

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Oliver Paadik 155100IASB

**KOHVIMASINA ANDMELOGGERITE
ÜHENDAMINE LORA TEHNOLOOGIAT
KASUTADES**

Bakalaureusetöö

Juhendaja: Uljana Reinsalu
PhD

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Oliver Paadik

28.12.2020

Annotatsioon

Käesolevas bakalaureusetöö peamine eesmärk oli valmis arendada prototüüp, mis võimaldaks kohvimasinade sisse paigaldatud andmeloggeritel omavahelist ühendust luua, kasutamata kontoris/kodus olemasolevat kohtvõrku (WiFi, Ethernet). Lahenduse puhul on seatud peamiseks eesmärgiks, et masinad suudaksid siseruumides suhelda üksteisega vähemalt 100 meetri kaugusel nii, et nende vahel puudub otsenähtavus. Töös võrreldakse levinumaid traadita andmeside protokolle ja tehnoloogiaid nagu WiFi, Bluetooth, ZigBee ja LoRa ning kõrvutatakse neid ülesande püstituses seatud kriteeriumite alusel. Lõpuks põhjendatakse miks langes valik just LoRa tehnoloogia peale.

Semtechi SX1276 kiibistikku kasutava LoRa transiiver mooduli põhjal arendatakse prototüüp, kus üks seade kuulab litsentsivaba 868 MHz sagedusala kasutatavat kanalit ja lõppsõlmedena käituvad seadmed saadavad selle poole teatud intervallide tagant sõnumeid. Testimise läbi veendutakse, et LoRa tehnoloogia võimaldab tõepoolest hoonesiseselt 100 meetriselt distantsilt sõnumite saatmist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 8 peatükki, 7 joonist ja 3 tabelit.

Abstract

CONNECTING COFFEE MACHINE DATALOGGERS USING LORA TECHNOLOGY

The aim of this bachelor's thesis is to develop a prototype which would allow to create a point-to-point wireless connection between two or more Raspberry Pi-based dataloggers installed in the coffee machines, without using existing LAN (WiFi, Ethernet). Main requirement for the connection is covering the distance of 100 meters between the machines in indoor environment, without line of sight. More common wireless technologies and protocols like WiFi, Bluetooth, ZigBee and LoRa are described. Considering the requirements posed for the solution, mentioned technologies are compared and a decision is made to use LoRa modules for connecting the devices.

Using the LoRa transceiver modules based on the Semtech SX1276 chip, a prototype is developed where one of the devices listens to the license-free ISM band of 868 MHz and the other devices, acting as endpoints, will send the payloads at certain intervals. Through testing in TalTech main building, necessary information for default configuration and setup of the LoRa connection is gathered and it is demonstrated that the LoRa connection indeed allows to setup a network of two or more devices in indoor environment where the distance between the devices is approximately 100 meters.

Thesis is written in Estonian and contains 29 pages of text, 8 chapters, 7 figures and 3 tables.

Lühendite ja mõistete sõnastik

BLE	<i>Bluetooth Low Energy</i>
CSMA/CA	<i>Carrier-sense multiple access with collision avoidance</i>
CSS	<i>Chirp Spread Spectrum</i>
DIO	<i>Digital Input-output</i>
EDR	<i>Enhanced Data Rate</i>
FIFO	<i>First-in First-out</i>
FSK	<i>Frequency-shift keying</i>
GFSK	<i>Gaussian frequency-shift keying</i>
GPIO	<i>General Purpose Input-output</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System of Mobile Communications</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
LPWAN	<i>Low Power Wide Area Network</i>
OOK	<i>On-off keying</i>
RPi	<i>Raspberry Pi</i>
RSSI	<i>Received signal strength indicator</i>
SMA	<i>SubMiniature version A</i>
SNR	<i>Signal-to-noise ratio</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver-transmitter</i>
UHF	<i>Ultra High Frequency</i>
WLAN	<i>Wireless Local Area Network</i>
WNAN	<i>Wireless Neighborhood Area Network</i>
WPAN	<i>Wireless Personal Area Network</i>
WWAN	<i>Wireless Wide Area Network</i>

Sisukord

1 Sissejuhatus	9
1.1 Olemasolev lahendus.....	9
1.2 Ülesande püstitus	10
1.3 Töö struktuur.....	10
2 Traadita võrgu protokollide võrdlus	12
2.1 WiFi.....	13
2.2 Bluetooth.....	14
2.3 ZigBee	15
2.4 LoRa	18
2.5 Protokollide võrdluse kokkuvõte	20
3 Lahenduse arhitektuur ja toimimispõhimõtted	22
4 LoRa moodul	24
4.1 SX1276 kiip	25
4.2 Ühendus Rasperry Pi-ga	26
5 Programmi arendus	29
5.1 pySX127x	29
5.2 Lüüs (<i>gateway</i>).....	30
5.3 Lõppsõlm (<i>endpoint</i>)	32
5.4 Käivitusprogramm.....	33
5.5 LoRa ühenduse parameetrid	33
6 Testimine.....	35
7 Edasised võimalikud arendused.....	37
8 Kokkuvõte	38
Kasutatud kirjandus	39
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	41
Lisa 2 – Vastuvõtja voodiagramm.....	42
Lisa 3 – Saatja voodiagramm	43

Jooniste loetelu

Joonis 1. Olemasolev lahendus.....	9
Joonis 2. ZigBee arhitektuur.....	16
Joonis 3. LoRaWAN arhitektuur	18
Joonis 4. Lahenduse arhitektuur	22
Joonis 5. Adapter-trükkplaadile joodetud LoRa moodul	26
Joonis 6. LoRa mooduli ühendus Raspberry Pi-ga.....	27
Joonis 7. Testimiseks kasutatud asukohad TalTech peahoones	35

Tabelite loetelu

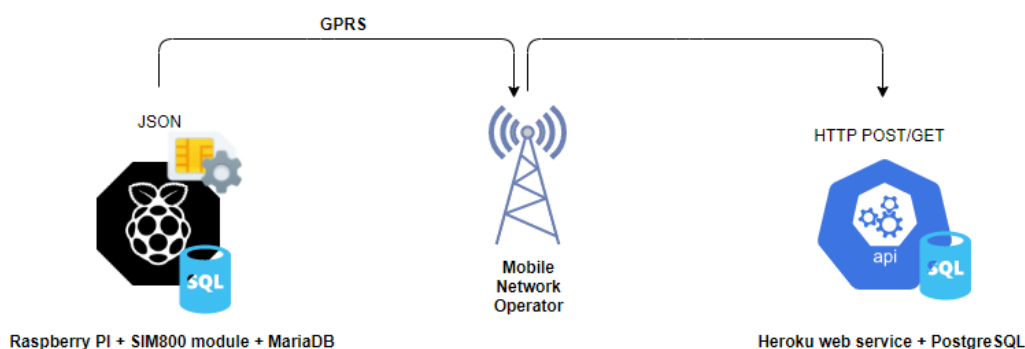
Tabel 1. Traadita võrgu protokollide võrdlus.....	21
Tabel 2. LoRa mooduli tehnilised näitajad	24
Tabel 3. Andmebaasi tabeli motor_log struktuur	32

1 Sissejuhatus

Kui veel ligi 10 aastat tagasi oli tavalises majapidamises internetiga ühenduses vaid personaalarvuti, siis tänapäevaks on selliste (kodu)masinate arv hüppeliselt kasvanud. Tolmuimejad, külmkapid, pesumasinad ja valvesüsteemid on vaid väike osa seadmetest, mille puhul on internetiühendust kasutatud, et automatiseerida või hõlpsamaks muuta mingisugune aeganõudev või lihtsalt tüütu tegevus. Sellesse nimekirja võib lisada ka kohvimasinad. Informatsioon masinast võetud jookide kohta võib väga väärtuslikuks osutuda näiteks suure kollektiiviga ettevõtetele, kus töötajatele on ette nähtud kindel arv soojasid jooke päevas. Andmed kohvimasina seisukorrast või kohviubade, kakaopulbri vmt jäägi üle masinas on aga olulised näiteks kohvimasinate rentimisega tegelevatele ettevõtetele. Need on vaid mõned näited, kus kodumasinade omavaheline ühendus või ühendus internetiga võib kergemaks muuta teatavaid (töö)protsesse.

1.1 Olemasolev lahendus

Antud lõputöö on jätkuks varasemale projektile (tehtud ainete IAS1420 Arvutite ja süsteemide projekt ja IXX1430 Arvutisüsteemid – projekt raames), kus sai ehitatud lahendus, mille abil sai kohvimasinatest võetud jookide üle arvet pidada. Seda ilma kodu/kontori kohtvõrku (WiFi, Ethernet) kasutamata.



Joonis 1. Olemasolev lahendus

Kohvimasinatele on paigaldatud RPi, mis kogub reaalajas andmeid masinast võetud jookide kohta. Andmed salvestatakse andmebaasi, kust need fikseeritud intervallide järel üles korjatakse, nendest JSON-formaadis sõnum genereeritakse ja GPRS kasutades

üle SIM800 mooduli need Heroku pilveplatvormil olevasse PostgreSQL andmebaasi talletatakse. Andmete vaatamiseks on ReactJS abil loodud kasutajaliides, mis lubab andmeid sorteerida ja filtreerida, võimaldades seega mugavalt näha ülevaadet kindla masina kohta.

1.2 Ülesande püstitus

Käesoleva töö eesmärgiks on uurida võimalusi, kuidas olemasolevat lahendust täiendada nii, et mitme kohvimasina puhul ei tuleks igale masinale külge panna SIM800 moodulit, mis tähendaks lisa-GSM kõnekaarte ja igakuist lisakulu mobiilioperaatori kuutasude näol. Üheks tingimuseks on ka kohvimasinate omavahelise võrgu sõltumatus välistest kohtvõrkudest, st masinad peaksid saama andmeid vahetada ka siis, kui puudub WiFi-võrgule ligipääs.

Eeltingimused lahenduse puhul on järgmised:

1. Kohvimasinad ei paikne keskest kohvimasinast, mis SIM800 mooduli abil andmeid üle GPRS-i pilvesse saadab, kaugemal kui 100 meetrit;
2. Kohvimasinate vahel puudub otsenähtavus, st võivad asuda erinevates ruumides, mis on üksteisest eraldatud ühe või mitme seinaga ja/või korrusega.

1.3 Töö struktuur

Töö esimeses osas võrreldakse levinumaid traadita ühenduse protokolle, tuues välja peamised tehnilised erisused ning plussid-miinused antud lahenduse kontekstis. Võrreldakse nii WiFi-t (peatükk 2.1), Bluetooth-i (peatükk 2.2), ZigBee-d (peatükk 2.3) kui ka LoRa (peatükk 2.4). Viimaks tuuakse välja põhjused, miks kohvimasina(te) andmeloggerite ühendamiseks just LoRa tehnoloogia valiti.

