tallentaminen

Antureiden aktiivisuus selvitetään MAP_ROUTE_XML-kyselyn avulla saaduista mittauksetiedoista. Jos anturi on lähettänyt tietoa tukiaseman kautta palvelimelle viimeisen sadan minuutin sisällä, todetaan että anturi on aktiivisessa tilassa. Samalla tallennetaan mittauksetieto ja sen aikaleima. Kun laite todetaan aktiiviseksi, asetetaan status-taulukon arvoksi nolla (esimerkki 5). Tällöin sovelluksen listatessa laitteita muuttuu aktiiviseksi todetun laitteen tila punaisesta vihreäksi (näkyvä esitellään kappaleessa 5.2).

```
for (var m = 0; m < 100; m++) { // markers = 100
  for (var i = 1; i <= DEVCOUNT; i++){

    if(tempLock[i]==false){
      if(isNaN(parseFloat(markers[m].getAttribute("xlog_temp"+i))))==false){
        temp[i]= parseFloat(markers[m].getAttribute("xlog_temp"+i));
        tempT[i]=markers[m].getAttribute("xlog_date");
        tempLock[i]=true;
        status[i]=0; // valo vihreäksi
      }
    }
    if(humLock[i]==false){
      if(isNaN(parseFloat(markers[m].getAttribute("xlog_humidity"+i))))==false){
        humidity[i]=parseFloat(markers[m].getAttribute("xlog_humidity"+i));
        humidityT[i]=markers[m].getAttribute("xlog_date");
        humLock[i]=true;
        status[i]=0; // valo vihreäksi
      }
    }
    if(co2Lock[i]==false){
      if(isNaN(parseFloat(markers[m].getAttribute("xlog_co"+i))))==false){
        co2[i]=parseFloat(markers[m].getAttribute("xlog_co"+i));
        co2T[i]=markers[m].getAttribute("xlog_date");
        co2Lock[i]=true;
        status[i]=0; // valo vihreäksi
      }
    }
  }
}
```

Esimerkki 5: Selvitetään laitteen aktiivisuus ja tallennetaan mittauksetietoja ja aikaleimat.

4.3.3 Mittaustietojen tulostaminen

Mittaustiedot tulostetaan sivulle HTML:n DIV:n avulla. Divit on sijoitettu sivuille HTML-osiossa ja DATA-osiossa tieto tulostetaan. Esimerkissä 6 voidaan nähdä työssä käytetyn tiedontulostustavan yksinkertainen esimerkki. Esimerkissä tulostetaan laitteen 5 nimi, lämpötila ja sen aikaleima Status-sivulle DIV-kenttään nimeltä "statusTable2".

```
var print2 ;  
print2="<b>"+name[5]+"</b><br>Temp: "+temp[5]+"&deg;C<br><i>"+tempT[5]+"</i>";  
document.getElementById("statusTable2").innerHTML = print2;
```

Esimerkki 6: Tiedon tulostaminen sivulle.

5 Käyttöliittymän käyttö

Tässä osiossa esitellään käyttöliittymän toiminta käyttäjän näkökulmasta. Käyttöliittymän näkymä voidaan määritellä palvelimelta ”yrityskohtaisesti”. Jokainen käyttäjä voi kuulua johonkin yritykseen ja yritykselle voidaan määritellä tietynlainen näkymä palveluun. Tässä työssä tehty näkymä voidaan siis määrittää avautuvaksi kenelle tahansa palvelimen käyttäjälle.

5.1 Kirjautuminen

Palvelimelle kirjaututaan osoitteella ”http://www/TAMK/”. Palvelimelle yhdistäminen vaatii että portti ”7777” on avoinna. Tunnuksia palveluun voidaan luoda tarpeiden mukaan eri oikeuksilla peruskäyttäjille ja kehittäjille (kuvio 14).



Kuvio 14: Kirjautuminen

5.2 Pääsivu

Käyttäjän kirjautuessa palvelimelle, aukeaa tässä työssä tehdyn käyttöliittymän päänäkyvä, josta voidaan nähdä yhdistettyjä tietoja monilta eri laitteilta. Näkymän yläosassa nähdään kaikkien laitteiden ja aktiivisten laitteiden määrä. Kuviossa 15 voidaan nähdä että 21 laitetta 27:stä järjestelmään kytketystä laitteesta on ollut tuolloin aktiivisina. Osa laitteista on poissa käytöstä ja verkon kantama-alueen ulkopuolella.

