

Antti Hantula

Etähallittava ovikoodijärjestelmä

Opinnäytetyö

Syksy 2012

Tekniikan yksikkö

Tietotekniikan koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietotekniikan koulutusohjelma

Suuntautumisvaihtoehto: Tietoverkkotekniikka

Tekijä: Antti Hantula

Työn nimi: Etähallittava ovikoodijärjestelmä

Ohjaaja: Petteri Mäkelä

Vuosi: 2012

Sivumäärä: 32

Liitteiden lukumäärä: 0

Tämän opinnäytetyön tavoitteena oli kehittää yksinkertainen ja edullinen etähallittava ovikoodijärjestelmä. Laitteisto koostuu oven vieressä olevasta numeronäppäimistöstä sekä langattomasta lähettimestä ja vastaanottimesta, jotka on kytketty Raspberry Pi -nimiseen yhden piirinlevyn Linux-tietokoneeseen ja palveluntarjoajan tietokoneeseen. Hallintasovellusta käyttämällä palveluntarjoaja voi määrätä ovikoodin ja sen voimassaoloajan. Ovi avataan jos käyttäjä syöttää koodin oikein, ja suljetaan kun voimassaoloaika päättyy. Tiedonsiirto tapahtuu matkapuhelinverkon kautta tekstiviestillä. Ohjelmoinnissa käytettiin C#-kieltä hallintasovelluksen osalta, ja C++:aa vastaanotettujen viestien käsittelyyn.

Avainsanat: kulunvalvonta, ovikoodi, Raspberry Pi

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Telecommunication

Author: Antti Hantula

Title of thesis: Remote controlled door code system

Supervisor: Petteri Mäkelä

Year: 2012 Number of pages: 32 Number of appendices: 0

The aim of this thesis was to develop a cheap and simple remote controlled door code system. The hardware of the system consists of a numeric matrix keyboard next to the door and two GSM USB modems connected to the Raspberry Pi single board Linux computer and operator's PC. By using an administration program the operator can set the code and its validity period. The door will open if the user enters the correct code and close when the code has expired. Data transmission takes place via SMS. Programming languages were C# for administration software and C++ for handling received messages.

Keywords: access control, door code, Raspberry Pi

SISÄLTÖ

| | |
|---|----|
| Opinnäytetyön tiivistelmä..... | 2 |
| Thesis abstract..... | 3 |
| SISÄLTÖ | 4 |
| KUVIO- JA TAULUKKOLUETTELO | 5 |
| KÄYTETYT TERMIT JA LYHENTEET..... | 6 |
| 1 JOHDANTO | 8 |
| 1.1 Työn tausta | 8 |
| 1.2 Työn tavoitteet..... | 8 |
| 1.3 Työn rakenne | 8 |
| 2 TIETOLIIKENNEYHTEYDET | 9 |
| 2.1 Tekstiviestit | 9 |
| 2.2 GSM-modeemi..... | 10 |
| 2.3 GSM-modeemin ohjaaminen AT-komennoilla..... | 11 |
| 3 RASPBERRY PI | 13 |
| 4 OVIKOODIJÄRJESTELMÄ | 15 |
| 4.1 Järjestelmän osat..... | 15 |
| 4.2 Toimintaperiaate | 16 |
| 5. TYÖN TOTEUTUS | 17 |
| 5.1 Elektroniikka | 17 |
| 5.2. Ohjelmointi | 19 |
| 5.2.1 Kehitysympäristö..... | 19 |
| 5.2.2 Hallintasovellus | 23 |
| 5.2.3 Vastaanotto | 25 |
| 6 TYÖN TULOKSET | 28 |
| 6.1 Tulokset | 28 |
| 6.2. Yhteenveto..... | 28 |
| LÄHTEET | 29 |

KUVIO- JA TAULUKKOLUETTELO

| | |
|---|----|
| Taulukko 1. PDU-datan tulkinta..... | 10 |
| Kuvio 1. Raspberry Pi | 14 |
| Kuvio 2. Ovikoodijärjestelmän osat | 16 |
| Kuvio 3. GPIO-pinnit | 17 |
| Kuvio 4. Releen kytkentä | 19 |
| Kuvio 5. Win32 Disk Imager..... | 21 |
| Kuvio 6. dmesg-komennon näyttämää tietoa..... | 23 |
| Kuvio 7. Hallintasovelluksen pääikkuna | 24 |

KÄYTETYT TERMIT JA LYHENTEET

| | |
|--------------------|---|
| ASCII | American Standard Code for Information Interchange. Erilaisten merkkien esittämistä tietokoneella kuvaava standardi, jossa kutakin merkkiä vastaa yksi kymmenjärjestelmän luku (Jaakkola 5.8.2004). |
| ARM | Advanced RISC Machines. Tietokonevalmistaja Acornin 1980-luvulla kehittämä suoritinarkkitehtuuri (ARM Company Milestones 2012). |
| GPIO | General Purpose Input/Output. Mikrokontrollerin tai -prosessorin yleiskäyttöinen nasta. Voidaan ohjelmoida joko tuloksi tai lähdöksi. (RPi Low-level peripherals 21.5.2012). |
| GSM | Global System for Mobile Communications. Maailmanlaajuinen digitaalinen matkaviestinverkko (Teknologiasanasto 2012). |
| HDMI | High-Definition Multimedia Interface. Liitäntästandardi digitaalisen kuvan ja äänen siirtoon (Teknologiasanasto 2012). |
| LED | Light-emitting Diode. Puolijohdekomponentti, joka säteilee valoa kun sähkövirta kulkee sen läpi (Huhtama [viitattu 21.9.2012]). |
| USB | Universal Serial Bus. Sarjamuotoinen väylä, jota pitkin laitteet voivat vaihtaa tietoja keskenään ja saada itselleen virtaa. (Teknologiasanasto 2012). |
| Sarjaportti | Sarjamuotoinen tiedonsiirtotapa. Huomattavasti hitaampi ja vanhempi tekniikka kuin USB (McNamara [viitattu 23.9.2012]). |

- SIM** Subscriber Identity Module. Matkapuhelinverkossa liittymän tunnistamiseen käytetty siru (Learn about Subscriber Identity Module 28.7.2010).
- SSH** Tietoliikenneprotokolla muun muassa salatun etäkäyttöyhdyden muodostamiseksi toiseen tietokoneeseen (Pitkänen 16.8.2012).
- PIN** Personal Identification Number. Numerokoodi, joka suojaa matkapuhelinta tai muuta laitetta tai järjestelmää väärinkäytöltä (PIN code [viitattu 24.9.2012].)
- x86** Intelin kehittämä suoritinarkkitehtuuri eli eräs tapa, miten tietokoneiden prosessoreja on rakennettu (Kurri 9.6.2008).

1 JOHDANTO

1.1 Työn tausta

Hotelleissa ja vuokramökeissä on säännöllisesti ongelmia kadonneiden avaimien tai kulkukorttien kanssa. Yksi ratkaisu on ovikoodijärjestelmä. Aiemmin sellaisen toteuttaminen ei välttämättä ollut kovin halpaa, vaan saattoi vaatia kalliin tietokantapalvelimen hankkimista. Viime vuosina on kuitenkin julkaistu muutamia erittäin edullisia ja pieniä tietokoneita, esimerkiksi Raspberry Pi. Ne alentavat kustannuksia merkittävästi. Tästä syntyi idea tuotekehitysprojektiin ja opinnäytetyöhön.

1.2 Työn tavoitteet

Tavoitteena oli valmistaa yksinkertainen ja edullinen etähallittava ovikoodijärjestelmä, jonka tiedonsiirrossa hyödynnetään GSM-modeemia. Käyttäjä syöttää oven vieressä olevaa näppäimistöä käyttäen koodin. Jos koodi on syötetty oikein, ovi aukeaa. Koodi ja sen voimassaoloaika voidaan määrätä palveluntarjoajan tietokoneelta hallintasovelluksen avulla. Huoneistoon sijoitettu Raspberry Pi tai vastaava laite saa koodin tiedot sovellukselta tekstiviestinä ja ohjaa oven lukkoa.

1.3 Työn rakenne

Luku 2 käsittelee työhön liittyvää teoriaa. Se sisältää perustietoa ovikoodijärjestelmän käyttämistä tietoliikenneyhteyksistä.

Luvussa 3 kerrotaan Raspberry Pi -tietokoneesta.

Luvussa 4 esitellään järjestelmän osat ja toimintaperiaate

Luku 5 kertoo, miten työ käytännössä eteni.

Luvussa 6 käsitellään työn tuloksia.

2 TIETOLIIKENNEYHTEYDET

2.1 Tekstiviestit

Matkapuhelinten kehityksen myötä tekstiviesteistä on 20 viime vuoden aikana tullut kiinteä osa arkielämää. Henkilökohtaisen viestinnän lisäksi niitä käytetään yhä enemmän muuan muassa automaatiolaitteiden ohjauksessa ja erilaisten sisältö- sekä maksupalveluiden toteutuksessa.

