

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
IT Kolledž

Aleksandr Pikalo
175264IDAR

Teehoolduse valgusfoori töörežiimide automatiseerimine

Diplomitöö

Juhendaja: Kalle Tammemäe
tehnikateaduste doktor

Konsultant: Roman Pikalo
Eesti Infotehnoloogia
Kolledži vilistlane

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: Aleksandr Pikalo

17.05.2021

Lõputöö ülesandeleht

Lõputöö autor sattus situatsiooni, kus teeremondi kohas keset ööd pidi seisma 15 minutit, kuid tegelikult teeremonti sellel hetkel ei toimunud ning peale seda tuligi autoril mõte seda probleemi lahendada tehes sellest lõputöö. Tööülesandeks oli luua teehoolduse valgusfoori juhtsüsteemi prototüüp, mille abil oleks võimalik valgusfooride töörežiimide automaatne juhtimine sõltuvalt teetöölise olemasolust objekti. Kasutades tööliste poolt kantavaid autonoomseid raadiomajakaid saab raadiovastuvõtja infot, mis töörežiimil peab valgusfoor töötama. Andmevahetuseks tuleb kasutada litsentsivaba raadiosagedust. Keskne juhtseade koos raadiovastuvõtjaga peab säilitama autonoomse töövõime vähemalt kaheks ööpäevaks, millele vastavalt tuleb valida toiteaku energiamahutavus. Kaasaskantav raadiomajakas peaks töötama autonoomselt vähemalt 10 tundi ning olema võimalikult väikeste mõõtudega, et mahuks vabalt näiteks rinnataskusse. Kuna süsteemi toimimise huvides on vajalik, et iga objektile olev teetööline kannab raadiomajakat, peab tagama, et majaka mass oleks väike, soovitatavalt kuni 100 grammi. Prototüüp peab demonstreerima töövõimet 100–200 m pikkusel teelõigul kuiva ilma korral, prototüübi tolmu- ja niiskuskindlus pole nõutav. Prototüübi loomisel on oluline pöörata tähelepanu energiatarbele, kuna süsteemi rakendatakse välitingimustes, kus reeglina tuleb kasutada autonoomset toidet.

Annotatsioon

Tänapäeval puudub teehoolduses kasutatavatel liiklusfooridel info hooldustööde tegeliku käigu suhtes, seetõttu ei muutu fooride töörežiim automaatselt, kui teehooldajad objektilt lahkuvad. Antud lõputöö prototüüp oli kokku pandud selleks, et vähendada teetöölise lahkumisel objektilt ooteaega väikestel teeremondi territooriumitel, mille ulatus oleks 100–200 meetrit.

Lõputöö eesmärgiks oli luua raadiovastuvõtja ehk valgusfoori ja raadiosaatja prototüüp, millega oleks võimalik valgusfoori töörežiimi automatiseerida. Töö käigus arendati välja modifitseeritud valgusfoori automaatika koos raadiovastuvõtja ja raadiosaatjatega, ning seadistati need tarkvaraliselt omavahel.

Teeremondi piirkonnas on igal töötajal üks NodeMCU V3 Lolin raadiosaatja moodul, mis suhtleb sama NodeMCU V3 Lolin raadiovastuvõtja mooduliga, mis on paigutatud teeremondi piirkonnas olevatesse valgusfooridesse. Lõputöö prototüübi töö ulatus oli 150 meetri. Raadiovastuvõtja tööaeg akuga 10000 mAh 5 V oli 2 päeva 23 tundi ja 15 minutit temperatuuril + 20°C (Celsius) ning raadiosaatja tööaeg akuga 1350 mAh 3,7 V oli kuni 21 tundi temperatuuril + 20°C.

Lõputöö on kirjutatud eesti keeles, ning sisaldab teksti 37-1 töö põhiosa leheküljel, 7 peatükki, 11 joonist, 5 tabelit.

Abstract

Automating the Operating Modes Of The Traffic Lights For Road Maintenance.

Nowadays, traffic lights used in road maintenance do not have information about the actual progress of maintenance work. Which means that the mode of operation of traffic lights does not change automatically when road maintenance workers leave the site. The capacity of the road section will not change and driver will be forced to wait for the right to drive behind the traffic lights, although regulation is no longer needed. This diploma thesis describes how automated traffic lights system was engineered and built to handle traffic during road repairs in small repair area of 100 –200 meters.

The aim of the diploma thesis was to create a prototype with which it would be possible to automate the operation mode of a traffic light. In the course of the work, the modified traffic light automation with the receiver and the transmitters was developed, and they were configured together in software.

In the repair area, each employee has one NodeMCU V3 Lolin radio transmitter module that communicates with the same NodeMCU V3 Lolin receiver module and is located in traffic lights in the repair area. The prototype of the diploma thesis completed its task and its work area is up to 150 meters. According to conducted experiments, the operating time of the receiver with a lithium ion battery of 10,000 mAh 5V was 2 days 23 hours and 15 minutes at +20°C (Celsius). The operating time of the transmitter with a 1350 mAh 3.7V battery was up to 21 hours and at a temperature of +20°C.

The thesis is in Estonian, and contains 37 pages of text, 7 chapters, 11 figures, 5 tables.

Lühendite ja mõistete sõnastik

AES	<i>Advanced Encryption Standard</i> , Täiustatud krüptimise standard
Arduino IDE	<i>Arduino Integrated Development Environment</i> , Arduino integreeritud arenduskeskkond
I2C	<i>Inter-Integrated Circuit</i> , Kahejuhtmeliides
LCD	<i>Liquid Crystal Display</i> , Vedelkristallekraan
LED	<i>Light-Emitting Diode</i> , Valgusdiod
MAC	<i>Media Access Control Address</i> , Füüsiline aadress
Mac OS	<i>Macintosh Operating System</i> , Macintosh operatsioonisüsteem
MPPT	<i>Maximum Power Point Tracking</i> , Maksimaalne võimsuspunkti jälgimine
MQTT	<i>Message Queuing Telemetry Transport</i> , Sõnumite järjekorrastamis protokoll
RF	<i>Radio Frequency</i> , Raadiosagedus
RFID	<i>Radio Frequency IDentification</i> , Raadiosagedustuvastus
SSID	<i>Service Set Identifier</i> , Traadita võrgu nimi
TKIP	<i>Temporal Key Integrity Protocol</i> , Ajaline võtmetervikluse protokoll
USB	<i>Universal Serial Bus</i> , Universaalne jadaliides
WiFi	<i>Wireless Fidelity</i> , Traadita andmeside standard
WPA	<i>WiFi Protected Access</i> , WiFi kaitstud juurdepääs

Sisukord

1. Sissejuhatus	11
2. Automatiseerimine	13
2.1 Automatiseerimise aktuaalsus	14
2.2 Olemasolevad lahendused	14
2.2.1 “KOMKON SDM“ raadiojuhitav valgusfoor.....	15
2.2.2 Turvalisus raudteel	15
2.3 Lõputöö lahenduse tööpõhimõte	16
3. Moodulite valik ja analüüs	18
3.1 Raadiolainete levimine	18
3.2 SA100 Dual/IP66 Remote EC moodulite katsetus	19
3.3 433 MHz moodulite katsetus	19
3.4 Bluetooth moodulite katsetus	21
3.5 NodeMCU V3 Lolin mooduli katsetus.....	21
3.6 Raadiomoodulite võrdlus.....	22
4. Lahenduse kirjeldus	24
4.1 NodeMCU V3 mooduli plussid.....	24
4.2 Moodulite kirjeldus	25
4.2.1 NodeMCU V3 moodul	25
4.2.2 RTC moodul	26
4.3 Moodulite programmeerimine	27
5. Eksperimendi kirjeldus	30
5.1 NodeMCU V3 Lolin töökäik.....	30
5.2 Valmis prototüübi katsetamine.....	36
5.3 Programmeerimisega kaasnenud probleemid.....	38
5.3.1 Mac operatsioonisüsteem	38
5.3.2 Ooteaja lihtsustatud seadistamine.....	39
6. Prototüübi parendamise võimalused.....	41
6.1 Lahendused foori autonoomse tööaja pikendamiseks	41
6.2 Antenn	43
6.3 Numbripaneel ja patarei nupp.....	43
6.4 LCD	44
6.5 MAC aadress ja ooteaja seadistamine	44

6.6 Prototüübi veekindlus ja vastupidavus	45
6.7 Prototüübi pilveteenusega ühendamine	45
6.8 Prototüübi võrgu turvalisus	46
7. Kokkuvõte	47
Kasutatud kirjandus	48

Jooniste loetelu

Joonis 1. Lahenduse tööpõhimõte	16
Joonis 2. Raadiovastuvõtja programmeerimis algoritm	29
Joonis 3. Raadiosaatja programmeerimise algoritm	29
Joonis 4. Raadiovastuvõtja ühendamise skeem.....	30
Joonis 5. Raadiosaatja ühendamise skeem.....	31
Joonis 6. Prototüübi raadiosaatja koos akupangaga	32
Joonis 7. Prototüübi valgusfoor koos akupangaga ning RTC mooduliga.....	32
Joonis 8. Tarkvara üleslaadimise tekkinud vaeteade Mac OS süsteemis.....	36
Joonis 9. Valgusfoori töörežiimi sisestamine Arduino IDE programmi.....	37
Joonis 10. Arduino IDE programmi valgusfoori töörežiimi sisestamise topelt küsimine.....	37
Joonis 11. Programmis Arduino IDE vastuse printimisel kaasnenud probleem.....	37

Tabelite loetelu

Tabel 1. Raadiomoodulite võrdlus	21
Tabel 2. NodeMCU V3 Lolin tehnilised andmed	24
Tabel 3. RTC mooduli tehnilised andmed	25
Tabel 4. Prototüübi komponendid	26
Tabel 5. Päikesepaneeli komponendid	39

1. Sissejuhatus

Tänapäeval on sõiduteed tähtsal kohal meie elus, kuid ei või ära unustada seda, et neid on vaja hooldada või vajadusel ka remontida. Firmad, kes on spetsialiseerunud teeremontidele, peavad tähele panema mitte ainult seda, et uus tee oleks kvaliteetne, vaid ka seda, kuidas liiklejad saaksid oma teekonda jätkata kohas, kus teeremont käib. Selleks ongi teetööde firmas projekteerijate tööks liiklejatele teekonna lahendamine, kas teeremondist mööda sõites või organiseerides liiklust nii, et liiklejad saaksid oma teekonda jätkata teeremondi kohast nii, et liiklejad ei segaks töölistel teetöid teha. Autojuhtide jaoks pannakse teetööde alas ajutised märgid, mis on juhtidele abiks, et nende liiklus oleks minimaalselt häiritud. Mõningatel juhtudel tuleb teetöolistel panna ikkagi mõlemad suunad kinni ning sellisel juhul projekteerijad puutuvad kokku probleemiga, kus tuleb otsida kaardilt uus tee, mille kaudu liiklejad saaksid teeremondist mööda sõita. Nii saavad töölistel teha oma tööd maksimaalselt, et liiklejad neid ei segaks. Alati ei ole võimalik otsida uut teekonda, siis tuleb firmal panna ajutised reguleerijad, kes korraldavad liiklust sellel tee vahemikul. Reguleerijate ülesandeks võib-olla veel täielik liikluse peatamine vajalikul ajal lühikest aega, näiteks mingi transpordi ühest kohast teise transportimisel, kuid selleks võib olla ka teetööline, kes pole spetsialiseerunud reguleerimisele. Pikema aja reguleerimiseks on ikka välja koolitatud inimene, kes on spetsialiseerunud liikluse reguleerimisele ning, kes näitab juhtidele kuhu ja kuna nad võivad sõita. Mõned firmad aga kasutavad spetsiaalseid valgusfoore, mis reguleerivad oma signaalidega liiklejate liiklust — seda, kas juhil on luba sõita edasi või mitte. Tavaliselt on need valgusfoorid programmeeritud mingi teatud ajaintervalli peale, ehk mingi aeg põleb punane tuli ja seejärel sama pikalt põleb roheline tuli. Sama süsteem on ka vastassuuna liiklejatele.

Töö autor viis läbi intervjuu ühe teedehitis firma objektijuhiga, ning vestluse käigus tuli välja see, et hetkel kasutuses olevad valgusfoorid töötavad ainult ühes režiimis. Isegi siis, kui töö on selleks päevaks lõppenud ning teeremondi piirkonnas pole enam teetöölisi, töötab valgusfoor samamoodi nagu teeremondi ajal. Objektijuhi sõnul ei soovita kasutada reguleerimiseks inimressursi, kuna see on lisakulu. Seega valgusfooride tellimine teatud ajaks on palju soodsam ja efektiivsem, kui tellida reguleerijaid. Intervjuu käigus selgus, et lõputöö prototüüp oleks kasulik teeremondi firmadele. (Intervjuu kestvus 10 minutit, Jõgevamaa, Raja küla, juuli 2019).

Lõputöö eesmärk oli automatiseerida teeremondi ajal liiklust kahesuunalisel teel pikkusega 100–200 meetrit.

Lõputöö ülesanded:

- Luua automatiseeritud valgusfoori prototüüp, mis suudaks reguleerida liiklust teeremondi ajal kasutades raadiomoduleid, sõltuvalt teetöölise olemasolust objektil;
- testida prototüübi raadiovastuvõtja ja -saatja signaali edastuse kaugust ning signaali läbivus läbi erinevate materjalide;
- testida prototüübi tööaega kasutades liitiumioonakut.

Lõputöö on üles ehitatud järgmisel kujul: sissejuhatus, kus tutvustatakse töö teemat, lahendatavat probleemi, töö eesmärki ning ülesandeid. Autor kirjeldab automatiseerimise olulisust ning tegi ülevaate olemasolevatest lahendustest, kus oli automatiseerimisel kasutatud raadiomoduleid. Kasutades saadud infot olemasolevatest lahendusest ning analüüsides kättesaadavaid modulite valikut, valis autor vajaliku mooduli, et lahendada lõputöö probleem. Sellele järgnes praktiline osa ehk eksperimentide kirjeldused. Eelviimaseks osaks toob autor välja soovitusi, kuidas saaks paremaks muuta antud lõputöö käigus valminud prototüüpi. Kokkuvõtte annab ülevaate olulisematest tulemustest ja järeldustest.

