Janne Ylänne

Ajoneuvojen kulunvalvonta ja tiedonkeruu

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikka
Insinöörityö
11.2.2016

Tekijä(t)	Janne Ylänne
Otsikko	Ajoneuvojen kulunvalvonta ja tiedonkeruu
Sivumäärä	28 sivua + 8 liitettä
Aika	11.2.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja(t)	Lehtori Marko Uusitalo Projektipäällikkö Ville Eskelinen

Ajoneuvojen testaus- ja kehitystyössä on toisinaan suuria käytännön ongelmia vikojen ja häiriötilanteiden tallentamisessa etenkin vikojen, jotka ilmenevät harvoin ja ennalta arvaamatta. Tiedonkeruujärjestelmä, joka olisi käytön ajan kytkettynä, ja joka keräisi koko ajalta kaiken tiedon talteen, tarvitsisi aivan liikaa tallennustilaa. Tähän sopivat laitteistot ovat edelleen myös hyvin arvokkaita.

Työssä tavoitteena oli luoda kohtuullisen hintainen, mutta kuitenkin monipuolinen laite tiedonkeruuta varten, joka tallentaisi tarpeellisia tietoja koko ajan. Vian tai häiriön ilmetessä olisi määritelty aikajakso, jonka käyttäjä kykenisi tallentamaan. Tallennettu tieto olisi tämän jälkeen tarkoitus lähettää yhteiseen tallennustilaan myöhempää tarkastelua varten. Samalla oli tarkoitus tallentaa mobiililaitteiden käytöstä muitakin tietoja, kuten esimerkiksi käyttäjä, ajankohta ja sijainti. Toisena toiveena oli luoda tietoturvallinen ja luotettava tietoliikenneyhteys, jolla laitteiden käyttöjärjestelmä olisi mahdollista päivittää verkon yli.

Työn lopputuloksena syntyi laite, joka toimi rajallisesti toiveiden mukaan, mutta kehitystä toivottavasti jatketaan muiden projektien yhteydessä vielä pitkään.
Työn tilaajana on Metropolia Oy, jonka projektien yhteydessä laitetta on kehitetty ja testattu.

Avainsanat	CANtact, Raspberry Pi, CANopen

Author(s)	Janne Ylänne
Title	Access control and diagnostics of a mobile vehicles
Number of Pages	28 pages + 8 appendices
Date	11 Feb 2016
Degree	Bachelor of Engineering
Degree Programme	Communication Networks and Applications
Specialisation option	Networks
Instructor(s)	Marko Uusitalo, Senior Lecturer Ville Eskelinen, Project Manager

Mobile vehicle testing and development is sometimes difficult because of capture of faults and warnings in the public roads and other remote locations. Most difficult to capture are faults that occur less often and without a warning. Data logging system that could store all the data all times and would be available at every moment a device runs, would take just too much of a storage space. The devices that are available at the moment are also very expensive.

In this thesis the main goal was to create a reasonable priced but still a device that can do different kind of tasks. In the moment of fault or malfunctions, the need is to have a possibility to record the moment before and after. The recorded data should be send to a storage where it can be analyzed with a proper program. At the same time there could be a chance to record other kind of things about the usage of a vehicle. The location, time and the user of the device at the moment of malfunction would help to process the data. Second hope in the thesis was to create a remote update possibility from a website or other easy to access location.

The end product was a device and it was proven to work. The time limitation caused that most of the testing to be done later.

The order for this thesis came from Metropolia Oy, and the device was created as part of projects of the company.

Keywords	CANtact, Raspberry Pi, CANopen

Sisällys

Liite 8. CANopen-viestitunnukset ja esimerkki tiedonsiirrosta

Lyhenteet

3G	Third Generation / kolmannen sukupolven teknologia.
4G(DC)	*Dual Carrier* käyttää kahta 3G verkkoa rinnakkain.
4G(LTE)	4. sukupolven matkapuhelinteknologia (*Long Term Evolution*).
ADSL	*Assymetric Digital Subscriper Line*, epäsymmetrinen tilaajalinja.
AT käskyt	Modeemien käyttöön kehitetty alkeellinen käskykanta.
CAN	*Controller Area Network*, automaatioväylä.
CANard	Python ohjelmointikielen CAN-väylän tulkintaa helpottava kirjasto.
CANlow / CANhigh	CAN-väylän jännite-ero (1,5V ja 3,5V CANopen).
CANopen	Avoimen koodin CAN verkkoprotokolla.
CANtact	CAN-väylä sovitin.
CPU	*Central Processing Unit*, Tietokoneen prosessori.
DB9-liitin	Sarjaportin 9-pinninen liitin.
EMCY	*Emergency Object*, CANopen-viestityyppi.
GUI	*Graphic User Interface*, Graafinen käyttöliittymä.
GPIO	*General Purpose Input/Output* yleisnimitys ohjattaville liitoksille.

GSM *Groupe Special Mobile* / Global System for Mobile Communications.

HSPA *High Speed Packet Access* on Kokoelma viestintäprotokollia.

I2C *I two C tai I-I-C* sarjamuotoinen kaksisuuntainen väylä.

Interface Kahden tiedonsiirtomuodon yhdyspiste.

MAC *Media Access Control*, yksilöity laitetunnus ethernet verkossa.

MIMO *Multiple-Input Multiple-Output*, usean antennin tiedonsiirto.

NAT *Network Address Translation*, Osoitteenmuutos sisäverkkoon.

NMT *Network Management*, CANopen-viestityyppi

Node/solmu Verkon liitospiste.

OSI *Open Systems Interconnection Reference Model*, OSI-malli.

PLC *Programmable Logic Controller* Ohjelmoitava Logiikka Ohjain.

RAM *Random Access Memory*, käyttömuisti.

Raspbian Linux Debianiin perustuva yksinkertaistettu käyttöjärjestelmä.

RDT / LXDE X11 *Remote desktop*, kevyt X11-työpöytäympäristö.

RPDO *Receive Process Data Object*, CANopen-viestityyppi.

RS232 Sarjaportti, esim. Windows koneissa COM1 ja COM2.

SDO *Service Data Object*, CANopen-viestityyppi.

SPI	*Serial Peripheral Interface*, synkronoitu tiedonsiirto väyläprotokolla.
SSH	*Secure Shell*, Tiedonsiirron suojausprotokolla.
SYNC	*Syncronization Object*, CANopen-viestityyppi.
TPDO	*Transmit Process Data Object*, CANopen-viestityyppi.
TIME	*Time Stamp Object*, CANopen-viestityyppi.
USB	*Universal Serial Bus*, yleistyyppinen liitin.
WLAN	*Wireless Area Local Area Network*, langaton sisäverkko.

1 Johdanto

Insinöörityön aiheena oli kehittää laite, jolla seurataan Metropolian projekteissa työn alla olevia ajoneuvoja. Tulevaisuudessa olisi myös toivottavaa saada yhteys työn alla oleviin laitteisiin, jotta ohjelmointi ja tarkastelu olisi mahdollista tehdä toimistosta käsin. Samalla olisi tarkoitus mahdollistaa vikatilanteiden tallennus nykyistä paremmin ja yksinkertaisemmin. Viat ja häiriöt tapahtuvat usein sellaisissa tilanteissa, joissa vian selvittäminen on jälkikäteen hankalaa, koska itse vikatilannetta ei saada tallennettua. Mainitun kaltaisten vikojen etsiminen on erittäin työlästä. Tämän vuoksi olisi hyvä olla sellainen puskurimuisti, jonka sisällön voisi tallentaa aina vian tai häiriön ilmaantuessa. Suurimpana haasteena työssäni oli saada aikaiseksi käyttöjärjestelmän päivitysmahdollisuus. Päivitys olisi hyvä saada suoritettavaksi verkon yli tietoturvallisesti, tähän tarkoitetun nettisivuston kautta.

Työn tilaajana on Metropolia Ammattikorkeakoulu Oy. Metropolia Ammattikorkeakoululla on lukuisia projekteja, joissa on syntynyt vuosien saatossa suuri määrä toinen toistaan hienompia saavutuksia. Aikojen saatossa on kellareiden syvyyksissä ja pienissä työpajoissa valmistettu monipuolisia ja ainutlaatuisia laitteita opiskelijoiden toimesta. Projektien parissa opiskelijat ovat voineet saattaa opintojaan loppuun käytännön aiheiden parissa. Huomioitavaa on myös, että projekteissa luodaan suhteita ja opitaan oman alan ulkopuolisia asioita auttamalla muita ja seuraamalla toisten työskentelyä.

2 Tekniikka

2.1 Laitteen käyttöjärjestelmät

2.1.1 Raspbian

Raspbian on erityisesti Raspberry Pi -tietokoneeseen tarkoitettu unix-pohjainen käyttöjärjestelmä. Pääosin se perustuu Debian käyttöjärjestelmään, joka taas on eräs Linuxin jakelupaketti. Debian on vuosien aikana päivittynyt ja tällä hetkellä sen uusin päivitys on Jessie, koska päivitykset on nimetty Toy Story -elokuvien hahmojen mukaan. Jessie tuli saatavaksi 25.4 2015. Raspbian on tästä kevennetty versio eikä se siis tue kaikkia Debianin ominaisuuksia. Tämä mahdollistaa käytön myös pienempi tehoisissa laitteissa kuten Raspberry Pi. Tietoa käytännön ongelmiin on hyvin tarjolla, kuten kaikista Linuxin käyttöjärjestelmistä. Lähes jokaiseen ongelmaan löytyy useita ratkaisuja keskustelufoorumeilta. GPIO-pinnien, toiminta on myös hyvin yleistä, ja niiden erilaiset käyttötarkoitukset vaativat hiukan opiskelua. Suurin ongelma on pinnien huono jännitteen tai virran sietokyky, jolloin laite on helppo saada hajoamaan vääränlaisella kytkennällä tai kokoonpanolla.

2.1.2 Codesys

Codesys on Saksalaisen 3S-Smart Software Solutions –yhtiön vuonna 1994 esittelemä ohjelmoitavan logiikan konfigurointiin suunniteltu ohjelmisto. Se mahdollistaa teollisuuden automaatiojärjestelmien ohjelmoinnin käyttäen sekä kaikkia viittä IEC 6113-1 –standardissa määriteltyä tekstipohjaista ohjelmointitapaa, että standardin ulkopuolista graafista editoria Continuos Function Chart. Väyläpohjaiset järjestelmät yleistyvät teollisuuden ulkopuolisissa sovelluksissa, jolloin ohjelma näyttäisi saavuttavan lisää suosiota. Esimerkiksi ABB:n ohjelmointiympäristöön on implementoitu rajapinta ohjelmalle. Codesys on monipuolinen työkalu, jolla voidaan luoda toimivia ja muokattavissa olevia järjestelmiä. Tällä hetkellä

uutuutena on myös ilmainen ja kaikille tarjolla oleva sovellus Raspberry Pin käyttöön osana Codesys-ympäristöä. Käyttöjärjestelmän asennus on yhtä helppoa kuin muidenkin Raspberry Pi -käyttöjärjestelmien. Asennetaan image SD-kortille ja kortti paikalleen. Tämän jälkeen laite on valmis käyttöön ja ohjelmoitavissa ohjelmalla.

