

Henri Vares

Moottorinohjainlaitteen diagnoosi etäyhteydellä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Auto- ja kuljetustekniikka

Insinöörityö

22.4.2014

Tekijä(t) Otsikko	Henri Vares Moottorinohjainlaitteen diagnoosi etäyhteydellä
Sivumäärä Aika	16 sivua + 1 liite 22.4.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Auto- ja kuljetustekniikka
Suuntautumisvaihtoehto	Autosähkötekniikka
Ohjaaja(t)	Lehtori Vesa Linja-aho, Metropolia-amk
<p>Tässä insinöörityössä suunniteltiin ja toteutettiin ajoneuvon moottorinohjainlaitteen etä-diagnoosin mahdollistava laite. Laite pystyy lukemaan ja poistamaan vikakoodit useimmista autoista auton sijainnista riippumatta etäyhteyden avulla.</p> <p>Tässä työssä tarkastellaan Raspberry Pi:n ja ELM327:n toimintaa, vikakoodien lukuun soveltuvia vapaan lähdekoodin ohjelmia ja ohjeistetaan laitteen toteutus ja käyttö. Työssä ei syvennytä Raspberry Pi:n komentoihin tai käytettyjen ohjelmien toimintaperiaatteeseen.</p> <p>Työn tavoitteena oli toteuttaa laite, joka mahdollistaa auton vikakoodien ja muuttujien lukemisen vikatilanteiden sattuessa auton ollessa muualla kuin korjaamolla. Laitteen tarkoitus on mahdollistaa auton alustava diagnoosi auton vikaantuessa tien päällä. Etäyhteyden muodostanut henkilö voi näin ollen ohjeistaa asiakasta puhelimitse jatkotoimenpiteissä. Laite on tarkoitettu autoon lisävarusteeksi asennettavaksi, eikä se vaadi käyttäjältään autoalan tuntemusta.</p> <p>Laitteen toteutus onnistui suunnitelmien mukaisesti ja lopputulos täyttää sille asetetut tavoitteet. Laite muodostaa etäyhteyden nappia painettaessa, minkä jälkeen on mahdollista esimerkiksi lukea ajoneuvosta vikakoodit ja poistaa ne. Toinen painike asetettiin käynnistämään ajoneuvon anturitietojen tallennus ajon aikana.</p> <p>Laitteen toteutukseen on käytetty Raspberry Pi -tietokonetta, ELM327-OBD-lukijaa ja 3G-modeemia. Käytettävät ohjelmat ovat kaikki vapaaseen lähdekoodiin perustuvia ja ovat saatavilla kaikille.</p>	
Avainsanat	OBD, etäyhteys, Raspberry Pi, ELM327, 3G

Author(s) Title	Henri Vares Engine Control Unit Diagnose with Remote Access
Number of Pages Date	16 pages + 1 appendix 15 September 2012
Degree	Bachelor of Engineering
Degree Programme	Automotive and Transport Engineering
Specialisation option	Automotive Electronics Engineering
Instructor(s)	Vesa Linja-aho, Senior Lecturer in Automotive Electronics
<p>The aim of this Bachelor`s Thesis was to design and implement a device that enables remote diagnose of the Engine Control Unit in all vehicles. The device can read and erase fault codes regardless of the vehicles location using remote access.</p> <p>The thesis looks into the operation of the Raspberry Pi and ELM327, open-source software suitable for fault code reading and instructs the implementation and use of the device. The thesis does not look into the commands of Raspberry Pi or the principles of the open-source software.</p> <p>The aim of the thesis was to implement a device that enables an automotive professional to remotely read fault codes and variables of the engine control unit if the vehicle malfunctions on the road. The person who opens the remote access can therefore instruct the customer for follow-up actions, possibly without the need for the customer to take the vehicle to a repair shop. The device is designed to be installed to vehicles as an accessory.</p> <p>As a result the device implemented in this thesis met its requirements. The device forms a remote access connection when a button is pressed and can then be used to read vehicles fault codes and erase them. Another button was set to start a data logging software which starts recording data from vehicles sensors.</p> <p>Components used for the implementation of the device were Raspberry Pi, ELM327 OBD-reader and 3G-modem. The used software is all open-source, and therefore, free to be used by anyone.</p>	
Keywords	OBD, remote access, Raspberry Pi, ELM327, 3G, open-source

Sisällys

Lyhenteet

1	Johdanto	1
2	Käytetyt komponentit	2
2.1	Raspberry Pi	2
2.2	3G-modeemi	2
2.3	ELM327	3
3	Laitteen toteutus	3
3.1	Raspbianin asennus	3
3.2	3G-modeemin asennus	4
3.3	Etäyhteys	7
3.4	OpenOBD ja Perl OBD-II Logger	10
3.5	Painikkeet ja ohjelmien käynnistys	12
4	Yhteenveto	15
	Lähteet	16

Liitteet

Liite 1. Valmis laite ilman kotelo

Lyhenteet

OBD	<i>On Board Diagnostics</i> . Auton ohjainlaitteiden komentoja ja vastauksia valvova järjestelmä.
Linux	Unix-pohjainen käyttöjärjestelmä.
GPIO	<i>General Purpose I/O</i> . Yleiskäyttöinen portti mikrokontrollereissa ja mikroprosessoreissa, joka voidaan ohjelmoida signaalin vastaanottajaksi tai lähettäjäksi.
ARM	<i>Advanced RISC Machines</i> . 32-bittinen mikroprosessoriarkkitehtuuri.
3G	Yleinen lyhenne "kolmannen sukupolven" matkapuhelinteknologiasta joka on tarkoitettu tiedonsiirtoon
IP-osoite	Numerosarja joka yksilöi jokaisen Internet-verkkoon kytketyn laitteen.
APN	<i>Access Point Name</i> . Yhteysportin nimi 3G-mobiiliverkon ja muun Internet-verkossa olevan laitteen välillä.

1 Johdanto

Tämän toiminnallisen opinnäytetyön tavoitteena oli toteuttaa laite, joka mahdollistaa auton vikakoodien ja muuttujien lukemisen vikatilanteiden sattuesssa auton ollessa muualla kuin korjaamolla. Laitteen tarkoitus on mahdollistaa auton alustava diagnoosi auton vikaantuessa tien päällä ja mahdollisesti ohjeistaa asiakasta puhelimitse jatkotoimenpiteissä. Laite on tarkoitettu autoon lisävarusteeksi asennettavaksi, eikä se vaadi käyttäjältään autoalan tuntemusta. Vastaavia laitteita on saatavilla esimerkiksi raskaan kaluston käyttöön [1]. Tällä laitteella mahdollistetaan vastaavien ominaisuuksien käyttö kuluttajatasolla.