2. Automatiseerimine

Automatiseerimine on masinate kasutamine ülesannete tegemiseks, mida eelnevalt on tehtud inimeste poolt või on olnud teostamiseks liiga keerulised. Kui mõistet mehhaniseerimine kasutatakse sageli selleks, et viidata inimtööjõu asendamisele masinatega, siis automaatika siiski eeldab üldjuhul seda, et tulevikus ei peaks kasutama inimtööjõudu. Automaatikat võib määratleda kui tehnoloogiat, mis on seotud protsessi teostamisega programmeeritud käskude abil koos automaatse tagasiside kontrolliga, et tagada juhiste nõuetekohane täitmine. Sellise tehnoloogia areng on muutunud inimesi üha sõltavamaks arvutitest ja arvutitega seotud tehnoloogiate kasutamisest. Selle tõttu on automatiseeritud süsteemid muutunud üha keerukamaks. Täiustatud süsteemid kujutavad endast võimekuse ja jõudluse taset, mis ületab mitmel viisil inimeste võimed samade tegevuste teostamiseks [3].

Lihtne automatiseerimine toimib kasutades algseid ülesandeid, automatiseerides neid. Selline automatiseerimise tase on töö digitaliseerimine tööriistade abil rutiinsete ülesannete sujuvamaks muutmiseks. Äriprotsesside haldus ja robotprotsesside automatiseerimine on põhilised automatiseerimise tüübid [2]. Töörežiimide automaatikasüsteemides määravad ära, kuidas süsteem (või selle alamsüsteemid) töötab vastavalt sellele väljastatud käskudele [1].

Toimingute kogumi määravad ära programmeeritud juhised, mida süsteem peab automaatselt tegema. Vastav programm määrab ära, mida peab tegema automatiseeritud süsteem ning, kuidas erinevad komponendid peavad soovitud tulemuse saavutamiseks toimima. Suhteliselt lihtsates süsteemides koosneb programm piiratud arvust hästi määratletud toimingutest, mida teostatakse pidevalt ja korduvalt õiges järjestuses, kusjuures ühest tsüklis järgmisse ei ole kõrvalekaldeid. Keerulisemates süsteemides võiks käskude arv olla üsna suur ning detailsuse tase igas käsus võiks olla oluliselt suurem. Suhteliselt keerulistes süsteemides näeb programm ette, et toorainete või muude töötingimuste muutustele reageerides tuleb muuta tegevuste järjestust [3].

2.1 Automatiseerimise aktuaalsus

Digitaalne ümberkujundamine, mis jätkab äri - ja tööstustavade ümberkujundamist, on toonud kaasa digitaalsete andmete hulga hüppelise suurenemise, mida tuleb hallata. Eriti infotehnoloogiat on oluliselt mõjutanud ja see on toonud kaasa kriitilise vajaduse automatiseerimise järele. IT-automatiseerimine viitab IT-asjatundjate poolt mitmesuguste IT-alaste ülesannete käsitsi täitmise eemaldamisele ja nende protsesside asendamisele keerukate tarkvaratööriistade ja komplektidega, mis saavad toiminguid automaatselt sooritada. See automaatika saavutati tavaliselt programmilise tagapõhja kaudu ja sagedamini on tänapäeval teostatud tehisintellekti ja/või masinõppe algoritmidega. Rohkem informatsiooni selle kohta on võimalik näha järgmisel lingil [4].

2.2 Olemasolevad lahendused

Selles alapeatükis toob autor välja olemasolevad lahendused, mis on kokku pandud kasutades mooduleid, mis töötavad raadiosagedustel 433 MHz ja 2.4 GHz. Kui võrrelda 2.4 GHz ja 433 MHz raadiomooduleid, siis 433 MHz mooduleid kasutatakse rohkem väikeste seadmete jaoks näiteks, raadiojuhitavad mänguasjad, mõningal juhul õue- ja garaaživäravad. Autori pool teostatud katsetusest lähtuvalt oli FS1000A raadiomoodulite signaali ulatus kuni 20 m, näiteks raadiojuhitavad mänguasjad, kus kasutatakse ühte raadiovastuvõtjat ja ühte raadiosaatjat. 2.4 GHz moodul, näiteks ESP8266 või NodeMCU V3 Lolin on kasutusel rohkem võrgu ehitamiseks suurema signaali raadiusega ilma juurde lisatud antennita. Olemasolevad lahendused on valitud selle põhjal, et need kasutavad litsentsivaba sagedust raadioside parameetrite edastamiseks. Antud alapeatükis on välja toodud lahendused, millega on automatiseeritud liiklust.

2.2.1 “KOMKON SDM“ raadiojuhitav valgusfoor

Töö autor leidis sarnase ülesandega seadme nagu lõputöö prototüüp, kuid nende tööpõhimõtte erineb. Seadme illustreeriv foto on Lisas 5. [39]. Selles seadmes kasutati 433 MHz raadiomoduleid, millega sai reguleerida liiklust kitsa teeremondi ajal. Selleks oli tehtud kaasaskantavad valgusfoorid, mille süsteem koosnes kahest komplektist (*master* ja *slave*), mis oli ühendatud raadiosignaali kaudu ning mõlemale valgusfoorile oli sisse ehitatud antenn. Keskseadme külge oli ühendatud valgusfoor ning keskseade andis teada valgusfoorile, kui kaua peab põlema punane tuli, roheline tuli ning missugune intervall on nende vahel. Need ajad ja intervallid olid seadistatud programmiga. Juhtplaadid, raadiomoodulid ja antennid oli sisseehitatud valgusfoori, mis oli väikese mõõduga karbiki umbes tikukarbi suurune, mis asetses peavalgusfoori juures. Kogu süsteem sai toidet akult [33, 40].

Esimeseks puuduseks selle süsteemi juures oli see, et kui oli vaja valgusfoori töörežiimi vahetada, siis selleks oli vaja programmeerimiskoodi sisu muuta. Teiseks valgusfoor ei saanud iseseisvalt aru, kas töötajad olid teeremondi piirkonnas või olid lahkunud [ibid].

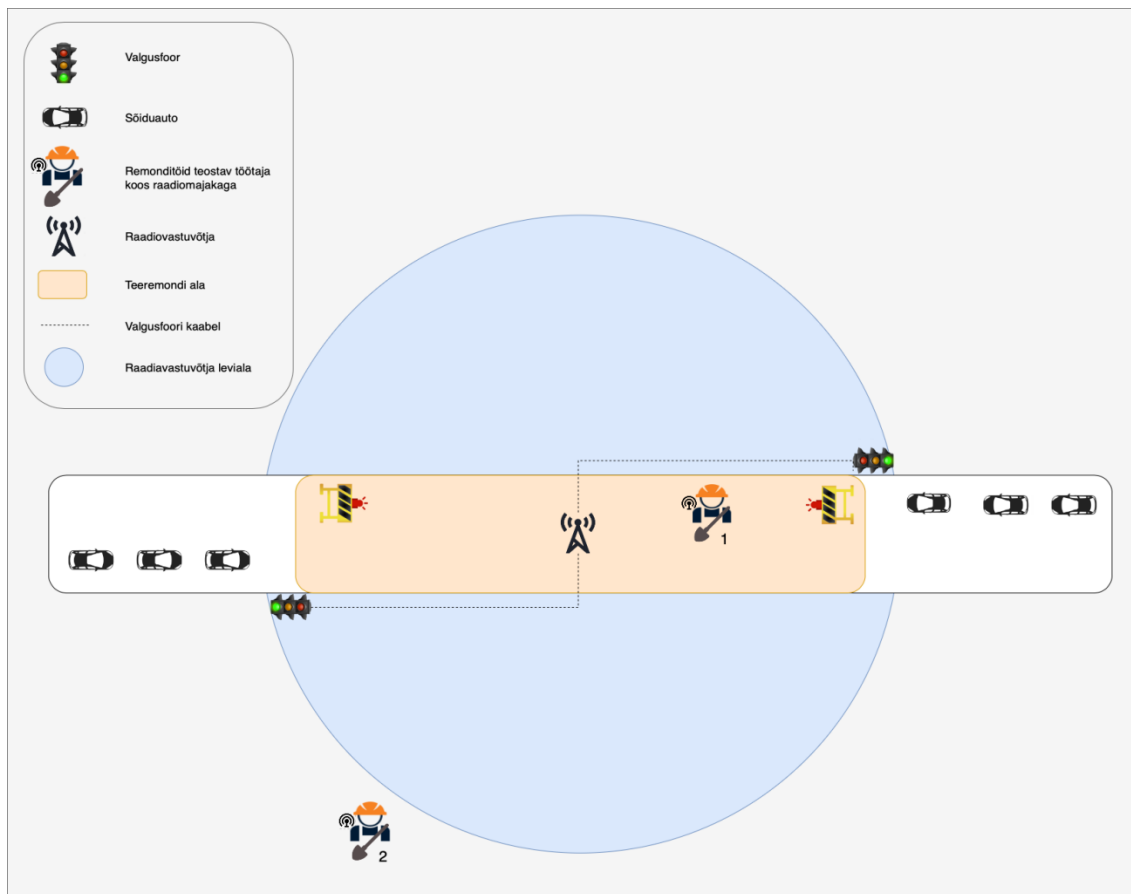
2.2.2 Turvalisus raudteel

Kasvanud rongiliiklusega kaasneb ka rohkem avariisid raudteel. Süsteem [34] vähendas sõiduautode ooteaega ning avariisid raudtee peal. Süsteem koosnes ISM 2.4 GHz Xbee moodulitest, infrapuna-, ultraheli- ja liikumisanduritest. Seadme topoloogia oli järgmine: raudtee ja autosõidutee ristmikul teatud kaugusele paigaldati raudtee külgedele üksteisele vastu kolmeliigiline andur, mis koosnes infrapuna-, ultraheli- ja liikumisanduritest. Antud lahenduse skeemiga saab tutvuda Lisa 2. Rongil oli samale kõrgusele asetatud ka kiirgur, et raudtee küljes olevad andurid saaks aru, et see, kes andurist mööda sõitis oli rong ning radio signaali abil edastatakse info keskseadmele, mis asus ristmikul. Kui keskseade sai signaali, et rong läheneb ristmikule, läksid tõkkepuud alla. Enne tõkkepuude alla laskumist kontrollis süsteem veel, ega keegi pole jäänud raudteel kahe tõkkepuu vahele. Seda kontrolliti ultraheliandurite abil, mis paiknesid ristmiku tee all. Kui ultraheliandur sai aru, et keegi oli kahe tõkkepuu vahel, siis sellest teavitati kohe rongi häiresignaaliga ning vilkuva punase signaaliga, et juht suudaks varakult reageerida. Kui kahe tõkkepuu vahel ei olnud kedagi, liikus rong edasi nii nagu ennegi [34].

Plussideks oli see, et mitu 2.4 GHz moodulit suutsid omavahel edastada üksteisele infot liikluseolu kohta. Teiseks selle süsteemiga oli tagatud turvalisus, et juhul kui auto jääb tõkkepuu vahel, siis süsteem teavitab rongijuhti. Võrreldes lõputöö süsteemiga saab tuua välja erinevuseks selle, et selle süsteemi puhul ei ole energiaprobleemi, nagu näiteks antud lõputöös.

2.3 Lõputöö lahenduse tööpõhimõte

Lõputöö autor tegi enda jaoks selgeks, kuidas automatiseeritud valgusfoori prototüübi süsteem peaks töötama ning see on esitatud Joonisel 1. Raadiovastuvõtja asub teeremondi ala keskel, ning võtab vastu signaale, mida jagavad raadiomajakad, mida kannavadteetöölised. Mõlemale valgusfoorile oli veetud kaabel raadiovastuvõtjalt. See oli vajalik, et kui töölised asuvad tee töö tsoonis (Joonisel remonditöid teostav töötaja koos raadiomajakaga nr 1), siis valgusfoorid töötavad ühel töörežiimil ning, kui tee töölisi ei ole töö tsoonis (Joonisel remonditöid teostav töötaja koos raadiomajakaga nr 2), siis valgusfoorid hakkavad töötama teises töörežiimis.



Joonis 1. Lahenduse tööpõhimõte.

3. Moodulite valik ja analüüs

Lõputöös otsitakse lahendust, mille abil oleks võimalik automatiseerida teeremondi ajal liiklust kahe-suunalisel teel pikkusega 100–200 meetrit. Selles peatükis toob autor välja milliseid tulemusi katse-eksitus meetod andis ning kirjeldab, mis probleemid kaasnesid nende katsetustega.

3.1 Raadiolainete levimine

Autori jaoks oli oluline selgeks teha, kuidas raadiolained levivad erinevates tingimustes, eriti õues, kuna lõputöö prototüüpi hakatakse kasutama pigem väliskeskkonnas. Näiteks, mis tingimustes levib signaal kõige paremini, mis takistused võivad segada signaali levimist ning kuidas oleks võimalik ületada neid takistusi. Tutvudes järgnevate artiklitega [32], [35], sai autor teada, et:

- otsenähtavuse korral levivad raadiolained kõige paremini;
- raadiolaineid summutavad kõige enam kõrged metallist või terasbetoonist ehitised või maastiku kõrgendused;
- raadiolaineid võivad mõjutada ilmastikuolud, näiteks vihm ja udu;
- seadmed, mis töötavad samal sagedusel võivad tekitada liigset müra ja summutada raadiolaineid.

Lisaks sai autor teada, et signaali levimine sõltub ka antenni tüübist ja pikkusest [35].

3.2 SA100 Dual/IP66 Remote EC moodulite katsetus

Algselt kasutas töö autor SA100 Dual/IP66 Remote EC mooduleid, kuna oli eelnevalt oma tavatöös neid modifitseerinud ja automatiseerinud nende abil sisse- ja väljasõidu väravaid. Tänu sellele teadis autor, et nendes moodulites oli sisse ehitatud 433 MHz raadiovastuvõtja ja eraldi 433 MHz raadiosaatja, mis olid juba eelnevalt programmeeritud.

Antud süsteemis oli hea, et raadiovastuvõtjal võimaldas juhtida kahte releed puldi abi, mis kontrollisid pinget 12 V. Rohkem infot SA100 Dual/IP66 Remote EC mooduli kohta on võimalik saada järgmiselt lingilt [21]. Tänu releele oli autoril võimalus ühendada ühe releega kaks Arduino Uno arendusplaati. Iga arendusplaat töötas erinevas režiimis ja oli ühendatud oma valgusfooriga. Vastuvõtja küljes oleva releega oli võimalik kontrollida, kumb Arduino Uno arendusplaat on ühendatud voluringi, ehk kui üks Arduino saab 12 V, siis teine arendusplaat ei saanud voolu.

