

Marcus Casagrande

ABB:n säätöyksikön mobiili kontrollijärjestelmä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

10.5.2013

Tekijä Otsikko	Marcus Casagrande ABB:n säätöyksikön mobiili kontrollijärjestelmä
Sivumäärä Aika	24 sivua 10.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmointi
Ohjaaja	yliopettaja Kari Salo
<p>Insinööriyönä kehitettiin prototyyppi älypuhelinsovelluksesta ABB:lle, joka on automaatiotekniikkaan ja sähkövoimatekniikkaan keskittyvä teollisuuskonserni. Sovellus pystyy etänä monitoroimaan ABB:n säätöyksikössä kiinni olevalta moottorilta tullutta dataa ja muuttamaan sen tilaa sekä tiettyjä parametreja. Sovellus tehtiin Android-ohjelmointialustalle. Sovelluksen ohjelmointiin käytettiin Java-ohjelmointikieltä.</p> <p>Työn tavoitteena oli selvittää, pystyykö moottorin arvoja monitoroimaan ja muuttamaan mobiilipäätelaitteella. Tavoitteena oli myös toteuttaa toimiva sovellus tähän tarkoitukseen.</p> <p>Sovelluksen käyttämä järjestelmä koostuu kahdesta eri osasta: taustajärjestelmästä sekä mobiilipäätelaitteesta. Taustajärjestelmä toimii automaattisesti taustaprosessina ja kerää sekä lähettää dataa moottorilta erilliseen etämonitorointiportaaliin, josta sitä voidaan käydä lukemassa. Mobiilipäätelaitteella oleva sovellus taas antaa käyttäjälle mahdollisuuden vaikuttaa taustajärjestelmässä olevan moottorin toimintaan ja tarjoaa tähän yksinkertaisen käyttöliittymän.</p> <p>Mobiilipäätelaite kommunikoi taustajärjestelmän kanssa ABB:n oman etämonitorointiportaalin sekä virtuaalipalvelimella olevan hallintasivun kautta. Nämä toimivat sovelluksen rajapintoina ja niihin otetaan yhteyttä käyttäen HTTPS- sekä HTTP-protokollaa.</p> <p>Insinööriyön lopputuloksena oli toimiva monitorointisovellus maailman suosituimmalle älypuhelinohjelmistoalustalle, jolla voidaan monitoroida moottorilta tullutta dataa ja muuttaa moottorin tilaa.</p>	
Avainsanat	ABB , Android, Kontrolli, Mobiili, Monitorointi, Moottori

Author(s) Title	Marcus Casagrande Mobile controlling for ABB motor drive
Number of Pages Date	24 pages 10 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Programming
Instructor(s)	Kari Salo, Principal Lecturer
<p>The purpose of the project described in this thesis was to develop a prototype smartphone application for the ABB group, a company that specializes in automation and power engineering. The application can monitor data sent by a motor connected to an ABB drive, change the motor's state and certain parameters. The application was developed for the Android platform. Android Java was used as the programming language.</p> <p>The objective of this project was to find out if it is possible to monitor and change motor values using a mobile device. The objective was also to produce a working prototype for this purpose.</p> <p>The system, which the application uses, consists of two parts: the background system and the mobile device. The background system works automatically in the background where it collects and sends data from the motor to a separate remote monitoring portal where it can be read. The application in the mobile device gives the user an option to affect motor functionality running in the background and provides a simple user interface to do it.</p> <p>The mobile device communicates with the background system through ABB's own remote monitoring portal and a control page located in a virtual host. These serve as application interfaces and are connected by using HTTPS and HTTP protocols.</p> <p>The final outcome of the thesis is a working monitoring application to the world's most popular smartphone platform that can be used to monitor data sent by a motor and change the state of the motor.</p>	
Keywords	ABB, Android, Control, Mobile, Monitoring, Motor

Sisällys

Lyhenteet

1	Johdanto	1
2	Android-käyttöjärjestelmä	2
2.1	Yleistä Androidista	2
2.2	Android-ohjelmistoalusta	3
2.3	Android-versio 2.2	4
2.4	Android-sovelluskehitystyökalut	5
2.5	HttpClient vs WebView	7
3	Kommunikaatioprotokollat	8
3.1	HTTP ja suojattu HTTPS	8
4	Arkkitehtuuri	10
4.1	Järjestelmän yleinen kuvaus	10
4.2	Taustajärjestelmä	11
5	Mobiilipäätte	14
5.1	Vaatimukset	14
6	Suunnittelu	16
6.1	Toteutus	17
7	Testaus	22
8	Jatkokehitys	22
9	Yhteenveto	23
	Lähteet	24

Lyhenteet

ADT	<i>Android Development Tools</i> . Androidin kehitystyökalut. Eclipse IDE-liitännäinen, joka on tarkoitettu Android-sovellusten kehitykseen.
API	<i>Application Programming Interface</i> . Yksi ohjelmointirajapinnan päätarkoituksista on tarjota käyttömahdollisuus yleisimmille toiminnoille.
CA	<i>Certificate Authority</i> . Virallinen sertifioija, joka myöntää digitaalisia sertifikaatteja sivustojen käyttöön.
HTTP	<i>Hypertext Transfer Protocol</i> . Protokolla, jota internet-selaimet ja palvelimet käyttävät tiedonsiirtoon.
HTTPS	<i>Hypertext Transfer Protocol Secure</i> . HTTP-protokollan salattu versio.
IDE	<i>Integrated Development Environment</i> . Ohjelmointiympäristö. IDE on yleensä yksi sovellus, johon on paketoitu mukaan sovelluskehityksessä tarvittavia ominaisuuksia, kuten tekstieditori, kääntäjä ja virheenjäljitin sekä tuki eri kehitystyökalukokoelmille.
SDK	<i>Software Development Kit</i> . Kehitystyökalupaketti. Ohjelmistokehittäjälle tarkoitettu paketti, joka sisältää kehitystyökalut tiettyä alustaa tai käyttöjärjestelmää varten.
SSL	<i>Secure Sockets Layer</i> . Salausprotokolla, jolla voidaan suojata internet-sovellusten tietoliikenne IP-verkoissa.
TLS	<i>Transport Layer Security</i> . Salausprotokolla, SSL-salauksen uudempi versio.

1 Johdanto

Tämä insinööri työ on tehty ABB oy:lle, joka on automaatiotekniikkaan ja sähkövoimatekniikkaan keskittyvä teollisuuskonserni.

Yksi ABB:n sähkövoimatekniikkaan liittyvistä alueista on moottoreiden sekä pumppujen energiatehokkuus. Tehokkuutta on mahdollista tehostaa taajuusmuuttajan avulla. Taajuusmuuttaja on laite, joka säätelee portaattomasti moottorin pyörimisnopeutta ja vääntömomenttia. Teollisuudessa ja toimistoissa eri puolilla maailmaa on käytössä miljoonia moottoreita. Moottoreita käytetään, muun muassa, jätevesi- ja kastelupumpuissa, lypsykoneissa ja hiihtohisseissä, paperikoneissa ja voimaloiden puhaltimissa sekä sahojen kuljettimissa ja sairaaloiden ilmanvaihtojärjestelmissä [1].