Nykyään autojen itsediagnoosijärjestelmät ovat niin laajoja, että ne valvovat kaikkia auton toimintoja. Jonkin toiminnon vikaantuessa auton mittaristoon syttyy valo merkiksi viasta ja ohjekirjassa kehoitetaan ottamaan yhteyttä merkkiliikkeeseen. Tällaisissa tapauksissa merkkiliikkeen työnjohtaja voisi ottaa tässä työssä toteutetun laitteen avulla etäyhteyden asiakkaan autoon ja lukea autosta vikakoodit. Tämän jälkeen työnjohtaja pystyy kirjaamaan saamansa tiedot työmääräykseen tai ohjeistamaan asiakasta tarvittavista jatkotoimenpiteistä. Auton käyttäjät voivat jossain määrin olla tietämättömiä vikavalon syttymisen tarkoituksesta ja sen vakavuudesta. Esimerkiksi joissain automalleissa sama valo voi syttyä, kun auton moottorissa on sytytyskatkoksia tai kun tuulilasinpesuneste on loppunut. Jos auton vikavalon on syttynyt sytytyskatkoksen seurauksena, asiakas ohjeistetaan lähimpään korjaamoon. Jos taas valo on syttynyt tuulilasinpesunesteen loppumisen vuoksi, asiakasta ohjeistetaan lisäämään nestettä, ettei asiakkaan tarvitse viedä autoa korjaamolle. Tällä toimintatavalla asiakas on tietoinen vikavalon syttymisen syystä eikä jätä mahdollisesti vakavaa vikaa huomiotta.

Laite on toteutettu käyttäen Raspberry Pi -tietokonetta, 3G-modeemia ja ELM327-OBD-adapteria. Vikakoodien luku tapahtuu vapaaseen lähdekoodiin perustuvalla OpenOBD-ohjelmalla ja tiedonkeruu Perl OBD-II Logger -ohjelmalla.

Työssä käydään läpi laitteen toiminnan mahdollistavat komponentit ja niiden asennus, sekä laitteen ohjelmointi ja kokoaminen. Työssä ei perehdytä OpenOBD:n- ja Perl OBD-II Logger -ohjelmien toimintaan.

2 Käytetyt komponentit

2.1 Raspberry Pi

Raspberry Pi on yhden piirilevyn tietokone, joka on noin luottokortin kokoinen. Sen on kehittänyt Raspberry Pi Foundation tarkoituksenaan luoda alusta ohjelmoinnin opettamiselle oppilaitoksissa. Tietokone sisältää muun muassa 700MHz:n ARM-prosessorin, 512 MB käyttömuistia, SD-korttipaikan, Ethernet-liittimen, HDMI-liittimen ja 2 USB-porttia. Raspberry Pi:n käyttöympäristö on ARM-arkkitehtuuriin perustuva Linux-käyttöjärjestelmä. Siitä on saatavilla eri versioita, kuten Arch, Raspbian ja Fedora. [2.] Tässä työssä on käytetty Raspbian-käyttöjärjestelmää. Raspberry Pi saa käyttövirtansa MicroUSB-liittimen kautta, jolloin se voidaan kytkeä esimerkiksi matkapuhelimen laturiin. Kaikki laitteen käyttämä tieto asennetaan SD-muistikortille. Laitteeseen tehtävät asennukset on suositeltavaa tehdä käyttäen kiinteää internetyhteyttä Raspberry Pi:n Ethernet-liittimen kautta.

2.2 3G-modeemi

Tässä työssä rakennettu laite muodostaa internetyhteyden 3G-modeemin avulla. Työssä käytetty modeemi on Huaweiin valmistama E367-3G-modeemi. Se tarvitsee toimiakseen matkapuhelimen SIM-kortin ja 3G-liittymän. Työssä käytetty 3G-liittymä on Saunalahden Mobiililaajakaista Tablet. Tämä liittymä valittiin laitteeseen käytettäväksi, koska laskutus perustuu siinä 3G-yhteyden käyttöpäiviin. Näin ollen liittymästä ei joudu maksamaan kuukausittain kiinteää summaa, vaan maksu määräytyy niiden päivien osalta, jolloin liittymää on käytetty. Tällä tavalla laitteen käyttö on mahdollisimman edullista. Mobiililaajakaista Tabletin hinta on 0,99 €/pv.

3G-modeemin toiminta perustuu langattomaan, matkapuhelinverkkojen tukiasemien kautta siirrettävään verkkoyhteyteen [3]. Modeemin yhdistyessä tukipisteeseen se saa IP-osoitteen, jolloin internet-yhteys on mahdollista. 3G-modeemin IP-osoite muuttuu aina kun yhteys katkaistaan ja muodostetaan.

Huawei E367 on vain osittain yhteensopiva Raspberry Pi:n kanssa johtuen modeemin ominaisuudesta käynnistyä ensin USB-muistina. Tämä ominaisuus johtuu siitä, että modeemin ajurit Windows-käyttöjärjestelmälle ovat ladattuna E367:n sisäiseen muistiin.

Tästä syystä modeemin asennus vaatii ylimääräisiä vaiheita ja sen toiminta voi ajoittain olla epävakaata. Huawei E220 on 3G-modeemi, joka on paremmin yhteensopiva Raspberry Pi:n kanssa. Tässä työssä käytettiin E367-modeemia sen edullisuuden ja saatavuuden vuoksi.

2.3 ELM327

ELM327 on OBD-sovitin jota käytetään auton ja tietokoneen väliseen tiedonsiirtoon, viikkoodien lukemiseen ja poistamiseen sekä moottorin parametrien lukemiseen. ELM327 tukee yhdeksää käytössä olevaa OBD-protokollaa, ja sen on valmistanut kanadalainen ELM Electronics. [4, s. 1.] ELM327:n ensimmäinen versio 1.0 julkaistiin vuonna 2005 ja tällä hetkellä viimeisin versio on 2.1. Tässä työssä on käytetty versiota 1.4b sen edullisuuden ja saatavuuden vuoksi. ELM327 käyttää tiedonsiirtoon RS232- tai USB-yhteyttä. Molemmat yhteysmuodot käyttävät sarjamuotoista tiedonsiirtoa, ja ne näkyvät kytkeytyessä tietokoneessa aina RS232-yhteytenä [4, s.7]. Tiedonsiirto tapahtuu lähettämällä ASCII-muodossa olevia pyyntöjä ELM327-laitteesta auton moottorinohjainlaitteeseen, joihin moottorinohjainlaite vastaa. Esimerkiksi jäähdytinnesteen lämpötilatieto saadaan lähettämällä ELM327:sta komento `>01 05` moottorinohjainlaitteelle. Moottorinohjainlaite vastaa tähän pyyntöön muodossa `41 05 7B`. Pyydetty jäähdytinnesteen lämpötila annetaan muodossa `7B`, joka vastaa desimaalimuodossa numeroa 123. Mutta koska arvot voivat myös olla negatiivisia, täytyy numerosta 123 vähentää 40, jolloin saadaan 83. Tämä on moottorin jäähdytinnesteen lämpötila pyydetyllä hetkellä. [4, s. 28, 29.]