Töö teine osa keskendub lahenduse arhitektuurile ja arendusele. Peatükis 4 tutvustatakse Semtechi SX1276 kiibil põhinevat LoRa moodulit ja näidatakse, kuidas on lahendatud selle ühendamine RPi-ga. Järgnevas peatükis on ära kirjeldatud nii lüüsi (*gateway*) kui ka lõppsõlme (*endpoint*) funktsionaalsus – esimene kuulab kanalit pidevalt ja reageerib sissetulevatele sõnumitele, teine kogub teatud intervallide tagant andmebaasist uued

andmed ja saadab need siis lüüsi poole tee. Välja toodud ja põhjendatud on ka parameetrite valikud LoRa ühenduse loomiseks masinate vahel. Valmis lahenduse testimist on kirjeldatud peatükis 6. Erinevate katsete läbi sai selgeks, millise LoRa laotusteguri (*spreading factor*) valida vaikeseadistusse.

Töö lõpetab peatükk 7, kus käiakse välja mõned ideed, millises suunas võiks valmis saanud lahendust edasi arendada.

2 Traadita võrgu protokollide võrdlus

Tänu Asjade Interneti (IoT) kiirele arengule viimase 10 aasta jooksul, mida peamiselt veab Industry 4.0 ehk tööstuse automatiseerimine ja ühendamine internetti, on pilt turul olevatest traadita kommunikatsiooni protokollidest ja tehnoloogiatest võrdlemisi kirju. Iga protokoll on loodud mingit kindlat eesmärki, kasutusala/kasutusjuhtu silmas pidades ning suures plaanis võib välja tuua kolm aspekti, millele uut traadita võrgu tehnoloogiat arendades kõige enam rõhku asetatakse:

1. Leviala ehk kui kaugele signaal levib. Siin võib maksimaalse leviala järgi eristada WPAN (Bluetooth, Thread, ZigBee, Z-Wave), WLAN (WiFi, White-Fi, WiFi HaLow), WMAN (JupiterMesh) ja WWAN (mobiilsidevõrgu tehnoloogiad nagu 3G, 4G, NB-IoT ja vähese energiatarbe ja laia ulatusega traadita sidevõrgu tehnoloogiad nagu LoRa, SigFox, DASH7) protokolle [1];
2. Energiatarve ehk kui palju voolu on tarvis, et pakette üle kanali saata ja seadet töös hoida. Protokolle, mis võimalikult madalale voolutarbele rõhku panevad, võib tuua ühise nimetaja alla kui LPWAN-tehnoloogiad. Nende kasutusala on peamiselt patareitoiteliste mõõdikute/sensorite pealt info kogumine;
3. Andmeedastuskiirus ehk kui kiirelt on võimalik andmeid üle kanali saata.

Käesolevas töös esitatava lahenduse puhul olen esmatähtsaks seadnud tehnoloogia poolt võimaldatava leviala. Kuna seade, mis andmeid saatma hakkab on pidevalt vooluvõrgus, ei valmista probleeme voolutarve. Samuti pole niivõrd tähtis andmeedastuskiirus, kuna saadetavad andmemahud ehk informatsioon teatud ajahihikus võetud jookide kohta, jäävad väikseks ja pole kindlasti antud juhul ajakriitilised.

Võrdlen järgnevalt nelja erinevat traadita võrgu protokollit, lähtudes eelkõige lahenduse vajadustest (ühendus kahe masina vahel, mis asetsevad üksteisest *ca* 100 m kaugusel), tehnoloogia (s.o moodulite) kättesaadavusest ja hinnast ning valitava tehnoloogia kasutuslihtsusest (Pythoni ja/või C teegid, varasemalt tehtud projektid jms). Kindlasti pole see valik IoT turul valitsevat hetkeolukorda arvestades ammendav, kuid töö eesmärk pole niivõrd erinevate tehnoloogiate näitajate võrdlemine, kuivõrd kaalutletud

otsus, mille põhjal oleks võimalik kokku panna sobiv lahendus kahe andmelogeri omavaheliseks kommunikatsiooniks.

2.1 WiFi

WiFi on IEEE 802.11 standardeid kasutatav traadita võrgu protokoll, mis on kujunenud *de facto* kodu- või kontorikohtvõrgu protokolliks. Olenevalt standardist, kasutab WiFi tavaliselt kas 2.4 GHz ja/või 5 GHz sagedusala. Sagedusalast sõltub suuresti ka WiFi leviala – kuigi 5 GHz lubab suuremaid kiiruseid, katab madalamatel sagedusaladel töötav võrk suurema ala. Käesoleva lahenduse juures on mõistlik hakata rääkima minimaalselt 4. põlvkonnast¹ ehk standardist 802.11n, mis lubab WiFi võrgu levialaks kuni 250 m, siseruumides kuni 75 m [2].

2016. aastal WiFi Alliance poolt välja kuulutatud WiFi HaLow (802.11ah) on katse WiFi võimekust ja võimalusi laiendada ka IoT maailmale. Kasutades 1 GHz alla jäävaid litsentsivabasid sagedusalasid, võimaldab see, võrreldes teiste 802.11 standarditega, võrguga katta mitmeid kordi suuremaid vahemaid. Ahmed *et al.* koostatud mudelite järgi on 802.11ah võimalik leviala siseruumides kuni 543 m [3]. Kahjuks on see võrdlemisi uus tehnoloogia – esimene laiemale turule mõeldud moodul kuulutati välja 2020. aasta alguses Silex Technology poolt [4] – ja seega ka liiga kallis, et seda antud töö kontekstis käsitleda.

Kuigi RPi omab juba sisseehitatud WiFi võimekust alates Pi 3-st, jääb tänu plaadile integreeritud antennile selle võimsus väikseks ja standardiga 802.11n lubatud signaali maksimaalne leviala kättesaamatuks. Üheks võimalikuks lahenduseks oleks (antenniga) WiFi moodul, mille saab arvutiga ühendada läbi USB. Valikukriteeriumiteks oleks seejuures, et mooduli kiibistik oleks toetatud Raspbiani distro poolt ja et see toetaks *ad hoc* võrgu loomist kahe masina vahel. Paljude müügil olevate WiFi pulkade puhul on

¹ Wi-Fi Alliance kasutab alates 2018. aastast standardite hõlpsamaks eristamiseks versiooninumbreid – 802.11n ehk 4. põlvkond, 802.11ac ehk 5. põlvkond jne. Varasemad standardid nummerdati tagasiulatuvalt.

juba ära märgitud sobivus RPi-ga, kuid informatsioon *ad hoc* võrgu toe kohta on üsna fragmenteeritud ja tuleb toetuda teiste kasutajate kogemustele.

Üheks potentsiaalseks mooduliks võiks olla RT5370 kiibistikku kasutatav WiFi pulk, mille puhul on garanteeritud ühildumine Raspbiani (või muu Linuxi distroga) [5] ja teiste kasutajate edukad katsed kahe masina vahelise võrguühenduse loomiseks. Sellise mooduli maksumus jääb vahemikku 2,47 € (Aliexpress¹) kuni 8,88 € (Amazon²).

802.11 standardi puhul on suureks plussiks kasutusele võtmise hõlpsus. Kuna volutarve pole antud lahenduse puhul probleemiks, võivad masinad olla pidevas ühenduses ja teatud intervallide tagant saadab üks masin internetiga ühenduses olevasse masinasse vajaliku informatsiooni. WiFi kasuks räägib ka see, et Raspbiani operatsioonisüsteem teeb ise palju võrguühenduse loomise ja hoidmise puhul ära ja seega oleks suures osas vaja keskenduda vaid andmete kokkukogumise- ja saatmise loogikale. Takistuseks võib saada aga piiratud leviala ja seda eriti siseruumides, kus on kasutusel ka teised WiFi-võrgud ja muid raadiolaineid häirivaid faktoreid.

2.2 Bluetooth

Bluetooth (IEEE 802.15.1) on lähiala raadiolink, mis toimib sagedusvahemikus 2,402-2.485 GHz, mis on üsna sarnane WiFi-võrkude sagedusele. Bluetooth ühenduse puhul jaotatakse andmed pakettideks ja iga pakett saadetakse üle ühe kindla kanali (ribalaiusega 1 MHz, uuematel versioonidel 2 MHz), mida Bluetoothil on kokku 79 (varasemad versioonid) või 40 (uuemad versioonid, alates versioonist 4).

Bluetooth hakkas kanda kinnitama 2004. aastal, kui välja tuli versioon 2.0, mis tänu EDR (*Enhanced Data Rate*) tehnoloogiale lubas teoreetilisi andmeedastuskiirusi kuni 3 Mb/s. Lisaks sellele vähenes eelmise versiooniga võrreldes volutarve. Versioon 3.0

¹ <https://www.aliexpress.com/item/32834257814.html> (külastatud 04.12.20)

² <https://www.amazon.com/Ralink-RT5370-Raspberry-Adapter-Function/dp/B019XUDHFC> (külastatud 04.12.20)

2009. aastal mitmekordistas kiirusi tänu 802.11 WiFi raadiolingile, samas kui uusi ühendusi loodi, hallati ja seadistati ikka läbi Bluetoothi raadiolingi. Koos 2010. aastal välja tulnud versioon 4.0-ga, ilmus ka Bluetooth LE (*Low Energy*), mis on juba suurem samm IoT vajaduste poole – tänu protokollu uuendustele suudeti Bluetoothi voolutarvet märkimisväärselt vähendada, säilitades 4.0 poolt võimaldatud leviala [6].

2016. aastal tutvustatud versioon 5 on eelkõige Bluetooth LE protokollu edasiarendused – säilitades madala voolutarbe, võimaldab uus versioon valida nelja erineva andmeedastuskiiruse vahel (2 Mb/s, 1 Mb/s, 500 kb/s või 125 kb/s), kus madalamatel kiirustel on selle võrra suurem maksimaalne vahemaa kahe Bluetooth seadme vahel [7]. Kui veel eelmise versiooni juures jäi võimalik maksimaalne vahemaa kahe seadme vahel 10–30 meetri vahele, siis uue versiooniga on kõige madalama kiirusega võimalik see pea kahekordistada 30–50 meetri peale [8].