Sellist süsteemi kasutades avastas autor, et raadiovastuvõtja oli eelnevalt programmeeritud töötama ainult kahe raadiosaatjaga ning selle vastuvõtjaga ei olnud võimalik rohkem raadiosaatjaid ühendada. Kuna tegelikkuses peaks lõputöö süsteemiga ühendust saama rohkem kui kaks raadiosaatjat, pidi autor otsima uue lahenduse töö süsteemi teostamiseks. Samas selle katse tulemusena otsustas autor edasi ehitada oma lahendusi kasutades Arduino Uno arendusplaati, kuna nii mikrokontrolleri riistvara kui ka tarkvara oli mugav kasutada. Veel üheks põhjuseks Arduino Uno kasuks oli see, et seal oli palju andmeväljundeid, kuhu oli võimalik ühendada palju teisi asju: näiteks lambid, ekraanid ja moodulid.

3.3 433 MHz moodulite katsetus

Peale eelnevas alapeatükis mainitud katsetust otsustas autor ise arendada raadiovastuvõtja ja raadiosaatja omavahelist suhtlust. Autor leidis oma hoovist ühe raadiojuhitava auto koos puldiga ja võttis nad lahti. Eesmärk oli teada saada, mis sagedusel raadiosaatja ja vastuvõtja töötasid, mis moodulid seal sees olid ja kas neid saab kasutada Arduino Unoga. Selgus, et puldis ehk raadiosaatjas oli moodul FS1000A, autos ehk raadiovastuvõtjas oli XY-MK-5V ning need sobisid antud lahendusse. Plussideks eelmainitud moodulite puhul oli ka see, et neid oli lihtne ühendada ja neil oli

eelprogrammeeritud alglaadur. Lisaks moodulid ei pea olema üksteise vaateväljas, vaid saavad suhelda omavahel ka läbi takistuste [57].

Seejärel otsustas autor katsetada sama raadiojuhitava autoga signaali ulatust. Katse toimus järgnevalt: autor kauges autost ning iga meetri järel vajutas raadiopuldil nuppu. Kui auto liikus, siis järelikut oli ühendus olemas ja autor astus meetri jagu edasi. Autor liikus nii kaugemale, kuni auto enam ei reageerinud nupuvajutusele, mis tähendas et raadisaatja ja -vastuvõtja vahel kadus ühendus ära. Katsesest selgus, et FS1000A moodulite omavaheline signaali ulatus oli kuni 20 meetrit.

Peale katsetust otsustas autor seadme ise kokku panna, programmeerida ning parendada signaali ulatust antenni abil. Töö tegija pani kokku algselt 433 MHz raadisaatja Arduino Uno arendusplaadiga, kasutades toite- ning andmeväljundeid. Sama tuli teha ka 433 MHz raadiovastuvõtjaga, mis ühendati teise Arduino Uno plaadiga. Järgmiseks tuli panna 433 MHz moodulitele antennid, et suurendada nende vastuvõtmise raadiust. Kui kõik sai ühendatud, tekkis probleem sellega, et 433 MHz sagedus oli avalik ning kõik seadmed, mis töötavad juhtmevaba juhtimisega võivad tekitada segavaid häireid. 433 MHz mooduli käivitamiseks oli vaja juurde eraldi Arduino Uno arendusplaat, mis ei teinud kokkupandud seadmeid kompaktsuks. Kuna kompaktsus oli autori jaoks tähtis, siis see oli ka üheks probleemiks, miks autor otsustas üle minna teisele moodulile.

Veel tekkis oht, et Arduino Uno ja 433 MHz moodul olid omavahel kokku ühendatud juhtmetega ning, et kogu süsteemi muuta kompaksemaks, tuli 433 MHz moodul asetada Arduino Uno mooduli peale, kus mooduli komponentide kontaktid olid kaitsmata. Kokkuühendatud süsteemi oleks pidanud käsitlema õrnalt, kuna võis tekkida olukord, et juhtmed tulevad lahti või Arduino Uno moodulite komponendid oleks võinud teetöoliste töö tegemise ajal kokku puutuda tema külge ühendatud 433 MHz mooduli komponentidega. Sellise olukorra tulemusel oleks võinud kogu süsteem minna lühisesse või isegi halvemal juhul põlema.

3.4 Bluetooth moodulite katsetus

Peale katsetust 433 MHz moodulitega võttis lõputöö autor veel ühe mooduli võrdluseks, mida ta ka plaanis prototüübi tegemisel kasutada. Selleks oli juhtmevabas hiires olev Bluetooth moodul HC-05. Autor katsetas Bluetooth moodulit järgmiselt: ühendas juhtmevaba arvutihiire sülearvutiga ning hoides käes hiirt liikus järjest sülearvutist kaugemale. Katse tulemus ei sobinud autori prototüübi eesmärgiga, kuna sülearvuti sees olev USB vastuvõtja ning juhtmevabas hiires olev saatja suhtlesid omavahel 30 meetrit.

3.5 NodeMCU V3 Lolin mooduli katsetus

Katsetuse kohaks valis autor Kristiine keskuse parkla, kus oli nii avalik tsoon, kui ka erinevatest materjalidest takistused. Järgmiseks soovis autor teada saada, kui suure raadiuse ulatuses suudavad moodulid signaali üksteisega jagada. See katse viidi läbi avalikus tsoonis ehk kohas, kus puudusid takistused. Selle katse tulemusel sai autor teada, et moodulid suudavad omavahel signaali jagada 75 meetri raadiuses ehk kui vaadata tulemust diameetrites, siis selleks on 150 meetrit. Veel selgus katsest, et NodeMCU V3 moodulid ei pea olema üksteise poole suunatud sisse ehitatud antennidega, et signaal oleks parem. Seepärast otsustas töö tegija võrrelda kolme raadiomoodulit omavahel, et teha õige otsus uue mooduli valimiseks. Moodulite võrdlust on võimalik näha Tabelis 1 [36] - [38].

3.6 Raadiomoodulite võrdlus

Pannes moodulite katsetuste tulemused ja tehnilised spetsifikatsiooni koondtabelisse (vt. Tabel 1), sai autor ülevaatlikult võrrelda katsetud moodulid omavahel. Võrreldes teiste tabelis välja toodud raadiomoodulitega on NodeMCU V3 moodulil ainukesena sisseehitatud 4 MB mälu, mis tähendas seda, et moodulile ei olnud vaja juurde lisada mikrokontrollerit. Lisaks sellele saab NodeMCU moodulit seadistada nii raadiosaatjaks kui ka vastuvõtjaks. Suureks eeliseks NodeMCU kasuks on samuti ka see, et tal ei ole ühenduste arvu piirangut võrreldes teistega. Veel on 2.4 GHz moodulite plussiks, et neil on juba sisse ehitatud antenn, kuid 433 MHz moodulitel seda ei ole. Tulemusena on näha, et 2.4 GHz moodulid sisseehitatud antenniga näitavad paremaid tulemusi signaali ulatavuse poole pealt.

Peale moodulite analüüsi otsustas autor kokku panna automatiseeritud valgusfoori prototüübi NodeMCU V3 Lolin WiFi mooduli põhjal. Töö tegija valis just 2.4 GHz sagedusel töötava mooduli, kuna sellel sagedusel kasutatakse teeremondi piirkonnas vähem seadmeid võrreldes 433 MHz sagedusega. Kasutades NodeMCU V3 Lolin 2.4 GHz WiFi mooduleid on võimalik vältida teiste seadmete poolt põhjustatud häireid.

Tabel 1. Raadiomoodulite võrdlus. Selles tabelis on võrreldud kolme erinevat tüüpi raadiomooduleid, milleks on: 433 MHz, HC-05, ESP8266(WiFi). Tabel annab ülevaate raadiomoodulite plussidest ja miinustest.

	HC-05	FS1000A ja XY-MK-5V	NodeMCU V3 Lolin
Sagedus	2.450GHz	433,92 MHz	2.4 GHz
Edastuskiirus	2,1 Mb/s	10 kb/s	4.5 Mb/s
Moodulite kogus	Raadiosaatja/-vastuvõtja ühes moodulis	Raadiosaatja/-vastuvõtja eraldi moodulid	Raadiosaatja/-vastuvõtja ühes moodulis
Pinge	3,3V	Saatja 3V - 12V Vastuvõtja 5V	3,3V
Mitu raadiosaatjat saavad suhelda ühe raadiovastuvõtjaga	7	2	Ei ole piiratud
Distants (välistingimustes otsenähtavuse korral)	Sissehitatud antenniga kuni 30m	Antenniga kuni 20m	Sissehitatud antenniga kuni 75m
Kas vajab mikrokontrollerit ?	Jah	Jah	Ei
Mälu	0 MB	0 MB	4 MB

4. Lahenduse kirjeldus

Selle töö jaoks oli kasutusele võetud RF (*Radio Frequency*, raadiosagedus) traadita side tehnoloogia, millega on võimalik tutvuda järgmiselt lingilt [49].

WiFi moodulite suhtlemismeetodiga tutvus lõputöö autor antud linkidelt [44], [6].

4.1 NodeMCU V3 mooduli plussid

Autor katsetas erinevaid raadiomoduleid ja valis nende hulgast NodeMCU V3 Lolin, sest need raadiomoodulid on usaldusväärsed ja nii mikrokontrolleri riistvara, kui ka tarkvara oli mugav kasutada. Nendel moodulitel oli omad programmeeritavad väljundid ehk ei olnud vaja lisada programmeeritava andmeväljundiga plaate, milleks võib olla näiteks Arduino Uno. Moodul ise oli töö tegija jaoks sobivate mõõtudega 60 mm x 30 mm [48]. See oli vajalik selleks, et lõpptulemuseks tuleks kompaktne seade. NodeMCU V3 moodulil saab seadistada oma WiFi võrgu, mis töötab 2.4 GHz sagedusel. WiFi võrku saab kaitsta WPA2 turvaprotokolliga, mis on siiani kõige turvalisem turvaprotokoll [59].

Lisaks NodeMCU V3 Lolin moodulitel on sisseehitatud programmeeritav LED indikaator, millest abil saab edastada vähemalt kolm visuaalset signaali lõppkasutajale raadiomajaka olekust:

- ei põle (aku on tühi või seade on rikkis);
- vilgub (seade on töökorras, kuid ei ole ühendatud WiFi võrku);
- ja põleb (seade on töökorras ja ühendatud WiFi võrku).

Seeläbi saab töötaja kiiresti visuaalselt hinnata, kas tema raadiomajakas on töökorras ja ühendatud vastuvõtjaga. See oli veel üheks plussiks NodeMCU V3 valikul. See võimaldas autoril mooduli ühendamise monitoorimist teha ilma arvutita.

Autor valis NodeMCU V3 Lolin moodulid, kuna ühte moodulit oli võimalik kasutada nii raadiosaatja, kui ka raadiovastuvõtjana. Selleks tuleb kasutada programmis vastavat funktsiooni, mille abil saab moodulit seadistada raadiovastuvõtjaks või raadiosaatjaks.

Lõputöö üheks tähtsaks kriteeriumiks oli teha kompaktne seade, et seda oleks mugav kaasas kanda. Kui moodul oli raadiosaatja funktsioonis, siis ei olnud vajalik enam talle rohkem juhtmeid juurde ühendada, mis tegi lõpptulemuse jällegi kompaktsemaks. Veel plussiks NodeMCU V3 Lolin moodulil oli see, et sisseehitatud antenni raadius oli isegi parem, kui mõnel moodulil, millel oli juurde joodetud antenn [31]. Moodulite ühenduse saamiseks tuli raadiosaatjal olla raadiovastuvõtja piirkonnas ning sisse lülitatud.

Seade saab olla sisse lülitatud kahel võimalusel, milleks on:

- Arvuti abil läbi USB pordi (näiteks koodide laadimiseks, mooduli seadistamiseks, testimiseks või töölise nime vahetamiseks)
- Akuga otse ühenduses 3-12V.

4.2 Moodulite kirjeldus

Lõputöös oli kasutatud kahte eritüüpi mooduleid. Üheks oli NodeMCU V3 WiFi moodul, mida kasutati raadiosaatja ja -vastuvõtja jaoks. Teiseks oli RTC (Real Time Clock / reaalaaja kella) moodul, mida kasutati töös katse läbiviimiseks, millega saab lähemalt tutvuda 4.3.2 RTC mooduli alapeatükis.

4.2.1 NodeMCU V3 moodul

NodeMCU V3 tähendab *Node MicroController Unit*, mis on avatud lähtekoodiga tarkvara ja riistvara arendamise keskkond. Sellesse moodulisse on juba sisse ehitatud ESP8266 moodul, milles on protsessor, RAM (*random access memory*, muutmälu), võrguühendus (WiFi), kaasaegne operatsioonisüsteem ja SDK (software development kit). NodeMCU V3 Lolin moodulil on kokku maksimaalselt 30 sisendit ja väljundit [14]. Põhjalikumate tehniliste andmetega on võimalik tutvuda Tabelis 2.

Tabel 2. NodeMCU V3 Lolin tehnilised andmed

Mikrokontrolleri tüüp	ESP-8266 32-bit
Mooduli suurus	58mm x 32mm
Taktisagedus	80 MHz
USB (Universal Serial Bus)	CH340G
USB tüüp	Micro USB
Pinge	3.3 V
SRAM (<i>Static random access memory</i> , staatiline muutmälu)	4 MB / 64 KB
Sisend/väljund viike	11
ADC (<i>analog-to-digital converter</i> , analoog-digitaalmuundur) vahemik	0-3.3V
UART (<i>Universal Asynchronous Receiver-Transmitter</i> , Universaalne asünkroonne vastuvõtja-saatja)	1
SPI (<i>Serial Peripheral Interface</i> , Järjestikune perifeeria liides)	1
I2C (<i>Inter-Integrated Circuit</i> , Integreeritud vooluahel)	1
Sisseehitatud WiFi	802.11 b/g/n
Temperatuuri vahemik	-40°C - 125°C

NodeMCU V3 Lolin mooduli tehnilised andmed leidis töö tegija järgmiselt lingilt [ibid].

