TARTU ÜLIKOOL

Loodus- ja tehnoloogiateaduskond

Tehnoloogiainstituut

Sander Tammesoo

Suhtlusprotokoll ESTCube-2 alamsüsteemide vaheliseks suhtluseks

Bakalaureusetöö (12 EAP)

Juhendajad:

M.Sc. Indrek Sünter

Ph.D. Mart Noorma

Tartu 2015

Resümee

Eesti Tudengisatelliidi projekt alustas 2008. aasta suvel Tartu Ülikoolis eesmärgiga edendada kosmosetehnoloogiaalaseid teadmisi. Viis aastat hiljem lennutati Maa orbiidile Eesti esimene satelliit ESTCube-1. ESTCube-1 missiooniks oli testida elektrilise päikesepurje printsiipi Maa ionosfääri plasmas. Programmi järgmiseks etapiks on viia läbi sarnane eksperiment Maast vähemalt 90 000 km kaugusel, et vältida magnetvälja mõjutusi, näiteks Kuu orbiidil. Korrektselt töötava satelliidi eelduseks on alamsüsteemide omavaheline usaldusväärne suhtlus, mille tagab satelliidisisene suhtlusprotokoll. ESTCube-1 ja ESTCube-2 arhitektuurid on aga niivõrd erinevad, et esimese jaoks väljatöötatud suhtlusprotokolli ei ole võimalik integreerida teisele satelliidile. Käesoleva töö eesmärgiks on välja selgitada võimalikud andmelülikihi protokollid, mida oleks võimalik ESTCube-2 puhul kasutada. Vajadusel kirja panna ESTCube-2 alamsüsteemide vahelise uue andmelülikihi protokolli spetsifikatsioon ning implementeerida prototüüplahendus protokolli testimiseks. Antud töö jaguneb kuueks peatükiks. Esimene peatükk on sissejuhatus. Teine peatükk kirjeldab planeeritava ESTCube-2 riistvara ja suhtlussiini ülesehitust. Kolmandas peatükis käsitletakse tõstatatud probleemi ning arutletakse erinevate standardsete protokollide kasutatavuse üle. Uue protokolli spetsifikatsioon ja kirjeldus on leitav neljandast peatükist. Viiendas peatükis on käsitletud töö käigus esinenud suurimaid probleeme, kasutusjuhtude analüüsist ilmnenud vigadest tingitud protokolli muutusi ja implementatsiooni testimist. Viimases peatükis on välja pakutud edasise arenduse sammud.

Sisukord

Resümee **2**

Lühendid **5**

1 Sissejuhatus **7**

1.1 Töö eesmärk ja ülevaade 7

2 ESTCube-2 **9**

2.1 ESTCube-2 missioon 9

2.1.1 Elektriline päikesepuri 9

2.1.2 Missiooni eesmärk 10

2.2 ESTCube-2 kirjeldus 10

2.2.1 ESTCube-2 kasutatav riistvara kirjeldus 10

2.2.2 ESTCube-2 võrgutopoloogia kirjeldus 11

2.2.3 ESTCube-2 suhtlussiini kirjeldus 12

3 Ülevaade probleemist **15**

4 ICP-2 spetsifikatsioon **16**

4.1 ICP-2 ülesanded ja temaga seotud OSI kihid 16

4.2 ICP-2 kaadri arhitektuuri kirjeldus 17

4.3 ICP-2 loogika ja kasutuslugude kirjeldus 18

4.3.1 Siinipöördus ja kaadri saatmise loogika kahe otspunkti vahel 18

4.3.2 Kaadri leviedastus 19

4.3.3 Suhtlus kõrgema prioriteediga kaadri saatmisel 20

4.3.4 Kui põhiline suhtlussiin on katki 20

4.3.5 Kokkupõrked siinil 21

4.3.6 Saatja katkestab saatmise 21

5 Töö käik ja tulemused **22**

5.0.7 ICP-2 ja teiste protokollide *overhead* võrdlus 23

5.1 Suurimad probleemid ja muutused arenduses 23

5.1.1 Vastuvõtja kinnituse saatmise loogika viga 24

5.1.2 Muutus pakettide puhverdamises 25

5.2 Testimine 26

6 Edasine arendus **29**

Kokkuvõte **30**

Summary **31**

Tänuavaldused **32**

Viited **33**

Lisa 1 - Lähtekood **35**

Lihtlitsents **36**

Lühendid

ESTCube - Elektrilise päikesepurje test satelliit (ing.k. *Electric Sail Test Cube*)

ICP - Sisemine suhtluse protokoll (ing.k. *Internal Communication Protocol*)

UART - Universaalne asünkroontransiiver (ing.k. *Universal Asynchronous Receiver/Transmitter*)

HDLC - Andmelüli kõrgjuhtimine, ISO standard (ISO 3309, ISO 4335, ISO 6159 , ISO 6256) (ing.k. *High-level Data Link Control*)

CAN - Kontrollervõrk, ISO standard (ISO 11898) (ing.k. *Controller Area Network*) protokoll

USB - Universaalne järjestiksiin (ing.k. *Universal Serial Bus*)

CSP - ing.k. *Cubesat Space Protocol*

EPS - Toitesüsteem (ing.k. *Eletrical Power Subsystem*)

CAM - Kaamerasüsteem (ing.k. *tether end mass imaging system*)

ST - Tähejälgija (ing.k. *Star Tracker*)

OBCS - Pardaarvuti (ing.k. *On-board Computing Saystem*)

COM - Suhtlussüsteem (ing.k. *Communication subsystem*)

AP - Liidestus port (ing.k. *Access Port*)

AOCS - Asendi ja orbiidi määramise ja juhtimise süsteem (ing.k. *Attitude Determination and Control Subsystem*)

PL - Kasulik last (ing.k. *Payload*)

SP - Külgpaneel (ing.k. *Side Panel*)

OSI - Avatud süsteemide sidumise arhitektuur (ing.k. *Open Systems Interconnection model*)

MQTT - Sõnumi järjekorra telemeetriline transport protokoll (ing.k. *Message Queuing Telemetry Transport*)

MAC - Meediumipöörduse juhtimine (ing.k. *Media Access Control*)

1 Sissejuhatus

Kevad 2013 lendas Maa orbiidile Eesti esimene satelliit *Electric Sail Test Cube* 1 (ESTCube-1), mis valmis Tartu Ülikooli eestvedamisel 2008. aastal alustatud Eesti tudengisatelliidi programmi raames [1]. ESTCube programmi jooksul ehitatud satelliitide suurem eesmärk on testida Pekka Janhuneni leiutatud elektrilist päikesepurje [2, 3]. ESTCube-1 lõpetas oma missiooni käesoleva aasta veebruaris. Tänaseks on alustatud järgmise projektiga, et saata kosmosesse kaks uut tudengisatelliiti: ESTCube-2 ja ESTCube-3 (vaata peatükki 2).

Satelliitide pardal on mitmeid alamsüsteeme, mis peavad missiooni kordaminekuks omavahel käsklusi ja andmeid vahetama. Seoses fundamentaalsete erinevustega ESTCube-1 ja tulevate satelliitide füüsilises kihis (vaata peatükki 2.2), ei ole võimalik ESTCube-1 pardal kasutatud satelliidisisest suhtlusprotokolli *Internal Communication Protocol* 1 (ICP-1) uutel satelliitidel kasutada. ESTCube-2 ja ESTCube-3 jaoks on vaja arendada uut suhtlusprotokolli, mis võimaldaks alamsüsteemidel minimaalse ressurssikuluga ühisel siinil käske ja andmeid vahetada. Hetkel tegeleb ESTCube meeskond ESTCube-2 alamsüsteemide prototüüpide arendamisega ning seetõttu vaatleme antud töö raames vaid ESTCube-2 ja selle alamsüsteemide vahelist suhtlust.