RPi-d omavad sissehitatud Bluetooth võimekust alates Pi 3 Model B-st (Bluetooth versioon 4). Alates Pi 4. põlvkonnast on plaadi peal Bluetooth versioon 5, kuigi see ei toeta suuremate vahemaade jaoks vajalikku BLE Coded PHY-d, mistõttu tuleks pigem vaadata väliste moodulite poole. Üheks potentsiaalselt sobivaks mooduliks võiks olla Nordic Semiconductor programmeeritav NRF52840 USB pulk, mille hind jääb 10 € kanti (Farnell¹). NRF52840 toetab Bluetooth LE kaugrežiimi (*long range mode*) ja isegi Bluetooth silmvõrgu (*mesh network*) loomist.

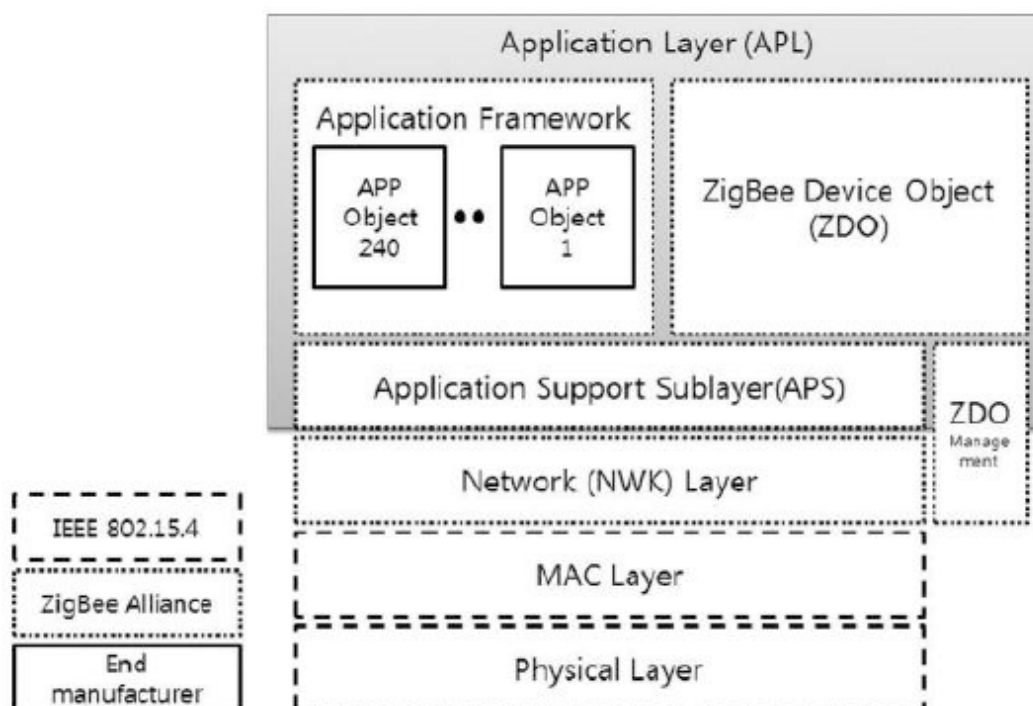
Plussid ja miinused on Bluetooth ühenduse puhul üsna sarnased WiFi-le, kõige suuremaks takistuseks jääb kindlasti Bluetoothi piiratud leviala.

2.3 ZigBee

ZigBee protokollu juures määrab füüsilise- ja MAC-kihi ära standard IEEE 802.15.4 ning võrgu- ja rakenduskihi ZigBee Alliance poolt hallatavad standardid. IEEE 802.15.4

¹ <https://ee.farnell.com/nordic-semiconductor/nrf52840-dongle/bluetooth-module-v5-2mbps/dp/2902521?st=nrf52840> (külastatud 04.12.20)

standardi järgi defineeritud füüsiline kiht töötab kahel ISM sagedusalal 868/915 MHz (esimene Euroopa ja teine USA/Austraalia jt riikide jaoks) ja 2.4 GHz. MAC-kiht kontrollib juurdepääsu raadiokanalile kasutades CSMA/CA protokoll. Rakenduskiht koosneb rakendust toetavast alamkihist (APS) ja ZigBee seadme objektist (ZDO) – siin tehakse ära kogu töö, mis on seotud teiste ZigBee seadmete leidmise ja ühendamisega [9]. ZigBee loodi eesmärgil pakkuda väikese voolutarbega ja odavamaid lahendusi WPAN loomiseks, seda eelkõige võrreldes teiste populaarsemate traadita ühendustega nagu WiFi ja Bluetooth.



Joonis 2. ZigBee arhitektuur

Andmeedastuskiirused sõltuvad peamiselt sagedusalast – alla 1 GHz töötaval sagedusalal on teoreetiline kiirus 20 kb/s, 2.4 GHz puhul 250 kb/s. ZigBee toetab kolme erinevat võrgutopoloogiat: silmvõrku (*mesh network*), tähtvõrku (*star network*) ja puuvõrku (*tree network*). Olenemata võrgu arhitektuurist koosneb ZigBee võrk alati kolme tüüpi seadmetest: koordinaatorist (võrgu kõige suuremate õiguste ja võimekusega seade, mis kontrollib võrgu turvalisust, liikmeid jmt), ruuteritest (peale rakenduslike funktsioonide toimib ka võrguliikluse vahelülina) ja lõppseadmetest (funktsionaalsus piirdub andmete kogumise ja nende edastamisega vanemale võrgusõlmele ehk siis kas ruuterile või koordinaatorile). Igal seadmepool on 16-bitine võrguaadress.

Kasutusmugavuse ja -lihtsuse seisukohast vaadates oleks kõige mõistlikum vaadata Digi¹ XBee-seeria poole, mis pakub lihtsalt seadistatavaid ZigBee protokolliga kasutatavaid mooduleid. Näiteks Digi XBee SX 868 RF moodul töötab Eestis mittespetsiifilistele lähetoimeseadmetele eraldatud 868 MHz sagedusel. Toote tehniline spetsifikatsioon lubab linnakeskkonnas teoreetiliseks maksimaalseks distantsiks kuni 2,5 km [10]. Tuleb märkida, et tegemist on ideaalses keskkonnas mõõdetud tulemusega, kus üheks kriteeriumiks seadmete vaheline otsenähtavus (*line of sight*) ja häirevaba keskkond. Andmeedastuskiiruseks lubatakse 10-80 kb/s. Mooduliga on võimalik suhelda läbi SPI või UART-liidese ja mooduli seadistamise teeb eriti mugavaks graafilise seadistusplatvormi XCTU olemasolu, aga võimalik on seda teha ka läbi Raspbiani käsurea. Hinnaks on umbes 27 € (Mouser²).

Teine võimalus on Xbee 3 Zigbee 3.0 moodul, mis kasutab 2.4 GHz leviala ja mille leviala on seetõttu SX 868-st väiksem – kuni 60 m ja 90 m XBee 3 PRO Zigbee 3.0 puhul. Nagu ka SX 868 RF puhul, on spetsifikatsioonis toodud teoreetilised distantsid ideaalse, häirevaba keskkonna puhul. Esimese maksumus on ca 13 € (Mouser³) ja PRO ca 16 € (Mouser⁴).

Kuna ZigBee on mõeldud peamiselt koduautomaatika jaoks, on andmepaketis eraldatud ruum edastatava info jaoks piiratud – vaid 127 baiti [11]. See omakorda tähendab seda, et andmed, mis kohvimasinast üle ZigBee võrgu saadetakse, peavad olema pikemate intervallide tagant saates tükeldatud väiksemateks osadeks või saadetud tihedamini. Viimase puhul võib takistuseks saada sagedusalale määratud töötsükli piirangud. Xbee kasuks räägib aga juba valmis kirjutatud Pythoni teegid⁵, mis hõlbustavad Xbee moodulite vahelise võrgu loomist.

¹ Digi on ka ZigBee Alliance liige

² <https://www.mouser.ee/ProductDetail/DIGI/XB8X-DMRS-001?qs=sGAEpiMZZMu3sxp5v1qroZc48vGwABcx7yIge1%252B4WI%3D> (külastatud 04.12.20)

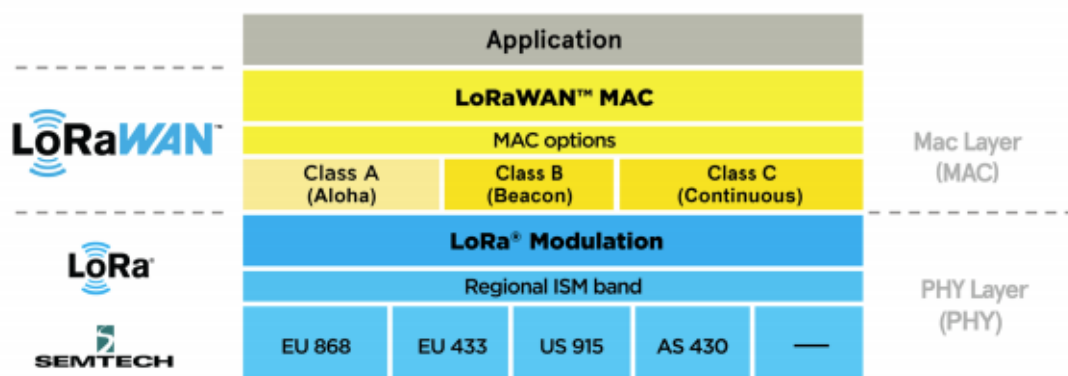
³ <https://www.mouser.ee/ProductDetail/DIGI/XB3-24Z8UM-J?qs=W0yvOO0ixfGVfR7tnmf86w%3D%3D> (külastatud 04.12.20)

⁴ <https://www.mouser.ee/ProductDetail/DIGI/XB3-24Z8UM?qs=W0yvOO0ixfEfyhbmVUqhjQ%3D%3D> (külastatud 04.12.20)

⁵ <https://xbplib.readthedocs.io/en/latest/index.html>

2.4 LoRa

LoRa on Semtechi poolt arendatud vähese energiatarbe ja laia ulatusega traadita sidevõrgu (LPWAN) protokoll. LoRa võrguühenduse füüsiline kiht kasutab CSS (*Chirp Spread Spectrum*) modulatsiooni ja töötab litsentsivabadel 1 GHz alla jäävatel ISM sagedusaladel. LoRaWAN spetsifikatsioon defineerib ära MAC-kihi, mis haldab lüüsi (*gateway*) ja lõppsõlme (*endpoint*) vahelist ühendust ning võrgu üldise ülesehituse. Protokoll ja selle standardeid haldab LoRa Alliance, 2015. aastal loodud ja enam kui 500 liikmest koosnev mittetulundusühendus [12].