Tekstiviesti välitetetään vastaanottajalle operaattorin viestikeskukseen kautta. Lähettämiseen on kaksi eri tapaa: ASCII-muotoinen teksti (text mode) ja PDU-koodaus (PDU mode). Yhden tekstiviestin maksimipituus on 160 merkkiä, mukaan lukien välilyönnit ja rivinvaihdot. On silti mahdollista lähettää pidempiäkin viestejä, jotka jaetaan 160 merkin osiin. Text mode ei kuitenkaan tue viestin jakamista, mikä tarkoittaa, ettei ylipitkää viestiä paloitella automaattisesti. (Janssen [viitattu 24.9.2012].) Rajoitus voidaan kiertää ohjelmiston suunnittelulla. Lisäksi PDU:n hyötynä on se, että voidaan käyttää mitä tahansa merkistöä. Toisaalta aivan kaikki puhelimet ja muut päätelaitteet eivät tue text modea. (Pettersson [viitattu 23.7.2012].)

PDU-koodattu data sisältää varsinaisen viestin lisäksi paljon muutakin tietoa. Voidaan esimerkiksi määritellä aika, jonka jälkeen viesti poistetaan keskuksesta, ellei sitä ole saatu toimitettua eteenpäin. Data koostuu pääosin 8-bittisistä heksadesimaaliarvoista eli okteteista, mutta mukana on myös puolioktetteja, joita voi tulkita helposti kääntämällä ne toisin päin (esim. 72 tarkoittaa kymmenjärjestelmän lukua 27). Viestin koodaaminen lähetystä varten tapahtuu niin, että kunkin merkin ASCII-arvo muutetaan binääriluvuksi. Ensimmäisen merkin binääriarvoon lisätään toisesta merkistä viimeinen bitti, kolmanteen neljännestä 2 viimeistä bittiä jne. Näin saadaan muodostettua oktetit. Kun ne vielä muunnetaan heksadesimaaliluvuiksi, koodaus on valmis. Vastaanotettaessa oktetit muunnetaan ASCII-arvoiksi, jolloin taulukosta nähdään arvoja vastaavat merkit. (Pettersson [viitattu 23.7.2012].)

Esimerkiksi lähetetty data:

0011000B916407281553F80000AA0AE8329BFD4697D9EC37

Taulukko 1. PDU-datan tulkinta (Pettersson [viitattu 23.7.2012])

| Oktetti | Kuvaus |
|--------------------|--|
| 00 | Viestikeskuksen numeron pituus |
| 11 | Määritetään mm. viestin voimassaoloajan muoto ja se, halutaanko saada toimitusraportteja |
| 00 | Viestin tunnistenumero |
| 0B | Puhelinnumeron pituus (0B = 11) |
| 91 | Puhelinnumeron muoto (91 = kansainv.) |
| 6407281553F8 | Puhelinnumero (puolioktetteina) |
| 00 | Käytetty protokolla |
| 00 | Käytetty merkistö (00 = oletusmerkistö 7 bit, standardi GSM 03.38) |
| AA | Viestin voimassaoloaika (AA = 4 pv.) |
| 0A | Viestin pituus merkkeinä (0A = 10) |
| E8329BFD4697D9EC37 | Viesti |

Vastaanotetussa datassa toinen oktetti on viestikeskuksen numeron muoto, ja kolmas itse numero. Sen jälkeen tulee uusi oktetti, joka muun muassa kertoo, onko muita viestin osia tai toimitusraportti vielä tulossa. Voimassaoloaika korvautuu aikaleimalla. Muutoin oktettien järjestys ja merkitys pysyy samana. Viestikeskuksen numeron pituus 0 tarkoittaa, että käytetään päätelaitteeseen tallennettua numeroa. (Pettersson [viitattu 23.7.2012].)

2.2 GSM-modeemi

GSM-modeemi on tietokoneen USB-porttiin liitettävä laite, jonka pääasiallinen tehtävä on yhdistää tietokone matkaviestinverkkoon langatonta Internetin käyttöä varten. Tämän lisäksi se voi myös lähettää ja vastaanottaa tekstiviestejä samoin kuin matkapuhelin. Sitä ennen modeemissa pitää tietysti olla SIM-kortti, jonka PIN-koodi tiedetään, jos sen kysely on käytössä. (SMS Tutorial: Introduction [viitattu 21.9.2012].)

GSM-modeemin mukana toimitetaan usein operaattorikohtainen yhteysohjelmisto, jolla ei sinänsä ole tässä työssä merkitystä, mutta sen mukana asentuu modeemin toiminnan kannalta välttämätön laiteajuri. Asennus käynnistyy kun modeemi liitetään tietokoneeseen. Modeemin sisäinen muisti, jossa asennustiedostot ovat, näkyy tietokoneelle CD-asemana tai muuna massamuistina. (Vinet 9.6.2009.) Tällöin tekstiviestien vastaanotto ja lähettämiseen käytettävät AT-komennot eivät toimi aina heti. Muilla komennoilla (ks. kohta 2.3) saadaan modeemi tarvittaessa vaihtamaan tilaansa niin, että viestit kulkevat. Linuxissa vaihdon hoitaa ohjelma nimeltä `usb-modeswitch` (Harvey 28.4.2010).

Jos laitteelle on ajuri suoraan käyttöjärjestelmässä, mitään tällaisia erityistoimenpiteitä ei tarvita. Normaalisti uusimmat järjestelmät sisältävät enemmän laiteajureita valmiina, jolloin operaattorin tarjoamaa ohjelmistoa ei välttämättä ole pakko asentaa, eikä se käynnistyessään pääse varaamaan modeemia itselleen. Usein yhteysohjelma avautuu kun käyttöjärjestelmä on latautunut tai modeemi kytketään tietokoneeseen. Käyttäytymistä voi yleensä kuitenkin säädellä ohjelman asetuksista. (Harvey 28.4.2010.)

2.3 GSM-modeemin ohjaaminen AT-komennoilla

Kun laiteajuri on asennettu, tietokone tunnistaa modeemin sarjaporttina, jonka välityksellä modeemille voidaan antaa komentoja. Komentojen lähettämiseen käsin tarvitaan pääteohjelma (esim. PuTTY tai Microsoft HyperTerminal). Jokainen komento alkaa kirjaimilla "AT". Jos komento menee perille ja suoritetaan onnistuneesti, modeemi vastaa "OK" ja usein lähettää myös muuta tietoa. Virhetilanteessa vastaus on "ERROR" tai muu tarkempi virheilmoitus tai vastausta ei tule ollenkaan. Käytettävissä olevat komennot, parametrit ja niiden arvot voivat vaihdella modeemista riippuen. (Send and receiving SMS [viitattu 22.5.2012].)

Tämän työn kannalta tärkeimmät komennot ovat:

- AT+ZOPRT=5 ja AT+ZCDRUN=8 (asettavat modeemin tilan viestien lähetystä ja vastaanottoa varten)
- AT+CMGF=1 (valitsee viestin muodoksi tekstin)
- AT+CMGS="numero" (lähettää viestin annettuun puhelinnumeroon)
- AT+CMGR=x (lukee viestin annetusta indeksistä x)
- AT+CMGD=x (poistaa viestin annetusta indeksistä x)
- AT+CPMS="ME" (määrittää saapuvien viestien tallennuspaikaksi modeemin sisäisen muistin). (Send and receiving SMS; Vinet 9.6.2009.)

Myös matkapuhelimet käyttävät näitä komentoja toimintansa ohjaamiseen. Jokainen komento, paitsi tekstiviestin lähetys, päättyy rivinvaihtoon. Kun annetaan komento AT+CMGS="numero", modeemi lähettää merkin ">" ja alkaa odottamaan, että viesti on kirjoitettu. Viimeiseksi tarvitaan loppumerkki, jonka saa muodostettua pääteohjelmassa näppäinyhdistelmällä CTRL+Z. Viestiin voi tehdä uuden rivin normaalisti Enter-näppäimellä, koska modeemi lähettää viestin vasta saatuaan loppumerkin. (SMS Tutorial: Example [viitattu 24.9.2012].)

3 RASPBERRY PI

Raspberry Pi on vuonna 2012 julkaistu pieni ja halpa yhden piirilevyn tietokone. Se perustuu Broadcom BCM2835 -järjestelmäpiiriin ja sen käyttöjärjestelmäksi voidaan valita jokin Linuxin jakeluversio. Massamuistina toimii Flash-tekniikkaan perustuva SD-muistikortti. (FAQs [viitattu 23.9.2012].)

Raspberry Pi ei ole täydellinen pöytäkoneen korvaaja, mutta sitä voidaan käyttää moneen eri tarkoitukseen, kuten Internetin selailu, tekstinkäsittely, taulukkolaskenta, viihdekäyttö ja pienimuotoiset palvelimet (FAQs [viitattu 23.9.2012]).

