

SATAKUNNAN AMMATTIKORKEAKOULU



Lauri Hartonen

Mika Impola

2007

MOBIILI RFID-LUKIJA

Tekniikka Rauma
Tietotekniikan koulutusohjelma

MOBIILI RFID-LUKIJA

Hartonen, Lauri & Impola, Mika

Satakunnan ammattikorkeakoulu

Tekniikka Rauma

Tietotekniikan koulutusohjelma

Kesäkuu 2007

Ohjaajat: Mikko Javanainen ja Kauko Peltonen

UDK: 621.39

Sivumäärä: 101

Avainsanat: ohjelmistokehitys, mobiililaitteet, tuotekehitys, elektroniikkalaitteet, etätunnistus

Tämän opinnäytetyön tavoitteena oli rakentaa prototyyppi mobiilista RFID-lukijasta. Tarkoituksena oli yhdistää RFID-lukija älykkääseen päätelaitteeseen ja kokonaisuudesta pyrittiin tekemään kätevä ja helppokäyttöinen. Laitteen käyttövaatimuksiksi määriteltiin kaksi asiaa: laitetta on pystyttävä käyttämään yhdellä kädellä ja lukuetaisyyden tulee olla standardien määrittelemissä rajoissa.

Helppokäyttöisen mobiilin RFID-lukijan tarve tuli esille tutkimusprojektissa. Projekti oli Satakunnan ammattikorkeakoulun ja Satakunnan sairaanhoitopiirin poikkitieteellinen yhteistyö, joka tutki yhtenä osa-alueenaan RFID-tekniikan käyttöönottoa Porissa Satakunnan sairaanhoitopiirin uuden päiväkirurgisen osaston tiloissa. RFID-tekniikkaa käytetään kasvavassa määrin myös teollisuuden logistisissa ratkaisuissa. Näin ollen mobiilille RFID-lukijalle löytyy sovelluskohteita muualtakin kuin terveydenhuollon piiristä.

Päätelaitteena käytettiin Nokia 770 Internet Tablettia. Päätelaitteeseen kiinnitettiin opinnäytetyön tuloksena syntynyt mobiili RFID-lukija nimeltään IDENTLE. IDENTLE koostuu OpenPCD-lukijasta, paristosta ja virransyöttöyksiköstä. Nokia 770 -päätelaitteelle ohjelmoitiin ohjelmisto, joka lukee RFID-tunnisteita ja esittää tunnisteen yksilöivään numeroon liitetyt tiedot käyttäjälle. Ohjelmisto noutaa tunnisteseen liitetyt tiedot palvelimen WebServices-rajapinnalta. Käytettävyyden parantamiseksi laite suunniteltiin kätevästi.

Laitteen toiminnallisuus testattiin testiympäristössä, jolla pyrittiin jäljittelemään mahdollista käyttöympäristöä sairaanhoidon piiristä. Testiympäristö osoitti laitteen toiminnan ja havainnollisti, miten mobiilia RFID-lukijaa voidaan mahdollisesti käyttää osana sairaalan tietojärjestelmää.

Lopputuloksena syntyi perusominaisuuksiltaan toimiva mobiili RFID-lukija, jota voidaan käyttää ideoinnin ja tuotekehityksen pohjana esimerkiksi osana laajempaa RFID-tekniikkaan pohjautuvaa järjestelmää.

MOBILE RFID READER

Hartonen, Lauri & Impola, Mika

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Tutors: Mikko Javanainen and Kauko Peltonen

June 2007

UDC: 621.39

Keywords: software development, mobile devices, research and development, electronic appliances, RFID

The purpose of this thesis was to develop a prototype of a mobile RFID reader. This was to be achieved by combining a RFID reader with an intelligent terminal into an easy to handle and useable device. Two main operational requirements were defined: it should useable by one hand and the operating distance should be within the requirements of the standard in use.

A need for an easy to use mobile RFID reader came up in a student research project in 2005. The project, which was a collaboration of Satakunta University of Applied Sciences and Satakunta Hospital District, studied the possibility to use RFID technology in the daytime surgical department at Pori. The overall interest and use of RFID technology in solving logistical problems has spread to different fields of industry as well. This indicates a growing demand for a mobile RFID reader outside the healthcare sector.

The terminal used in this project was Nokia 770 Internet Tablet. Hardware, consisting of an OpenPCD reader, a battery and a power injector PCB, was enclosed in an easy to handle casing to improve the usability. This formed a device called IDENTLE, which was attached to the Nokia 770 Internet Tablet. The terminal runs software that has a user interface and controls the RFID reader. As its main functions the IDENTLE reads RFID tags and presents the data linked with the identification number of the tag. The software retrieves the linked data from a server based on the collected identification numbers.

The result was a working mobile RFID reader, which was put through series of tests. A test environment was built to test and demonstrate the feasibility of the mobile RFID reader in a healthcare environment. The tests proved the capabilities of the device and showed how it could function as a complementary part of the information system of a hospital. IDENTLE could be used as a foundation for the future product development projects.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENTEET	7
1 JOHDANTO	8
2 RADIO FREQUENCY IDENTIFICATION	10
2.1 Tekniikka yleisesti	10
2.2 Tagi	11
2.3 Lukija	13
2.4 Kommunikointi	13
2.4.1 Passiivisten tagien toimintataajuudet ja -etäisyydet	15
2.5 Standardit	16
2.5.1 Mifare	16
2.5.2 ISO 14443	17
2.5.3 ISO 15693	17
2.6 Tietoturva	18
3 NOKIA 770	20
3.1 Ominaisuudet	20
3.1.1 Käytettävyys	21
3.1.2 Liitettävyys	21
3.2 Ohjelmisto	22
3.3 Ohjelmointiympäristö	23
3.3.1 MAEMO	23
3.3.2 Hildon application framework	24
4 OpenPCD – RFID-LUKIJA	26
4.1 Bitmanufaktur lyhyesti	26
4.2 OpenPCD-lukija	26
4.2.1 Tekniset ominaisuudet	27
4.2.2 Ohjelmisto	28
4.3 Lukijan toiminta	29

4.3.1 Librfid	30
5 TUTKIMUSONGELMA	32
5.1 Projektin taustaa	32
5.2 Projektin esittely	32
5.2.1 Laitteistovaatimukset	34
5.2.2 Ohjelmiston toiminnallinen määrittely	35
6 IDENTLE	37
6.1 Nokia 770 & OpenPCD lukija	37
6.1.1 Yhteensopivuus	37
6.1.3 Ohjelmistovaatimukset	39
6.2 USB Power Injector	39
6.2.1 Piirikaavio	40
6.2.2 Toteutus	42
6.2.3 USB power injectorin testaus	43
6.2.4 IDENTLE:n käyttöaika	45
6.3 Tuotteistaminen	46
6.3.1 Vaatimukset	47
6.3.2 Toteutus	47
6.4 Laitteistotestaus	52
6.4.1 Käytännön lukuetaisyys	52
6.4.2 Kotelon testaus	52
7 IDENTLEN OHJELMISTO	54
7.1 Nokia 770:n ohjelmiston kehitystyökalut	54
7.1.1 Eclipse	54
7.1.2 Scratchbox	55
7.1.3 Gazpacho	56
7.2 Suunnittelu	56
7.2.1 Arkkitehtuuri	57
7.3 Moduulit	59
7.3.1 RFID communication	60
7.3.2 WebServices	61

7.3.3 User Interface.....	62
7.3.4 IDENTLE.....	62
7.4 Toteutus.....	63
7.4.1 RFID communication.....	64
7.4.2 WebServices	66
7.4.3 User Interface.....	67
7.4.4 IDENTLE.....	69
7.5 Ohjelmistotestaus.....	70
8 KOKONAISTESTAUS	72
8.1 Testiskenaario	72
8.2 Testaustulosten tarkastelu	73
9 JATKOKEHITYS.....	74
9.1 IDENTLE.....	74
9.2 Aktiivitagisovellukset	76
10 YHTEENVETO	77
LÄHTEET.....	79
KUVA-, TAULUKKO- JA KUVIOLUETTELO	82
LIITTEET	84

LYHENTEET

ASK	Amplitude Shift Keying
CDT	C/C++ Development Toolkit
EEPROM	Electrically Erasable Programmable Read-Only Memory
FDX	Full Duplex tiedonsiirto
FSK	Frequency Shift Keying
GTK	Gimp Toolkit
HAF	Hildon Application framework
HF	High Frequency
ISO	International Standardization Organization
PCB	Printed Circuit Board
PCD	Proximity Coupling Devices
PICC	Proximity Integrated Circuit Card
PSK	Phase Shift Keying
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer
SAMK	Satakunnan ammattikorkeakoulu
SOAP	Simple Object Access Protocol
SRAM	Static Random Access Memory
Tagi	Transponderi, RFID-tunniste
UHF	Ultra High Frequency
WSDL	Web Services Description Language

1 JOHDANTO

Radio Frequency Identification, lyhyemmin RFID, on monille aloille sovellettu langaton tunnistustekniikka. Viime aikoina RFID on noussut mm. logistiikka- ja terveydenhuoltoalojen kiinnostuksen huipulle. Tekniikkaa sovelletaan moneen eri tarkoitukseen, vaikka se ei itsessään ole monille tuttu. Edellä mainitut seikat tekevät tekniikasta kiinnostavan, ja sille löytynevät tulevaisuudessa laajat markkinat.

SAMK:n ja Satakunnan sairaanhoitopiirin yhteinen projekti nimeltään Päiväkirurgiapoliklinikan logistiikka Satakunnan sairaanhoitopiirissä tutki yhtenä osa-alueenaan RFID-tekniikan käyttöönottoa Porin uuden päiväkirurgisen osaston tiloissa. Tutkimuksessa ilmeni mahdollinen tarve mobiilipäätelaitteelle, joka kulkisi esim. hoitajan mukana ja kuljettaisi ajantasaista potilasdataa suoraan tilanteiden ytimeen ja joka olisi silti luotettava. Kannettavalle tiedonkeruu, -hallinta sekä -verifiointilaitteelle voi löytyä tarvetta myös toimitusketjunhallintajärjestelmän useilla eri osa-alueilla.

Projektin tarkoituksena on tuottaa toimiva mobiili RFID-lukijaratkaisu. Toimivuus osoitetaan rakentamalla testiympäristö, jolla pyritään jäljittelemään mahdollista käyttötilannetta. Tavoitteena on, että tuotteen toiminnallisuutta voidaan muokata eri käyttötarkoituksiin ja -kohteisiin asiakkaan toiveiden mukaisesti. Testiympäristö rakennetaan demonstraatiotarkoituksessa oppilaitoksemme tiloihin.

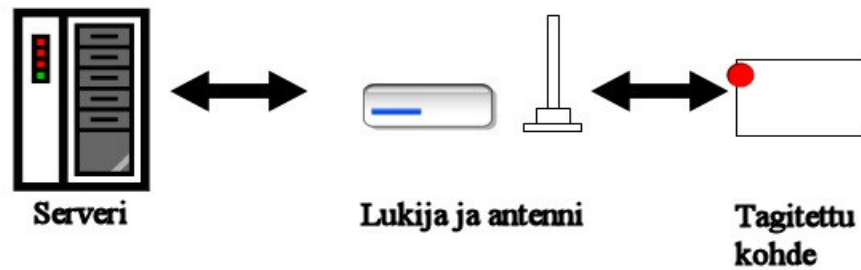
Insinööriä aloitettiin kartoittamalla mahdolliset toteutustavat ja -teknologiat. Laitetta ei lähdetty kehittämään tyhjältä pöydältä, vaan niin laitteiston kuin ohjelmistonkin osalta pyrittiin ajan säästämiseksi tukeutumaan valmiisiin komponentteihin, kuten saksalaisvalmisteiseen RFID-lukijaan ja sen ohjelmistokirjastoihin. Laitteen ohjelmisto, elektronikka ja kotelointi suunniteltiin ja testattiin kokonaisuutta silmällä pitäen. Testiympäristössä testattiin laitteen toimivuus ja havainnollistettiin mobiilin RFID-lukijan liikeidea.

Opinnäytetyö tehtiin parityönä. Mika Impola vastasi projektin testiympäristön tietokannan luonnista ja ohjelmistojen toteutuksesta sekä palvelimelle että päätelaitteelle. Lauri Hartosen vastuualueeseen kuuluivat laitteistosuunnittelu, -kehitys ja toteutus sekä niiden testaus ja testausraportointi. Molemmat vastasivat omien vastuualueidensa raportoinnista. Yhteisinä vastuualueina olivat testiympäristön ja ohjelmiston määrittely.

2 RADIO FREQUENCY IDENTIFICATION

2.1 Tekniikka yleisesti

Radio Frequency Identification, lyhyemmin RFID, merkitsee mitä tahansa tunnistussysteemiä, johon integroitu elektroninen väline käyttää radiotaajuuksissa tai magneettikentässä tapahtuvia muutoksia kommunikoidakseen. Kaksi oleellista tekijää RFID-tekniikassa ovat *transponderi* eli *tagi*, joka on seurattavan kohteen yksilöivä tunniste, ja *readeri* eli *lukija*, joka tunnistaa RFID-tagien läsnäolon sekä niiden sisällön rajatulla alueella. Lukija voi välittää tiedon RFID-tagien tunnistamisesta eteenpäin järjestelmälle, joka sitä osaa käsitellä. Perusidea on esitelty Kuvassa 1. (Glover & Bhatt 2006, 1-2.)



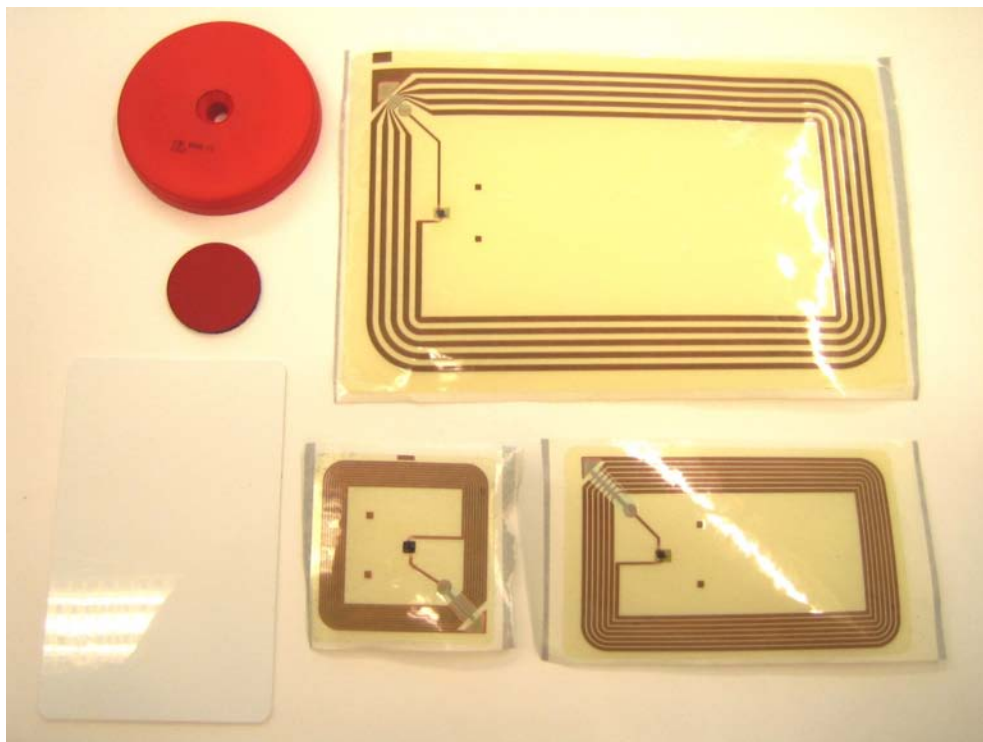
Kuva 1. RFID-systeemi

Tunnistustilanteessa lukijan sekä tagitetun kohteen välillä lukija lähettää signaalia ennalta määrätyllä taajuudella, jonka tähän taajuudelle viritetyt tagit vastaanottavat. Signaalin vastaanottamat tagit lähettävät tiedon läsnäolostaan takaisin lukijalle. Käytössä oleva kommunikointistandardi, tagin tyyppi ja lukialueella olevien tagien lukumäärä vaikuttavat siihen, miten tagi käyttäytyy, lähettääkö se mikrosirulleen tallennetun datan heti ensikontaktin jälkeen vai jääkö se odottamaan vuoroaan. Lukija tunnistaa lukialueella olevan tagin tunnistedatan perusteella. Itse tunnistus voi tapahtua lukijassa itsessään tai serveril-

lä, johon tageilta poimitut tiedot lähetetään. (Glover & Bhatt 2006, 21-52; Rinta-Runsala & Tallgren 2004.)

2.2 Tagi

Tagin tehtävä yksinkertaisuudessaan on liittää tieto fyysisesti kohteesta kohteeseen. Jokaisella tagilla on jokin sisäinen mekanismi datan säilömiseksi ja tyyli kommunikoida data eteenpäin. Kommunikointi tapahtuu eri tekniikoin riippuen tagista, mutta jokaisella tagilla on kela tai jokin muu antenni, kuten kuvassa 2 nähdään. (Glover & Bhatt 2006, 55.)



Kuva 2. RFID-tunnisteita

Vain tagin kiinnitettävyyys ja luettavuus ovat universaaleja tehtäviä, mutta kuten kommunikointi, tagin tarjoamat palvelut varioivat tagin ominaisuuksien mukaan. Tagille voidaan

kirjoittaa uutta dataa, se voidaan deaktivoida ns. tappokoodilla sekä se osaa osallistua salattuun kommunikointiin ja vaatia autentikointia. (Glover & Bhatt 2006, 55-56.)

Tagit luokitellaan yleensä niiden teholähteiden mukaan. Tämä määrää myös pitkälti tagin hinnan ja kestävyuden. *Passiivinen* tagi ottaa käyttövirtansa lukijan lähettämästä signaalista. *Semi-passiivinen* tagi ottaa lähetystehonsa lukijan signaalista, mutta sillä on oma akku esim. muistin ylläpitämiseen. *Aktiivinen* tagi ottaa tehonsa omasta akusta, jonka virtaa se käyttää kommunikoimiseen ja prosessorin, muistin sekä mahdollisten sensoreiden ylläpitämiseen. Ns. *two-way* tagit mahdollistavat kaiken yllämainitun lisäksi kommunikoinnin tagien välillä ilman lukijan tukea. (Glover & Bhatt 2006, 58.)

Tagien ominaisuuskirjo mahdollistaa niiden käytön vaihtelevissa ympäristöissä, joten myös fyysiset ominaisuudet vaihtelevat käyttötarkoituksen mukaan. Ulkoiselta olemukseltaan tageja on monenlaisia, kuten

- PVC- tai muovikielkoja, jotka ovat kestäviä ja uudelleenkäytettäviä
- luottokortin muotoisia
- liimapintaisia tarratageja
- pieniä, suurempiin kokonaisuuksiin upotetut tageja, kuten kulkuavaimet
- syövyttävän ympäristön kestäviä tageja lasikapseleissa. (Glover & Bhatt 2006, 57.)

RFID-tagit voivat häiritä toisiaan. Jos lukijan lukuetaisyydellä on useampi samaan taajuuteen reagoiva tagi, lukija ei välttämättä kykene purkamaan ja erottelemaan vastaan otettuja signaaleja. Tietyissä sovelluksissa on oleellista lukea useampi kuin yksi tagi pienessä ajassa. Näissä sovelluksissa tagien tulee tukea joko törmäyksenestoprotokollaa (anticollision) tai vuoronvalintaprotokollaa (singulation). Törmäyksenestoprotokolla säättää ja ajoittaa tagit vastaamaan vuorollaan kuitenkin yksilöimättä niitä erikseen. Vuoronvalintaprotokolla pyrkii yksilöimään tagit ja kutsumalla joko yksilö ID:n perusteella tai jakamalla tageille vuoronumerot. (Garfinkel & Rosenberg, 2005, 20; Glover & Bhatt 2006, 78.)

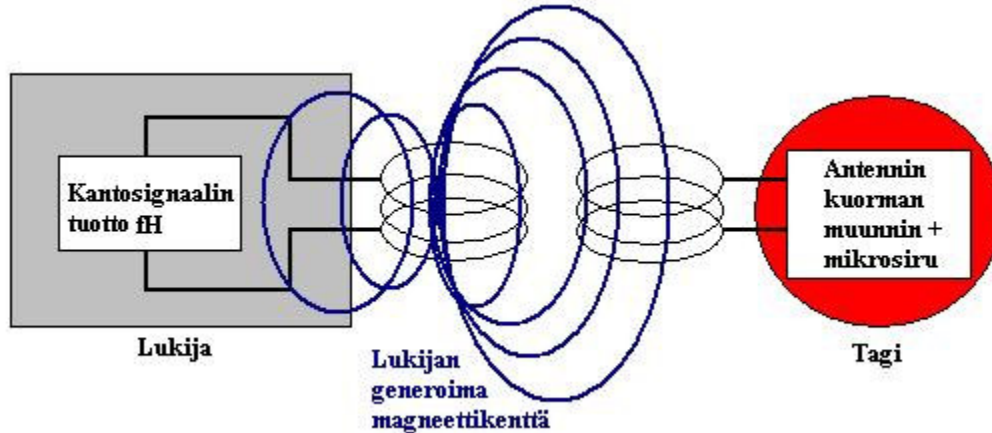
2.3 Lukija

Lukijan tehtävä RFID-systeemissä on luoda tageille kommunikointiin vaadittava virta, kerätä lähetetty data ja toimia porttina systeemin ulkopuoliseen verkkoon. Koska lukija kommunikoi radiotaajuuksilla, täytyy sillä olla yksi tai useampi antenni joko kiinni itsessään tai lyhyen matkan päässä. Rajoittava tekijä antennien määrässä on kaapeloinnista aiheutuva signaalihäviö. Kommunikointi ulkopuoliseen verkkoon vaatii esim. sarjakaapeliliitännän (RC 232) tietokoneeseen tai mahdollisuuden RJ45-kaapeliliitintään verkkoyhteyttä varten. Liitintämahdollisuudet vaihtelevat lukijakohtaisesti, ja myös langatonta tekniikkaa, kuten Bluetooth-tekniikkaa sekä IEEE 802.11 -standardeja, tuetaan. Datan keräämiseen ja kommunikointiprotokollien hallintaan tarvitaan mikrokontrolleri, joka sijaitsee joko itse lukijalla tai lukijaan yhdistetyllä päätteellä. (Glover & Bhatt 2006, 107-109.)

Riippuen RFID-tekniikan sovelluksesta, lukija on joko aina päällä tai ulkoisesti päälle kytkettävissä. Kulunvalvonnassa, jossa tagin lukialueelle osuminen on satunnaista, lukijan tulee olla päällä jatkuvasti. Mutta esimerkiksi käsilaitteissa on tavallisesti kytkin, jolla lukija kytketään päälle. Näin menetellen säästetään kannettavien laitteiden virtalähteitä.

2.4 Kommunikointi

Normaalisti radiokommunikointiin tarvitaan lähetin ja vastaanotin. RFID-tekniikassa tästä on hyötyä lähinnä aktiivitagi-sovelluksilla pitkillä etäisyyksillä. Tästä poiketen passiivitagi-sovelluksissa lukija ja tagi ”pariutetaan” sähkömagneettista induktiota hyväksikäyttäen. Menetelmää kutsutaan sähkömagneettiseksi kytkennäksi (coupling) ja sen käyttö RFID-tekniikassa nähdään kuvassa 3.



Kuva 3. Sähkömagneettinen induktio (Finkenzeller 2006).

Lukijan ja tagin antennit ovat molemmat muotoiltu ohuesta johtimesta kelan muotoiseksi. Kuvan 3 kaltaisessa tapauksessa, lukijan kelassa oleva sähkövirta muodostaa magneettikentän ympärilleen. Tagin kelaan indusoituu jännite, joka tasasuunnataan tagin käyttöjännitteeksi. Lukija kommunikoi tagin kanssa moduloimalla luomaansa kantaaltoa muuttaen kantaallon amplitudia, taajuutta tai vaihetta riippuen käytössä olevasta tekniikasta.

Tagin joutuessa lukijan luomaan magneettikenttään huomataan lukijan antennissa jännitetasonmuutos. Joko lukijan pyynnöstä tai automaattisesti, tagi vastaa lukijalle lukijan kantaallosta eroavalla taajuudella. Eroava taajuus muodostuu, kun tagi muuttaa antenninsa kuormitusta nopealla taajuudella f_1 . Lukija havaitsee tagin luoman $\pm f_1$ -taajuudet kantaallonsa ympärillä, joita kutsutaan radiotermein apukantaalloiksi. Kuvassa 8, joka on sivulla 31, nämä apukantaalloit nähdään f_0 kantaallon ympärillä. Tagi moduloi yleensä toiseen näistä apukantaalloista oman datansa. Muutoksia ohjataan tagilla olevan datan mukaan, joko ASK-, FSK- tai PSK-modulaatiolla. Kuvan 8 tapauksessa, käytetään FSK-modulaatiota ja taajuuden muutokset ovat $f_1 \pm 106$ kHz. Nämä taajuuden muutokset ovat nähtävissä lukijan antennissa apukantaallon amplitudin muutoksina. (Garfinkel & Rosenberg 2005, 23-24; Finkenzeller 2006; Meriac, M. 2006.)