2.1.3 CAN-väylä

Vuonna 1986 Robert Bosch GmbH esitteli SAE-kongressissa Sarjaväylätekniikkaan perustuvan 8-bittisen Controller Area Network (CAN) -järjestelmän. 1987 Intel toimitti ensimmäisen CAN-sirun, joka oli malliltaan 82526. Tämän jälkeen nopeasti Philips Semiconductors esitteli 82C200 Can -ohjaimen. Nykyään yli 20 valmistajaa tekee kyseisen protokollan mukaisia laitteita. Vuonna 1999 lähes 60 miljoonaa ohjainta valmistettiin ja vuonna 2000 myytiin yli 100 miljoonaa laitetta, jotka toimivat CAN-väylässä. Vaikka alkuperäinen tarkoitus oli luoda ajoneuvokäyttöön sopiva järjestelmä, yleistyivät CAN-väylät nopeasti myös teollisuuden ja muun automatiikan sovelluksissa. 1990 alettiin selvittää standardin luomista, ja vuonna 1992 Holger Zeltwanger toi käyttäjät ja valmistajat yhteen perustaakseen CiA eli CAN in Automation järjestön. Tämä johti standardin luomiseen ja myöhemmin CANopenin syntymiseen.[2.]

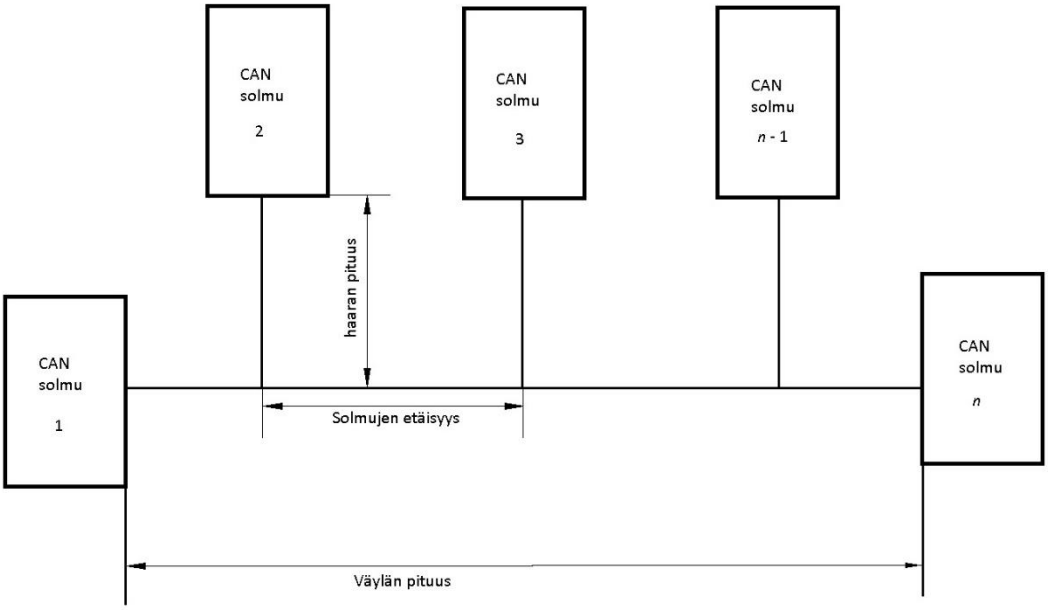
CAN-väylä ja muut väyläpohjaiset protokollat perustuvat jännite-eroihin siten, että tieto siirretään bitteinä eli jonona ykkösiä ja nollia. Vaikka alkuperäinen CAN-väylä toimisi kahdella johdolla, niin lisätään nykyään maajohdin järjestelmään parantamaan toimintavarmuutta. Silloin väylässä on kolme johdinta, joista yksi on CANHigh- toinen CANLow-johdin, ja näiden lisäksi tuo mainittu maajohdin josta saadaan nollapotentiaali järjestelmälle. Jännite-ero johdinten välillä loogisessa tilassa (1) on kaksi volttia, jolloin johtimien jännitteet ovat 1,5 ja 3,5 volttia. Tätä tilaa kutsutaan joissain yhteyksissä myös Dominanttitilaksi. Looginen tila (0) tai resessiivinen tila ilmoitetaan kaapelissa niin, että molemmissa johtimissa on 2,5

volttia, jolloin jännite-eroa ei ole lainkaan vaan johtimissa on sama jännite. Jännite-eroon perustuva tiedonsiirto sietää hyvin häiriöitä ja on muutenkin todella luotettavaksi todettu. Tiedonsiirtonopeuksia rajoittaa väylän pituus, joka seuraa CAN-järjestelmässä seuraavaa kaaviota.

Taulukko 1. CAN-väylän etäisyydet

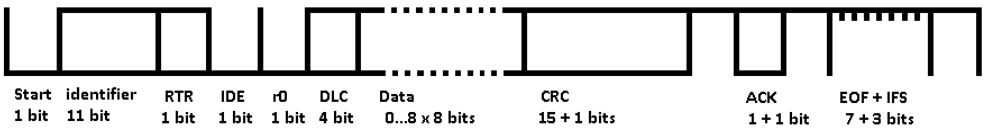
Väylän nopeus	Väylän pituus	Sivuhaaran pituus	Solmun etäisyys
1 Megabittiä/sekunti	40 metriä	0,3 metriä	40 metriä
500 kilobittiä/sekunti	100 metriä	0,3 metriä	100 metriä
100 kilobittiä/sekunti	500 metriä	0,3 metriä	500 metriä
50 kilobittiä/sekunti	1000 metriä	0,3 metriä	1000 metriä

Taulukosta selviää että CAN-tyyppisessä väylässä suuremmat nopeudet eivät mahdollista kovin laajaa verkkoa, mutta esimerkiksi ajoneuvoissa voidaan käyttää 1Mb/s nopeutta.



Kuva 1. CAN-väylän rakenne. Kuvasta selviää CAN-verkon perusrakenne ja termien selitykset.

CAN-väyläprotokolla määrittelee OSI-mallin ensimmäistä kahta tasoa eli fyysistä ja siirtoa koskevia asioita. Jokainen CAN –viesti alkaa start of frame bitillä (SOF), jota seuraa ID. Viestien törmätessä väylän sisällä pienimmällä ID-tunnuksella on etu, eli pienempi tunnus menee läpi. Tämän jälkeen suuremman ID-tunnuksen viestin lähettänyt laite odottaa hetken, ja koittaa lähettää uudelleen oman viestinsä.



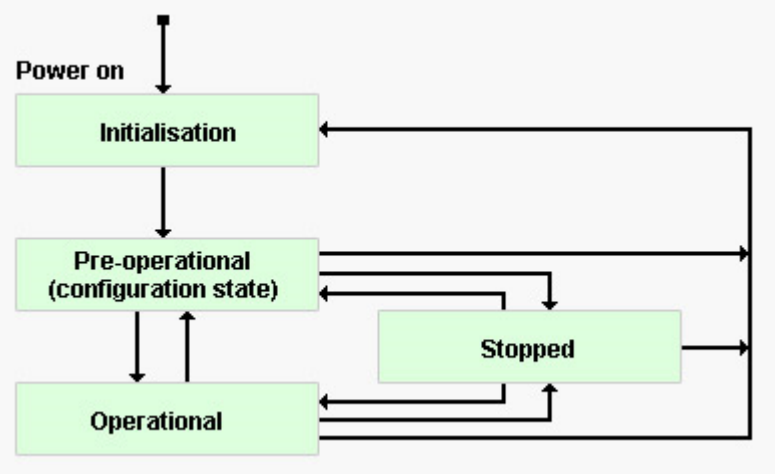
Kuva 2. CAN-frame rakenne 11-bittisessä standard form CAN –protokollassa. [6.]

2.1.4 CANopen

CANopen-väylässä viestien rakenne on samanlainen kuin itse CAN-protokollassa, mutta CANopen on käytännössä ylemmän tason protokolla, joka OSI-mallin mukaisesti ilmaistuna käyttää tasoja 3-7. On myös mahdollista lisätä CAN-väylään ISO 11898-2 tai uudemman standardin laitteita, jotka toimivat CANopen protokollan mukaisesti. Nämä laitteet soveltuvat yhteen ja se lisää molempien standardien käytettävyyttä erilaisissa sovelluksissa.

CANopen toimii 11-bittisellä tunnisteosalla, joista ensimmäiset 7-bittiä muodostavat node-id:n, josta johtuen laitteiden lukumäärä on rajattu 127 kappaleeseen verkkoa kohti. PDO-viestejä on pääluokassa kahta eri tyyppiä jotka rakenteeltaan ja topologialtaan ovat samankaltaiset. Ensimmäinen eli TPDO-viesti (Transmit Process Data Object) lähettää kaikkien laitteiden vastaanotettavaksi tietoa. RPDO-viestin (Receive Process Data Object) on taas vastaanotettava viestityyppi. Käytännössä TPDO on laitteelta lähtevä ja RPDO laitteelle saapuva viesti.

NMT eli Network Management -tyypin viesteillä laitteet ilmoittavat käynnistyttyään olevansa valmiina, jonka jälkeen Masteriksi määritelty yksikkö asettaa väylässä oleville laitteille erilaisia toimintatiloja. CANopen-laitteet voidaan asettaa kolmeen erilaiseen tilaan jotka ovat operational, stop ja pre-operational. Käynnistymisen eli initialisation-tilan jälkeen laite menee automaattisesti pre-operational-tilaan ja ilmoittaa olevansa valmis toimintaan. NMT-viesteillä myös seurataan verkon tilaa, eli että kaikki laitteet ovat toiminnassa ja toivotussa tilassa.



Kuva 3. CANopen laitteiden tilakone, eli kaaviosta selviää NMT-viestien toiminta ja laitteiden erilaisten tilojen käyttömallli. [7].

SDO (Service Data Object) viesteillä voidaan määritellä laitteiden asetuksia, kuten väylänopeuksia ja ID- RDPO- TPDO- asetuksia. CANopen-laitteiissa on lisäksi sisäinen muistikenttä, johon on mahdollista kirjoittaa tai lukea suoraan SDO-käskyillä. SDO-viestien käytössä on hyvä huomata että niiden käyttö on mahdollista vain pre-operational tilassa.