Koska sulautetut järjestelmät on usein suunniteltu olemaan jatkuvasti päällä, ei Raspberry:ssakaan ole erillistä virtapainiketta. Käyttöjännite on 5 V ja sähkönkulutus toiminnassa korkeintaan 3,5 W.

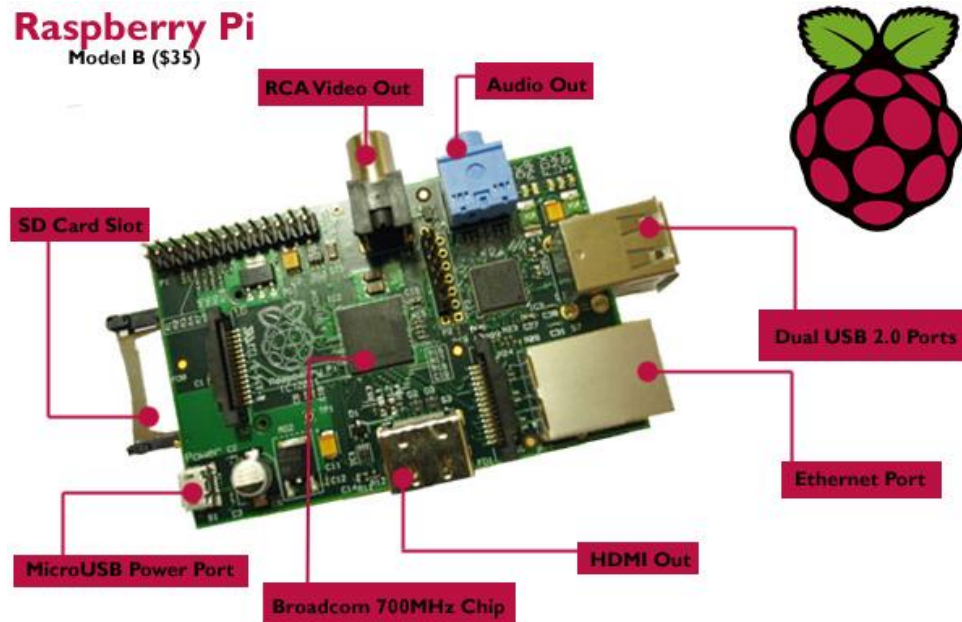
Raspberry Pin tärkeimmät tekniset tiedot:

Broadcom BCM2835 -järjestelmäpiiri, johon kuuluu:

- Prosessori: yksiytiminen ARM 1176JZ-F, kellotaajuus 700 MHz
- Näytönohjain: Broadcom VideoCore IV (muun muassa täysteräväpiirtoviideon toisto)
- Keskusmuisti: 256 Mt

Lisäksi:

- SD-muistikortti
- Liitännät: muun muassa USB (malli A 1 kpl, B 2 kpl), HDMI, Ethernet (vain malli B). (Halfacree 29.2.2012).



Kuvio 1. Raspberry Pi (Seghers 10.4.2012)

Mallin A hinta on 25 \$ ja mallin B 35 \$ (Seghers 10.4.2012).

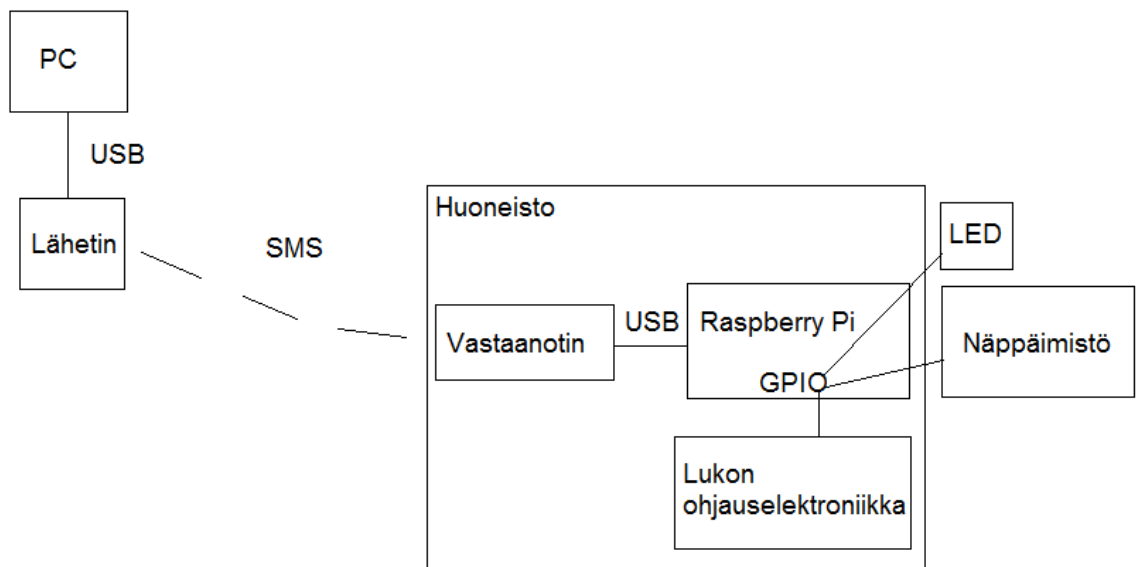
Muita vastaavia tuotteita ovat muun muassa Beagleboard xM ja Pandaboard. Raspberryyntä verrattuna niissä on vähän tehokkaampi prosessori ja enemmän keskusmuistia. Työn tavoitteena oli kuitenkin edullisuus, eivätkä erot ole niin merkittäviä, että olisi kannattanut maksaa moninkertainen hinta. Raspberry valittiin myös siksi, että siinä on hyvät liitännät ja se ehkä soveltuu tulevaisuudessa muihinkin projekteihin.

4 OVIKODIJÄRJESTELMÄ

4.1 Järjestelmän osat

Ovikoodijärjestelmä koostuu langattomasta lähettimestä ja vastaanottimesta sekä oven vieressä sijaitsevasta numeronäppäimistöstä. Lähetin, kuten vastaanotinkin, on USB-liitännäinen GSM-modeemi. Aluksi työ oli tarkoitus tehdä Siemensin MC35- ja TC45-GSM/GPRS-moduuleilla. Testikäyttöön ne olisivatkin varmasti olleet hyviä, mutta GSM-modeemin valintaa tuki edullinen hinta ja uudempi teknologia, sekä se, että monella mahdollisista asiakkaista todennäköisesti on jo sopiva laite valmiiksi hankittuna. Periaatteessa mikä tahansa matkapuhelinkin voi toimia lähettimenä tai vastaanottimena.

Lähetin on kytketty palveluntarjoajan tietokoneeseen, ja vastaanotin sekä näppäimistö Raspberryn. Tiedonsiirto laitteiden välillä toteutettiin matkapuhelinverkon kautta tekstiviestinä. Tämä on varmin keino, koska internetyhteys ei ole käytettävissä kaikkialla. Näppäimistö on 3x4 matriisinäppäimistö, jossa on numerot 0 - 9 (aluksi perusmalli, tuotantovaiheessa täytyy olla ulkokäyttöön tarkoitettu ja säänkestävä). Lisäksi järjestelmään kuuluu LED-valo ja lukon ohjauselektronikka, joita käsitellään tarkemmin luvussa 5.



Kuvio 2. Ovikoodijärjestelmän osat

4.2 Toimintaperiaate

Hallintasovellus pitää kirjaa tiloista, joihin järjestelmä on asennettu, ja tilojen ovi-koodeista. Kun lisätään uusi koodi, muokataan koodia tai poistetaan koodi, ohjelma lähettää tekstiviestin valitulle tilalle määriteltyyn puhelinnumeroon.

Vastaanottopään ohjelmisto odottaa koko ajan viestiä lähettimeltä ja käyttäjän syötettä näppäimistöltä sekä tarkkailee koodien voimassaoloaikoja. Syötettyä numerosarjaa verrataan kulloinkin käytössä olevaan koodiin, ja jos ne täsmäävät, lukko avataan ja vihreä LED-valo syttyy. Voimassaoloajan umpeuduttua ovi lukittuu ja valo sammuu.

5. TYÖN TOTEUTUS

5.1 Elektroniikka

Näppäimistö, LED-valo ja lukon ohjauselektroniikka liitettiin Rapberry Pi:n GPIO-pinneihin. Kytkentäkaaviosta (Raspberry Pi Schematics, 17.4.2012, 2) täytyi selvittää, mitä Broadcom BCM2835 –järjestelmäpiirin nastoja voi käyttää siihen tarkoitukseen. Jos piirilevyn asento on kuvion 1 mukainen, GPIO-pinnit sijaitsevat vasemmassa yläkulmassa. Niillä voidaan toteuttaa myös pulssinleveysmodulaatiota (PWM) ja erilaisia väyliä, kuten SPI ja I²C. Näitä ei työssä kuitenkaan tarvittu, eikä käsitellä tässä raportissa. Tarkempia ohjeita on järjestelmäpiirin datalehdessä (Broadcom Europe Ltd. 6.2.2012, 138 – 171). Kuviossa 3 esitetään pinnien ryhmittely.