2.4.1 Passiivisten tagien toimintataajuudet ja -etäisyydet

Jotta passiivinen tagi pystytään lukemaan, tulee sen saada riittävä määrä tehoa pyörittämään elektroniikkaansa ja luomaan paluusignaali lukijalle. RFID-systeemin käyttöön vaikuttaa rajoittavasti joko tagille saatavilla oleva teho tai lukijan herkkyys havaita tagien lähetyksiä. RFID-järjestelmien kehityksessä on rajoittavana tekijänä tagien hinta, joten lukijan herkkyyttä on pyritty tehostamaan sen sijaan, että tagiin lisättäisiin kallista tekniikkaa. Lukuetaisyyttä pystytään kasvattamaan suurentamalla lukijan antennia. Tällä tavoin lukija pystyy keräämään enemmän tehoa tagilta. (Garfinkel, S. & Rosenberg, B. 2005, 20-25.) Taulukossa 1 ovat listattuina RFID-tekniikassa käytetyt taajuusalueet.

Taulukko 1. RFID-systeemien toimintataajuudet ja -etäisyydet (Glover & Bhatt 2006, 59.)

Nimi	Taajuusalue	ISM taajuudet	Maksimi lukuetaisyys	Käyttökohde esim.
LF	30 – 300 kHz	< 135 kHz	50 cm	Lemmikki- sekä lyhyen matkan nestepitoisen kohteen tunnistus
HF	3 – 30 MHz	6.78 MHz 13.56 MHz 27.125 MHz 40.680 MHz	3 m	Rakennusten ja tilojen kulunvalvonta
UHF	300 MHz – 3 Ghz	433.920 MHz 869 MHz 915 MHz	9 m	Laatikot ja kuljetuslavat
Microwave	> 3 GHz	2.45 GHz 5.8 GHz 24.125 GHz	> 10 m	Ajoneuvotunnistus (korkeat nopeudet)

Toimintataajuus tarkoittaa sähkömagneettista taajuutta, jolla RFID-systeemit toimivat. Koska RFID-järjestelmä lähettää sähkömagneettista kenttää, on se siis radiolaitte. RFID-systeemit eivät saa häiritä muita toimivia radiolaitteita. Häirinnän estämiseksi on määritelty erityiset ISM-taajuudet (Industrial-Scientific-Medical), jotka ovat alun perin tarkoitettu käytettäväksi teollisuus-, tieteis- sekä terveydenhuoltosovelluksissa tai muissa lyhyen kantaman sovelluksissa. (Glover & Bhatt 2006, 57; Finkenzeller 2006.)

Kuten yleensä radiosignaalien kanssa, ympäristötekijät vaikuttavat suuresti RFID-systeemin lukuetaisyyskysymyksiin. Erityisesti niihin vaikuttavat lukijan ja tagin välissä olevat materiaalit. HF- ja UHF-taajuuksilla tapahtuva kommunikointi häiriintyy helposti vedestä ja metallista. Esimerkiksi kosteat materiaalit eristävät merkittävästi signaalia ja voivat katkaista yhteyden. (Garfinkel & Rosenberg 2005, 26.)

2.5 Standardit

Maailmanlaajuisista standardointielintä ei ole, vaan standardointi on tähän asti tapahtunut aluekohtaisesti. Tämä seikka hidastaa tekniikan yleistymistä sekä yhteisten toimintaprotokollien kehittymistä. Euroopassa RFID-standardoinnista huolehtivat ISO-, ERO-, CEPT- ja ETSI-järjestöt. LF- (low frequency) ja HF- (high frequency) taajuudet ovat ainoat, joita saa käyttää maailmanlaajuisesti ilman lisenssiä. Alaluvuissa on esitelty oleellimmat RFID standardit. (Wikipedia 2007.)

2.5.1 Mifare

Mifare on rajoitetun lukuetaisyyden sirutyypin, jota käytetään kontaktittomissa älykortteissa. Mifare pohjautuu ISO 14443 -standardiin ja on Philipsin omistama. Mifare-kortti toimii 13.56 MHz:n taajuudella. Tiedon siirto tapahtuu 106kbit/s ja toimintaetaisyys rajoittuu 100 millimetriin. Mifare Standard-kortilla on yhden tai neljän kilotavun EEPROM muisti, joka käydään tarkemmin läpi kohdassa 7.4.1. Keskeisimmät käyttökohteet ovat julkisen liikenteen maksujärjestelmät sekä kulunvalvonta. Käyttökohteiden vuoksi Mifaressa on keskitytty erityisesti käyttöturvallisuuteen. Sekä lukijan että kortin puoleiset haaste-autentikoinnit ja data salaaminen varmistavat, ettei tietoa käytetä väärin. (RFIDLab Finland 2007; NXP Semiconductors 2007.)

2.5.2 ISO 14443

ISO 14443 -standardi määrittelee 13,56 MHz:n taajuudella toimivien PICC-sirukorttien ja PCD-lukijalaitteiden fyysiset ominaisuudet, radiotaajuusrajan, yhteyden hallinnan sekä tiedonsiirtoprotokollat. Standardin mukaan korttien tulisi toimia 0 °C - +50 °C lämpötilassa. ISO 14443 -standardi on jaoteltu kahteen osaan, 14443 A:han ja 14443 B:hen. Jaottelu perustuu magneettikentän modulointitavan, koodauksen, sekä törmäyksenhallinnan osalta. ISO 14443 -standardi rajoittaa lukuetaisyyden alla 10 senttimetriin. (Bashan, O.)

2.5.3 ISO 15693

ISO 15693 -standardia ei suunniteltu alun perin yksilötason tunnistamiseen, kuten esim. kirjastoihin, joissa sitä kuitenkin käytetään yleisesti. ISO 15693 -standardi suunniteltiin toimitusketjun (supply chain) sovelluksiin. Standardi kuvaa RFID-korttien fyysiset ominaisuudet, ilmaitse tapahtuvan yhdistämisen, kommunikoinnin sekä yhteyden hallinnan menetelmät. (Garfinkel & Rosenberg 2005, 233.)

Fyysisiltä ominaisuuksiltaan ISO 15693 -standardin mukainen RFID-kortti on pankkikortin kokoinen ja koostuu mikropiiristä sekä antennista. Se toimii 13.56 MHz:n taajuudella ja saa käyttövirtansa induktiivisesti lukijan magneettikentästä. Standardi määrittelee maksimi toimintaetäisyydekseen 1-1.5 metriä. Jos lukijan lukuetaisyydellä on useampi tagi, standardi määrittelee tietyt törmäyksenestön (anticollision) menettelytavat. (Glover & Bhatt 2006, 74; Wikipedia. 2007b.)

2.6 Tietoturva

RFID-tekniikan tietoturvan ongelmia ovat mm. RFID-tagien luvaton jäljittäminen ja lukeminen, RFID-tagin taikka lukijan kloonaminen ja tagin ja lukijan välisen kommunikoinnin salakuuntelu. (Glover & Bhatt 2006, 197–214.)

Tagin ja lukijan välisen kommunikoinnin salakuuntelun estämiseksi voidaan kommunikointi salata. Tämä kuitenkin edellyttää tagilta lisä-älyä, mikä nostaa tagin valmistuskustannuksia. Salakuuntelua rajoittaa myös tagin lyhyt lukuetaisyys. Tagin lukuetaisyyden ollessa lyhyt ovat myös lukijan ja tagin lähettämät signaalit heikkoja. Heikkojen signaalien salakuuntelemiseksi tulee salakuuntelulaitteen olla niin lähellä tagia, että se hankaloittaa luvaton lukijan ja tagin välisen kommunikoinnin kuuntelua.

Lukijan tai tagin kloonamiseen liittyvät uhat voidaan joissakin tapauksissa kiertää muuttuvilla autentikointimenetelmillä. Muuttuva autentikointimenetelmä voisi olla esimerkiksi sellainen, jossa tagin autentikointivastaus riippuu lukijan antamasta avaimesta. Tagi siis sisältäisi algoritmin, joka laskee autentikointivastauksen riippuen lukijan antamasta avaimesta. Tällöin pelkkä tagin kommunikointiviestien kloonaminen ei riitä, koska autentikointivaiheessa tagin vastaus on aina erilainen riippuen lukijan antamasta avaimesta. Kloonamisen lisäksi pitäisi murtaa myös autentikointi algoritmi. (Glover & Bhatt 2006, 211.)

Tagin luvaton jäljittäminen ja luvaton lukeminen pystytään estämään esimerkiksi sijoittamalla tagi Faradayn häkkiin, joka estää tagin lukemiseen käytettävien radioaaltojen läpipääsyn. Tagi voidaan myös haluttaessa tuhota kohdistamalla siihen voimakkaita elektromagneettisia pulsseja esim. mikroaaltouunilla. Suurta apua tagin luvattomaan lukemiseen tuo myös se, että katkaisee tai murtaa tagin antennin. Tällöin tagin lukuetaisyys pienenee huomattaavasti. Nämä toimenpiteet eivät tietenkään ole kovin soveliaita eivätkä tule kysymykseen, mikäli tagia vielä pitäisi käyttää hyödyksi. Nämä ratkaisut ovat enemmänkin käyttäjän tapoja torjua luvaton tagin lukua. Järjestelmän suunnittelijan puoleltakin asialle voidaan jotakin tehdä. Voidaan käyttää tagia, joka ei lähetä mitään tie-

toa ennen kuin lukija on lähettänyt oikeanlaisen avaimen eli tunnistanut itsensä tagille. Tagin tiedon lukeminen voi myös edellyttää salasanan käyttöä. (Glover & Bhatt 2006.)

RFID voi väärin käytettynä olla yksityisyyttä pahastikin loukkaava teknologia. Oikein käytettynä sen yksityisyydelle aiheuttamat uhat ovat kuitenkin minimaalisia hyötyihin verrattuna.

3 NOKIA 770

3.1 Ominaisuudet

Nokia esittelee tuotteensa Internet-sivuillaan oheisesti: ” Nokia 770 Internet Tablet ja laajakaistainen Wi-Fi-yhteys on kaikki mitä tarvitset langattomaan internetin käyttöön. Voit suoratoistaa tiedostoja, kuunnella internet-radiokanavia, lukea uutisia tai toistaa lempimusiikkiasi ja -videoitasi. Se kulkee vaivattomasti mukana, näyttää tyylikkäältä ja soveltuu erityisen hyvin verkkosisältöjen katseluun, kiitos äärimmäisen terävän laajakuvanäytön.”(Nokia 2007.) Nokia 770 on Internetin selailuun suunniteltu laite, joka ottaa yhteyden Internetiin joko 802.11b/g WLANin kautta tai puhelimen kautta Bluetooth-tekniikan välityksellä (Littler 2006). Laite ei siis ole puhelin, vaan mm. Internetin selailuun tarkoitettu multimedialaite, tästä nimi Internet Tablet.



Kuva 4. Nokia 770 Internet Tablet

Laite koostuu ulkoisesti kahdesta osasta; itse laitteesta ja sen liu'utettavasta suojakuoresta. Laitteen mitat ovat 141 x 79 x 19 mm (pituus x leveys x paksuus), ilman suojakuorta 135 x 78 x 14 mm, ja se painaa 230 grammaa. Käyttölämpötilaksi Nokia ilmoittaa -10°C - +55°C. Oletuksena muistiominaisuuksiltaan käyttäjällä on käytettävissä 64Mt flash-muistia sekä 64Mt:n RS-MMC – muistikortti. (Nokia 2007.)

3.1.1 Käytettävyys

Pääfunktioitaan eli internetin selailua Nokia 770 tukee suurella 65 536 värin (800 x 480 pikseliä) kosketusnäytöllä, jota kontrolloidaan joko mukana tulevalla kynällä tai sormella. Toiminta-ajaksi Nokia ilmoittaa 1500 milliampeeritunnin akulla selausajaksi kolme tuntia ja valmiusajaksi seitsemän vuorokautta. Vaikka akku on kapasiteetiltaan suuri, kookas näyttö vie paljon tehoja. Näytön hallintamahdollisuudet, kuten kirkkauden säätö sekä näytön itsesammutus, ovat oleellisia käyttöajan optimoijia. Kuitenkaan näytön automaattista kirkkauden ja kontrastin säätöominaisuutta ei tässä laitteessa ole, kuten esimerkiksi Nokian E61 mallissa, jossa tämä niin sanottu näytön laatu säätyy ympäristön valovoimakkuuden mukaan automaattisesti.

Nokia 770:n tukemat tiedostomuodot ovat yleisimmät internetistä löytyvät formaatit. Näistä oleellisimmat ovat ääniformaateista MP3, WAV ja WMA sekä videoformaateista AVI, MPEG-1 ja MPEG-4.

3.1.2 Liitettävyys

Kuten jo edellä selvitettiin, Nokia 770 ei ole puhelin. Tällä tarkoitetaan siis sitä, ettei se ole kykeneväinen muodostamaan yhteyttä GSM- tai 3G-verkkoon. Laite tarvitsee toimi-

akseen selaimena ulkoisen yhteyden. Yhteys hoituu joko 802.11b/g eli WLAN-tekniikan avulla tai yhteydellä puhelimeen Bluetooth-tekniikkaa käyttämällä.

Tietokoneeseen 770:n saa liitettyä Bluetoothin kautta tai DKE-2 USB-kaapelilla. Liitettäessä Nokia 770 toimii ns. slave-tilassa, eli se ei ole oletuksena kykenevä hallinnoimaan lisälaitteita, kuten esimerkiksi USB-näppäimistöä.

3.2 Ohjelmisto

Nokia 770 Internet Tabletin mukana tulee melko kattava määrä ohjelmia internetin peruskäyttäjälle. Seuraavassa on lista pääohjelmista, jotka ovat valmiiksi asennettuna Nokia 770:ssa:

- Nettiselain, joka pohjautuu, suosittuun open source selaimen, Operaan. Selain tukee myöskin Flashiä
- Internet puhelu -ohjelmisto
- Pikaviestin
- Sähköposti
- Internet radio
- RSS uutissyötteiden lukija
- Ohjelmistot äänen, videon, kuvan ja pdf-tiedostojen katselua ja toistoa varten
- Ohjelmien hallintaohjelmisto, joka on tuttu toimintaperiaatteiltaan monesta Linux jakelusta. (Nokia 2005a.)

Näiden lisäksi Nokia 770 sisältää monia pienempiä ja käytännöllisiä ohjelmia, kuten kellon, laskimen ja useita pelejä.

Koska Nokia 770:n ohjelmointiympäristö on vapaa (kts. kohta 3.3) ja sille on kohtalaisen helppo tehdä ohjelmia, ovat käyttäjät kehitelleet paljon sovelluksia tälle alustalle. Nämä

ohjelmat ovat joko varta vasten tälle laitealustalle tehtyjä taikka sitten käännettyjä ja ehkä hieman muokattuja tavallisia Linux-ohjelmia. (Maemo-yhteisö 2007a.)

Ohjelmistotarjonta tälle laitteelle on siis verrattain laaja ja monipuolinen.

3.3 Ohjelmointiympäristö

Nokia 770:n käyttöjärjestelmänä toimii Nokian oma Internet tablet -käyttöjärjestelmä, joka pohjautuu Linuxiin (Nokia 2005b). Linuxiin pohjautumisen johdosta ohjelmistokehitys tälle alustalle on miltei samantyyppistä, kuin ohjelmistokehitys yleensä on Linux-alustalla. Kuitenkin on otettava huomioon, että Nokia 770 on sulautettu järjestelmä ja näin ollen rajallinen resursseiltaan. (Maemo-yhteisö 2007b.)

Ohjelmistokehittäjälle on suurta apua siitä, että laite ohjelmistoltaan pohjautuu Linux-käyttöjärjestelmään. Linuxille on tehty paljon kirjastoja ja ohjelmistoja, jotka käännettävissä pienillä muutoksilla Nokia 770:n tarjoamalle Linux alustalle. Kirjastoja ja valmiita ohjelmia kääntämällä ohjelmoija saa Linux maailman laajan ohjelmisto- ja kirjastotarjonnan käyttöönsä helpottamaan ja nopeuttamaan mobiilia sovelluskehitystyötä.

3.3.1 MAEMO

Maemo on avoimen lähdekoodin sovelluskehitysalusta, joka on tarkoitettu Linuxiin pohjautuville kämmenlaitteille. Käytännössä tällä hetkellä ainut laite, jolle maemo alustalla voi ohjelmia tehdä, on Nokia 770 Internet Tablet. Periaatteessa kuitenkin kyseessä on alusta, jolla ohjelmiston kehitys onnistuisi muillekin laitteille. (Maemo-yhteisö 2007b.)

Maemo rakentuu monesta erilaisesta avoimen lähdekoodin komponentista. Se tarjoaa käyttäjälleen

- mahdollisuuden testata kohde laitteen ohjelmia tavallisella PC:llä
- helpotusta graafisten käyttöliittymien ohjelmointiin
- mahdollistaa perinteisten Linux ohjelmien käännöksen kohdealustalle
- mahdollistaa testatun ja ohjelman pakkaamisen ja kääntämisen kohdelaitteen ymmärtämään muotoon. (Maemo-yhteisö 2007b.)

3.3.2 Hildon application framework

Hildon application framework eli lyhennettynä HAF on osa maemo-sovelluskehitysalustaa. HAF perustuu GNOME:n perusteknologioihin, kuten esim. GTK+ widgets. (Maemo-yhteisö 2007c.)



Kuva 5. Ohjelmistoalustan rakenne

Kuvassa 5 on esitetty, miten asian voi yleistäen hahmottaa. Kaikkein alimpana on aivan tavallinen Linux-käyttöjärjestelmä, joka on toteutettu Nokia 770:n alustalle. Sen perustoiminnallisuudet on tietenkin muokattu vastamaan laitteen resursseja. Tämän päällä on Debian-jakelulle ominaisia paketteja. GTK, joka on siis käyttöliittymäkomponenttien kirjasto, ja D-Bus, joka on sovellusten välisen viestinnän kirjasto, ovat kaiken tämän päällä. Hildon kirjastot, jotka siis kuuluvat HAF:iin, on rakennettu kaikkien näiden teknologioiden päälle. Käyttäessään HAF:ia ohjelmoija voi käyttää hyväksi niin paljon kuin pystyy ylimmän kerroksen eli Hildon-kirjaston palveluja, koska tämä helpottaa huomattavasti

ohjelmointia. Kaikkea ei kuitenkaan ole abstraktoitu Hildon-kirjastoon, vaan esim. käyttöliittymää ohjelmoitaessa ohjelmoija tulee käyttämään paljon suoraan GTK+ -kirjaston komponentteja. Järjestelmätason viesteihin, kuten esimerkiksi patterin virran vähyyteen, ohjelma saadaan reagoimaan ainoastaan, kun käytetään suoraan D-Bus:n tarjoamaa rajapintaa.

Muita ohjelmointia helpottavia peruskirjastoja, joita ohjelmoinnin apuna voi käyttää, on mm. Glib. Se on kirjasto, joka on tarkoitettu helpottamaan C-ohjelmointia. Se tarjoaa tietotyyppejä, tietotyyppi konversioita, makroja, merkkijono-ominaisuuksia, tiedostojen käsittelyominaisuuksia jne. (Gnome-kehittäjäyhteisö 2007.)

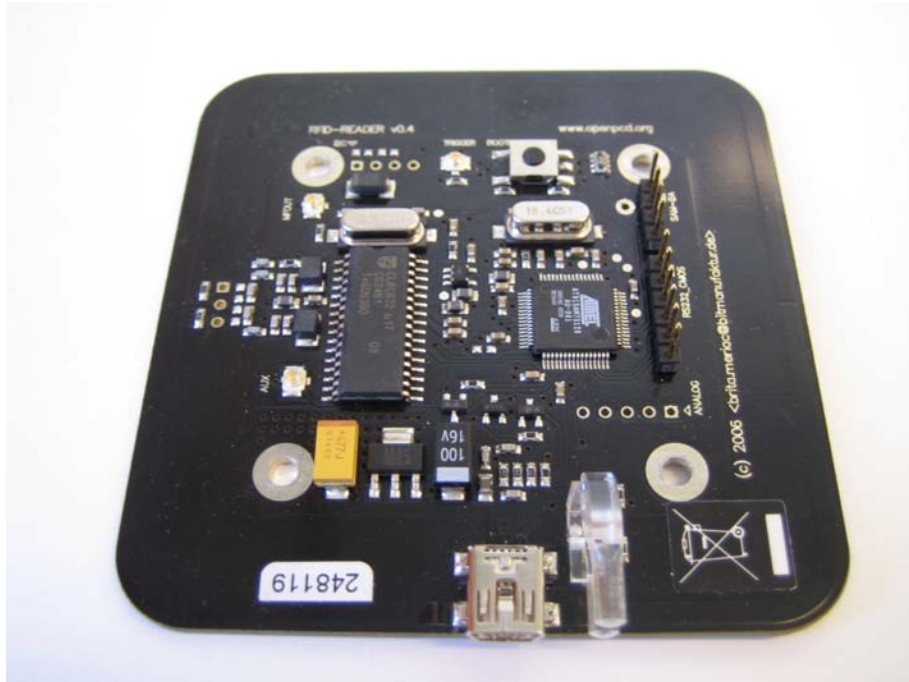
4 OpenPCD – RFID-LUKIJA

4.1 Bitmanufaktur lyhyesti

Bitmanufaktur on saksalainen yritys, joka on OpenPCD-lukijan laitteiston ja ohjelmiston suunnittelijoiden perustama yritys. Se valmistaa ja toimittaa OpenPCD-lukijan tilauksesta. OpenPCD-lukija ei sinällään kuulu yrityksen tarjoamiin tuotteisiin siksi, että se on suunniteltu yhteisöllisesti avoimilla lisensseillä. Yritys tarjoaa kuitenkin konsultaatiota RFID-teknologioihin liittyen. Yrityksen varsinaisia tuotteita ovat mm. aktiivitagien seurantajärjestelmä, aktiivitagien valmistus ja palomuurisovellukset. Yritys tarjoaa myös hardware suunnittelua ja sulautettujen järjestelmien osaamista. (Bitmanufaktur 2007.)

4.2 OpenPCD-lukija

Valmistajan kotisivuilla OpenPCD (Proximity Coupling Device, PCD) esitellään lähilukulaitemalliksi, joka käyttää 13,56 MHz:n taajuutta kommunikointiin. Laite kykenee keräämään tietoa kontaktittomilta lähilukuteknologiaan perustuvilta sirukortteilta (Proximity Integrated Circuit Card, PICC) käyttäen ISO 14443 ja ISO 15693 -standardeja. (OpenPCD:n kehittäjät 2007a.)

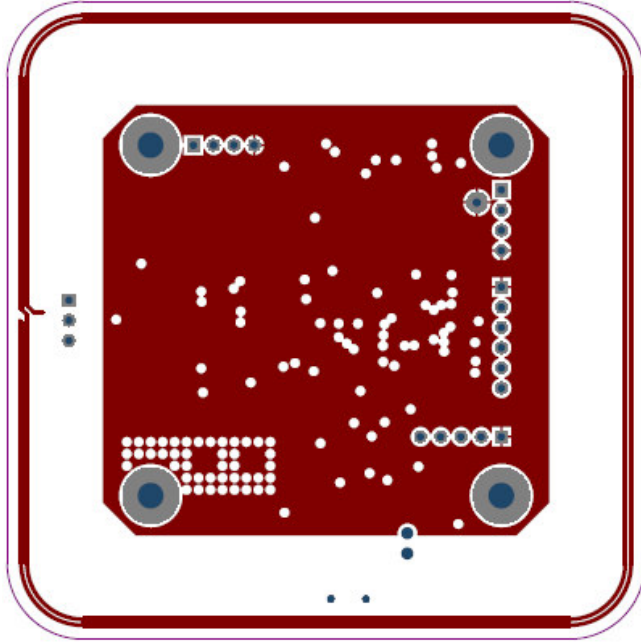


Kuva 6. OpenPCD-lukija

OpenPCD-lukija, joka on nähtävillä kuvassa 6, on tarkoitettu lähinnä tutkimus- ja esitteilykäyttöön. Käytännön käyttökohteeksi on listattu esim. kulunvalvontajärjestelmät. Lukija on kooltaan 80 mm * 80 mm * 27,5 mm.