4.2.2 RTC moodul

RTC moodul ehk reaalaja kell on seade, mis tegeleb hetkel oleva aja arvestamisega. Selle mooduli plussiks oli see, et juhul kui välistoide kadus ära, siis moodulil ei olnud vaja uuesti hetkel olevat kellaaega seadistada, kuna moodulil oli varu CR2032 nööp-patarei. Patarei abil hoidis moodul hetkel olevat aega meeles kui põhitoiteallikas oli eemaldatud [10]. RTC on oluline komponent paljudes ajakriitilistes rakendustes ja seadmetes, nagu serverid, GPS, andmelogerid. Andmete saatmiseks moodul kasutab I2C protokollit [11]. RTC tehniliste andmetega on võimalik tutvuda Tabelis 3[ibid].

Tabel 3. RTC mooduli tehnilised andmed

Kiip	24C32N
Pinge	3.3 V
Mälu	32Kb
Temperatuuri vahemik	-40°C kuni +85°C

4.3 Moodulite programmeerimine

Enne NodeMCU V3 Lolin programmeerimist soovis lõputöö autor tutvuda ja viia ennast kurssi moodulite programmeerimise töökäiguga. Autor sai teada, et kõige parem oleks neid moodulid programmeerida Arduino IDE arenduskeskkonnas. Lõputöö tegija sai teada, kuidas mooduleid kasutada ja mida oli vaja, et mooduleid programmeerida. Täpsema loetelu komponentidest, mida oli vaja selleks, et luua prototüüp leiab Tabelis 4.

Tabel 4. Prototüübi komponendid. Vajalikud esemed ja seadmed lõputöö läbiviimiseks olid ja nende ligikaudne maksumus.

Komponendid	Kogus	Hind (€)
NodeMCU V3 Lolin 2.4 GHz WiFi moodul	1 tk	2,5
RTC DS3231 moodul	1 tk	3,0
ENERGIZER Patareid CR2032	1 tk	1,3
Akupank Esperanza 2000 mAh	1 tk	4,5
LED (punane)	1 tk	0,6
LED (kollane)	1 tk	0,6
LED (roheline)	1 tk	0,6
Takistid	3 tk	0,6
Maketeerimislauda juhtmed (ema - ema)	1 komplekt	4,5
Maketeerimislauda juhtmed (isa - isa)	1 komplekt	4,5
	Kokku	22,7

NodeMCU Lolin 2.4 GHz WiFi moodul tuli tellida e-poest (Aliexpress), kuna selliseid ei olnud tol ajal Eestis müügil. Ülejäänud komponendid ostis autor Oomipoest. Kui kõik ettevalmistused programmeerimiseks olid tehtud tuli lõputöö tegijal hakata programmiliselt seadistama NodeMCU V3 Lolin mooduleid. Lisaks sai lõputöö tegija teada sellelt veebilehelt, kuidas oleks turvaline programmeerida Arduino IDE (*Arduino Integrated Development Environment*, Arduino integreeritud arenduskeskkond) arendusplaati. Enne koodi loomist oli vaja teatud teegid alla laadida ning lisada õigesti need koodi. Selleks oli töö tegijal abiks järgmine link, kus oli võimalik teada saada, kuidas lisada teeke [53].

Antud lõputöö koodi loomiseks kasutati järgmisi teeke:

- ESP8266WiFi.h - teek, mida kasutatakse kui tegemist on WiFi moodulitega, selle teegi abil sai saata ja võta vastu signaali [54].
- Wire.h - kasutati koodis, et ei tuleks ette tundmatuid sümboleid [55].
- RTCLib.h - kasutati koodis RTC mooduli korrektseks töötamiseks [56].

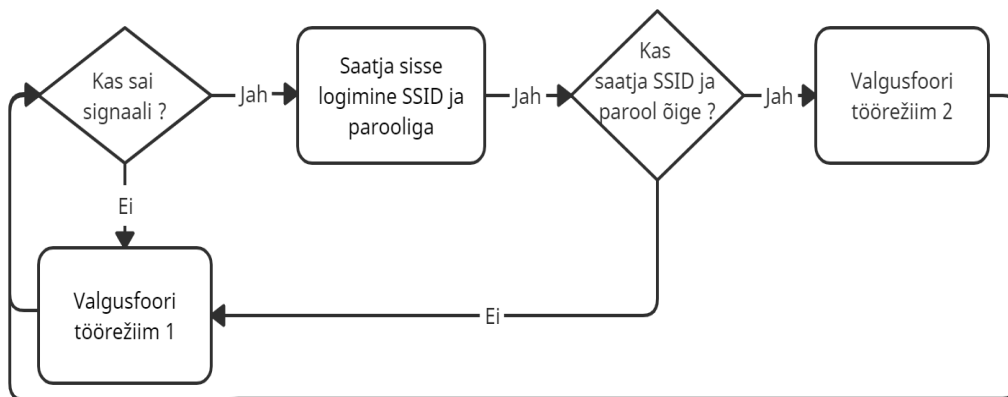
5. Eksperimendi kirjeldus

Selles peatükis autor kirjeldab moodulite valikut, toob välja tabeli kujul moodulite võrdluse ning selgitab, miks just selline valik tehti antud prototüübi tegemiseks. Järgmiseks räägib töö tegija NodeMCU V3 Lolin töökäigust, prototüübi katsetamistest ja tähtsamatest tulemustest. Prototüübi programmeerimisel kaasnes mõningaid vigu, millest töö autor räägib lühidalt.

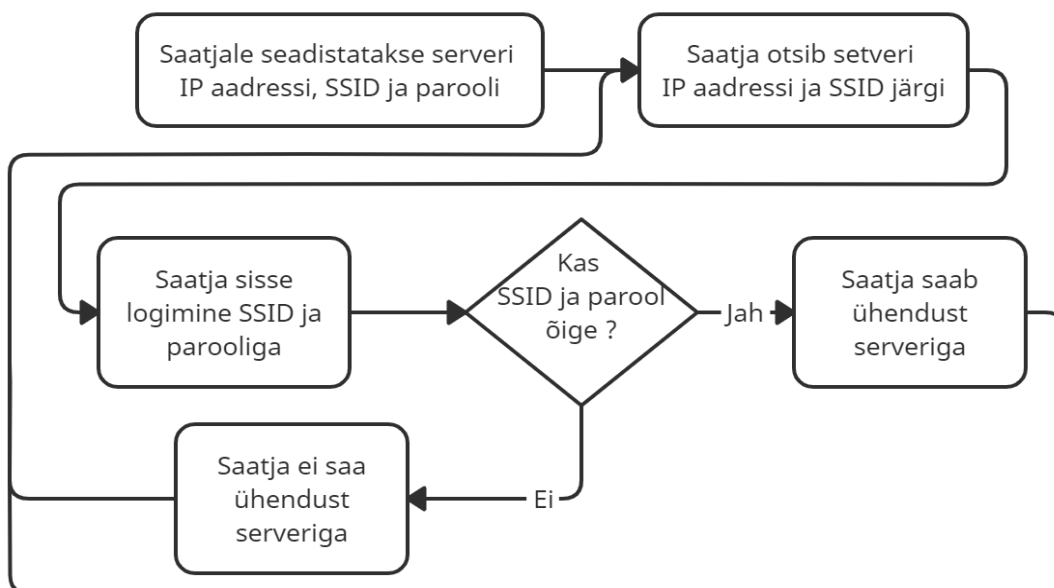
5.1 NodeMCU V3 Lolin töökäik

Kõigepealt tuli autoril alla laadida programm, milleks oli Arduino IDE. See oli vajalik, et programmeerida mooduleid. Järgmiseks tuli programmi seadistada vastavalt programmeeritava mooduli järgi. Kuna autor kasutas NodeMCU V3 Lolin moodulit, siis tuli seadetest valida õige programmeeritav moodul ja valikutest teha õiged seaded, mille õpetus on vastava lingi alusel kättesaadav [20]. Edasi tuli programmis pakutavatest baastekidest lisada õiged teeked, sobivate teekide puudumisel tuli need ise sinna alla laadida. Autoril programmiga töötamisel puudusid vajalikud teegid Arduino IDE valikutes, seepärast tuli töö tegijal minna arenduskeskkonna otsingusse, kus oli suurem teekide kogum. Autor puutus kokku olukorraga, kus ka arenduskeskkonna otsingus puudusid vajalikud teegid ning autor pidi vajaliku teeked internetist üles otsima ZIP (faili formaat)-failina, aili arvutisse alla laadima ja programmis olevate teekide kogumikesse lisama.

Autor oli praktilise poole jaganud seitsmeks etapiks, mida tehti järk-järgult. Esmalt oli vaja luua struktuurskeemid, et näha, kuidas kogu süsteemi loogika käib vastavalt Joonisele 2. ja Joonisele 3. Struktuurskeemid abistasid töö tegijat programmeerimisel.



Joonis 2. Raadiovastuvõtja programmeerimis algoritm. Autori poolt tehtud.



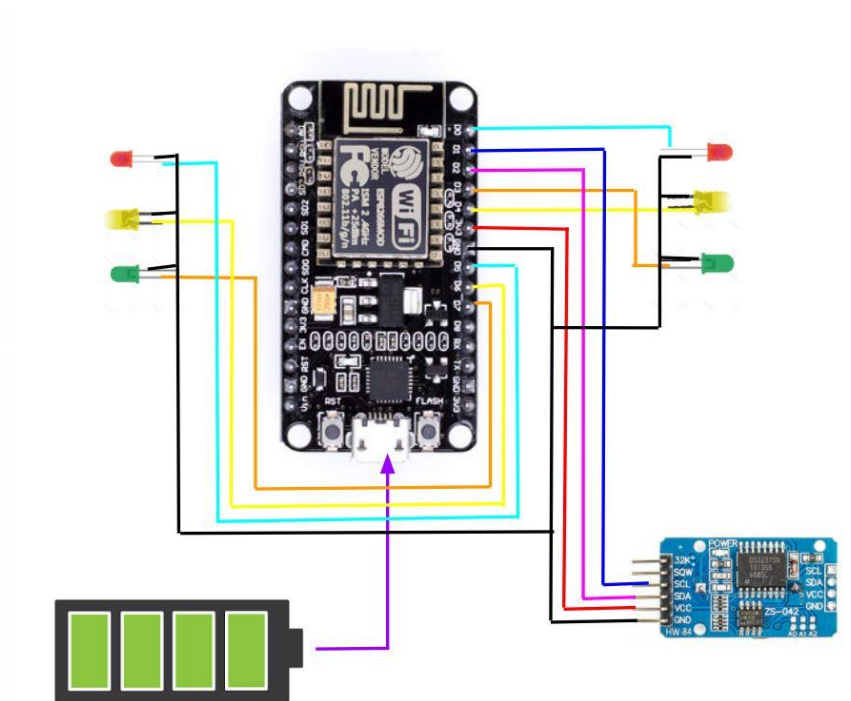
Joonis 3. Raadiosaatja programmeerimise algoritm. Autori poolt tehtud.

Teiseks ehk ka kõige tähtsamaks osaks oli kahe mooduli suhtlus omavahel juhtmevaba WiFi ühendusega, millest oli ka varasemalt juttu. Autor otsis internetist võimalusi, kuidas saaks panna suhtlema kaks moodulit omavahel. See sai realiseeritud NodeMCU vastuvõtja baaskoodiga, mida autor muutis oma vajaduste järgi. Kasutatud baaskoodiga on võimalik tutvuda järgmisel lingil [19]. Kolmandaks, kuna tegemist oli ise ehitatud lokaalvõrguga, siis autoril oli kindlasti vaja kaitsta võrk vähemalt parooliga. Autor saavutas selle WiFi nime ja raadiovastuvõtja poolt kontrollitava parooli kombinatsiooniga, et keegi võõras ei saaks ühendust objektile oleva raadiovastuvõtjaga isegi siis, kui kirjutatakse täpselt sama programmeerimiskood raadiosaatja jaoks.

Neljandaks pidi autor ühendama RTC mooduli olemasoleva süsteemiga ja programmeerima selle. Autor kasutas RTC mooduli baaskoodi põhjana, millega on võimalik tutvuda järgmisel lingil [17]. Joonis 4. ja Joonis 5. on kujutletud, kuidas töö tegija ühendas kõik komponendid omavahel.

Joonise 4. oleva valgusfoori raadiovastuvõtja kokku panemiseks oli vaja autoril järgmisi komponente:

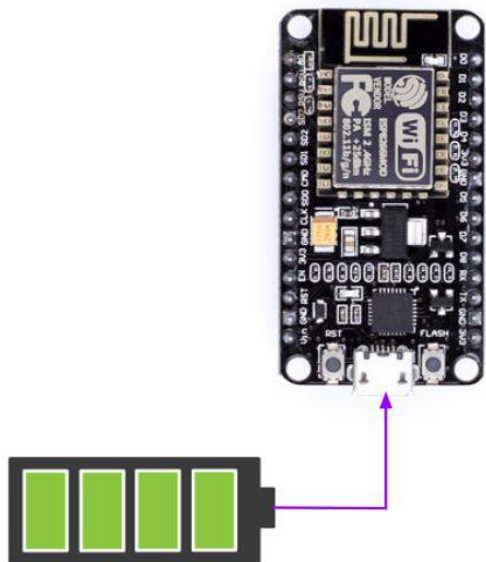
- NodeMCU V3 Lolin moodulit,
- RTC DS3231 moodulit,
- LED lambid,
- akupanka.



Joonis 4. Raadiovastuvõtja ühendamise skeem. Skeemi komponentide pildid on pärit allikatest [25] - [30]. LED-i ühendusega autor tutvus antud lingil [12]. RTC mooduli ühendusega lõputöö tegija sai informatsiooni järgnevalt veebilehelt [17]. Kasutades allikat [52], sai lõputöö tegija teada, kuidas ohutult vooluallikat kasutada.

Joonise 5. oleva raadiosaatja kokku panemiseks tuli kasutada järgmisi asju:

- NodeMCU V3 Lolin moodulit,
- akupanka.



Joonis 5. Raadiosaatja ühendamise skeem. Skeemi komponentide pildid on pärit allikatest [25], [28].