Osa näistä pumpuista tai moottoreista voi sijaita syrjäisillä alueilla, minne on henkilökunnan on hankala päästä tarkastamaan moottorin toimintaa tai muuttamaan asetuksia. Yksi ratkaisu tähän voisi olla mobiilisovellus, jolla käyttäjä voi etänä nähdä moottorin tilan ja tarvittaessa muuttaa sen asetuksia.

Insinööri työnsä tarkoituksena oli selvittää, pystyykö etänä olevaa moottoria kontrolloimaan mobiilipäätelaitteella, joka tässä projektissa oli Android-puhelin. Työssä suunniteltiin ja toteutettiin toimiva prototyyppi mobiililaitteelle, jolla voidaan monitoroida ja kontrolloida etänä moottorin toimintaa.

Insinööri työnsä järjestelmä tehtiin kahden ihmisen työryhmässä. Projektin osa-alueet jaettiin siten, että itse vastasin sovelluksen toiminnallisuudesta, johon kuului muun muassa HTTP- ja HTTPS-yhteydet sekä varmenteiden käsittely. Työtoveri taas vastasi käyttöliittymästä.

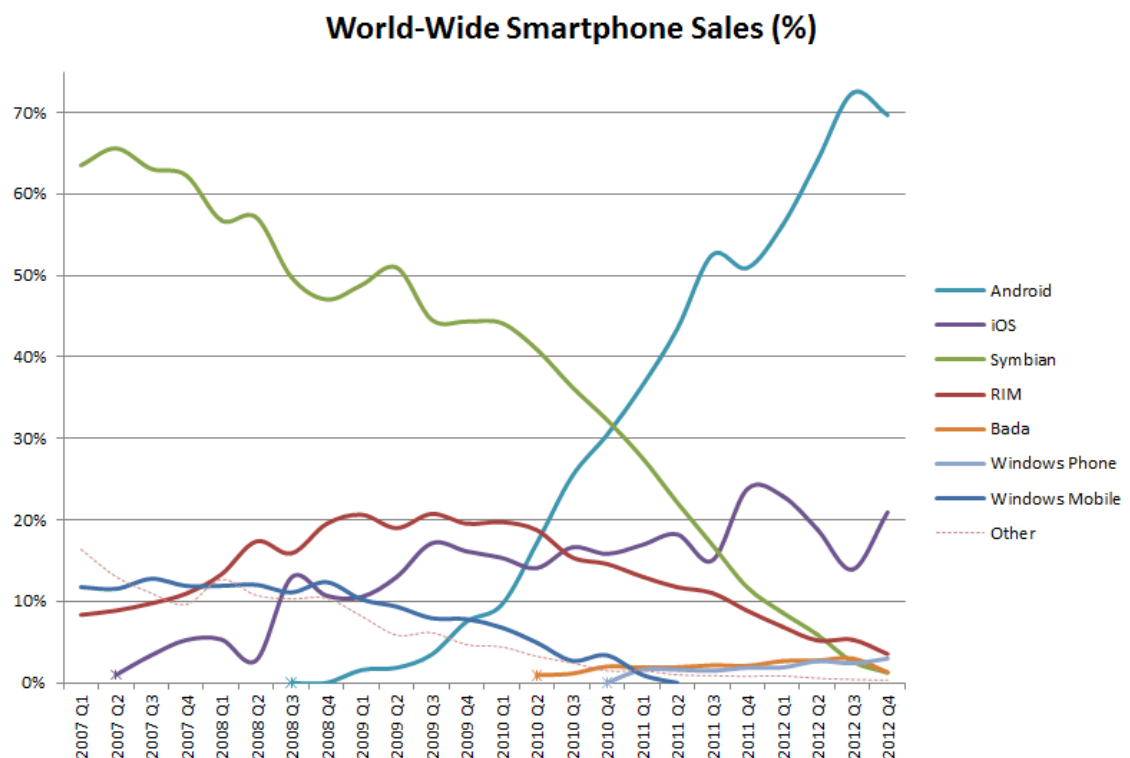
2 Android-käyttöjärjestelmä

2.1 Yleistä Androidista

Nykypäivän älypuhelimissa on suuri kilpailu, ja yksi keskeisimmistä osista puhelimen käytettävyyteen liittyen on luonnollisesti käyttöjärjestelmä. Android julkistettiin 5. marraskuuta 2007, ja se sai tasaisesti suosiota käyttäjien keskuudessa, kunnes nousi räjähdysmäisesti älypuhelinien markkinajohtajaksi vuonna 2010.

Vuoden 2012 alussa tehtyjen mittausten mukaan Android on maailman suosituin käyttöjärjestelmä älypuhelimissa.

Vuonna 2012 Android oli myydyin käyttöjärjestelmä älypuhelimissa ja voitti toisena olleen Applen IOS-käyttöjärjestelmän (kuva 1). Joulukuussa 2012 Android-käyttöjärjestelmän osuus älypuhelinmarkkinoilla oli 68,3 prosenttia, kun taas toisena tulleen Apple IOS:n osuus oli 18,8 prosenttia [2].



Kuva 1: Älypuhelimien myynti vuosittain [2].

Yksi syy Androidin suosioon on epäilemättä sen vapaa lähdekoodi. Tämä tarkoittaa sitä, että kuka tahansa voi itse kehittää ja tehdä sovelluksia Androidille. Siihen tarvitaan ainoastaan oikeat ohjelmointi työkalut sekä ohjelmointiosaamista. Nykyään on kuitenkin helppoa löytää ohjeita Android-sovellusten tekoon.

Google on myös kehittänyt Google Play -nimisen, internetissä toimivan sovelluskaupan, johon käyttäjät ja valmistajat voivat ladata omia sovelluksiaan. Google Play syntyi maaliskuussa 2012, kun Android Market, Google Music sekä eBookstore yhdistyivät. Google Play on valmiiksi esiasennettuna suurimmassa osassa Android-puhelimia ja sen käyttäminen vaatii ainoastaan internet yhteyden. [3.]

2.2 Android-ohjelmistoalusta

Älypuhelimien tekniikka ja ominaisuudet kehittyvät huimaa vauhtia, ja vuodesta 2008 lähtien Android-ohjelmistoalustasta on tullut monia eri versioita, jotka ovat parantaneet käyttöjärjestelmää, tuoneet uusia ominaisuuksia ja korjauksia vanhoihin ominaisuuksiin. Suuret versiopäivitykset ovat kaikki nimetty aakkosjärjestyksessä jälkiruuan tai sokerisen herkun mukaan. Esimerkiksi versio 1.5 oli nimeltään ”Cupcake” (vuokaleivos) ja sitä seuraava versio 1.6 ”Donut” (donitsi).

Ensimmäinen kaupalliseen myyntiin tullut Androidia käyttävä älypuhelin oli HTC Dream, joka julkaistiin 22.10.2008 (kuva 2)



Kuva 2: HTC Dream, ensimmäinen Android puhelin.[4]

2.3 Android-versio 2.2

Tämän insinööriyön sovellus on tehty käyttäen Android-versiota 2.2. Versiossa 2.2 on muun muassa laajennetut multimediaominaisuudet, parannuksia järjestelmän nopeuteen sekä käytettävyyteen ja parannuksia ohjelmointirajapinnan käyttöön.