4.2.1 Tekniset ominaisuudet

Lukijan 13.56 MHz:n taajuudella tapahtuva luku- ja kirjoitustoiminta perustuu Philipsin CL RC632 -mikropiiriin, joka tukee ISO 14443 A&B, ISO 15693, Mifare jaICODE-protokollia. Prosessorina käytetään Atmelin AT91SAM7S128 prosessoria 32-bittisellä RISC arkkitehtuurilla. Käytössä on 128 kilotavua Flash-muistia ja 32 kilotavua SRAM-muistia. Yhteyden Atmelin mikroprosessoriin saa USB-B-MINI liittimen kautta tai 20-pinnisen JTAG liittimen kautta. Tutkimusta sekä signaaliseurantaa varten OpenPCD-lukijassa on lukuisia rajapintoja. Lukijaan saa lisättyä myös ylimääräisen ulkoisen antennin. (OpenPCD:n kehittäjät 2007a.)



Kuva 7. OpenPCD-lukijan antenni (OpenPCD:n kehittäjät 2007a).

Lukijan oma antenni kehystää koko systeemiä ja on nähtävissä kuvassa 7. Kuvassa on nähtävissä OpenPCD-lukijan piirilevyn kolmannen kerroksen layout.

4.2.2 Ohjelmisto

OpenPCD-lukijalle löytyy monta erilaista firmware-ohjelmistoa. Nämä firmwaret ovat kaikki lisensoitu avoimen lähdekoodin lisenssien mukaan. Esimerkkeinä firmwareista voidaan mainita main_mifare:n, main_librfid:n ja main_dumbreader:n. (OpenPCD:n kehittäjät 2007b.)

Main_librfid on firmware, joka itsessään sisältää koko RFID-pinon toiminnot. Se on tarkoitettu pohjaksi ohjelmistoille, jotka tekevät OpenPCD-lukijasta autonomisen lukijan ilman kytkentää tietokoneeseen. (OpenPCD:n kehittäjät 2007c.)

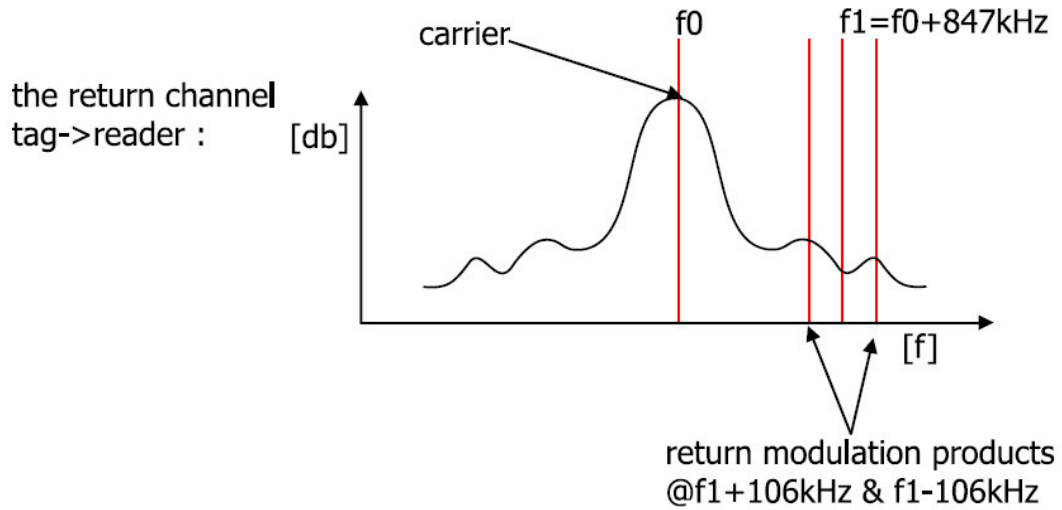
Main_mifare firmware on oikeastaan sama asia kuin main_librfid. Siihen kuitenkin lisätty toimintoja, jotka tekevät siitä yhteensopivan Philipsin mifare nimeä kantavien rfid-älykorttien kanssa. (OpenPCD:n kehittäjät 2007d.)

Main_dumbreader on firmwareista yksinkertaisin. Se tarjoaa ainoastaan primitiivejä RC632-piirin käsittelyyn usb-portin kautta kytketyille laitteille. Nämä primitiivit ovat yksinkertaisia funktioita, jotka mm. mahdollistavat RC632-piirin rekisterien käsittelyn. Main_dumbreader firmwarea käytettäessä tulee varsinainen RFID-pinon äly olla laitteessa johon OpenPCD-lukija on kytketty. (OpenPCD:n kehittäjät 2007e.) Tämä antaa etuna mm. sen, että RFID-pinon ohjelmistopäivitykset ovat paljon helpompia. Tavallisimmin lukija on kytkettynä tietokoneeseen. Tietokoneen ohjelmistopäivitys on helpompaa kuin lukijan firmwaren.

4.3 Lukijan toiminta

Milosch Meriac, yksi OpenPCD-lukijan kehittäjistä, tiivistää laitteensa toiminnan esitelmämateriaalissaan oheisesti.

Kommunikointiin lukijan ja tagin välillä 13.56 MHz:n taajuudella sovelletaan ISO 14443 tai ISO 15693 (ei tuettu projektin aikana) -standardeja. Lukija ja tagi yhdistyvät toisiinsa ilmateitse magneettisen kentän kautta sähkömagneettista induktiota hyväksi käyttäen. Kohdassa 2.4 on selitetty tämä tarkemmin. Lukija lähettää 13.56 Mhz:n amplitudimoduloidun kantosignaalin, jota tagi käyttää virtalähteenään. Tagi muuttaa antenninsa kuormaa tagin datan mukaan. Milosch Meriac selvittää tätä kuvassa 8. (Meriac, M. 2006)



Kuva 8. OpenPCD-lukijan ISO 14443 taajuus spektri tagi-lukija kommunikoinnissa (Meriac, M. 2006).

Lukija–tagi-kommunikointi on FDX- (Full Duplex) moodissa. Tämä tarkoittaa, että kommunikointi on samanaikaista molempiin suuntiin. Tästä syystä vastaussignaalin taajuutta tulee muuttaa, koska muuten lukija ei tunnista signaalia. Jollei paluusignaalia muutettaisi lukijan lähettäessä signaalia tunnistaakseen kuuloetäisyydellä olevat tagit, sen tulisi lopettaa kuuntelu aika ajoin (Half Duplex). Tällöin tageissa tulee olla oma tehovarasto, esim. kondensaattoreita, jotta niillä riittäisi energia paluusignaalin generointiin ja lähettämiseen. Passiivisten tagien hintojen minimoimiseksi käytetään paluusignaalin muuttamista, jolloin tagi saa tarvittavan tehon lukijalta ja lukija kykenee tunnistamaan tagit keskeyttämättä kantosignaalin lähetystä. (Garfinkel & Rosenberg 2005, 23.)

4.3.1 Librfid

Librfid on vapaan ohjelmistoperiaatteen RFID-kirjasto. Se toteuttaa lähietäisyyden lukulaitteiden puolen RFID-protokollista seuraavat: ISO 14443 A, ISO 14443 B, ISO 15693, Mifare Ultralight ja Mifare Classic. Librfid kirjastoa suositellaan käyttämään OpenPCD-lukijan kanssa vaikkakin se tukee myös muita lukijoita. (Welte, H. 2006.)

Koska librfid-kirjasto toteuttaa RFID-pinon itsessään, tulee OpenPCD-lukijan firmwarena käyttää main_dumbreader:ä (kts. kohta 4.2.2.).

5 TUTKIMUSONGELMA

5.1 Projektin taustaa

RFID on tunnettu, mutta suursovellusasteeltaan vähälle jäänyt tekniikka. Viimeaikoina RFID on noussut mm. logistiikka- ja terveydenhuoltoalojen kiinnostuksen huipulle. Tekniikkaa sovelletaan moneen eri tarkoitukseen, vaikka tekniikka itsessään ei ole monille tuttu. Edellä mainitut seikat tekevät tekniikasta kiinnostavan, ja sille löytyvät laajat markkinat.

SAMK:n ja Satakunnan sairaanhoitopiirin yhteinen projekti nimeltään Päiväkirurgiapoliklinikan logistiikka Satakunnan sairaanhoitopiirissä tutki yhtenä osa-alueenaan RFID-tekniikan käyttöönottoa Porin uuden päiväkirurgia-osaston tiloissa. Tutkimuksessa ilmeni mahdollinen tarve mobiilipäätelaitteelle, joka kulkisi esim. hoitajan mukana tuoden tuoretta potilasdataa suoraan tilanteiden ytimeen ja joka olisi silti luotettava. Kannettavalle tiedonkeruu, -hallinta sekä -verifiointilaitteelle voi kuvitella olevan tarvetta myös toimitusketjunhallintajärjestelmän eri osa-alueilla.

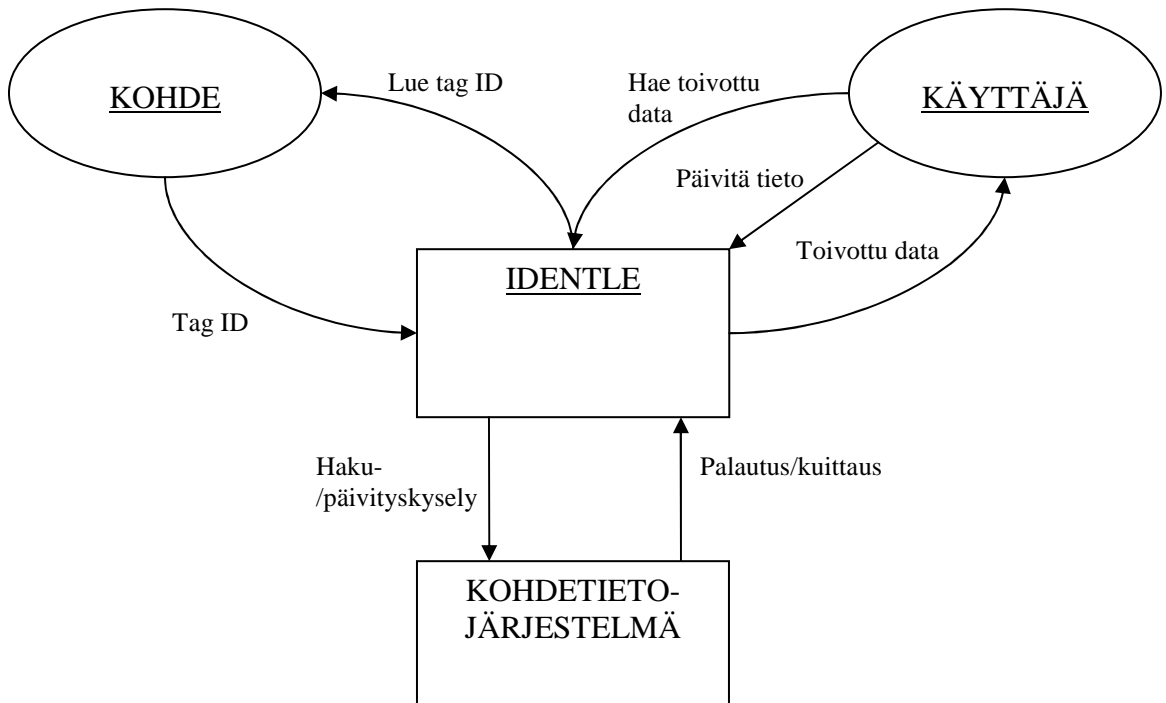
5.2 Projektin esittely

Lähdemme insinööriyössämme luomaan ratkaisua, joka täyttäisi em. kannettavan lukijan tarpeen teollisuuden ja terveydenhuollon eri osa-alueilla. Lähtökohtana on yhdistää Nokia 770 Internet Tabletin prosessointikyvykkyudet sekä OpenPCD RFID-lukija yhdeksi kokonaisuudeksi, joka on yhteydessä Nokia 770 Internet Tabletin luomin mahdollisuuksin palvelimeen. Kutsumme projektin lopputuotetta nimellä IDENTLE.

IDENTLE on tarkoitus saattaa myytävään muotoon koteloimalla se Nokia 770:n takakuoreen istuvaksi ja itse Nokia 770-laitteesta irrotettavaksi yksiköksi. Näin IDENTLE:ä voidaan myydä irrallisena lisälaitteena. IDENTLE:n ohjelmisto sijoitettaisiin RS-MMC-muistikortille, josta se on asennettavissa Nokia 770 Internet Tablettiin tai vastaavaan isäntälaitteeseen. Vastaavuudella tarkoitetaan laitetta, joka pyörittää Hildon Application frameworkia.

IDENTLE:n Nokia 770 toteutettu ohjelmiston perustoiminnallisuus on RFID-tunnisteiden lukeminen ja tunnistetietoja vastaavan, käyttäjän pyytämän, tiedon hakeminen palvelimelta ja tietojen esittäminen graafisena käyttöliittymänä käyttäjälle Nokia 770:n ruudulla.

IDENTLE:n yleiskuvaus on esitelty Kuvassa 9.



Kuva 9. IDENTLE:n toiminnan yleiskuvaus.

Toiminta koostuu neljästä päätekijästä, jotka esitellään myös taulukossa 2: *KOHDE*, *KÄYTTÄJÄ*, *IDENTLE* ja *KOHDETIETOJÄRJESTELMÄ*. *IDENTLE* on *KÄYTTÄJÄN* päälle ja pois päältä kytkettävissä, *KÄYTTÄJÄ* kontrolloi *KOHEIDEN* lukua. *KOHDE* on *IDENTLE* laitteella tunnistettava yksilö, joita voi olla yhdellä lukukerralla monia. *KOHEIDEN* tunnisteet eli tagID:t listataan *KÄYTTÄJÄLLE*, joka voi pyytää *IDENTLE* laitteen hakemaan tai päivittämään *KOHDETIETOJÄRJESTELMÄLTÄ* tietyn tunnisteiden dataa. *IDENTLE* ottaa yhteyden *KOHDETIETOJÄRJESTELMÄÄN*, joka palauttaa halutut tiedot *IDENTLE* laitteelle. *IDENTLE* esittää tiedot *KÄYTTÄJÄLLE* halutussa muodossa.

Taulukko 2: IDENTLE-projektin entiteetit ja niiden tehtävät

ENTITEETTI	KUVAUS
<i>KOHDE</i>	Kantaa yksilöityä tag ID:tä.
<i>KÄYTTÄJÄ</i>	Kohteen tagin luku Kohteen tietojen tarkastelu
<i>IDENTLE</i>	Listaa luettujen tagien ID:t Tallenna luettujen tagien ID:t Kohteen tietojen näyttö
<i>KOHDETIETOJÄRJESTELMÄ</i>	Kohteen tietojen varastointi

5.2.1 Laitteistovaatimukset

Lukijan laitteistovaatimukseen vaikuttaa osittain ratkaisussa käytetty Nokia 770 Internet Tablet, johon lukijan tulee olla liitettävissä. Tosin tulevaisuudessa *IDENTLE* tulee olla yhdistettävissä myös jonkin vastaavan, Hildon Application frameworkia tukevan laitteen kanssa. Lukijalle tulee löytyä ajurit USB-väylällä.

Lukijan tulee tukea joko ISO 14443 tai 15693 -standardeja ja lukuetaisyyden tulee olla minimissään 5 cm laitteesta. Kooltaan lukijan kiinnityspisteet eivät saa olla Nokia 770:n suoja kuoren mittoja suuremmat ja sen tulee tukea USB-liitäntää.

USB-portteihin liitetyt laitteet vaativat toimiakseen viiden voltin jännitteen. Tämän jännitteen syöttö tulee olla tasaista sekä USB-standardien mukaista.

5.2.2 Ohjelmiston toiminnallinen määrittely

Toiminnallisessa määrittelyssä on pyritty kuvaamaan ohjelmiston toiminnot mahdollisimman kattavasti. Tämän lisäksi määrittelyssä on käyty läpi kaikki ohjelmiston liitännät sekä käyttöliittymän näkymät. Toiminnallisen määrittelyn tarkoituksena on antaa käytännön toteutukselle perusteet ja rajat. Tätä työtä varten tehtiin kaksi toiminnallista määrittelyä. Päätelaitteelle ja palvelimelle tehtiin molemmille omat määrittelynsä. IDENTLE:n päätelaitteen toiminnallinen määrittely on esitetty liitteissä.

Mahdollisesti poiketen normaaleista toiminnallisista määrittelyistä päätelaitteen määrittelyssä on paljon toteutustapoja rajoittavia määrittelyjä. Suurimman rajoituksen aiheuttaa tietenkin päätelaitteena käytetty Nokia 770 -laite. Sen ohjelmointiympäristön rajoitukset rajoittavat käytettävissä olevia toteutustekniikoita. Toisena merkittävänä rajoittavana tekijänä on RFID-lukija. Lukijaksi valittiin OpenPDC-lukija ennen ohjelmiston määrittelyn aloittamista. Tämä johti siihen, että RFID-lukija kommunikointiin käytettävä kirjastona oli käytettävä librfid-kirjastoa vaihtoehtojen puuttuessa.

Määrittelyssä on ohjenuoraksi sovittu valmiiden avoimenlähdekoodin komponenttien käyttäminen. Valmiiden komponenttien käyttämisen suosittelemisen taustalla on ajatus, että ohjelmiston toteuttamiseen käytettävä aika pienenee, mitä enemmän käytetään valmiita komponentteja. Projektin edetessä käytäntö kuitenkin osoitti, että saavutettu aika-voitto käytettäessä valmiita komponentteja oli odotettua pienempi. Valmiiden komponenttien käytön opiskeluja ja tarpeellisten toiminnallisuuksien löytäminen osoittautui oletettua hitaammaksi prosessiksi.

Ohjelmiston ominaisuudet voisi tiivistää kolmeen käyttöliittymänäkymään ja niiden toiminnallisuuksiin. Näkymiä ovat aloitus-, listaus- ja tietojen näyttönäkymä. Aloitusnäky-

män toiminnallisuuksia on joko yhden taikka useamman RFID-tunnisteen lukemisen aloittaminen. Listausnäkyvä listaa luetut tagit silloin, kun on valittu monen tagin lukemisvaihtoehto. Listausnäkyvä myös avaa tietojen näyttönäkymän silloin, kun valitaan jokin luetuista tunnisteista. Tietojen näyttönäkymä esittää potilaan tiedot.

6 IDENTLE

6.1 Nokia 770 & OpenPCD lukija

Nokia 770 on perussovelluksiltaan ja toiminnoiltaan tarkoitettu internetin selailuun sekä sähköpostin lukemiseen. Käteen istuvuus, suuri kosketusnäyttö sekä selailua helpottava käyttöliittymä antavat käyttäjälle kannettavan tietokoneen ominaisuudet kevyessä pake-tissa. Kuten kannettava tietokone, eri palveluihin ja palvelimiin yhteyden muodostami-nessa, Nokia 770 on riippuvainen muista laitteista. IP-puhelun muodostamiseen tulee luo-da yhteys joko Bluetoothin kautta GSM-verkossa kiinni olevaan kännykkään tai ottaa luoda puhelu WLAN:in kautta. Yksi projektimme kannalta olennainen ero kannettaviin tietokoneisiin on ulkoisten laitteiden liittäminen Nokia 770:een. Pienellä muokkauksella Nokia 770 saadaan isännöimään muita laitteita, kuten ulkoista näppäimistöä tai hiirtä Bluetooth-tekniikan kautta.

Open PCD-lukija toimii datakommunikaatiossa kuten Nokia 770:aan kytkettynä ns. ”tyhjänä laitteena.” Nokia 770 on kykenevä itsenäiseen tiedonkäsittelyyn sekä pyörittä-mään muistikapasiteettinsa rajoittaman määrän sovelluksia oman akkuvirtansa avulla. Open PCD-lukija taas ottaa virtansa USB-liittimen kautta ja suorittaa tietyn määrän en-nalta määrättyjä funktiota, jotka se lähettää eteenpäin.

6.1.1 Yhteensopivuus

Nokia 770 Internet Tablet on tehdasasetuksiltaan ns. client-laite eli se ei tue ulkoisia lait-teita. Jotta saamme Nokia 770:n ja OpenPCD-lukijan yhdistettyä, tulee meidän asettaa Nokia 770 isännäksi (ns. host-mode). Nokia 770 ohjelmisto saadaan isännöimään USB -laitteita asettamalla sen USB-väylä host-tilaan. Oiletusarvoisesti Nokia 770:n USB-väylä

on asetettu ns. slave-tilaan. Tilan aktivointi tapahtuu samalla työkalulla, millä laitteeseen fläshätään uusia käyttöjärjestelmäversioita. Kokonaista uutta käyttöjärjestelmäversiota ei kuitenkaan tarvitse laitteelle fläshätä, jotta USB-väylän saisi host-tilaan. Riittää, että ajaa fläsherin ”-enable-usb-host-mode”-parametrilla.

Tämän jälkeen USB-väylään voidaan kytkeä slave-laitteita ja Nokia 770 kykenee toimimaan niiden isäntänä. Tämä edellyttää tietenkin, että kappaleessa 6.1.1 mainittu USB-väylän virransyöttö on kunnossa. (Salminen 2005.)

Nokia 770:n USB-portti ei anna esim. OpenPCD-lukijalle tarvittavaa +5 voltin jännitettä. Tämä luo tarpeen välikappaleelle, joka syöttää tarvittavan jännitteen USB-porttiin aktioiden näin laitteen USB-piiriin. Laite luulee olevansa kytketty tietokoneeseen ja kommunikointi on mahdollista. Väliin tulevaa laitetta kutsutaan USB power injectoriksi. Siitä enemmän kohdassa 6.2.

Nokia 770 Internet Tablet sekä OpenPCD-lukija on tarkoitus yhdistää yhdeksi laitteeksi. Fyysiseltä kooltaan laitteet ovat samankokoisia. Kuvassa 10 näemme laitteet vierekkäin.



Kuva 10. IDENTLE:n koon määrittävät komponentit

RFID-lukijan signaali voi häiriytyä Nokia 770 läheisyydessä monella eri tavoin. Nokia 770:n akku, näyttö ja kuoret voivat vaikuttaa lukijan lähettämään signaaliin häiritsevästi. Testaus käydään läpi kohdassa 6.4.

6.1.3 Ohjelmistovaatimukset

Kun USB-väylä on saatu toimintaan raudan osalta, tarvitsee laitteen ohjelmistoa päivittää. Kuten edellisessä kappaleessa todettiin, kaupasta ostettuna Nokia 770 USB-väylä ei ole host-tilassa. Tästä syystä laitteen käyttöjärjestelmäkään ei sisällä host-tilassa toimimiseen tarvittavia kirjastoja. Nämä kirjasot on kuitenkin helppo asentaa laitteelle.

Koska Nokia 770-laite pohjautuu Debian linuxiin, (kts. luku 6.3) on siinä käytössä samat työkalut ohjelmien ja kirjastojen asennukseen kuin pöytäkoneille tarkoitetuissa järjestelmissä. Debian-pohjaisissa järjestelmissä tämä työkalu on apt-get. Apt-get ohjelman avulla pitää laitteelle asentaa libusb-kirjasto. Tämä kirjasto tarjoaa ohjelmille rajapinnan USB-väylän laitteisiin. (Erdfelt 2007.)

OpenPCD-lukijan kanssa kommunikointiin käytetään librfid-kirjastoa. Tätä kirjastoa ei löydy valmiiksi paketoituna Nokia 770 laitteelle, joten sen asentaminen käyttäen apt-get työkalua ei onnistu. Nopein tapa saada kirjasto käyttöön Nokia 770 laitteelle on kääntää se lähdekoodista laitteelle sopivaksi käyttäen ristikäntäjää, joka on esitelty kohdassa 7.1.2. Tämän jälkeen kirjasto vain kopioidaan laitteeseen oikealle käyttöpolulle.

6.2 USB Power Injector

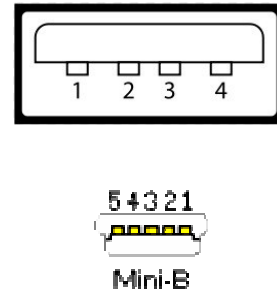
Jos Nokia 770:een halutaan liittää ulkoisia laitteita, joita sen halutaan hallinnoivan, tulee Nokia 770 asettaa isännöimään (em. host-mode), ja USB-portista on saatava +5V jännite. Sellaisenaan Nokia 770 ei tähän pysty, vaan väliin täytyy tehdä laite, joka tuottaa tasaisen viiden voltin jännitteen 9-12V paristosta. Laitetta kutsutaan USB power injectoriksi.