Emergency Object (EMCY) protocol on vian ilmoitus protokolla, joka antaa kaikille laitteille mahdollisuuden ilmoittaa ongelmista. Protokolla lähettää vain yhden ilmoituksen jokaista vikaa kohti. Koska viestin prioriteetti on niin korkea, EMCY tarjoaa jokaiselle laitteelle mahdollisuuden korkean prioriteetin viesteihin, riippumatta laitteeseen asetetusta node-id:stä. Tämä toimii erittäin hyvin ja nopeasti, koska laitteen id:stä huolimatta kyseisen viestin prioriteetti on niin korkea, että se menee varmasti läpi.

Törmäysten välttämiseksi on erittäin tärkeää synkronoida viestiliikenne. Kunkin väylässä toimivan laitteen kommunikointiasetuksissa määritellään TPDO viestien lähetystyyppi ja -sykli. Laite voidaan ohjelmoida lähettämään viestinsä joko asynkronisesti määrätyin väliajoin tai synkronisesti tietoa pyydettäessä. CA-

Nopen järjestelmässä synkronoitua viestiliikennettä ohjaa Synchronization Object (SYNC) -protokolla. Kukin CANopen laite voidaan asettaa lähettämään tietonsa tietyllä SYNC-viestin monikerralla. Monikerrat ovat rajoitettu arvoihin 1 – 240. Laite voi siis vastata esimerkiksi jokaiseen synkronointiviestiin tai joka kymmenenteen, sadanteen ja niin edelleen. SYNC-viesti sisältää myös toisen tärkeän ominaisuuden, joka liittyy tiedon tallentamiseen. Tietoa keräävä ja synkroniseen kommunikointitilaan asetettu CANopen-laite tallentaa sisääntulonsa jokaisella SYNC-viestillä laitteelle asetetusta vastausmonikerrasta huolimatta. SYNC-protokollalla voidaan täten määrittää laitteiden tiedonkeruusykli.

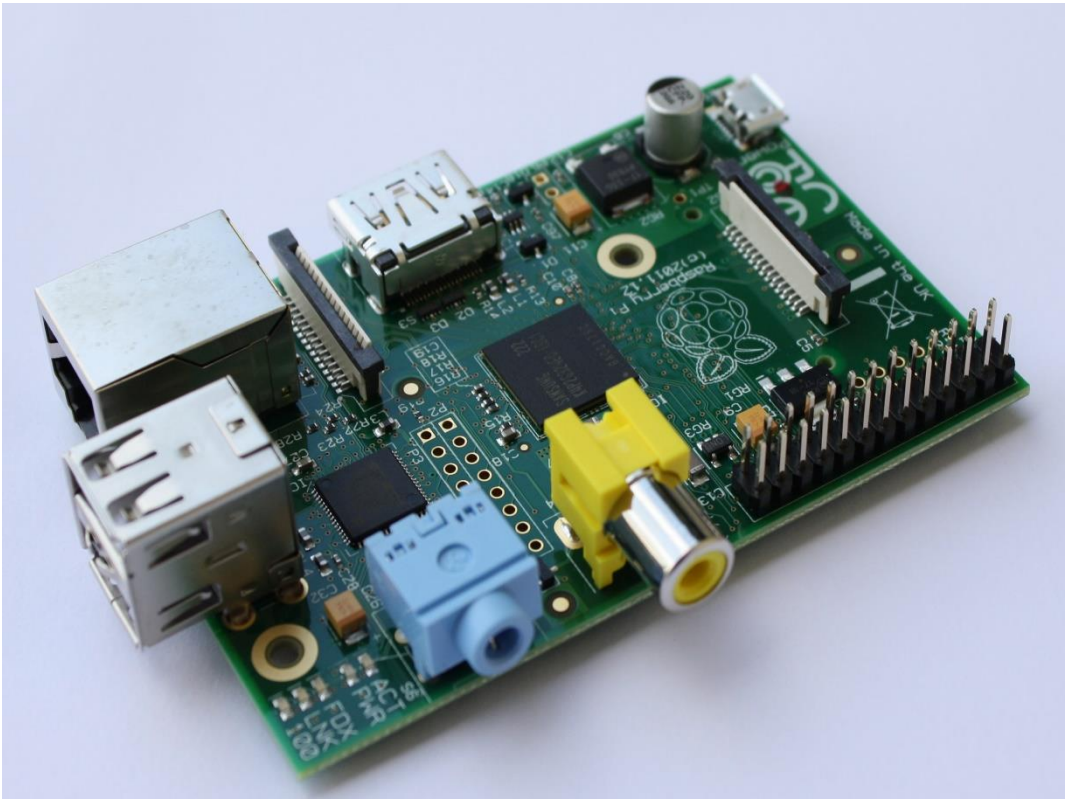
Time Stamp Object (TIME) protocol määrittää ajan hetken mittaamalla erotuksen millisekunneissa keskiyöhön ja päivien määrän 1.1.1984 lähtien. Viesti sisältää 6 tavua eli 48 bittiä ja sillä voidaan ajoittaa pidempiä aikoja vaativia tehtäviä ja muutenkin saadaan suurempi joukko toimimaan yhteisessä ajan hetkessä.

CANopen-laitteissa on standardin mukaiset valmiiksi asetetut muistilohkot, joista Master-laite voi kirjoittaa tietoa tai lukea sitä pienellä viiveellä ja yksinkertaisella ohjelmalla. Tämän mahdollistaa CANopenin object dictionary -kirjasto, joka on 16-bittinen indeksi. Indeksi määritellään nelinumeroisella heksadesimaalisella arvolla 1000h:sta 9FFFh:hon. Indeksissä on erilaisiin tarkoituksiin jaettuja osia, joista ensimmäinen osa 1000h - 1FFFh määrittää laitteiden viestintää. Toinen osa on ohjelmien vaatimia arvoja varten, ja käsittää lohkot 2000h - 9FFFh. Toisessa osassa on vielä eritelty 2000h - 5FFFh alue vapaaseen käyttöön ohjelmia ja käyttäjän toimintaa varten, ja lohkot 6000h - 9FFFh on standardisoiduille CANopen lohkoille varattuna.

2.2 Laitteisto

Laitteistoksi valitsin Raspberry Pin lähinnä sen alhaisen kustannustason vuoksi. Oletettavasti jokaisella laitetta tulevaisuudessa käyttävällä on jonkinlaiset tietotekniset taidot, joten tämän vuoksi laitteen ei tarvitse olla aivan yksinkertaisin

mahdollinen. Koitin saada aikaiseksi laitteen, jonka käyttö olisi sovitettavissa monenlaisiin sovelluksiin mahdollisimman pienin muutoksin, jotta se myös sopisi uusiin opiskelijaprojekteihin tulevaisuudessa. Toinen syy valintaan on kyseisen tietokoneen pieni virrankulutus. Projektissa alkuun käytössä ollut 3G/GPS-Moduuli olisi myös ollut ihanteellinen pienen sähköntarpeensa vuoksi. 4G-reititin jolla lopullinen työ tehtiin, ei merkittävästi ollut tehonkulutukseltaan suurempi. Laite on myös helppo muuttaa monenlaiseen lisäkäyttöön ja automatisoida toimimaan lähes itsestään. Huomasin työn loppuvaiheessa, että äskettäin oli ilmestynyt päivitys Codesys-käyttöjärjestelmään, joka saattaa olla ratkaisu tiedonkeruun ja tietojen hallinnan kanssa tulevaisuudessa projektien ajoneuvoista.



Kuva 4. Raspberry Pi Model B versio, joka on tunnistettavissa 26 GPIO pinnistä ja kahdesta USB-portista.

2.2.1 Raspberry Pi

Vuonna 2006 Cambridgen yliopistossa tultiin siihen tulokseen, että opiskelijat joutuvat ylittämään liian suuren kuilun aloittaessaan harjoitukset tietokoneiden parissa. Aikaisemmat sukupolvet aloittivat ohjelmoinnin yksinkertaisilla tietokoneilla. Ensimmäiset harjoitukset olivat todella helppoja ja myös mahdollisia ymmärtää täydellisesti. Tämän vuoksi Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft [1] päättivät tehdä yksinkertaisen, edullisen ja helposti lähestyttävän, mutta kuitenkin riittävän laskentatehon omaavan tietokoneen. Pyrittiin tekemään laite, jolla jokainen tietotekniikkaa opiskeleva tai harrastava pääsee ohjelmoinnissa alkuun, ja näin käsittää pohjatason ohjelmien rakenteen ja toiminnan. Ryhmään liittyi vuonna 2008 Pete Lomas ja David Braben, minkä jälkeen meni vielä kolme vuotta ennen massatuotannon aloitusta. Laitteen suosio oli ilmestyessään valtava. Kahden ensimmäisen vuoden myynti oli yli kaksi miljoonaa Model B -mallin yksikköä.

Hankintahinta pelkälle peruslaitteelle oli aluksi kahdenkymmenen euron luokkaa. Nykyään paranneltu tehokkaampi versio siihen liittyvine tarvikkeineen maksaa noin viisikymmentä euroa. Kyseisen laitteen ympärille on tehty paljon erilaisia projekteja niin oppilaitoksissa kuin harrastajienkin parissa. Lähitulevaisuudessa projektien määrä tulee vielä varmasti kasvamaan. Viimeisessä päivityksessä laitteen prosessoritehoa nostettiin jälleen, mikä mahdollistaa taas monimutkaisempia tehtäviä. Laitteen asennus on hyvin yksinkertaista ja helppoa, myös ilman aiempaa kokemusta käyttöjärjestelmän asennuksesta. Käyn insinöörityössä perusasennuksen läpi ja lisäksi sellaiset toimenpiteet, jotka ovat välttämättömiä laitteen toiminnan kannalta tämän kaltaisessa käytössä. On myös mainittava tässä yhteydessä Raspberry Pi Zero, joka juuri julkaistiin. Zero on lähes vastaava laite kuin alkuperäinen Pi, mutta maksaa tällä hetkellä viisi euroa ja on todella pienikokoinen sisältäen kuitenkin nykyisen 40 GPIO-pinnin mahdollisuudet. Tämän laitteen tulevaisuuden mahdollisuudet ovat todella vaikeita ennustaa.