Kaikki Android-sovellukset ovat tiettyyn pisteeseen asti taaksepäin yhteensopivia, joten versiolla 2.2 kehitettyjä ohjelmia voi käyttää vanhemman version laitteissa ja ne toimivat myös uudemmissa laitteissa. Tietenkin jos sovelluksessa on käytetty uudemman version ominaisuutta, jota vanhemmassa ei ole, ei se toimi vanhassa versiossa.

Taulukosta 1 voi nähdä Android käyttöjärjestelmän erot versioon 2.2 asti.

Taulukko 1. Android versioiden erot versioon 2.2 asti [5].

	Version name	Key user features added	Release date	API Level	Adobe Flash support
Android 2.2	Froyo	Speed improvements JIT implementation USB Tethering Applications installation to the expandable memory Upload file support in the browser Animated GIFs	2010 May 20	8	10.1
Android 2.1	Eclair	Updated UI	2010 Jan 12	7	No, Flash Lite 4 on some device
Android 2.0.1	Eclair		2009 Dec 3	6	No, Flash Lite 4 on some device
Android 2.0.1	Eclair	HTML Digital zoom Microsoft Exchange support Bluetooth 2.1 Live Wallpapers Updated UI	2009 Oct 26	5	No, Flash Lite 4 on some device
Android 1.6	Donut	Gesture framework Turn-by-turn navigation	2009 Sep 15	4	No, Flash Lite 4 on some device
Android 1.5	Cupcake	Bluetooth A2DP, AVRCP support Soft-keyboard with text-prediction Record/watch videos	2009 Apr 30	3	No, Flash Lite 4 on some device
Android 1.1	Banana bread	"Show" & "Hide" numeric keyboard, in caller application Ability to save MMS attachments	2009 Feb 9	2	No
Android 1.0	Apple pie	Download and updates via Android Market Web Browser Camera support Gmail, Contacts and Google Agenda synchronization Google Maps YouTube application	2008 Sep 23	1	No
Android 0.9			2008 Aug 18		No

2.4 Android-sovelluskehitystyökalut

Android-ohjelmien kehittäminen on helppo aloittaa ja siihen tarvitaan ainoastaan sovelluskehitystyökalupaketti. Sovelluskehitystyökalupaketti, eli Software Development

Kit (SDK) tarjoaa kaikki tarpeelliset ohjelmointirajapinnat (API-kirjastot) sekä kehitystyökalut ohjelmien rakentamiseen ja testaamiseen.

Suosituksena on tosin käyttää varsinaista Androidin kehitystyökalua (Android Development Tools, ADT). ADT on Eclipsen liitännäinen, joka tukee Android-sovellusten rakennusta käyttäen Eclipsen omaa ohjelmointiympäristöä (Interactive Development Environment, IDE) [6, s.11].

Kuvassa 3 näkyy sovelluskehityksessä käytetty emulaattori, jonka avulla ohjelmia voi testata suoraan tietokoneella.



Kuva 3: Sovelluskehityksessä käytetty emulaattori.

Eclipsen ADT on täysin ilmainen ja tarjoaa monia etuja verrattuna perinteiseen komentolinjakehitystyökaluihin. Näitä ovat muun muassa graafinen käyttöliittymä sekä erillinen sovelluksen käyttöliittymä (UI) suunnittelutyökalu. Tämän graafisen UI-työkalun

avulla voi rakentaa nopeasti prototyyppejä ja suunnitella sekä rakentaa sovelluksen käyttöliittymän.

Koska Android-koodi tehdään Java-kielellä, auttaa Javan aikaisempi tuntemus sovellusten tekemisessä. Aikaisemmin Java-sovelluksia tehneet käyttäjät huomaavat tuttujen tekniikoiden, merkintöjen ja kieliopin kääntyvän suoraan Androidille. Tosin tähänkin löytyy joitain poikkeuksia.

Myös olio-ohjelmointiin suuntautuneiden kehittäjien on helppo lähestyä Androidia. Androidin voima tulee sen laajasta ohjelmointirajapinnasta, joten kehittäjän ei tarvitse suoraan muistaa kaikkia Java-kohtaisia luokkia. [7, s.18.]

2.5 HttpClient vs Webview

Ohjelmat, jotka vaativat jatkuvaa verkkoyhteyttä toimiakseen voivat olla suhteellisen hankalia toteuttaa. Androidilla on tarjolla eri työkaluja verkkoyhteyden toteuttamiseen ja näistä kaksi ovat httpclient- sekä Webview- luokat.

HttpClient

Suurin osa verkkoa tarvitsevista Android-sovelluksista käyttävät HTTP-yhteyttä lähettääkseen ja vastaanottaakseen dataa. Android tarjoaa tähän kahta erilaista HTTP Client:ia: HttpURLConnectionia tai Apachen HTTP Client:ia. Molemmat pystyvät toteuttamaan HTTP-pyyntöjä, käsittelemään evästeitä sekä varmennuksia sekä hallitsemaan yhteyksiä [8].

Tämän projektin kommunikaatio hoidettiin käyttämällä Apachen HTTP Client:ia, tarkemmin sanottuna DefaultHttpClient-luokkaa, jolle tehtiin oma kustomoitu CustomHttpClient-olio. Yksi syy Apachen valitsemiseen oli sen laaja sekä joustava ohjelmointirajapinta (API).

Yksi työn isommista ongelmista oli saada httpclient-luokka hyväksymään SSL-varmenteet ja näyttämään suojattuja HTTPS-sivuja. Tämä saatiin toteutettua käyttämällä EasySSLConnectionFactory- sekä EasyX509TrustManager-luokkaa. Luokan

avulla ohjelma hyväksyy kaikki mahdolliset varmenteet, eikä erillistä vahvistusta tarvita. Tämä ei tietenkään ole ideaalinen tilanne, mutta riittävä tämän ohjelman testaukseen.

Webview

Webview on jatke Androidin View-luokalle, ja sen avulla sovellukseen voi suoraan upottaa Internet-sivun. Webview-luokkaa käytetään siis Internet-sivujen näyttämiseen ohjelmassa. Webview luokka toimii periaatteessa selaimena, mutta vakiona luokalla ei ole oikean selaimen toiminnallisuuksia, kuten osoitepalkkia tai eteen/taakse-nappeja.

Webview-luoka on helppo tapa toimittaa käyttäjälle dataa, joka vaatii jatkuvan internetyhteyden toimiakseen. Webview-luokalla ei erikseen tarvitse tehdä työläitä pyyntöjä palvelimelle, eikä kehittäjän tarvitse erikseen käsitellä ja näyttää pyydettyä dataa.

3 Kommunikaatioprotokollat

3.1 HTTP ja suojattu HTTPS

HTTP eli Hypertext Transfer Protocol on yleisesti käytetty kommunikaatioprotokolla, jota käytetään tiedon välitykseen verkossa. Ongelma HTTP:n kanssa on kuitenkin se, että HTTP ei ole täysin turvallinen, vaan on altis kolmannen osapuolen häirinnälle. Esimerkiksi kolmas osapuoli voi ”kuunnella” verkkoliikennettä ja tätä kautta saada selville, mitä käyttäjä sekä palvelin viestittävät toisilleen.

HTTPS eli Hypertext Transfer Protocol Secure on tästä turvallisempi muoto. Ennen tiedon lähettämistä se salataan erikseen käyttäen SSL (Secure Socket Layer) -protokollaa tai sen uudempaa muotoa SSL/TSL (Transport Security Layer).