1.1 Töö eesmärk ja ülevaade

Töö üheks eesmärgiks on välja selgitada võimalikud andmelülikihi protokollid, mida saaks ESTCube-2 puhul kasutada. Vajadusel panna kirja ESTCube-2 alamsüsteemide vahelise uue andmelülikihi protokolli spetsifikatsioon ning alustada protokolli implementeerimisega ja testimisega.

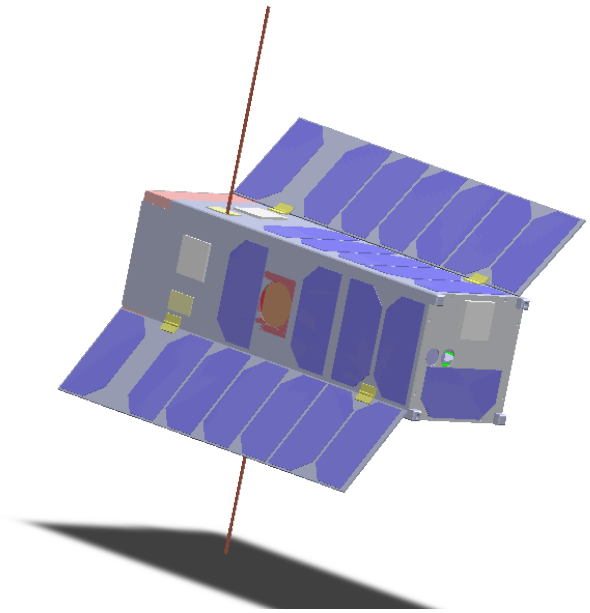
Teise osana tööst on luua testprogramm, mille abil on võimalik suhtlusprotokolli kasutada, testida ja mängida läbi võimalikke stsenaariumeid.

Töö autori panus:

- suhtlusprotokolli teegi spetsifikatsiooni loomine ja kirjeldamine,
- suhtlusprotokolli teegi ja testkeskkonna implementatsiooni alustamine,
- suhtlusprotokolli teegi ja testkeskkonna implementatsiooni testimine ja dokumenteerimine.

2 ESTCube-2

Tudengisateliidi programmi praeguse projekti eesmärgiks on ehitada satelliit nimega ESTCube-2 ja saata see Maa lähisorbiidile. Uus sateliit on eelmise, ESTCube-1 edasiarendus ning järgmise, ESTCube-3 osaline koopia. ESTCube-2 on kolmeühikuline satelliit (Joonis 2.1), mõõtmetega 340x100x100 mm ja kaaluga <4 kg, ja vastab kuupsatelliidi standardile [4].



Joonis 2.1: Illustratsioon kolmeühikulisest kuupsatelliidist (Paul Liias).

2.1 ESTCube-2 missioon

2.1.1 Elektriline päikesepuri

Elektriline päikesepuri on Soome Meteroloogia Instituudi teadlase Pekka Janhuneni innovaatiline tehnoloogia, mille abil on teoreetiliselt võimalik kosmosesõidukeid liigutada. Elektriline

päikesepuri kasutab jõuallikana päikesetuult ehk päikeselt lenduvaid prootoneid, ning liigutab kosmosesõidukeid sarnaselt tuule abil sõidukeid liigutava purjega. Päikese puri koosneb mitmest pikast ja peenikesest lõast, mida hoitakse elektronkahuriga kõrgel positiivsel potentsiaalil. Samanimelised laengud päikesetuules olevatel prootonitel ning purjel tõukavad kosmosesõidukid edasi [3, 5].

2.1.2 Missiooni eesmärk

ESTCube-2 missiooni eesmärk on testida kõikide alamsüsteemide korrasolekut ning käitumist Maa orbiidil enne ESTCube-3 missiooni[1].

2.2 ESTCube-2 kirjeldus

2.2.1 ESTCube-2 kasutatav riistvara kirjeldus

ESTCube-2 pardal on kokku üle 14-ne alamsüsteemi. Nende hulka kuuluvad *Eletrical Power Subsystem* (EPS), *tether end mass imaging system* (CAM), *Communication subsystem* (COM1 ja COM2), *Access Port* (AP), *Attitude and Orbit Control System* (AOCS), *Payload* (PL), *Onboard Computing System* (OBCS), *Star Tracker* (ST) ja kuus külgpaneeli (*Side Panel* ehk SP).

EPSi ülesandeks on voolu tootluse ja tarbivuse haldamine ning teiste alamsüsteemide varustamine vooluga. COM1 sidesüsteemi kasutab satelliit Maalt tulevate käskluste vastuvõtmiseks ning vastuste tagasi saatmiseks. COM2 on kõrgema kandesagedusega eksperimentaalne sidesüsteem. AP on liides satelliidi testimiseks ja hooldamiseks enne orbiidile saatmist. AOCS tegeleb satelliidi asendi ja orbiidi määramise ning muutmisega. PL on satelliidi päikesepurje teaduslikuks eksperimendiks vajalik last ning sellega seonduv. CAM tegeleb elektripurje lahtikerimise ja Maa pildistamisega. OBCS juhib teiste alamsüsteemide tegevust ning salvestab nende poolt tehtud mõõdiseid. Kuus SP-d aitavad koguda andmeid satelliidi külgpaneelide kohta.

Alamsüsteemidel on erisugused nõuded ja vajadused, seetõttu kasutatakse neil osaliselt erinevaid mikroprotsessoreid. Tabelis 2.1 on kirjeldatud ESTCube-2 peal kasutatavate mikroprot-

[1] ESTCube-3 missioon on viia läbi elektrilise päikesepurje eksperiment Maa magnetvälja mõjusfäärist piisavalt kaugel (90 000 km) [6], näiteks Kuu orbiidil.

sessorite põhilised parameetrid. Peamiseks riistvaraliseks piirajaks on alamsüsteemides kasutatavad väikese muutmäluga (kuni 2KiB) mikroprotsessorid.

Tabel 2.1: ESTCube-2 peal kasutatavate mikroprotsessorite põhilised parameetrid

	STM32F4xxxx [7]	MSP430FR5969 [8]	MSP430FR6989 [9]
Arhitektuur	ARM 32-bit Cortex-M4	16-Bit RISC	16-Bit RISC
Taktsagedus	168 MHz	16 MHz	16 MHz
Püsimälu	2 MiB	64 KiB	128 KiB
Muutmälu	256+4 KiB	2 KiB	2 KiB
Riistvaraline CRC16	Ei	Jah	Jah
Riistvaraline CRC32	Jah	Ei	Ei
Toetab riistvaras CAN	Jah	Ei	Ei
Toetab riistvaras USB	Jah	Ei	Ei

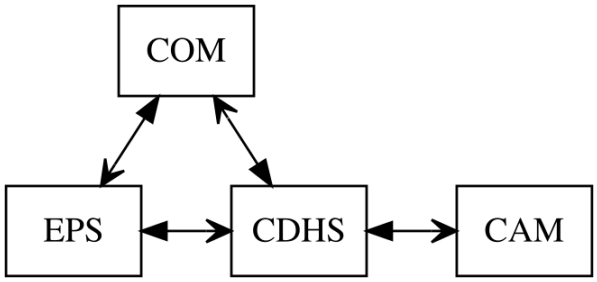
2.2.2 ESTCube-2 võrgutopoloogia kirjeldus

Satelliidil ESTCube-1 kasutati alamsüsteemide vahel kakspunkt[2] *Universal Asynchronous Receiver/Transmitter* (UART) ühendustest koosnevat detsentraliseeritud silmusvõrk võrgutopoloogiat (Joonis 2.2). Sellest tulenevalt oli tarvis tegeleda alamsüsteemide vahelise liikluse marsruutimisega. ESTCube-1 marsruutimine ei olnud küll klassikalises mõistes marsruutimise algoritm, kuna kõik marsruudid olid staatiliselt alamsüsteemide mällu salvestatud [10]. Olenemata staatilistest marsruutimistabelitest pidid alamsüsteemid kulutama eraldi aega pakettide vahendamisele ning vajaduse korral alternatiivsete marsruutide leidmisele.