2.2.2 3G / GPS -moduuli

Hankin työtäni varten Cooking-hacks-nimiseltä yritykseltä etäyhteyttä ja tiedonsiirtoa varten Arduino-kehityslaudalle tarkoitetun 3G/GPS-moduulin, jonka yhdistin tarkoitukseen valmistetulla välisovittimella Raspberry Pi -pienoistietokoneeseen. Pitkään kuvittelin, että voisin tällä laitteella muodostaa jatkuvan yhteyden 3G-verkon yli, mutta se osoittautui mahdottomaksi. Yhteyttä varten moduuli olisi yhdistettävä Raspberry Piihin usb-liittimen kautta eikä tämä muista syistä ollut järkevää, joten vaihdoin laitteen 4G-teknologiaa tukevaan WLAN-reitittimeen. Tiedonsiirto langattomassa verkossa on tällä hetkellä saavuttanut suuren nopeuden. Olemme osittain siirtyneet kolmannesta sukupolvesta neljänteen sukupolveen matkapuhelinverkoissa. Ensimmäinen sukupolvi oli perinteinen analoginen verkko, jossa toimivat mm. NMT-puhelimet. Paras tiedonsiirto tietoliikenteessä saavutettiin ADSL-tyyppisillä modeemeilla. Nämä kykenivät loppuvaiheessa jopa yli 20 megabitin sekuntinopeuteen verkosta ladattaessa. Suurin ongelma oli todella suuri epäsymmetrisuus latausnopeuksissa verrattuna lähetysnopeuteen. Lähetysnopeudet olivat kymmenyksen luokkaa lataukseen verrattuna. Internetin käyttöön ADSL sopi hyvin mutta tiedonsiirtoon ja jakamiseen huonosti. Toinen sukupolvi oli sitten ensimmäinen digitaalinen verkko, jonka suurimmaksi menestykseksi osoittautui GSM. Tätä standardia käytetään eniten verkossa kommunikointiin maailmassa vielä nykyäänkin. Suomessa myydään tällä hetkellä jo 4G-tekniikkana kahta erilaista 3G-jatkosovellusta, joten 4G-termi on hiukan harhaanjohtava. 4G LTE perustuu MIMO-tekniikkaan, jossa erityisen algoritmin avulla tieto hajautetaan ja lähetetään usealla lähteellä ja vastaanotetaan samalla määrällä vastaanottimia. Järjestelmän erikoisuus on, että se voi hyväksikäyttää Space Time Coding -menetelmää, jolloin viiveen haittoja voidaan vähentää. Samalla sekä virheenkorjaus että kantama paranee. Toisena 4G-tekniikkana Suomessa myydään kahden 3G-verkon rinnakkaista toimintamallia eli 4G dual carrier -tekniikkaa. Nämä tekniikat pystyvät tällä hetkellä maksimissaan 100 megabitin sekuntinopeuteen. Huomattavaa järjestelmissä on, että yleensä lähetysnopeus

on latausta suurempi. Todellisuudessa latausnopeudet kyllä jäävät alle 50 megabittiin sekunnissa, mikä riittää tällä hetkellä kuitenkin suurimpaan osaan sovelluksista.

Lopullisessa työssä päädyin käyttämään Sierra Wireless AirCard 762s 4G-tukiasemaa, koska kyseisessä laitteessa ominaisuuksia oli huomattavasti enemmän. Sillä voidaan luoda oma verkko. Verkkoa käyttävät laitteet oli helppo yhdistää kiinteällä osoitteella, jolloin ne automaattisesti löytävät toisensa. Myös asetukset pysyvät muuttumattomina. Tällöin on suojaus myös paremmin järjestettävissä. Laite pystyy pienimuotoiseen reititykseen, joten se mahdollistaa monenlaiset lisäsovellukset. Valmistaja lupaa myös laitteen pystyvän tuohon 100 megabitin nopeuteen, mutta käytännön testeissä ei päästy yli 40 megabitin nopeuteen, vaikka myös operaattori lupaa saman 100 megabitin nopeuden. Työssä testaus rajoittui vain DNA:n verkon testaukseen pienellä alueella. Testaamatta jäi myös tiedon siirron toiminta liikkuvassa autossa. Etenkin suuremmilla nopeuksilla, koska 3G-verkon tekninen rajoitus on 250 kilometriä tunnissa, jonka jälkeen se ei pysty enää säilyttämään yhteyttä. Olisi ollut mielenkiintoista selvittää, mikä on käytännön toimintaraja. Katkeilua ilmenee kuitenkin pienemmissäkin nopeuksissa, etenkin kaupunkiolosuhteissa.

2.2.3 CANtact

Ajoneuvoissa ja teollisuudessa on jo pitkään käytetty monenlaista väylätekniikkaa. Näiden lukemisessa ja etenkin tulkitsemisessa on vielä useampia tekniikoita. CANtact syntyi tähän tarpeeseen Kickstarter-projektina Eric Evenchickin toimesta [4]. Tarkoituksena oli luoda yhteisö, jossa jaetaan kaikki tieto ilmaiseksi ja laite, joka on helppo muuntaa monenlaisiin tarkoituksiin. CANtact sopii siis käytännössä kaikkiin CAN-sovelluksiin, vaikka onkin pääosin tarkoitettu ajoneuvojen väylien tarkasteluun. CANtact on suunniteltu niin, että se soveltuu suurimman osan CAN-pohjaisten laitteiden lukemiseen ja konfigurointiin. Esimerkiksi DB-9-tyypin liittimen kytkentä CANtactissa voidaan muuttaa toivottuihin asetuksiin,

koska näiden kytkennöissä on laitekohtaisia eroja. OBD-2 eli On Board Diagnosticsin toinen standardi on myytävissä ajoneuvoissa pakollinen. Standardin mukaisessa liittimessä on 16 pinniä, joista seitsemän on valmistajan itse määräämiä. Lisäksi joissain yhteyksissä väylässä on oltava 120 ohmin vastus. Vastuksen asettaminen onnistuu tässä laitteessa vain jumpperin siirrolla. Parasta kuitenkin laitteessa on, että kaikki dokumentit ja ohjelmat ovat kaikille avoimia ja ilmaisia. Lisäksi keskustelufoorumeilta löytyy apua moniin ongelmiin. CANtactin piirikaavio on liite 5.

CANtactin yhdistämiseksi täytyy muodostaa rajapinta, joka ymmärtää väylän tuottamaa dataa.

```
– sudo slcand -o -c -s6 /dev/ttyACM0 can0
```

Käsky nimeää /dev/ttyACM0 eli USB-portin can0:ksi, jolloin on mahdollista lukea CANtactiin kytkettyä väylää. `–s6` on väylänopeus 500 kb/s. `–o` lähettää käynnistys käskyn avatessa ja `–c` sulkee yhteyden ohjelman sammutuksen yhteydessä.

Kuva 5. CANtact lukija mahdollistaa CAN-väylän tulkinnan muuttamalla tiedon luettavaan muotoon tavalliselle tietokoneelle. CANtact osoittautui todella hyväksi ja toimivaksi laitteeksi. Sen muuttaminen erilaisten väylien tutkimiseen on tehty yksinkertaiseksi.

2.2.4 Elektrobit

Oulussa perustettiin vuonna 1985 yritys, joka aluksi erikoistui langattoman tietoliikenteen kehitykseen ja laitteiden valmistukseen. Yhteistyö Nokian kanssa ja lukuisat yrityskaupat, erityisesti JOT Automation Oyj:n kanssa vuonna 2002 johtivat kehitykseen myös autojen ohjausjärjestelmien kanssa. Vuonna 2015 Elektrobit myytiin Continental AG:lle Saksaan ja tässä vaiheessa oli luovuttu langattomista verkoista ja keskitytty jo kokonaan ajoneuvojen ohjausjärjestelmiin. Elektrobitin

EB6110-ohjauslaite, jollaista Metropolian sähköautoprojekteissa on hyvällä menestyksellä käytetty, on tarkoitettu kehitystyöhön. Laitteessa on kaksi mikrokontrollerityyppistä piiriä, joilla ajoneuvon ohjaus toteutetaan. Mikrokontrollerin suurin etu on, ettei se tarvitse toiminnassaan ulkoista muistia, vaan ainoastaan virtalähteen. Lisäksi mikrokontrollerit ovat sovellusta varten suunniteltuja, kuten Elektrobitin MPC5200 on tehty vastaamaan ajoneuvojen tarvitsemia ominaisuuksia. Lisäksi laitteessa on FPGA-tyyppinen Alteran valmistama Cyclone EP2C70 -mikrokontrolleri. FPGA-piiri mahdollistaa kriittisimmän ohjauslogiikan ohjelmoinnin suoraan piiriin, jolloin se toimii nopeasti ja pienellä kulutuksella. Molemmat kontrollerit ovat erittäin toimintavarmoja, jonka vuoksi paljon käytettyjä sellaisissa sovelluksissa, joissa toimintavarmuus täytyy olla korkea. Laitetta voidaan ohjelmoida Mathworksin luomalla Matlab Simulink -ohjelmointiympäristöllä. Simtools sisältää tarvittavat lisäosat CAN-väyläsovelluksiin. Simtarget on ohjelma jolla koodi käännetään lopuksi laitteen omalle kielelle. Myös koko EB6000-sarja on hyvin kompaktin kokoinen ja on sijoitettavissa erittäin pieneen tilaan.

Kuva 6. EB 6100- sarjan ohjainlaite edestäpäin kuvattuna Laitteen koko on kompakti 150mm * 100mm * 25mm.

2.2.5 Muut lisälaitteet

Työssä kokeiltiin monenlaisia releitä ja yhteysmuotoja. Lopputuloksena kuitenkin päädyttiin melko yksinkertaiseen rakenteeseen parhaan toimintavarmuuden saavuttamiseksi. Sivutyönä syntyi toinen laite uudemmasta Raspberry Pi 2 Model B:stä johon tuli kamera ja 4,3 tuuman näytön. Tämä toinen laite voisi toimia graafisena liittymänä kulunvalvonnassa. Toinen laite kykenisi toimimaan WLAN-verkossa omana laitteena. Tosin siitä puuttuisi tuo rele, joka mahdollistaa käyttöjärjestelmän lataamisen. Alkuperäinen 3G/GPS-moduuli oli todella kompaktin kokoinen, jos siihen vain olisi saanut jatkuvan yhteyden. Moduulin kanssa käskyjen vaihto tapahtui AT-käskyjen avulla jotka olivat rajalliset kaikin puolin. Kun tietokoneiden välinen yhteydenpito tuli tarpeelliseksi verkossa, kehitettiin suuri määrä protokollia, joista osa on jo jäänyt historiaan. Matkapuhelinverkossa tiedonsiirtoon kehitettiin modeemit. Nykyään kuitenkin vielä GSM- ja 3G-verkoissa on edelleen käytössä todella vanhoja ja yksinkertaisia laitteita. Eräs näistä protokollista on Hayes command set. Käskykannan kehitti vuonna 1981 Dennis Hayes kehittämäänsä Hayes Smartmodem -laitteeseen. Tämä laite kykeni 300 bittiä/sekunti siirtonopeuteen, ja käskykantaa jouduttiin laajentamaan laitteiden kehittyessä. AT-komento tulee siis sanasta ”attension” ja onkin yleensä jonkin käskyn etuliite. Esimerkiksi laitteen käynnistyminen varmistetaan pelkällä AT-käskyllä, johon laite vastaa ollessaan käynnissä OK. Tavallinen puhelinsoitto on ATDT<numero johon tahdotaan soittaa>. Signaalin voimakkuuden ja operaattorin tiedot saa käskyllä AT+COPS? Muita esimerkkikäskyjä on liitteessä lisää. Cooking-hacksin laite soveltuisi ilmoittamaan ongelmista kentällä. Toisaalta modeemi voisi esimerkiksi lähettää kuvan sen hetkisestä kuljettajasta mutta tiedonsiirtoon se ei tekniikaltaan soveltunut.