Joonis 3. LoRaWAN arhitektuur

Lõppsõlmed võivad olla kolme erinevat tüüpi:

- **Klass A** ehk madalaima energiatarbega, kahesuunalise suhtlusega lõppsõlmed. Lõppseadmel on vaid kaks sõnumite vastuvõtmisakent, alati peale üleslink saatmist. Muul ajal on seade unerežiimil. Üleslink avatakse vastavalt vajadusele. Kõik LoRa seadmed kuuluvad vähemalt klassi A.
- **Klass B** ehk deterministliku allalink latentsusega (*deterministic downlink latency*) kahesuunalise suhtlusega lõppsõlmed. Lisaks klassis A määratud allalink vastuvõtuakendele avab lõppsõlm allalink suhtluse pärast lüüsi poolt saadetud perioodilist pingi.
- **Klass C** ehk madala latentsusega, kahesuunalise suhtlusega lõppsõlmed. Võrreldes Klass A seadmetega jäetakse kanal pärast üleslinki lahtiseks ja lõppsõlm on valmis andmeid vastu võtma põhimõtteliselt igal ajal. Klass C on

kõige energiakulukam, mistõttu sobib ta vaid kasutamiseks, kui lõppseade on vooluvõrku ühendatud.

LoRa peamised kasutusala on sellised, kus esmatähtis on lahenduse madal voolutarve (näitude saatmine, sensoritelt lihtsa info kogumine jmt) – patareitoitel LoRa lõppsõlm võib töötada üle 10 aasta. Käesolevas projektis aga on LoRa kasuks rääkivaks argumentiks pigem maksimaalne distants kahe sõlme vahel – Augustin jt on eksperimentaalselt linnakeskkonnas erinevaid laotustegureid (*spreading factor*) kasutades saanud pakette saata kuni 3 kilomeetri kaugusele [13]. Sõltuvalt laotustegurist, mida on kuus ehk täisarvud 7-12, on maksimaalseks andmesidekiiruseks 27 kb/s kuni 0.3 kb/s [14].

Kuna LoRa on peamiselt mõeldud sensoritelt tuleva info saatmiseks, on tema suurimaks puudujäägiks ühe sõnumiga saadava info hulga piiratus. Kasutades varieeruva suurusega pakettide režiimi (*Variable Length Packet Format*) on sõnumi suurus ära kirjeldatud päises eraldiseisvas 8-bitisel väljal, mis teeb maksimaalseks sõnumi suuruseks 255 baiti [15].

RPi-ga ühilduv LoRa moodul võib maksta umbes 5 € (AliExpress¹) aga on ka kallimaid mooduleid nagu SparkFun WRL-15836 (Mouser², 25,37 €) või Adafruiti RFM95W (Mouser³, 17 €). Moodulid baseeruvad enamasti kas Semtechi SX1276 või HopeRF RFM95 transsiiver kiipidel. Ühendus mikrokontrolleriga toimub üle SPI-liidese, kuid on võimalik leida ka UART-ühendust kasutavaid mooduleid. Eelotsingut tehes jälgisin, et moodul võimaldaks ühendada ka lisaantenni suurema leviala kättesaamiseks.

¹ <https://www.aliexpress.com/item/32859550198.html?spm=a2g0s.9042311.0.0.480e4c4dw2FcSq>
(külastatud 04.12.20)

² <https://www.mouser.ee/ProductDetail/SparkFun/WRL-15836?qs=%2Fha2pyFadugWJ46KWECErQla0s6pYgzo%252BWv4WTyRHqW67gVks5wfdxw%3D%3D>
(külastatud 04.12.20)

³ <https://www.mouser.ee/ProductDetail/RF-Solutions/RFM95W-868S2?qs=%2Fha2pyFadug%2FaviqoHKJzxmdRQIyyHW3dnqBtsJgZXl6O%252Bh5v3AuHQ%3D%3D>
(külastatud 04.12.20)

Tänu sellele, et Semtech on paljud draiverid avaldanud enda GitHub kontol¹, on nende peale ehitatud ka kõrgemas keeles olevad teegid, mis teevad kasutamise võrdlemisi hõlpsaks. Üheks selliseks on näiteks pyLora².

2.5 Protokollide võrdluse kokkuvõte

Erinevate protokollide tehnilised näitajad ja maksumused on kokkuvõtvalt esitatud Tabelis 1. Kokkuvõtvalt võib öelda, et kuigi WiFi või Bluetoothi kasutamine oleks tehniliselt kõige lihtsam, ei suuda nad suure tõenäosusega rahuldada ülesande püstituses eesmärgiks seatud 100 meetrist vahemaad kahe masina vahel. Seega tuleks vaadata LPWAN tehnoloogiate poole nagu ZigBee või LoRa. Nende kahe puhul räägivad nii mooduli hind kui ka leviala tugevalt LoRa kasuks.

LoRa-t on edukalt testitud siseruumides ja sellele tuginedes võib kindel olla, et 100-meetrise raadiusega ala suudab antud tehnoloogiaga garanteeritult katta. Kuna käesoleva lahenduse puhul on keskne seade juba internetiga ühenduses (SIM800 mooduli abil loodud GPRS ühendus), siis pole vaja kasutada teiste teenusepakkujate LoRaWAN *gateway*'d, mis tähendab, et puudub lisamaksumus teenusepakkujate litsentsi- ja kuutasude näol. LoRa kasuks räägib ka autori varasem kokkupuude tehnoloogiaga.

¹ <https://github.com/Lora-net>

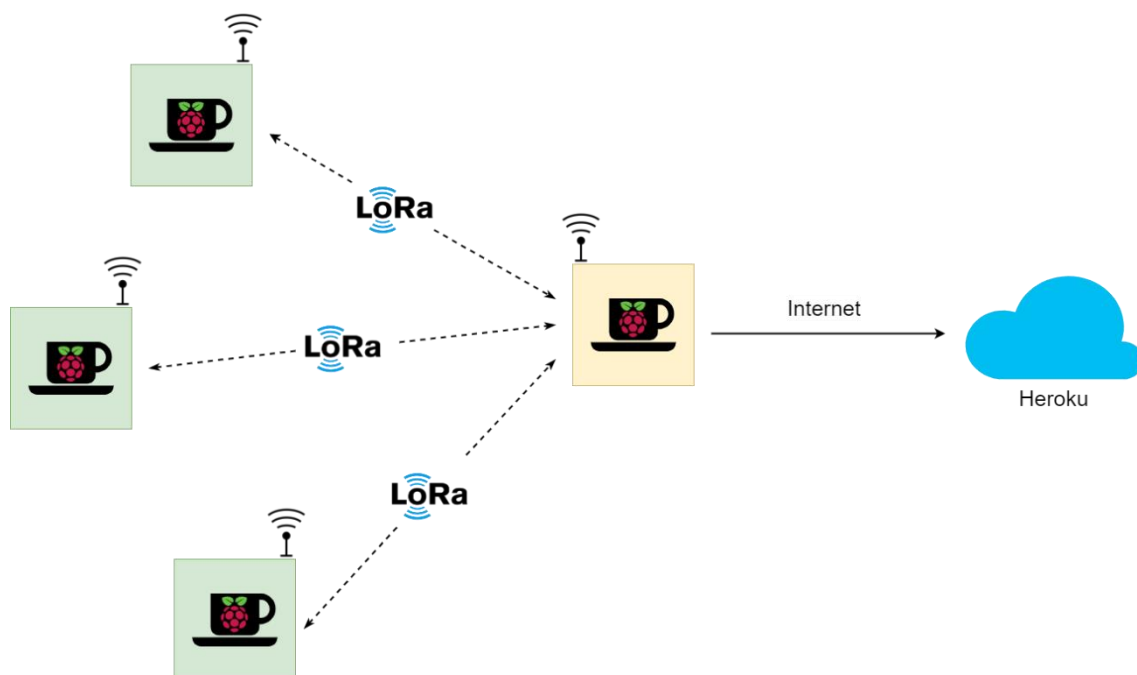
² <https://pypi.org/project/pyLora/>

Tabel 1. Traadita võrgu protokollide võrdlus

	WiFi (802.11n)	Bluetooth (v5)	ZigBee / Xbee	LoRa
Sagedusala	2.4 GHz	2.4 GHz	868 MHz / 2.4 GHz	868 MHz
Kaugus (siseruumides)	kuni 70 m	kuni 50 m	teoreetiliselt kuni 2,5 km (Xbee SX 868) / kuni 90 m (XBee 3 PRO Zigbee 3.0)	kuni 3 km
Tugi	OS tugi, Pythoni teegid	OS tugi, Pythoni teegid	Pythoni teegid, graafiline seadistamisplatvor m	Pythoni/C teegid
Andme- edastuskiirus	kuni 150 Mb/s	125 kb/s (LE Coded PHY S=8)	kuni 80 kb/s (Xbee SX 868 RF) / kuni 250 kb/s (XBee 3 PRO Zigbee 3.0)	kuni 27 kb/s
Tasud (litsentsid, teenused)	-	-	-	Ainult mõne teenuspakkuja LoRaWAN <i>gateway</i> 'd kasutades
Mooduli maksumus	~ 3 €	~ 10 €	13-27 €	~ 5 €

3 Lahenduse arhitektuur ja toimimispõhimõtted

Lähtudes olemasolevast lahendusest, kus üks seade on juba SIM800 GSM mooduli abil internetiga ühenduses, laieneb võrgu arhitektuur teiste, internetiga mitteühenduses olevate masinate läbi. Selguse ja lihtsuse huvides kasutan internetiga ühenduses oleva masina nimetamiseks edaspidi terminit lüüs (*gateway*) ja teisi, mis üle LoRa raadioside andmeid peaseadmesse saadavad, mõistega lõppsõlmed (*endpoint*). Joonisel 4 on välja toodud lahenduse arhitektuur, kus roheliselt on märgitud lõppsõlmed ja kollasega lüüs.



Joonis 4. Lahenduse arhitektuur

Lüüsi ja lõppsõlmede vahel on ühendus kahe-suunaline, lõppsõlmed saadavad lüüsi andmeid ja lüüs vastab kinnitusega, kui andmete edastamine õnnestus. Õnnestunud saadetud andmed märgitakse lõppsõlmede andmebaasis ajatempliga (see toimib samal põhimõttel nagu vastuvõtjast andmete pilve saatmine). Igal lõppsõlmel on unikaalne identifikaator, mis saadetakse koos andmetega lüüsi – see jätab võimaluse hiljem info ühest kindlast kohvimasinast võetud jookide kohta eraldada.

Võrgu ülesehitus eeldab seda, et lüüs peab pidevalt kanalit kuulama ja peab alati olema valmis vastu võtma andmeid lõppsõlmede poolt. Lõppsõlmed kuulavad kanalit ainult pärast andmete edastamist, et olla valmis vastu võtma kinnitussõnumit lüüsi – muul ajal

kanalit nad kasutama ei pea. Kuna lahenduse puhul ei kasutata LoRaWAN-i, mis suudab hallata mitme erineva kanali peal toimivat liiklust, vaid ühendus toimub kõikide lõppsõlmede ja lüüsi vahel ühe kanali peal, tähendab see seda, et lõppsõlmed peavad kanalit kasutama ükshaaval ehk jagama kanalit aja järgi.