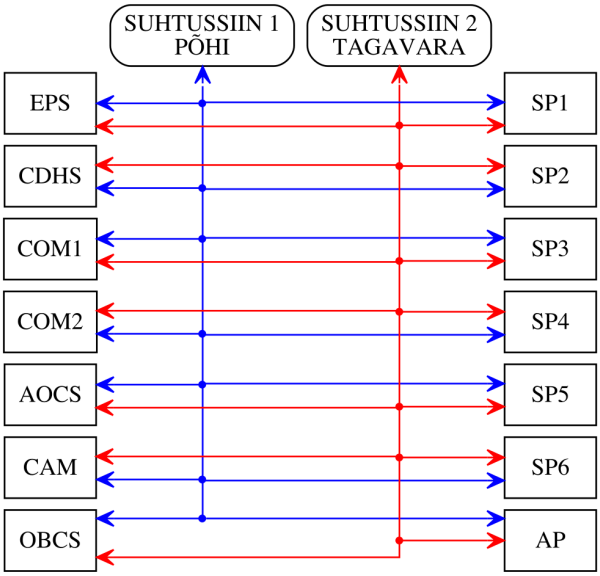
Pakettide marsruutimise vajaduse vältimiseks ning satelliidi võrgutopoloogia lihtsustamiseks planeeritakse alates ESTCube-2 kasutada kahte ühist suhtlussiini, millest üks on liiasuse tagamiseks (Joonis 2.3).

Jagatud suhtlussiini kasutamisel on esmatähtis lahendada siinipöörduse haldus. Vigane siinipöörduse haldamine võib põhjustada kaadrite kokkupõrkeid ja siini lukustumist teadmata ajaks.

[2]Kakspunkt ehk *point-to-point* on ainult kahe otspunktiga ühendusliik.



Joonis 2.2: ESTCube-1 alamsüsteemide vaheline võrgutopoloogia.



Joonis 2.3: ESTCube-2 alamsüsteemide vaheline võrgutopoloogia.

2.2.3 ESTCube-2 suhtlussiini kirjeldus

Tabelis 2.2 on välja toodud mitmete laialt levinud suhtlussiinide põhilised parameetrid. ESTCube-2 hulkpunkt topoloogia tõttu ei ole RS232 standardi kasutamine võimalik. Kuigi I^2C on väikeste kuupsatelliitide suhtlussiiniks popularne valik [11–13], esineb selle puhul tihti tõrkeid. Kui üks või enam alamsüsteemi lülitatakse edastatava kaadri ajal välja, võib sõltuvalt mirkoprotsessorite konfiguratsioonist I^2C siin lukustuda hõivatud olekusse [10]. Teised levinumad suhtlussiinid on *Universal Serial Bus* (USB) ja *Controller Area Network* (CAN). Kuigi USB ja CAN on hea läbilaskevõimega, ei sobi neist kumbki ESTCube-2 alamsüsteemide vaheliseks ühenduseks. USB-l puudub mitme ülemaga võrkude jaoks vajalik võimekus. Mitme ülemaga võrku kasutatakse ESTCube-2 tagamaks kogu võrgu ja suhtluse liiasuse ning vältimaks *sing-*

le point of failure probleemi. Muuhulgas tarbib USB liigselt voolu. ESTCube-2 töös esineb suurte andmemahtude edastamist, milleks ei ole CAN ettenähtud. Lisaks on selle protokolli keerukusest tingitud vajaminev muutmälu hulk osade satelliidi alamsüsteemide mirkoprotsessorite jaoks liiga suur (Tabel 2.1). Lisaks ei toetata riistvaraliselt MSP430FR mikroprotsessorite peal ei USB ega CAN siini (Tabel 2.1).

Eelnevat arvesse võttes kasutatakse ESTCube-2 suhtlussiinide põhiosana kahte jagatud pool-dupleks RS485 siini. Igal alamsüsteemil on RS485 siini kasutamiseks transiiver. Transiiver tegeleb siinilt andmete mahalugemisega ja saatmisega. Vastuvõetud baitide puhvri täitumisel annab transiiver alamsüsteemi mikroprotsessorile sellest teada. Alamsüsteem suhtleb RS485 transiiveriga üle UART-i [14].

Lisaks on ICP-2 jaoks mõlema RS485 siini juures kaks jagatud *active-low* signaaliliini: ACCESS ja SHUDUP.

ACCESS liini kasutatakse suhtlussiini oleku kontrollimiseks ja saatele minnes hõivamiseks ning võimaldab minimaliseerida kokkupõrkeid ja optimiseerida energiatarvet. Selline signaaliliin likvideerib kaadri alguse ja lõpu eraldusbaitide kasutamisvajaduse, tänu millele ei pea tegelema *byte escaping*-ga. Kõik see vähendab kaadri päist.

SHUDUP liini kasutatakse suhtlussiini vabastamiseks suvalisel ajahetkel. Juhul kui suhtlussiin on hõivatud ning see on vaja jõuga vabastada, saab poolelioleva suhtluse katkestada SHUDUP liiniga. Siinil ei toimu suhtlust vähemalt seni kaua kuni SHUDUP liin on aktiivne.

Tervikliku suhtlussiini moodustavad üks RS485, üks ACCESS ja üks SHUDUP liin. Teine samasugune komplekt on liiasuse tagamiseks.

Tabel 2.2: Erinevate suhtlussiini standardite võrdlus

	RS232	RS485	USB	CAN	I^2C
Diferentsiaalne	Ei	Jah	Jah	Jah	Ei
Ülemate maksimum	1	32	1	>1	>1
Alluvate maksimum	1	32	127	>1	>1
Andmeedastus	pooldupleks	pooldupleks	pooldupleks	pooldupleks	pooldupleks
reziim	täisdupleks				
Võrgutopoloogia	kakspunkt	hulkpunkt	hulkpunkt	hulkpunkt	hulkpunkt
Maks. kiirus	20 kbps	35 Mbps	480 Mbps	1 Mbps	400 kbps

3 Ülevaade probleemist

Käesolevas peatükis kirjeldatakse lühidalt ICP-1 kasutuse käigus selgunud probleeme ja arutletakse olemasolevate protokollide kasutatavuse üle ESTCube-2 pardal.

Seoses põhimõtteliste erinevustega võrgutopoloogias ESTCube-1 ja 2 vahel, ei ole võimalik ICP-1 ilma põhjaliku ümberkohandamiseta ESTCube-2 peal kasutada 2.2.2.

Lisaks ICP-1 protokollile kaaluti ESTCube-2 pardal ka teiste laialt levinud suhtlusprotokollide nagu Modbusi ja *Cubesat Space Protocol*-i (CSP) kasutamist. ModBus välistati sarnaselt USB-ga, kuna ei toeta mitme ülemaga võrke [15]. CSP on küll kuupsatelliitide võrgukihi protokoll ning disainitud võttes arvesse väiksemaid mikroprotsessoried. Kuid kuna tegu on kõrgema kihi protokolliga, siis madalamatel kihtidel toetub see ESTCube-2 mittesobivatele teistele standarditele nagu CAN, I^2C, RS232. Kuna RS232 ja RS485 liidestus mikroprotsessoriga on samasugune, võiks teoreetiliselt CSP-d kasutada ka RS485 peal. Kuid kuna hetkel toetub CSP kaadrite kokkupõrgete vältimiseks kas andmelülikihile (CAN) või ei toeta kokkupõrkeid üldse (RS232), siis peaks RS485 kasutamisel tegelema eraldi vahekihi arendusega vältimaks kaadrite kokkupõrkeid.