USB power injector -laite voidaan rakentaa yksinkertaisimmillaan yhden 7805 jänniteregulaattorin ja kahden USB-naarasliittimen kytkennällä. Kuitenkin lisäämällä diodi paristolta tulevaan vetoon ja kondensaattorit jänniteregulaattorin sisään menoon ja ulostuloon, saadaan kytkennästä huomattavasti vakaampi sekä turvallisempi. Kondensaattorit tasoittavat regulaattorilta mahdollisesti tulevat tehopiikit, ja diodi estää pariston väärinkytkenän aiheuttamat vahingot.

Power injektoriin kytketään sekä Nokia 770 ja RFID-lukija. Nokia 770 liitetään USB-naarasliittimeen ja lukija mini-B urosliittimen kautta. USB-liittimien lankajärjestys on esitetty Kuvassa 11 ja selitykset taulukossa 3.

Taulukko 3. USB- ja USB-mini-B-liittimen lankajärjestys (Wikipedia. 2007a.)

Lanka#	Väri	Funktio USB/mini-B
1	Punainen	V_{SUB} (4.75-5.25 voltia)
2	Valkoinen	Data -
3	Vihreä	Data +
4	Musta	Ground / NC
5(mini-B)	Musta	- / Ground

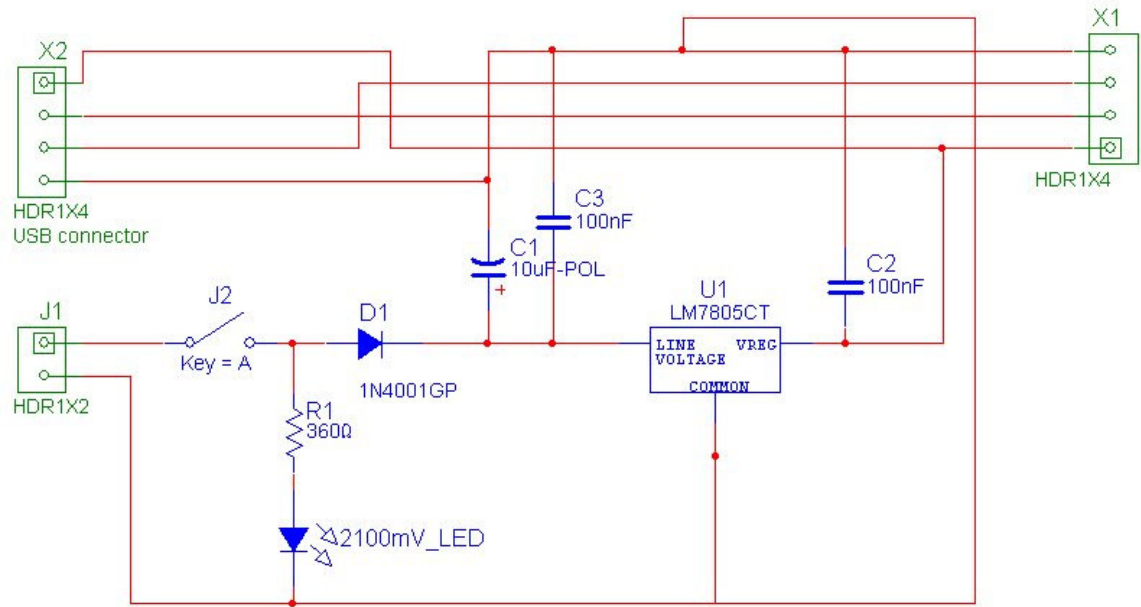


Kuva 11. USB- ja USB-mini-B-liittimen lankajärjestys (Wikipedia. 2007a.)

Langat 2 ja 3 ovat datakommunikointia varten. Vastakkaiset USB-liittimet kommunikoi-
vat kierretyn parikaapelin kautta. Dataliikenteessä arvoa 0 indikoivat jännitteet ovat 0.0-
0.3 voltia. 2.8-3.6 voltin jännitteet indikoivat ykköstä tai arvoa 1. Mini-USB:n neljättä
lankaa ei ole kytketty. (Wikipedia. 2007a.)

6.2.1 Piirikaavio

USB power injector -toteutus koostuu kolmesta kondensaattorista, diodista, 7805 jännite-
regulaattorista, kahdesta USB-liitännästä, lukittuvasta virtakytkimestä, ledistä sekä ledin
etuvastuksesta.



Kuva 12. USB power injectorin piirikaavio.

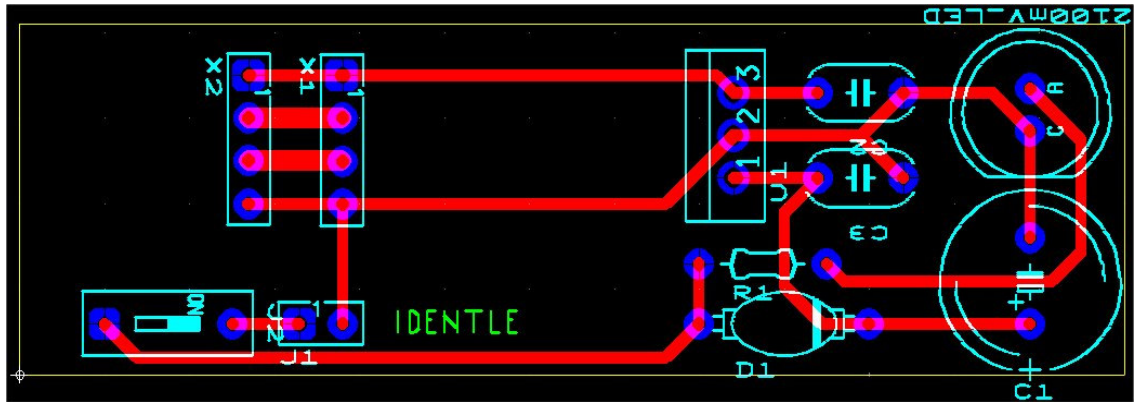
Kytkenän piirikaavio on esitetty kuvassa 12 ja taulukosta 4 löytyvät piirikaavion komponentit, niiden arvot sekä funktiot. Piirikaavio on uudelleensuunniteltu Multisim 9-ohjelmalla. Alkuperäinen piirikaavio löytyy osoitteesta: <http://www.hcilab.org/projects/nokia770/nokia770.htm>.

Taulukko 4: USB power injectorin piirikaavion komponentit funktioineen.

Tunnus	Komponentti	Arvo	Funktio
C1	Elektrolyytti-kondensaattori		Jänniteregulaattorille tulevan jännitteen tasaus
C2&C3	Kondensaattori		Jänniteregulaattorin in- ja outputien jännitteen tasaus
D1	Diodi 1N4007	-	Päästösuuntaan kytketty diodi estää väärin kytketyn virtalähteen aiheuttamat ongelmat
X1	Mini-B USB-johto	-	Open PCD lukija liitetään Mini-B USB-plugin kautta
X2	USB-naarasliitin	-	Nokia 770 liitetään ulkoisesti USBn kautta
J1	Virtalähteen liitin	-	Ulkoinen virtalähde (9-12V)
J2	Kytkin	-	Lukkiutuva virtakytkin
U1	7805 regulaattori	-	Muuttaa syötetyn jännitteen ~5 voltin jännitteeksi
R1	LEDn etuvastus	360 Ω	Rajoittaa LEDille menevää jännitettä 2,1 volttiin.
LED	Vihreä LED	-	Kynnysjännite: 2,1 V, virta: 20mA

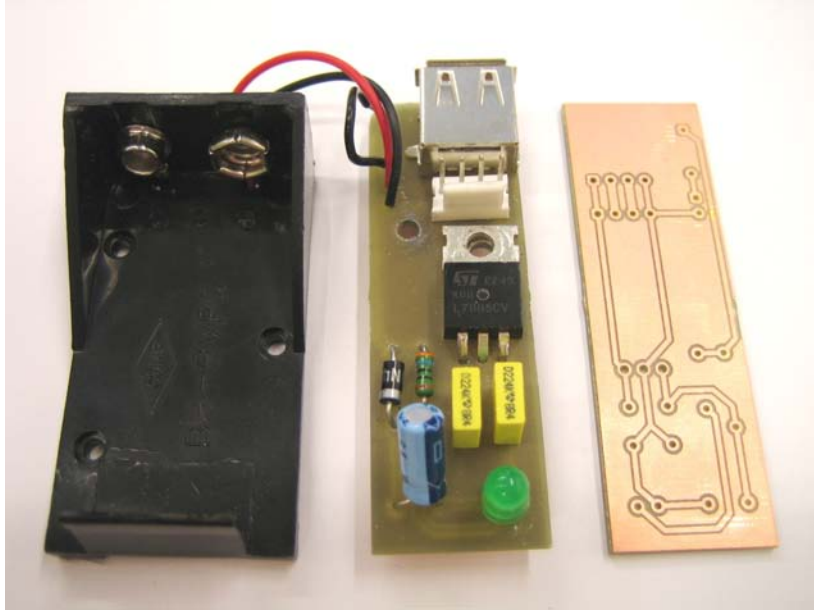
6.2.2 Toteutus

Alustava toteutus tehtiin testialustalle, kuten kuvassa 15 nähdään. Testialustalta saatiin ensikäsitys kytkennän toiminnasta käytännössä ja tehtiin tehonkulutus-mittauksia. Lopullinen versio suunniteltiin Ultiboard 9 ohjelmalla. Huomioitavaa oli datalinjojen paksuus sekä juotospadien suurentaminen juottamisen helpottamiseksi. Kuvassa 13 nähdään lopullisen version layout.



Kuva13. USB power injectorin lopullinen layout Ultiboard 9:llä.

Jyrsiminen tapahtui EMCLAB:ssa Raumalla piirikorttijyrsimellä. Kuvassa 14 nähdään jyrsitty kortti ja valmis versio samassa kuvassa.



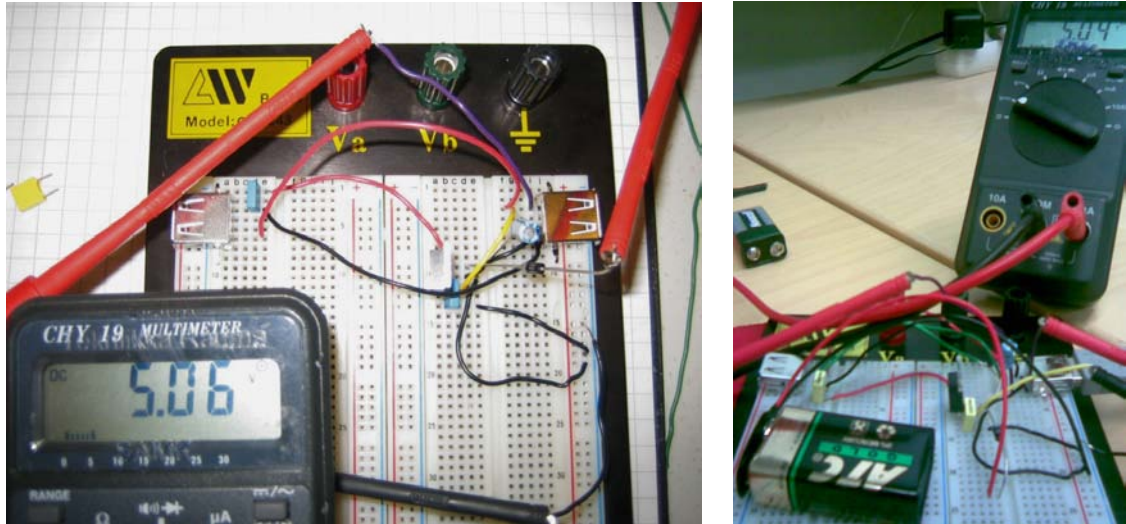
Kuva 14. Jyrsitty ja käyttövalmis power injector.

X1-liitin toteutettiin neljäpiikkisellä massaterminointiliittimellä. Katkaistu USB-johtoon kohdistuu suurehko vääntävä voima, kun se liitetään RFID-lukijaan. Johtimien pelkkä juottaminen ei siis riitä, koska ajan mittaan juotokset vääntyisivät poikki. Erillisellä liittimellä voima saadaan kohdistettua pois juotoskohdasta.

6.2.3 USB power injectorin testaus

Kehitysvaiheessa kytkeä testattiin turvallisuuden ja laitteistotuhojen välttämiseksi. Lähde, josta saimme idean USB power injectorin tekemiseen, toistuvasti muistuttaa riskeistä, joita tämän tyyppiset tee-se-itse elektroniikkaratkaisut tuovat mukanaan. Siksi USB power injectorin testit jaettiin *alustaviin* ja *lopullisiin* testeihin.

Alustavan testauksen ensimmäinen vaihe suoritettiin sähkölaboratoriossa. Kytkeä tehtiin testausalustalle, ja testauksen tarkoituksena oli mitata, saadaanko USB-portin ensimmäisestä langasta tarvittava viiden voltin jännite. Testauskytkentä nähdään kuvassa 15.



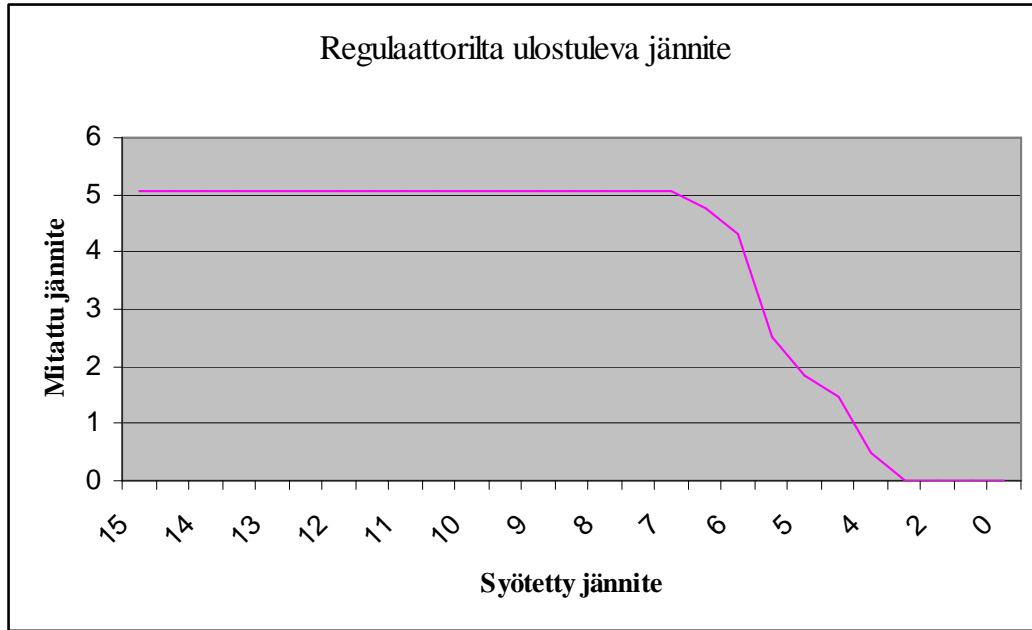
Kuva 15. Testauskytkennät alustavassa testauksessa.

Tässä vaiheessa jännite syötettiin säädettävästä virtalähteestä. Kytkentään syötettyä jännitettä vaihdeltiin yhdeksän ja kahdentoista voltin välillä. Näin toimivuus myös 12 V:n paristolla varmistettiin. Seuraavaksi jännitelähde vaihdettiin kiinteästä 9 voltin paristoon. Alustavan testauksen tulokset nähdään Taulukossa 5. OpenPCD-lukija kytkettiin onnistuneesti kytkentään.

Taulukko 5: Alustavan testauksen tulokset.

Syötetty jännite (V)	Virtamittarin lukema (V)
9 (kiinteälähde)	5.06
12 (kiinteälähde)	5.05
9 (paristosta)	5.04

Lopullinen testaus suoritettiin USB power injectorin toteutuksen jälkeen. Aluksi suoritettiin juotos- ja oikosulkutestit. Varsinaisessa testissä seurattiin USB-liittimien V_{SUB} lan-goissa (ks. kuva 15 ja taulukko 5) tapahtuvaa jännitteen muutosta, kun syöttöjännitettä lasketaan 15 voltista aina nolnaan volttiin saakka. Tällä testillä todetaan laitteen toimivuus, joka pettää jos USB-porteista mitattu jännite tippuu alle 4,75 voltin. Testin tulokset nähdään kuviossa 1.



Kuvio 1. USB power injectorin USB porteista mitattu jännite.

Jännite pysyy tasaisena 5,06 voltissa välillä 15-7,5 voltia. 7805 jänniteregulaattori muuttaa ylijäävän jännitteen lämmöksi, joka testin aikana ei kuitenkaan ollut huomattavaa. Toiminnallisuuden kannalta oleellinen raja tulee 6,5 voltin kohdalla, jolloin regulaattorilta mitataan 4,75 voltia. Tästä alemmilla jännitteillä USB power injectoriin liitetyt laitteet eivät enää saa käyttöjännitettään, eivätkä täten toimi.

6.2.4 IDENTLE:n käyttöaika

Käytön kannalta on tärkeä tietää, miten kauan laite pysyy toiminnassa. Kuten jo edellä selvitettiin, Nokia 770 ei kykene tuottamaan OpenPCD-lukijalle virtaa USB-porttinsa kautta, vaan väliin tulee USB power injector tuottamaan vaadittavat viisi voltia. Nokia 770:n käyttöaika ei siis ole riippuvainen OpenPCD-lukijasta, vaan toimii luvattujen spesifikaatioiden mukaisesti. Kuitenkin IDENTLE on riippuvainen paristonsa kapasiteetista. Käyttöaika-arvio on suoritettu 1000 milliampeeritunnin kapasiteetin omaavalla paristolla. Olennaista on tietää mitä OpenPCD-lukija kuluttaa valmius- ja kuormitusilassa. Virran-

kulutusmittaus-kytkennässä virtamittari kytkettiin sarjaan 7805 jänniteregulaattorilta X1 USB-liittimelle menevään V_{SUB} -lankaan.

Taulukosta 6 ovat luettavissa mittaustulokset ja niiden pohjalta tehdyt käyttöaika-arviot. IDENTLE:n suunnittelussa on otettu huomioon rajallinen käyttöaika virtakytkimellä. Käyttäjä voi sulkea paristolta tulevan kytkennän ja näin sammuttaa lukijan haluamanaan ajankohtana. USB power injectorin led kuluttaa 20 milliampeeria, joten se on lisättävä käyttöaika-arvioon. Myös 7805 jänniteregulaattori muuttaa ylijäävän jännitteen lämmöksi, mutta tätä hävikkiä ei oteta huomioon.

Taulukko 6: IDENTLE:n virrankulutus ja käyttöaika-arvio.

OpenPCD-lukijan tila	Virrankulutus (mA)	Arvioitu käyttöaika
Valmiustila	57,8	12 h 51 min
Kuormitustila	74,3	10 h 36 min

Käyttöaika laskettiin kaavalla (1).

$$(1) \text{ Käyttöaika} = \frac{I_h (mAh)}{I_{OpenPCD} (mA) + I_{LED} (mA)}$$

6.3 Tuotteistaminen

IDENTLE:n koteloiti oli ehdotonta sen arkojen osien peittämiseksi, käyttöiän pidentämiseksi, ulkonäön kohentamiseksi sekä käyttäjäystävällisyyden parantamiseksi. Kotelo ei ole ainoastaan suoja vaan toimii laitteen uskottavuuden sekä laadun mittapuuna – IDENTLE:n brändin luojana.

6.3.1 Vaatimukset

Tarkoituksena oli luoda ergonominen kotelo, joka mahdollistaa IDENTLE:n käytön yhdellä kädellä. Kooltaan kotelon tuli peittää kaikki osat, 770:tä ja IDENTLE:n yhdistävää USB-johtoa lukuun ottamatta. IDENTLE:n virtanäppäimen ja -ledin tuli olla sijoitettuna käyttäjälle tarkoituksenmukaisesti: virta on helposti päälle/pois kytkettävissä ja led indikoi laitteen päällä oloa.

IDENTLE on kannettava RFID-lukija; sen tulee siis istua vaivattomasti ja varmasti käteen sekä olla helposti käytettävä. Käytettävyys yhdellä kädellä tulee ottaa huomioon suunnittelemalla kotelon muoto Nokia 770 näppäinten puolelta erilaiseksi otteen parantamiseksi. Virtanäppäin tulee sijoittaa siten, että käyttäjä kykenee hallitsemaan virran käyttöä vasemman käden peukalolla.

Koska IDENTLE itsessään käyttää kotelon alle sijoitettavaa 9V:n paristoa, tulee kuoren irrotettavuus olla helppoa ja johdonmukaista. Pultit, jotka pitävät kotelon kiinni Nokia 770:ssa, tulee merkitä selkeästi tai tulee olla sijoitettu siten, ettei käyttäjä irrota IDENTLE:n osien kiinnitysruuveja.

IDENTLE on tarkoitettu erilliseksi lisäosaksi Nokia 770:tä. Kotelon tulee siis olla väri- ja muotoon sopiva, eikä se saa loukata Nokian brändiä. Kotelon ja kuoren välinen rakotoleranssi säädettiin kahteen millimetriin, ja kotelon liikkuvuus kuoren suhteen haluttiin pitää nollassa.

6.3.2 Toteutus

Kotelon työstö aloitettiin Nokia 770:n suojakuoren mitoituksella. Kuoren leveys määrää kotelon maksimileveyden, mutta kuten kuvasta 16 nähdään, RFID-lukija sen korkeuden. Muut osat mahtuvat hyvin suojakuoren ulkopinnalle.



Kuva 16. Nokia 770:n suojakuorimuokkaus ja IDENTLE:n laitteistomitoitus

Taulukossa 7 ovat koteloitavien osien mitat, jotka määrittävät kotelon sisätilavuuden: 80 mm * 135 mm * 35 mm (pituus * leveys * korkeus). Kotelon sisätilavuus tulee olla hieman lopullista koteloita pienempi ja tästä muodostunut kokoero kurotaan kotelomateriaalilla. Malli, jonka päälle kotelo tullaan tekemään, kaiverrettiin solumuovista. Mallin raakaversio nähdään kuvassa 17.

Taulukko 7: IDENTLE:n osien mitat

Osa	Pituus (mm)	Leveys	Korkeus
Suojakuori	67,0	140,0	-
RFID-lukija	80,5	80,5	27,5
USB power injector	65,0	21,0	23,5
Paristokotelo	52,1	30,5	22,5



Kuva 17. IDENTLE:n kotelon ensivaihe.

Kotelon ulkopinta tulee vaatimusten mukaisesti olla käteen sopiva. Kuten kuvassa 17 nähdään, solumuovimalli muotoiltiin laitteen vasemmalta puolelta pyöreäksi käyttäjän otteen parantamiseksi.

Kotelointimateriaalina käytettiin lasikuituvahvisteista polyesterikittiiä ja viimeistelevää muovikittiiä, joka kuvasta 17 vielä puuttuu. Muovikittiiä levitettiin solumuovimallin päälle pohjakerrokseksi. Tämän päälle tasoitettiin noin viiden millimetrin lasikuitukerros, joka muodostaa varsinaisen kotelon. Muovikittiiä levitetään ainoastaan viimeistely- ja tasoitus-tarkoituksessa, koska on helposti hiottavissa ja se tasoittaa lasikuitukitin epätasaisuudet, mutta ei ole tarpeeksi vahvaa vahvistaakseen koteloa. Tästä syystä pohjimmainen ja päällimmäinen kerros ovat molemmat muovikittiiä.

Nokia 770:n suojakuori ei ole reunoiltaan tasainen vaan kaareva (vrt. kuva 16 edellisellä sivulla), joka muodostaa haasteen kotelon istuvuudelle. Suojakuorta ympäröivä tummemman harmaa muovilista, joka on hieman muusta kuoresta koholla, toimii kotelon rajana. Lista varmistaa myös, että kotelo sijoittuu paikalleen ja pysyy tukevasti suojakuorta vasten. Kotelon istuvuus nähdään kuvassa 18, joka on otettu ennen päällimmäisen muovikittikerroksen levittämistä.