3 Laitteen toteutus

3.1 Raspbianin asennus

Ensimmäinen vaihe laitteen toteutuksessa on asentaa SD-kortille Raspbian-käyttöjärjestelmä. Tämä voidaan tehdä joko ostamalla muistikortti, johon Raspbian on asennettu valmiiksi, tai lataamalla tarvittavat tiedostot Raspberry Pi:n. Sivuilta löytyy aina viimeisin versio kustakin saatavilla olevasta käyttöjärjestelmästä. Asentamista varten on käytetty

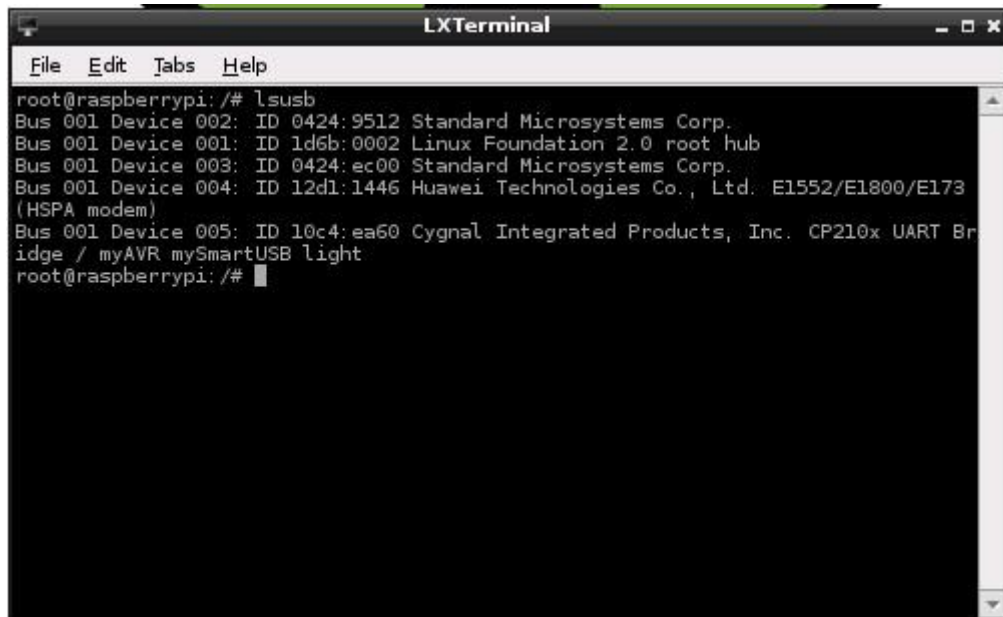
Windows-tietokonetta ja WIN32DiskImager-ohjelmaa. Kun Raspbian on ladattu, kyseinen IMG-päätteinen tiedosto kirjoitetaan muistikortille käyttäen Win32DiskImager-ohjelmaa. [4.]

Tämän jälkeen muistikortti asetetaan Raspberry Pi:n muistikorttipaikkaan, liitetään HDMI-johto, Ethernet-johto, hiiri ja näppäimistö. Raspberry Pi käynnistyy heti kun virtajohto kytketään siihen kiinni, jolloin se alkaa asentaa muistikortilla olevaa käyttöjärjestelmää. Asennuksen jälkeen avautuu konfiguraatioikkuna josta valitaan kohdat 2 ja 10. Valinta tapahtuu liikuttamalla kursoria näppäimistön nuolinäppäimillä ja valitsemalla haluttu kohta ENTER-näppäimellä. Konfiguraatioikkunan kohta 2 laajentaa käyttöjärjestelmälle varattua tilaa muistikortilta niin, että koko muistikortin tila tulee käytettäväksi. Kohta 10 määrittää Raspberry Pi:n käynnistymään aina työpöytätilaan eli graafiseen käyttöliittymään. Valittavissa on myös vaihtoehto, joka ei sisällä graafista käyttöliittymää. Tällöin Raspberry Pi käynnistyy vain komentorivitalaan. Asetukset tallennetaan valitsemalla kohta *<Finish>*, minkä jälkeen laite käynnistyy uudelleen ja kirjautuu aina automaattisesti graafiseen työpöytätilaan.

3.2 3G-modeemin asennus

Kaikki vaadittavat asennukset ja ohjelmat tehdään LXTerminal-ohjelmalla. Kirjoittamalla LXTerminalin komentoriville *sudo apt-get update*, päivitetään Raspbianiin saatavilla olevien laajennusten kirjasto viimeisimpään versioon. 3G-modeemin vaadittavat ohjelmat ja laajennukset ladataan ja asennetaan komennolla *sudo apt-get install ppp usb-modeswitch wvdial*. Jos käytössä on E220-modeemi, niin USB-Modeswitchin asennus voidaan ohittaa. PPP on protokolla, joka mahdollistaa suoran yhteyden kahden verkkolaitteen välillä. USB-Modeswitch on ohjelma, joka vaihtaa 3G-modeemin muistitilasta modeemi-tilaan. WVDial on 3G-yhteyden muodostamiseen käytettävä ohjelma.

Asennusten jälkeen E367 voidaan kytkeä Raspberry Pi:n USB-porttiin. Kirjoittamalla LXTerminalin komentoriville *lsusb*-komento, tulostuu lista kytketyistä laitteista (kuva 1). Tarpeellista tietoa tästä listasta ovat ID-numerot. Näitä käytetään USB-Modeswitch-ohjelman kokoonpanon muuttamisessa. Tämän jälkeen Raspberry Pi käynnistetään uudelleen komennolla *reboot*. Uudelleenkäynnistytyn jälkeen tarkastellaan taas *lsusb*-listaa, josta huomataan, että aikaisemmat ID-numerot ovat muuttuneet. Tämä tarkoittaa sitä, että USB-Modeswitch tekee tehtävänsä eli vaihtaa modeemin oikeaan tilaan.



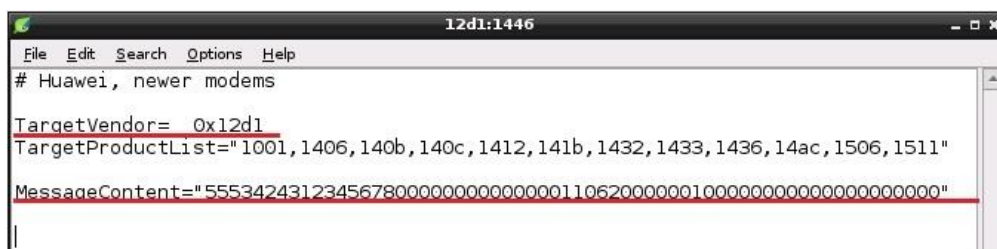
```

LXTerminal
File Edit Tabs Help
root@raspberrypi: /# lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 12d1:1446 Huawei Technologies Co., Ltd. E1552/E1800/E173
(HSPA modem)
Bus 001 Device 005: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Br
idge / myAVR mySmartUSB light
root@raspberrypi: /#

```

Kuva 1. Lista *lsusb*-komennolla tulostuvista laitteista. E367 on Device 004, jonka ID-numerot ovat 12d1:1446.

Kokoonpanoa voidaan muuttaa menemällä komentorivillä kansioon *tmp* ja purkamalla tiedosto, joka sisältää eri 3G-modeemien tarvittavat tiedot. Tämä tapahtuu syöttämällä komennot *cd /tmp* ja *tar -xzf /usr/share/usb_modeswitch/configPack.tar.gz xxxx\:yyyy*. *Xxxx* ja *yyyy* ovat ensimmäisellä *lsusb*-komennolla saadut ID-numerot. Purettu tiedosto avataan tekstieditorissa komennolla *leafpad xxxx:yyyy*. Avatusta tiedostosta kopioidaan kuvassa 2 näkyvät rivit. Tämän jälkeen avataan *usb_modeswitch.conf*-tiedosto komennolla *sudo leafpad /etc/usb_modeswitch.conf*. Tänne liitetään edellä mainitut tiedot kuvan 3 osoittamalla tavalla. Nyt käynnistäessä USB-Modeswitch vaihtaa 3G-modeemin aina oikeaan tilaan.