Kuvio 3. GPIO-pinnit (RPi Low-level peripherals 21.5.2012)

Pinneihin voi viitata joko niiden fyysisen sijainnin mukaan numeroilla 1 – 26, sen mukaan mihin järjestelmäpiirin nastoihin ne on kytketty tai dokumentaatioissa annetuilla nimillä. Pinnit on yhdistetty suoraan järjestelmäpiiriin, joten vaurioiden välttämiseksi kytkentöjen suunnitteluun tulee kiinnittää erityistä huomiota. Looginen jännitetaso on 0 – 3,3 V. (RPi Low-level peripherals 21.5.2012.)

Yleismittarilla löydettiin kuoritusta näppäimistön liitántäkaapelista rivejä ja sarakkeita vastaavat ulostulojohtimet. Myöhemmin tämän tiedon voi katsoa valmistajan datalehdessä, jota ei kehityksessä käytetylle näppäimistölle ollut saatavilla. Näppäimistö liitettiin Raspberryyyn vanhan tietokoneen 34-pinnisellä levykeasemakaapelilla. Kaapelin väritetty reuna kertoo kummalla puolella on nasta 1. Kaikkia pinnnejä ei ole kytketty mihinkään. Ylimääräiset pinnit on varattu tulevaisuuteen Raspberryn uusia versiota varten.

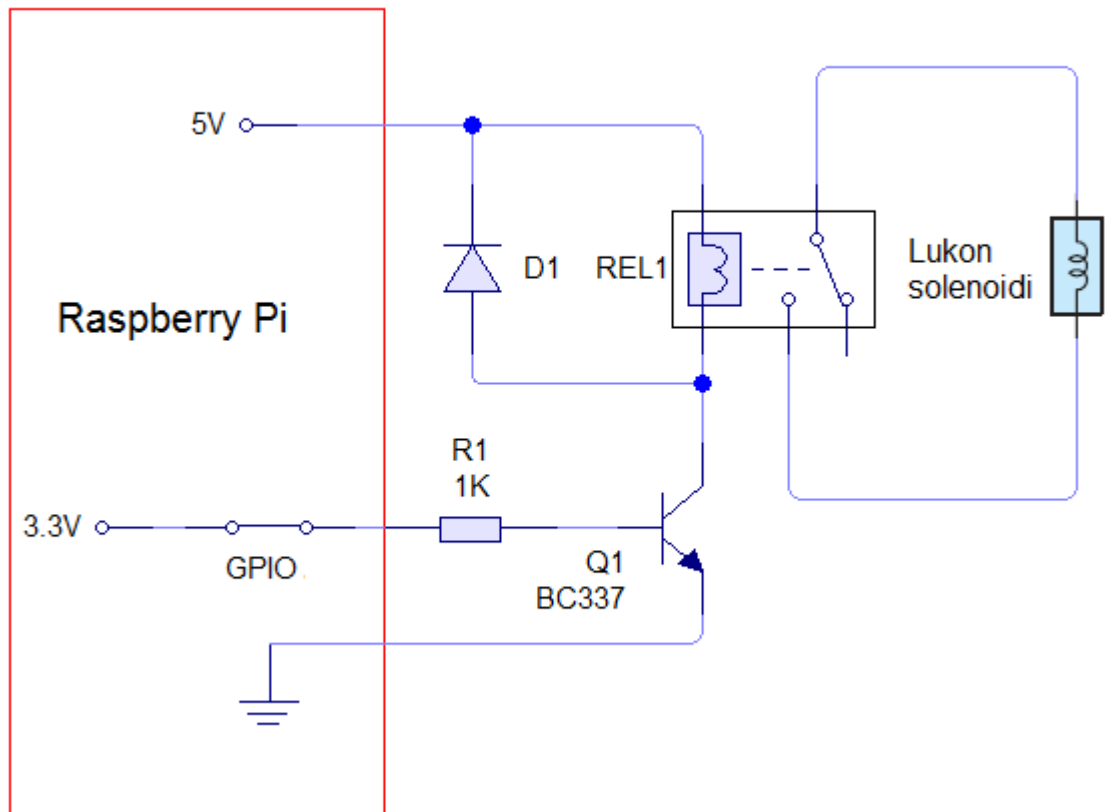
Matriisinäppäimistö toimii siten, että kullekin riville kirjoitetaan yksitellen looginen 1, muut rivit nollataan ja sitten luetaan sarakkeista arvot. Jokaisen lukukerran jälkeen pidetään 0,5 sekunnin tauko, ettei yhtä painallusta rekisteröidä moneen kertaan. Jotta arvo olisi vakio myös lepotilassa, eli silloin kun mitään näppäintä ei paineta, tarvitaan sarakkeisiin vastukset kytkettynä maihin ("alasetovastus") tai 3,3 V:n jännitteeseen ("ylösvetovastus"). Muuten se jää "kellumaan" ja vaihtelee satunnaisesti johtuen sähkömagneettisista häiriöistä. Sen sarakkeen arvo, jolla painettu näppäin sijaitsee, on 1. Muut arvot ovat 0, ja kun näppäinkartta on tiedossa, saadaan nappi selville päättelemällä. Menetelmää kutsutaan näppäimistön skannaamiseksi. (Rantala 2010, 3-7.)

LED-valolle tarvitaan etuvastus R1, ettei virta kasva liian suureksi ja hajota LED:iä. Jännite, joka tarvitaan LED:n sytyttämiseen, ns. kynnysjännite, vaihtelee värin mukaan, ja on vihreälle LED:lle n. 2 V. Vastusarvon laskemiseksi se täytyy vähentää GPIO-pinnistä saatavasta 3,3 V jännitteestä. Tämän jälkeen etuvastuksen arvo saadaan Ohmin laista:

$$U = R1 * I \Rightarrow R1 = \frac{U}{I} \Rightarrow R1 = \frac{1,3V}{I} \quad (1)$$

missä U on jännite ja I virta, jolla LED:iä halutaan käyttää (esim. 0,01 A = 10 mA). Mitä suurempi virta, sitä kirkkaammin valo loistaa. LED:n katodi kytketään maihin ja anodi GPIO-pinniin. Katodin tunnistaa siitä, että sen puolella LED:ssä on viiste. Ei ole väliä kummalle puolelle LED:iä etuvastus laitetaan.

Lukon solenoidia ohjataan releellä. Lukko avautuu kun releen käämiin johdetun sähkövirran synnyttämä magneettikenttä vetää koskettimet yhteen, ja sulkeutuu kun virta katkaistaan, jolloin kenttä häviää. Releen kytkentä esitetään kuviossa 4.



Kuvio 4. Releen kytkentä (Sangeelee 26.6.2012)

5.2. Ohjelmointi

Tässä luvussa kerrotaan käytetystä kehitysympäristöstä sekä hallinta- ja vastaanotto-sovelluksien toteutuksesta.

5.2.1 Kehitysympäristö

Ohjelmointikielinä päätettiin käyttää hallintasovelluksen osalta C#-kieltä ja vastaanotossa C++ :aa. C# soveltuu hyvin graafisen käyttöliittymän ohjelmointiin, kun

taas C++:n vahvuudet ovat enemmän sulautetuissa järjestelmissä. Perinteisesti mikrokontrollereita ja -prosessoreja on ohjelmoitu C-kielellä, mutta kummassakin valitussa kielessä on joitakin C-kieleen verrattuna parempia merkkijonon käsittelyyn ja olio-ohjelmointiin liittyviä ominaisuuksia.

Ohjelmoinnissa käytettiin Microsoft Visual Studio- ja Eclipse-kehitysympäristöjä sekä Raspberrylle tehtyä virallista C++-kääntäjää **arm-bcm2708-linux-gnueabi**. Ympäristöistä erityisesti ensin mainittu on tullut opiskelun aikana tutuksi.