Antud töö raames vaadeldud laialt levinud suhtlusprotokollid ei sobi ESTCube-2 alamsüsteemide vaheliseks suhtluseks. Seega otsustati ESTCube-2 jaoks väljatöötada uus minimalistlik suhtlusprotokoll.

4 ICP-2 spetsifikatsioon

Käesolevas peatükis kirjeldatakse ICP-2 suhtlus protokolli, tema ülesandied, arhitektuuri, siinipöörduse loogikat ning erinevaid kasutuslugusid näitlike alamsüsteemide A, B ja C puhul.

4.1 ICP-2 ülesanded ja temaga seotud OSI kihid

ICP-2 eesmärgiks on tagada mitteajakriitiliste andmete vahetus ESTCube-2 alamsüsteemide vahel. Normaalseks andmete saatmise ajaks võib lugeda 100(+-50) ms. ESTCube-2 alamsüsteemid saadavad üksteisele erinevate suurustega andmeid: alates ühebaidistest käskudest kuni mitme megabaidiste andmeplokkideni. Seetõttu peab ICP-2 olema suuteline saatma ka suuremahulisi andmepakette. Näiteks edastatakse CAM alamsüsteemi poolt tehtud foto 64 KiB tükkidena COM alamsüsteemile, mis saadab need tükid Maa peal olevale juhtimiskeskusele.

ICP-2 on *Open Systems Interconnection model* (OSI) mudeli andmelülikihi protokoll, mille ülesandeks on tagada universaalne liidestumine füüsilise ja kõrgemate kihtide vahel. ICP-2 ülesanded:

- kõrgemalt kihilt saadud andmepakettide ettevalmistamisega kaadriteks enne füüsilisele kihile saatmist,
- siinipöörduse juhtimisega,
- vastuvõetud kaadrites esinevate vigade tuvastamisega ja haldusega,
- korrektsest kaadrist andmepaketi eraldamisega,
- kaadrist eraldatud andmepaketi kõrgematele kihtidele edastamisega.

4.2 ICP-2 kaadri arhitektuuri kirjeldus

ICP-2 kaader koosneb nelja baidi suurusest päisest, kasulikust lastist ja kahebaidisest kontrollsummast (Joonis 4.1).

Päise esimene bait on adressaadi MAC aadress. Selle alusel otsutab alamsüsteem A, kellele vastuvõetud kaader on mõeldud ning vajadusel jätkab järgnevate baitide salvestamist.

Päise teine bait on saatja MAC aadress. Seda infot on vaja kaadri saatja identiteedi määramiseks ja kontrollimiseks. Kui alamsüsteem A saadab alamsüsteemile B päringu, siis alamsüsteem B teab saata vastust päringu esitanud alamsüsteemile A. Omakorda saab alamsüsteem A kontrollida vastuse saatjat.

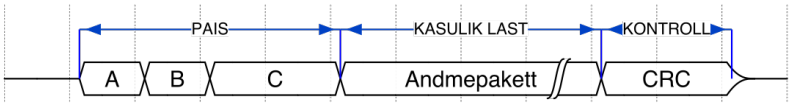
Päise kaks viimast baiti iseloomustavad kasuliku lasti pikkust. See arv on võrde baitide arvuga, mis on kasulikus lastis lisaks kohustuslikule andmemahule. Kasuliku lasti pikkus on oluline kaadriga kaasa panna, et vastuvõttev alamsüsteem teaks, millal kõik edastatavad baidid on temani jõudnud.

Kaadri kasuliku lasti osa sisaldab andmeid, mida saatja vastuvõtjale edastada tahab. Kasuliku lasti suurus valitakse vastavalt saadetavate andmete hulgale. Kuna kaadri päises oleva kahe kasuliku lasti suurust iseloomustava kahe baidiga on võimalik loendada vaid kuni 65535 baidini, siis on kasuliku lasti miinimumpikkus üks bait ning maksimaalne pikkus on 64 KiB.

Kaadri kaks viimast baiti on kontrollsumma, mille abil vastuvõttev alamsüsteem kontrollib kaadri terviklikkust. Kontrollsumma arvutamisel võetakse arvesse kõiki teisi kaadri osi alates adressaadist kuni viimase kasuliku lasti baidini. Kontrollsumma lisatakse kaadrile alamsüsteemi A poolt vahetult enne kaadri saatmist. Vastuvõttev alamsüsteem B arvutab samuti kontrollsumma, järgides sama algoritmi, ning võrdleb seda kaadriga kaasas olnud kontrollsummaga. Kui kontrollsummad võrduvad on kaader korrektselt kohale jõudnud.

Kontrollsumma arvutatakse vastavalt CRC16-CCITT algoritmile [16]. Antud algoritm valiti seetõttu, et satelliidil kasutatavad TI MSP430 seeria mirkoprotsessorid võimaldavad CRC16 kontrollsummat arvutada riistvara tasemel [8, 9]. Suurematel ja kiirematel mirkoprotsessoritel pole probleemi kontrollsummat tarkvaraliselt arvutada.

Kokkuvõttes on ICP-2 kaadris peale kasuliku lasti kuus lisabaiti ning suurimaks kasulikuks lastiks võib olla 64 KiB.



A - Adressaadi MAC; B - Saatja MAC; C - Kasuliku lasti pikkus;

Joonis 4.1: ICP-2 kaader.

4.3 ICP-2 loogika ja kasutuslugude kirjeldus

Käesolevas alapeatükis kirjeldatakse täpsemalt ICP-2-te kasutavate süsteemide saatmisel ja vastuvõtmisel tehtavaid tegevusi nii tavatingimustel kui ka erandjuhtudel.

4.3.1 Siinipöördus ja kaadri saatmise loogika kahe otspunkti vahel

Joonisel 4.2 on kujutatud üldist suhtlussiini pöörduse aegdiagrammi. Joonisel on märgitud RS485 siin, signaalid ACCESS ja SHUDUP ning ülemises ääres olevate baitide järjekorranumbrid.

Oletame, et alamsüsteem A tahab saata alamsüsteemile B andmeid. Kõigepealt kontrollib alamsüsteem A, kas suhtlussiin on vaba. Selleks tuleb määrata kas ACCESS ja SHUDUP liinid on kõrges olekus ehk inaktiivsed (Joonis 4.2 takt 0). Kui suhtussiin ei ole vaba, lõpetab alamsüsteem tegevuse ja annab kõrgemale programmile teada, et kaadri saatmine ei õnnestunud. Kui suhtlussiin on vaba hõivab alamsüsteem A siini, muutes ACCESS liini aktiivseks (Joonis 4.2 takt 1). Seejärel ootab alamsüsteem A ettemääratud aja t_1[1], mille möödudes alustab kaadri saatmist.

Kaadri saatmise aeg t_2[2] sõltub kaadri pikkusest. Selle aja möödumisel peab olema kaader edastatud.

[1] Aeg t_1 on igal alamsüsteemil eraldi konfigureeritav.

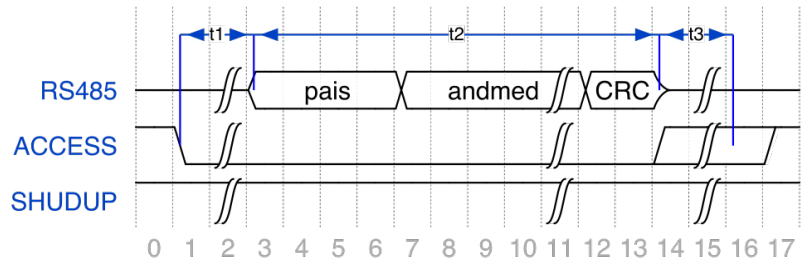
[2] Aeg t_2 sõltub kaadri pikkusest ning suurimaks lubatud väärtuseks on maksimaalse lubatud kaadri edastusaeg pluss X% lisa. Suurus X on protokollis konfigureeritav suurus.