HTTPS on käytännössä normaali HTTP-protokolla, johon on lisätty SSL-salaus, ja varmenteet ovat yksi tarpeellinen osa turvallisen yhteyden luomiseen.

Jos sivusto haluaa luoda käyttäjien välille turvallisen yhteyden, tarvitaan siihen ensimmäiseksi SSL-varmenne. Varmenteen (sertifikaatin) voi hakea kahdella eri tavalla:

- Jos sivuston ylläpitäjällä (Host) on jo virallisen sertifioijan eli Certificate Authority:n (CA) hyväksymä varmenne, voi sivusto pyytää varmennetta suoraan ylläpitäjältä.
- Sivusto voi myös hakea SSL-varmenteen viralliselta sertifioijalta (CA). Tämä kolmas osapuoli tutkii sivuston myöntää SSL-varmenteen.

Varmenteen voi siis ajatella olevan verkossa toimiva ID-kortti, jolla sivusto todistaa olevansa juuri se sivusto, miksi itseään kutsuu. Varmenne sisältää tietoa sen omistajasta kuten sähköpostiosoitteen, varmenteen haltijan nimen, varmenteen käyttötarkoituksen, voimassaoloajan sekä tietoa itse verkkotunnuksesta. Varmenteessa on myös erillinen CA:n varmenne ID, josta voi nähdä, kuka (CA) on hyväksynyt kyseisen sivuston varmenteen.

Varmenne sisältää myös yleisen avaimen eli Public Key:n, jota käytetään itse salauksessa hyväksi.

Käytännössä turvallisen HTTPS-yhteyden luominen sujuu seuraavalla tavalla:

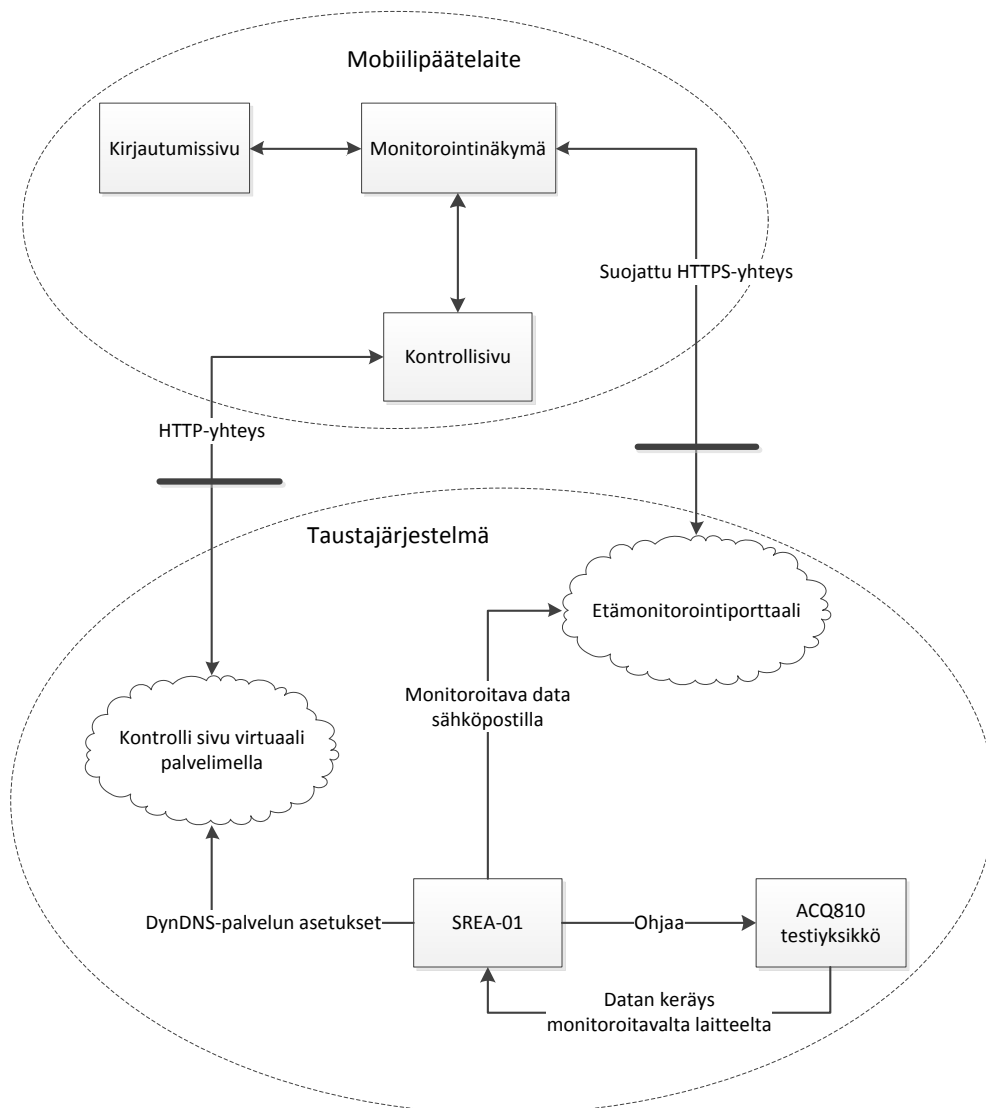
1. Selain pyytää palvelimelta turvallista yhteyttä, joka tapahtuu yleensä käyttämällä https:// -muotoa osoitteessa.
2. Palvelin lähettää selaimelle varmenteen ja yleisen avaimen.
3. Selain tarkistaa, että vastaanotettu varmenne on aito ja sen tiedot täsmäävät sivustoon.
4. Selain käyttää yleistä avainta tehdäkseen satunnaisen salausavaimen ja lähettää sen salatun URL:n välityksellä palvelimelle.
5. Palvelin kääntää selaimen lähettämän salatun datan luettavaksi käyttäen sen omaa salausavainta.
6. Palvelin lähettää pyydetyn dokumentin tai datan takaisin selaimelle salatussa muodossa.

7. Selain kääntää salatun datan luettavaksi ja näyttää sen käyttäjälle.

4 Arkkitehtuuri

4.1 Järjestelmän yleinen kuvaus

Insinöörityön kontrollijärjestelmän yleisen tason kuvaus koostuu kahdesta osasta: mobiilipäätelaitteesta sekä taustajärjestelmästä (kuva 4).



Kuva 4: Järjestelmän yleinen kuvaus.

Taustajärjestelmässä pyörii ACQ810-testiyksikkö, jossa on erikseen kiinni SREA-01-etäkäyttöliittymälaite. SREA-01 ohjaa moottoria, kerää sieltä tulevan datan ja lähettää sen eteenpäin ABB:n etämonitorointiportaaliin sähköpostin välityksellä. SREA-01-hallintatilaa ylläpidetään myös virtuaalipalvelimella, jota kautta käyttäjä voi etänä hallita moottorin toimintaa.

Mobiilipäätteellä käyttäjä kirjautuu taustajärjestelmässä pyöriin palveluihin ja pystyy tätä kautta käyttämään niitä. Mobiilipäätelaite hakee etämonitorointiportaalissa olevan datan käyttämällä suojattua HTTPS-yhteyttä ja näyttää sen käyttäjälle helposti luettavana listana.