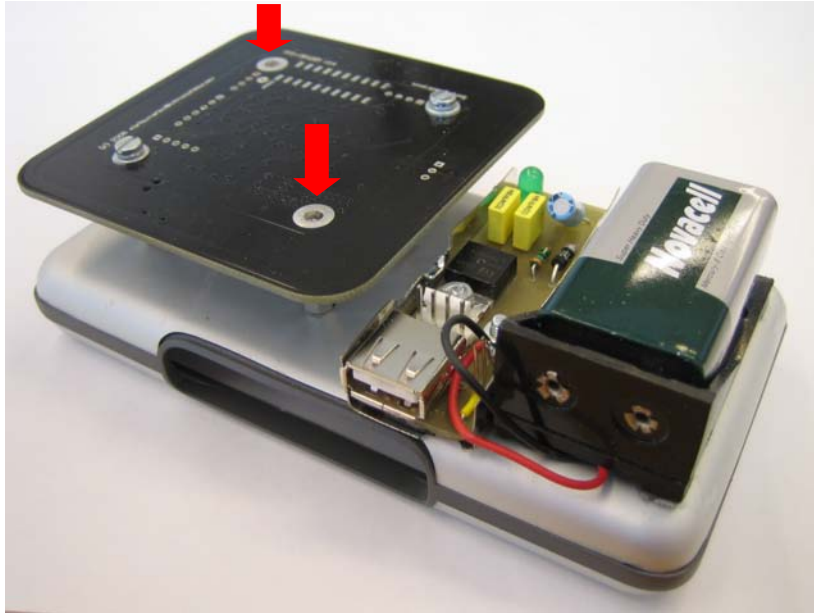
Kuva 18. IDENTLE:n kotelon istuvuus Nokia 770:n suojakuorta vasten.

Kotelon työstössä tuli ottaa huomioon sen ulkoiset osat, kuten virtakytkin sekä erityisesti USB-liitin sen pohjaosassa. Virtakytkin vaatii halkaisijaltaan 12 millimetrin reiän. USB-liitin tulee kotelosta hieman ulos, joten USB power injector voi rikkoutua huolimattomasta kotelon irrottamisesta. Kotelo jätettiin avonaiseksi suojakuoren ja ulkonevan USB-liittimen väliltä, jotta liitin ei repeytyisi irti.

Nokia 770:n suojakuoren ulkokerros on magnesiumseosta, joka on hyvin kevyttä, mutta kestävä. Ulkokerroksen paksuus on noin 0,6 mm. Suojakuoren sisäpuolella on muovikerros, jonka paksuus on noin kaksi millimetriä. Suunnitelman mukaan IDENTLE:n osat kiinnitetään suojakuoreen sisältä päin. Pulttien kannat tasoitetaan sisäkerroksen tasolle, koska ylimääräistä tilaa suojakuoren ja Nokia 770:n välillä ei ole. Suojakuoreen porattiin halkaisijaltaan kolmen millimetrin reiät ja muovikerrokseen sorvattiin sisältä päin pultin kannan muotoinen potero. IDENTLE:n osat kiristetään suojakuoreen pienillä muttereilla RFID-lukijaan lukuun ottamatta. Lukija kiinnitetään sen mukana tuleviin jalkoihin. USB power injectorin ympärille asetettiin metalloitu maataso häiriösuojaksi. Maadoitettu alumiinilokero suojaa lukijan ja pariston magneettikentiltä.

Valmis kotelo liitettiin IDENTLE:n osilla varustettuun suojakuoreen kahdella pultilla lukijan tukijalosta. Kiinnityskohdat osoitetaan punaisilla nuolilla kuvassa 19. Kiinnitysku-

vio puristaa koteloä tasaisesti suoäakuorta vasten. Kotelo on helposti irrotettavissa, koska kaikki sen kiinnitysruuvit ovat koteloä etupuolella.



Kuva 19. IDENTLE:n koteloä kiinnityskohdat suoäakuoreen.

Maalaus ja viimeistely tehtiin Nokia 770:n värimaailmaa ja tyyliä silmällä pitäen. Kotelo maalattiin mustaksi. Lopputuote on nähtävissä kuvassa 20.



Kuva 20. IDENTLE

6.4 Laitteistotestaus

IDENTLE:n laitteistotestauksen tarkoituksena oli saada varmuus laitteen riittävästä toimivuudesta, luotettavuudesta ja käytännöllisyydestä. Laitteen käyttötestiä varten luotiin todellista käyttöympäristöä jäljittelevä testiskenaario, joka käydään läpi luvussa 8. Toimivuustesti riittävän toiminnan varmistamiseksi tehtiin alustavasti silmämääräisellä etäisyysmittauksella.

6.4.1 Käytännön lukuetaisyys

ISO 14443 -standardissa, jota OpenPCD-lukija testihetkellä ainoastaan tukee, lukuetaisyys on rajoitettu n. 3-4 senttimetriin. Testi koostui tagin lukemisesta kotelon kanssa ja ilman. Lukuetaisyys mitattiin silmämääräisesti arvioiden.

Testissä kotelolla ei huomattu olevan vaikutusta lukuetaisyyteen. Lukuetaisyydeksi arviointiin noin 10 senttimetriä kotelon kanssa ja ilman.

Testaus jäi ajan puutteen vuoksi vajanaiseksi. Perusteellinen testaus tulisi suorittaa Rauman EMC-laboratoriossa, joka on tälle projektille entuudestaan tuttu USB power injectorin piirilevyn jyräpaikkana. IDENTLE:lle ei tarvitsisi suorittaa varsinaisia EMC-testauksia, vaan laboratorion testilaitteita käytettäisiin em. testaustarkoitukseen.

6.4.2 Kotelon testaus

Testiä varten rakennettiin testikotelo, joka on ominaisuuksiltaan IDENTLE:ä jäljittelevä. Testien tarkoitus on saada kuvaa, miten tuote kestää pudotusta. Testi koskee vain protokotelon kestävyyttä. Mitään takuuta RFID-lukijan tai Nokia 770:n pudotuskestävyydestä ei ole. Niihin tätä testiä ei kohdistettu.

Pudotustestissä rakennettiin kuvan 21 mukainen, painoltaan ja kulmiltaan IDENTLE:ä imitoiva kotelo, joskin pienemmässä koossa. Kuoren paksuus oli sama ja painoa lisättiin painoilla.



Kuva 21. Testikotelo

IDENTLE painoi noin 490 grammaa. Testikotelo täytettiin painolla oikeaa imitoivan iskun saavuttamiseksi. Tuote pudotettiin noin metrin korkeudesta betonilattialle. Testin mukaan kotelo kestää tasaisesti jakautuvan paineen, kun laite laskeutuu kyljelleen eikä kulmalleen. Testikotelo oli avoin yläpäästä, joten seinien tukevuus ei ole IDENTLE:n luokkaa. Kotelon laskeutuessa kulmalleen repeytyi toinen sivuseinistä kokonaan irti. Testin johtopäätöksenä voidaan todeta lasikuidun olevan liian heikkoa materiaalia todellista käyttöä kestäväksi.

7 IDENTLEN OHJELMISTO

7.1 Nokia 770:n ohjelmiston kehitystyökalut

Ohjelmistokehitystyön nopeuttamiseksi ja helpottamiseksi käytetään jo melko yleisesti graafisia työkaluja. Nokia 770:n alustalle graafisia ohjelmistotyökaluja on tarjolla rajallisesti, mutta ne kuitenkin löytyvät. Tässä luvussa on käyty läpi ohjelmistonkehitystyökaluja, joita on käytetty tämän projektin toteuttamisessa.

7.1.1 Eclipse

Eclipse on avoimen lähdekoodin projekti, jonka tarkoituksena on tuottaa helposti eri käyttötarkoituksiin ja ohjelmointikieliin skaalaantuva sovelluskehitysympäristö. Se on ohjelmoitu Javalla, ja suurin tuki siitä löytyy juuri Java-pohjaisten ohjelmistojen kehitykseen. Eclipselle on kuitenkin olemassa laajennus nimeltään CDT, joka mahdollistaa sovelluskehityksen myös C ja C++-kielillä.

Laika on Eclipsen laajennus, joka yhdessä CDT-laajennuksen kanssa mahdollistaa sovelluskehityksen Nokia 770-laitteelle. Laika on Scratchboxin (kts. kohta 7.1.2) graafinen käyttöliittymä. Se helpottaa ja nopeuttaa monta scratchboxin toimintaa. Periaatteessa Laika tarjoaa scratchboxin toiminnallisuudet Eclipse-ympäristöön sulautettuna. Laika tarjoaa myös projektipohjia, joiden päälle on helppo rakentaa oma sovellus.

Laikassa on useita Nokia 770 -sovelluskehitystä helpottavia toimintoja. Omien Nokia 770:lle tehtyjen sovellusten ajaminen scratchbox-ympäristössä onnistuu alkuasetusten jälkeen muutamalla napin painalluksella. Omasta ohjelmasta Debian-paketin luominen Nokia 770:lle asentamista varten onnistuu myös helposti.

Gazpacho-käyttöliittymän suunnittelutyökalu (kts kohta 7.1.3) on integroituna Laika-liitännäisessä. Laika tarjoaa nopean tavan käynnistää käyttöliittymän suunnittelu Nokia 770:n sovelluksille scratchbox istunnon sisällä.

Eclipseen on olemassa myös liitännäinen subversion-versionhallintajärjestelmää varten nimeltään Subclipse. Tätä projektia on kehitetty versionhallinnan alaisena, ja tästä syystä Subclipsen tarjoama mahdollisuus käyttää versionhallintaa integroituna osana itse kehitystyökalua on ollut suureksi avuksi.

Eclipse siis tarjoaa mainiot apuvälineet Nokia 770:n ohjelmistokehitykselle. Oikeastaan mitään kilpailijaa ei Eclipselle Nokia 770 -ohjelmistokehityssaralla ole. Toki ohjelmistoja voi Nokia 770 -alustalle kehittää. Eclipse yhdessä sen liitännäisten kanssa tarjoaa kuitenkin lähes kokonaisvaltaisen työkalun Nokia 770 ohjelmistoprojekteille.

7.1.2 Scratchbox

Scratchbox on ristikäntäjätyökalu, joka on suunniteltu helpottamaan sulautettujen Linux-sovellusten kehitystä. Se tarjoaa täyden pakillisen työkaluja kokonaisen Linux jake-
lun integrointiin ja ristikäntämiseen. (Movial 2007.)

Ristikäntämisellä tarkoitetaan sitä, että käännetään korkeantason kieli jonkin toisen prosessoriarkkitehtuurin konekielelle kuin mikä käännöstä suorittavan koneen prosessoriarkkitehtuuri on. Nokia 770:n prosessori on ARM-prosessori. Scratchbox-ristikäntäjätyökalujen ansiosta voidaan normaalilla pöytäkoneella kääntää ohjelmia ARM-arkkitehtuuriin sopiviksi.

Ennen kohdelaitteelle kääntämistä voidaan ohjelmasta kääntää pöytäkoneen arkkitehtuurille sopiva versio ja ajaa tätä versiota scratchboxissa, joka tarjoaa ainoastaan resursseja,

mitkä olisivat tarjolla myös itse kohdelaitteessa. Tämä mahdollistaa ohjelmiston kevyen ja nopean testauksen ennen kohdelaitteelle kääntämistä. Scratchbox tukee myös itse laitteella tapahtuvaa testausta.

7.1.3 Gazpacho

Gazpacho on käyttöliittymän suunnittelutyökalu GTK-kirjastolle (kts. kohta 3.3.2). Se on suunniteltu helppokäyttöiseksi ja yksinkertaiseksi. Rajoitettujen resurssien ympäristöissä, kuten Nokia 770 -laitteessa, käyttöliittymä on yleensä yksinkertainen ja kevyt. Tämän tyyppisen käyttöliittymän suunnitteluun Gazpacho-työkalu sopii mainiosti juuri sen takia, että siinä ei ole mitään ylimääräistä ja monimutkaista.

Gazpacho tuottaa käyttöliittymästä xml-tiedoston. Tämä xml-tiedosto ladataan ohjelman suorituksen aikana libglade-kirjaston avulla. Tällainen menettely tarjoaa ohjelmoijalle mahdollisuuden tehdä muutoksia käyttöliittymään ilman ohjelman uudelleen kääntämistä. Xml-tiedosto on myös helppolukuinen tapa esittää käyttöliittymä, ja se mahdollistaa sen, että ohjelmoija voi tehdä myös käsin pieniä muutoksia käyttöliittymään nopeasti ja helposti.

7.2 Suunnittelu

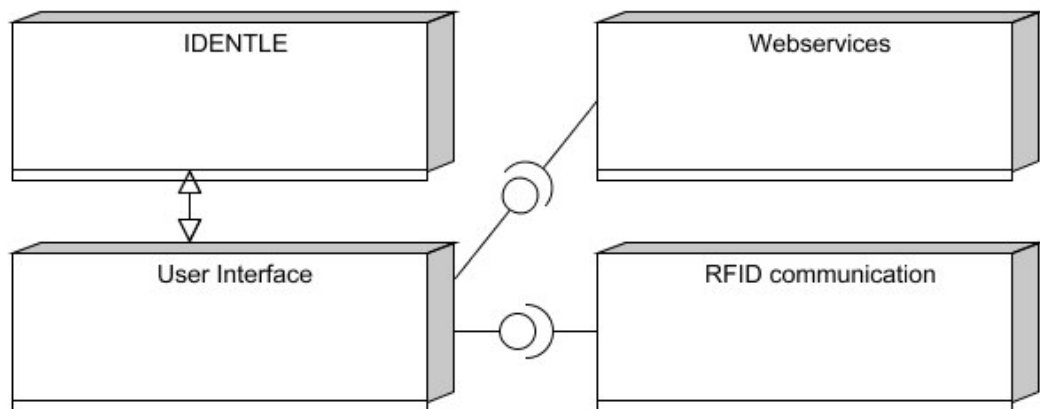
Tämä projekti on luonteeltaan tutkiva ja kokeellinen. Projektin alussa ei aina välttämättä tiedetty, mitkä toteutustavat tai -tekniikat täyttäisivät vaatimukset. Tämä teki suunnittelusta erittäin haastavaa. Projektin haasteellisuuden ja kokeellisuuden tiedostaminen jo projektin alussa johti siihen, että ohjelmisto päätettiin toteuttaa ns. evo-mallia mukaillen. Evo-mallissa toteutetaan yksi ohjelmiston ominaisuus ja tarkastetaan määrittelyt. Tämän jälkeen lähdetään toteuttamaan seuraava ominaisuutta. (Haikala & Märijärvi 2000, 30.)

Yhtä ominaisuutta toteutettaessa eri tekniikoita ja toteutusmalleja testattiin ja kokeiltiin. Näistä valittiin paras ja tämän toteutus jätettiin voimaan. Näin ollen projektin kulku tarkasteltaessa yhden ominaisuuden toteuttamista muistutti myös protoilumallia.

Ohjelmiston suunnitteludokumentteihin palattiin projektin edetessä useita kertoja. Niihin tehtiin tarkennuksia ja lisäyksiä, ja niitä myös päivitettiin vastaamaan toteutettua järjestelmää. Suunnitteludokumentit haluttiin pitää ajan tasalla, jotta niiden pohjalta voitaisiin helpommin lähteä suunnittelemaan uusia ominaisuuksia. Testaussuunnitelmat pohjautuvat ohjelmiston suunnittelu- ja määrittelydokumentaatioihin. Tästä syystä oli tärkeää pitää ne ajan tasalla.

7.2.1 Arkkitehtuuri

Ohjelmiston arkkitehtuuri on kohtalaisen yksinkertainen ja selkeä. Se koostuu neljästä moduulista. Nämä moduulit ovat IDENTLE, RFID communication, User Interface ja Webservices. Moduulit on esitelty Kuvassa 22. Kuvasta näkyy myös moduulien väliset riippuvuudet. Ohjelmistoarkkitehtuuri pyrittiin pitämään selkeänä ja helposti ymmärrettävänä. Selkeydellä ja helppoudella tavoiteltiin ohjelmiston ylläpidettävyyttä ja muokattavuutta.



Kuva 22. IDENTLE:n ohjelmiston arkkitehtuuri

Kuten luvussa kahdeksan kuvataan, IDENTLEN yksi päämäärinä on olla mukautuvainen eri käyttötarkoituksiin ja -ympäristöihin. Tämä on erityisen haastavaa toteuttaa ohjelmiston tasolla. Päämäärän tavoittamista kuitenkin helpottaa juuri selkeä ja yksinkertainen suunnittelu. Moduulit on pyritty rajaamaan selkeästi ja niiden rajapinnat on tehty mahdollisimman yksinkertaisiksi. Tämä tietenkin on aiheuttanut ylimääräistä ohjelmointia ja abstraktaatioita.

Abstraktaatiot eivät yleisesti ottaen ole suositeltavia, kun puhutaan rajoitettujen resurssien laitteista. Tällaiseksi laitteeksi voidaan laskea Nokia 770. Abstraktaatiot aiheuttavat pitkiä funktioiden kutsujonoja ja näin ollen kuormittavat turhaan laitetta. Abstraktaatioiden hyötynä on se, että ohjelmistolla on selkeä moduulinen rakenne ja yksinkertaiset rajapinnat. Näin ohjelmiston ylläpidettävyys ja muokattavuus on parempi. Ohjelmiston suunnitteluvaiheessa pitikin miettiä näiden kahden asian välillä. Pitäisikö tähän kohtaan tehdä yksinkertainen rajapinta vai säästää yksi abstraktaatiokerros ja antaa asiakasmoduulin kutsua palvelinmoduulin funktioita suoraan ilman rajapintaa.

IDENTLE:n palvelinkommunikaatioteknologiaksi on valittu WebServices. Webservices on muista riippumaton itsensä kuvaava teknologia. Se on tarkoitettu ohjelmien väliseen kommunikointiin internetin yli ja se perustuu avoimiin protokollisiin. Xml on WebServices-teknologian perusta. (W3Schools 2007a.) WebServices-liitännän tarjoamisesta ohjelmalle vastaa WebServices-moduuli. Se on itsenäinen moduuli, joka toimiakseen ei vaadi mitään muita moduuleja. Moduulille tehtiin rajapinta, jonka ansiosta se on tarvittaessa korvattavissa jollakin muulla halutulla teknologialla.

Olellaisena osana IDENTLEN ohjelmistoa on tietenkin vuorovaikutus OpenPCD-lukijan kanssa ja sen kautta RFID-tagien hallinta. Tätä kommunikointia varten on arkkitehtuurisuunnittelussa varattu oma moduulinsa nimeltään RFID communication. Tämän moduulin rajapinta tarjoaa RFID-tagien lukemisen ja kirjoittamisen muutaman funktiokutsun avulla.

IDENTLE:n käyttöliittymää varten on varattu oma moduulinsa. Selkeyden vuoksi käyttöliittymän toiminnot on haluttu erottaa itse ohjelmasta. IDENTLE on melko käyttöliittymäpainotteinen sovellus, ja näin ollen kaikki sen toiminnot ovat tiukasti sidottuna juuri käyttöliittymään. Tästä johtuen käyttöliittymämoduuli on melko keskeisenä osana ohjelmistoa.

Ohjelman oman nimen mukaan nimetty moduuli IDENTLE sisältää osan ohjelmiston toiminnallisuudesta. Tarkoituksena on, että tämä moduuli sisältää ohjelmiston älyn. Suurin osa ohjelmiston älystä kuitenkin on käyttäjän herätteisiin reagoimista käyttäen apuna WebServices ja RFID communication -moduuleita. Näiden toimintojen kierrättäminen IDENTLE-moduulin kautta on käytännössä katsoen turhaa. IDENTLE-moduuli sisältää kuitenkin joitakin toiminnallisuuksia ja tulevaisuudessa ohjelman kehittyessä yhä eteenpäin se tulee kätkemään aina vain enemmän ohjelman älyä sisälleen.

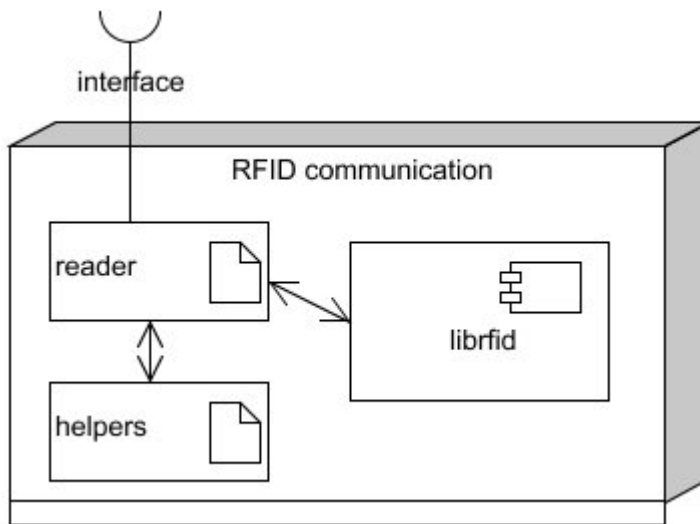
7.3 Moduulit

Arkkitehtuurisuunnittelussa esitellyt IDENTLE:n moduulit ovat kaikki oikeastaan ylemmän tason moduuleja, jotka kätkevät sisälleen alemman tason moduuleja. Tämän tapaisella sisäkkäisellä moduulien suunnittelulla on tahdottu saada aikaan ylemmälle tasolle moduuleja, jotka tarvittaessa ovat korvattavissa jollakin muulla toteutustavalla tai -tekniikalla. Tarve korvata moduuleja ja tehdä ohjelmasta geneerinen lähtee ohjelmiston toiminnallisen määrittelyn vaatimuksista (Liite 1).

Alemman tason moduulijaolla on pyritty selkeyttämään moduulin sisäistä toimintaa ja aikaan saada helposti hallittavia kokonaisuuksia. Jaoteltaessa ongelmia pieniin osiin ne yleensä tulevat helpommin hallittaviksi ja hahmotettaviksi. Tämä moduuleihin jako toivottavasti johtaa myös helpommin ylläpidettävään ohjelmistoon.

7.3.1 RFID communication

RFID communication-moduulin toteutus pohjautuu librfid-kirjastoon. Kirjastoa käsiteltiin tarkemmin kohdassa 7.3.1. Moduuli tarjoaa reader-rajapinnan, joka kätkee librfid-kirjaston toiminnan taakseen. Moduulin asiakkaiden tarvitsee vain tutustua reader rajapintaan ja heillä on käytössä librfid-kirjaston toiminnallisuus.



Kuva 23. RFID communication -moduuli

Reader-rajapinta sisältää myös librfid-kirjastoa helpottavia ja yksinkertaistavia toimintoja. Rajapinta ei siis ole pelkästään yksi abstraktaatiokerros lisää, vaan se sisältää myös älyä. Moduuli ei tarjoa muita rajapintoja. Mikäli moduulin palveluja haluaa käyttää hyödyksi, tulee käyttää tätä rajapintaa. Kuvassa 23 on esitelty RFID communication-moduulin sisälleen piilottamat moduulit. Librfid-kirjasto ei sinällään ole osa moduulia, mutta se on kuvassa mukana hahmottamisen helpottamiseksi. Kuvan kaksi varsinaista alimoduulia ovat reader ja helpers-moduulit. Alimoduuleista reader toimii RFID communication -moduulin ulkoisena rajapintana, ja alimoduuli helpers tarjoaa rajapinnalle tukevia toimintoja librfid-kirjaston kanssa kommunikointiin.

7.3.2 WebServices

WebServices-moduulin toiminnallisuuden toteuttamisesta vastaa täysin gSOAP-kirjasto. gSOAP WebServices -kehitystyökalu tarjoaa XML-sidoksen C ja C++ -kielille SOAP/XML WebServices rajapintojen kehittämisen helpottamiseksi. (Engelen 2007.) gSOAP siis tarjoaa valmiin paketin rajapinnan toteuttamiselle. Se oli merkittävin syy juuri tämän kirjaston valitsemiseksi moduulin toteutuksesta vastaavaksi osaksi.

SOAP-viestit, joita käytetään WebServices-rajapinnan viestien lähettämisessä, ovat viestejä, jotka perustuvat xml-kuvauskieleen. SOAP-viestit tarjoavat tavan ohjelmien väliseen kommunikointiin riippumatta käyttöjärjestelmästä, teknologiasta tai käytetystä ohjelmointikielestä. (W3Schools 2007b.) Juuri tämä SOAP-viestien käyttämisen tuoma vapaus alustasta ja ohjelmointikielestä oli vahvin syy siihen, miksi WebServices valittiin käytettäväksi tämän projektin palvelinkommunikointimenetelmäksi. WebServices-rajapinta on kohtalaisen helppo toteuttaa palvelimelle ja se tuo ennen kaikkea joustavuutta palvelimen rajapintaan. Joustavuudella tarkoitetaan tässä sitä, että, koska Web servicet on itsensä kuvaava protokolla, on uusien metodien käyttöönotto ja lisääminen rajapintaan nopeampaa kuin kilpailevissa toteutustavoissa. Koska IDENTLE laitteena on tarkoitus integroida osaksi isompaa järjestelmää, tarvittiin palvelinkommunikoinnin toteuttamiseen jokin teknologia, joka olisi mahdollisimman yleinen ja näin ollen tekisi IDENTLE:n tuomisen osaksi järjestelmää helpommaksi. WebServices on nouseva sovellustenvälisen kommunikoinnin tapa, joten tästäkin syystä sen käyttöön ottaminen tässä projektissa koettiin hyväksi.