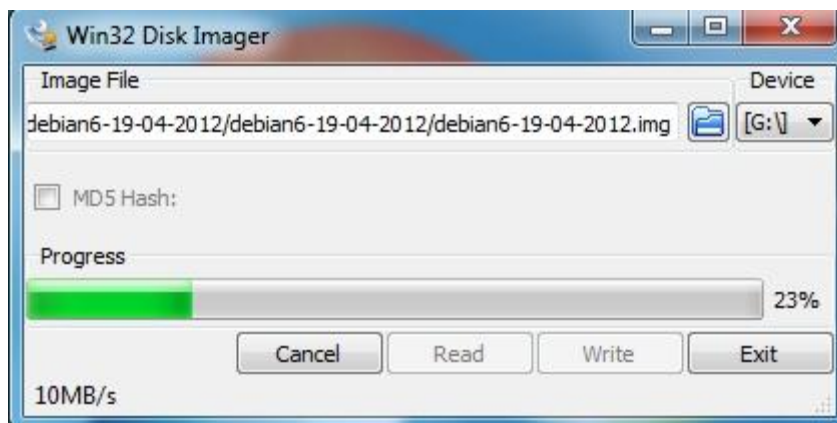
Debian Linux -käyttöjärjestelmän ja Eclipsen asentaminen pöytäkoneelle oli hyvin suoraviivaista: haettiin asennuslevykuva Debianin kotisivuilta ja kirjoitettiin se CD-levylle, jolta tietokone käynnistettiin. Tämän jälkeen täytyi vain noudattaa ruudulla näkyneitä ohjeita. Eclipse löytyi Debianin pakettienhallinnasta asentuen muutamalla hiiren klikkauksella. Kääntäjän asentamiseen perehtyminen oli työläämpää, mutta lopulta asennus eteni kivuttomasti Lakeview Researchin ohjetta (Using Eclipse to Cross-compile [viitattu 15.5.2012]) seuraten ja soveltaen.

Vaikka kääntäjää ajetaan x86-arkkitehtuurin tietokoneella, se tuottaa Raspberryn ARM-prosessorille soveltuvaa konekoodia, joka vain siirretään Raspberryn kun ohjelma on valmis. Näin kehitystyötä pystyttiin tekemään jo ennen kuin laitteeseen pääsi fyysisesti käsiksi, eikä ilmenneiden toimitusvaikeuksien takia mennyt niin paljon aikaa hukkaan.

Debian on vakaa, Raspberry Pi -säätöön suosittelema käyttöjärjestelmäversio. Sen asentamiseksi piti ladata levykuvatiedosto Raspberryn nettisivuilta ja kirjoittaa SD-muistikortille Win32 Disk Imagerilla. Tässä vaiheessa oli syytä olla tarkkana, ettei valitse väärää asemakirjainta ja tuhoa koneen kovalevyä. Saatavilla on myös valmiiksi asennetun käyttöjärjestelmän sisältäviä kortteja. Muistikortti voidaan kirjoittaa kokonaan uudelleen n. 100 000 kertaa. Lukea voi rajattomasti, Flash-muisti ei siitä kulu. Kortin kirjoitusnopeudeksi mitattiin n. 10 Mt/s (ks. Kuvio 5) ja lukunopeudeksi n. 19 Mt/s. Alkuperäisen SD-standardin mukaan muistikortin suurin mahdollinen tallennuskapasiteetti on 2 Gt. Rasperry tukee kuitenkin myös uudempia SDHC-kortteja, jotka tarjoavat enintään 32 Gt kapasiteetin. Debian Linux tarvitsee 1,6 Gt vapaata tilaa. Suuret kortit pitää asennuksen jälkeen osioida uudestaan

esim. Gparted-ohjelmalla, jotta täysi kapasiteetti saadaan käyttöön, sillä levykuva on vain 2 Gt:n kokoinen (Resize SD Card 1.6.2012; Capacity 2012.)

Raspberry käynnistyy alle 30 sekunnissa. Oletuksena on käytössä brittiläinen näppäimistöasettelu ja GMT-aika 12 h:n kellolla. Jostain syystä käynnistysaika pitenee lähes 2 minuuttiin, jos asettelu vaihdetaan suomalaiseksi, mutta siihen toki vaikuttaa myös muistikortin nopeus. Tämän työn kannalta tärkeämpää on oikea kellonaika, joka tulee internetin aikapalvelimelta, koska Raspberrysssä ei ole reaaliaikakelloa. Aika haetaan käynnistyksen yhteydessä GSM-modeemilla. Dimopoulos (2010) on kehittänyt Sakis3G-komentojonoskriptin, jolla yhteys muodostuu vaivatta, kunhan modeemi on ensin saatu toimimaan oikein. Skripti sisältää graafisen käyttöliittymän, mutta on silti kevyt, eikä vaadi juuri mitään konfigurointia, vain tukiaseman nimi täytyy valita. Muut asetukset skripti tunnistaa automaattisesti.



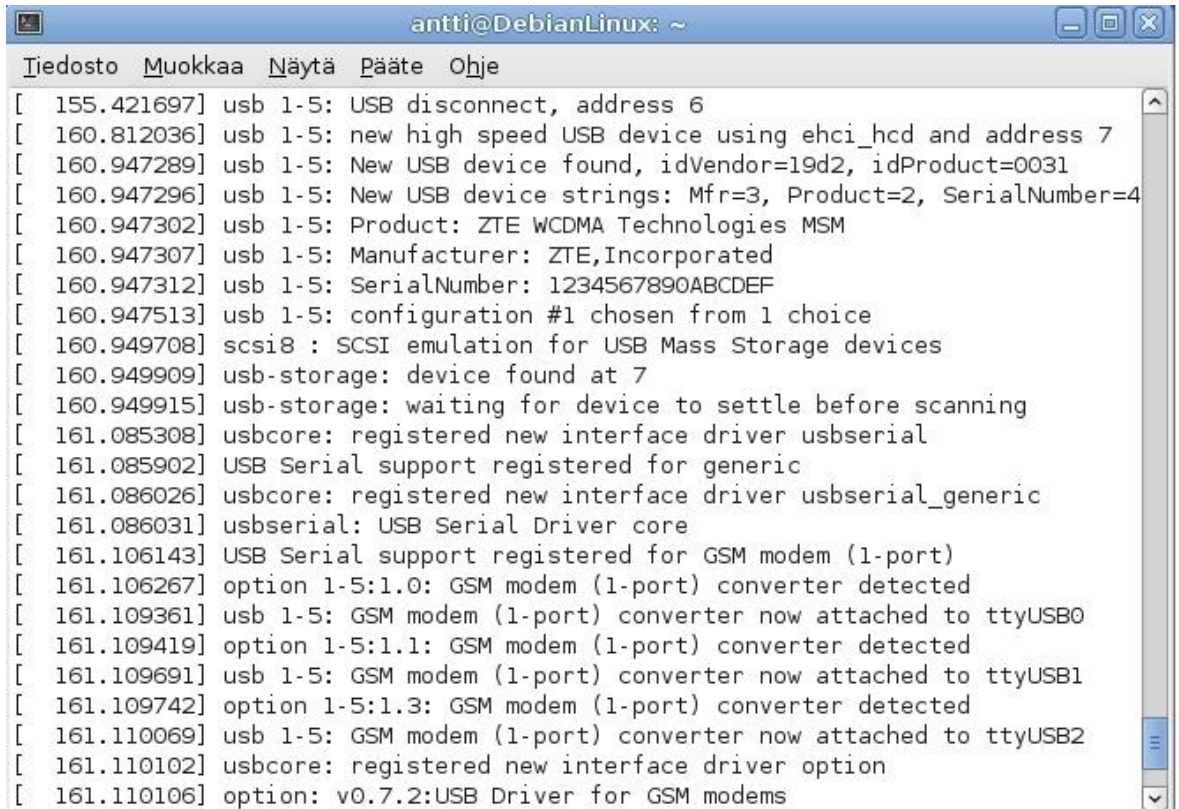
Kuvio 5. Win32 Disk Imager

Ovikoodijärjestelmää kehitettiin ja testattiin ensin ZTE MF636+ GSM-modeemilla, joka toimi hyvin pöytäkoneessa sekä Windows- että Linux-käyttöjärjestelmällä, mutta huonosti Raspberryn kanssa. Vastauksen saaminen AT-komentoihin oli todella hidasta ja epävarmaa vaikka graafinen työpöytäympäristö ei ollut käynnissä. Myös lähiverkkoyhteys katkeili kun modeemi oli liitettynä. Seuraavaksi yritettiin Huawei E367:lla, joka ei toiminut, vaan käyttäytyi kuten MF636+. Ongelmien syyinä oli liian suuri virrankulutus. Raspberryn USB-portti pystyy antamaan vain 100 mA, koska sen kaikkia pinnejä ei ole kytketty. Muuten maksimivirta voisi USB 2.0 -standardin mukaan olla jopa 500 mA eli aivan liikaa, jos virtalähteestä otetaan 700 mA koko laitteistolle. Toimimaan saatiin vain Huawei E1552.

Kehittäjien sivustolla Huaweiin mallit E220 ja E160 (AT-komentojen osalta) on vahvistettu toimiviksi. Erittäin todennäköisesti myös mallit E169, E1550, E620, E270 ja E230 toimivat, mutta laitteiden saatavuudesta johtuen niitä ei ehditty kokeilla. Kuitenkin modeemin sisäiset komponentit pysyvät usein samoina, ja ero saattaa olla vain siinä, voiko malliin liittää ulkoisen antennin tai muistikortin, tai mitä tiedonsiirtonopeutta ja verkon taajuuksia malli tukee.