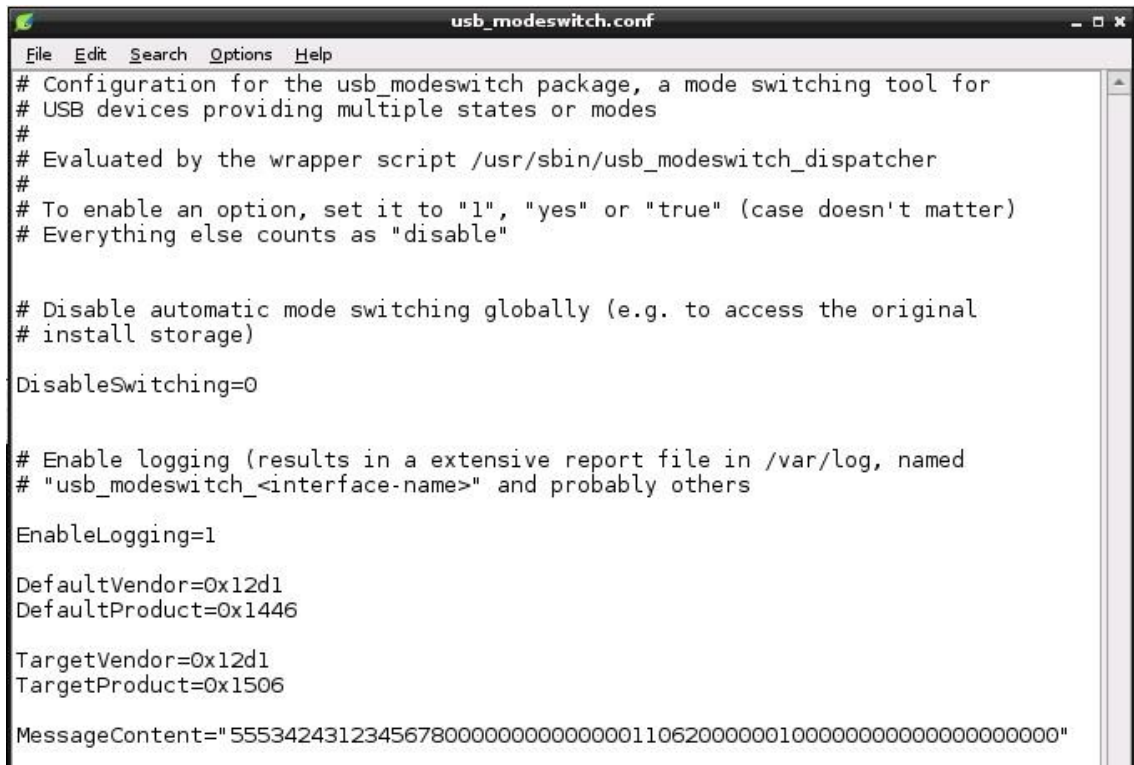
```

12d1:1446
File Edit Search Options Help
# Huawei, newer modems

TargetVendor= 0x12d1
TargetProductList="1001,1406,140b,140c,1412,141b,1432,1433,1436,14ac,1506,1511"
MessageContent="5553424312345678000000000000000011062000000100000000000000000000"

```

Kuva 2. Tiedostosta 12d1:1446 kopioitavat rivit.



```

usb_modeswitch.conf
File Edit Search Options Help
# Configuration for the usb_modeswitch package, a mode switching tool for
# USB devices providing multiple states or modes
#
# Evaluated by the wrapper script /usr/sbin/usb_modeswitch_dispatcher
#
# To enable an option, set it to "1", "yes" or "true" (case doesn't matter)
# Everything else counts as "disable"

# Disable automatic mode switching globally (e.g. to access the original
# install storage)
DisableSwitching=0

# Enable logging (results in a extensive report file in /var/log, named
# "usb_modeswitch_<interface-name>" and probably others
EnableLogging=1

DefaultVendor=0x12d1
DefaultProduct=0x1446

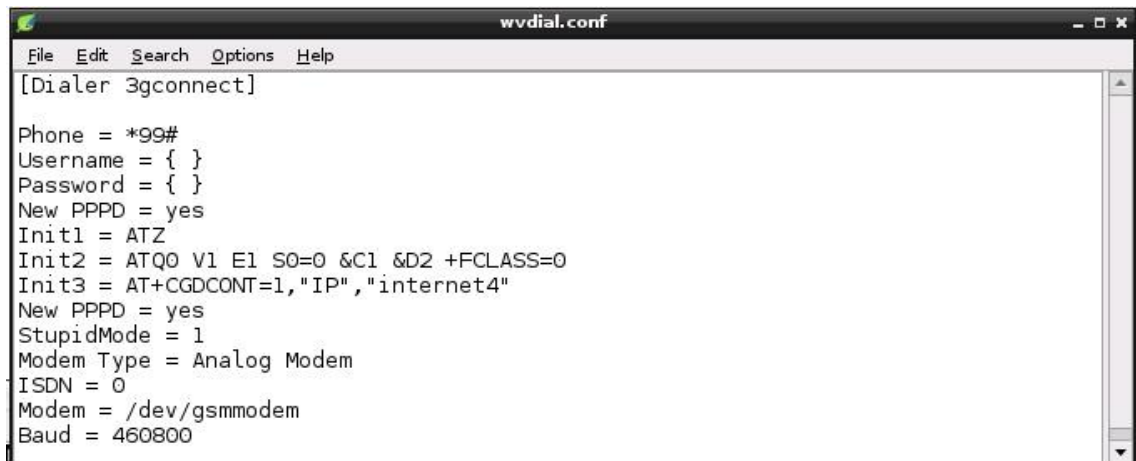
TargetVendor=0x12d1
TargetProduct=0x1506

MessageContent="555342431234567800000000000000011062000000100000000000000000000"

```

Kuva 3. Tiedosto `usb_modeswitch.conf`, johon on lisätty vaadittavat rivit.

Seuraavassa vaiheessa asetetaan WVDial toimimaan 3G-modeemin ja käytettävän liittymän kanssa. *Sudo leafpad /etc/wvdial.conf*-komennolla avataan tiedosto, joka sisältää WVDialin kokoonpanotiedot. Tähän tarvitaan muun muassa liittymän APN, puhelinnumero, käyttäjänimi ja salasana, jotka ovat saatavilla operaattorilta. Tätä työtä varten IP-osoitteen tulee olla julkinen. Saunalahden liittymät voidaan asettaa käyttämään julkista IP-osoitetta OmaSaunalahti-sivuilta. Kuvassa 4 on esimerkki `wvdial.conf`-tiedoston täyttämisestä. Ensimmäisellä rivillä annetaan tälle yhteydelle vapaavalinnainen nimi, jota käytetään jatkossa komentona 3G-yhteyden muodostamiseen. Internet-yhteyden muodostaminen tapahtuu komennolla *wvdial 3gconnect*.



```

wvdial.conf
File Edit Search Options Help
[Dialer 3gconnect]
Phone = *99#
Username = { }
Password = { }
New PPPD = yes
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","internet4"
New PPPD = yes
StupidMode = 1
Modem Type = Analog Modem
ISDN = 0
Modem = /dev/gsmmodem
Baud = 460800

```

Kuva 4. Esimerkki työssä käytetystä wvdial.conf-tiedostosta.