Kuten RFID communication -moduulissa myös tässä moduulissa on moduulin toiminnallisuus piilotettu rajapinnan taakse. Rajapinnan käytöllä haetaan samoja etuja kuin RFID communication -moduulin kohdalla.

7.3.3 User Interface

User Interface -moduuli koostuu neljästä alimoduulista. Nämä moduulit ovat listview-, dataview-, startter-i ja ui-moduuli. Ui-moduuli on moduuli, joka sisältää käyttöliittymän yleisiä toimintoja ja tätä moduulia käyttää hyväkseen kaikki kolme muuta moduulia. Listview-, dataview- ja startteri-moduulit ovat HildonWindow-widgetin esiintymiä. HildonWindow on osa hildon-kirjastoa ja on GTK-widget, joka esittää ylemmän tason ikkunan HAF:ssa (kts. kohta 3.3.2). (Maemo-yhteisö 2007d.) Nämä kolme HildonWindow:n esiintymää vastaavat toiminnallisessa määrittelyssä esitettyjä käyttöliittymän määrittelyjä.

Listview- ja startteri-moduuleista ohjelma luo ainoastaan yhden instanssin ja säilyttää viittausta näihin instansseihin ohjelman perusrakenteessa, joka on määritelty IDENTLE-moduulissa (kts. kohta 7.3.4). Dataview-moduulista ohjelmalla voi olla useampiakin instansseja. Näiden instanssien viittaukset säilytetään linkitetystä listassa, jonka viittaus säilytetään ohjelman perusrakenteessa. Monesta mahdollisesta ilmentymästä johtuen dataview-moduulin rakenne eroaa listview- ja startteri-moduuleista.

7.3.4 IDENTLE

Yleensä ottaen hyvänä ohjelmistosuunnittelun käytäntönä pidetään varsinaisen ohjelmiston liikeideaälyn eriyttämistä käyttöliittymästä. IDENTLE:n ohjelmistossa äly on eriytetty pois käyttöliittymämoduulista jo kahteen moduuliin. Nämä ovat RFID communication- ja WebServices-moduulit. Tämän lisäksi suunnitteluvaiheessa katsottiin tarpeelliseksi tehdä vielä kolmas käyttöliittymästä erillään oleva moduuli, johon sijoitettiin se osa ohjelmiston älystä, joka ei edellä mainittuihin kahteen moduuliin kuulunut. Tämä moduuli nimettiin IDENTLE:ksi.

Projektin aikataulun ja tavoitteiden vuoksi ohjelmiston liikeideaäly ei paljoakaan ulotu RFID communication- ja WebServices-moduulien ulkopuolelle. Tästä johtuen IDENTLE-moduuli on ohjelmiston kehityskaaren tässä vaiheessa vielä aika suppea. Kun ohjelmistoon aletaan lisätä enemmän liikeideaälyä, tulee IDENTLE-moduuli huolehtimaan sen älyn toteutuksesta. Tärkein IDENTLE-moduulin sisältä toiminnallisuus on ohjelmiston perusrakenteen ja siihen liittyvien apufunktioiden toteutus.

7.4 Toteutus

Ohjelmiston toteutuksessa lähdettiin liikkeelle sovelluskehitysympäristön pystyttämiseksi. Kaikki sovelluskehitystyökalut olivat Linux-käyttöjärjestelmälle. Helpoimmaksi tavaksi osoittautui erityisesti emulointia ajateltaessa se, että Nokia 770:n ohjelmistokehitys suoritetaan mahdollisimman samankaltaisella alustalla kuin itse laitteen alusta. Koska Nokia 770 perustuu Linuxin Debian-jakeluun, lähdettiin sovelluskehitysympäristöä koamaan suosituimman ja käyttäjäystävällisimmän Debian-jakelun johdannaisselle Ubuntuille. Sovelluskehitystyökalut pohjautuivat Eclipsen ja Scratchboxin (kts. kohta 7.1) yhdistelmään. Erityisesti scratchboxin asennus ja toimintakuntoon saaminen vaativat omien shell-scriptien tekemistä ja pitkällistä perehtymistä. Lopputuloksena oli kuitenkin toimiva ympäristö, joka antoi mahdollisuuden testata Internet Tabletin ohjelmia emuloiden pöytäkoneessa.

Johtuen jo pelkästään siitä, että itse käytännön ohjelmistokehitystä tehtiin useammassa kuin yhdessä paikassa ja että projekti tehtiin parityönä, tarvittiin projektille joku menetelmä sen tiedostojen jakamiseen. Versionhallintatyökalut ovat hyödyllisiä ohjelmistokehityksessä, koska ne tarjoavat mahdollisuuden muutosten seuraamiseen ja mahdollisuuden entisten toimivien versioiden palauttamiseen. Versionhallintatyökalu toimii hyvin projektin tiedostojen keskeisenä säilytyspaikkana. Näin ollen päätettiin ottaa käyttöön versionhallintaohjelmisto tämän projektin toteutuksen apuvälineenä. Versionhallintaan käytettiin subversion-ohjelmaa. Tähän päädyttiin sen takia, että työkalu koettiin tu-

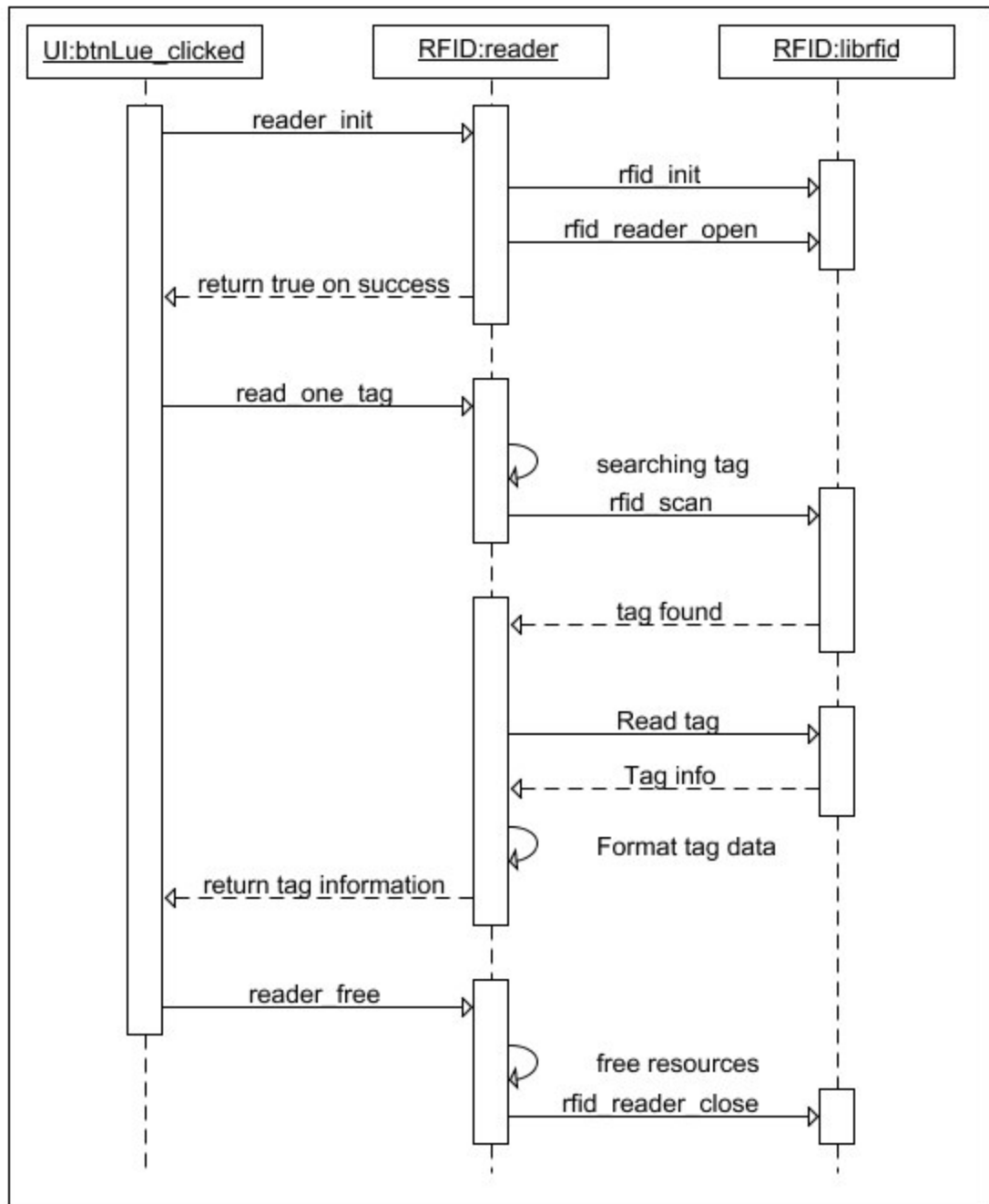
tuksi, helpoksi käyttää ja se myös toimi hyvin yhteen Eclipsen kanssa Eclipselle tehdyn liitännäisen ansiosta.

Projektin ohjelmiston dokumentoinnissa apuna käytettiin doxygen-työkalua. Doxygen generoi html-dokumentin suoraan oikealla tavalla dokumentoidusta lähdekoodista (Heesch 2007). Doxygen kykenee piirtämään ohjelmasta myös ns. funktiokutsujonoja, jotka auttoivat liiallisen abstraktaation poistamisessa ohjelmiston moduulien välisistä rajapintakutsuista. Doxygenin käyttö auttaa myös ohjelman hahmottamisessa muillakin osa-alueilla. Se on myös omiaan ohjelmiston toiminnan ymmärtämisessä, mikäli ei muutoin ohjelmaa tunne.

7.4.1 RFID communication

Tämä moduuli on pääosin toteutettu tekemällä rajapinta librfid-kirjastolle. Rajapinnan tarkoituksena oli yksinkertaistaa librfid-kirjaston käyttöä ja edesauttaa sitä, että kirjasto on tarvittaessa korvattavissa. Kirjasto voidaan joutua korvaamaan tilanteessa, jossa itse lukija vaihdetaan johonkin toiseen, jota ei librfid-kirjasto tue.

Kuvassa 24 näkyy sekvenssikaavio, joka kuvaa rajapinnan käyttöä ja moduulin sisäistä toimintaa. Kuvassa on esitelty ainoastaan osa rajapinnasta. Tässä ei ole tarkoitus käydä koko rajapinnan toimintaa läpi, vaan ainoastaan antaa mielikuva esimerkin avulla siitä, miten se on toteutettu. Kuvassa näkyvät funktiokutsut vastaavat todellisuutta User Interface -moduulin, joka on kuvassa lyhennetty UI, btnLue_clicked-funktion ja RFID communication -moduulin, joka on kuvassa lyhennetty RFID, Reader-rajapinnan osalta. RFID communication -moduulin sisäisiä funktiokutsuja on huomattavasti yksinkertaistettu. Nämä kutsut ovat siis reader-rajapinnan ja librfid-kirjaston välisiä kutsuja.



Kuva 24. Yhden tagin lukuprosessi.

RFID communication -moduulin ulkopuolelle yhden tagin lukeminen näyttää siis juuri kuvan mukaiselta. Kuvan btnLue_clicked-metodia kutsutaan, kun starteri-moduulin Lue-nappia on klikattu (kts. Liite 1, Toiminnallinen määrittely). Ensin alustetaan lukija. Tä-

män jälkeen kutsutaan tagin luku-funktiota. Lukijan alustusvaiheessa on määritelty kuinka kauan tagia yritetään lukea eli kauanko tagia etsitään. Funktio ei siis palaudu ennen kuin tagi on saatu luettua tai alustettu tagin etsintäaika on mennyt umpeen. Tämä mahdollistaa sen, että klikattaessa Lue-nappia ei tagin tarvitse olla heti lukuetaisyysdellä, vaan lukija voidaan siirtää lähemmäs tagia. Kuitenkin, jos lukija on tarpeeksi lähellä tagia, ohjelman suoritus jatkuu välittömästi.

RFID communication -moduulin reader-rajapinnan ja librfid-kirjaston välisten kutsujen yksinkertaistamisesta voisi esimerkkinä ottaa Read_tag-kutsun. Luettaessa esim. Philipsin Mifare Classic tagia (kts. luku 2.5.1) tapahtuu rajapinnan ja kirjaston välillä paljon enemmän kuin vain yksi kutsu. Mifare Classicin muisti, kun puhutaan 4 kilotavun tagista, on jaettu 32 sektoriin, joissa on 4 lohkoa ja 8 sektoriin, joissa on 16 lohkoa. Yhdessä lohossa on 16 tavua. Luettaessa tagia ensimmäisen sektorin ensimmäisestä lohokosta löytyy tagin tunniste ja muut tagin tiedot, jotka ovat kirjoitussuojattuja. Ensimmäisen sektorin toisesta lohokosta alkaa varsinainen muisti. (Philips 2002.) Tagille tallennettua muistia luettaessa tarvitsee se lukea lohko kerrallaan. Yksi Read_tag-kutsu ei kata kaikkien Mifare Classic tagin lohkojen lukua. Yksi luku lukee ainoastaan yhden lohkon. Rajapinta osaa askeltaa lohkot ja sektorit läpi tagille tallennetun tiedon hankkimiseksi.

7.4.2 WebServices

WebServices-moduulin rajapinnan taakseen piilottama gSOAP-kirjaston toiminnallisuus luodaan vastaamaan palvelimen WSDL-kuvaustiedostoa gSOAP-kirjaston apuohjelmien avulla. WSDL-kuvaustiedosto on xml-tiedosto, joka kuvaa palvelimen WebServices -rajapinnan menet, WebServices-rajapinnan sijainnin ja sen miten metodeita käytetään. (W3Schools 2007a.)

Palvelimen WSDL-tiedostosta luodaan ensin C-kielen otsikkotiedosto käyttäen gSOAP-kirjaston mukana tulevaa wsdl2h-ohjelmaa. Tästä otsikkotiedosta saadaan luotua tarvitta-

vat tietorakenteet ja metodit käyttäen soapcpp2-ohjelmaa. Luodut ohjelmakooditiedostot linkitetään rajapintaan ja rajapinnan metodit toteutettiin käyttäen apuna luotuja gSOAP-kirjaston metodeja. gSOAP-kirjasto voidaan kääntää ohjelman sisälle, joten mitään ylimääräisiä asennuksia ei kohdejärjestelmään tarvitse gSOAP-kirjaston käyttöönottamiseksi tehdä. Webservice-liittymän kautta palvelimelta tulevat viestit ovat xml-formaatissa. Tästä syystä tämä moduuli sisältää myös xml-viestin parseroinnin c-kielen rakenteellisiksi tietorakenteiksi. Rajapinnan käyttäjä saa siis palautteena pyytämänsä tiedot rakenteellisina tietorakenteina.

7.4.3 User Interface

Käyttöliittymän näkymät luodaan käyttöliittymän suunnittelutyökalulla Gazpacho (kts. kohta 7.1.3), joka luo käyttöliittymästä xml-tiedoston. Tämä xml-tiedosto luetaan ohjelman suorituksen aikana libglade-kirjaston avulla. Kirjasto muodostaa xml-tiedostosta näkymien GTK- ja Hildon-widgetit. Kuten luvussa 7.3.3 mainitaan, kaikki ohjelman ikkunat ovat HildonWindow-widgetin instansseja. Jostakin syystä libglade-kirjasto ei tue HildonWindow-widgettiä. Kaikki ikkunat on siis rakennettu GtkWindow-widgettiin perustuen. Sen jälkeen, kun käyttöliittymän widgetit on libglade-kirjastolla luotu, korvataan GtkWindow-widgettit HildonWindow-widgeteillä.

Käyttöliittymänäkymän tapahtumiin reagoidaan callback-funktioilla, jotka sidotaan käyttöliittymän komponentteihin joko automaattisesti libglade-kirjaston widgettien luonnin yhteydessä tai erillistä g_signal_connect-funktiota käyttäen. Kunkin näkymän callback-funktiot on paketoitu näkymää edustavaan moduuliin.

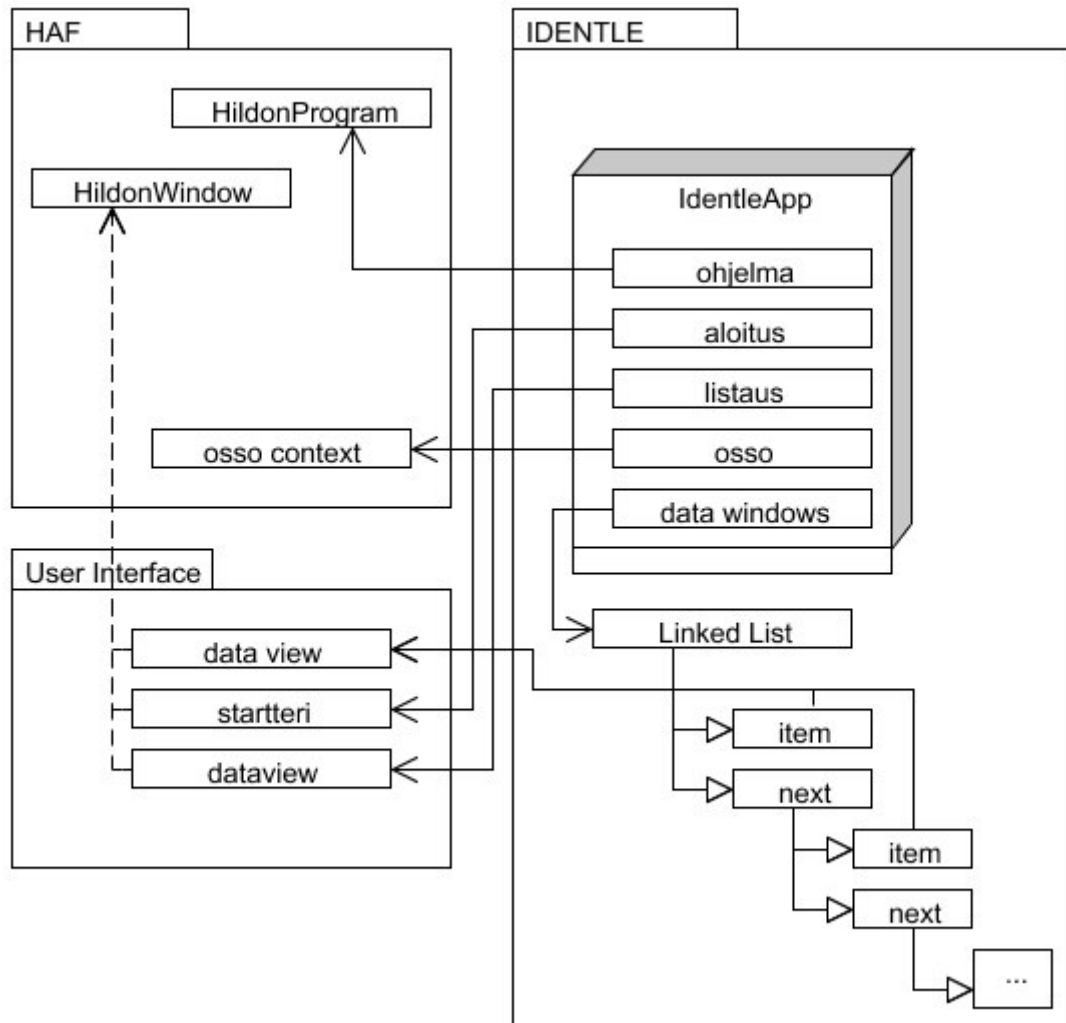
```
typedef struct DataView_{
    HildonWindow *window;
    GladeXML *xmlGladeUI;
    struct communication_data *rfid_data;
    struct koetulokset *pvk_kokeet;
    struct koetulokset *inr_kokeet;
}DataView;
```

Kuva 25. Ohjelmalistaus: dataview-rakenne

User Interface-moduulin sisältämistä alimoduuleista mielenkiintoisin on dataview-moduuli. Se on moduuli, joka edustaa näkymää, jossa näytetään tagiin liitetyn potilaan tiedot. Kuvassa 25 on esitelty moduulin sisältämä tietorakenne, joka on keskeisessä asemassa moduulin funktioiden kutsuissa ja toiminnoissa. Ensimmäisenä on viittaus HildonWindow-widgettiin. Toisena on viittaus libglade-kirjastolla luotuun widgettien kokoelmaan. Näillä kahdella viittauksella päästään käsiksi käyttöliittymänäkymän widgetteihin. Koska näkymän tiedot pohjautuvat tagiin, on luontevaa, että myös tiedot tagista säilytetään moduulin sisällä. Tietorakenteen kolmas muuttuja on viittaus communication_data-tietorakenteeseen. Tämä tietorakenne pitää sisällä mm. tagin tunnusteen ja muistin sisällön. Tagin tunnusteen avulla voidaan suorittaa kyselyitä potilaan tiedoista palvelimelta web services-moduulin kautta. Kaksi viimeistä viittausta ovat taulukoita, joihin on talletettu testiskenaariossa käytettyjen kahden erilaisen verikokeen tiedot. Näihin kumpaankin taulukkoon voidaan hakea palvelimelta viittaukset kyseisen verikokeen muuttamaan viimeisimpään tulokseen. Nämä tulokset on listattu näkymän valikossa koetuloksen ottamishetken mukaan eriteltyinä. Valittaessa koetulos haetaan kokeen tulokset palvelimelta ja esitetään käyttäjälle.

7.4.4 IDENTLE

Merkittävin IDENTLE-moduulissa toteutettu asia on ohjelmiston perusrakenteen määrittely ja siihen liittyvien perustoimintojen toteutus. Kuvassa 26 on esitelty ohjelmiston perusrakenne. Perusrakennetta kuvaa laatikko, joka on nimetty IdentleApp-nimellä. Kuvassa on merkitty katkoviivalla perimistä kuvaavat yhteydet ja yhtenäisellä viivalla viittaukset toteutuksiin. Ylemmän tason moduulit ovat kuvassa näkyvissä moduulijaon hahmottamiseksi. Hildon application framework, lyhennettynä HAF, on kuvassa selkeyden vuoksi näkyvissä omana moduulinaan.



Kuva 26. Ohjelman perusrakenne.

Perusrakenteessa on viittaus HildonProgram-widgettiin. HildonProgram on olio, joka edustaa Hildon frameworkissa ajettavaa ohjelmaa. Se tarjoaa siihen liitetyille ikkunoille yhteisiä ominaisuuksia, kuten esimerkiksi valikon ja työkalupalkit, jotka ovat samat kaikissa HildonProgram instanssiin liitetyissä ikkunoissa. (Maemo-yhteisö 2007e.) Tämän widgetin ansiosta voidaan yhtenäistää ohjelmien eri ikkunoiden näkymiä.

Tämä lisäksi perusrakenteessa ovat viittaukset User Interface -moduulin startteri- ja list-view-alimoduuleihin, jotka edustavat ohjelman ikkunanäkymää. Näistä ikkunoista on yhdessä ohjelmassa ainoastaan yksi esiintymä. Viittaukset näihin esiintymiin säilytetään ohjelman perusrakenteessa, jotta niihin päästään käsiksi eripuolilta ohjelmaa. Dataview-moduulin edustama ikkuna on ikkuna, josta voi periaatteessa olla määrittelemätön määrä esiintymiä yhdessä ohjelmassa. Tästä syystä viittaukset kyseiseen moduulin säilytetään linkitetystä listassa. Viittaus tähän listaan säilytetään ohjelman perusrakenteessa. IDENTLE-moduuli sisältää linkitetyn listan toteutuksen ja listan käsittelyyn tarkoitetut apufunktiot. Apufunktioihin kuuluu uuden moduulin lisääminen listaan, moduulin poistaminen listasta, listan muistivarauksen vapauttaminen sekä aktiivisen eli näkyvässä olevan dataview-moduulin hakeminen listasta.

7.5 Ohjelmistotestaus

Perusmoduulitestausta helpotti huomattavasti se, että yksittäisiä moduuleja voitiin sinälään testata tavalliselle pöytäkoneelle asennetulla Linux-käyttöjärjestelmällä. Esimerkiksi RFID communication -moduuli ohjelmoitiin ensin pöytäkoneelle. Moduulille tehtiin testiohjelma ja moduulin toiminnallisuutta testattiin pöytäkoneella. Tämän jälkeen kyseistä moduulia ei enää testattu erillisenä moduulina itse laitteella. Vasta integraatiotestausvaiheessa suoritettiin testauksia laitteella. Näissä testauksissa varmistui se, että moduulin toiminnallisuus ei muuttunut vaikka se käännettiin ja suoritettiin eri laitealustalla. Samantapaisen testausprosessin kävi läpi myös Webservices-moduuli. Nämä molemmat moduulit oli mahdollista testata ensin pöytäkoneella.