3 Tekninen toteutus

3.1 Laitteen käyttöönotto

Käyttöönottoa varten tarvitaan Raspberry Pi -tietokone, SD-kortti, HDMI-johto, näyttö ja näppäimistö. Lisäksi tarvitaan toinen tietokone, jossa on yhteys ulkoverkkoon tiedostojen lataamiseksi SD-kortille. On myös mahdollista tilata SD-kortti, jossa on valmiiksi asennettuna NOOBS eli New Out Of the Box Software, jolloin tietokone käynnistyy suoraan graafiseen käynnistysvalikkoon.

Raspberry Pin käyttöönotto on tehty yksinkertaiseksi, mutta lisäominaisuudet ovat toisinaan vähän haastavia. Esimerkiksi GPIO-pinnejä käyttävät lisälaitteet yleensä käyttävät samoja pinnejä. Pinnien muuttaminen on melko yksinkertaista ja onnistui pienellä tutkimisella.

Tämän työn laitteen käyttöönottoon saattamiseen kuului seuraavat 12 vaihetta.

1. Levyn alustus (formatointi)

- Tehdään esimerkiksi koodivirheen jälkeen ja suositellaan muulloinkin. SD-kortti on alustettava, koska muuten sen kaikki ominaisuudet eivät ole käytössä. Minä käytin tähän diskparttia, joka löytyy Windows-koneista valmiina. Diskpart käynnistetään ja valitaan oikea levy. Oikean levyn valinta on hyvin tärkeää, ettei tyhjennä vahingossa käytettävän tietokoneen kovalevyä. Esimerkissä SD-kortti löytyy disk 1 -nimellä listauksessa ja siinä on kaksi osiota, jotka siis tyhjennetään. SD-kortti on kokonsa normaalitilanteessa melko helppo löytää. Seuraavat käskyt tyhjentävät osiot kortilta, jonka jälkeen se on valmis alustettavaksi:

- `list disk`
- `select disk 1`
- `list part`
- `select part 0`
- `delete part`
- `select part 1`
- `delete part` (`-delete partition override`)

2. Imagen teko

- Tähän käytin Win32DiskImager:ia yksinkertaisuutensa vuoksi. Käyttöjärjestelmä ladataan esimerkiksi sivulta

https://www.raspberrypi.org/downloads/raspbian/

- Tämä asennetaan SD-kortille. Tämän jälkeen kortti asennetaan Raspberry Pihin ja laite käynnistetään.

3. Network-asetukset

- Eli tiedosto /etc/interfaces/network muutetaan, jotta SSH-yhteys onnistuu ja laite on käytettävissä verkon kautta. Tiedoston esimerkki löytyy liitteistä. Tähän toimintaan tarvitaan näyttö, jonka jälkeen laite on konfiguroitavissa verkon kautta eikä näyttöä siis enää tarvita.

4. Näppäimistön muutos

- Avataan tiedosto.

  – `sudo nano /etc/default/keyboard`

- Tämän jälkeen vaihdetaan tiedostosta maakohtainen tunnus.

  `XKBLAYOUT="gb"` muutetaan muotoon `XKBLAYOUT="fi"`

- Tällöin näppäimistö on Suomen asetuksissa.

5. Raspberry Pi:n update, ladataan uusimmat päivitykset

   – `sudo apt-get install update`

6. Raspberry Pi:n upgrade, asennetaan päivitykset käyttöön

   – `sudo apt-get install upgrade`

7. SPI-pinnien käyttöönotto

   – `sudo raspi-config`

   Käsky avaa graafisen konfigurointivalikon, josta suoritetaan seuraavat:

- SD-kortin laajennus (expand file system)
- käyttäjänimen vaihto
- SSH-yhteyden salliminen
- maa-asetukset.
- SPI interfacen hallinta.

Valikosta on helppoa asettaa ja hallita monia asetuksia myöhemminkin.

8. Kirjastojen lataus ja käännökset

- Ensin ladataan kirjastot.

  www.cooking-hacks.com/skin/frontend/default/cooking/images/catalog/documentation/raspberry_arduino_shield/arduPi_1-5.tar.gz

  www.cooking-hacks.com/skin/frontend/default/cooking/images/catalog/documentation/raspberry_arduino_shield/arduPi_changelog.txt

- Kirjastojen purku (x=purkaminen f=tiedosto tyyppi z=gunzip purku samalla).

  - `tar -xfz arduPi_1-5.tar.gz`

  - `tar xfz arduPi_changelog.txt`

- Kun nämä on purettu, kirjasto täytyy vielä kääntää, jotta sitä voi käyttää.

  - `g++ -c arduPi.cpp -o`

- Ohjelmat voidaan kääntää seuraavalla käskyllä ajettavaan muotoon.

  - `g++ -lrt -lphtread oma_ohjelma.cpp ardiPi.o -o oma_ohjelma`

9. Ohjelmien lataus advanced packaging toolin avulla

   - `sudo apt-get minicom`

   - `sudo apt-get vwdial`

   - `sudo apt-get slcand`

   - `sudo apt-get install python-wxtools`

10. Serial-asetusten muutos

    - `sudo systemctl stop serial-getty@ttyAMA0.service`

11. Käynnistys tiedoston -muutos komentoriviltä käsin

    - `sudo nano /boot/cmdline.txt`

    - Josta poistetaan teksti ”console = ttyAMA0”. Tämä tehdään, jotta serial-portti ei lähetä käynnistysvaiheessa mitään käskyjä eteenpäin.

12. GPIO-pinnien resetointi käynnistyksen yhteydessä

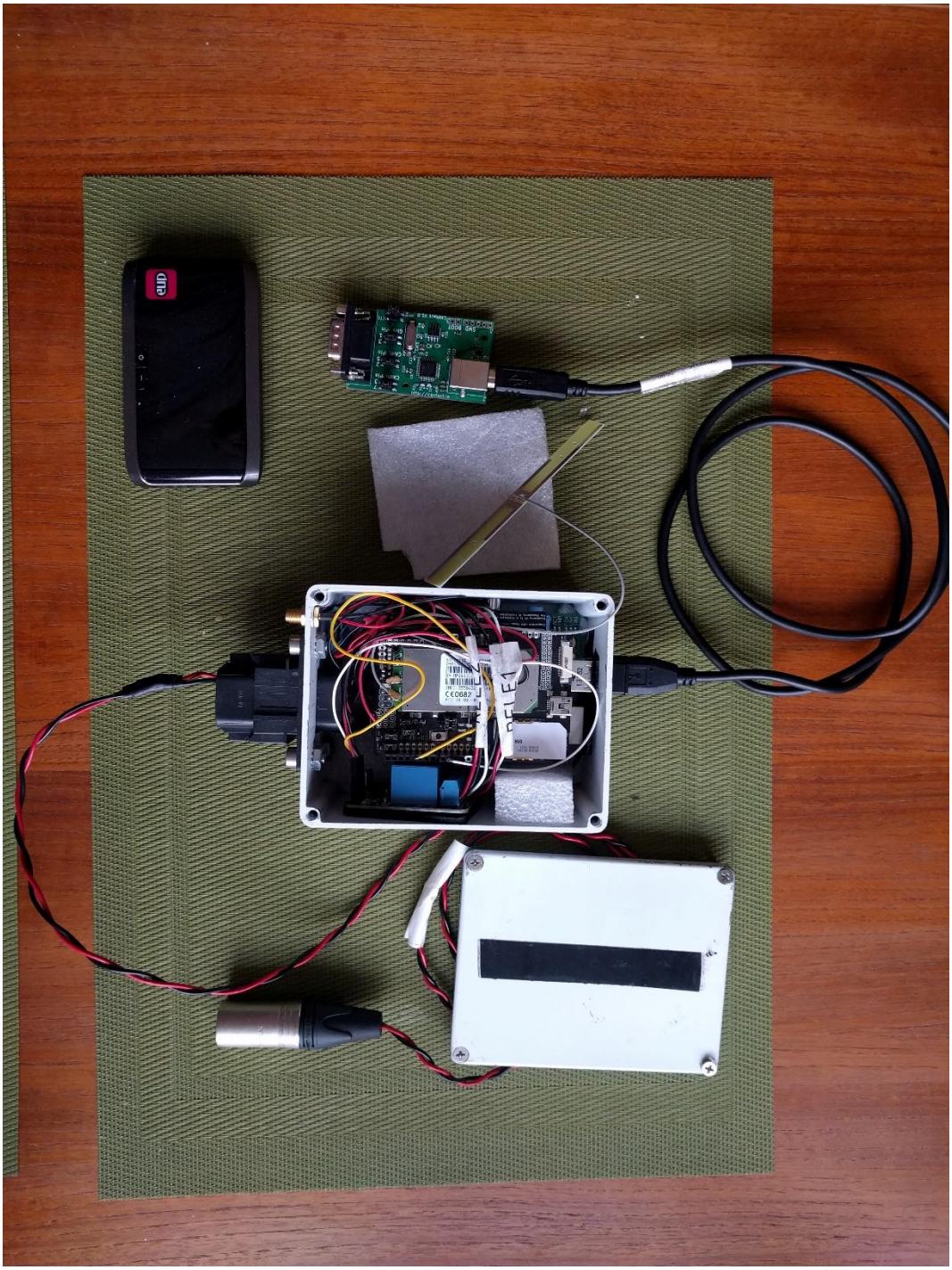
    - Työssä päädyttiin ratkaisuun, jossa luotiin tiedosto nimeltä `startup` [liite 4.], asettamaan GPIO-pinneistä ne, joita tarvitaan kytkennöissä 0-tilaa

vastaavaan tilaan. Työssä käytettiin ulkopuolista relettä ajoneuvon laitteiden ohjaukseen. Resetointi oli pakollinen, koska releen kärki avautuessaan katkaisisi muutoin jännitteen toiminnoilta, jotka eivät normaalitilanteessa saa katketa. Nykyisellä järjestelyllä releen kärki pysyy kiinni, kunnes ohjelma sen aukaisee. Tiedosto `startup` lisätään tekstinä tiedostoon `/etc/rc.local`, jolloin se suoritetaan aina käynnistyksen yhteydessä. Esimerkkitiedosto on liitteessä 3.