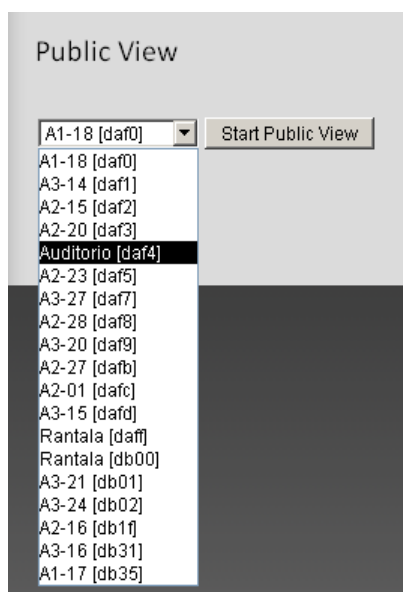
Sivulta löytyy myös diagrammi, jossa on vertailtavana kaikkien aktiivisten THB-antureiden lämpötila-arvot. Samanlaiset diagrammit löytyvät myös ilmankosteudesta sekä hiilidioksidista. Sivun alareunasta pääsee aloittamaan julkisen näkymän.



Kuvio 15: Käyttöliittymän pääsivu

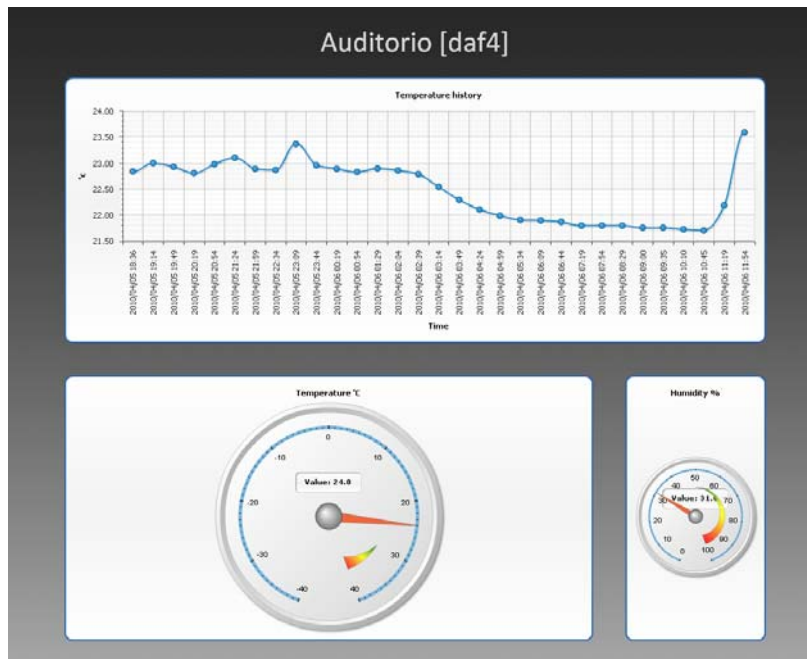
5.3 Julkinen näkymä

Julkisessa näkymässä tiedot näkyvät tilakohtaisesti. Alasvetovalikosta voidaan valita näytettävä tila ja painamalla ”Start Public View” -nappia, sovellus siirtyy näyttämään tietoja (kuvio 16).



Kuvio 16: Julkisen näkymän valinta

Näkymän käynnistyttyä ruudulle ilmestyvät lämpötilan ja kosteuden mittarit, jotka näyttävät viimeisimpiä mittaustietoja tilasta. Tällä välin sovellus parsii XML-tietoa tilan lämpötilan historiasta. Lämpötilan historiatiedon valmistuttua sovellus tuo näkyviin diagrammin (kuvio 17). Mittaustiedot päivittyvät minuutin välein. Tässä näkymässä lämpötilan historiatiedot näkyvät vuorokauden taaksepäin (tämä on asetus, joka on helposti muutettavissa kyselyn ehdoista).



Kuvio 17: Julkinen näkymä

5.4 Laitelista

Laitteiden listauksessa sovellus tulostaa käyttäjän näkymään kaikki langattomat anturit ja jakaa ne tyyppin mukaan kahteen ryhmään: THB- ja THCLR-laitteisiin. Sovellus ei ole riippuvainen siitä, kuinka monta laitetta verkkoon on liitetty (kuvio 18).

Laitelistasta voidaan ensimmäisenä nähdä laitteiden nimet, jotka tulevat palvelimelta käyttäjän määritysten mukaan. Lisäksi nähdään laitteista muita tietoja, kuten lämpötila, kosteus, patteriarvo, hiilidioksidi ja tässä tapauksessa signaalin voimakkuus (XRR).