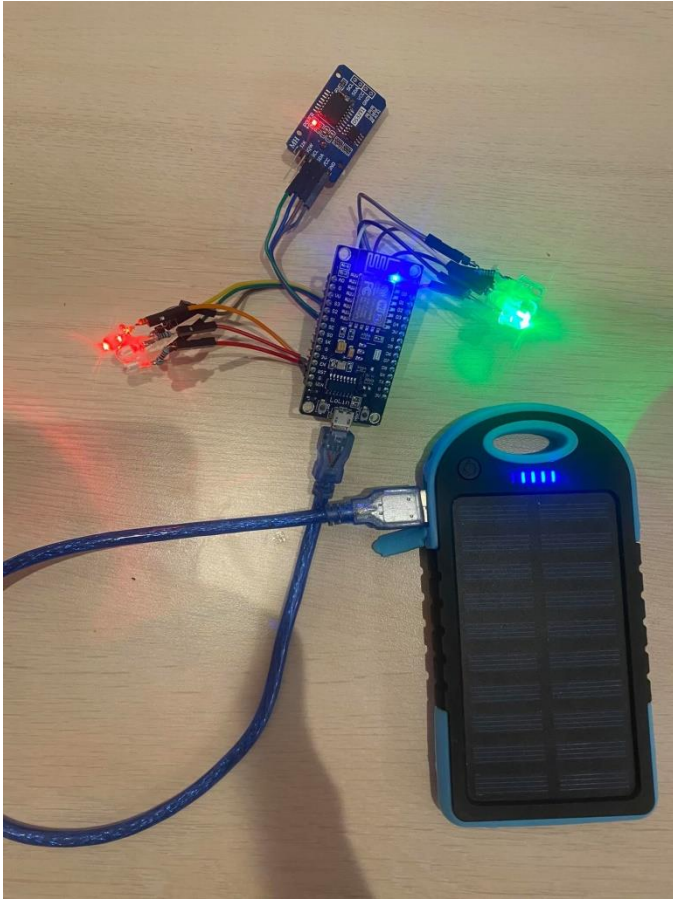
Viiendaks etapiks tuli autoril ise välja mõelda valgusfoori algoritm, mille abil töötaksid valgusfoorid mõlemas sõidusuunas. Kuuendaks, kuna raadiovastuvõtjaga olid ühendatud valgusfoor ja RTC moodul, tuli kõige lõpuks Arduino IDE programmis kõik neli koodiosa kokku panna ning teha sellest üks ühine programmeerimise kood, mis lõpuks moodustaks raadiovastuvõtja koodi.

Viimaseks, kuna seadmed said toidet otse ühest arvutist, millega toimus ka seadmete programmeerimine, siis autoril oli vaja lahendada toiteallika probleem, sest prototüübi komponendid ei ole lähestikku, seega ei saa olla nende mõlema toiteallikana olla ühine arvuti. Töö autor lahendas selle probleemi järgmiselt, et ühendas prototüübi kaks komponenti ehk nii raadiosaatja kui ka raadiovastuvõtja akupangaga. Raadiovastuvõtja jaoks kasutas autor akupanka, mille mahuks oli 10000 mAh pingega 5 V. Raadiosaatja oli vaja ühendada aku külge, mis oli väiksema mahuga, sest kogu seadme mõõdud ei võinud olla suured ja, et seade saaks kiiresti ennast täis laadida ja kestaks vähemalt 8 tundi. Autor valis raadiosaatja jaoks akupanga, mille mahuks oli 1350 mAh pingega 3,7 V.

Kõigi esemete ühendused ning nende programmeerimiskoodid on võimalik leida järgnevalt lingilt [17], [8]. Jooniselt 6 ja Jooniselt 7 on võimalik näha lõputöö käigus valminud prototüüpi.



Joonis 6. Prototüübi raadiosaatja koos akupangaga.



Joonis 7. Prototüübi valgusfoor koos akupangaga ning RTC mooduliga.

5.2 Valmis prototüübi katsetamine

Autor teostas valmis süsteemiga ka mitmeid erinevaid katsetusi. Ühe katse läbiviimiseks oli autoril vaja kasutada RTC moodulit. Kasutades RTC moodulit autor teostas katseid, millega sai teada, kui kaua peavad vastu raadiosaatja ja -vastuvõtja akud. Antud moodul oli raadiovastuvõtja küljes, mis näitas kuupäeva ja kellaega, millal NodeMCU V3 moodulid (raadiosaatjad) olid serveriga ühendust saanud. Programmeerimise ajal oli autor NodeMCU V3 moodulitele andnud unikaalsed nimed, millega oli võimalik teada saada, kes on saanud ühendust serveriga. Tänu RTC moodulile oli võimalik ühendust saanud NodeMCU V3 mooduleid logida. Logist oli võimalik täpselt jälgida kellaajaliselt ja kuupäevaliselt, millal mingi moodul oli saanud ühendust serveriga ning ka seda, millal signaal katkes. Selle katsega sai autor teada, mis kuupäeval ja mis kellast hakkas moodul signaali saama (täislaetud aku) ning, mis kell signaal kadus. Kui signaal kadus tähendas see seda, et aku oli saanud tühjaks ehk autor sai teada viimase kuupäeva ja kellaaja, millal oli ühendus veel olemas. Teades alguskuupäeva ja aega tänu RTC moodulile ja lõppkuupäeva ja aega, sai autor välja arvutada, kui kaua peab raadiosaatja aku vastu. Kuigi autor oli teadlik, et programmis Arduino IDE oli olemas vastav funktsioon, millega saab jälgida kellaajaliselt mooduli ühendust serveriga, kuid autorile selline variant ei sobinud, kuna katsetamine toimus rohkem kui 24 tundi.

Järgmiseks soovis autor teada saada, kui palju võtab aega moodulite käivitamine, õige võrgu leidmine, parooli saatmine raadiovastuvõtjale, ning selle kontrollimine ja vastuse saamine. Esimese katse tulemuseks sai autor teada, et kõik eelnevad protsessid võtavad kokku 5 sekundit aega.

Seejärel katsetas autor, kuidas signaal levib, kui seisev auto või parklas õhukesest plastmassist oleva ostukäru hoiustamise punkt oli takistuseks. Nende signaali edastamise ulatus oli 75 meetrit. Kolmandaks takistuseks oli metallkattega võileib-tüüpi sein ehk selleks oli maja nurk. Selle tulemus oli kahjuks täielik signaali kaotus ehk läbi sellise paksu takistuse moodulid ei saanud signaali üksteisega jagada. Viimaseks katsetas autor, kuidas signaal levib moodulite vahel, kui raadiosaatja moodul asetseb autos. Selle katse puhul tehti kaks erinevat katset. Esimeseks hoiti raadiosaatja moodul üleval ehk läbi aknaklaasi oli raadiovastuvõtjale nähtav raadiosaatjat ning selle tulemuseks oli, et signaali suutsid moodulid üksteisele jagada 50 meetri raadiuses.

Teiseks asetati raadiosaatja mitte nähtavale kohale raadiovastuvõtjale, milleks oli seestpoolt auto uks tasku. Kahjuks kohe, kui raadiosaatja pandi auto ukse metallist ala juurde kadus signaal raadiosaatja ja -vastuvõtja vahel ära.

Autor tellis kolm NodeMCU V3 Lolin moodulit, millest üks oli programmeeritud raadiovastuvõtjaks ning kaks raadiosaatjateks. Nii sai testida andmete liikumist ühele raadiovastuvõtjale. Pandeemia tõttu tulid moodulid väga kaua, ligikaudu üheksa kuud. Autor teostas kolme mooduliga järgmisi katseid:

- toiteallika, väljundite ja kahe raadiosaatja ja ühe raadiovastuvõtja omavahelise suhtluse kontroll;
- nende omavahelist suhtluse kauguse testimine.

Kõige tähtsamaks katseks oli vaja teada saada, kuidas raadiovastuvõtja käitub kui sellega on ühendatud kaks raadiosaatjat. Peale testimist sobisid autorile NodeMCU V3 Lolin moodulid, et täita lõputöö ülesanded.

Tegelikuses oli autoril plaanis lõputöös kasutada minimaalselt kahte raadiovastuvõtjat ja mitu raadiosaatjat, mille abil raadiovastuvõtja hakkaks ise valima, mis režiimil valgusfoor töötab. Kahjuks, kuna tekkisid eelnevalt juba mainitud tarneraskused, siis autori poolt oli otsustatud teha valgusfoori prototüüp ainult ühe raadiovastuvõtjaga ning vedada 75 meetrit kaablit valgusfoorideni. Täpsemat skeemi ja ülesehitust näeb Jooniselt 1. Lõputöö prototüüp oli mõeldud lühikese tööloigu jaoks ning tegelikult sobis see variant isegi paremini lõputöö praktilise osa läbiviimiseks. Juhul, kui teelõigul oli ka ristmik, siis raadiovastuvõtja tuleks panna keset ristmikku, ning lisada raadiovastuvõtjale kolm või enam valgusfoori.

5.3 Programmeerimisega kaasnenud probleemid

Programmeerimise käigus tekkis töö tegijal mõningaid probleeme, millega saab tutvuda järgnevates alapeatükides.

5.3.1 Mac operatsioonisüsteem

Alguses autor tegi oma lõputöö programmeerimis osa Apple Macbook Pro 2018 sülearvutis, kuid sellega tekkis mõningaid probleeme. Autor programmeeris kõik moodulid nii, nagu oli eelnevalt välja toodud lõigus 5.1 NodeMCU V3 Lolin töökaik. Jõudes etapini, kus tuli kood üles laadida kokku pandud seadmetele, siis autoril ei õnnestunud seda teha. Programm andis laadimise ajal veateate, mis oli autori jaoks tundmatu. Joonisel 8 on võimalik tutvuda veateatega. Töötegija hakkas otsima lahendusi sellele veateatele internetist ning sai mitu lahendust sellele probleemile. Ühe lahendusega, mida autor proovis on võimalik näha järgnevalt lingilt [41]. Lõputöö autor järgis kõiki leitud juhendeid, mis internetis leidis selle veateate kohta, kuid ükski lahendustest ei toiminud. Kuna internetist saadud lahendused ei toimunud, siis otsustas töötegija Arduino IDE programmi arvutist täielikult kustutada ning uuesti programm arvutisse alla laadida, installeerida, ning ka teeke alla laadida ja installeerida. Kui kõik oli alla laetud, installeeritud ja seadistatud proovis autor uuesti koodi üles laadida, kuid midagi ei muutunud ning sama veateade tuli ette. Peale ebaõnnestunud katseid töötegija vahetas *Mac OS*-i (*Macintosh Operating System*, Macintosh operatsioonisüsteem) *Windows 10 Home*-i operatsioonisüsteemi vastu. *Windows 10* operatsioonisüsteemiga autor teostas samad etapid, mis *Mac OS*-iga. Lõpuks autoril õnnestus üles laadida sama kood, mida ta oli juba eelnevalt mitmeid kordi proovinud üles laadida *Mac OS*-iga. Töötegija tegi järelduse, et Arduino IDE programm *Mac OS*-il ei tööta nii hästi, kui *Windows 10* operatsioonisüsteemil. Kuna autoril olid ajapiirangud, siis kahjuks polnud aega probleemi otsida ning autor laadis koodi üles *Windows 10* operatsioonisüsteemi abil ning jätkas oma programmeerimis tööd sellega.

```
An error occurred while uploading the sketch Copy error messages
Executable segment sizes:
IROM : 246924 - code in flash (default or ICACHE_FLASH)
IRAM : 27260 / 32768 - code in IRAM (ICACHE_RAM_ATTR, ISRs)
DATA : 1256 ) - initialized variables (global, static) in RAM
RODATA : 876 ) / 81920 - constants (global, static) in RAM
BSS : 25216 ) - zeroed variables (global, static) in RAM
Sketch uses 276316 bytes (26%) of program storage space. Maximum is 1048576.
Global variables use 27348 bytes (33%) of dynamic memory, leaving 54572 bytes
pyserial or esptool directories not found next to this upload.py tool.
An error occurred while uploading the sketch

33 NodeMCU 1.0 (ESP-12E Module) on /dev/cu.usbserial-14120
```

Joonis 8. Tarkvara üleslaadimise tekkinud vaeteade Mac OS süsteemis.

5.3.2 Ooteaja lihtsustatud seadistamine

Lõputöö käigus autor puutus kokku probleemiga, kus üritas lihtsustada programmeerimiskoodi ooteaja seadistamisel, kasutades internetist saadud baaskoodi, millega on võimalik tutvuda järgmisel lingil [5]. Modifitseerimiskoodi tööpõhimõte seisneb selles, et süsteem küsib ooteaja kohta üksiklaadil programmeerija käest küsimusi ning süsteem ootab mõne minuti vastuse saamiseks. Lõputöö autor puutus kokku vastuse sisestamisel arusaamatu probleemiga Joonis 9, kus süsteem ei küsinud üksiklaadil küsimusi, vaid küsis kaks järjestikust küsimust korraga Joonis 10. Autor ei saanud aru sisestamise ajal, kumba vastust süsteem ootab. Süsteemi küsimuste esitamise järjekord oli programmeeritud nii, et süsteem esitab valgusfoori seadistamise küsimused üksiklaadil ning peale vastuste saamist saadab tööliste sisestatud vastused jadaporti, et tööline saaks kontrollida, kas sisestatud vastused on õiged. Lihtsa teksti printimisel kaasnes ka mõningaid probleeme, milleks oli pooliku sõna printimine ning topelt küsimuste korral vastuse väljastamine ainult ühele küsimusele jadapordis Joonis 11. Kuna probleeme tekkis kohe mitu, siis autor otsustas jätta kõik nii nagu enne oli. Antud lõputöö raadiosaatja, kui ka raadiovastuvõtja pseudokoodide info on võimalik kätte saada antud GitHubi lingilt [9]. Lõputöös prototüübi koodidega saate tutvuda Lisa 2. ja Lisa 3.

COM4

2|

```
Sisesta kaua peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta lühikese peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta sõitmise aeg minutites mõlema valgusfoori jaoks, kui objektil on kedagi
```

Joonis 9. Valgusfoori töörežiimi sisestamine Arduino IDE programmi. Programm küsis töö tegijalt valgusfoori töörežiimi, kui kaua valgusfoor sellel režiimil töötama peab. Autor sisestas soovitud väärtuse.

COM4

|

```
Sisesta kaua peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta lühikese peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta sõitmise aeg minutites mõlema valgusfoori jaoks, kui objektil on kedagi
Sisesta sõitmise aeg minutites mõlema valgusfoori jaoks, kui objektil ei ole kedagi
Sisesta peatuse aeg minutites mõlema valgusfoori jaoks kui objektil ei ole kedagi
```

Joonis 10. Arduino IDE programmi valgusfoori töörežiimi sisestamise topelt küsimine. Järgmiseks, kui Joonisel 7. oli näha, et autor vastas kolmandale küsimusele ära ning peale enter-i vajutamist küsis programm kohe neljandat ja viiendat küsimust. Jäädes ootama autoril viienda vastust mitte neljanda.