Laite on konfiguroitavissa tämän jälkeen SSH-yhteydellä, joten se on valmis asennettavaksi ajoneuvoon tai muuhun mobiililaitteeseen. Yhteydenottoa varten tarvitaan joko ethernet-kaapeli tai WLAN-sovitin ja käyttöön soveltuva WLAN-verkko. Tarvittavat lisälaitteet ovat toimintakunnossa. Kirjastojen lataus ja muut käskyt löytyvät liitteenä. Työn tekovaiheessa havaittiin, että GPIO-pinnien käyttöönotto ja perusasetus ovat hankalin osa. Tämän vuoksi nämä erityisen tarkasti kuvattuna liitteinä. Vaikka lähes kaikki ohjelmat, joita tarvitaan, löytyvät valmiina, joutuu aina tiedostoja muokkaamaan käyttötarkoitukseensa sopiviksi asettamalla omat tiedot ohjelmiin, ja kääntämään ne suoritettavaan muotoon.

3.2 Laitteen testaus

Käytännön testit ovat melko alkutaipaleella eli niitä suoritettiin pieni määrä. Todettua on, että laite toimii odotetulla tavalla. Lisäksi on tutkittu kerätyn tiedon tallennusmuotoja ja lopputuloksena todettiin, että suositeltavaa olisi tallentaa väylän liikenne yksinkertaisimmassa mahdollisessa muodossa, jonka jälkeen tarvittaessa siirtää se tallennettavaksi esimerkiksi palvelimelle. Tästä tiedostosta voi sitten jälkikäteen tarvittaessa kohtuullisen helposti oikeanlaisella ohjelmalla tutkia tapahtumia. Tiedonsiirto vaatii aina aikaa ja kaistanleveyttä. Toisena vaihtoehtona olisi käytännössä muuttaa kerätty ja tarpeelliseksi todettu tieto, ja muuttaa se luettavaan muotoon, esimerkiksi kaavioksi, ja vasta sitten lähettää se edelleen. Kuitenkin laitteen teho on rajallinen, jonka vuoksi tämä ei ole yhtä tehokasta. Kulunvalvontaa on testattu vain toimisto-olosuhteissa, joissa se vaikuttaisi toimivan odotetusti.

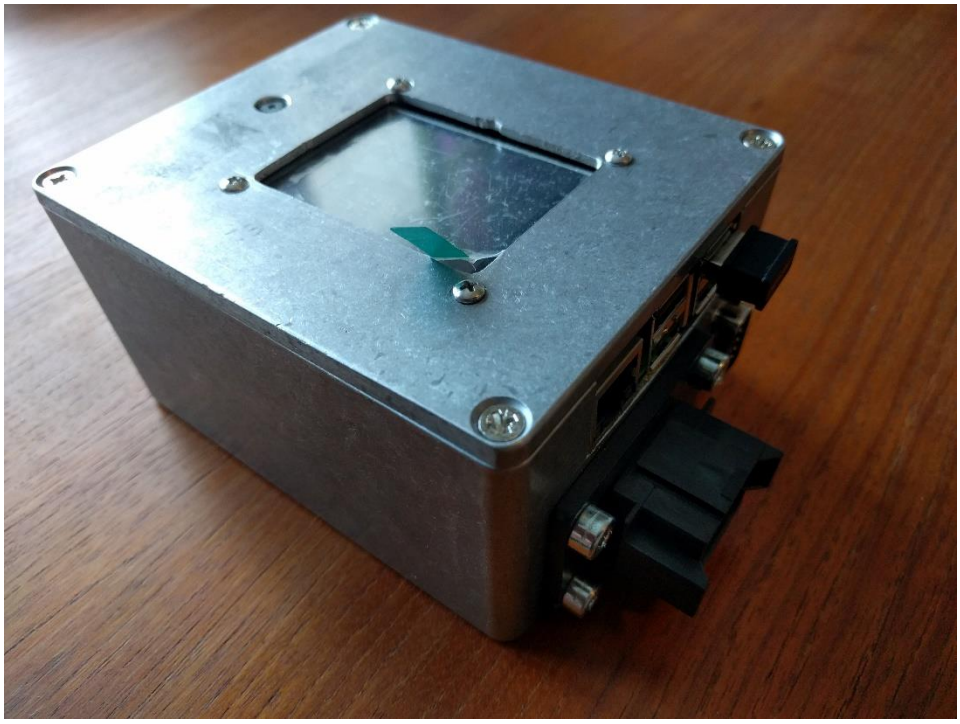


Kuva 7. Laitteiston rakenne. Työssä käytettiin alumiinista valmistettua rasiaa tekniikan suojaamiseksi. Rasian koko on 120 * 92 * 56 mm

4 Työn tulokset

4.1 Kulunvalvonta

Kulunvalvonnan tavoitteita on vikojen yhdistäminen käyttäjiin ja ajoneuvojen käyttömäärien seuraaminen tilastotarkoituksessa. Tarkka kirjanpito antaisi arvokasta tietoa laitteiden eliniän määrityksessä, kuten myös laitteistojen kulumisen tutkimisessa. Myös häiriöiden vikatiheys kilometrien ja tuntien suhteessa olisi hyödyllistä. Ei ole tarkoituksen mukaista henkilöidä vikoja vaan enemmänkin tutkia tapahtuuko jokin vika samalla henkilöllä vai yleisesti kaikilla. Paljon auttaisi esimerkiksi kuva ajoneuvon ulkopuolesta tasaisin väliajoin, koska kuva antaisi tehokasta tietoa sääolosuhteista ja vuodenajasta kuten myös ajoneuvon sijainnista. Tämä kuva ja kerätty data ajoneuvon väylästä voitaisiin yhdistää, jolloin saavutettaisiin huomattavaa hyötyä.



Kuva 8. Toinen laite jossa Raspberry Pi 2 –malli sisällä. Tämä osoittautui huomattavasti nopeammaksi ja kosketusnäyttö toi paljon lisämahdollisuuksia.

4.2 Tiedonkeruu

Työ tehtiin osana Metropolian sähköisen liikenteen projektia. Ajoneuvoja on projektin käytössä tällä hetkellä kolme. Kaksi Fiat Dobloa joihin italialainen yritys Microvett on joitakin vuosia sitten vaihtanut dieselmoottorin tilalle sähköisen voimanlähteen, eli konvertoinut sähköisiksi Italiassa. Nykyään ne sisältävät suomalaista osaamista niin European Batteriesin akkujen kuin ABB:n taajuusmuuntajien osalta. Lisäksi kehityksessä mukana on Electric RaceAbout -sähköurheiluauto, jolla on tehty monenlaista uraa uurtavaa kehitystä sähköautojen ja niiden voimansiirron kehityksessä. Näissä ajoneuvoissa on kaksi CAN-väylää, joista toinen on taajuusmuuntajaa eli moottorin ohjausta varten ja toinen ajoneuvon muiden laitteiden ohjaukseen. Jatkuvaan valvontaan on siis oltava kaksi väyläsovitinta. Väylän tarkkailu on melko yksinkertaista tiedonsiirtoa ja onnistuu kätevästi esimerkiksi minicom-nimisellä ohjelmalla. Ohjelmalla voi myös tallentaa tekstimuotoisena tiedostona luetun väylän koko tiedonsiirron, jolloin sitä voidaan tutkia erilaisilla analysointiohjelmilla. Tiedonsiirron määrä on väylää kohti noin 40 megatavua tunnissa.

5 Lopputulos

Insinöörityön lopputuloksena syntyi käytännössä kaksi laitetta. Molempien laitteiden kehitysmahdollisuudet tuntuvat hyvin laajoilta, mutta toisaalta myös toimivilta. Ajanpuutteen vuoksi työtä olikin rajattava melkoisesti. Verkkosivu ajoneuvojen sijainnin esitykseen olisi mahdollinen, koska molemmat laitteet keräävät jatkuvaa tietoa sijainnistaan. Käyttöjärjestelmän päivitys onnistuisi samoin, koska siihen vaaditut elementit toimivat hyvin ja luotettavasti.

Myös Codesys-ohjelman tarjoamat mahdollisuudet vaikuttavat toimivilta. Yhteys Wlanin kautta on myös mahdollinen molemmissa laitteissa, ja mainitun 4G-reitittimen kanssa voidaan muodostaa helposti oma verkko. Tiedostojen jakaminen ja laitteiden käyttö on omassa verkossa erittäin toimivaa.

Käytännön testit jäivät melko vähäisiksi ajanpuutteesta johtuen. Testaus suoritettiin suuressa hallissa ja ajamalla normaalin liikenteen seassa. Suurin yksittäinen työ oli erillisten releiden hankinta ja näiden ohjelmointi. Kytkentä aiheutti myös jonkin verran työtä. GPIO-pinneistä lopuksi käytettiin pinnejä 22 ja 24, kuten liitteestä 3 nähdään.

Releillä on tarkoitus ohjata ajoneuvojen ohjauslaite ohjelmointitilaan. Laitteessa joka tähän toimintoon saatiin työssä tehtyä, on varattu tähän kaksi relettä. Molemmissa releissä on sulkeutuva ja avautuva kärki käytettävissä. Lisäksi toisella laitteella, jossa oli käytössä tehokkaampi Raspberry Pi 2 -versio, tehtiin testejä. Testeissä ilmeni, että sen käynnistymisajat ovat nopeammat, ja se jaksoi jo helposti pyörittää kameraa ja kosketusnäyttöä. Tämä mahdollistaisi kulunvalvonnan toteutuksen, jossa laitteeseen kirjattaisiin käyttäjä ja muita tietoja käytöstä.

Sovelluksia on todella paljon, joihin tämän kaltainen laite sopisi. Myös nykyisiä ominaisuuksia voisi kehittää, mutta uusia ideoita olisi hienoa nähdä uusien opiskelijoiden toimesta tulevaisuudessa.

Lähteet

Raspberry Pin virallinen verkkosivu <https://www.raspberrypi.org/about/>. Luettu 7.9.2015.

CANopen-väyläprotokollan virallinen sivusto <http://www.canopen.org/>. Luettu 10.1.2016.

Verkkodokumentti 1.12.2011 päivitetty 2.6.2014 <http://digital.ni.com/public.nsf/allkb/D5DD09186EBBFA128625795A000FC025>. Luettu 2.12.2015.

Eric Evenchikin omat sivut, joilta löytyy enemmän tietoa hänen omista projekteistaan <http://www.evenchick.com/>. Luettu 20.11.2015.