Linuxissa modeemin käyttämän portin näkee kirjoittamalla komentotulkkiin komennon **dmesg**, joka näyttää kaikki tietokoneen osat ja koneeseen liitetyt laitteet (Kuvio 6). Itse asiassa MF636+ otti komentoja vastaan sekä portissa ttyUSB0 että ttyUSB1. Portti saataa muuttua, jos modeemi irrotetaan ja liitetään sitten takaisin ilman, että konetta käynnistetään välillä uudelleen.

Eclipsellä kehitetyn vastaanottosovelluksen siirtämiseksi Rasperryyn täytyi laittaa päälle SSH-palvelu, joka pöytäkoneella oli jo valmiina. Rasperryn kansiossa /boot oleva skripti **boot_enable_ssh.rc** vain nimetiin uudelleen muotoon **boot.rc**. Sen jälkeen siirto kävi helposti komennolla **scp tiedosto käyttäjänimi@palvelin:kansio**, missä palvelin on Rasperryn IP-osoite, jonka näki käynnistyksen yhteydessä.



```

antti@DebianLinux: ~
Tiedosto Muokkaa Näytä Pääte Ohje
[ 155.421697] usb 1-5: USB disconnect, address 6
[ 160.812036] usb 1-5: new high speed USB device using ehci_hcd and address 7
[ 160.947289] usb 1-5: New USB device found, idVendor=19d2, idProduct=0031
[ 160.947296] usb 1-5: New USB device strings: Mfr=3, Product=2, SerialNumber=4
[ 160.947302] usb 1-5: Product: ZTE WCDMA Technologies MSM
[ 160.947307] usb 1-5: Manufacturer: ZTE, Incorporated
[ 160.947312] usb 1-5: SerialNumber: 1234567890ABCDEF
[ 160.947513] usb 1-5: configuration #1 chosen from 1 choice
[ 160.949708] scsi8 : SCSI emulation for USB Mass Storage devices
[ 160.949909] usb-storage: device found at 7
[ 160.949915] usb-storage: waiting for device to settle before scanning
[ 161.085308] usbcore: registered new interface driver usbserial
[ 161.085902] USB Serial support registered for generic
[ 161.086026] usbcore: registered new interface driver usbserial_generic
[ 161.086031] usbserial: USB Serial Driver core
[ 161.106143] USB Serial support registered for GSM modem (1-port)
[ 161.106267] option 1-5:1.0: GSM modem (1-port) converter detected
[ 161.109361] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB0
[ 161.109419] option 1-5:1.1: GSM modem (1-port) converter detected
[ 161.109691] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB1
[ 161.109742] option 1-5:1.3: GSM modem (1-port) converter detected
[ 161.110069] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB2
[ 161.110102] usbcore: registered new interface driver option
[ 161.110106] option: v0.7.2:USB Driver for GSM modems

```

Kuvio 6. dmesg-komennon näyttämää tietoa

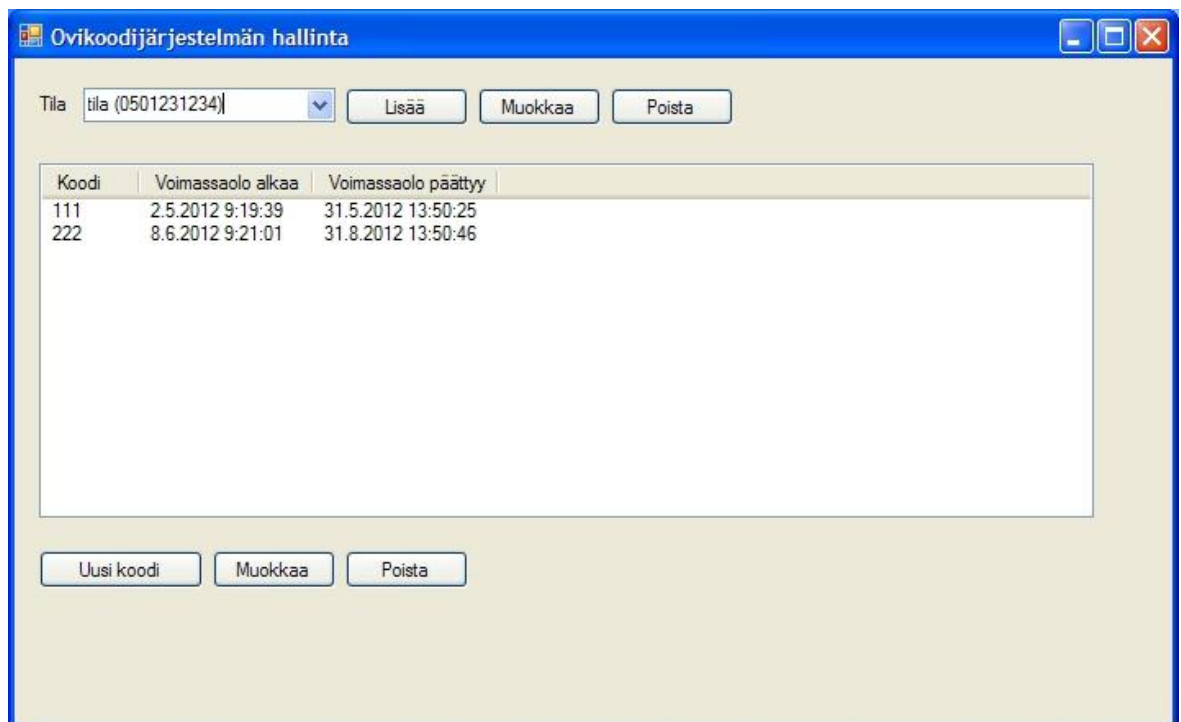
5.2.2 Hallintasovellus

Hallintasovellus, kuten kaikki muutkin Visual Studiolla kehitetyt sovellukset, perustuu Microsoftin .NET-arkkitehtuuriin, ja toimii näin ollen vain Windowsissa. Arkkitehtuuri asettaa tietokoneelle vähimmäisvaatimuksia. Näitä ovat:

- Prosessorin kellotaajuus vähintään 400 MHz
- Keskusmuistia vähintään 256 Mt
- Levytilaa muutama Mt itse ohjelmalle, .NET Framework –ohjelmiston asennus kuluttaa vähintään 280 Mt
- Windowsin versio XP tai uudempi, kuluttaa väh. 1,5 Gt levytilaa
- .NET Framework versio 3.5 tai uudempi asennettuna (.NET framework 2011; Windows XP 2010.)

Lähes kaikki nykyisin käytössä olevat tietokoneet ylittävät nämä vaatimukset reilusti.

Sovelluksen pääikkuna esitetään kuviossa 7. Sovellus kysyy käyttäjältä vastaanottimen puhelinnumeron ja tallentaa sen tiedostoon annetulla nimellä. Nämä tiedot yhdessä muodostavat ”tilan”. Tilat luetaan tiedostosta aina kun sovellus käynnistetään. Ovikoodit tallennetaan eri tiedostoon ja luetaan aina kun käsiteltävä tila vaihtuu. Käyttäjälle näytetään siis vain valittuun tilaan liittyvät koodit. Taustasäie poistaa niitä sitä mukaa kun ne vanhenevat ohjelman käytön aikana. Voimassaolo tarkistetaan myös jokaisen käynnistyksen ja tilan vaihtumisen yhteydessä.



Kuvio 7. Hallintasovelluksen pääikkuna

Ohjelmoitaessa piti päättää, millainen SMS-viesti vastaanottimelle lähetetään. PDU-koodausta ei käytetty, koska tekstimuoto on ohjelmoinnin kannalta yksinkertaisempi. Viestissä on välilyönnillä eroteltuna avainsana, koodin tunnistenumero, koodi, sekä koodin alkamis- ja päättymisaikat.

Avainsanan perusteella päätellään onko kyseessä uuden koodin lisääminen tai muokkaaminen (CODE) vai olemassa olevan koodin poistaminen (DELETE). Tunniste auttaa kohdistamaan halutun toiminnon oikeaan koodiin. Voimassaolon alkamis- ja päättymisaika esitetään muodossa pp.kk.vvvv tt:mm:ss (24 h kello). DELETE-komennossa avainsanan jälkeen luetellaan vain tunnistenumerot väli-

lyönnillä erotettuna. Muokkauksen jälkeen näkyvillä olevat koodit järjestetään voimassaolon alkamisajan mukaan.

Koska laitteisto on jokaisella asiakkaalla hieman erilainen, AT-komentojen lähettämiseen käytetty sarjaportti voi vaihdella. Sen nimi on kuitenkin muotoa **COMx**, missä x on positiivinen kokonaisluku. Oikea portti selvitetään ohjelmallisesti SystemManagementObjectSearcher-oliota käyttämällä (Salmon 8.9.2010).