COM4

3|

```
Sisesta kaua peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta lühikese peatuse aeg minutites mõlema valgusfoori jaoks
Sisesta sõitmise aeg minutites mõlema valgusfoori jaoks, kui objektil on kedagi
Sisesta sõitmise aeg minutites mõlema valgusfoori jaoks, kui objektil ei ole kedagi
Sisesta peatuse aeg minutites mõlema valgusfoori jaoks kui objektil ei ole kedagi
-----
kk peatus, kui objektil käib töö:
Lühike peatus, kui objektil käib töö:
itmis aeg, kui objektil käib töö:
atus aeg kui objektil ei ole kedagi:
Soitmis aeg kui objektil ei ole kedagi:
```

Joonis 11. Programmis Arduino IDE vastuse printimisel kaasnud probleem. Joonisel on näha, kuidas mõningatel lausetel, kus tööline saab kontrollida enda sisestatud vastused oli poolikud sõnad, näiteks “kk” asemel tegelikult peab olema sõna “Pikk”.

6. Prototüübi parendamise võimalused

Kahjuks jäi lõputöö tegijal mõningad funktsioonid tegemata ja prototüübi lihtsustamise etapid vahele, kuna koroonaviiruse tarneraskuste pärast ei jõudnud moodulid kohale, lisaks sellele veel mõnede lisaseadmete lisamisel oleks lõputöö liiga mahukaks läinud. Lõputöö autor soovis jagada oma mõtteid, kuidas prototüüpi parendada ning sellest tulebki juttu järgmises peatükis.

6.1 Lahendused foori autonoomse tööaja pikendamiseks

Autori kokku pandud seadet saab tulevikus edasi arendada sellega, et tuleks kasutusele võtta parem aku lahendus raadiovastuvõtjale, kuna hetkel olevat akut tuleb käia laadimas. Seda probleemi lahendaks autor kasutades päikesepaneeli, kuna see suurendab aku töö kestvust. Päikesepaneel oleks hea alternatiiv, kuna enamasti tehakse teeremonte valgel, sageli päikeselisel ajal. Päikesepaneeliga ning tema tööpõhimõttega tutvus autor järgmises töös [45, lk 14].

Ainult päikesepaneeli lisamisest ei piisa, et laadida raadiovastuvõtja akut, selleks tuleks veel juurde lisada päikesepaneeli kontrolleri, näiteks LCD 20A 12V/24V USB. Kontrolleri abil saaks päikesepaneeliga kogutud energiat edasi anda raadiovastuvõtja akudele. Prototüübi paremaks muutmisel päikesepaneeliga tuleks silmas pidada seda, et teeremondi ajal on vaja valgusfooride ehk raadiovastuvõtjate asukohta muuta ning seetõttu kasutada tuleks mitte väga suuri päikesepaneele. Suured paneelid teeksid selle ebamugavaks ning võib juhtuda olukord, et suure tuulega võib päikesepaneelidega valgusfoor ümber kukkuda. Eelnevalt välja toodud päikesepaneeli kontrolleri on kahe USB väljundiga, millega on võimalik veel seadet kasutada nt nutitelefoni ja muude moodulite laadimiseks. Infot kontrolleri, aku ja päikesepaneeli ühendamise meetodi kohta on võimalik saada järgnevalt lingilt [7]. Autor soovib päikesepaneeli paigaldamiseks kasutada järgnevaid tarvikuid kirjeldatud Tabelis 5.

Tabel 5. Päikesepaneeli komponendid

Nimetus	Kogus	Hind (€)
Päikesepaneel polükristall 10W 18.2V 0.59A 354*251*17mm	1 tk	17
Päikesepaneeli kontrolleri LCD 20A 12V/24V USB	1tk	42
	Kokku	59

Päikesepaneeli ühendamise peaks olema korrektne, sest kui ühendada paneel akuga jadamisi ainult läbi takisti, siis võib aku ennast üle laadida ning tagajärjeks võib aku rikneda [46]. Selleks, et vältida seda olukorda tuleks ühenda MPPT (maximum power point tracking) kontrolleri. Tsiteerides Mihkel Salem: “MPPT ehk *maximum power point tracking*-uga kontrolleri, mis otsib maksimaalse võimsuse punkti ehk korrutab pinget ja voolu ning nii saadakse akudesse 30...40% rohkem energiat” [46, lk 17]. Rohkem infot päikesepaneeli kohta saab siit [45], [47].

Lõputöö prototüübi arendamiseks oleks töö praktilises osas suureks abiks lisaks päikesepaneelile veel kasutusele võtta tuulegeneraator. Prototüübile oleks väiketuulikut suur kasu. Väiketuulik ehk seade, mis muudab tuuleenergia elektrienergiaks, mille kogukõrgus Eestis peab jääma kuni 30m. Selleks, et tuuleturbiinid hakkaks elektrit genereerima on vaja vähemalt 3-4 m/s tuult [18, lk 21]. Tsiteerides Anti Tiiki: “Suurbritannias ja mujal maailmas tehtud uuringutele tuginedes võib väita, et avamaastikule (hajaasustuspiirkonda) püstitatud väiketuuliku tootlikkus on 17–30 % võimalikust, sõltuvalt tuuleoludest ja generaatori juhtimislahendusest. Tiheasustuslal või lausa linnas jääb väiketuuliku tootlikkus alla 10 % ning selle põhjus on turbulentses tuules.” Autor sai teada rohkem informatsiooni aku valikust ning energia jääkidest järgmiselt veebilehelt [16].

Kuna ilmastikku on raske ette näha, siis valgusfoori kontrolleri reaalses kasutuses aitaks kaasa, kui oleks nii tuulegeneraatori, kui ka päikesepaneeli ühendus. Selleks, et saaks mõlemat süsteemi omavahel ühendada tuleks nende vahele lisada eriotstarbeline kontrolleri. Selle abiga saaks hakata tootma vahelduvat voolu, ning millega saaks laadida akusid. Tuulegeneraatori ja päikesepaneeli ühenduste kohta on võimalik tutvuda järgmisest allikast [15].

6.2 Antenn

Raadiosignaali edastuse parendamiseks oleks vaja kasutusele võtta antenni. Töö tegija ei kasutanud lisa antenni moodulite jaoks, kuid näiteks müügivõrgus on saadaval kvaliteetseid WiFi antenne, mis tagavad usaldusväärse signaali ülekande. Antenne võiks panna nii raadiosaatjatele, kui ka raadiovastuvõtjatele [58]. Signaali edastuse kohta käiva info leiate järgnevalt lehelt [43]. Autor leidis allika, milles viidi läbi katse, kus testiti signaali edastust. Seade oli kokku pandud NRF24L01 moodulitega, mille sagedus oli 2.4 GHz, signaali võimendi ja antenniga. Katse tehti järgmiselt, et võeti kaks moodulit üks, kus oli raadiosaatja koos signaali võimendi ja antenniga, mis saatis koguaeg signaali maksimaalse võimsusega ja minimaalse kiirusega ning teiseks mooduliks oli vastuvõtja, mille külge oli ühendatud ekraan. Ekraani peal oli näha, kui palju pakette vastuvõtja võttis vastu. Esimene katse oli tehtud sisseehitatud antenniga, mille tingimus oli 600 meetrit ning signaali edastuse tulemuseks oli umbes 90%. Teiseks katse distantiks oli 1 kilomeeter ja sisseehitatud antenniga ühendus moodulite vahel kadus, aga kui moodulile lisati 10 sentimeetriline SMA antenn, siis signaali edastuse tulemus paranes kuni 100% [24].

6.3 Numbripaneel ja patarei nupp

Prototüüpi saaks veelgi paremaks muuta sellega, kui panna valgusfoorile 4x4 *matrix keypad* ehk väike numbripaneel. Sellega oleks võimalik teetöölisel ise manuaalselt määrata ooteaega. See muudaks ooteaja muutmise palju kergemaks, kuna siis ei oleks vaja minna spetsiaalsesse programmi, et muuta programmeerimiskoodi. Turvalisuse tagamiseks on võimalik välja mõelda *master*-kood. Vastav kood tuleks sisestada numbripaneeli, mis asub valgusfoori küljes ning õige *master*-koodi sisestamise järel avaneb menüü, kus ongi teetöölisel võimalik sisestada ooteaeg. Rohkem infot ühendamise ja programmeerimise kohta sai töö autor järgmiselt lingilt [51]. Raadiosaatjale ja -vastuvõtjale väikese nupu lisamine võimaldaks näha aku staatust. See parendaks prototüüpi nii, et oleks võimalik jälgida aku laetavust, kuna raadiosaatja aku tühjaks saamisel raadiovastuvõtja kaotab ühenduse ning valgusfoori töörežiimi muutub.

6.4 LCD

Autojuhtide informeerimiseks on võimalik ühendada LCD valgusfoorile, tänu millele näeksid juhid ooteaega ning selle põhjust. Kuna NodeMCU V3 Lolin moodulil ei ole piisavalt andmeväljundeid LCD ühendamiseks, siis on olemas I2C LCD üleminek, millega saaks vähendada LCD väljundeid nelja väljundini. Valgusfoori küljes olevale LCD-le on võimalik lisada nupp, millega saab raadiovastuvõtjat teavitada liiklusolukorrast. Näiteks kui objektile on saabunud suur kaubaauto, mille tõttu on seiskunud liiklus, siis raadiovastuvõtjal olevale nupule vajutades lisatakse teatud ooteaega juurde mõlemale teesuunale. Samuti tuleb autojuhtidele ekraanile teade, mis põhjusel on ooteaega pikendatud. Kui kaup on maha laetud, siis vajutades kõrvalolevat nuppu, saab raadiovastuvõtja signaali tagasi töörežiimile minemiseks. LCD ühendamise kohta on võimalik tutvuda järgneval lingil [50].

6.5 MAC aadress ja ooteaja seadistamine

Prototüübi paremaks muutmiseks võib juurde lisada ühe raadiovastuvõtja, nii saaks kasutada kaht või enam eraldiseisvat valgusfoori. Nii oleks võimalik valgusfoorid tuua autojuhtidele lähemale, et juhid näeksid paremini lugeda LCD-l olevaid märkusi. Samuti raadiovastuvõtja töötamine aku pealt võimaldab objektile mugavalt muuta valgusfooride asukohta.

Igal NodeMCU V3 Lolin moodulil on oma MAC (*Media Access Control Address*, Meediumipöörduse juhtimise aadress) aadress, mida saab kasutada informatsiooni edastamiseks kahe raadiovastuvõtja vahel. Eelnevalt peaks üks NodeMCU V3 Lolin moodul olema programmeeritud autori poolt peamooduliks. Teetöö tsooni jõudnud moodulitelt "küsit" peamoodul nende MAC aadressit. Peamoodulit saab eelnevalt programmeerida teise NodeMCU V3 Lolin raadiovastuvõtja MAC aadressiga, millega peamooduli raadiovastuvõtja hakkab võrdlema teiste moodulite MAC aadressitega, mis talle saadetakse raadiosaatja poolt. Juhul kui peamooduli raadiovastuvõtja saab sama MAC aadressi, mis oli talle eelnevalt programmeeritud, siis peamoodil hakkab saatma käskusid. Näiteks milline on teise valgusfoori ehk raadiovastuvõtja ooteaeg ja, mis töörežiimil valgusfoor peab töötama.

Kokkupandud seadmel on hetkel ooteaja seadistamine kliendi jaoks ebamugav. Praegu toimub ooteaja seadistamine nii, et töötajal või kliendil on vaja avada programmeerimiskood ja käsitsi muuta koodi sisu. Koodi muutmisel ei ole hetkel midagi rasket, ainult viie numbriga sisestamine, kuid klient ei peaks üldse saama koodi sisu muutma, sest nii võivad tekkida süsteemi vead. Selle parandamiseks on vaja programmeerimiskoodi muuta nii, et pärast koodi üles laadimist süsteem peaks küsima ooteaja kestvust.

6.6 Prototüübi veekindlus ja vastupidavus

Kuna teetöölised töötavad välitingimustes, siis tuleks sellega arvestada prototüübi tegemisel tulevikus. Ilmastikuolud on väga erinevad ning arvestades seda, peaks seade olema veekindel. Kuna on suur oht, et tööline võib seadmega ehk raadisaatjaga kuhugi pihta minna või see võib maha kukkuda, siis tuleks teha vastupidav seade. Vaadates veekindlust ja vastupidavust, siis esiteks on vaja konstrueerida väike karbik, mis on hermeetiliselt kinni pandud ja samas ka vastupidav. Probleem võib tekkida sellega, et tööline võib unustada oma seadme koju või näiteks objektile, selleks oleks vaja tööriide külge õmmelda väike krõpsuga tasku, kuhu seade sisse panna. Koht, kuhu tasku õmmelda tuleks valida nii, et see karbik ei jääks segama töölise töö tegemisel. Selleks kohaks võib näiteks olla õlavars.

6.7 Prototüübi pilveteenusega ühendamine

Lõputöö prototüübi süsteemi kasutades RTC moodulit saaks täiendada ning muuta paremaks, kuna hetkel on võimalik jälgida töötajaid kasutades programmi Arduino IDE arenduskeskkonda. Kuid selline lahendus ei ole õige realses tööolukorras, kuna see on mõeldud tarkvaraarendajate töövahendiks, mitte teederemondi firma ülemuste töövahendiks. Kui automatiseeritud valgusfoori prototüüpist saata andmed *Easy IOT server*-isse, siis oleks võimalik ülemustel reaalajas jälgida läbi pilveteenuse haldusliidese töötajate viibimist objektile näiteks kuu jooksul. Järgneval lingil on võimalik tutvuda, kuidas seadistada *Easy IOT server*-it [13].