Rivin oikeassa laidassa on sarake ”Status”. Siitä voidaan nähdä, onko laite aktiivisena. Jos laite on ollut tarpeeksi pitkään poissa käytöstä tai verkon kantaman alueen ulkopuolella (100 minuuttia), sen tila muuttuu punaiseksi. Lisäksi ”Details”-sarakkeesta voidaan päästä tarkastelemaan laitteiden mittaustietojen lisätietoja.

	NAME	TEMP	HUMIDITY	BATTERY	XRR	Status	DETAILS
1	A1-18 [daf0]	21.5	29.4	6.1	-8.6	●	▶
2	A3-14 [daf1]	24.9	23.1	6.4	-9.4	●	▶
3	A2-15 [daf2]	23.1	29.9	3.6	-10.2	●	▶
4	A2-20 [daf3]	20.8	30	6	-9.3	●	▶
5	Auditorio [daf4]	23.7	31.9	6.2	-6	●	▶
6	A2-23 [daf5]	21.3	29.6	6.2	-8.4	●	▶
7	A3-27 [daf7]	22.3	26.8	6.1	-9.4	●	▶
8	A2-28 [daf8]	22.1	31.9	6.2	-8.8	●	▶
9	A3-20 [daf9]	23.2	26.8	5.9	-10	●	▶
10	A1-21 [dafa]	NaN	18	3.6	NaN	●	▶

Kuvio 18: Laitelistaus

5.5 Lisätiedot

Yhden laitteen mittaustietoja ja lisätietoja voidaan tarkastella erikseen ”Details”-sivulla (kuvio 19). Sivun avulla voidaan tarkastella kaikkia anturin lähettämiä mittaustietoja. Sivun on lajiteltu otsikoihin nimen, arvon ja ajan mukaan. Näiden mittaustietojen lisätietoina tässä työssä on ainoastaan aikaleima, joka on näillä mittauksilla sama, mutta järjestelmän määritysten mahdollisesti muuttuessa voi tulevaisuudessa olla tilanne, että esimerkiksi lämpötila ja kosteus olisi mitattu eri aikaan.

Name	Value	Time
Location:	A1-18 [daf0]	
Temperature:	21.5	2010-04-06 12:39
Humidity:	29.4	2010-04-06 12:39
Battery:	6.1	
XRR:	-8.6	

Kuvio 19: Anturin mittaustiedot

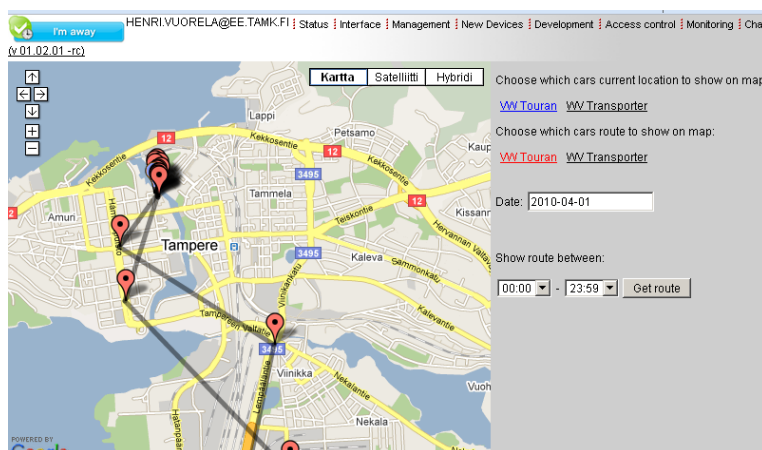
6 Verkon hyötykäyttö TAMKissa

Langatonta ZigBee-verkkoa on hyödynnetty TAMKissa syksyllä 2009 käynnistyneen sensoriverkkoprojektin osalta. Tässä kappaleessa tutustutaan hieman tähän asti tehtyihin töihin ja hyötykäytön mahdollisuuksiin TAMKissa yleisesti.

Projektin töiden tuloksina on saatu aikaiseksi esimerkiksi karttasovellus, joka hyödyntää Tampereen ammattikorkeakoulun autoihin asennettuja Gateway-laitteita, joiden lähettämän paikannustiedon avulla on voitu luoda sovellus, jonka avulla kuljettua reittiä on voitu seurata tietyltä ajanjaksolta. Toinen projektiin liittyvä sovellus oli S60-alustalle kosketusnäyttösarjan puhelimille tehty mobiilisovellus, joka hyödyntää Web Runtime (WRT) -teknologiaa.