Virtuaalipalvelimella olevaan hallintasivustoon otetaan yhteys käyttäen normaalia HTTP-yhteyttä.

4.2 Taustajärjestelmä

Jotta ohjelmaa pystyy käyttämään, täytyy taustalla pyörivien järjestelmien olla kunnossa.

Taustajärjestelmä koostuu neljästä eri komponentista, jotka ovat ACQ810-testiyksikkö (jossa moottori), SREA-01 ethernet adapter, ABB:n etämonitorointiportaali, joka toimii osoitteessa <https://www.mydrives.info> sekä virtuaalipalvelimella toimiva moottorin kontrollisivu.

ACQ810-testiyksikkö

ACQ810-testiyksikköä (kuva 5) käytetään moottorin tai pumpun simulointiin. Yksikössä on sisäänrakennettu moottori, jota käytetään hyväksi testauksessa. Normaaleissa ACQ810-yksiköissä ei ole kyseistä moottoria.



Kuva 5: ACQ810 testiyksikkö.

ACQ810-yksikköä käytetään esimerkiksi vesi- ja jätevesijärjestelmissä pumpun tai moottorin pyörimisnopeuden sekä vääntömomentin säätämiseen. Tällä voidaan tehostaa pumpun energiankulutusta.

SREA-01 Ethernet Adapter

SREA-01 ethernet adapter (kuva 6) on ABB:n käyttämä etäkäyttöliittymälaitte, jota käytetään laitteiden, kuten moottorien tai pumppujen, monitoroimiseen sekä ohjaukseen. SREA-01 tarjoaa selainpohjaisen palvelu- ja monitorointikäyttöliittymän.



Kuva 6: SREA-01 Ethernet Adapter.

Käyttöliittymän avulla käyttäjä pääsee käsiksi muun muassa. laitteen eri parametreihin, arvoihin ja tilaan. Tämän lisäksi SREA-01 pystyy keräämään laitteelta dataa, lähettämään hälytyksiä sekä toimimaan Modbus/TCP-porttina.

Yksi SREA-01:n tärkeimmistä ominaisuuksista liittyen tähän projektiin on datan kerääminen laitteelta. SREA-01:n asetuksista voi suoraan määrittää, kuinka usein näytteitä otetaan laitteelta. Datanäytteiden tiheyden voi määrittää 10 sekunnin ja 60 minuutin välille. Kerätty data voidaan tämän jälkeen lähettää eteenpäin sähköpostin avulla.

Toinen projektissa tarvittu ominaisuus on selainpohjainen etämonitorointi. SREA-01 pystyy siis näyttämään laitteilta tulleita arvoja, parametreja, laitteen nykyisen tilan sekä virhelokeja. Jos SREA-01 on kiinni useammassa laitteessa, voi selaimessa näyttää yhteenvetona kaikkien monitoroitavien laitteiden tilat erikseen. Asetuksista voi myös luoda kustomoituja näkymiä, joissa näytetään ainoastaan tärkeimmät arvot liittyen laitteeseen [9, s. 17].

Etämonitorointiportaali ja kontrollisivu

Sovelluksen taustajärjestelmään kuuluu ACQ810:n ja SREA-01 lisäksi ABB:n oma etämonitorointiportaali sekä virtuaalipalvelimella oleva kontrollisivu.

ABB:n etämonitorointiportaali on selainpohjainen työkalu, jonka avulla käyttäjä voi tarkkailla ja tutkia ABB-laitteiden (kuten ACQ810) toimintaa. Koska portaali toimii www-sivulla osoitteessa mydrives.info, sen käyttö on suhteellisen helppoa. Portaalin lukemiseen tarvitaan ainostaan selain ja toimiva internetyhteys.

Tässä projektissa laitteelta tullut data välitetään portaaliin sähköpostin välityksellä ja portaali näyttää datan käyttäjälle. Portaaliin tulevien sähköpostien tiheys on erikseen määritelty SREA-01-asetuksista.

Kontrollisivun ylläpito sijaitsee dyndns-virtuaalipalvelimella, mutta itse sisältö ja sen määrytykset tuleva SREA-01:n kautta.

Sovelluksessa käytettävä kontrollisivu on itseasiassa SREA-01:n normaali selainpohjainen hallintasivu. Hallintatilan etusivuksi on laitettu kustomoitu näkymä, joka avautuu ensimmäisenä käyttäjän kirjautuessa hallintaan. Hallintatila on laitettu toimintaan virtuaalipalvelimelle, jotta siihen pystyisi saamaan yhteyden etänä mistä tahansa.

5 Mobiilipäätte

5.1 Vaatimukset

Mobiilisovellusten kehittämisessä on otettava huomioon paljon eri asioita verrattuna esimerkiksi PC-sovelluskehitykseen. Mobiililaitteissa on muun muassa paljon pienemmät resurssit muistin sekä tehojen suhteen. Myös pienempi näyttö sekä käyttöliittymän rajoitteet tulee ottaa huomioon. Mobiilisovelluksia tulisi voida käyttää helposti yhdellä kädellä, joten sovelluksen käyttöliittymään tulee kiinnittää erityistä huomiota.

Mobiilisovelluksissa on myös tärkeää, että käyttäjälle annetaan tarpeeksi tietoa ohjelman senhetkisestä toiminnasta. Esimerkiksi taustalla suoritettavien tehtävien aikana on hyvä näyttää käyttäjälle prosessin latauspalkki.

Kirjautumisnäkyvä

Kun ohjelma käynnistetään, tulisi ensimmäisen sivun olla kirjautumissivu. Sivun pitäisi olla yksinkertainen ja sisältää ainoastaan käyttäjätunnus- ja salasana-tekstikentät sekä Kirjautu-napin, jolla käyttäjä pääsee eteenpäin.

Tekstikenttiin laitettavilla tunnuksilla käyttäjä kirjautuu <https://www.mydrives.info/> -sivustolle. Kirjautumisnäkyvän pitäisi siis olla yksinkertaistettu versio [mydrives.info](https://www.mydrives.info/) -kirjautumissivusta.

Jos käyttäjän antamat tunnukset ovat oikein, ohjelmassa siirrytään datan monitorointi sivulle. Muussa tapauksessa ohjelma ilmoittaa käyttäjälle, että tunnukset ovat väärät.

Käyttäjän täytyy pystyä lopettamaan ohjelma joko painamalla puhelimen valikkonappia tai sitten painamalla puhelimesta olevaa takaisin-nappia.

Monitorointinäkyvä

Monitorointisivulla käyttäjälle näytetään selkeästi ja yksinkertaistettuna moottorilta tullut data. Ohjelman tulisi itse huolehtia siitä, että sivulla oleva data päivitetään tietyn ajan välein.

Jos moottorilta ei ole tullut yhtään dataa, voi se johtua muun muassa siitä, että moottoriin on tullut vika tai se ei ole lainkaan toiminnassa. Tässä tapauksessa käyttäjälle pitäisi ilmoittaa, että uutta dataa ei ole ja kehottaa häntä tarkastamaan moottorin tila. Käyttäjän täytyy pystyä siirtymään kontrollisivulle, josta hän voi tarkistaa moottorin tilan ja esimerkiksi käynnistää moottorin, jos se ei ole päällä. Käyttäjällä täytyy myös olla vaihtoehto palata takaisin aloitussivulle.