6.8 Prototüübi võrgu turvalisus

Kuna tegemist on prototüübiga, mis on algusjärgus ning protsess, mida lõputööga luuakse on ajutine, seega turvariskid on madalad. Kuna lõputöö eesmärkide hulka ei kuulunud võrgu turvalisus, siis autor ei rääkinud sellest põhjalikumalt. Kuid selles peatükis autor toob välja lahenduse, kuidas saaks võrku teha turvalisemaks. Selleks lahenduseks on WPA (*WiFi Protected Access*, WiFi kaitstud juurdepääs), mis on võrgu krüpteerimise tüüp, mis kasutab terviklikku ajalist võtme protokollit TKIP (*Temporal Key Integrity Protocol*). Seda kasutatakse, et muuta veelgi turvalisemaks krüpteerimist. WPA-d on muudetud veelgi turvalisemaks ning on saadud WPA2, mida on ka kergem seadistada. Põhiline erinevus WPA2 on see, et ta kasutab laiemat krüpteerimisstandardit (AES)-i TKIP asemel.

Kui võrrelda WPA ja WPA2, siis saab järgmised erinevused:

- WPA- on vähem turvalisem meetod krüptimisel ning nõuab lühemat parooli, mis muudabki selle vähem turvalisemaks.
- WPA2- on uuendatud variant WPA-st, mis kasutab krüpteerimiseks AES-i ning pikki paroole, et muuta võrku turvalisemaks.

Rohkem infot võrgu kaitsmise kohta on võimalik leida järgnevalt lingilt [22]. WPA2 seadistamine NodeMCU V3 mooduliga saab tutvuda järgnevalt lingilt [23]. Kuid tuleb arvestada sellega, et saadetud pakettide AES krüpteerimise ja dekrüpteerimise peale tuleb kulutada lisa energiat ehk akud saavad kiiremini tühjaks.

7. Kokkuvõte

Lõputöö põhieesmärgiks oli luua automatiseeritud valgusfoori prototüüp, millega saaks automatiseerida teetööde tsoonis oleva valgusfoori töörežiimi. Töö käigus arendati välja modifitseeritud valgusfoori automaatika, kasutades raadiosaatjat ja -vastuvõtjat. Viimaseks seadistati need tarkvaraliselt omavahel ning testiti.

Testi tulemuste põhjal võib väita, et antud lõputöö käigus välja arendatud automatiseeritud valgusfoori prototüüp on võimeline:

- vahetama töörežiimi sõltuvalt sellest, kas teetööline on raadiovastuvõtja levialas või mitte;
- kasutades lõputöös loetletud komponente, suutsid moodulid omavahel signaali jagada üksteisega avatud tsoonis ulatusega 75 meetrit;
- signaal levis läbi väikeste takistuste moodulite vahel, kuid läbi metallistesemetete ei levinud;
- autor suutis saavutada raadiosaatja kompaktsuse, mis mahuks rinnataskusse ning massiks oli 100 grammi;
- raadiovastuvõtja aku suutis pidada vastu 2 päeva 23 tundi ja 15 minutit 10000 mAh ja 5 V akupangaga;
- raadiosaatja aku pidas vastu 21 tundi 1350 mAh ja 3.7 V akupangaga.

Aku kestvust mõõdeti mõlema mooduli puhul töörežiimis ehk infovahetus toimus samal ajal, kui testiti aku kestvust. Autor testis prototüüpi, et mõõta selle energiatarvet. Raadiovastuvõtjal oli voolutarve 0,08 A ja 3,96 V juures ning vastuvõtjal vastavalt 0,06 A ja 3,96 V. Lõputöö prototüüp sai oma ülesandega hakkama.

Kasutatud kirjandus

- [1] Seno Panjaitan, "OPERATION MODES HANDLING IN DISTRIBUTED AUTOMATION SYSTEMS", 15.06.2007 https://www.uni-saarland.de/fileadmin/user_upload/Professoren/FreyG/GF_SP_DCDS_jun_2007.pdf
- [2] IBM, "What is automation?" <https://www.ibm.com/topics/automation>
- [3] Mikell P. Groover, "Automation", 22.10.2020 <https://www.britannica.com/technology/automation>
- [4] Rob Stone, "IT Automation – What You Should Know" <https://www.mushroomnetworks.com/blog/it-automation-what-you-should-know/>
- [5] Kurt Kiefer, "ESP433: WiFi to 433MHz bridge", 23.02.2018 <https://www.c-sharpcorner.com/article/reading-input-from-serial-monitor-in-arduino/>
- [6] Sherenaz Al-Haj Baddar, "Wireless Communication Protocol", 2018 <https://www.sciencedirect.com/topics/engineering/wireless-communication-protocol>
- [7] vuaeco, "Charge Li-ion batteries with regular Charge controller made for Lead Acid batteries", 26.06.2016 https://www.youtube.com/watch?v=cBAY_dSpd5Y&t=0s&ab_channel=vuaeco
- [8] osoyooproduct, "NodeMCU Basic Project-Blink a LED" <https://www.instructables.com/NodeMCU-Basic-Project-Blink-a-LED/>
- [9] Aleksandr Pikalo, "NodeMCU-V3", 10.03.2021 <https://github.com/Apikalo77/NodeMCU-V3>
- [10] Jan Mallari, "HOW TO USE A REAL-TIME CLOCK MODULE WITH THE ARDUINO" <https://www.circuitbasics.com/how-to-use-a-real-time-clock-module-with-the-arduino/>
- [11] Administrator, "Arduino DS3231 RTC Module Tutorial", 29.08.2018 <https://www.electronicshub.org/arduino-ds3231-rtc-module-tutorial/>
- [12] "Simple Arduino Projects For Beginners" <https://www.makerspaces.com/simple-arduino-projects-beginners/>
- [13] Roland Pelayo, "Display Sensor Data to NodeMCU Web Server", <https://www.teachmemicro.com/display-sensor-data-nodemcu-web-server/>
- [14] "NodeMCU ESP8266 Detailed Review" <https://www.make-it.ca/nodemcu-arduino/nodemcu-details-specifications/>
- [15] "500W 12V Off Grid Wind & Solar Panel Hybrid System w/ Controller" <https://www.eco-worthy.com/products/500w-12v-off-grid-wind-solar-panel-hybrid-system-w-controller>

- [16] ANTI ТИИК, “Väikeste tuulegeneraatorite kasutusvõimalustest Eestis“, 13.09.2019 <https://keskkonnatehnika.ee/vaikeste-tuulegeneraatorite-kasutusvoimalustest-ees/is/>
- [17] СЕРГЕЙ, “Подключение DS3231 к NodeMCU (ESP8266)”, 13.03.2020 <https://robotchip.ru/podklyuchenie-ds3231-k-nodemcu/>
- [18] Kristina Aprelkova, “Väiketuulikute tasuvuse arvutamine (vajalike tuuletingimuste ja omatarbena kasutatud elektrienergia leidmine)”, 2016 <https://digikogu.taltech.ee/et/Item/3361b539-9494-4dfe-96be-818ccae076f9>
- [19] Julien, “COMMUNICATION BETWEEN TWO NODEMCU TOO SLOW??”, 8.07.2017 <https://www.esp8266.com/viewtopic.php?p=68132>
- [20] “How to Setup the ESP8266 Arduino IDE with the NodeMCU V3 (ESP8266 ESP12e)” <https://www.best-microcontroller-projects.com/esp8266-arduino-ide.html>
- [21] “Model SA100 Dual/IP66 Remote EC ”, 13.09.2008 <https://fccid.io/ANATEL/01405-10-00594/Manual-T/2802991C-F9E0-4E26-9EEE-666800D55840/PDF>
- [22] “WPA vs WPA2: Which WiFi Security Should You Use?”, 8.08.2020 <https://www.google.com/amp/s/www.pandasecurity.com/en/mediacenter/security/wpa-vs-wpa2/amp/>
- [23] simkard69, “WPA2 enterprise with credentials auth ok with radius but still get STATION_WRONG_PASSWORD after 30s“, 18.11.2019 <https://github.com/esp8266/Arduino/issues/6803>
- [24] Заметки Ардуинщика, ”Arduino and nRF24L01. Wiring and wireless control by Arduino”, 24.11.2016 <https://www.youtube.com/watch?v=ACVtKDjVXS4&feature=youtu.be>
- [25] youmat, “Free battery icon Stock Photo” <https://www.freeimages.com/photo/battery-icon-1241933gmail>
- [26] “led png LED Light Transparent Background | PNG Mart” <https://outlets.eoutlet2021.com/content?c=led%20png&id=15>
- [27] “Vector - green led icon on white background. led sign. light emitting diode. flat style. LED.” https://www.123rf.com/photo_142833992_stock-vector-green-led-icon-on-white-background-led-sign-light-emitting-diode-flat-style-led-.html
- [28] Achim Pieters, “NodeMcu v3 – ESP-12E Development Board” <https://www.studiopieters.nl/nodemcu-v3-esp-12e-development-board/>
- [29] howardthegeek, “Yellow LED Lamp (Off) PNG icon”, 28.10.2017 <https://www.iconspng.com/images/yellow-led-lamp-off/yellow-led-lamp-off.jpg>
- [30] Ардуино для начинающих, “Часы rtc ds3231 подключение к ардуино”, 6.03.2020 <http://arduino-kid.ru/blog/ds3231-podklyuchenie>
- [31] Adnan Aqeel “Introduction to NodeMCU V3” 11.10.2018 <https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html>

- [32] Владимир Приходько, “Явления влияющие на прохождение радиоволн”
<https://www.qrz.ru/solar/articles/ew8au.shtml>
- [33] Тамара Кравченко, “Переносные светофоры: в автоматическом режиме, а все равно человек стоит, останавливает непослушных”, 6.09.2018
<http://topnews.cn.ua/society/2018/09/10/107637.html>
- [34] Green Robot Machinery, ”Automated Railroad Crossing“, 30.06.2015
<https://www.hackster.io/GreenRobot/automated-railroad-crossing-093072>
- [35] “Радиоволны и распространение волн”, 12.11.2018
<https://meanders.ru/radiovolny.shtml>
- [36] „NodeMCU ESP8266 Detailed Review“, <https://www.make-it.ca/nodemcu-arduino/nodemcu-details-specifications/>
- [37] Basamma B, “HC-05 Serial Bluetooth Module“, 11.01.2021
<https://electrosome.com/hc-05-serial-bluetooth-module/>
- [38] “433 MHz RF Transmitter Module”, 21.02.2018 <https://components101.com/433-mhz-rf-transmitter-module>
- [39] ООО "КОМКОН ТРАФИК", „Traffic controllers“,
<http://komkon.ua/en/products/controllers/>
- [40] ООО "КОМКОН ТРАФИК", “Mobile traffic lights”,
<http://komkon.ua/en/products/controllers/sdm/>
- [41] Hans, „macOS - Aruino IDE - How to fix "pyserial or esptool directories not found next to this upload.py tool" error (ESP8266)“, 2.03.2021
<https://www.tweaking4all.com/forum/arduino/mac-os-aruino-ide-how-to-fix-pyserial-or-esptool-directories-not-found-next-to-this-upload-py-tool-error-esp8266/>
- [42] “Getting Started with ESP-NOW (ESP8266 NodeMCU with Arduino IDE)”
<https://randomnerdtutorials.com/esp-now-esp8266-nodemcu-arduino-ide/>
- [43] atesmeH, “A Guide for Understanding Range in RF Devices”, 21.01.2015
<https://www.ecnmag.com/article/2015/01/guide-understanding-range-rf-devices>
- [44] Marco Zennaro, “Introduction to WiFi Networking”
<https://www.internetsociety.org/wp-content/uploads/2017/10/Wifi-Intro-v3.pdf>
- [45] Jefim Birjuk, “ÜHE JA KAHE PÕÖRAMISTELJEGA PV-PANEELIDE ASENDI MUUTMISE JUHTIMISSÜSTEEMIDE TÖÖ TULEMUSTE VÕRDLUS”, 2019
http://dspace.emu.ee/xmlui/bitstream/handle/10492/5157/Jefim_Birjuk_2019MA_EK_täistekst.pdf?sequence=1&isAllowed=y
- [46] Mihkel Salem, “ELAMU AUTONOOMNE ELEKTRIVARUSTUSSÜSTEEM“, 2017
https://dspace.emu.ee/xmlui/bitstream/handle/10492/3574/Mihkel_Salem_2017BA_EK_täistekst.pdf?sequence=1&isAllowed=y

- [47] "SIZING YOUR CHARGE CONTROLLER & POWER INVERTER"
<https://resystech.com/sizing-your-charge-controller--power-inverter.html>
- [48] Артур Сальбах, "NodeMCU (ESP8266) для начинающих: что такое, как подключить", 21.04.2017 <http://edurobots.ru/2017/04/nodemcu-esp8266/>
- [49] Mouser Electronics, "RF Wireless Technology" <https://www.mouser.ee/applications/rf-wireless-technology/>
- [50] CodeChamp, "Interface I2C LCD Using NodeMCU"
<https://www.instructables.com/Interface-LCD-Using-NodeMCU/>
- [51] "ESP8266: Interfacing with a 4x4 Matrix Keypad",
<https://techtutorialsx.com/2017/03/18/esp8266-interfacing-with-a-4x4-matrix-keypad/>
- [52] TheElectromania, "Programming ESP8266 ESP-12E NodeMCU Using Arduino IDE - a Tutorial" <https://www.instructables.com/Programming-ESP8266-ESP-12E-NodeMCU-Using-Arduino-/>
- [53] Arduino, "Installing Additional Arduino Libraries"
<https://www.arduino.cc/en/guide/libraries>
- [54] Ivan Grokhotkov, ESP8266WiFi library, 2017 <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
- [55] Arduino, "Wire Library" <https://www.arduino.cc/en/reference/wire>
- [56] Arduino, "RTCLib", <https://www.arduino.cc/reference/en/libraries/rtclib/>
- [57] "RF 433 MHZ TRANSMITTER AND RECEIVER (WITH DEMO)", 28.02.2019
<https://www.robocraze.com/blog/rf-transmitter-receiver-experiment.html>
- [58] tom_1, „How to add an external antenna to an ESP board“, 08.2019
<https://community.home-assistant.io/t/how-to-add-an-external-antenna-to-an-esp-board/131601>
- [59] NortonLifeLock, „What you need to do about the WPA2 Wi-Fi network vulnerability“, 18.01.2018 <https://us.norton.com/internetsecurity-emerging-threats-what-to-do-about-krack-vulnerability.html#:~:text=Security%20researchers%20have%20discovered,client%20that%20connects%20to%20it>
-

Lisa

Lisa 1-Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Aleksandr Pikalo

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Teehoolduse foori töörežiimide automatiseerimine ", mille juhendaja on Kalle Tammemäe.

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.