6.1 GPS-paikannus

Gateway-tukiasemia asennettiin kahteen Tampereen ammattikorkeakoulun autoon. Asennus suoritettiin sensoriverkko-projektin tehtävänä. Tehtävän tarkoituksena oli kerätä paikannustietoa autojen liikkeistä. TappIT Oy hallinnoimalle palvelimelle kirjoitettiin HTML:ää ja Javascriptiä käyttämällä sovellus, josta käyttäjä pystyy valitsemaan haluamansa päivän ajohistoriatiedot Google Maps -karttaan sijoitettuna. Sovellus tuli hyötykäyttöön TAMK:n henkilökunnalle.



Kuvio 20: GPS-paikannuskartta

6.2 Mobiilisovellus

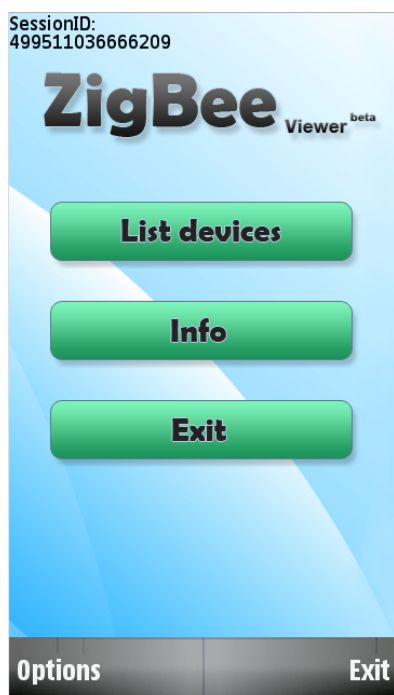
Tämän opinnäytetyön lisäosana toteutimme Sensoriverkko-projektin ja Graafiset käyttöliittymät -kurssin tehtävänä mobiilisovelluksen S60-alustalle kosketusnäyttösarjan puhelimille. Työ painottui mobiilikäyttöliittymän suunnitteluun ja toteutukseen. Työssä käytimme Web Runtime (WRT) -teknologiaa, jonka avulla on mahdollista tehdä widget-pienoissovelluksia. Ne yhdistävät liikkuvuuden edut Internetin tarjoamiin mahdollisuuksiin. Tämän tyyllisiä pienoissovelluksia ovat esimerkiksi Weatherbug ja RSS-lukija. WRT tarjoaa mahdollisuuden tehdä nopeasti ja helposti Internetissä toimivat palvelut käyttäjän laitteeseen, jolloin palvelun tieto on myös käyttäjäkohtaista.

Widgetien tekoon käytetään perinteisiä web-teknologioita kuten HTML:ää, CSS:ää ja JavaScriptiä. Web-teknologioita käyttämällä ohjelmistokehitys on nopeaa ja sillä voidaan maksimoida alustariippumattomuus suhteessa työn määrään.

Toteutimme Nokian S60 5th -sarjan puhelimeen (kosketusnäyttöpuhelin) ZigBee-anturiverkon mobiilikäyttöliittymän, jolla käyttäjä voi selata reaaliaikaisesti eri tilojen mittaustietoja. Anturiverkko mittaa Tampereen ammattikorkeakoululla useista eri luokista mittausrvoja, kuten lämpötilaa, kosteutta, patteriarvoa ja joidenkin laitteiden hiilidioksidipitoisuutta. Ohjelmointiympäristönä käytimme Aptana web 2.0:aa, johon asennettiin Nokian WRT-plugin.

Anturiverkon tiedot tallentuvat palvelimelle, josta ne voidaan XML-kyselyiden avulla hakea mihin tahansa sovellukseen. Sovellus kirjautuu ensin TappIT Oy:n hallinnoimalle palvelimelle ja hakee sitten XML-tiedoston, parsii sen ja tuo tiedot mobiilikäyttöliittymään.