Kontrollinäkymä

Kontrollinäkymässä käyttäjä voi halutessaan muuttaa moottorin tilaa (päälle/pois) sekä tiettyjä moottorin toimintaan liittyviä arvoja, kuten esimerkiksi nopeutta. Myös mahdollisten hälytysten ja virhetilanteiden näyttäminen olisi tarpeellista.

Näkymän tulisi olla helposti käytettävä ja ulkoasultaan yksinkertaistettu. Käyttäjän täytyy myös pystyä palaamaan takaisin edelliseen näkymään.

6 Suunnittelu

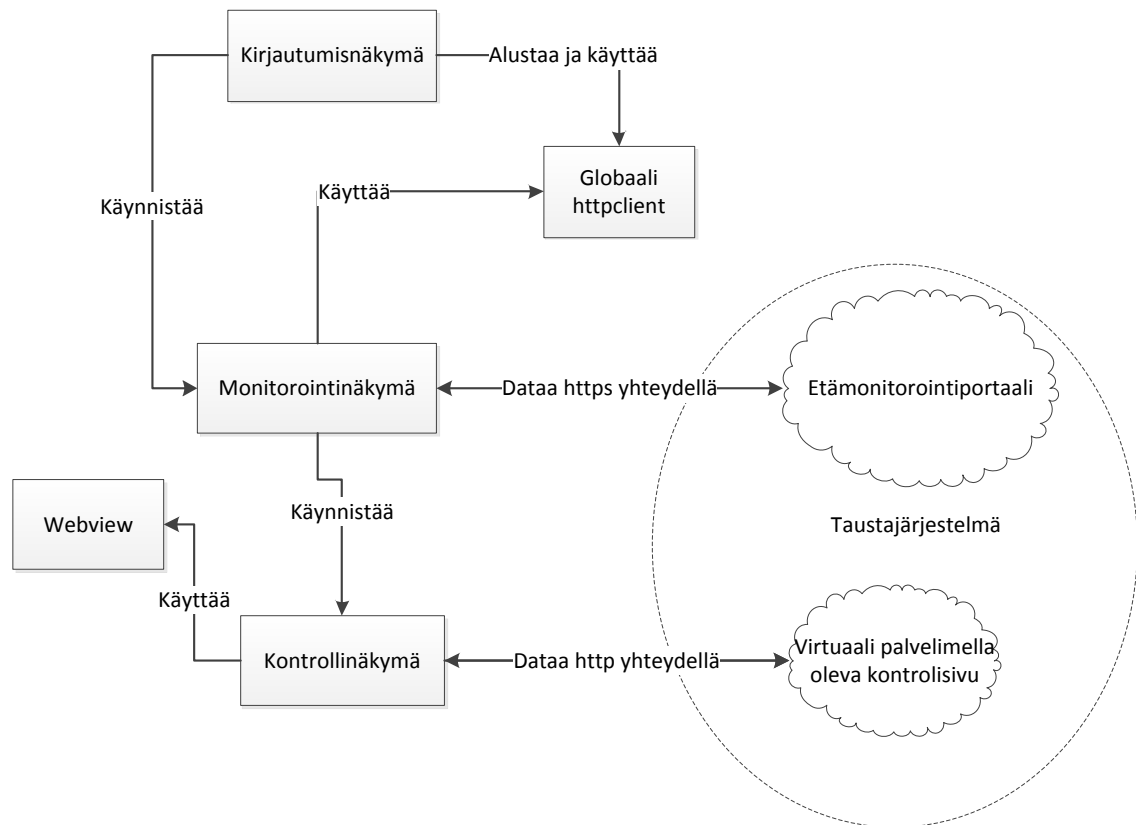
Sovelluksen suunnittelu aloitettiin kartoittamalla taustajärjestelmässä toimivat komponentit sekä selvittämällä niiden toimintaa. Tähän kuului muun muassa ACQ810-testiyksikön ja etämonitorointiportaalin toiminnan testaaminen sekä SREA-01:n asetusten määrittely. Samalla selvitettiin komponenttien yhteydet toisiinsa ja se, miten tieto niiden välillä liikkuu.

Taustajärjestelmän jälkeen siirryttiin sovelluksen käyttöliittymään. Käyttöliittymää suunniteltaessa oli tavoitteena tehdä siitä mahdollisimman yksinkertainen, jotta sitä olisi helppo käyttää. Vaikka käyttöliittymä on yksinkertainen, pitää käyttäjälle kuitenkin jatkuvasti antaa tietoa ohjelman tilasta ja esimerkiksi onnistuneesta tai epäonnistuneesta kirjautumisesta täytyy antaa erillinen ilmoitus.

Sovelluksen toiminnallisuuden suunnittelu oli helppo aloittaa, kun tarvittavien näkymien määrä sekä taustajärjestelmän toiminta oli tiedossa.

6.1 Toteutus

Kuvassa 6 näkyy yleinen kuvaus järjestelmän teknisestä toteutuksesta. Tässä työssä keskitytään kuitenkin tarkemmin mobiilipäätelaitteilla sijaitseviin toimintoihin.



Kuva 7: Järjestelmän tekninen toteutus.

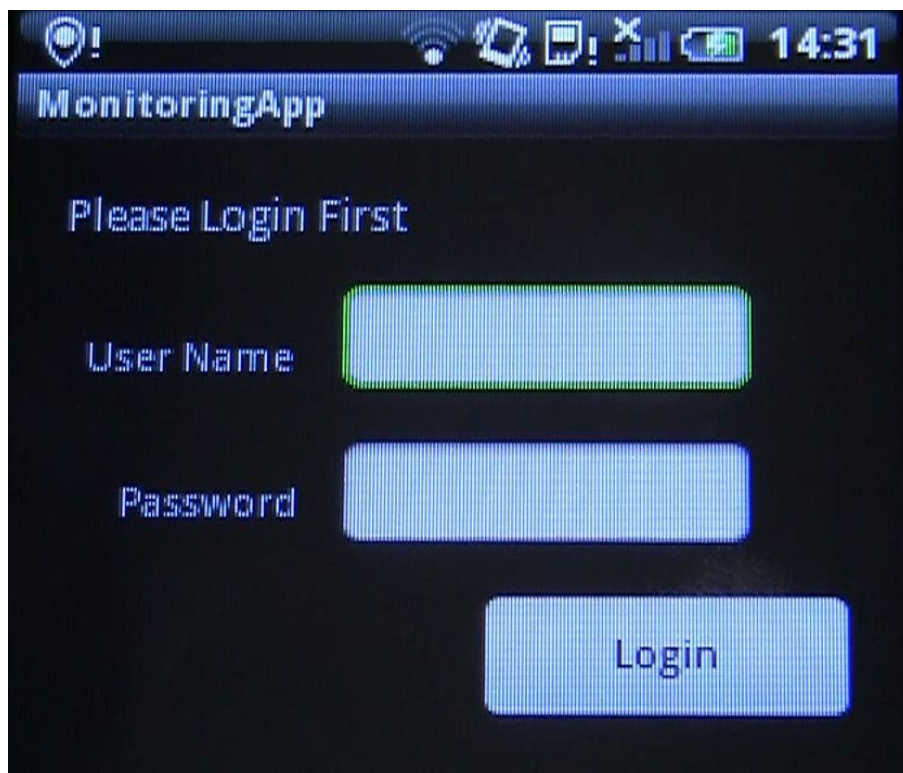
Kirjautumisnäkömä

Kirjautumisnäkömän pääasialliset tehtävät on alustaa globaali customhttpclient sekä kirjautua ABB:n etämonitorointiportaaliin, josta ohjelma hakee moottorin datan.