Pärast kaadri päise kohale jõudmist otsustab alamsüsteem B, kas kaader on mõeldud talle. Kui kaader on adresseeritud talle, siis lülitab alamsüsteem B oma ACCESS liini viigu väljundiks ning hakkab ka ACCESS liini aktiivsena hoidma. Kui ei ole, ei tee alamsüsteem B midagi.

Vahetult pärast kaadri lõppu (Joonis 4.2 takt 14) kontrollib kaadri saatnud alamsüsteem A, kas keegi hoiab ACCESS liini aktiivsena. Kui signaaliliin on inaktiivne eeldab alamsüsteem A, et alamsüsteem B ei saanud saadetud kaadrit kätte ning annab sellest programmile teada. Kui ACCESS on aktiivne tähendab see, et alamsüsteem B tegeleb kaadri töötlemisega. Sellest hetkest hakkab alamsüsteem A kuulama ACCESS liini peal tõusvat fronti kinnituseks, et alamsüsteem B on kaadri kätte saanud. Pärast kaadri edastust ootab alamsüsteem A vähemalt ajahulga t_3[3] ning kontrollib, kas alamsüsteem B jätkuvalt hoiab ACCESS liini aktiivsena, mis tähendab kaadri kontrollsummade ebavõrdsust. Sellisel juhul visati kaader minema[4].

Kui korrekte kaader on kohale jõudnud (ning edastatud juba kõrgemale programmile), muudab alamsüsteem B enne ajahetke t_3 möödumist ACCESS liini inaktiivseks.

Olenemeta olukorrast lülitab alamsüsteem B ACCESS liini inaktiivseks alates 15. taktist joonisel 4.2. See vabastab suhtlussiin teistele alamsüsteemidele.



Joonis 4.2: Siinipöördus aegdiagramm ühe kaadri saatmisel.

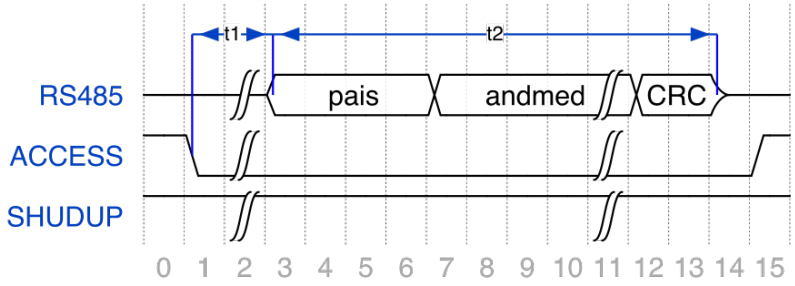
4.3.2 Kaadri leviedastus

Leviedastus töötab sarnaselt nagu kaadri edastus kahe otspunkti vahel. Küll aga ei toimu kaadri kättesaamise kontrolli. Saatev alamsüsteem lülitab ACCESS liini kohe pärast kaadri saatmist

[3] Aeg t_3 on andmete hulgast sõltuv aeg, sest erinevate pikkustega andmete korral kulub kontrollsumma arvutamiseks erinev aeg.

[4] Põhjused miks kinnitust tähistab inaktiivne ACCESS / tõusev front ACCESS liinil tulevad välja alapeatükis 4.3.5.

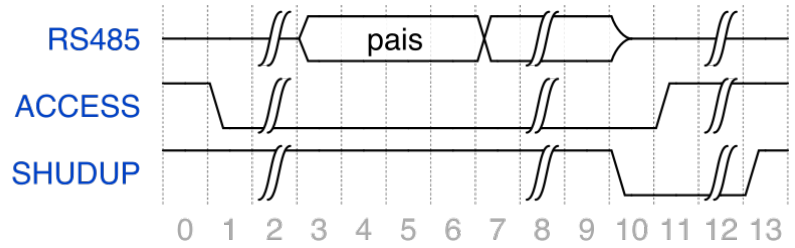
inaktiivseks ning ei jää ootama kinnitust (Joonis 4.3 takt 15).



Joonis 4.3: Siinipöördus aegdiagramm leviedastuse kaadri saatmisel.

4.3.3 Suhtlus kõrgema prioriteediga kaadri saatmisel

Kui alamsüsteem A tahab edastada kõrgema prioriteediga kaadrit põhisuhtlussiini peal, aga see on hõivatud, siis kasutab ta suhtlussiini vabastamiseks SHUDUP liini (Joonis 4.4 takt 10). SHUDUP liin aktiveeritakse, käimasolev suhtlus lõpetatakse ning kõrgema prioriteediga kaader saadetakse välja. Kui alamsüsteem A on kõik kõrgema prioriteediga kaadrid välja saatnud laseb ta SHUDUP signaali lahti ja jätkub tavatingimustel siinipöördus (Joonis 4.4 takt 13).



Joonis 4.4: Siinipöördus aegdiagramm kõrge prioriteediga kaadri saatmisel.

4.3.4 Kui põhiline suhtlussiin on katki

Tuvastades, et siin on "katki"[5], lülitab alamsüsteem A ümber tagavara suhtlussiini peale ning saadab välja leviedastuskaadri sisuga: "Põhisiin on hetkel katki, kasutame suhtluseks tagavara

[5]Suhtlussiin on "katki" kui siini ei ole korduvalt võimalik tavatingimustel kasutada. Näiteks kui alamsüsteem A tagastab N korda järjest veateate, et siin on hõivatud ning ei vabane ka SHUDUP liini aktiveerides, või kui alamsüsteem B ei tagasta M korda järjest alamsüsteemi A poolt saadetud kaadrile kinnitust. Suurused N ja M on protokollis konfigureeritavad suurused.

siini seni, kuni põhisiini viga kõrvaldatakse". Selle peale lülitavad ka kõik teised alamsüsteemid põhisuhtluse ümber tagavara siini peale.

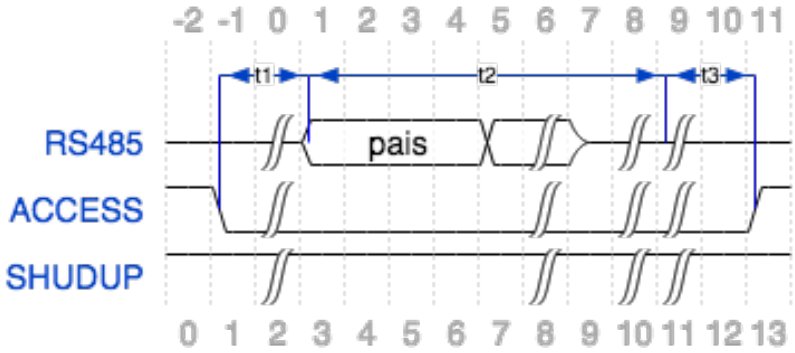
4.3.5 Kokkupõrked siinil

Kahe alamsüsteemi samaaegsele saatele minemisel toimub edastatavate kaadrite kokkupõrge ning vastuvõtjatel ei ole võimalik paralleelselt saadetud kaadreid terviklikuna siinilt maha lugeda. Vastuvõtvad alamsüsteemid loevad maha vigased kaadrid, kuid ei teavita saatjaid kaadri kohale jõudmisest. Saatmise lõpust t_3 möödudes vabastavad mõlemad saatjad ise ACCESS liini. Seetõttu eraldi kokkupõrgete tuvastamisega ja käsitlemisega tegelema ei pea.

4.3.6 Saatja katkestab saatmise

Alamsüsteem A võib poole kaadri pealt edastamise lõpetada ning seega vabastab omalt poolt ACCESS liini (Joonis 4.5). Kui ükski teine alamsüsteem ei ole saadetava kaadri vastuvõtjaks ennast tunnistanud ja ei hoia ACCESS liini aktiivsena, vabaneb suhtlussiin teistele alamsüsteemidele.