User Interface- moduulin moduulitestausta ei ollut mahdollista suorittaa pöytäkoneen ympäristössä johtuen siitä, että käyttöliittymän toteutuksessa käytettiin ainoastaan Nokia 770:n alustalle toteutettuja käyttöliittymäkomponentteja. Pääasiassa moduulitestaukseen kuitenkin käytettiin varsinaisen laitteen sijasta pöytäkoneella emuloitua ympäristöä. Koska User Interface -moduuli on toiminnoissaan riippuvainen sekä RFID communication että Webservices-moduuleista, tehtiin molemmista moduuleista ns. tynkäversiot. Kun varsinaiset moduulit oli saatu toteutettua ja testattua, liitettiin ne mukaan projektiin. Makroja apuna käyttäen tehtiin mahdolliseksi joko tynkämoduulin tai varsinaisen moduulin käyttäminen. Tällä ratkaisulla pyrittiin nopeuttamaan testausta.

Integraatiotestaus suoritettiin alustavasti emuloidussa ympäristössä. Oikeastaan kaikki muut ohjelmiston osa-alueet kyettiin testaamaan integrointitestausvaiheessa emuloidussa ympäristössä paitsi RFID communication-moduuli. Näin siksi, että USB-liitäntää lukijan ja pöytäkoneen emuloidun ympäristön kanssa ei saatu toimimaan. Täydellinen integraatiotestaus suoritettiin siis varsinaisella laitteella.

Järjestelmän kokonaisvaltainen testaaminen suoritettiin testiskenaarion avulla. Järjestelmätestaamisella pyrittiin löytämään vielä jäljelle jääneet ohjelmiston merkittävimmät ongelmat. Järjestelmän testaamisesta on kerrottu enemmän luvussa 8.

8 KOKONAISTESTAUS

8.1 Testiskenaario

Testiskenaariolla tässä yhteydessä tarkoitetaan testiympäristössä ennalta määrätyn käyttötapausten suorittamista. Testiskenaariolla pyritään havainnollistamaan mobiilin RFID-lukijan toimintaa ja sen soveltuvuutta käyttötapausten kuvaamaan tarkoitukseen. Testiskenaarion aikana otetut kuvankaappaukset esitetään Liitteessä 2.

Testiympäristö rakennettiin Satakunnan ammattikorkeakoulun Tekniikka Rauman tiloihin. Testiympäristö koostui palvelimesta, langattomasta WLAN-verkosta sekä IDENTLE:stä. Palvelimena toimi Internet Information Server, johon tehtiin MySQL-tietokanta sekä Webservice rajapinta C#-ohjelmointikielellä. Tietokantaan tallennettiin potilastietoja, ja Webservice rajapinta tehtiin tarjoamaan kyseisiä potilastietoja rajapinnan asiakkaille. Testiympäristön langaton WLAN-verkko tarjosi verkkoyhteyden mobiilille RFID-lukijalla mahdollistaen lukijan Webservice-kyselyt palvelimelta.

Testiskenaarion ensimmäinen käyttötapaus käsittää yhden tagin lukemisen ja siihen liittyvien oheistoimintojen suorittamisen. Käyttötapausten on tarkoitus jäljitellä potilaan tunnistamista tagin avulla ja perustietojen selailua. Potilaan rintataskuun taikka ranteeseen on kiinnitetty mifare-standardin mukainen RFID-tagin. Potilaan tiedot on tallennettu testiympäristöpalvelimeen, ja tagin tunnistetiedon ja potilaan tietojen välille on luotu referenssi. Yksinkertaistettuna käyttötapausten toiminnalliset vaiheet ovat seuraavat:

- Käynnistetään mobiilin RFID-lukijan ohjelmisto (kts. Liite 2. Kuva 1.)
- Käynnistetään RFID-lukija virtakytkimestä
- Painetaan Lue-nappia ja siirretään lukija potilaan tagin lukuetaisyysdelle.
- Kun tagi on luettu, suljetaan RFID-lukija virtakytkimestä
- Selailaan potilaan perustietoja ja haetaan palvelimelta potilaan verenhyytymisar-

vokokeen viimeisin tulos. (kts. Liite 2. Kuva 2. & 3.)

- Suljetaan ohjelmisto.

Toinen käyttötapaus ei eroa ensimmäisestä käyttötapauksesta, muuten kuin että Lue-napin painamisen sijasta painetaan Etsi-nappia. Tämä käynnistää ohjelmistossa monen tagin lukemistoiminnon. Yhden tagin lukemisen sijasta luetaan kaksi tagia. Tagin tunnistetta vastaavan perustietoja tarkastellaan samalla tavoin kuin ensimmäisessä käyttötapauksessa.

8.2 Testaustulosten tarkastelu

Luvussa 5 esitetyt toimintavaatimukset ja testiskenaariosuunnitelman kautta luotiin ympäristö, johon IDENTLE oletetusti joutuisi työkäytössä. Testiaika oli lyhyt, joten todellista käyttöikää testissä ei todettu. Testiympäristö on testaukselle ihanteellinen, koska häiriötekijät oli pyritty minimoimaan. Laitteen toimivuus todellisessa käyttöympäristössä jää vielä myöhemmin todettavaksi.

Testaus käytiin läpi askel askeleelta. Huomiot sekä mahdolliset kehitystarpeet kirjattiin. Laite toimi asetettujen vaatimusten mukaisesti ja tulokset todettiin hyväksyttäviksi ja luotettaviksi. Laitteen ohjelmistossa havaittiin muutamia epäjatkuvuuksia ja käytännön luoketäisyydeksi todettiin noin 10 cm. Vaatteilla ei todettu olevan huomattavaa merkitystä laitteen kykyyn tunnistaa luettava kohde.

Laitteen käyttäminen osoittautui vaatimusten mukaiseksi:

- Käyttö yhdellä kädellä on mahdollista.
- Virtakytkin ei ole tiellä, mutta tarvittaessa ulottuvissa.
- Pariston nopea vaihto vaatii laitteen tuntemista, mutta on riittävän ohjeistuksen jälkeen mahdollista.

9 JATKOKEHITYS

9.1 IDENTLE

Laitteen jatkotutkimukseen ja -kehitykseen tulisi panostaa sekä ohjelmiston että raudan osa-alueilla. IDENTLE:n protoversion kehityksen yhteydessä tuli uusia toteuttamisisideoita, joihin projektin aikataulu ei riittänyt. Alla on lueteltu niistä oleellisimmat.

IDENTLE:n teholähde toi esiin monta kysymystä. Olisiko mahdollista asentaa analoginen komparaattori tutkimaan 9V:n pariston jännitetasoa? Vertailujännitteet saataisiin 7805 jänniteregulaattorin reg-outputista ja suoraan paristosta. Kun jännite laskee alle seitsemän ja puolen voltin, antaa mikroprosessori hälytyksen. Hälytys näkyisi käyttäjälle ”vaihda paristo” -tekstinä. IDENTLE:n ohjelmisto kyselisi hälytysrekisterinarvoa määrättyin väliajoin.

Olisiko mahdollista kehittää laitteeseen erilainen akkuratkaisu? Kytkemällä AA-paristoja sarjaan ja täten saataisiin nostettua kokonaisvirtakapasiteettia. Tästä seuraa tietysti pidempi käyttöikä. Kotelon muotoa tulisi uudelleen harkita paristojen luoman tilatarpeen vuoksi.

USB power injector yllätti haasteellisuudellaan. Kotelointi ja osien pakkaaminen tiiviisti toi esiin aspekteja, joita ei kuvitellut power injectoria Ultiboardilla suunniteltaessa. Piirilevyn layout vaatisi uudelleen suunnittelua, jossa otettaisiin huomioon RFID-lukijan liittäntä, tarkemmin sen sijainti ja kaapelin jäykkyyden aiheuttamat haasteet. Myös häiriösuojaan maadoitus tulisi suunnitella jo piirilevyä suunnitellessa. Olisi olennaista saada levyille yhtenäinen maapotentiaalialue.

Nokia 770 Internet Tabletin ja IDENTLE:n välinen USB-kaapeli tulisi lyhentää ja saada mahdollisesti kokonaan kotelon sisään. IDENTLE:n ensimmäisessä versiossa kaapeli osoittautui epäkäytännölliseksi ja tiellä olevaksi.

IDENTLE:n lukuetaisyyttä tulisi parantaa. Olimme luottavaisia, että OpenPCD-lukijan kehittäjät saisivat koodattua lukijan firmwaresta version, joka tukisi ISO 15693 -standardia. Tämä teoriassa mahdollistaisi 0.5 - 1.5 metrin lukuetaisyyden. Toinen kokeilemisen arvoinen idea voisi olla oman, alkuperäistä suuremman antennin kehittäminen. OpenPCD-lukijaan on mahdollista kytkeä ulkoinen antenni. Uusi antenni tulisi IDENTLE:n koteloon, ja näin antennille saataisiin suurempi halkaisija.

Ohjelmistossa on vielä paljon asioita, joita voi parantaa ja lisätä. IDENTLE:n ohjelmiston muutamit löydettyt epäjatkuvuusongelmat tulisi korjata. Sinällään ohjelmisto on toimintoiltaan tällä hetkellä melko suppea. Uusien ominaisuuksien lisääminen ohjelmistoon olisi siis paikallaan. Nokia 770 ei ole ainut laite, joka pyörii Maemo-alustalla. IDENTLE:n nykyinen ohjelmisto on tehty Maemo versioon 2.1. Uusin Maemon versio on kuitenkin 3.1 (Maemo-yhteisö 2007f). Nokian uusi N800 Internet Tablet on laite, joka toimii Maemon versiolla 3.1. Mielenkiintoista olisi selvittää, miten hyvin käytännössä nykyinen ohjelmisto olisi käännettävissä viimeisimmälle Maemo-alustalle. Periaatteessa sen pitäisi onnistua ilman suuria muutoksia.

Ohjelmiston asennettavuudessa Nokia 770:tä tulisi helpottaa. Kaikki ohjelmistot asennetaan Nokia 770 -laitteelle Debian-paketteina. IDENTLE:n ohjelmisto on pakattuna Debian-paketiksi, mutta ohjelmiston tarvitsemaa librfid-kirjastoa ei ole paketoitu Nokia 770:tä varten. Se on varta vasten tätä projektia silmällä pitäen käännetty Nokia 770-laitteelle ja asennettu käsin. Kirjaston paketointi helpottaisi ohjelmiston asennusta.

Käyttöliittymän toteutuksessa voisi tutkia Pythonin käyttöä. Ohjelmiston tarkoituksena on olla helposti muokattavissa oleva. Pythonin käyttö käyttöliittymän toteutuksessa saattaisi helpottaa helposti muokattavuuden saavuttamista.

9.2 Aktiivitagisovellukset

Aktiivitagilla tarkoitetaan RFID-tunnistetta, joka sisältää virtalähteen. Virtalähteen ansiosta tagin lukuetaisyttä voidaan kasvattaa, taikka sen toiminnallisuutta voidaan lisätä liittämällä siihen esim. lämpötilaa mittaava anturi. Aktiivitageilla IDENTLE:n toiminnallisuutta voidaan huomattavasti laajentaa pelkästä tunnistamisesta myös tietojen reaaliaikaiseen keräämiseen ja analysointiin.

Terveydenhuollossa aktiivitageja voidaan soveltaa esimerkiksi siten, että potilaalle istutetaan peruselintoimintoja, kuten pulssia seuraava tai tallentava aktiivitag. Pulssia tallentava tagi tekisi omatoimisesti mittauksia, jonka tulokset se tallentaisi omaan muistiinsa. Nämä tulokset olisivat sitten luettavissa IDENTLE:n avulla. Pulssia seuraava aktiivitag ei tallenna tuloksiaan, vaan luettaessa tagia IDENTLE:n avulla tagi näyttäisi potilaan sen hetkisen pulssin.

Logistiikan yksi sovellusmahdollisuus voisi olla kylmäkontin hetkellisen lämpötilan tarkastaminen ja lämpötilavaihteluiden seuraaminen sekä tutkiminen koko kuljetushistorian ajalta käyttäen IDENTLE:ä tagin lukemiseen.

10 YHTEENVETO

Projektin tavoitteena oli kehittää kannettavan RFID-lukijan prototyyppi teollisuuden ja terveydenhuollon eri osa-alueiden käyttöön. Tarkoituksena oli yhdistää RFID-lukija älykkääseen yhdellä kädellä käytettävään laitteeseen. Markkinoilla on lukuisia kannettavia RFID-lukijoita rajoitetulla suorituskyvyllä. Laitteemme kilpailee näiden tuotteiden kanssa päätelaitteen monipuolisuudella ja vuorovaikutusominaisuuksilla.

Projekti koostuu kahdesta pääosasta, IDENTLE:stä ja Nokia 770 Internet Tabletista. IDENTLE käsittää kotelon, johon Nokia 770 on liitettävissä. Kotelon sisälle on sijoitettu OpenPCD-lukija, paristo ja virransyöttöyksikkö. Nokia 770 valittiin päätelaitteeksi pääasiassa sen suuren kosketusnäytön, ohjelmointialustan ja tietoliikenneominaisuuksien vuoksi.

OpenPCD-lukijan valintaan päädyttiin sen edullisuuden ja avoimen ohjelmistokirjaston takia. Näiden lisäksi lukijan fyysiset mitat mahdollistivat helpon koteloinnin Nokia 770 kanssa ja se oli liitettävissä Nokia 770 Internet Tabletiin USB:llä.

IDENTLE:n kehityksessä haluttiin korostaa käytettävyyttä ja informatiivisuutta. Tämä loi laitteiston kehitykselle haasteita koon, painon ja materiaalin suhteen. Nokia 770 Internet Tablet asetti laitteen fyysiset rajat. IDENTLE:n komponentit oli saatava mahtumaan Nokian suojakuoreen, ja kokonaisuus oli koteloitava tavalla, joka istuu hyvin käyttäjän käteen. Kaksi päätekijää, RFID-lukija ja Nokia 770, eivät olleet sellaisenaan yhdistettävissä. Molemmat laitteet odottavat ulkoisia käskyjä, eivät luo niitä. RFID-lukija ottaa käyttövirransa USB-portista, jota 770 ei oletuksena kykene antamaan. Väliin oli luotava ratkaisu, joka muuntaa ulkoisesta paristosta USB-portille oikean jännitteen ja huolehtii laitteiden välillä tapahtuvasta datan siirrosta. Haastavinta kuitenkin oli saada kaikki kolme osaa koteloitua näyttämään yhtenäiseltä Nokia 770:n kanssa. Kotelo tehtiin lasikuidusta, joka osoittautui hyväksi ratkaisuksi materiaalin muokattavuuden vuoksi. Käteen istuvuus var-

mistettiin tartuntapuolen pyörityksellä. Kotelo saatiin kiinnitettyä onnistuneesti Nokia 770 suojakuorta vasten ja laitteen käyttö on kohtalaisen helppoa.

Ohjelmiston haasteet painottuivat testiskenaarion toiminnallisuuksien toteuttamiseen. Nokia 770 -ohjelmiston kehitysympäristön asentaminen ja itse ohjelmointiympäristön opiskelu olivat projektin haastavimmat osa-alueet ohjelmiston osalta. Haastetta uuden ohjelmistoympäristön lisäksi toi viimeisimpien teknologioiden käyttö, kuten palvelin kommunikatioissa käytetty WebServices- ja RFID-teknologiaa itsessään.

Ohjelmiston toteutuksessa nojaututtiin valmiisiin komponentteihin. Valmiiden komponenttien käytön suurimpana syynä oli se, että projektin ohjelmisto on laajuudeltaan niin suuri, ettei sen toteuttaminen kokonaisuudessaan olisi ollut mahdollista käytettävissä olevassa ajassa. Toisaalta valmiiden komponenttien käyttöönotto vaati opettelua, joka puolestaan vei aikaa. Annetussa ajassa kuitenkin ehdittiin toteuttaa ohjelmiston perustoiminnot, kuten projektin toiminnallisessa määrittelyssä esitetään. Ohjelmisto pyrittiin rakentamaan modulaariseksi, jotta sen käyttöönottaminen testiskenaariorista eroavassa ympäristössä olisi mahdollista mahdollisimman pienellä määrällä ohjelmointia. Modulaarisuuden toteuttamisessa onnistuttiin kohtalaisesti. Käytetyt tekniikat helpottavat omalta osaltaan käyttöönottoa erilaisissa ympäristöissä. Esimerkiksi palvelinkommunikaatioon käytetty WebServices kasvattaa koko ajan suosiotaan järjestelmien välisenä tietoliikenne standardina. Näin voi olettaa, että tulevaisuudessa järjestelmä jonka osaksi IDENTLE tulee saattaa jo käyttää WebServices-palvelin rajapintana.

Projektissa onnistuttiin toteuttamaan perusominaisuuksiltaan toimiva mobiili RFID-lukija, joka oli tavoitteiden mukaisesti helposti käytettävä. Testiskenaariossa lukija täytti sille projektin alussa osoitetut tavoitteet.

LÄHTEET

- Bashan, O. ISO 14443: An introduction to the contactless standard for smart cards and its relevance to customers [verkkodokumentti]. [Viitattu 18.4.2007]. Saatavissa: <http://www.otiglobal.com/objects/ISO%2014443%20WP%204.11.pdf>
- Bitmanufaktur 2007. Yritysesittely [verkkodokumentti]. [Viitattu 19.3.2007]. Saatavissa: <http://www.bitmanufaktur.de/>
- Engelen, R. 2007. gSOAP: C/C++ Web Services and Clients [verkkodokumentti]. [Viitattu 11.3.2007]. Saatavissa: <http://www.cs.fsu.edu/~engelen/soap.html>
- Erdfelt, J. 2007. About libusb [verkkodokumentti]. [Viitattu 13.2.2007]. Saatavissa: <http://libusb.sourceforge.net/>
- Finkenzeller, K. 2006. RFID Handbook 2nd Edition [verkkodokumentti]. [Viitattu 3.1.2007]. Saatavissa: <http://www.rfid-handbook.de/rfid/frequencies.html>.
- Garfinkel, S. & Rosenberg, B. 2005. RFID Applications, Security, and Privacy. Addison-Wesley.
- Glover, B. & Bhatt, H. 2006. RFID Essentials. O'Reilly.
- Gnome-kehittäjäyhteisö 2007. GLib Reference Manual: GLib Overview [verkkodokumentti]. [Viitattu 16.3.2007]. Saatavissa: <http://developer.gnome.org/doc/API/2.0/glib/glib.html>
- Haikala, I. & Märijärvi, J. 2000. Ohjelmistotuotanto. Satku.
- Heesch, D. 2007. Source code documentation generator tool [verkkodokumentti]. [Viitattu 15.3.2007]. Saatavissa: <http://www.stack.nl/~dimitri/doxygen/index.html>
- Littler, J. 2006. Linux on the Nokia 770 Internet Tablet [verkkodokumentti]. LinuxDevCenter [viitattu 5.2.2007]. Saatavissa: <http://www.linuxdevcenter.com/pub/a/linux/2006/07/20/nokia-770.html>
- Myerson, J. M. 2007. RFID in the Supply Chain. New York. Auerbach Publications
- Maemo-yhteisö 2007a. Repository [verkkodokumentti]. Maemo.org [viitattu 23.1.2007]. Saatavissa: <https://garage.maemo.org/>.
- Maemo-yhteisö 2007b. Maemo 2.1 tutoriaali [verkkodokumentti]. Maemo.org [viitattu 16.3.2007]. Saatavissa: http://maemo.org/platform/docs/tutorials/Maemo_tutorial.html

Maemo-yhteisö 2007c. Maemo Development Platform: White paper [verkkodokumentti]. Maemo.org [viitattu 16.3.2007]. Saatavissa: http://maemo.org/platform/docs/maemo_exec_whitepaper.html

Maemo-yhteisö 2007d. HildonWindow API [verkkodokumentti]. Maemo.org [viitattu 14.3.2007]. Saatavissa: <https://stage.maemo.org/svn/maemo/projects/haf/doc/api/hildon-libs/hildon-libs-HildonWindow.html>

Maemo-yhteisö 2007e. HildonProgram API [verkkodokumentti]. Maemo.org [viitattu 14.3.2007]. Saatavissa: <https://stage.maemo.org/svn/maemo/projects/haf/doc/api/hildon-libs/hildon-libs-HildonProgram.html>

Maemo-yhteisö 2007f. Maemo is the application development platform for Nokia Internet Tablet products [verkkodokumentti]. Maemo.org [viitattu 18.4.2007]. Saatavissa: <http://www.maemo.org/>

Meriac, M. 2006. OpenPCD & PICC [verkkodokumentti]. [Viitattu 19.3.2007]. Saatavissa: <http://www.openpcd.org/dl/foss.in-2006.pdf>

Movial 2007. Scratchbox homepage [verkkodokumentti]. [Viitattu 15.2.2007]. Saatavissa: <http://www.scratchbox.org/>

Nokia 2005a. Nokia 770 Internet Tablet: Technical specifications [verkkodokumentti]. [Viitattu 12.1.2007]. Saatavissa: <http://europe.nokia.com/A4145105>

Nokia 2005b. A new product category - The Nokia 770 Internet Tablet [lehdistöjulkaisu]. Nokia Oyj 07.11.2005 [viitattu 23.1.2007]. Saatavissa: http://press.nokia.com/PR/200511/1019865_5.html

Nokia 2007. Nokia 770 Internet Tablet: Perusominaisuudet [verkkodokumentti]. [Viitattu 5.2.2007]. Saatavissa: <http://www.nokia.fi/puhelimet/puhelinmallit/770/perusominaisuudet.html>

NXP Semiconductors 2007. [Verkkodokumentti]. [Viitattu 21.3.2007]. Saatavissa: <http://www.nxp.com/>

OpenPCD:n kehittäjät 2007a. OpenPCD.org a free 13.56MHz RFID reader & writer design [verkkodokumentti]. [Viitattu 19.3.2007]. Saatavissa: www.openpcd.org

OpenPCD:n kehittäjät 2007b. OpenPCD wiki [verkkodokumentti]. [Viitattu 10.3.2007]. Saatavissa: http://wiki.openpcd.org/wiki/Main_Page

OpenPCD:n kehittäjät 2007c. main_libbrfid [verkkodokumentti]. [Viitattu 10.3.2007]. Saatavissa: http://wiki.openpcd.org/wiki/Main_libbrfid

OpenPCD:n kehittäjät 2007d. main_mifare [verkkodokumentti]. [Viitattu 10.3.2007]. Saatavissa: http://wiki.openpcd.org/wiki/Main_mifare

OpenPCD:n kehittäjät 2007e. main_dumbreader [verkkodokumentti]. [Viitattu 10.3.2007]. Saatavissa: http://wiki.openpcd.org/wiki/Main_dumbreader

Philips 2002. Datasheet: Mifare Standard 4kByte Card IC MF1 IC S70 [verkkodokumentti]. [Viitattu 20.3.2007]. Saatavissa: http://www.nxp.com/acrobat_download/other/identification/m043531.pdf

RFIDLab Finland 2007 [verkkodokumentti]. [Viitattu 22.2.2007]. Saatavissa: <http://www.rfidlab.fi>

Rinta-Runsala, E. & Tallgren, M. 2004. RFID -tekniikan hyödyntäminen asiakkuudenhallinnassa [verkkodokumentti]. [Viitattu 2.1.2007]. Saatavissa: <http://virtual.vtt.fi/inf/julkaisut/muut/2004/rfid-raportti.pdf>.

Salminen, K. 2005. Karoliina's blog [verkkodokumentti]. [Viitattu 13.2.2007]. Saatavissa: http://www.karoliinasalminen.com/blog/?page_id=68

W3Schools 2007a. Introduction to Web Services [verkkodokumentti]. [Viitattu 14.3.2007]. Saatavissa: http://www.w3schools.com/webservices/ws_intro.asp

W3Schools 2007b. Introduction to SOAP [verkkodokumentti]. [Viitattu 11.3.2007]. Saatavissa: http://www.w3schools.com/webservices/soap_intro.asp

Welte, H. 2006. What is librfid [verkkodokumentti]. [Viitattu 10.3.2007]. Saatavissa: <http://openmrd.org/projects/librfid/>

Wikipedia. 2007a. Universal Serial Bus [verkkodokumentti]. [Viitattu 3.12.2006]. Saatavissa: <http://en.wikipedia.org/wiki/USB>.

Wikipedia. 2007b. ISO 15693 [verkkodokumentti]. [Viitattu 3.12.2006]. Saatavissa: http://en.wikipedia.org/wiki/ISO_15693.