4 LoRa moodul

Lõplik valik langes Aliexpressis müüdava¹ SX1276 kiipi kasutava mooduli peale. SX1276 oli esimene Semtechi LoRa kiip ja kuigi Semtech on vahepeal turule tulnud juba uuemate versioonidega, on SX1276 laialt kasutuses paljudes turul müüdivates LoRa moodulites (nt HopeRF RFM95w). Mooduli hinnaks kujunes umbes 5 €.

Kuigi moodulil puudub ametlik tehniline dokumentatsioon (välja arvatud müügikuulutuses olev parameetrite nimistu), saab kasutada kas Semtechi enda SX1276 kiibi või HopeRF RFM95w mooduli dokumentatsiooni² – mõlemad on suures osas samad. Peamised tehnilised parameetrid on välja toodud tabelis 2.

Tabel 2. LoRa mooduli tehnilised näitajad

Tööpinge	1.8-3.6 V
Kesksagedus	868 MHz
Modulatsioon	FSK, OOK, GFSK, LoRa®
Väljundvõimsus	max 20 dBm
Vastuvõtja tundlikkus	-139 dBm
Voolutarve	IDLE: 1.5 μ A RECEIVE: 10.8-12 mA TRANSMIT: 20-120 mA
Signaali ribalaius	7.8-300 kHz
Töötemperatuuri vahemik	-40...+85 °C
Mõõtmed	15 x 15 mm

¹ <https://www.aliexpress.com/item/32859550198.html?spm=a2g0s.12269583.0.0.4ad834ddXWMMhF> (külastatud 06.12.20)

² <https://www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf> (külastatud 15.12.20)

4.1 SX1276 kiip

SX1276 on pooldupleks kommunikatsiooni kasutatav transiiver. Saatmise ja vastuvõtmise jaoks on kaks eraldi sagedussüntesaatorit (*frequency synthesizer*), mis genereerivad kohaliku ostsillaatori (*local oscillator*) sagedust nii ülemiste (alates 779 MHz) kui ka alumiste (kuni 535 MHz) UHF-sagedusalade jaoks. SX1276-l on kolm eraldiseisvat raadiosagedus-võimendit. Kaks nendest on ühendatud RFO_LF (madalamad sagedused) ja RFO_HF (kõrgemad sagedused) viikudega ja pakuvad +14 dBm signaalivõimendust, madalat energiatarvet silmas pidades. Kolmas on ühendatud PA_BOOST viiguga ja suudab pakkuda +20 dBm võimendust. Peale standardsete modulatsioonitehnikate nagu OOK, FSK, GFSK, MSK ja GMSK, on SX1276 varustatud ka LoRa modemiga, mis võimaldab kasutada Semtechi patenteeritud CSS-modulatsiooni. Kõik SX1276 kiibi seadistamisregistrid on ligipääsetavad läbi SPI-liidese. [15]

Kiibil on 8 erinevat töörežiimi:

- **SLEEP:** energiasäästlik režiim. SPI ja seadistamisregistrid on ligipääsetavad aga FIFO andmepuhver pole.
- **STANDBY:** ooterežiim.
- **FSTX:** sagedussünteesimisrežiim saatmisele.
- **FSRX:** sagedussünteesimisrežiim vastuvõtmisele.
- **TX:** saatmisrežiim. Aktiveerides saadetakse pakett ja pannakse transiiver tagasi STANDBY-režiimi.
- **RXCONT:** pideva vastuvõtmise režiim. Sissetulevaid andmeid töödeldakse pidevalt ja režiimist tullaakse välja alles pärast manuaalset ümberlülitamist uuele režiimile.
- **RXSINGLE:** ühekordse vastuvõtmise režiim. Pärast paketi saamist lülitatakse tagasi STANDBY režiimile.
- **CAD:** kuulatakse kanalit sobiva LoRa signaali tuvastamiseks.

4.2 Ühendus Raspberry Pi-ga

Kuna RFM95w ja tema erinevad kloonid pole makettplaadile sobivad väga kitsaste viikude tõttu, oli prototüübi mugavamaks testimiseks ja ühendamiseks vaja osta ka RFM95w moodulile mõeldud adapter-trükkplaat. Internetipood Diycon müüb selliseid¹ 2,95 € eest. Jootes trükkplaadile LoRa mooduli ja standardsed 2.54 mm isased kontaktiribad, saab mooduli muuta kerge vaevaga makettplaadile sobivaks. Lisaks sellele saab plaadi külge joota ka emase SMA otsiku, et hiljem sobiv antenn sinna külge ühendada (Joonis 5).



Joonis 5. Adapter-trükkplaadile joodetud LoRa moodul

RPi-ga ühendus on välja toodud joonisel 6. Moodul saab toidet RPi 3.3 V klemmilt. Mooduli DIO0-3 klemmid on ühenduses järgnevate RPi GPIO klemmidega:

- DIO0 → GPIO 4
- DIO1 → GPIO17
- DIO2 → GPIO18
- DIO3 → GPIO27

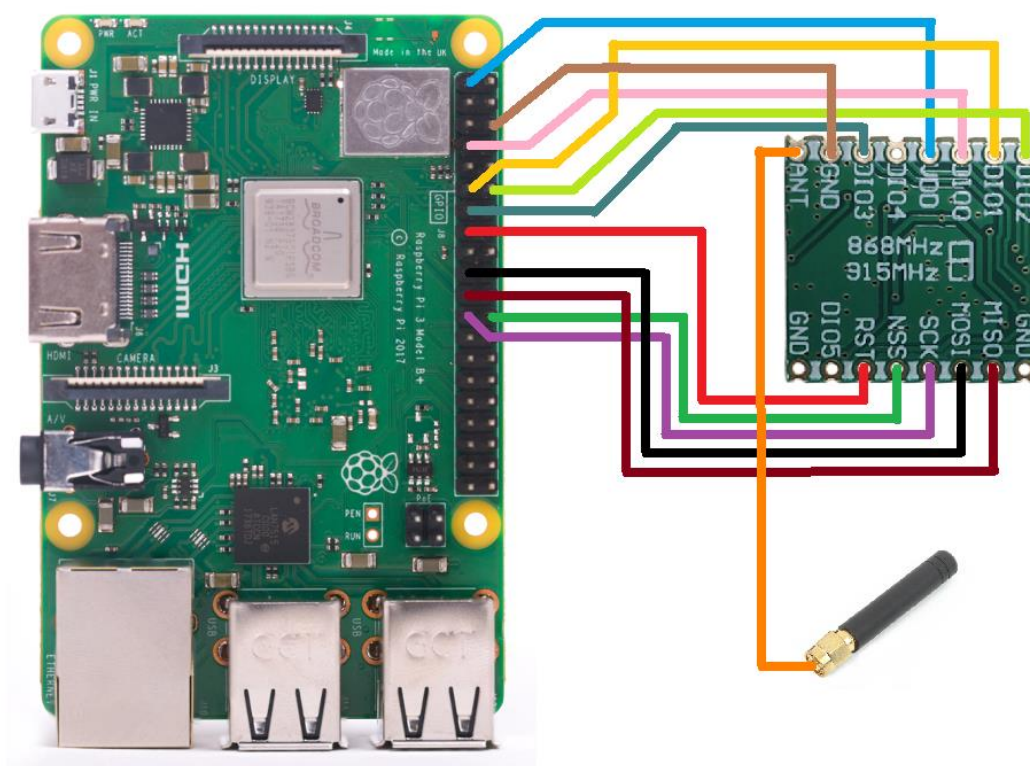
¹ <https://diycon.nl/product/lora-node-pcb-100-shield-only-rfm92-95> (külastatud 09.12.20)

DIO4 ja DIO5 toetavad režiime ja funktsionaalsusi, mis pole antud lahenduse puhul vajalikud ja on seega jäetud ühendamata.

RPi ja mooduli omavaheline suhtlus toimub läbi SPI-liidese, mis sobib hästi käesoleva lahenduse puhul, kuna vastuvõtja rollis oleval RPi-l on UART liides juba hõivatud – sinna on ühendatud SIM800 moodul, mis andmeid pilvesse saadab. Moodul on ühendatud RPi SPI0 klemmide külge järgnevalt:

- MISO → GPIO10 (SPI0 MOSI)
- MOSI → GPIO9 (SPI0 MISO)
- SCK → GPIO11 (SPI0 SCLK)
- NSS → GPIO08 (SPI0 CS0)

Mooduli taaskäivitamist saab kontrollida läbi RPi GPIO22, mis on ühendatud kiibi RST viiguga.



Joonis 6. LoRa mooduli ühendus Raspberry Pi-ga

Adapter-trükkplaadi külge joodetud SMA antennipesasse on ühendatud MIKROE-2349/MIKROE-2350 2dBi võimendusteguriga kummist antenn, mis on mõeldud just LoRa seadmetele ehk optimeeritud sagedusalale 868 MHz.

5 Programmi arendus

GitHub koodivaramust märksõnaga „SX1276“ otsingut tehes tuleb vasteid 149, millest seisuga 10.12.20 on 100 kirjutatud C/C++ programmeerimiskeeles ja 6 on kirjutatud Pythonis. Märksõna „SX127x“ annab 60 projekti, millest 17 on kirjutatud C-s ja 16 Pythonis. Suur osa nendest on kirjutatud LoRaWAN lüüsi jaoks või EBYTE LoRa moodulite jaoks, mis kasutavad UART-liidest ning seega käesoleva lahenduse puhul ei sobi.

Kuna lüüsis andmeid pilve saatev programm on kirjutatud juba Pythonis, sai tehtud otsus ka LoRa ühendust loov ja haldav programm kirjutada samas keeles. Pythoni distros PyPI on saadaval pyLora¹ teek, mis omakorda on ehitatud mayeranalyticsi pySX127x² peale. PyLora võimaldab RPi külge ühendada kuni 2 LoRa moodulit ja pakub võimalusi testida nii kiibi korrektset töötamist kui ka ühendust kahe mooduli vahel. Kahjuks on pyLora *fork*'itud vanast pySX127x versioonist ja pole vahepeal enda lähtekoodi uuendanud, mistõttu on seal veel mõned vead, mis tööd kindlasti mõjutavad. Näiteks on pySX127x puhul vahepeal parandatud SNR arvutusviisi. Kuna PyPI-st installeeritud kolmandate osapoolte teeki on keeruline ise parandada ja peab lootma teegi haldaja/arendaja peale ning tegelikult pole käesoleva lahenduse puhul ka valmiskirjutatud pyLora lisafunktsionaalsusi tarvis, siis sai tehtud otsus kasutada pySX127x projekti, kloonides koodirepo otse GitHub-ist.