Kui aga mõni alamsüsteem on eelnevalt tuvastanud, et kaader on talle mõeldud, toimib vastuvõttev alamsüsteem nagu eelpool kirjeldatud alapeatükis 4.3.1.



Joonis 4.5: Siinipöördus aegdiagramm saatja poolt katkestatud kaadrist.

5 Töö käik ja tulemused

Käesolevas peatükis on kirjeldatud ICP-2 teegi arendust, arenduse käigus tekkinud probleeme, arendatud testprogrammi ning programmide ja teegi testimist.

Algselt võeti ICP-2 arenduse aluseks ICP-1 lähtekood, et mõista täpsemalt kuidas oli suhtlusprotokoll lahendatud ESTCube-1 alamsüsteemidel. Esimeste edusammudena suudeti testprogrammis teostada kahe programmi andmevahetust. Esialgu saadeti ainult üksikuid baite, hiljem baitide massiive.

Pärast põhjalikku ICP-1 loogika uurimist ning ESTCube-2 meeskonnaga arutlemist jõuti järeldusele, et ICP-2 teegi jaoks ei ole ICP-1 loogikast palju võimalik uuesti kasutada. Seega kirjutati programmi loogika ning *Application Programming Interface* (API) arhitektuur peaaegu täielikult puhtalt lehelt. Kirjutatud teegis on mõningad sarnasused ICP-1ga, nagu näiteks mõndade API funktsioonide ja failide nimed ning teegi integreerimise protseduur.

ICP-2 arenduse käigus läbiti järgmised sammud:

1. tutvuti ICP-1 protokolliga,
2. loodi testkeskkond alamsüsteemide ja siini simuleerimiseks,
3. implementeeriti lihtsamad funktsioonid ja andmemudelid,
4. paralleelselt arendati ja testiti siinpöörduse loogikat,
5. implementeeriti saatja poolne kaadrite komplekteerimise loogika,
6. implementeeriti vastuvõtja poolne kaadrite töötlemise loogika,
7. testiti võimalike stsenaariumeid.

Muus osas implementeeriti protokolli API vastavalt spetsifikatsioonile.

5.0.7 ICP-2 ja teiste protokollide *overhead* võrdlus

Järgmise valemiga on võimalik arvutada kui suur osa kogu saadetavast kaadrist on *overhead*.

$$\begin{cases} n = \text{asendatud baidid} + \text{päise suurus} * \frac{\text{andmete suurus}}{\text{maksimaalne kasulik last}} \\ \frac{n}{\text{andmete suurus}+n} = \textit{overhead}\text{-i } \% \end{cases} \tag{5.1}$$

Tulenevalt ESTCube-1 võrgutopoloogiast, kasutati alamsüsteemide vaheliseks suhtlemiseks suhtlusprotokolli ICP-1, mis oli ühtlasi nii andmelüli- kui ka võrgukihi protokoll. ICP-1 kasutas *High-level Data Link Control*i (HDLC) eeskujul kaadrite kapseldamist 0x7E baitidega kaadrite teineteisest eraldamiseks mälus. Ühe kaadri sees esinevate 0x7E ja 0x7D baitidega tehti välistav-või tehe 0x20 baidiga ning tulemuse ette lisati 0x7D [10]. Kirjeldatud algoritmi nimetatakse *byte escapeing*-uks. Selliselt on võimalik kaadreid teineteisest lihtsasti eraldada, kuid puudusena kasvab kaadri mittekasulike baitide arv (*overhead*) ja puhverdamiseks vajalik mälumaht. ICP-1 päise suurus oli 9 baiti ning maksimaalne kasulik last 255 baiti.

64 KiB juhuslike andmete edastamisel on umbes 500 baiti 0x7E või 0x7D[1]. Kasutades valemit 5.1 saame arvutada, et keskmiselt on ICP-1 *overhead* 64 KiB andmete edastamisel 4.1%. Halvimal juhul on kõik andmetes sisalduvad baidid kas 0x7E või 0x7D ning seega on *overhead* umbes 50.87%. Võrdluseks võib tuua, et CAN protokolli *overhead* on umbes 50% [17].

ICP-2 kaadris on peale kasuliku lasti kuus lisabaiti ning suurimaks kasulikuks lastiks võib olla 64 KiB. Seega kasutades valemit 5.1 saame arvutada ICP-2 teoreetilise *overhead*i 64 KiB andmete edastamisel. Tulemuseks on 0.00915%, mis on umbes 450 korda vähem kui ICP-1 ning 5460 korda vähem kui CAN protokollil.

5.1 Suurimad probleemid ja muutused arenduses

Käesolevas alapeatükis kirjeldadakse ICP-2 teegi arenduses ette tulnud suurimaid muutuseid, probleeme ning nende lahendusi.

[1] Leitud kasutades programmi, mis genereeris suvaliselt 64 KiB andmeid ning luges kõik 0x7E ja 0x7D kokku. Katse tehti 10 korda järjest ning tulemused keskmistati.

5.1.1 Vastuvõtja kinnituse saatmise loogika viga

Esialgse visiooni järgi pidi alamsüsteem B korrektse kaadri kinnituse alamsüsteemile A saatmiseks ACCESS signaali liini lihtsalt lahti laskma. Pärast ACCESS liini tõusva frondi tuvastamist sai alamsüsteem A aru, et korretne kaader on kohale jõudnud. Juhul kui alamsüsteem B ei oleks kinnitust saatnud, siis oleks alamsüsteem A pärast t_3 möödumist ACCESS liini ise lahti lasknud.

Arenduse ja spetsifikatsiooni kirja panemise käigus ilmnes antud loogikas viga. Kuna ACCESS signaaliliin on *active-low open-drain*, siis piisab vaid ühest alamsüsteemist, et hoida ACCESS signaaliliini aktiivsena (madalal). Mis omakorda tähendab, et on stsenaariumeid, kus ACCESS signaali pöördus ei ole defineeritud:

1. Kohe pärast kaadri saatmist lülitab alamsüsteem A oma ACCESS liini küljes oleva viigu sisendiks ning jääb kinnitust ootama (Joonisel 5.1 takt 14). Oletame, et alamsüsteem B ei tuvasta, et kaader on mõeldud talle ning ei hakka ACCESS liini juhtima. Sellest hetkest alates ei juhi ACCESS liini enam keegi. *Active-low open-drain* signaalide küljes on *pull-up* takistid, mis juhul kui keegi signaali ei juhi tõmbavad ise signaali kõrgele nivootasemele. Seeläbi hakkab ACCESS vaikselt kõrgeks triivima[2]. See võimaldab alamsüsteemil A tuvastada ACCESS liini tõusvat fronti enne t_3 täitumist. Teisisõnu, alamsüsteem A arvab, et alamsüsteem B saatis talle kinnituse, kuigi ta seda ei teinud.

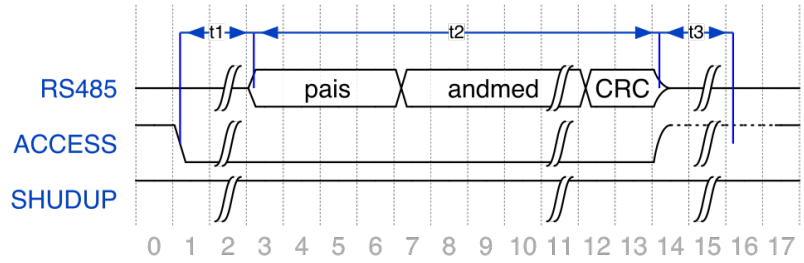
2. Kohe pärast kaadri saatmist lülitab alamsüsteem A oma ACCESS liini küljes oleva viigu sisendiks ning jääb kinnitust ootama. Alamsüsteem B tuvastab, et kaader on mõeldud talle, hakkab ACCESS liini juhtima ning aktiivsena hoidma. Oletame, et alamsüsteem B kontrollib kaadri sisu ning tuvastab, et kaadri kontrollsummad ei ole võrdsed ning seega ei muuda ACCESS liini inaktiivseks (sest see saadaks alamüsteemile A kinnituse korrektsest kaadrist). Kui alamsüsteem A ei ole aja möödudes kinnitust saanud hakkab ta ise uuesti ACCESS liini juhtima ning üritab seda lülitada inaktiivseks. Kuna alamsüsteem B hoiab ACCESS liini aktiivsena ning tal pole ühtegi võimalust kontrollida kas alamsüsteem A on lõpetanud kinnituse ootamise või mitte, siis jääb ikka ACCESS aktiivseks ning kogu siin hõivatuks (Joonisel 5.2 takt 16).