Kaikki MF636+:n asetukset eivät tallennu muistiin. Ainakin viestin muodon asetus katoaa, jos modeemista katkaistaan virta. Niinpä komento AT+CMGF=1 annetaan modeemille joka kerta ennen viestin lähetystä.

Eräs tärkeä asia on käyttäjän syötteen validointi. Se estää ohjelmaa kaatumasta virheellisesti syötetyn tiedon vuoksi. Hallintasovellus tarkistaa, että puhelinnumero on 10 merkkiä pitkä ja sisältää vain numeroita kuten koodikin, ja että sama tilan nimi esiintyy vain kerran. Nimellä erotetaan eri tilojen koodit toisistaan. Uutta koodia ei voi luoda, jos mitään tilaa ei ole valittuna. Tilaa kohti voi olla voimassa vain yksi koodi kerrallaan. Käyttäjälle ilmoitetaan, jos tekstiviestin lähetys ei onnistu. Koodin pituus on rajoitettu 25 numeroon.

5.2.3 Vastaanotto

Vastaanottopäässä on viestin ja käyttäjän syötteen odottamista sekä koodin voimassaolon tarkistamista varten kolme ikuista while-silmukkaa. Koska silmukka pysäyttää aina muun ohjelman suorituksen kunnes toistoehto ei enää päde, tarvitaan säikeet, jotta silmukoita voidaan suorittaa samanaikaisesti. Jokaiselle silmukalle on siis yksi säie. Säikeiden luomisessa hyödynnettiin pthread.h-kirjastoa ja siitä löytynyttä esimerkkiä (Ippolito [viitattu 22.5.2012]). Käyttöä helpottamiseksi ja sähkökatkon varalta ohjelma määritettiin käynnistymään automaattisesti Debian Linuxin mukana luomalla siihen symbolinen linkki (vastaa Windowsin pikakuvaketta) kansioon /etc/rc2.d. Graafista käyttöliittymää sovellukselle ei tarvita. Erityisesti sulautettuja järjestelmiä ohjelmoitaessa on pyrittävä säästämään laitteiston resursseja.

Viestin vastaanotossa hyödynnettiin Keijo Kasvin (2010) kehittämää SMS Server Tools 3 -ohjelmistoa. Se perustuu AT-komentoihin ja toimii Linuxissa taustapalveluna (daemon). Kun viesti saapuu, SMS Server Tools hakee sen modeemilta ja luo uuden tiedoston kansioon, jota tarkkaillaan jatkuvasti. Vastaanotto-ohjelmisto lukee viestin, poistaa sen kansiota ja käsittelee kooditiedostoa. Muokkauksen jälkeen koodit järjestetään voimassaolon alkamisajan mukaan. Yhden viestin vastaanottaminen kestää n. 5 – 10 s (Kasvi 2010). Kesto riippuu matkapuhelinverkon kuormituksesta.

Linux-käyttöjärjestelmässä SMS Server Toolsin lähdekoodi piti asennusohjeen (Kasvi 2010) mukaisesti purkaa paketista ja kääntää. Windowsille on valmis exe-tiedosto. Ratkaisuun päädyttiin, koska Huaweiin modeemilta ei muuten saatu minikäänlaista merkkiä saapuneesta viestistä. Myöskään sellainen lähestymistapa, että viestejä olisi käyty tarkistamassa vähän väliä, ei oikein toiminut. Toisaalta se on joka tapauksessa loogisesti huono vaihtoehto, ja kuluttaa turhaan resursseja.

Tässä tapauksessa on epätodennäköistä, että useampi säie yrittäisi kirjoittaa samaan tiedostoon yhtä aikaa. Silti kirjoitusoperaatiot on kaiken varalta suojattu siten, ettei sellaista pääse tapahtumaan. Samanaikainen kirjoitus voi aiheuttaa ottamattomia virheitä ohjelman toiminassa. Tämän vuoksi tiedoston käsittelyyn liittyvät käskyt koottiin yhteen funktioon. Suojaus tehtiin Mutexilla. Mutex on synkronointimekanismi, joka keskeyttää kaikkien niiden säikeiden toiminnan, jotka yrittävät kutsua funktiota silloin kun toinen säie suorittaa operaatiota. (Ippolito [viitattu 22.5.2012].)

GPIO-pinnien ohjaukseen on Raspberry Pi -kehittäjien sivustolla (RPI Low-level peripherals 21.5.2012) esitelty useita eri tapoja. Tavat eroavat toisistaan siinä, millä ohjelmointikielellä ja miten laiteläheisesti ne on toteutettu, mitä käyttöoikeuksia vaaditaan sekä millainen pinnien numerointi (ks kohta 5.1) on valittu. Helpoin tapa ei vaadi varsinaista ohjelmointia, ainoastaan komentorivikäskyjä.

Esimerkki: asetetaan GPIO pin 4 tuloksi ja GPIO pin 5 lähdeksi, luetaan pin 4:n arvo ja kirjoitetaan pin 5:een arvo 1:

```
Kirjoitetaan "23" tiedostoon /sys/class/gpio/export
Kirjoitetaan "in" tiedostoon /sys/class/gpio/gpio23/direction
```

```
Kirjoitetaan "24" tiedostoon /sys/class/gpio/export
Kirjoitetaan "out" tiedostoon /sys/class/gpio24/direction
Luetaan tiedosto /sys/class/gpio/gpio23/value
Kirjoitetaan "1" tiedostoon /sys/class/gpio/gpio24/value
```

23 ja 24 ovat GPIO-pinnejä 4 ja 5 vastaavat järjestelmäpiirin nastat, kuten kohdassa 5.1 todettiin. Nämä numerot tiedoston polussa luonnollisesti vaihtelevat sen mukaan, mitä nastaa halutaan ohjata. Pinnit voidaan vapauttaa käytöstä kirjoittamalla niiden numerot tiedostoon /sys/class/gpio/unexport. Tällöin ne eivät ole tuloja eivätkä lähtöjä. Mikään toinen ohjelma ei voi käyttää pinnejä ennen kuin ne on vapautettu.

Pihlajamaan (3.7.2012) testin perusteella voidaan päätellä, että komentorivipohjaisen lähestymistavan huono puoli on melko vaatimaton suorituskyky. Lisäksi ohjaus on suoritettava pääkäyttäjän (root) oikeuksilla tai ainakin käyttäjätillillä, jonka alaisuudessa ohjelmaa ajetaan, pitää olla luku- ja kirjoitusoikeudet edellä mainittuihin tiedostoihin. Yksi mahdollisuus on käsitellä suoraan prosessorin rekisterejä. Siinäkin tarvitaan root-oikeudet, mutta se on paljon tehokkaampaa, ja Gordon Hendersonin kehittämä wiringPi-kirjasto (Henderson 2012) helpottaa sitä huomattavasti. Kyseinen kirjasto on tehty alun perin C-kielellä, mutta sen käyttäminen onnistuu myös C++:lla. Se oli siis varsin hyödyllinen.

6 TYÖN TULOKSET

6.1 Tulokset

Projekti ei aivan pysynyt aikataulussa, mutta onnistui hyvin. Eniten aikaa kului Raspberryn toimituksen odotteluun, vaikka sinä aikana voitiinkin tehdä taustatyötä ja keskittyä muihin osa-alueisiin. Myöskään GSM-modeemin toimimaan saaminen Raspberrissä ei ollut täysin ongelmaton, kuten luvussa 5.2.1. todettiin. Vastoin-käymisistä huolimatta päästiin tavoitteeseen.

6.2. Yhteenveto

Yhteenvetona voidaan todeta, että vähäisen keskusmuistin ja virransyöttökyvyn takia Raspberrillä ei välttämättä pysty toteuttamaan kovin vaativia sovelluksia. Myös laitetuessa on vielä kehittämisen varaa. Silti kyseessä on hintaansa nähden kelvollinen alusta, koska aina ei tarvita suurta tehoa. Positiivisia asioita Raspberrissä ovat halpa hinta, helppo käyttöönotto, monipuoliset liitännät, äänettömyys, alhainen virrankulutus ja grafiikkapiirin suorituskyky.

Työ vaikutti alkuun melko haastavalta, mutta lähemmässä tarkastelussa ei osoittautunut lainkaan mahdottomaksi. Työssä tuli kerrattua Linuxin peruskäyttöä. Tiedonhaku helpotti Debianin samankaltaisuus verrattuna Ubuntuun. En ollut aikaisemmin käyttänyt AT-komentoja tai matriisinäppäimistöä tai ohjelmoinut Linux-tietokoneella, joten työ tarjosi myös paljon uutta tietoa ja arvokasta kokemusta.