5.1 pySX127x

PySX127x pakub Pythonis kirjutatud liidestust Semtechi 1276/1277/1278/1279 LoRa transiivritele. Projekt loodi 2015. aastal ja viimane üleslaadimine on (10.12.20 seisuga) tehtud koodireposse 2019. aasta oktoobris. Arendaja nimetab Pythoni eelisteks C/C++ kirjutatud teekide ees just kiiret prototüüpimist ja testimist. [16]

¹ <https://pypi.org/project/pyLoRa/>

² <https://github.com/mayeranalytics/pySX127x>

PySX127x vajab töötamiseks kahte teeki: LoRa transiiveriga suhtlemiseks üle SPI-liidese teeki `Spidev`¹ ja RPi GPIO klemmide kontrollimiseks teeki `RPi.GPIO`². Mõlemad on PyPI distros saadaval.

Pakett on arendatud objektorienteeritud programmeerimise põhimõtteid silmas pidades ja prototüübi loomiseks on vajalikud failis `board_config.py` asuv `BOARD` klass ja failis `LoRa.py` asuv `LoRa` klass. Lisaks on `constants.py` failis sisalduvates abiklassides registrite väärtused koos nimedega. `BOARD` klass on vajalik mooduli ja RPi-vahelise ühenduse seadistamiseks – siin määratakse ära, milliseid GPIO klemme kasutatakse DIO0...3 korral, seotakse nendega `LoRa` klassis olevad *callback*-funktsioonid, mis visatakse SX1276 katkestuste puhul ja luuakse SPI-ühendus `LoRa` mooduli ja RPi vahel. `LoRa` klassis on ära kirjeldatud meetodid, mille abil suheldakse `LoRa` mooduli registritega. Enamasti on need defineeritud *getter/setter* meetoditena, mis tähendab, et *setter*-ite abil saab edastada registritesse informatsiooni ja *getter*-ite abil saab pärida registrite sisu.

5.2 Lüüs (*gateway*)

Lüüsi tööd juhtiv programm on kirjutatud faili `receiver.py`³. Selles on `Receiver`-nimeline klass, mis on `LoRa` alamklass ja pärib sealt kõik meetodid. Objekti loomisel kutsutakse välja `BOARD.setup()`, mis seadistab ühenduse mooduli ja RPi vahel, tehakse moodulile restart (vajalik, kuna programmi mittekorrektusel lõpetamisel võib moodul jääda valesse olekusse ja uuesti käivitamisel veateateid andma hakata), `LoRa` klassi konstruktor seadistab ära registrid ja seadistamise lõpus pannakse moodul *SLEEP*-režiimi.

Kuna erinevad seadistamisparameetrid, nagu töösagedus, kanali ribalaius, mooduli CR (*Coding Rate*), laotustegur (SF) ja mooduli maksimaalne väljundvõimsus, on seadistatavad läbi `parameters.ini`, loetakse see samuti sisse `Receiver` objekti loomisel. Parameetrite hoidmine eraldi failis tekstilisel kujul annab võimaluse hõlpsalt neid muuta ka kasutajal, kes koodi lugeda ei oska.

¹ <https://pypi.org/project/spidev/>

² <https://pypi.org/project/RPi.GPIO/>

³ Kogu kood on kättesaadav ja nähtav autori GitHub repos: https://github.com/oliverpaadik/kohvi_lora

Receiver-klassis on üle kirjutatud LoRa klassi meetodid, mis on seotud mooduli DIO0...3 tulevate signaalidega ehk LoRa transiiveri katkestustega. Need meetodid on järgnevad:

- **on_rx_done** – meetod kutsutakse välja siis, kui transiiver viskab RxDone katkestuse ehk antakse märku saabunud sõnumist;
- **on_tx_done** – meetod kutsutakse välja siis, kui transiiver viskab TxDone katkestuse ehk antakse märku saadetud sõnumist;
- **on_cad_done** – meetod kutsutakse välja siis, kui transiiver viskab CadDone katkestuse ehk antakse teada, et CAD (*Channel Activity Detection*) protsess on lõpetatud. CAD kasutatakse, et hoida LoRa vastuvõtjat energiasäästlikus režiimis ja samal ajal kontrollida, et äkki on kanalis sõnumeid, mida peaks vastu võtma (sel juhul visatakse ka CadDetected katkestus);
- **on_rx_timeout** – kutsutakse välja RxTimeout katkestuse puhul, mis visatakse RXSINGLE (ühe sõnumi vastuvõtmine) režiimis pärast programmeeritava vastuvõtuakna lõppu;
- **on_valid_header** – kutsutakse välja ValidHeader katkestuse puhul ehk märgib saabunud sõnumi korrektset päist;
- **on_payload_crc_error** – seotud PayloadCrcError katkestusega, mis visatakse kui saabunud sõnum ei läbi CRC kontrolli.

Töövoog on üldjoontes järgnev: programmi käivitamisel seatakse FIFO andmepuhvri aadress-pointer FifoAddrPtr puhvri algusesse, mis tähendab, et moodul on valmis sõnumite vastuvõtmiseks. Seejärel pannakse moodul RXCONT-režiimi (ja programm läheb lõputusse tsüklisse. Tsüklist väljutakse, kui tuvastatakse RxDone katkestus, mis tähendab, et moodul on lõpetanud paketi vastuvõtmise. Logitakse RSSI ja SNR väärtused ja nullitakse RxDone lipp. Seejärel tehakse CRC kontroll ja kui vigasid ei tuvastata, loetakse sisse sõnum. Sõnum on saadetud CSV-formaadis sõnena, kus esimene element on saatja identifikaator ja sellele järgnev informatsioon võetud jookide kohta (võetud jook, kuupäev ja kirje transaktsiooni id). Näiteks 8. detsembril masinast nimega rpi-zero võetud kakao, kuuma vee ja kohvi võtmise järel genereeritud sõne oleks järgmine:

```
rpi-zero;876,cocoa,2020-12-0810:34:03;631,hot water,2020-12-08 10:36:03;1716,coffee,2020-12-08 10:38:02;
```

Saadud sõne jagatakse semikooloni järgi väiksemateks tähenduslikeks osadeks ja salvestatakse nad kirjetena andmebaasi. Kui andmebaasi salvestamine õnnestus, edastatakse lõppsõlmele kinnitav sõnum „ACK“ ja lülitatakse transiiver tagasi RXCONT-töörežiimi. Kogu tööprotsess on esitatud voodiagrammina Lisas 1.

5.3 Lõppsõlm (*endpoint*)

Lõppsõlmega seonduv on klassis Transmitter, mis asub failis transmitter.py. Sarnaselt lüüsiga on see LoRa klassi alamklass ja pärib sealt kõik tööks vajalikud meetodid. Lõppsõlme seadistamine toimub samadel põhimõtetel lüüsiga, kus Transmitter objekti loomisel seadistatakse nii ühendus mooduliga kui ka kanali parameetrid (failist parameters.ini).

Tabel 3. Andmebaasi tabeli motor_log struktuur

Veerg	Andmetüüp	Null lubatud	Primaarvõti
id	int(11)	Ei	Jah
motor_name	varchar(10)	Jah	
start_time	datetime	Jah	
machine	varchar(15)	Jah	
transmit	datetime	Jah	
external_id	int(11)	Jah	

Esimese asjana luuakse ühendus andmebaasiga ja loetakse tabelist motor_log (ülesehitus on kirjeldatud Tabelis 3) uued kirjed (veerus transmit on null väärtus). Kuna LoRa andmepaketi maksimaalseks suuruseks on 255 baiti, siis päritakse andmebaasist maksimaalselt 6 kirjet. Maksimaalne kirjete arv on välja arvatud tabeli motor_log veergude maksimaalse suuruse järgi, arvestades, et masina nimi (N_{machine}) lisatakse saadetavale sõnele vaid ühe korra:

$$\frac{255 - N_{machine}}{N_{id} + N_{motor_name} + N_{start_time}} = \frac{255 - 15}{(11 + 10 + 19)} = 6 \text{ [kirjet]}$$

Kirjetest genereeritakse CSV-formaadis sõne, mille ette pannakse masina nimi. Sõne saadetakse transiiveri FIFO andmepuhvrissi ja lülitatakse ümber TX töörežiimi peale, mille järel sõnum saadetakse välja. Seejärel lülitatakse transiiver RXCONT režiimi ja oodetakse kinnitavat ACK-sõnumit. Kinnitust oodatakse 15 sekundit ja kui see saabub, registreeritakse RSSI ja SNR väärtus, nullitakse RxDone lipp ja eelnevalt kokku kogutud kirjetele lisatakse transmit veergu ajatempel. Kui 15 sekundi jooksul ACK-sõnumit ei saabu, eeldab programm, et andmed ei jõudnud kohale ja ei märgi kirjeid saadetuks. Järgmisel korral, kui programm käivitatakse, korjatakse samad andmed uuesti andmebaasist ja üritatakse neid uuesti saata. Töövoog on kirjeldatud Lisas 2 olevas voodiagrammis.

5.4 Käivitusprogramm

Nii lõppsõlme kui lüüsi programmi saab käivitada läbi käivitusprogrammi, mis asub failis runner.py. Selle saab käivitada läbi käsurea, kasutades parameetreid, mis määravad ära, kas masin käitub lüüsina või lõppsõlmena. Kasutades parameetrit T (või t) läheb käima lõppsõlme programm ja parameetri R (või r) kasutamisel käivitatakse lüüsi programm. Lisaks sellele on võimalik sisse lülitada silumisrežiim, mille läbi programm töötamise ajal kasutajale rohkem informatsiooni protsesside kohta annab – see käib läbi parameetri --debug (või -d).

Näiteks kasutades programmi käivitamiseks käsklust „python3 runner.py -d r“, käitub masin lüüsina ja annab töö jooksul kasutajale lisainformatsiooni tegutsemise kohta.

5.5 LoRa ühenduse parameetrid

Vastavalt Euroopa Liidu lähitoimeseadmete regulatsioonidele on 868-868,6 MHz sagedusalas töötavad lähitoimeseadmetele lubatud maksimaalseks efektiivseks kiirgusvõimsuseks (ERP) 25 mW ja töösükli piiranguks 1% [17]. Sellele tuginedes on parameters.ini failis vaikimisi seadistatud masinate vahelise ühenduse sagedusalaks 868,3 MHz.