Seuraavat rivit täytyy vielä syöttää komentoriville, jotta kaikilla yhteyden muodostamiseen käytettävillä ohjelmilla olisi oikeudet avata yhteys:

- `chmod +s /usr/sbin/pppd`
- `chmod 777 /usr/sbin/pppd`
- `chmod 777 /etc/ppp/chap-secrets`
- `chmod 777 /etc/ppp/pap-secrets`
- `chmod 777 /etc/ppp/peers/`
- `chmod 777 /etc/ppp/peers/wvdial`
- `chmod 777 /etc/ppp/peers/wvdial-pipe.`

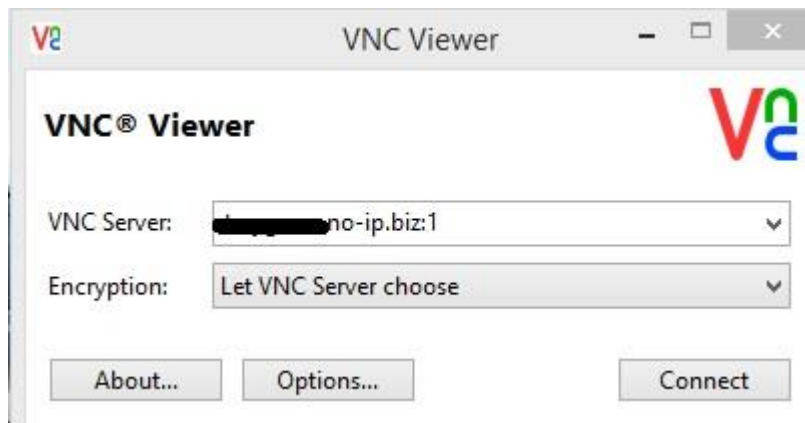
3.3 Etäyhteys

No-IP on palvelu, joka seuraa jatkuvasti käytössä olevan laitteen muuttuvaa IP-osoitetta sekä ohjaa IP-osoitteen käyttäjän määrittämään osoitteeseen [5]. Tätä palvelua tarvitaan tämän laitteen toimintaan, koska 3G-modeemin IP-osoite muuttuu joka kerta, kun yhteys avataan. Jos IP-osoite ei ole tiedossa, etäyhteyden muodostaminen ei onnistu. Palvelu voidaan ottaa käyttöön noip.com-sivustolla. Palvelu on ilmainen, jos seurattavia IP-osoitteita on vain yksi.

Asennusta varten täytyy tehdä NoIP:lle kansio komennolla `mkdir/home/pi/noip`, ja tähän kansioon siirrytään komennolla `cd /home/pi/noip`. Tämän jälkeen ladataan tarvittava palvelu suoraan NoIP:n sivuilta komennolla `wget http://www.no-ip.com/client/linux/noip-duc-linus.tar.gz` ja kyseinen tiedosto puretaan kansioon komennolla `tar vzx noip-duc-linus.tar.gz`. Seuraavaksi siirrytään juuri purettuun kansioon komennolla `cd noip-2.1.9-1` ja asennetaan palvelu komennoilla `sudo make` ja `sudo make install`. Asennuksen aikana täytyy syöttää No-IP-sivulle tehdyt käyttäjätunnukset ja aikaväli, jona No-IP päivittää IP-osoitteen. Palvelu käynnistetään komennolla `sudo noip2`.

Seuraavassa vaiheessa asennetaan ohjelma, joka mahdollistaa etätyöpöytäyhteyden muodostamisen. Tätä varten tarvitaan TightVNC-ohjelma, joka avaa Raspberry Pi:hin palvelimen, joka lähettää työpöytäyhteyden IP-osoitteen yli vastaanottavaan laitteeseen. Vastaanottamista varten tarvitaan TightVNC Viewer -ohjelma, joka on saatavilla Windows-tietokoneisiin ja esimerkiksi myös Android-puhelimiin.

TightVNC asennetaan Raspberry Pi:hin komennolla `sudo apt-get install tightvncserver`. Asennuksen aikana kysytään, että asennetaanko ilman varmistusta, ja tähän vastataan `y` ja painetaan Enter. Asennuksen jälkeen TightVNC-palvelin voidaan käynnistää. Tässä työssä käytettiin komentoa `vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565`. Tällä komennolla käynnistetään palvelin nimeltään "1", jonka resoluutio on 1280 x 800. Resoluutio voidaan tarvittaessa vaihtaa vastaamaan vastaanottavan laitteen näytön tarkkuutta. Nyt etätyöpöytäyhteys voidaan avata vastaanottavassa laitteessa esimerkiksi VNC Viewer -ohjelmalla. Etätyöpöytäyhteyden muodostamista varten tarvitaan palvelimen osoite, joka on muodossa `käyttäjänimi.no-ip.biz`. Tämä osoite luodaan No-IP:n verkkosivuilla. Tähän osoitteeseen No-IP-palvelu lähettää viimeisimmän IP-osoitteen. Osoite syötetään VNC Viewer -ohjelmaan muodossa `käyttäjänimi.no-ip.biz:1`. Osoitteen loppuun lisätään aikaisemmin avatun palvelimen nimi, tässä tapauksessa se on 1. (kuva 5). Jotta palvelin avautuisi automaattisesti Raspberry Pi:n käynnistyessä, täytyy sitä varten luoda oma tiedosto. Tiedosto luodaan komennolla `sudo nano /etc/init.d/vncboot`. Tämä tiedosto täytetään kuvan 6 osoittamalla tavalla. Valmis tiedosto on muutettava suoritettavaan muotoon komennolla `sudo chmod 755 /etc/init.d/vncboot`. Jotta tämä tiedosto käynnistyisi ja sammuisi Raspberry Pi:n mukana, täytyy se lisätä käynnistyvien ohjelmien listalle komennolla `sudo update-rc.d vncboot defaults`.



Kuva 5. Etättyöpöydän avaaminen VNC Viewer -ohjelmalla.

```

File Edit Tabs Help
GNU nano 2.2.6 File: /etc/init.d/vncboot

## BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: VNC Server start on boot
## END INIT INFO

#!/bin/sh
# /etc/init.d/vncboot

USER=pi@raspberrypi
HOME=/root

export USER HOME

case "$1" in
  start)
    echo "Starting VNC Server"
    /usr/bin/vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565
    ;;
  stop)
    echo "Stopping VNC Server"
    /usr/bin/vncserver -kill :1
    ;;
  *)
    echo "Usage: /etc/init.d/vncboot {start|stop}"
    exit 1
    ;;
esac

exit 0

```

[Read 35 lines]

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell

Kuva 6. Esimerkki tiedoston *vncboot* sisältämistä tiedoista.