Kun käyttäjä tulee kirjautumissivulle, alkaa ohjelma alustaa globaalia customhttpclienttia, joka huolehtii sovelluksen yhteyksien ylläpitämisestä. Customhttpclient on oma luokkansa, jolle on rakennettu omat metodit sekä konstruktorit. Jos customhttpclienttia ei ole aikaisemmin alustettu, tehdään sille normaalit alustustoiminnot, jossa käytetään erillisiä EasySSLSocketFactory- sekä EasyX509TrustManager-luokkia kaikkien varmenteiden hyväksymiseen. Jos customhttpclient on jo aikaisemmin alustettu, palautetaan käyttäjälle tämä aikaisemmin

alustettu customhttpclient. Vanhan customhttpclientin palauttaminen tarkoittaa sitä, että luokka tulee alustaa ainoastaan kerran ja kaikki sovelluksen verkkoyhteydet hoidetaan saman httpclientin kautta. Kaikki sovelluksen näkymät voivat vain kutsua luokkaa sen omalla get-metodilla.

Sivulla on kaksi EditText-luokan tekstikenttää, johon käyttäjä kirjoittaa kirjautumiseen tarvittavat käyttäjätunnuksen ja salasanan (kuva 8). Kun käyttäjä painaa login-nappia, siirretään käyttäjätunnus sekä salasana nimi-arvoparilistaan ja se välitetään httppost-menetelmällä etämonitorointiportaalin kirjautumissivulle.



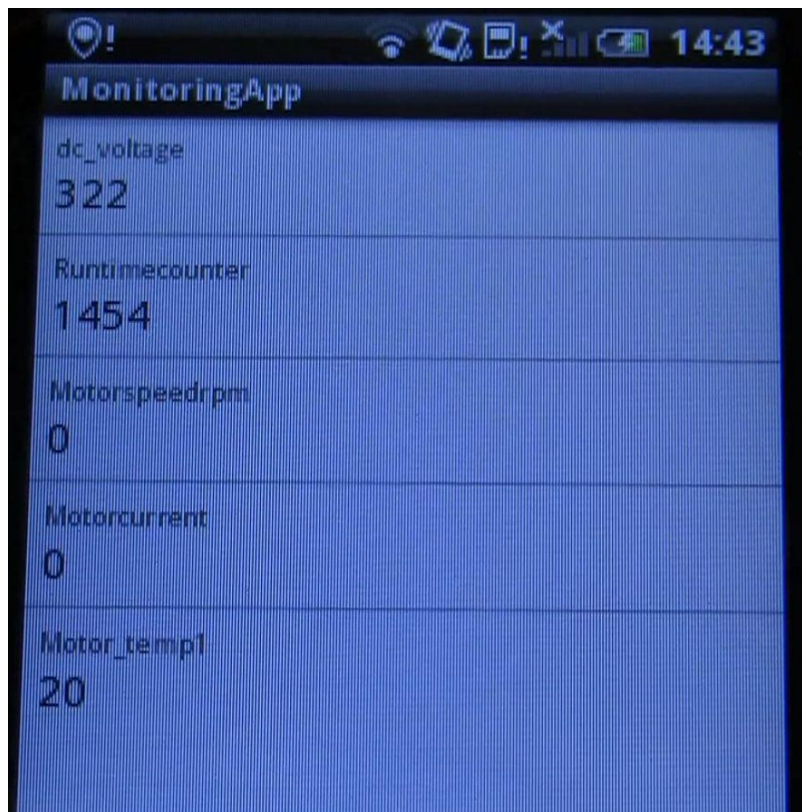
Kuva 8: Kirjautumisnäkyvä.

Palvelimen vastaus muutetaan yhdeksi merkkijonoksi käyttäen erillistä StringBuilder inputStreamToString -luokkaa. StringBuilder ottaa siis palvelimen vastauksena tulleen html koodin ja muuttaa sen yhdeksi merkkijonoksi. Saatu merkkijono sisältää koko sivuston html-koodin. Tästä merkkijonosta voidaan etsiä tietoa, onnistuiko kirjautuminen vai ei.

Jos kirjautuminen onnistui, löytyy sivuston html-koodin <title> tagista erillinen "Login" -teksti. Muussa tapauksessa kirjautuminen on epäonnistunut.

Monitorointinäkymä

Moottorilta tulleen datan näyttämiseen valittiin yksinkertainen listanäkymä (kuva 9) ja se toteutettiin Listview-luokan avulla. Listassa data näytetään allekkain nimi-arvo-pareina, joissa tulee ensin monitoroitavan parametrin nimi ja tämän alla itse arvo.



Kuva 9: Monitorointinäkymä.

Lista päivitetään Androidin oman Handler-luokan avulla aina minuutin välein. Handler ylläpitää ohjelman käskyjen ajoitusta, ja sen päätehtävät ovat ajoitettujen tehtävien suorittaminen taustalla sekä uuden tehtävän lisääminen ”tehtäväjonoon”. Kun käyttäjä ensin saapuu monitorointinäkymään, laitetaan listan päivitysoperaatio jonoon siten, että se suoritetaan minuutin päästä sivuille tulosta. Tämän jälkeen aina kun ohjelma käy päivittämässä listaa, lisätään uusi päivystehtävä ”tehtäväjonon” loppuun.

Käyttäjä voi siirtyä moottorin kontrollisivulle painamalla puhelimen valikko nappia, josta hän voi valita ”Control motor”-vaihtoehdon. Tämä käynnistää täysin uuden sivun johon käyttäjä ohjataan. Valikossa on myös erillinen logout-vaihtoehto, josta käyttäjä voi kirjautua ulos ohjelmasta ja palata takaisin kirjautumissivulle.

Kontrollinäkömää

Kontrollisivu sijaitsee dyndns-virtuaalipalvelimella ja itse sivun sisältö on erikseen määritelty SREA-01-asetusten kautta. Kontrollisivu on kokonaan javascriptia, joten kirjautumissivun yksinkertaistettua näkymää ei tässä tapauksessa ollut mahdollista toteuttaa.

Tästä syystä sivun näyttämiseen on käytetty Androidin WebView-luokkaa. Näkymästä on kuitenkin kokonaan poistettu selaimen normaalit toiminnot, kuten osoitteen vaihtaminen ja liikkuminen eteen/taakse-napeilla.

Virtuaalipalvelimella oleva sivu käyttää HTTP-yhteyttä, joten varmenteiden lataus ja hyväksyminen ei tässä tapauksessa ole tarpeellista.

Jotta käyttäjä pystyy muuttamaan moottorin asetuksia, täytyy hänen ensin kirjautua SREA-01-hallintatilaan. Kirjautumistunnukset on myös erikseen määritelty käyttäjälle SREA-01-asetusten kautta.