Wolfhard, Lawrenz. 2013. CAN System Engineering - From Theory to Practical Applications.

Verkkodokumentti http://www.canopensolutions.com/english/about_canopen/canopen-management.shtml Luettu 2.2.2016

Liite 1. Tiedosto /etc/network/interfaces

```
auto lo

iface lo inet static

address 192.168.0.110

netmask 255.255.255.0

#gateway 192.168.0.1

broadcast 192.168.0.255

#dns-nameservers 192.168.0.1 8.8.8.8 8.8.4.4


allow-hotplug wlan0

auto wlan0

iface wlan0 inet static

address 192.168.1.110

netmask 255.255 255.0

gateway 192.168.1.1

broadcast 102.168.1.255

dns-nameservers 192.168.1.1 8.8.8.8 8.8.4.4


     wpa-ssid" wlan ssid "

     wpa-psk" wlan passwd "
```

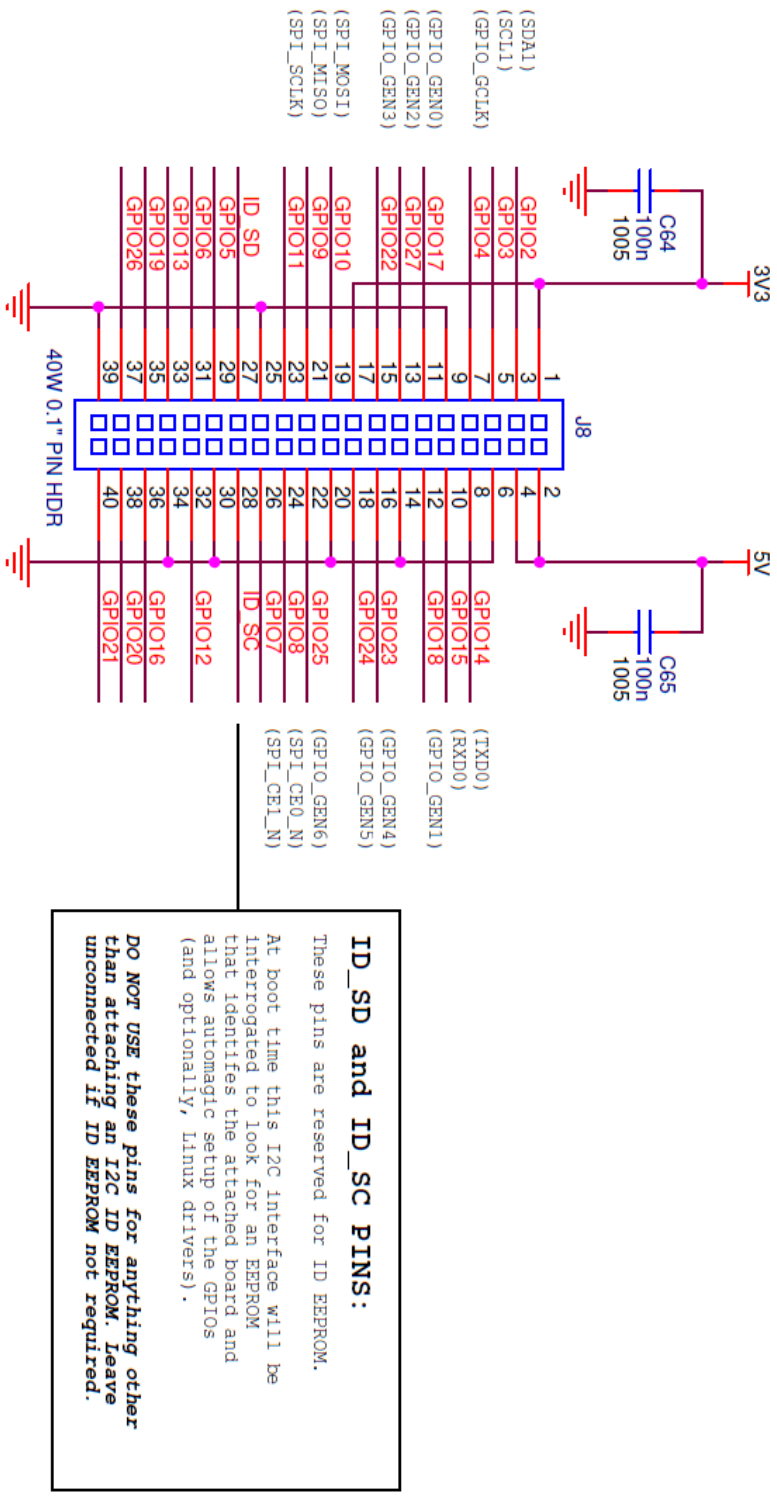
Liite 2. Tiedosto /etc/rc.local

```
#!/bin/sh -e

# rc.local

# This script is executed at the end of each multiuser run-
level.

# Make sure that the script will "exit 0" on success or any
other

# value on error.

# In order to enable or disable this script just change the
execution

# bits.

# By default this script does nothing.

# Print the IP address

_IP=$(hostname -I) || true

if [ "$_IP" ]; then

  printf "My IP address is %s\n" "$_IP"

fi



./etc/startup



exit 0
```

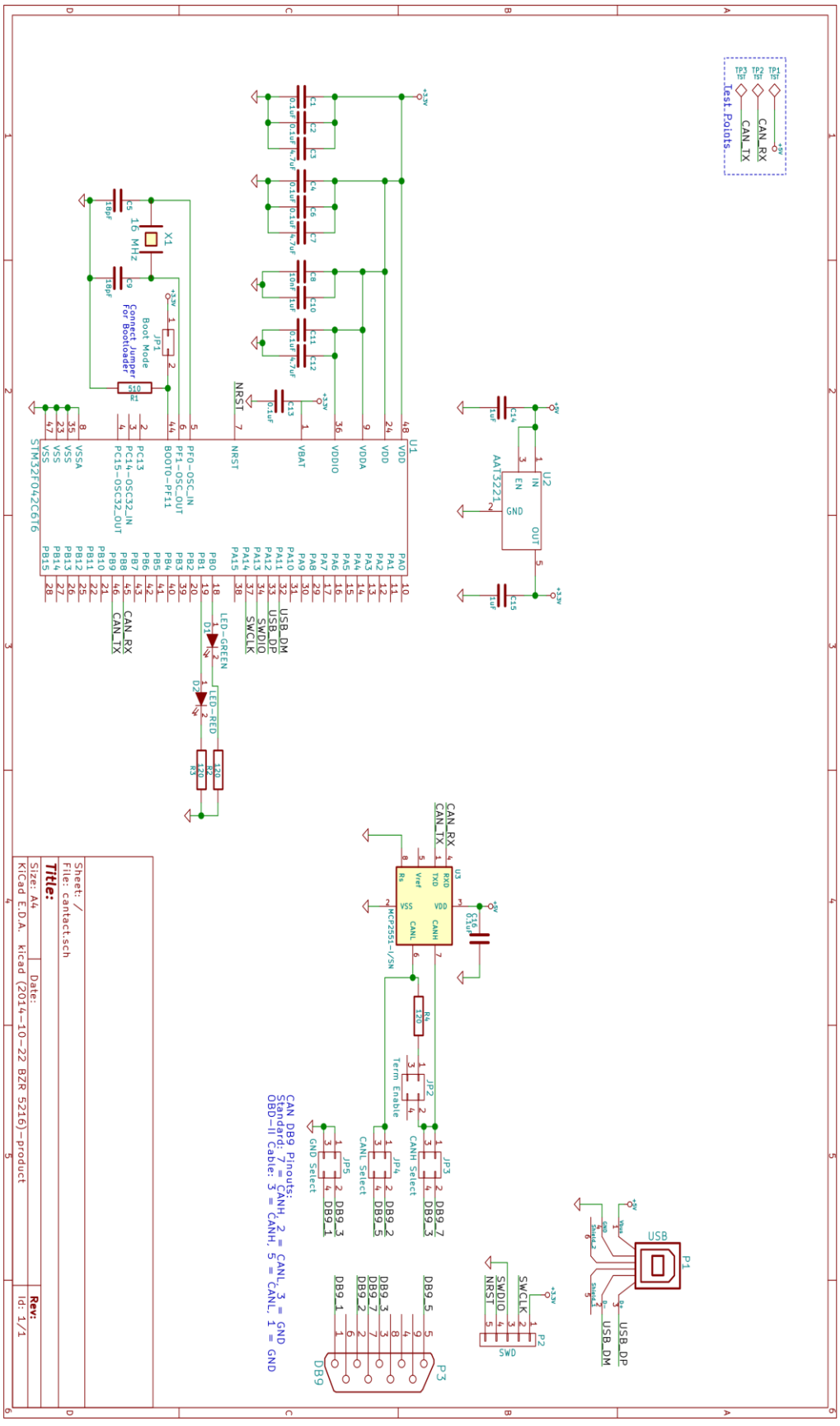
Liite 3. Tiedosto /etc/startup

```
echo 22 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio22/direction

echo 1 > /sys/class/gpio/gpio22/value



echo 24 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio24/direction

echo 1 > /sys/class/gpio/gpio24/value
```

Liite 4. Raspberry pi GPIO pinout



Liite 5. CANtact piirikaavio



Liite 6. Raspberryn peruskäskyt

General Commands

apt-get update	Updates your version of Raspbian.
apt-get upgrade	Upgrades all of the software packages you have installed.
clear	Clears the terminal screen of previously run commands and text.
date	Prints the current date.
find / -name example.txt	Searches the whole system for the file example.txt and outputs a list of all directories thatc ontain the file.
nano example.txt	Opens the file example.txt in "Nano", the Linux text editor.
poweroff	To shutdown immediately.
raspi-config	Opens the configuration settings menu.
reboot	To reboot immediately.
shutdown -h now	To shutdown immediately.
shutdown -h 01:22	To shutdown at 1:22 AM.
startx	Opens the GUI (Graphical User Interface).

File/Directory Commands

cat example.txt	Displays the contents of the file example.txt.
cd /abc/xyz	Changes the current directory to the /abc/xyz directory.
cp XXX	Copies the file or directory XXX and pastes it to a specified location; i.e.
cp examplefile.txt /home/pi/office/	copies examplefile.txt in the current directory and pastes it into the /home/pi/ directory. If the file is not in the current directory, add the path of the file's location (i.e.cp /home/pi/office/ copies the file from the documents directory to the office directory).
ls -l	Lists files in the current directory, along with file size, date modified, and permissions.
mkdir	example_directory: Creates a new directory named example_directory inside the current directory.
mv XXX	Moves the file or directory named XXX to a specified location. For example,mv examplefile.txt

	/home/pi/office/ moves example-file.txt in the current directory to the /home/pi/office directory. If the file is not in the current directory, add the path of the file's location (i.e. cp /home/pi/documents/examplefile.txt /home/pi/office/ moves the file from the documents directory to the office directory).This command can also be used to rename files (but only within the same directory). For example, mv example-file.txt newfile.txt renames examplefile.txt to newfile.txt, and keeps it in the same directory.
rm example.txt	Deletes the file example.txt.
rmdir example_directory	Deletes the directory example_directory (only if it is empty).
scp user@10.0.0.32:/some/path/file.txt	Copies a file over SSH. Can be used to download a file from a desktop/laptop to the Raspberry Pi. user@10.0.0.32 is the username and local IP address of the desktop/laptop and /some/path/file.txt is the path and file name of the file on the desktop/laptop.
touch	Creates a new, empty file in the current directory.