3. Alamsüsteemid A ja B satuvad korraga saatele. Oletame, et A üritab saata kaadrit alam-

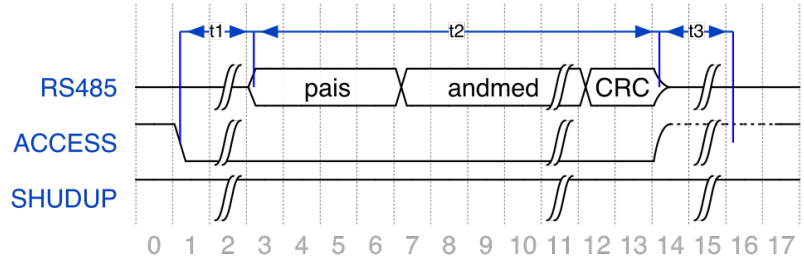
[2]Selle muutuse kiirus sõltub erinevate komponentide mahtuvusest.

süsteemile C ning B saadab välja leviedastuse. Sõltumata sellest, kas alamsüsteem C identifitseerib ennast kui kaadri aadressaati, ei ole saadetava kaadri kontrollsummad võrdsed ning alamsüsteem C ei saaks saata kinnitust. Küll aga on võimalik, et alamsüsteem A tuvastab kinnituse ning arvab, et C sai kaadri korrektselt kätte. Kui A lõpetab saatmise varem kui B ning jääb kinnitust ootama, siis pärast B leviedastuse lõppu laseb B ACCESS liini koheselt lahti, sest leviedastuse kaadrile kinnitust ei oodata. Alamsüsteem A tuvastab nüüd ACCESS liini tõusva frondi ning eeldab, et tegemist on kaadri kinnitusega alamsüsteemilt C.

Eelnevalt loetletud põhjuste tõttu sai ACCESS liini pöördumine ja kaadrite kinnituste saatmise loogika ümber tehtud (kirjeldatud alapeatükis 4.3).



Joonis 5.1: Siinipöördus aegdiagramm. Vale kinnituse saamine.



Joonis 5.2: Siinipöördus aegdiagramm. Siini lukustumine.

5.1.2 Muutus pakettide puhverdamises

Algselt prooviti ICP-1 eeskujul integreerida ICP-2 protokolli väljasaatmist ootavate ning vastuvõetud pakettide puhverdamise võimekust. Saates pandi pakett esmalt saatmisjärjekorda

ning määrati talle järjekorra number. Saatmisjärjekorrast võeti ükshaaval paketid ning vormistati kaadriks. Vormistatud kaadrite väljasaatmine toimus põhiprogrammi poolt API funktsiooni väljakutsumisel. Juhul kui järjekorras olevate pakettide saatmine ei õnnestunud, prooviti järgmisel API funktsiooni väljakutsumisel uuesti. Paketi edukal saatmisel edastati põhiprogrammile paketi järjekorranumber. Analoogiliselt pandi vastuvõetud kaadritest saadud paketid vastuvõtu järjekorda. Selline lahendus oli vajalik, kuna ESCube-1 puhul võis alamsüsteem saada korraga kahelt saatval alamsüsteemilt kaadrid, mida tuli kanda erinevatesse puhvritesse. Pärast vastuvõetud kaadrite korrektseks tunnistamist pandi need kokku ühisesse järjekorda ning anti põhiprogrammile teada. Seejärel sai põhiprogramm talle endale sobival ajal sealt paketid välja lugeda. Seega on tegemist asünkroonse lahendusega. Asünkroonsus võimaldab keerukate ja paindlike lahenduste implementeerimist, kuid samas võib tekitada ka muresid. Näiteks samaaegsest mälupöördusest tekkivad vigased andmed.

ESTCube-2 suhtlussiini disain ning kaadrite vahetamise loogika ei toeta asünkroonsust. Pole võimalik, et üks alamüsteem võtab korraga vastu rohkem kui ühte kaadrit. Samuti on siini hõivamisega kaadritöötluse ajal tagatud kaadri täielik töötlemine enne järgmise kaadri vastuvõttu.

Eelolevat arvesse võttes otsustati kaotada saadetavate ja vastuvõetavate pakettide järjekorrad.

5.2 Testimine

Käesolevas alapeatükis kirjeldatakse ICP-2 teegi arenduse käigus väljakujunenud testprogrammi ning selle abil läbi viidud teste.

ICP-2 teegi arendust alustades oli ESTCube-2 alamsüsteemide riistvara alles prototüüpimise faasis. Seetõttu oli tarvis simuleerida alamsüsteeme arvutis, kus ühtlasi on arenduse tsükkli aeg palju lühem kui arendades riistvaral.

Testprogramm implementeeriti C programmeerimiskeeles. Testprogramm kasutab suhtlussiini simuleerimiseks *Message Queuing Telemetry Transport* (MQTT) protokolli. MQTT on TCP/IP peale ehitatud *publish-subscribe* mudelil põhinev protokoll, mis võimaldab muuhulgas alamsüsteeme simuleerivatel programmidel suhelda teineteisega ka interneti vahendusel [18].

MQTT kasutamise puuduseks on suur latentsus, mis põhjustas arenduse jooksul siinipöörduse probleeme. Sõnum ühe alamsüsteemi siini hõivamise kohta jõudis teistele alamsüsteemidele umbes 0.5-2 millisekundiga. Selle aja jooksul jõudsid ka teised alamsüsteemid siini hõivamise soovist teada anda. Probleemi vältimiseks simuleeriti siin eraldi C programmis, mis tegeles *first-come, first-served*[3] põhimõttel alamsüsteemidele siini reserveerimisega ning vastavate simuleeritud signaaliliinide kontrollimisega.

Lisaks kirjutati juhtprogramm, millega juhiti testprogramme, suhtlussiini ning võimalikke stsenaariumeid. Juhtprogramm võimaldab saata simuleeritud siinile ja alamsüsteemide programmidele erinevaid korraldusi üle MQTT. Näited võimalikest käskudest:

- sätete muutmine: *[eps / cam / ... / all] set [sys / err / warn / info / debug / all] [true / false];*,
- paketi saatmine: *[eps / cam / ... / all] ping [eps / cam / ...];*,
- algsätete taastamine: *[eps / cam / ... / all] reset;*,
- programmi väljundi puhastamine: *[eps / cam / ... / all] clean;*,
- ajaarvamise sünkroniseerimine (nullimine): *[eps / cam / ... / all] sync;*.

"Set" käsuga on võimalik programmide erinevaid väljundeid sisse ja välja lülitada. Analoogiliselt saab muuta programmide konfiguratsiooni globaalseid muutujaid. Käsuga *"ping"* ütleme alamsüsteemile, et ta saadaks viimase parameetrina antud alamsüsteemile ping paketi. *"Reset"* käsk taastab programmi sätted, ilma et peaks programmi uuesti kompileerima või taaskäivitama. *"Clean"* käsib programmidel väljundisse panna 10 reavahetust, et puhastada väljundi ekraan. *"Sync"* käsuga nullivad ning seeläbi sünkroniseerivad kõik vastuvõtjad oma ajaarvamise kella.