KUVA-, TAULUKKO- JA KUVIOLUETTELO

Kuva 1	RFID-systeemi
Kuva 2	RFID-tunnisteita
Kuva 3	Sähkömagneettinen induktio
Kuva 4	Nokia 770 Internet Tablet
Kuva 5	Ohjelmistoalustan rakenne
Kuva 6	OpenPCD-lukija
Kuva 7	OpenPCD-lukijan antenni
Kuva 8	OpenPCD-lukijan ISO 14443 taajuusspektri tagi-lukija kommunikoinnissa
Kuva 9	IDENTLE:n toiminnan yleiskuvaus
Kuva 10	IDENTLE:n koon määrittävät komponentit
Kuva 11	USB- ja USB-mini-B-liittimen lankajärjestys
Kuva 12	USB power injectorin piirikaavio
Kuva 13	USB power injectorin lopullinen layout Ultiboard 9:llä
Kuva 14	Jyrsitty ja käyttövalmis power injector
Kuva 15	Testauskytkennät alustavassa testauksessa
Kuva 16	Nokia 770 suojakuorimuokkaus ja IDENTLE:n laitteistomitoitus
Kuva 17	IDENTLE:n kotelon ensivaihe
Kuva 18	IDENTLE:n kotelon istuvuus Nokia 770 suojakuorta vasten
Kuva 19	IDENTLE:n kotelon kiinnityskohdat suojakuoreen
Kuva 20	IDENTLE
Kuva 21	Testikotelo
Kuva 22	IDENTLE:n ohjelmiston arkkitehtuuri
Kuva 23	RFID communication-moduuli
Kuva 24	Yhden tagin lukuprosessi
Kuva 25	Ohjelmalistaus: dataview-rakenne
Kuva 26	Ohjelman perusrakenne

TAULUKKO 1	RFID-systeemien toimintataajuudet ja -etäisyydet
TAULUKKO 2	IDENTLE-projektin entiteetit ja niiden tehtävät
TAULUKKO 3	USB- ja USB-mini-B-liittimen lankajärjestys
TAULUKKO 4	USB power injectorin piirikaavion komponentit funktioineen
TAULUKKO 5	Alustavan testauksen tulokset
TAULUKKO 6	IDENTLE:n virrankulutus ja käyttöaika-arvio
TAULUKKO 7	IDENTLE:n osien mitat
Kuvio 1	USB power injectorin USB-porteista mitattu jännite

LIITTEET

Liite 1 Päätelaitteen toiminnallinen määrittely

Liite 2 Testiskenaarion kuvan kaappaukset

LIITE 1, Päätelaitteen toiminnallinen määrittely

Toiminnallinen määrittely

IDENTLE:n päätelaitteelle

Mika Impola
Lauri Hartonen

Versio 0.1

SISÄLLYS

1. Johdanto	88
1.1 Tämän dokumentin tarkoitus	88
1.2 Dokumentin laajuus	88
1.3 Tuote	88
1.4 Yleiskatsaus dokumenttiin	89
1.5 Viitteet.....	89
2. Yleiskuvaus.....	90
2.1 Määritelmän näkökulma	90
2.2 Ohjelmiston perustoiminnot.....	90
2.3 Käyttäjät.....	90
2.4 Alusta	91
2.5 Dokumentaatio	91
2.6 Oletukset ja riippuvuudet.....	91
3. Toiminnot.....	93
3.1 Yleiset toiminnot.....	93
3.2 Rfid-tagin lukeminen	93
3.2.1 Yhden tagin lukeminen	93
3.2.2 Useamman tagin lukeminen.....	93
3.3 Palvelinkyselyt.....	94
3.4 Tietojen näyttö	94
3.4.1 Perustietojen näyttö.....	95
3.4.2 Koetulosten näyttö	95
4. Rajapinnat	96
4.1 Käyttöliittymä	96
4.1.1 Aloitusnäkyvä	96
4.1.2 Listausnäkyvä	97
4.1.3 Tietojen esitysnäkyvä	97
4.2 Ohjelmisto rajapinta.....	98

4.2.1 RFID-lukija.....	98
4.2.2 Palvelin kommunikaatio	98
4.3 Verkkoyhteys	99

1. Johdanto

1.1 Tämän dokumentin tarkoitus

Tämä dokumentti on luotu määrittelemään IDENTLE-projektiin kuuluvan päätelaitteen eli Nokia 770 Internet Tabletin perustoiminnallisuudet.

1.2 Dokumentin laajuus

Dokumentti käsittelee ainoastaan IDENTLE-projektin päätelaitteen eli Nokia 770 Internet Tabletin ohjelmiston toiminnallisuudet. Projektin muiden osien toiminnallisuusmäärittelyt ovat niiden omissa toiminnallisissamäärittelyissä.

Määrittely rajoittuu ohjelmistomäärittelyyn.

1.3 Tuote

IDENTLE-projektin päätelaite on mobiili RFID -lukija. Se käyttää hyväksi erillistä lukija yksikköä RFID-tagien lukemiseen. Langattomalla internet yhteydellään se kommunikoi järjestelmän muiden toimijoiden kuten esimerkiksi tietovarastojen kanssa.

Testimielessä käytetään esimerkkiympäristönä sairaalaa. Tämä tarkoittaa sitä, että luettava RFID-tagit on potilaaseen kiinnitetty tunnistet ja että luettava, käsiteltävä ja esitettävä data on potilastietoja ja erilaisten lääketieteellisten kokeiden tuloksia.

1.4 Yleiskatsaus dokumenttiin

Johdannossa kuvataan lähinnä dokumenttia itseään. Toisessa luvussa käydään läpi määrittelyn yleiset tekijät. Luvussa kolme määritellään ohjelmiston toiminnallisuudet. Luvussa neljä kuvataan rajapinnat ja liittymät, joilla ohjelmisto kommunikoi projektin muiden komponenttien kanssa.

1.5 Viitteet

IDENTLE-projektin palvelimen toiminnallinen määrittely on saatavilla tämän projektin dokumentaatiossa.

IDENTLE-projektin RFID -lukijan tiedot: <http://www.openpcd.org/>

Nokia 770 Internet Tabletin tekniset ominaisuudet:

http://www.nokia.fi/puhelimet/puhelinmallit/770/tarkka_tuoteseloste.html

Hildon Application Framework:

http://www.maemo.org/community/UI_Style_Guide_Summary_2.0.pdf

GTK esittely: <http://www.gtk.org/>

Gtkmm: <http://www.gtkmm.org/>

Web services esittely: http://www.w3schools.com/webservices/ws_intro.asp

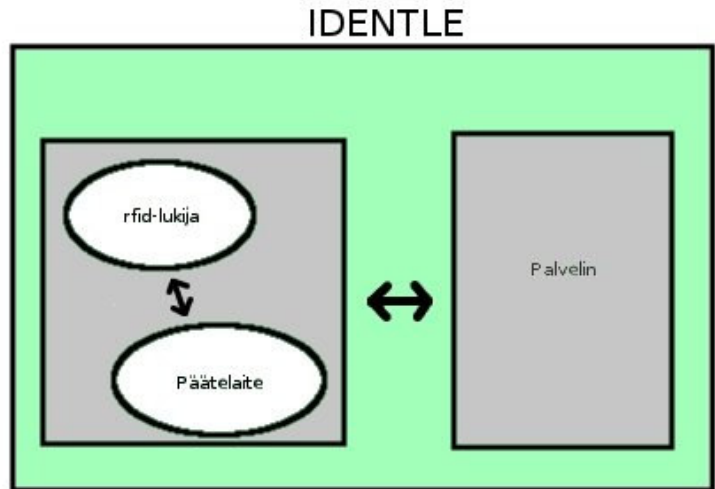
SOAP tietoa: http://www.w3schools.com/soap/soap_intro.asp

WSDL tietoa: http://www.w3schools.com/wSDL/wSDL_intro.asp

2. Yleiskuvaus

2.1 Määritelmän näkökulma

Tämä määritelmä määrittelee IDENTLE-projektin kolmesta pääkomponentista yhden. Projektin komponentteja ovat rfid-lukija, päätelaite ja palvelin. Määritelmä määrittelee päätelaitteen ohjelmiston toiminnallisuudet.



Kaikki komponentit ja määrittelyt tehdään alusta asti, vaikka toteutuksen tulee pohjautua jo valmiisiin avoimen lähdekoodin ratkaisuihin.

2.2 Ohjelmiston perustoiminnot

Tarkoituksena on, että päätelaite kykenee lukemaan rfid-tageja käyttäen lukijakomponenttia. Tämän lisäksi päätelaitteen tulee pystyä kommunikoimaan palvelimen kanssa langattoman yhteyden kautta.

2.3 Käyttäjät

Tässä esimerkkiympäristössä päätelaitteen käyttäjiä ovat terveydenhuollon ja sairaaloiden henkilökunta eli lääkärit ja hoitajat.

2.4 Alusta

Ohjelmiston alustana toimii Nokia 770 Internet Tablet. Nokia 770:ssa on sulautettu Linux käyttöjärjestelmä, joka pohjautuu suosittuun debian-jakelupakettiin. Käyttöjärjestelmä-versiona käytetään tällä hetkellä tuoreinta virallisesti julkaistua versiota, joka on Internet Tablet 2006. Ohjelmiston käyttöliittymän toteuttamiseen käytetään Hildon Application Frameworkiä eli HAF:iä. HAF on rakennettu mm. GTK:n päälle ja Nokia 770 on yhteen sopiva suurimmaksi osaksi GTK:n kanssa. HAF tarjoaa komponentteja käyttöliittymän toteuttamiseen, mutta ei kuitenkaan kaikkia tarpeellisia. Siinä voidaan käyttää siis myös perus GTK:n komponentteja.

Kommunikointi palvelimen kanssa suoritetaan palvelimen tarjoamien web servicien kautta. Rfid -lukijan kanssa kommunikointiin käytetään librfid-kirjastoa.

2.5 Dokumentaatio

Ohjelmistokoodi ja komponentit dokumentoidaan käyttäen apuna doxygen -työkalua. Käyttäjälle tarkoitettuja dokumentteja tehdään tarpeen mukaan.

2.6 Oletukset ja riippuvuudet

Tarkoituksena on käyttää mahdollisimman paljon hyödyksi avoimen lähdekoodin komponentteja.

Tavoitteena on luoda ohjelmakoodista mahdollisimman geneeristä, jotta sen käyttöönotto ja muokkaaminen muunlaiseen toimintaympäristöön olisi mahdollisimman helppoa.

RFID-lukija kommunikointiin käytettävä librfid-kirjasto on avoimen lähdekoodin projekti ja tällä hetkellä aktiivisen kehityksen alla. Ohjelmakoodissa tulee varmistaa yhteenso-

pivuus myös tulevien librfid-kirjaston versioiden kanssa tai ainakin pyrkiä vähentämään tarvittavien muutosten määrää suunnittelemalla sopivia rajapintoja.

Nokia 770:n käyttöjärjestelmä on avoimen lähdekoodin projekti ja jatkuvan kehityksen alla. Käyttöjärjestelmän kehitysversio on nimeltään Sardine. Herring on ominaisuusjäädytettyversio, josta parantelemalla muokataan seuraavia virallisia käyttöjärjestelmäjulkaisuja. Näitä kahta kehitysversiota tulee tarkkailla tätä ohjelmaa kehitettäessä, jottei uuteen käyttöjärjestelmäversioon siirtyminen olisi vaikeaa.

3. Toiminnot

3.1 Yleiset toiminnot

Ohjelman kaikissa näkymissä on valikko. Valikko sisältää ohjelman lopettamis- ja ohjelman perustietojen näyttövalinnat. Tämän lisäksi valikkoon voi lisätä kullekin näkymälle tarkoitettuja toimintoja.

3.2 RFID-tagin lukeminen

Lukeminen aloitetaan painamalla päänäkymässä joko tai useamman tagin lukemiseen tarkoitettua nappia.

Mikäli lukijaa ei löydy, ilmoitetaan asiasta käyttäjälle. Lukijan etsintä ja tagin etsintäprosessien etenemisestä pitää ilmoittaa käyttäjälle.

3.2.1 Yhden tagin lukeminen

Ohjelma lopettaa tagin etsinnän heti sen löydettyään ja aloittaa tarvittavat palvelinkyselyt (kts. kohta 3.3). Tagin lukuaika tulee olla alustettavissa. Mikäli yhtään tagia ei löydy alustetussa ajassa, ilmoitetaan asiasta käyttäjälle.

3.2.2 Useamman tagin lukeminen

Heti tagin lukemisen aktivoinnin jälkeen esitetään listausnäkyvä (kts. kohta 4.1.2). Listausnäkyvään tulostetaan jokaisen löydetyn tagin muistista ihmisen luettava tunnistetie-

toesimerkiksi potilaan nimi. Tagien tulostamista jatketaan, kunnes käyttäjä lopettaa tagien etsinnän painamalla etsinnän lopettavaa nappia listausnäkyvässä.

Tageja ei lueta yhtäjaksoisesti vaan tiheissä sykäyksissä. Sykäyksissä tapahtuva lukeminen säästää virtaa. Lukutiheys tulee olla alustettavissa.

Tietojen näyttöön siirtyminen tapahtuu siten, että käyttäjä valitsee jonkun listatuista tageista ja painaa tietojennäyttö nappia. Listausnäkyvä jätetään avoimeksi. Listausnäkyvä voi siis halutessaan palata ja avata jonkin toisen potilaan tiedot.

3.3 Palvelinkyselyt

Palvelinkyselyt luodaan palvelimen tarjoamasta WSDL-tiedostosta. Tästä johtuen palvelinrajapinnan toteuttamista palveluista saa tietoa palvelimen toiminnallisesta määrittelystä. Palvelinkyselyille tulee kuitenkin luoda rajapinta, joka tarjoaa seuraavat palvelut:

- RFID-tunnistetta vastaavien potilastietojen palautus
- Potilaan laboratoriotulosten listaus
- Käyttäjän valitseman laboratoriotuloksen listaus.

Rajapintaan voidaan toteuttaa muitakin palveluja, jotka palvelin tarjoaa. Näistä tarkemmin palvelimen toiminnallisessa määrittelyssä.

3.4 Tietojen näyttö

Potilastiedot näytetään aina uudessa ikkunassa. Tämä siksi, että haluttaessa voidaan tarkastella useamman kuin yhden potilaan tietoja. Käyttäjä voi siis halutessaan lukea uusia RFID-tunnisteita ja vanhan tunnisteen potilastietojen ikkuna on aina saatavilla esille, mikäli käyttäjä ei sitä ole erikseen sulkenut.

3.4.1 Perustietojen näyttö

Perustietonäytössä esitetään potilaasta ainakin seuraavat tiedot: nimi, henkilöturvatus ilman tunnisteosaa ja osoite. Tämän lisäksi näytetään potilaan kuva sekä haetaan valikkoon potilaan kokeet. Katso ulkonäkömäärittely kohdassa 4.1.3

3.4.2 Koetulosten näyttö

Koetulokset näytetään ns. ponnahdusikkunassa. Käyttäjän tarvitsee painaa ok-nappia ennen jatkamista. Kyseinen ikkuna on modaali koko ohjelmaan nähden.

4. Rajapinnat

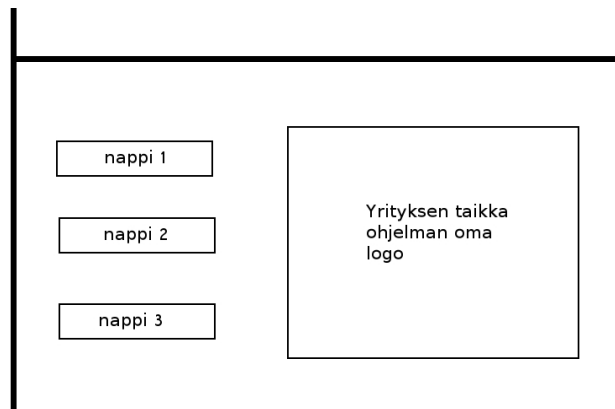
4.1 Käyttöliittymä

Käyttöliittymä toteutetaan käyttäen Hildon Application Frameworkiä. Käyttöliittymän moduulisuunnitteluvaiheessa tulee selvittää mahdollisuus käyttää gtkmm:ä käyttöliittymän toteuttamisvaihtoehtona. Tämä siksi, että oliopohjainen lähestyminen tapahtumaohjattuun käyttöliittymän tekoon on paljon helpompaa. On kuitenkin myös otettava huomioon, että kyseessä on rajoitettujen resurssien laite ja tästä johtuen kevyempi ratkaisu voisi tulla myös kysymykseen.

Ohjelman tulee olla sekä ikkuna että kokonäyttönäkymät.

4.1.1 Aloitusnäkyvä

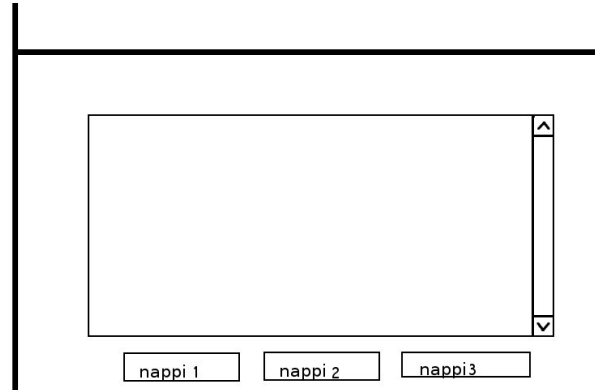
Aloitusnäkyvässä on kolme nappia. Yhdestä napista voi lukea yhden rfid-tagin. Mikäli tämä nappi valitaan, ohjelmisto siirtyy suoraan näkymään kolme. Toisesta napista ohjelmisto aloittaa lukea rfid-tageja ja lukee niitä niin kauan kunnes käyttäjä keskeyttää lukemisen (kts. näkymä2). Kolmas nappi avaa valikon, josta voi valita jonkun aikaisemman lukukerran



rfid-tagin, mikäli se on tallennettuna. Kun vanhan tagin valinta on tehty, siirtyy ohjelma suoraan näkymään kolme. Ohjelman ensimmäinen näkymä sisältää myös ohjelman tai sen tehneen yrityksen logon.

4.1.2 Listausnäkyvä

Tähän näkymään tullaan, kun on valittu ensimmäisestä näkymästä jatkuva RFID-tagien luku. Kuvassa näkyvään kontrolliin lisätään sitä mukaan RFID-tageja kun niitä on saatu luettua. Ensimmäisestä napista keskeytetään tagien lukeminen. Toinen nappi mahdollistaa

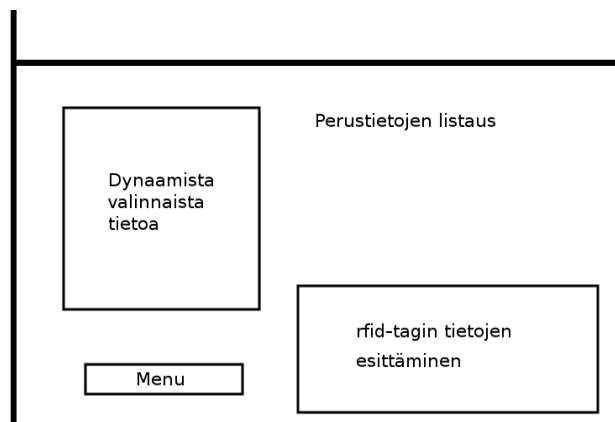


keskeytyksen jälkeen luettujen tagien listan tallentamisen. Kolmannella napilla siirrytään näkymään kolme ja siellä näytetään sen tagin tiedot, joka napin painamishetkellä on valittuna listauksessa. Sama toiminta saavutetaan myös tuplaklikkaamalla listauksen riviä.

4.1.3 Tietojen esitysnäkyvä

Tähän näkymään tullaan joko aloitusnäkymästä suoraan tai listausnäkyvän kautta. Tässä näkymässä on koottuna kaikki RFID-tagiin liittyvä informaatio. Näkyvä sisältää sekä tagiin tallennetut tiedot, että tagin tunnisteiden perusteella langattoman yhteyden kautta palvelimelta noudetut tiedot.

Näkymän vasemmassa alalaidassa on valikko. Tämä valikko sisältää tiedot potilaan laboratorioskokeista ja -testeistä. Valikosta aukeaa listaus kaikista tehdyistä kokeista aikaleimoilla eroteltuina. Valitsemalla listasta jonkin kokeen aikaleiman esitetään sen kokeen tulokset ponnahdusikkunassa.



Oikealla alhaalla on alue rfid-tagiin tallennettujen tietojen esittämiseksi. Tämän yläpuolelle tulostetaan potilaan perustiedot.

Vasemmalla ylhäällä oleva alue on valinnaista niin sanottua dynaamista dataa varten varattu tila. Tässä voidaan esittää esimerkiksi potilaan kuva tai muuta tarpeelliseksi katsottua tietoa.

4.2 Ohjelmisto rajapinta

Moduulisuunnittelussa ja -toteutuksessa tulee pyrkiä eriyttämään komponentteja ja luomaan niille selkeät rajapinnat, jotta ohjelmisto olisi mahdollisimman geneerinen.

4.2.1 RFID-lukija

Selkeästi oma ohjelmistorajapinta pitää luoda rfid-lukija kommunikointia varten. Librfid-kirjaston toiminnot kätketään tämän rajapinnan taakse

4.2.2 Palvelinkommunikaatio

Toinen eriytettävä komponentti on verkkoyhteydestä ja palvelinkommunikaatiosta huolehtiva ohjelman osa. Palvelinkyselyjen toteutus tulee kätkeä palvelun pyytäjältä rajapinnan taakse, jotta se tarvittaessa voidaan korvata jollakin muulla tekniikalla tai toteutustavalla.

4.3 Verkkoyhteys

Päätelaitteen tietoliikenne rajapinta on langaton verkkoyhteys palvelimeen. Tämän projektin palvelimen rajapinnan määrittelyn voi lukea palvelimen toiminnallisen määrittelyn dokumentaatiosta. Tämä rajapinta määrittyy pitkälti palvelimen tarjoamien palvelujen pohjalle.

Yhteystekniikkana käytetään SOAP:ia ja palvelimen rajapintakuvaus on toteuttu WSDL-määrittelynä. Mikäli tarvetta on monen muuttujan lähettämiseen tai lukemiseen yhdellä SOAP kutsulla, käytetään xml:ää muuttujien paketointiin.

Verkkoyhteys tulee olla salattu. Palvelin edellyttää kirjautumista.

LIITE 2, Testiskenaarion kuvan kaappaukset

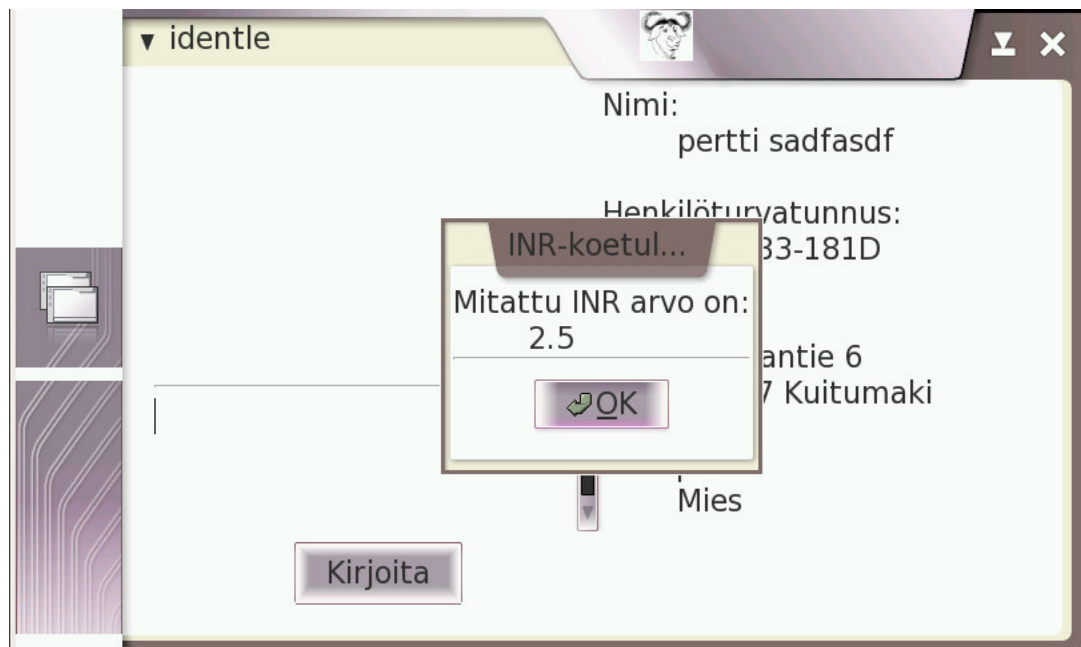
Tässä liitteessä esitetään IDENTLE:n testiskenaarion aikana otetut kuvankaappaukset Nokia 770 näytöltä.



Kuva 1. Aloitusnäky



Kuva 2. Potilaan testitulosten listaus



Kuva 3. Palautettu testiarvo serveriltä.