3.4 OpenOBD ja Perl OBD-II Logger

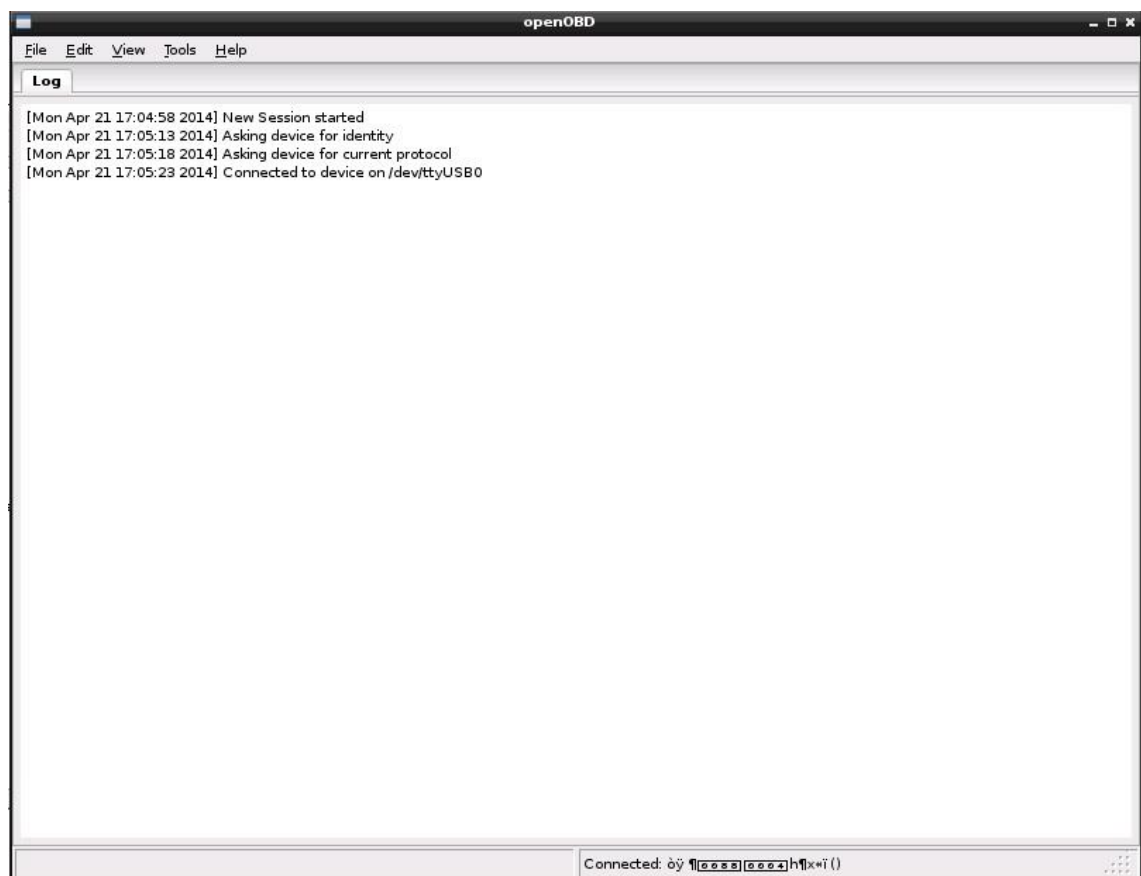
Tässä luvussa käsitellään vikakoodien ja parametrien lukuun käytettävien ohjelmien toimintaa ja asennusta. Käytettävät ohjelmat ovat openOBD ja Perl OBD-II Logger.

OpenOBD:n asennusta varten täytyy ladata ohjelman internet-sivuilta ohjelman lähdekoodi [6.]. Lähdekoodi on lista komennoista ja käskyistä, joilla ohjelma saadaan koottua ja asennettua. Jotta annetut käskyt olisivat tietokoneelle ymmärrettävässä muodossa, tarvitaan erilaisia apuohjelmia ja kirjastoja, jotka kokoavat asennettavan ohjelman käskyjen mukaisesti.

Kun ohjelman lähdekoodi on ladattu, se täytyy purkaa kansioon kuten esimerkiksi `/home/pi/openOBD/`. Tämä tapahtuu siirtymällä komentorivillä kansioon `/home/pi/` komennolla `cd /home/pi`. Tähän luodaan uusi kansio komennolla `mkdir openOBD`. Ladattu lähdekoodi on nyt siirrettävä tähän kansioon ja lähdekoodi puretaan komennolla `tar xvfz openobd-0.5.0.tar.gz`. Ohjelman asennusta varten vaadittavat ohjelmat ovat CMake, wxWidgets ja sqlite3. Kyseiset ohjelmat asennetaan komennolla `sudo apt-get install libsqlite3-dev wx-common gettext python-wxgtk2.8 libwxgtk2.8 libwxgtk2.8-dev libgtk2.0-dev`. Asennusten jälkeen siirrytään komentorivillä lähdekoodin sisältävään kansioon komennolla `cd /home/pi/openOBD/`. Tämän kansion sisälle luodaan asennusta vasten uusi kansio komennolla `mkdir build` ja siirrytään juuri luotuun kansioon komennolla `cd build`. Ohjelman asennus aloitetaan komennolla `cmake ../`. Tällä komennolla luetaan lähdekoodi ja varmistetaan, että vaadittavat kirjastot löytyvät. Komennolla `make` ohjelma asennetaan build-kansioon. Kun asennus on suoritettu loppuun, ohjelman toimintaan vaadittava kirjasto on siirrettävä kansioista `/home/pi/openOBD/data/` kansioon `build`. Tämä tiedosto on nimeltään `openobd.db`. Oletuksena openOBD-ohjelma käyttää yhdistämiseen RS232-liittimellä varustettua ELM327-laitetta. Koska tässä työssä on käytetty USB-liittimellä varustettua ELM327-laitetta, tulee tämä ominaisuus muuttua. Kansioista `build` löytyy tiedosto `openOBD.conf` (kuva 7). Muuttamalla kohta `Port` vastaamaan käytettävää USB-porttia, saadaan openOBD avaamaan yhteys oikeaan porttiin. Muuttamalla kohdassa `AtStartup` oleva numero 0 numeroksi 1, asetetaan openOBD käynnistymään automaattisesti silloin, kun etätyöpöytä avataan. Tämän jälkeen ohjelma voidaan avata komennolla `/root/build/openobd` (kuva 8).



Kuva 7. Tiedosto openOBD.conf josta löytyy OpenOBD-ohjelman asetukset.



Kuva 8. OpenOBD-ohjelma.