LÄHTEET

- ARM Company Milestones. 2012. [Verkkajulkaisu]. ARM Ltd. [Viitattu: 23.9.2012]. Saatavissa: <http://www.arm.com/about/company-profile/milestones.php>
- BCM2835 ARM Peripherals. 6.2.2012. [PDF-dokumentti]. Broadcom Europe Ltd. [Viitattu 22.7.2012]. Saatavissa: <http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>
- Capacity (SD/SDHC/SDXC). 2012. [Verkkajulkaisu]. SD Association. [Viitattu: 14.8.2012]. Saatavissa: <https://www.sdcard.org/developers/overview/capacity>
- Dimopoulos, S. Sakis3G. 2010. [Verkkajulkaisu]. [Viitattu: 20.8.2012]. Saatavissa: <http://www.sakis3g.org>
- FAQs | Raspberry Pi. Ei päiväystä. [Verkkajulkaisu]. [Viitattu: 23.9.2012]. Saatavissa: <http://www.raspberrypi.org/faqs>
- Halfacree, G. 29.2.2012. Raspberry Pi interview: Eben Upton reveals all. [Verkkajulkaisu]. Linux user. [Viitattu: 25.5.2012]. Saatavissa: <http://www.linuxuser.co.uk/features/raspberry-pi-interview-eban-upton-reveals-all>
- Harvey, G. 28.4.2010. Installing 3G USB Modems On Linux. [Verkkajulkaisu]. [Viitattu: 24.9.2012]. Saatavissa: <http://www.drupaler.co.uk/blog/installing-3g-usb-modems-linux/497>
- Henderson, G. 2012. Raspberry Pi | Wiring. [Verkkajulkaisu]. Gordons Projects. [Viitattu: 19.7.2012]. Saatavissa: <https://projects.drogon.net/raspberry-pi/wiringpi>
- Huhtama, K. Harraste Elektroniikka. Ei päiväystä. [Verkkajulkaisu]. [Viitattu: 21.9.2012]. Saatavissa: <http://koti.mbnet.fi/~huhtama/ele/index.php?si=ml21.sis>
- Ippolito, G. Ei päiväystä. Linux Tutorial: POSIX Threads. [Verkkajulkaisu]. [Viitattu: 22.5.2012]. Saatavissa: <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>
- Jaakkola, M. 5.8.2004. ASCII-koodi. [Verkkajulkaisu]. Päijät-Hämeen koulutuskeskerni. [Viitattu: 21.9.2012]. Saatavissa: <http://edu.phkk.fi/opiskelu/cppohjtekni/ascii.htm>

- Janssen, C. What is Concatenated Short Message Service? Ei päiväystä. [Verkkajulkaisu]. [Viitattu: 24.9.2012]. Saatavissa: <http://www.techopedia.com/definition/24193/concatenated-short-message-service-concatenated-sms>
- Kasvi, K. 2010. SMS Server Tools 3. [Verkkajulkaisu]. [Viitattu: 8.7.2012]. Saatavissa: <http://smstools3.kekekasvi.com>
- Kurri, S. X86-arkkitehtuuri täytti 30 vuotta. 9.6.2008. [Verkkajulkaisu]. [Viitattu: 21.9.2012]. Saatavissa: <http://muropaketti.com/x86-arkkitehtuuri-taytti-30-vuotta>
- Learn about Subscriber Identity Module in GSM | SIM. 28.7.2010. [Verkkajulkaisu]. Learn Telecom. [Viitattu: 21.9.2012]. Saatavissa: <http://learntelecom.com/telephony/gsm/subscriber-identity-module>
- McNamara, T. Ei päiväystä. What is the Difference Between Serial ports & USB Ports? [Verkkajulkaisu]. [Viitattu: 23.9.2012]. Saatavissa: http://www.ehow.com/facts_5790112_difference-serial-ports-usb-ports_.html
- .NET Framework system requirements. 2011. [Verkkajulkaisu]. Microsoft. [Viitattu: 25.5.2012]. Saatavissa: <http://msdn.microsoft.com/en-us/library/8z6watww%28v=vs.100%29.aspx>
- Pettersson, L. SMS messages and the PDU mode. Ei päiväystä. [Verkkajulkaisu]. [Viitattu 23.7.2012]. Saatavissa: <http://www.dreamfabric.com/sms/>
- Pihlajamaa, J. Benchmarking Raspberry Pi GPIO Speed. 3.7.2012. [Verkkajulkaisu]. Code and Life. [Viitattu: 3.9.2012]. Saatavissa: <http://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/>
- PIN code (Personal Identification Number) – Mobile terms glossary. Ei päiväystä. [Verkkajulkaisu]. GSMarena.com. [Viitattu 24.9.2012]. Saatavissa: <http://www.gsmarena.com/glossary.php3?term=pin-code>
- Pitkänen, S. H. SSH - Secure Shell - Tieto- ja viestintätekniiikan käytön oppaita ja oppimateriaaleja. 16.8.2012. [Verkkajulkaisu]. Itä-Suomen yliopisto. [Viitattu: 21.9.2012]. Saatavissa: <https://wiki.uef.fi/display/opkmateriaalit/SSH+-+Secure+Shell>
- Rantala, P. 2010. Tietokonetekniikan laboraatiot: Matriisinäppäimistön toimintaperiaate. [Powerpoint-esitys]. Oulun seudun ammattikorkeakoulu. [Viitattu: 15.5.2012]. Saatavissa: <http://www.tekniikka.oamk.fi/~pekkar/materiaalia/tietokonelabrat/Matriisinappaimisto.ppt>

- Raspberry Pi Schematics. 17.4.2012. [PDF-dokumentti]. Raspberry Pi Foundation. [Viitattu 15.5.2012]. Saatavissa: <http://www.raspberrypi.org/wp-content/uploads/2012/04/Raspberry-Pi-Schematics-R1.0.pdf>
- RPi Low-level peripherals - eLinux.org. 21.5.2012. [Verkkajulkaisu]. Raspberry Pi Wiki. [Viitattu: 22.5.2012]. Saatavissa: http://elinux.org/Rpi_Low-level_peripherals
- Resize SD Card Partitions. 1.6.2012. [Verkkajulkaisu]. Raspberry Pi Spy. [Viitattu: 13.8.2012]. Saatavissa: <http://www.raspberrypi-spy.co.uk/2012/06/resize-sd-card-partitions>
- Salmon, M. 8.9.2010. Matt Salmon's Tech Blog: .NET - Detecting Modem COM ports. [Verkkajulkaisu]. [Viitattu: 1.6.2012]. Saatavissa: <http://salmontech.blogspot.fi/2010/09/net-detecting-modem-com-ports.html>
- Sangeelee, K. 26.6.2012. Raspberry Pi – Driving a Relay using GPIO. [Verkkajulkaisu]. SusaNET. [Viitattu: 21.7.2012]. Saatavissa: <http://www.susa.net/wordpress/2012/06/raspberry-pi-relay-using-gpio>
- Seghers, A. 10.4.2012. Raspbery Pi Passes CE, Now finally Available – News – Trusted Reviews. [Verkkajulkaisu]. IPC Media. [Viitattu: 31.5.2012]. Saatavissa: <http://www.trustedreviews.com/news/raspberry-pi-passes-ce-now-finally-available>
- Send and receiving SMS using AT command with a GSM modem. Ei päiväystä. [Verkkajulkaisu]. TXTOnline. [Viitattu: 22.5.2012]. Saatavissa: http://www.txtonline.com/sms_at_commands.htm
- SMS Tutorial: Example Demonstrating How to Use the +CMGS AT Command to Send SMS Text Messages in SMS Text Mode. Ei päiväystä. [Verkkajulkaisu]. [Viitattu: 24.9.2012]. Saatavissa: <http://www.developershome.com/sms/cmgsCommand3.asp>
- SMS Tutorial: Introduction to wireless modems, GSM Modems and GPRS Modems. Comparison of Mobile Phones with GSM/GPRS Modems. Ei päiväystä. [Verkkajulkaisu]. [Viitattu: 21.9.2012]. Saatavissa: <http://www.developershome.com/sms/GSMModemIntro.asp>
- Teknologiasanasto. 2012. [Verkkajulkaisu]. AfterDawn Oy. [Viitattu 21.9.2012]. Saatavissa: <http://fin.afterdawn.com/sanasto/>
- Using Eclipse to Cross-compile, Part 3: Create and Configure a Project. Ei päiväystä. [Verkkajulkaisu]. Lakeview Research. [Viitattu 15.5.2012]. Saatavissa: <http://www.lvr.com/eclipse3.htm>

Vinet, J. zeroflux.org. 9.6.2009. [Verkkajulkaisu]. [Viitattu 15.5.2012]. Saatavissa:
<http://www.zeroflux.org/blog/post/255>

Windows XP –käyttöjärjestelmien järjestelmävaatimukset. 2010. [Verkkajulkaisu].
Microsoft. [Viitattu: 31.5.2012]. Saatavissa:
<http://support.microsoft.com/kb/314865>