Arenduse jooksul viidi läbi põhilised funktsionaalsed testid. Alamsüsteemid on võimelised:

- saatma teistele alamsüsteemidele pakette,
- saatma leviedastuse pakette,
- pakette vastu võtma,

[3] *first-come, first-served* on klient programmide päringute teeninduspoliitika, kus teenindatakse päringuid nende saabumise järjekorras, arvestamata muid parameetreid või eelistusi.

- arvutama ning võrdlema kaadrite kontrollsummat,
- tuvastama vigast kaadrit,
- tegelema siinipöördusega.

6 Edasine arendus

ICP-2 implementatsioon ei ole veel lõplik. Töö autori plaan on edasiste õpingute käigus implementeerida hetkel puuduolev spetsifikatsioonis kirjeldatud funktsionaalsus. Näiteks:

- tagavara siini kasutamine vastavalt spetsifikatsioonile,
- SHUDUP signaaliliini loogika.

Viimasena tuleb ICP-2 testida reaalsel ESTCube-2 riistvaral ning vajadusel täiendada protokolli spetsifikatsiooni ja implementatsiooni.

Kokkuvõte

Käesoleva töö raames anti ülevaade tulevase ESTCube-2 alamsüsteemide vahelise suhtlusega seotud riistvarast. Arutleti võimalike standardsete andmelülikihi protokollide kasutamise üle, mille tulemusena leiti, et olemasolevate protokollide integreerimine ESTCube-2 puhul on problemaatiline seoses varasemate riistvaraliste otsustega. Sellest tulenevalt alustati satelliidi alamsüsteemide vahelise suhtlusprotokolli ICP-2 implementeerimisega ning loodi esialgne spetsifikatsioon. Uue protokolli *overhead* 64 KiB andmepaketi juures on umbes 450 korda väiksem kui ESTCube-1 kasutatud ning umbes 5460 korda väiksem kui CAN protokollil. Töös väljatoodud suhtlusprotokolle võrreldes paistab ICP-2 silma oma lihtsuse, minimalistlikuse ja vähese ressurssinõudlikusega. ICP-2 protokolli arendus ja erinevate alamsüsteemide riistvaradel testimine jätkub ka tulevikus.

Kokkuvõttes said töös püstitatud eesmärgid täidetud. Töö käigus kirjutati ICP-2 spetsifikatsioon, alustati protokolli esialgse implementatsiooniga, loodi programm alamsüsteemide simuleerimiseks ja protokolli testimiseks ning dokumenteeriti programmi API programmi lähtekoodis.

Internal Communication Protocol for ESTCube-2 subsystems

Bachelor Thesis (12 ECTS)

Sander Tammesoo

Summary

Back in 2008 University of Tartu started the first Estonian Student satellite project. Five years later the first Estonian satellite ESTCube-1 was launched to low Earth orbit. ESTCube-1's mission was to perform the first in-orbit demonstration of electric solar wind sail (E-sail) technology. In the next few years the Estonian Student satellite program aims to conduct a similar experiment further away from Earth, so that the Earth's magnetic field would have minimal effect on the results. For the satellite to succeed it's subsystems need to be able to communicate. Due to fundamental differences in ESTCube-1 and ESTCube-2, the internal communication protocol can not be integrated to ESTCube-2. The purpose of this thesis is to find a suitable protocol to be used on ESTCube-2. If need-be describe the specification and start the implementation and testing of a new protocol. The thesis is divided into six chapters. First chapter is the introduction. Second chapter describes the hardware selection and the shared communication bus of the ESTCube-2 satellite. In the third chapter, an in depth overview of the problem and other standard protocols is given. The specification and description of the new proposed protocol can be found in chapter four. Chapter five gives an overview of the problems that occurred during development and a short description about the testing procedures. Future work and further development is proposed in the last chapter.

Tänuavaldused

Sooviksin tänada oma juhendajaid Indrek Sünterit ja Mart Noormad. Indrek oli töö kirjutamisel tohutult suureks abiks ning alati olemas kui tekkis küsimusi. Samuti tänan südamest oma vanemaid ja perekonda, kes on mulle alati toeks olnud. Tuhat tänu ka kõikidele sõpradele ja lähedastele, kes olid mulle käesoleva töö koostamise jooksul nõu ja jõuga abiks.

Viited

[1] S. Lätt, et al. “ESTCube-1 nanosatellite for electric solar wind sail in-orbit technology demonstration.”*Proc. Estonian Acad. Sci.*, 2014, **63**(2S), 200-209.

[2] P. Janhunen, “Electric sail for spacecraft propulsion,” *AIAA Journal of Propulsion and Power*, August 2004, **20** (4), 763–764.

[3] P. Janhunen, P. Toivanen, J. Envall, S. Merikallio, G. Montesanti, J. G. del Almo et al., “Overview of electric solar wind sail applications”, *Proc. Estonian Acad. Sci.*, 2014, **63**, 267-278.

[4] CubeSat Design Specification. http://www.cubesat.org/ Revision 13.

[5] P. Janhunen, et al. “Invited Article: Electric solar wind sail: Toward test missions,” *Review of Scientific Instruments*, 2010, **81**(11)

[6] V. Lobzin ja V. Krasnoselskikh, “Cluster reveals the reformation of the Earth’s bow shock.” *ESA Webstory*, 2007.

[7] STMicroelectronics. *STM32F427xx, STM32F429xx* datasheet, Revision 5

[8] Texas Instruments. *MSP430FR59xx Mixed-Signal Microcontrollers* DATASHEET, Revision E

[9] Texas Instruments. *MSP430FR698x(1), MSP430FR598x(1) Mixed-Signal Microcontrollers* datasheet, Revision B

[10] K. Laizans, I. Sünter, K. Zalite, H. Kuuste, M. Valgur, K. Tarbe, et al., “Design of the fault tolerant command and data handling subsystem for ESTCube-1.”, *Proc. Estonian Acad. Sci.*, 2014, **63**(2S), 222-231.

[11] Bouwmeester, J. and Guo, J., “Survey of worldwide pico- and nanosatellite missions, distributions and sub- system technology.”, *Acta Astronaut.*, 2010, **67**(7–8), 854–862.

[12] D. E. Holmstrøm, “Software and software architecture for a student satellite.” *Norwegian University of Science and Technology*, 2012.

[13] J. L. Tresvig, T. A. Bekkeng ja T. Lindem. “Cubestar—A nanosatellite for space weather monitoring.” *Proc. 1st IAA Conf. University Satellite Missions and CubeSat Workshop*. 2011.

[14] Linear Technology Corporation. *LTC2850/LTC2851/LTC2852 3.3V 20Mbps RS485/RS422 Transceivers* DATASHEET

[15] Modbus protocol. http://www.interlog.com/ speff/usefulinfo/modbus_protocol.pdf

[16] J. Geluso http://srecord.sourceforge.net/crc16-ccitt.html Viimati vaadatud 20.05.2015

[17] O. Pfeiffer, A. Ayre ja C. Keydel, “CAN Overview,” in *Embedded Networking with CAN and CANopen*, 1997, p. 205.

[18] OASIS Open http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf MQTT versioon 3.1.1 spetsifikatsioon, 29.10.2014 revision.

Lisa 1 - Lähtekood

Käesoleva töö käigus valminud ICP-2 teegi implementatsiooni ja selle testkeskkonna lähtekood on kättesaadavad ESTCube-2 meeskonnalt. Alternatiivselt on lähtekood üleslaetud aadressile https://github.com/sandertammesoo/ec2icp.git.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Sander Tammesoo

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

   "Suhtlusprotokoll ESTCube-2 alamsüsteemide vaheliseks suhtluseks"

   mille juhendajad on Indrek Sünter ja Mart Noorma

   (a) reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

   (b) üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile;

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **22.05.2015**