Selleks, et väljundvõimsus jääb lubatud piiridesse, on vaja kõigepealt lubatud 25 mW teisendada detsibellidesse:

$$P = 10 \times \log_{10}(P [mW]) = 10 \times \log_{10}(25) = 13,98 [dBm]$$

See tähendab, et LoRa mooduli väljundvõimsus ja antenni võimendus peavad jääma selle alla. MIKROE-2350 antenni võimendus on 2 dBi, mis tähendab, et mooduli maksimaalne väljundvõimsus peab jääma alla 11,98 dBm. Kuna moodulit saab seadistada vaid täisarvudega, on konfiguratsioonifailis vaikimisi selle väärtuseks 11.

Kanali ribalaiuseks on vaikimisi määratud 250 kHz, laotusteguriks 10 ja koodimiskiiruseks 4/8. Laotustegur määrab ära selle, kui pika aja peale on venitatud saadetud informatsioon ehk suurem laotusteguri väärtus tähendab suuremat kanali kasutusaega ja väiksemat andmeedastuskiirust. Koodimiskiirus fikseerib ära selle, kui palju on saadetaval informatsioonil lisabitte ennetava veaparanduse (*forward error correction*) jaoks ehk 4/8 tähendab seda, et pool paketist on andmed ja pool bitid veaparanduse jaoks. Suurem number nimetajas tähendab suuremat veakindlust ühenduse puhul. Nende parameetrite põhjal on võimalik arvutada maksimaalse andmeedastuskiiruse:

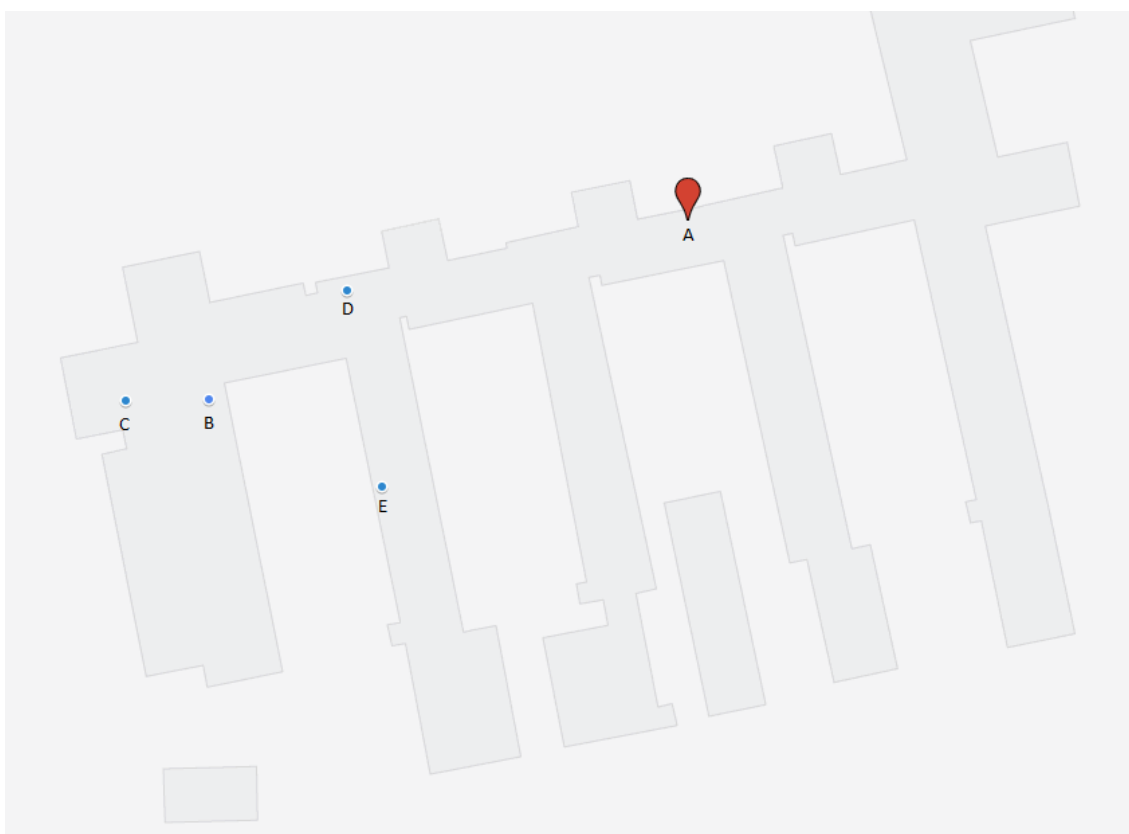
$$R_b = SF \times \frac{BW}{2^{SF}} \times CR = 10 \times \frac{250000}{2^{10}} \times \frac{4}{8} = 1220,70 [bit/s] \approx 1,2 [kbit/s]$$

Kasutades LoRaTools eetriaja kalkulaatorit¹ on võimalik nende väärtuste põhjal kergelt välja arvutada, kui palju kulutakse ühe 255 baidise sõnumi saatmiseks eetriaega ja kui pikad peavad olema eetrisoleku-vahelised perioodid 1%-lise töötütsükli piirangu puhul. Kasutades eelnevalt mainitud parameetreid saame maksimaalse suurusega sõnumi puhul eetriajaks 1,79 sekundit. Saatmiste vahele peab jääma vähemalt 2 minutit ja 59 sekundit – seda informatsiooni saab ära kasutada, kui seadistada masinate kanali kasutamise kord.

¹ <https://www.loratoools.nl/#/airtime> (külastatud 20.12.2020)

6 Testimine

Testimiseks kasutati kahte RPi-d, millest üks lüüsina käituv masin oli statsionaarne ja lõppsõlmena käituvat masinat liigutati erinevatesse kohtadesse TTÜ peamajas, mis on kujutatud Joonisel 5. Punkt B on esimesel korrusel ja vahemaa punktiga A on ca 116 meetrit. Punkt C on teisel korrusel ja vahemaa punktiga A on ca 134 meetrit. Punkt D asub neljandal korrusel ja on ca 79 meetri kaugusel punktist A. Viimane testimiseks kasutatud asukoht E asub punktist A 92 meetri kaugusel ja on samuti 4. korrusel. Statsionaarne RPi kuulas pidevalt kanalit ja tema poole saadeti pakette informatsiooniga, mis emuleerib reaalselt kasutusjuhtu ehk andmebaasi sisestatud jooke koos ajatemplitega. Saadetud sõnumite puhul mõõdeti ka RSSI-d ja signaali-müra suhet (SNR).



Joonis 7. Testimiseks kasutatud asukohad TalTech peahoones

Punktis B ehk samal korrusel asuvas punktis oli võimalik sõnumeid vahetada kõige väiksema laotusteguriga (SF=9), seejuures jõudsid kohale kõik sõnumid (n=15). Keskmine signaali-müra suhe oli seejuures -9,7 dB ja kõige suurem -10,75 dB ehk kõik

saadetud sõnumid jäid kindlalt alla dokumentatsioonis laotustegurile $SF=9$ määratud signaali-mürasuhte piirmäärale $-12,5$ dB [15]. See on väikseim suurus, mida vastuvõtja suudab demoduleerida. Väiksema laotusteguriga ($SF=8$) õnnestus küll sõnumeid lüüsi saata, kuid vastu saadetud ACK lõppsõlmeni ei jõudnud. Punkt C asus B-st korrus kõrgemal ja paarkümmend meetrit kaugemal ning sellest piisas, et $SF=9$ puhul kahe masina vaheline ühenduse töökindlus tuntavalt kannataks – õnnestunuks võis lugeda 50% sooritatud saatmistest. Nagu ka eelmise katsega, jõudis paljude ebaõnnestunud saatmiste korral pakett küll lüüsinini, kuid kinnitav vastus lõppsõlmeni ei jõudnud. Tõstes laotustegurit ($SF=10$), muutus ka kahe masina vahelise ühenduse töökindlus 100%-liseks.

Punkt D sai valitud, et testida, kuidas töötab LoRa ühendus juhul, kui masinate vahel on mitu korrust – antud katse puhul asus üks masin 1. korrusel ja teine masin 4. korrusel. Kõige väiksem laotustegur, mil kohale jõudsid kõik sõnumid oli $SF=10$. SNR oli nii lõppsõlmes kui ka lüüsis valdavalt positiivne või nullilähedane (dokumentatsiooni määratud piirmäär oli $SF=10$ puhul -15 db), mistõttu võib antud laotusteguri suurust ka vaikeseadistuses kasutada, kuna katab ära ülesande püstituses seatud eesmärgi luua ühendus nii 100 meetri kaugusel kui ka erinevatel korrustel asuvate masinate vahel. Punkt E puhul ei jõudnud lõppsõlmest lüüsi kohale ükski sõnum, seda ka kõige suurema laotusteguri puhul. Siin võis olulisteks faktoriteks olla kombinatsioon kaugusest kui ka raadiosignaali sumbuvalt põhjustavatest keskkondlikest teguritest (signaal pidi läbima nii mitme korruse betoonist põranda kui ülikoolihoone kahe tiiva seinad).

Kuna lahenduse puhul kasutatakse ühte kanalit, tuli testida ka seda, et kaks või enam masinat saaks edukalt hakkama kanali jagamisega nii, et ükski masin teise andmete edastamist/vastuvõtmist ei segaks. Selleks kasutati Linuxi tööriista `crontab`, mis lubab kasutajal ära seadistada programmide/skriptide käimapanekud kas siis kindlal kellaajal või teatud intervallide tagant. Testimiseks pandi kodustest tingimustes (masinate vaheline kaugus üksteisest *ca* 10 m) üks masin tööle lüüsinina ja kaks masinat tööle lõppsõlmedena. Peatükis 5.5 välja toodud töötsükli piirangutele tuginedes seadistati lõppsõlmed andmeid saatma nii, et üks saadab andmeid 4. minutil, teine 8. minutil, esimene 12. minutil jne. Peaaegu 24-tunnise töö ajal selle puhul mingisuguseid tõrkeid ei ilmnunud.

7 Edasised võimalikud arendused

Projektiga on algusest peale kaasas käinud eeldus, et lõplik lahendus peab olema võimalikult taskukohane. Seetõttu võiks mõelda, kuidas oleks võimalik kaotada kõige kallim komponent – Raspberry Pi. Olgugi, et RPi on piisavalt võimekas, et jooksutada nii andmebaasi, kui ka samal ajal paralleelselt tegeleda andmete vastuvõtmisega ja saatmisega internetti, ei suuda ta vähemalt hinna poolest konkureerida Arduino või Espressif Systems mikrokontrolleritega.