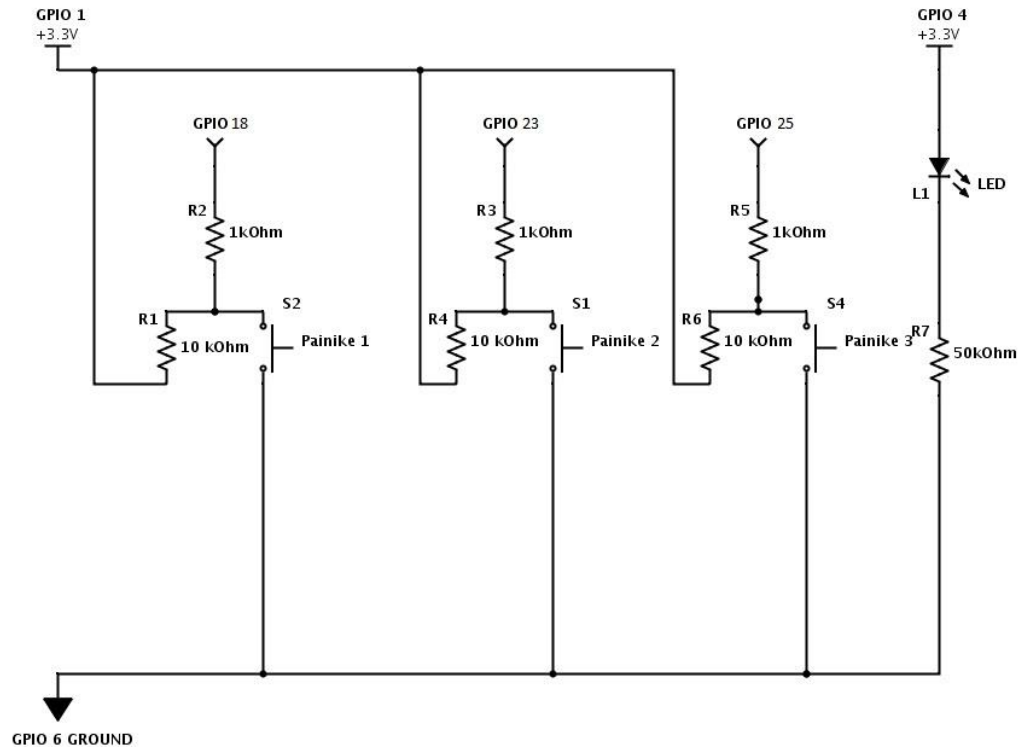
Perl OBD-II Logger -ohjelman asennus on hieman yksinkertaisempi, eikä se vaadi lisäkirjastoja tai asennusohjelmia. Ohjelman asennuskirjasto voidaan ladata ohjelman internetsivuilta. [7.] Ohjelman tiedostot puretaan komennolla `tar xvf pobd_logger-0.01.gz`. Purettuun kansioon siirrytään komennolla `cd pobd_logger-0.01`. Tässä kansiossa kun

annetaan komento `sudo make install-init.pi`, luetaan Raspberry Pi:lle tarkoitettu lähdekoodi ja aloitetaan asennus. Asennuksen jälkeen ohjelma käynnistetään komennolla `pobd_logger -v /dev/ttyUSB0 38400`.

3.5 Painikkeet ja ohjelmien käynnistys

Jotta laitetta voidaan käyttää ilman näyttöpaneelia, sen toimintaan vaaditaan painikkeet. Painikkeista voidaan valita halutut toiminnot kuten etäyhteyden avaus, reaaliaikatietojen tallennus ja laitteen sammutus. Laitteeseen tulee myös yksi LED-valo, jonka tehtävänä on ilmoittaa, koska laite on valmis vastaanottamaan käskyjä painikkeilta.

Painikkeet ja LED-valo kytketään Raspberry Pi:n GPIO-pinneihin. Kuvassa 9 on kytkentäkaavio työssä käytetystä kytkennästä. GPIO-pinnien ohjaamiseen käytetään Python-ohjelmointikieltä ja sitä varten Raspbianissa on ohjelma nimeltään Idle. Sillä kirjoitetaan vaadittavat koodirivit ja ne tallennetaan tiedostoon, jonka loppupääte on `.py`. Tämä tiedosto asetetaan käynnistyväksi Raspberry Pi:n mukana, minkä jälkeen se odottaa syötettä painikkeelta. Kuvassa 10 on esimerkki työssä käytetystä `wvdialconnect.py`-tiedostosta, joka tallennetaan `/home/pi/-`kansioon. Siinä asetetaan tiedosto kuuntelemaan muutosta GPIO-portissa 18. Kun portin 18 tila muuttuu, tiedosto suorittaa komennon `wvdial 3gconnect`, joka avaa 3G-yhteyden. Vastaavia koodirivejä käytetään pinneissä 23 ja 25. Komento `sudo shutdown -h now` sammuttaa laitteen ja komento `pobd_logger -v /dev/ttyUSB0 38400` käynnistää OBD-tietojen tallennuksen. Jotta painikkeet olisivat käytössä heti käynnistyksen jälkeen, täytyy `.py`-tiedostot lisätä käynnistettävien ohjelmien listalle. Tämä tapahtuu muokkaamalla `rc.local`-tiedostoa komennolla `sudo nano /etc/rc.local`. Tämän tiedoston loppuun ennen riviä `exit 0`, kirjoitetaan kaikki kolme aikaisemmin luotua `.py`-tiedostoa muodossa `python /home/pi/tiedoston_nimi.py &`. Nyt kun laite käynnistyy, tiedostot jäävät odottamaan komentoja painikkeilta.



Kuva 9. Kytkentäkaavio laitteen painikkeista ja LED-valosta.

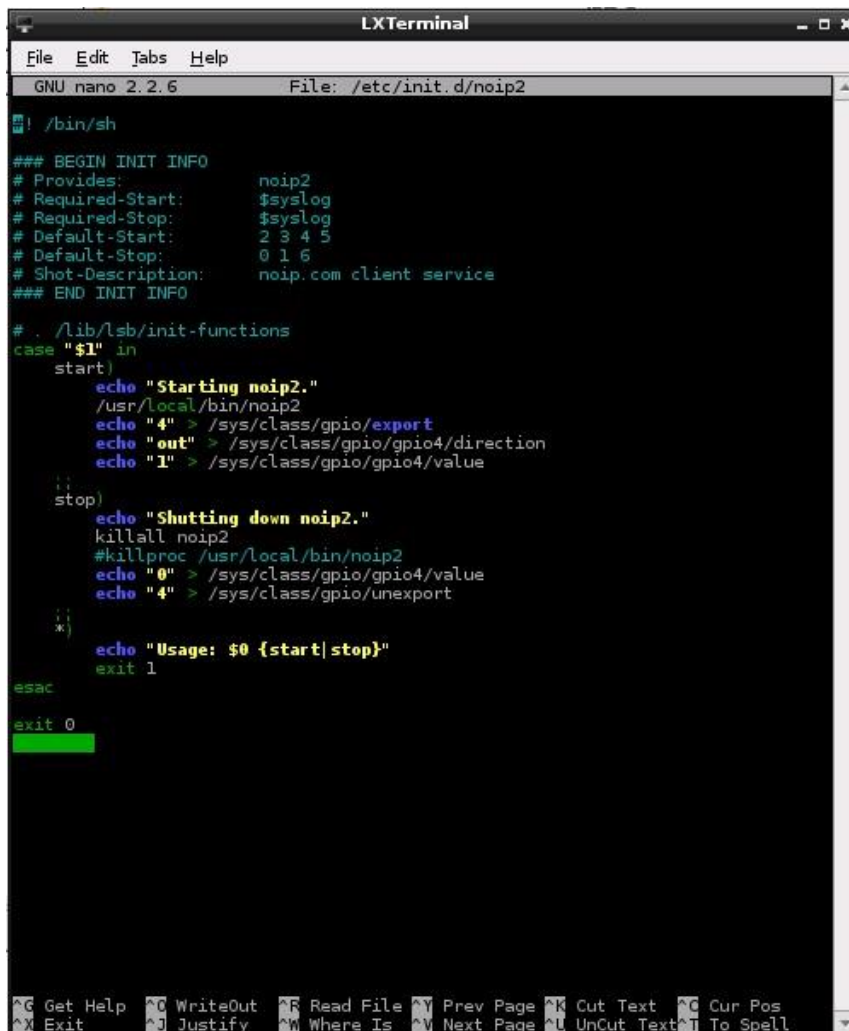
```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time
import os
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.IN)
while True:
    if (GPIO.input(18) == False):
        os.system("wvdial 3gconnect")
        time.sleep(0.4)
```

Kuva 10. Esimerkki työssä käytetystä wvdialconnect.py-tiedostosta.