Networking/Internet Commands

ifconfig	To check the status of the wireless connection you are using (to see if wlan0 has acquired an IP address).
iwconfig	To check which network the wireless adapter is using.
iwlist wlan0 scan	Prints a list of the currently available wireless networks.
iwlist wlan0 scan | grep ESSID	Use grep along with the name of a field to list only the fields you need(for example to just list the ESSIDs).
nmap	Scans your network and lists connected devices, port number, protocol, state (open or closed)operating system, MAC addresses, and other information.
ping	Tests connectivity between two devices connected on a network. For example, ping 10.0.0.32 will send a packet to the device at IP 10.0.0.32 and wait for a response.

	It also works with website addresses.
wget http://www.website.com/example.txt:	Downloads the file example.txt from the web and saves it to the current directory.

System Information Commands

cat /proc/meminfo	Shows details about your memory.
cat /proc/partitions	Shows the size and number of partitions on your SD card or hard drive.
cat /proc/version	Shows you which version of the Raspberry Pi you are using.
df -h	Shows information about the available disk space.
df /	Shows how much free disk space is available.
dpkg –get-selections | grep XXX	Shows all of the installed packages that are related to XXX.
dpkg –get-selections	Shows all of your installed packages.
free	Shows how much free memory is available.
hostname -I	Shows the IP address of your Raspberry Pi.
lsusb	Lists USB hardware connected to your Raspberry Pi.
UP key:	Pressing the UP key will enter the last command entered into the command prompt. This is a quick way to correct commands that were made in error.
vcgencmd measure_temp	Shows the temperature of the CPU.
vcgencmd get_mem arm && vcgencmd get_mem gpu	Shows the memory split between the CPU and GPU.

Liite 7. AT-käskyt

AT	Näyttää onko laite valmis, vastaus OK jos on
AT+CPIN=1234	Syöttää PIN-koodin laitteeseen
AT+CPWD="SC","vanha","uusi"	Vaihtaa laitteen PIN-koodin vanhasta uuteen
AT+CLCK="SC",0,"1234"	Poistaa PIN-koodin
AT&V	Näyttää laitteen tilan ja käskykannan
ATI	Tilatiedot (Valmistaja, Malli, Revision, IMEI, toiminta)
AT+COPS=?	Listaa verkkotiedot 0-Unknown/2-Current/3-Forbidden, Longname, Shortname, Numerical-ID, "AcT"
AT+CSQ	Näyttää verkon voimakkuuden.
ATD*99#	Puhelun soitto numeroon *99#
AT+CGDCONT=1,"IP","ap.name"	Määrittelee verkon asetukset
AT+CBC	näyttää laitteen lataus,varaus ja jännite tiedot

Liite 8. Esimerkki CANopen viestitunnuksista ja väylän tiedonsiirrosta.

CANopen tunnukset					
Viestin tyyppi	Funktion koodi (binary)	ID-tunnus	Database Communication Index	Database Mapping Index	Kommentit
NMT	0	000H	-	-	Tärkein viesti
SYNC	1	080H	(1005H)	-	
TIME STAMP	10	100H	-	-	
EMERGENCY	1	081-0FFH	-	-	node-id + 080H
PDO1 (tx)	11	181-1FFH	1800H	1A00H	node-id + 180H
PDO1 (rx)	0100	201-27FH	1400H	1600H	node-id + 200H
PDO2 (tx)	101	281-2FFH	1801H	1601H	node-id + 280H
PDO2 (rx)	110	301-37FH	1401H	1601H	node-id + 300H
SDO (tx) server	1011	581-5FFH	1200H	-	node-id + 580H
SDO (rx) client	1100	601-67FH	1280H	-	node-id + 600H
Nodeguard	1110	701-77FH	(100EH)	-	node-id + 700H

t703815500115500115001t1F587EEB000000000000t3018000000000000000t7388000000000FF7F0000t7408000000000000000t7328420F420FA712A712t7348000000000000000
t7398000000000000000t7418000000000000000t29E80205B3FFE5002200t5505030900078St7028000000000000000t1538244900FF00FFFF10t7178060008003C000000
t7388000000000FF7F0000t7318000000000006400t7308000000000000000t7378230023003200t7398000000000000000t19E80605B2FFFFFF0000
t1F38028000FFC1FF0008t5505030900078Bt5216010586010000t055800EE0221FF000000t524604028CBE0700t7408000000000000000t1F38028000FFC1FF0008
t7418000000000000000t550503090007A7t70080010260550000F0t7028000000000000000t7148000000000000000t7168000000000000000t7158000000000000000
t7188000000000000000t7408000000000000000t7328420F420FA712A712t7348000000000000000t7398000000000000000t55050309000758t1538244900FF00FFFF10
t055800EE0221FF000000t52260207AEFCFFFFt7388000000000FF7F0000t7408000000000000000t7328420F420FA712A712t7348000000000000000t1538244900FF00FFFF10
t1F0809600900090009000t19E80405B4FF02000000t5008000000000A47FA0t70164000006400000t700800002605300000F0t7138000000000000000t7148000000000000000
t1538244900FF00FFFF10t1F0809000900090009000t1F38028000FFC1FF0008t7188000000000000000t7348000000000000000t7368000000000000000t7418000000000000000
t19E80105B4FFFDFF0300t1538244900FF00FFFF10t5216010990010000t055800EE0221FF000000t5246040562BE0700t26A809E88B0100012313t7388000000000FF7F0000
t7368000000000000000t7318000000000006400t7308000000000000000t1538244900FF00FFFF10t7008000026056000000F0t7028000000000000000t7148000000000000000
t7188000000000000000t1F587EEB000000000000t5246040641BE0700t7388000000000FF7F0000t1538244900FF00FFFF10t1F0809600900090009000t7368000000000000000
t7318000000000006400t1008030349466FFFFFFFFt1028FDFF8CFFFFFFFFFFFt5216010B96010000t1F0809000900090009000t1F38028000FFC1FF0008t7388000000000FF7F0000
t7408000000000000000t7308000000000000000t5505030900070Dt1F0809200900090009000t7028000000000000000t5216010C93010000t7188000000000000000t059100
t1F587EEB000000000000t584807000E0005007202t1108FFCFFFFFFFFFFFFt1F38028000FFC1FF0008t7328420F420FA712A712t7308000000000000000t7378230023003200
t7398000000000000000t5008000000000A47FA0t5038E8030000F3010000t1F587EEB000000000000t3018000000000000000t7388000000FF7F0000t7408000000000000000
t7328420F420FA712A712t7378230023003200t7398000000000000000t7418000000000000000t5505030900072Ft19E8FE04B0FFFFFF0200t1538244900FF00FFFF10
t722800000000C1FFACFFt70551FFF0F000Ft055800EE0221FF000000t7158000000000000000t7178060008003C000000t7388000000FF7F0000t7368000000000000000
t7318000000000006400t1F0809200900090009000t1F38028000FFC1FF0008t5505030900070D0t055800EE0221FF000000t1538244900FF00FFFF10t1F080940090009000900
t1F38028000FFC1FF0008t7348000000000000000t7368000000000000000t7318000000000006400t19E80205B2FF00000600t32981168DB0500000000t7008000026055000000F0
t1F38028000FFC1FF0008t7168000000000000000t1F587EEB000000000000t5246040B52BE0700t7388000000000FF7F0000t7318000000000006400t1538244900FF00FFFF10
t19E80605B3FFFDFF0600t5505030900007F2t055800EE0221FF000000t27F301B501t7388000000000FF7F0000t1F38028000FFC1FF0008t7408000000000000000t7318000000000006400
t7308000000000000000t5226020823FDFFFFt7378230023003200t32981168DB0500000000t5505030900007E3t71250E0F0D0D0Ft7048155001144001150 0t58080400D2130200FB0F
t1F0809400900090009000t7148000000000000000t7188000000000000000t1F587EEB000000000000t7368000000000000000t7318000000000006400t55050309000794
t1F0809600900090009000t055800EE0221FF000000t1F587EEB000000000000t7388000000000FF7F0000t1538244900FF00FFFF10t7308000000000000000t26B5FA1A000000
t7378230023003200t5505030900078St70080000260550000 0F0t055800EE0221FF000000t1F0809200900090009000t1F587EEB000000000000t7388000000000FF7F0000
t7318000000000006400t7418000000000000000t5505030900007B6t1F0809400900090009000t055800EE0221FF000000t5216010595010000t7388000000000FF7F0000
t7348000000000000000t7368000000000000000t1538244900FF00FFFF10t1F0809600900090009000t19E80205B1FFFEFF0000t32981168DB0500000000t7008000026065000000F0
t7138000000000000000t7178060008003C000000t7198B4FD0000FFFF0000t1F38028000FFC1FF0008t1F587EEB000000000000t7368000000000000000t7418000000000000000
t055800EE0221FF000000t5038E8030000F3010000t1538244900FF00FFFF10t7388000000000FF7F0000t3018000000000000000t3028000000000000000t7348000000000000000
t7368000000000000000t7318000000000006400t19E80105AFFFFFFF0000t29E80205B0FFE4002200t1F38028000FFC1FF0008t700800F02504300000F0t7138000000000000000
t7178060008003C000000t7198B3FD0000000000000t7388000000000FF7F0000t7368000000000000000t7318000000000006400t7308000000000000000t7398000000000000000
t7418000000000000000t055800EE0221FF000000t1F587EEB000000000000t1538244900FF00FFFF10t7388000000000FF7F0000t7348000000000000000t7368000000000000000
t7318000000000006400t5236030811BE0700t70080000260830000F0t1F38028000FFC1FF0008t72280000000C1FFABFFt7168000000000000000t7158000000000000000
t7388000000000FF7F0000t7318000000000006400t1538244900FF00FFFF10t055800EE0221FF000000t5216010385010000t5505030900076Bt1538244900FF00FFFF10
t7408000000000000000t7308000000000000000t7378230023003200t7398000000000000000t32981168DB0500000000t524604066BBE0700t7016400000640000