Üheks potentsiaalseks alternatiiviks oleks näiteks Arduino Uno Rev3, mis maksab *ca* 22 eurot¹ ja võimaldab samuti kasutada SPI-liidest LoRa mooduliga suhtlemiseks. Praeguse MySQL/MariaDB andmebaasi annaks asendada näiteks sardseadmete piiratud ressursse arvestava relatsioonilise andmebaasiga LittleD². Arduinole on tehtud ka mitmeid teekes, mis teevad LoRa ühendust kasutava lahenduse arendamise lihtsamaks, näiteks LoRa³. Espressif ESP32 mikrokontrollerid on veelgi odavama hinnaga, Mouseris on saadaval ESP32-S2-Saola-1R mudel hinnaga 6,78 eurot⁴ ja leidub ka erinevaid kloone, mille hind on veelgi odavam. ESP32 kasutamine tähendaks samuti seda, et tuleks vaadata sardsüsteemile mõeldud andmebaasi poole.

Mõne teise mikrokontrolleri peale lahenduse tegemine eeldab muidugi järjekordset ajakulu, kuna olemasolevat lähenemist (Pythonis kirjutatud programm + andmebaas) enam kasutada ei saaks. Eeldatavasti vajab see juhtumi- ja kasutuspõhist analüüsi – suuremate masinate arvu puhul oleks arvatavasti mõistlik vähemalt lõppsõlmed ehitada mõne odavama mikrokontrolleri peale ja jätta andmete talletamine lüüsi rollis talitava RPi hoolde; väiksemate masinate arvu puhul on mõistlik kasutada olemasolevat lahendust.

¹ <https://www.mouser.ee/ProductDetail/Seed-Studio/102990189/?qs=W0yvOO0ixfEl8v9TJdk2SA%3D%3D> (külastatud 18.12.20)

² <https://github.com/graemedouglas/LittleD> (külastatud 28.12.20)

³ <https://www.arduino.cc/reference/en/libraries/lora/> (külastatud 19.12.20)

⁴ <https://www.mouser.ee/ProductDetail/Espressif-Systems/ESP32-S2-Saola-1R/?qs=GBLSl2AkirsMIxW%2FKjKKRw%3D%3D> (külastatud 28.12.20)

8 Kokkuvõte

Töö eesmärk oli valida sobivad tööriistad ja valmis ehitada IoT lahendus, kus kohvimasinatele paigaldatud RPi-l põhinevad andmeloggerid saaksid omavahel suhelda väljaspool tavapärasest kohtvõrku (LAN, Ethernet). Peamised eesmärgid lahenduse puhul olid seejuures:

- saavutada masinate vaheline ühendus ja andmete edastus 100 meetri pealt;
- ühendus peab toimima ka keskkonnas, kus masinate vahel on takistused (seinad ja/või korrused)

Sobiva tehnoloogia valimisel võrreldi nelja erinevat traadita andmeside tehnoloogiat/protokolli – WiFi, Bluetooth, ZigBee ja LoRa. Peamisteks valikukriteeriumiteks olid seejuures võimalik leviala, mida protokoll või tehnoloogia toetab, kasutuslihtsus (teekide olemasolu, eelnevalt tehtud projektid) ja hind. Selle põhjal sai valituks LPWAN tehnoloogia LoRa, mis teoreetiliselt võimaldab hoonestatud keskkonnas sõnumeid saata kuni 3 km kaugusele.

Lahenduse arhitektuur näeb ette, et kasutatakse kahte tüüpi masinaid – üks on keskne seade ehk lüüs, mis on SIM800 mooduli abil ühenduses internetiga ja ülejäänud võrgus olevad seadmed on lõppsõlmed ehk seadmed, mis saadavad teatud intervallide tagant andmeid lüüsi poole. Masinate vaheline ühendus luuakse LoRa mooduli abil, mis kasutab Semtech SX1276 arhitektuuril põhinevat kiipi. Funktsionaalsust kontrolliv programm kirjutati valmis Pythoni keeles ja kasutati peamiselt pySX127x teeki, mis hõlbustas suhtlemist LoRa mooduli registritega.

Testimise jaoks seadistati üks lüüsi rolli täitev masin, mis oli TalTechi peahoones statsionaarne ja erinevate katsete läbi liigutati teist, lõppsõlme rollis olevat, masinat. Testimise tulemusel leiti optimaalsed parameetrid, mida vaikeseadistuses kasutada ja veenduti, et LoRa tehnoloogia võimaldab tõepoolest hoonestatud üksteisest 100 meetri kaugusel asuvates seadmetes andmeid edastada. Testiti ka kanali jagamist aja põhjal, mis simuleerib reaalselt kasutusjuhtu, kus erinevad masinad peavad sama kanali peal lüüsi andmeid saatma.

Kasutatud kirjandus

- [1] i-SCOOP, „Comprehensive and visual classification of wireless IoT protocols and network technologies,“ i-SCOOP, [Võrgumaterjal]. Available: <https://www.i-scoop.eu/internet-of-things-guide/wireless-iot-protocols-technologies/>. [Kasutatud 29 September 2020].
- [2] S. Sendra, M. Garcia, C. Turro ja J. Lloret, „WLAN IEEE 802.11a/b/g/n Indoor Coverage and Interference Performance Study,“ *International Journal on Advances in Networks and Services*, kd. IV, nr 1 & 2, pp. 209-222, 2001.
- [3] N. Ahmed, H. Rahman ja M. Hussain, „A comparison of 802.11ah and 802.15.4 for IoT,“ *ICT Express*, kd. II, nr 3, pp. 100-102, 2016.
- [4] B. Hashim, „Industry’s First 802.11ah (HaLow) Wireless Module for IoT Devices,“ Silex Technology, 18 February 2020. [Võrgumaterjal]. Available: <https://www.silextechnology.com/unwired/industrys-first-802.11ah-halow-wireless-module-for-iot-devices>. [Kasutatud 05 November 2020].
- [5] „RPI USB Wi-Fi Adapters,“ [Võrgumaterjal]. Available: https://www.elinux.org/RPi_USB_Wi-Fi_Adapters. [Kasutatud 28 September 2020].
- [6] Jaycon Systems, „Bluetooth Technology: What Has Changed Over The Years,“ 28 September 2017. [Võrgumaterjal]. Available: <https://medium.com/jaycon-systems/bluetooth-technology-what-has-changed-over-the-years-385da7ec7154>. [Kasutatud 28 September 2020].
- [7] A. Nguyen, „Bluetooth 1.0 vs 2.0 vs 3.0 vs 4.0 vs 5.0 - How They Compare,“ Symmetry Electronics, 18 April 2018. [Võrgumaterjal]. Available: <https://www.semiconductorstore.com/blog/2018/Bluetooth-1-0-vs-2-0-vs-3-0-vs-4-0-vs-5-0-How-They-Differ-Symmetry-Blog/3147>. [Kasutatud 4 October 2020].
- [8] L. Frenzel, „5 Things You Must Know About the New Bluetooth 5,“ 4 January 2017. [Võrgumaterjal]. Available: <https://www.electronicdesign.com/technologies/wireless/article/21802861/5-things-you-must-know-about-the-new-bluetooth-5>. [Kasutatud 4 October 2020].
- [9] Y. Kim ja I. Moon, „ZigBee and IEEE 802.15.4 Standards,“ %1 *ZigBee Network Protocols and Applications*, London, CRC Press, 2014, pp. 32-52.
- [10] Digi International Inc, „XBee® SX 868 RF Module User Guide,“ 2018. [Võrgumaterjal]. Available: <https://cms.digi.com/resources/documentation/Digidocs/90001538/>. [Kasutatud 5 October 2020].

- [11] G. Yang, „Background Introduction,“ %1 *ZigBee Network Protocols and Applications*, London, CRC Press, 2014, pp. 2-29.
- [12] „About Us: LoRa Alliance,“ 2020. [Võrgumaterjal]. Available: <https://loralliance.org/about-lora-alliance>. [Kasutatud 24 October 2020].
- [13] A. Augustin, J. Yi, T. Clauzen and W. M. Townsley, “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,” *Sensors*, vol. IX, no. 16, pp. 1-18, 2016.
- [14] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui ja T. Watteyne, „Understanding the Limits of LoRaWAN,“ *IEEE Communications Magazine*, kd. 55, nr 9, pp. 34-40, 2017.
- [15] Semtech Corporation, “SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver datasheet,” May 2020. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoK FfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE. [Accessed 4 November 2020].
- [16] Mayeranalytics, „pySX127x Readme,“ October 2015. [Võrgumaterjal]. Available: <https://github.com/mayeranalytics/pySX127x/blob/master/README.md>. [Kasutatud 10 December 2020].
- [17] Euroopa Liidu Teataja, „KOMISJONI RAKENDUSOTSUS (EL) 2017/1483,“ 18 August 2017. [Võrgumaterjal]. Available: http://data.europa.eu/eli/dec_impl/2017/1483/oj. [Kasutatud 16 December 2020].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

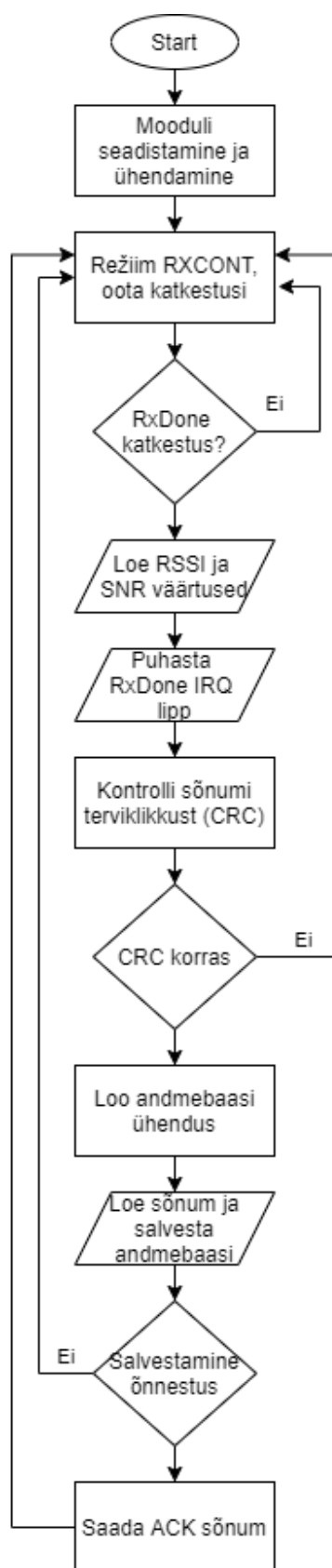
Mina, Oliver Paadik

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Kohvimasina andmeloggerite ühendamine LoRa tehnoloogiat kasutades“, mille juhendaja on Uljana Reinsalu
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

04.01.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Vastuvõtja voodiagramm



Lisa 3 – Saatja voodiagramm