LED-valo asetetaan syttyväksi samaan aikaan kun No-IP-palvelu käynnistyy Raspberry Pi:n mukana. Kuvassa 11 on esimerkki työssä käytetystä tiedostosta, joka on liitetty käynnistykseen. Tiedosto luodaan komennolla `sudo nano /etc/init.d/noip2`. Tiedosto suorittaa ensin No-IP-palvelun ja sen jälkeen asettaa GPIO-pinnin tilaan "1", jolloin LED-valo

syttyy. Kun tiedosto on luotu, se on muutettava suoritettavaan muotoon komennolla `sudo chmod 755 /etc/init.d/noip2`. Jotta tiedosto käynnistyisi automaattisesti laitteen mukana, täytyy noip2-tiedosto lisätä update-rc.d-tiedostoon. Tämä tapahtuu komennolla `sudo update-rc.d noip2 defaults`. Nyt kun laite käynnistetään, se suorittaa tiedoston noip2, joka käynnistää No-IP-palvelun ja sytyttää LED-valon merkiksi siitä, että laite on käynnistynyt ja on valmis vastaanottamaan komentoja painikkeilta.

Nyt kaikki laitteen toteutukseen vaadittavat kohdat on suoritettu ja laite on valmis käytettäväksi. Suorittamalla komennon `sudo raspi-config` avataan aikaisemmin Raspbianin asennuksen jälkeen käytetty kokoonpanoikkuna. Valikon kohdasta 10 voidaan asettaa Raspberry Pi käynnistymään työpöytäkymän sijaan komentoriviin. Etäyhteys käynnistyy kuitenkin aina työpöytäkymään.



```

LXTerminal
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/init.d/noip2

#!/bin/sh

### BEGIN INIT INFO
# Provides:          noip2
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: noip.com client service
### END INIT INFO

# . /lib/lsb/init-functions
case "$1" in
  start)
    echo "Starting noip2."
    /usr/local/bin/noip2
    echo "4" > /sys/class/gpio/export
    echo "out" > /sys/class/gpio/gpio4/direction
    echo "1" > /sys/class/gpio/gpio4/value
  ;;
  stop)
    echo "Shutting down noip2."
    killall noip2
    #killproc /usr/local/bin/noip2
    echo "0" > /sys/class/gpio/gpio4/value
    echo "4" > /sys/class/gpio/unexport
  ;;
  *)
    echo "Usage: $0 {start|stop}"
    exit 1
esac

exit 0

```

Kuva 11. Esimerkki työssä käytetystä noip2-tiedostosta.

4 Yhteenveto

Työn tavoitteena oli toteuttaa laite, joka mahdollistaa auton vikakoodien ja muuttujien lukemisen vikatilanteiden sattuessa auton ollessa muualla kuin korjaamolla. Laitteen oli tarkoitus olla ominaisuuksiltaan sellainen, että sen käyttö on yksinkertaista ja että se soveltuu asennettavaksi autoon lisävarusteeksi. Laitteen toteutus onnistui hyvin ja se saatiin vastaamaan haluttuja ominaisuuksia. Ongelmia toteutuksessa aiheutti Huaweiin E367-3G-modeemi, joka ei ajoittain toiminut Raspberry Pi:n USB-porttien ongelmien takia. Tämä ongelma korjautui vaihtamalla 3G-modeemi paremmin yhteensopivaan E220-malliin.

Vaikka laite on tarkoitettu asennettavaksi kaikkiin ajoneuvoihin, niin ELM327:n ominaisuuksista ja autovalmistajien erilaisista OBD-tuista johtuen se ei kuitenkaan ole mahdollista. Tämä johtuu siitä, että OBD-tukea on ryhdytty vaatimaan eri puolilla maailmaa eri aikaan. Suomessa OBD-järjestelmä vaaditaan autoissa, jotka on otettu käyttöön ennen 1.1.2001 [9]. Tätä ennen käyttöön otettujen ajoneuvojen OBD-tuki on vaihdellut merkin ja mallin mukaan. Vaikka autossa olisikin OBD2-pistoke, niin se ei tarkoita, että sen käyttämä protokolla kuuluisi yleisesti ELM327:n tukemiin protokolliin.

Laitteessa on vielä paljon erilaisia jatkokehitysmahdollisuuksia, kuten esimerkiksi laitteen liittäminen auton multimediajärjestelmään. Tämä mahdollistaisi muun muassa laitteessa olevan 3G-modeemin laajemman käytön erilaisissa musiikki-, video- ja karttapalveluissa. Jos laitteen ominaisuudet yhdistettäisiin jo autossa valmiina olevaan näyttöpäätteeseen, se toisi auton parametrit ja vikakoodit helpommin kuluttajalle saataviksi. Tämä mahdollistaisi sen, että auton vikavalon syttyessä auton käyttäjä saisi halutessaan vikavalon syttymisen aiheuttavan vikakoodin selville.

Lähteet

- 1 V3: Fleet GPS Tracking/Vehicle Diagnostics System. 2014. Verkkodokumentti. Zonar Systems. <<http://www.zonarsystems.com/products/v3-gps-fleet-tracking/>>. Luettu 20.4.2014.
- 2 Raspberry Pi - A Credit-Card Sized ARM Computer - Yours For Only \$25. 2012. Verkkodokumentti. Bookers Tim. <<http://www.makenseof.com/tag/raspberry-pi-creditcard-sized-arm-computer-25/>>. Luettu 20.4.2014.
- 3 Matkapuhelinverkko. 2014. Verkkodokumentti. STUK. <http://www.stuk.fi/sateilyn-hyodyntaminen/matkapuhelimet-ja-tukiase-mat/fi_FI/matkapuhelinverkko/>. Luettu 20.4.2014.
- 4 ELM327 ODB to RS232 Interpreter. 2005. Verkkodokumentti. Elm Electronics Inc. <<http://elmelectronics.com/DSheets/ELM327DS.pdf>>. Luettu 21.4.2014.
- 5 What does No-IP do? 2014. Verkkodokumentti. No-IP.com. <<http://www.noip.com/support/knowledgebase/what-does-no-ip-do/>>. Luettu 21.4.2014.
- 6 How to make a Raspberry Pi disk image to SD card with win32diskimager. 2014. Verkkodokumentti. RasPI.TV. <<http://raspi.tv/2012/how-to-make-a-raspberry-pi-disk-image-to-sd-card-with-win32diskimager>>. Luettu 21.4.2014.
- 7 openOBD: OBD-II Scan Tool. 2013. Verkkodokumentti. Simesb. <<http://sourceforge.net/projects/openobd/>>. Luettu 21.4.2014.
- 8 Perl OBD-II Logger. 2014. Verkkodokumentti. Kd6ekq. <<http://sourceforge.net/projects/pobd-logger/>>. Luettu 21.4.2014.
- 9 Ottomoottorikäyttöisten ajoneuvojen pakokaasupäästöjen tarkastus. 2011. Verkkodokumentti. Trafi. <http://www.trafi.fi/file-bank/a/1325147177/3fb9d1c954c8aab89c3d40b8cce5ca26/4756-Ottomoottorikayttoisten_pakokaasupaastojen_tarkastus.pdf>. Luettu 21.4.2014.

Valmis laite ilman koteloä