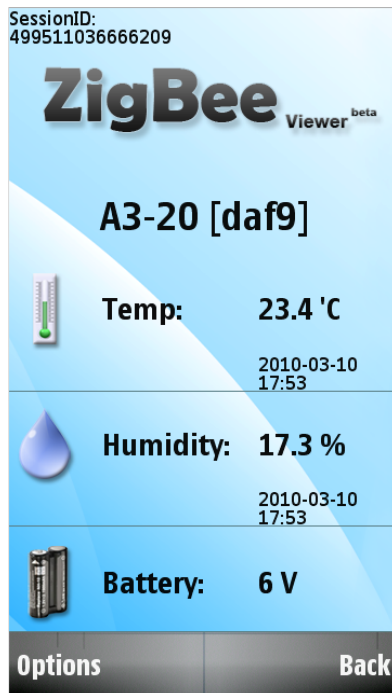
Sovelluksen päänäkymässä ylhäällä näkyvä sessionID kertoo, että yhteys on palvelimeen saatu ja tiedot haettu onnistuneesti. ”Lis devices” -linkki listaa kaikki anturit yhdelle sivulle. Sivua voi selata alaspäin vetämällä ruutua ylös. Laitetta klikattaessa ruudulla aukeaa näkymä, josta voi selata laitteen mittaamia tietoja. Esimerkiksi lämpötila on 23,4 °C tilassa A3-20 klo 17.53 päivämäärällä 10.3.2010.



Kuvio 21: Sovelluksen päänäköymä.



Kuvio 22: Sovelluksen laitelista. Sovellus listaa kaikki anturit yhdelle sivulle.



Kuvio 23: Laitteen mittaustietojen näkymä. Laitelistasta laitetta klikattaessa ruudulla aukeaa näkymä, josta voi selata laitteen mittaamia tietoja.

6.3 Hyötykäyttö opetuksessa

Laitteiden hyötykäyttöä opetuksessa olisi mahdollista toteuttaa monella tavalla. XML-tiedon parsiminen on yksi osa-alue, joka tulee esille monella eri kurssilla. Reaaliaikaiset mittaustiedot voisivat tuoda tällaisiin aiheisiin lisää sisältöä.

Selainpohjainen ohjelmointi on pienessä roolissa tietotekniikan koulutusohjelmassa. Tätä osaamista olisi mahdollista monipuolistaa erilaisilla projekteilla, jotka liittyvät TAMK:ssa sijaitsevan sensoriverkon anturien mittaustietojen hyödyntämiseen.

7. Yhteenveto

Vähän virtaa kuluttavat langattomat ZigBee-sensoriverkot voivat olla hyödyksi monilla eri tuotannollisilla aloilla. Mitä enemmän asiaan perehtyy, sitä enemmän käyttömahdollisuuksia verkolle ja tekniikalle voi löytää.

Langattomien ZigBee-verkkojen hyödyntäminen ja käyttö eri aloilla ja ympäristöissä rajoittuu lähinnä mielikuvitukseen, minkä vuoksi on ollut aiheena mielenkiintoinen ja hyödyllinen.

Käyttöliittymät ovat oleellinen osa kerätyn tiedon hyödyntämistä. Työssä tehdyn käyttöliittymän tarkoituksena on ollut tutustua hieman verkon eri hyödyntämismahdollisuuksiin ja saada myös kehitystyötä alulle verkon hyödyntämisen osalta Tampereen ammattikorkeakoulussa.

Opinnäytetyössä tehdyn käyttöliittymän jatkokehitys ja sen eri osa-alueiden hyödyntäminen on ideallisesti mahdollista monissa erilaisissa sovelluksissa ja järjestelmissä. Selainpohjaisten käyttöliittymien pohjalta on helppo saada ideoita esimerkiksi mobiilisovelluksiin ja toteuttaa osittain samoja palveluita. Tästä hyvänä esimerkkinä on tämän opinnäytetyön pohjalta syntynyt mobiilisovellus (esitellään kappaleessa 6.2), joka pystyi hyödyntämään lähes suoraan työssä käytettyä tiedonkeruutapaa.

Lähteet

Wikipedia: ZigBee [www-dokumentti]. [Viitattu 17.2.2010].

Saatavissa: <http://fi.wikipedia.org/wiki/ZigBee>

ZigBee Alliance: ZigBee [www-dokumentti]. [Viitattu 17.2.2010].

Saatavissa: <http://www.zigbee.org>

Tutorial-Reports: ZigBee Tutorial [www-dokumentti]. [Viitattu 17.2.2010].

Saatavissa: <http://www.tutorial-reports.com/wireless/zigbee/tutorial.php>

Wikipedia: XML [www-dokumentti]. [Viitattu 14.3.2010].

Saatavissa: <http://fi.wikipedia.org/wiki/XML>

Wikipedia: World Wide Web Consortium [www-dokumentti]. [Viitattu 14.3.2010].

Saatavissa: http://fi.wikipedia.org/wiki/World_Wide_Web_Consortium

Wikipedia: JavaScript [www-dokumentti]. [Viitattu 14.3.2010].

Saatavissa: <http://fi.wikipedia.org/wiki/JavaScript>