Kirjaututtuaan sisään hallintatilaan käyttäjä pystyy hallitsemaan moottorin toimintaa ja esimerkiksi käynnistämään taisammuttamaan moottorin. Kuvassa 10 näkyy moottorin kontrollisivu, josta arvoja voidaan muuttaa.



ABB

Logged in as: Administrator

Select page **Status Alarm Log Configuration Setup**

Overview

Drive control	
Description	Value
CONTROL	DISABLED <input type="button" value="v"/> set
01.01: Motor speed rpm	0.00
REFERENCE2[%]	<input type="text"/> set

Kuva 10: Moottorin ohjaussivu.

Kontrollisivun Control-arvolla ohjataan moottorin tilaa. Tätä arvoa vaihtamalla moottorin pysyy joko sammuttamaan tai käynnistämään. Moottoria käynnistäessä pitää käyttäjän ensin valikoida Control-alasvetovalikosta reset ja tämän jälkeen painaa Set-nappia. Kun moottori on palautettu alkutilaan, voi käyttäjä valikoida samasta valikosta Run-vaihtoehdon, joka käynnistää moottorin. Moottorin sammutus onnistuu suoraan Disabled-vaihtoehdolla.

Motor speed rpm-kohta näyttää käyttäjälle moottorin senhetkisen nopeuden. Motor speed rpm-kohdan alapuolella on Reference-kenttä, johon käyttäjä voi laittaa haluamansa moottorin nopeuden prosentteina. Jos käyttäjä laittaa Reference-kenttään arvon ja painaa Set-nappia, muuttuu moottorin nopeus käyttäjän antaman arvon mukaan.

7 Testaus

Koodia testattiin aina ensin SDK:n omassa emulaattorissa, ja jos siinä ei heti ilmennyt virheitä, siirrettiin koodi suoraan oikeaan päätelaitteeseen. Testauksessa käytettiin Android HTC Wildfire-puhelinta. Jos virhetilanteita syntyi puhelimesta, siirryttiin takaisin käyttämään SDK:n oman emulaattorin virheenjäljitintä (debugger).

Puhelimen testausta vaikeutti kuitenkin hieman Wildfiren ongelmat wifi-yhteyksien kanssa. Puhelin saattoi satunnaisesti katkaista wifi-yhteyden kokonaan, joka taas haittasi testausta ja tuotti virhetilanteita.

Testausta varten moottori säädettiin keräämään sekä lähettämään dataa minuutin välein, jotta varsinaisia tuloksia ei tarvitsisi odottaa niin kauaa. Monitoroitavaksi dataksi valittiin satunnaisia parametreja kuten moottorin tasavirata, ylläpitolaskuri, moottorin virta sekä moottorin lämpötila. Yksi erikseen valittu arvo oli moottorin pyörimisnopeus (rpm).

Moottorin pyörimisnopeus valittiin myös arvoksi, jota pystyttäisiin etänä muuttamaan. Tämä tehtiin sen takia, että moottorin pyörimisnopeus on helppo havainnoida suoraan moottorista ja se on muutenkin moottorin parametreista yksi selkeimmistä arvoista. Toinen etänä muutettava parametri on itse moottorin tila, eli moottorin pystyy käynnistämään tai sammuttamaan etänä.

Testauksen tarkoituksena oli varmistaa ohjelman toimivuus yleisimmissä tapauksissa.

8 Jatkokehitys

Jatkokehitystä ajatellen tärkeä osa olisi saada moottorin ohjaussivu yksinkertaisempaan geneeriseen muotoon. Virtuaalipalvelimella oleva sivu on tehty puhtaasti javascriptilla, joten sen näyttäminen yksinkertaisessa muodossa (kuten monitorointisivu) vaatii jatkokehitystä.

Monitorointinäkylässä olevan listan tietojen tallennus erilliseen tiedostoon olisi myös hyvä lisäys ohjelmaan. Tällä tavalla käyttäjä voi varastoida moottorilta tulevan datan myöhempää käyttöä varten.

HTTPS-yhteydessä ladattavat varmenteet tulisi myös käsitellä oikeaoppisesti ja varmistaa erikseen CA:n kautta.

Projektin alussa virtuaalipalvelin oli täysin ilmainen dyn.com:n kautta, mutta tämän jälkeen se on siirtynyt maksulliseksi. Mahdollisia vaihtoehtoja kontrollisivun ylläpitoon palvelimella kannattaa siis tutkia.

9 Yhteenveto

Tämän työn tarkoituksena oli selvittää, pystyykö etänä olevaa ABB-moottoria monitoroimaan sekä hallitsemaan etänä mobiilisovelluksen avulla. Lopputuloksena kehitettiin prototyyppi-Android-sovellus, jolla pystyttiin lukemaan moottorin dataa sekä muuttamaan sen arvoja.

Sovelluskehitys Android-laitteelle on yleisesti helppoa ja Android-ohjelmointirajapinta (API) on todella kattava. Usein ongelmatilanteissa onkin paras siirtyä suoraan Android Developers-sivuston API-osioon ja etsiä sieltä ratkaisuja luokkien käyttöön. Androidiin tarkoitettua koodia kirjoitetaan Java-kielellä, mutta Android kirjastot käyttävät myös C/C++ -kirjastoja, joten aikaisemmin C:tä käyttäneet kehittäjät voivat helposti oppia tuottamaan ohjelmistoja Androidille.

Yksi projektin haastavimmista osista oli ehdottomasti HTTPS-yhteyden luominen ABB:n etämonitorointiportaaliin. Ohjelmassa käytetty kaikkien SSL-varmenteiden hyväksyminen ei tosiaan ole ideaalinen tilanne, mutta prototyypin testausta varten riittävä.

Lähteet

- 1 Mikä taajuusmuuttaja on? 2008. Verkkodokumentti. ABB Suomeksi.
<<http://www.abb.fi/cawp/db0003db002698/d5b664f5dd909412c1257291003ef7cc.aspx>>. Päivitetty 23.6.2008. Luettu 1.5.2013
- 2 Smartphone. 2013. Verkkodokumentti. Wikipedia
<<http://en.wikipedia.org/wiki/Smartphone>>. Päivitetty 7.5.2013. Luettu 14.5.2013.
- 3 Google Play. 2013. Verkkodokumentti. Wikipedia.
<http://fi.wikipedia.org/wiki/Google_Play> Päivitetty 5.4.2013. Luettu 12.4.2013.
- 4 T-Mobile G1 – Full phone specifications. 2013. Verkkodokumentti. Gsmarena
<http://www.gsmarena.com/t_mobile_g1-2533.php> Luettu 10.5.2013.
- 5 Android versions comparison. 2013. Verkkodokumentti.
<<http://socialcompare.com/en/comparison/android-versions-comparison> >. Päivitetty 29.4.2013. Luettu 4.5.2013.
- 6 Sayed Y. Hashimi, Satya Komatineni, Dave MacLean. 2010 Pro Android 2, United States of America: Apress.
- 7 Reto, Meier 2010. Professional Android 2 Application Development. Indianapolis, Indiana: Wiley Publishing.
- 8 Android Developers Blog. 2011. Verkkodokumentti. Android's HTTP Clients
<<http://android-developers.blogspot.fi/2011/09/androids-http-clients.html>>. Päivitetty 29.11.2011. Luettu 20.3.2013.
- 9 ABB Drives, User's Manual Ethernet Adapter Module SREA-01. ABB, 2010.