17.05.2021

Lisa 2- Raadiovastuvõtja kood

```
// Lisatakse vajalikud teeke
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <RTClib.h>

// Defineeritakse väljundid
#define LED0 2
#define R D0
#define G D3
#define Y D4
#define R1 D5
#define Y1 D6
#define G1 D7

const char* TKDssid = "ESPsoftAP_01"; // Luuakse
Serverile WiFi nimi
const char* TKDpassword = "pass0077"; // Luuakse
Serverile parool

WiFiServer TKDServer(9001); // Määratakse serverile port
WiFiClient TKDClient[200]; // Määratakse maksimaalne arv
raadiosaatjaid, //mis kuvab jälgimis
monitooringul

RTC_DS3231 rtc; // Luuakse RTC objekti

void setup()
{
  // Seadistatakse jadapordis
  Serial.begin(115200);
  Wire.begin(); // Kasutatakse, et jadapordile ei
  tuleks ette tundmatuid sümboleid

  // Seadistatakse väljundeid
  pinMode(LED0, OUTPUT);
  pinMode(D0, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D7, OUTPUT);

  // Prinditakse välja kiri, et seade läks tööle
  Serial.println();
}
```

```

Serial.println("I/O Pins Modes Set .... Done");

// Seadistatakse automaatne kellaaeg
rtc.begin();
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

// WiFi seadistamine
SetWifi("ESPsoftAP_01", "");

}

void loop()
{
  //Kutsutakse välja funktsioon
  IsClients();
}

void SetWifi(char* TKDssid, char* TKDpassword)
{
  // Kõikide eelnevate protsesside peatamine
  WiFi.disconnect();

  // Seadistatakse WiFi režiimi
  WiFi.mode(WIFI_AP_STA);
  Serial.println("WIFI Mode : AccessPoint Station");

  // WiFi tööle rakendamine
  WiFi.softAP(TKDssid, TKDpassword);
  Serial.println("WIFI < " + String(TKDssid) + " > ... Started");

  // Määratakse viivitusaeg
  delay(1000);

  // Serveri IP aadressi saamine
  IPAddress IP = WiFi.softAPIP();

  // Prinditakse Serveri IP aadress
  Serial.print("AccessPoint IP : ");
  Serial.println(IP);

  // Serveri käivitamine
  TKDServer.begin();
  Serial.println("Server Started");
}

//Luuakse uus funktsioon
void IsClients()

```

```

{

//Serverile programmeeritakse sisse SSID ja parool.
boolean result = WiFi.softAP(TKDssid, TKDpassword);
if(result == true)
{
    //Kui serveril on juba klient ühendatud
    if(TKDServer.hasClient())
    {
//Oodatakse kuna klient logib sisse SSID-ga ning parooliga,
// mis tal on programmeeritud
        WiFiClient TKDClient = TKDServer.available();
//Kui klient logib sisse, siis saab ühendust serveriga
        if(TKDClient.available())
        {
//Prinditakse välja kiri, et klient on saanud ühendust
serveriga
            String Message = TKDClient.readStringUntil('\r');

            // Prinditakse kuupäeva ja kellaaja struktuuri jadapordis
            DateTime now = rtc.now();
            Serial.print(now.day(), DEC);
            Serial.print('/');
            Serial.print(now.month(), DEC);
            Serial.print('/');
            Serial.print(now.year(), DEC);
            Serial.print(' ');
            Serial.print(now.hour(), DEC);
            Serial.print(':');
            Serial.print(now.minute(), DEC);
            Serial.print(' ');
            Serial.print('-');
            Serial.print(' ');
            Serial.println(Message);

// Kliendile vastamine Serveri poolt, et sõnum tuli kohale
            TKDClient.print("\nWe Got Your Message");
            TKDClient.flush();

// Ehitatakse kahe valgusfoori algoritm ning kutsutakse
välja funktsioonid

                Molemad_R_Pikk();           //Mõlemal valgusfooril põleb
punane tuli pika viivitusega
                VF_R_Y();                   //Esimesel valgusfooril
põleb punane ja kollane tuli, teisel
                // põleb punane tuli

```

```

        VF_G();          //Esimesel valgusfooril põleb
roheline tuli, teisel
                                // põleb punane tuli
        VF_G_Vilgub();      //Esimesel valgusfooril
vilgub roheline tuli, teisel
                                // põleb punane LED
        VF_Y();          //Esimesel valgusfooril põleb kollane
LED, teisel
                                // põleb punane LED
        Molemad_R_Luhike(); //Mõlemal valgusfooril põleb
punane LED lühikese
                                //viivitusega
        VF1_R1_Y1();      //Teisel valgusfooril põleb punane
ja kollane LED, esimesel
                                // põleb punane LED
        VF1_G1();          //Teisel valgusfooril põleb
roheline LED, esimesel
                                // põleb punane LED
        VF1_G1_Vilgub();   //Teisel valgusfooril vilgub
roheline LED, esimesel
                                // põleb punane LED
        VF1_Y1();          //Teisel valgusfooril põleb
kollane LED, esimesel
                                // põleb punane LED
    }
}
else{
        Molemad_R_Tuhi();   //Mõlemal valgusfooril põleb
punane LED lühikese
                                //viivitusega
        VF_R_Y();          //Esimesel valgusfooril
põleb punane ja kollane LED,
                                //teisel põleb punane LED
        VF_G_tuhi();       //Esimesel
valgusfooril põleb roheline LED, teisel
                                // põleb punane LED
        VF_G_Vilgub();     //Esimesel valgusfooril
vilgub roheline LED, teisel
                                // põleb punane LED
        VF_Y();          //Esimesel valgusfooril
põleb kollane LED, teisel
                                // põleb punane LED
        Molemad_R_Tuhi();   //Mõlemal valgusfooril põleb
punane LED lühikese
                                //viivitusega
        VF1_R1_Y1();      //Teisel valgusfooril
põleb punane ja kollane LED,
                                //esimesel põleb punane LED

```



```

        VF1_G1_tuhi();                //Teisel valgusfooril
põleb roheline LED, esimesel
        VF1_G1_Vilgub();            // põleb punane LED
roheline LED, esimesel            //Teisel valgusfooril vilgub
        VF1_Y1();                    // põleb punane LED
kollane LED, esimesel            //Teisel valgusfooril põleb
        VF1_Y1();                    // põleb punane LED
    }
}
}

```

//Luuakse eraldi valgusfoori funktsioonid, et hiljem oleks lihtsam nendest luua valgusfoori algoritmi

//Luuakse funktsioonid mõlemale valgusfoorile, mis ei muutu

```

void VF_R_Y(){
    digitalWrite(R, HIGH);           //Esimese valgusfoori punase LED-i
süttimine
    digitalWrite(Y, HIGH);           //Esimese valgusfoori kollase LED-
i süttimine
    digitalWrite(G, LOW);            //Esimese valgusfoori rohelse
LED-i kustumine
    digitalWrite(R1, HIGH);          //Teise valgusfoori punase LED-i
süttimine
    digitalWrite(Y1, LOW);           //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, LOW);           //Teise valgusfoori rohelse LED-i
kustumine
    delay(500);                       //Funktsiooni viivitus aeg
}
void VF_G_Vilgub(){
    digitalWrite(R, LOW);             //Esimese valgusfoori punase LED-i
kustumine
    digitalWrite(Y, LOW);            //Esimese valgusfoori kollase LED-
i kustumine
    digitalWrite(G, LOW);            //Esimese valgusfoori rohelse
LED-i kustumine
    digitalWrite(R1, HIGH);          //Teise valgusfoori punase LED-i
süttimine
    digitalWrite(Y1, LOW);           //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, LOW);           //Teise valgusfoori rohelse LED-i
kustumine
    delay(250);                       //Funktsiooni viivitus aeg
    digitalWrite(R, LOW);            //Esimese valgusfoori punase LED-i
kustumine
}

```

```

        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, HIGH); //Esimese valgusfoori roheline
LED-i süttimine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, HIGH); //Esimese valgusfoori roheline
LED-i süttimine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg

```

```

        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, HIGH); //Esimese valgusfoori roheline
LED-i süttimine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg
    }
    void VF_Y() {
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, HIGH); //Esimese valgusfoori kollase LED-
i süttimine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(500); //Funktsiooni viivitus aeg
    }
    void VF1_R1_Y1() {
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustutamine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, HIGH); //Teise valgusfoori kollase LED-i
süttimine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(500); //Funktsiooni viivitus aeg
    }
    void VF1_G1_Vilgub() {
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine

```

```

    digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
    digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
    digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
    delay(250); //Funktsiooni viivitus aeg
    digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
    digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
    digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
    digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
    digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, HIGH); //Teise valgusfoori rohelise LED-i
süttimine
    delay(250); //Funktsiooni viivitus aeg
    digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
    digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
    digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
    digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
    digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
    delay(250); //Funktsiooni viivitus aeg
    digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
    digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
    digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
    digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
    digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
    digitalWrite(G1, HIGH); //Teise valgusfoori rohelise LED-i
süttimine
    delay(250); //Funktsiooni viivitus aeg
    digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine

```

```

        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(250); //Funktsiooni viivitus aeg
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, HIGH); //Teise valgusfoori roheline LED-i
süttimine
        delay(250); //Funktsiooni viivitus aeg
    }
    void VF1_Y1() {
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori roheline
LED-i kustumine
        digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
        digitalWrite(Y1, HIGH); //Teise valgusfoori kollase LED-i
süttimine
        digitalWrite(G1, LOW); //Teise valgusfoori roheline LED-i
kustumine
        delay(500); //Funktsiooni viivitus aeg
    }

    // Luuakse funktsioonid, kui objektile ei ole kedagi

    void Molemad_R_Tuhi(){
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine

```

```

        digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
        delay(1000); //Funktsiooni viivitus aeg
    }
    void VF_G_tuhi(){
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, HIGH); //Esimese valgusfoori rohelise
LED-i süttimine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
        delay(500); //Funktsiooni viivitus aeg
    }
    void VF1_G1_tuhi(){
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
        digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, HIGH); //Teise valgusfoori rohelise LED-i
süttimine
        delay(500); //Funktsiooni viivitus aeg
    }

    // Luuakse funktsioonid, kui objektil käib töö

    void Molemad_R_Pikk(){
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine

```

```

        digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
        delay(3000); //Funktsiooni viivitus aeg
    }
    void Molemad_R_Luhike(){
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW); //Teise valgusfoori rohelise LED-i
kustumine
        delay(2000); //Funktsiooni viivitus aeg
    }
    void VF1_G1() {
        digitalWrite(R, HIGH); //Esimese valgusfoori punase LED-i
süttimine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, LOW); //Esimese valgusfoori rohelise
LED-i kustumine
        digitalWrite(R1, LOW); //Teise valgusfoori punase LED-i
kustumine
        digitalWrite(Y1, LOW); //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, HIGH); //Teise valgusfoori rohelise LED-i
süttimine
        delay(1000); //Funktsiooni viivitus aeg
    }
    void VF_G(){
        digitalWrite(R, LOW); //Esimese valgusfoori punase LED-i
kustumine
        digitalWrite(Y, LOW); //Esimese valgusfoori kollase LED-
i kustumine
        digitalWrite(G, HIGH); //Esimese valgusfoori rohelise
LED-i süttimine
        digitalWrite(R1, HIGH); //Teise valgusfoori punase LED-i
süttimine

```

```
        digitalWrite(Y1, LOW);           //Teise valgusfoori kollase LED-i
kustumine
        digitalWrite(G1, LOW);         //Teise valgusfoori roheline LED-i
kustumine
        delay(1000);                   //Funktsiooni viivitus aeg
    }
```


Lisa 3- Raadiosaatja kood

```
// Lisatakse teatud teegi
#include <Wire.h>
#include <ESP8266WiFi.h>

// Defineeritakse väljundid
#define LED0 2
#define TWI_FREQ 400000L

//Defineeritakse muutujaid
int TootajaState;
int LastButtonState = LOW;
int LastDebounceTime = 0;
int DebounceDelay = 50;
const String TootajaNimi = "NIKITA";

int LEDState = LOW;
unsigned long CurrMillis = 0;
unsigned long PrevMillis = 0;
unsigned long Interval = 1000;

//Antakse kliendile IP aadressi, SSID ja parooli, ühenduse saamiseks
char* TKDssid = "ESPsoftAP_01";
char* TKDpassword = "pass0077";
IPAddress TKDServer(192,168,4,1);
WiFiClient TKDClient;

// Seadistuse funktsioon
void setup()
{
    Wire.begin();

    Wire.setClock(TWI_FREQ);

    //Seadistatakse jadapordi
    Serial.begin(115200);

    //Seadistatakse väljundeid
    pinMode(LED0, OUTPUT);
    digitalWrite(LED0, !LOW);

    //Seadistatakse kliendi, et ta oleks saatja rollis
    Serial.println("\nI/O Pins Modes Set .... Done");
    WiFi.mode(WIFI_STA);
    WiFi.begin(TKDssid, TKDpassword);
}
```

```

//Prinditakse välja, mis võrgule ühendatakse
Serial.println("!--- Connecting To " + WiFi.SSID() + " ---!");

//Seadistatakse sisseehitatud LED-i
digitalWrite(LED0, !HIGH);

//Prinditakse välja võrgu info, millega sai klient ühendatud
Serial.println("!-- Client Device Connected --!");
Serial.println("Connected To      : " + String(WiFi.SSID()));
Serial.println("Signal Strenght   : " + String(WiFi.RSSI()) +
" dBm");
Serial.print ("Server IP Address : ");
Serial.println(TKDServer);
Serial.print ("Device IP Address : ");
Serial.println(WiFi.localIP());
}

//Luuakse tsükli osa
void loop()
{
  //Kutsutakse välja funktsiooni
  CheckConnectivity();
  TKDRequest();
}

//Luuakse võrgu ühenduse kontrollimise funktsiooni
void CheckConnectivity()
{
  //Kontrollitakse, kas seade on ühendust saanud
  while(WiFi.status() != WL_CONNECTED)
  {
    for(int i=0; i < 10; i++)
    {
      digitalWrite(LED0, HIGH); //NodeMCU V3 moodulil
LED-i süttimine
      delay(250);                //Funktsiooni viivitus
aeg
      digitalWrite(LED0, LOW); //NodeMCU V3 moodulil
LED-i kustumine
      delay(250);                //Funktsiooni viivitus
aeg
      Serial.print(".");
    }
    Serial.println("");
  }
}
}

```

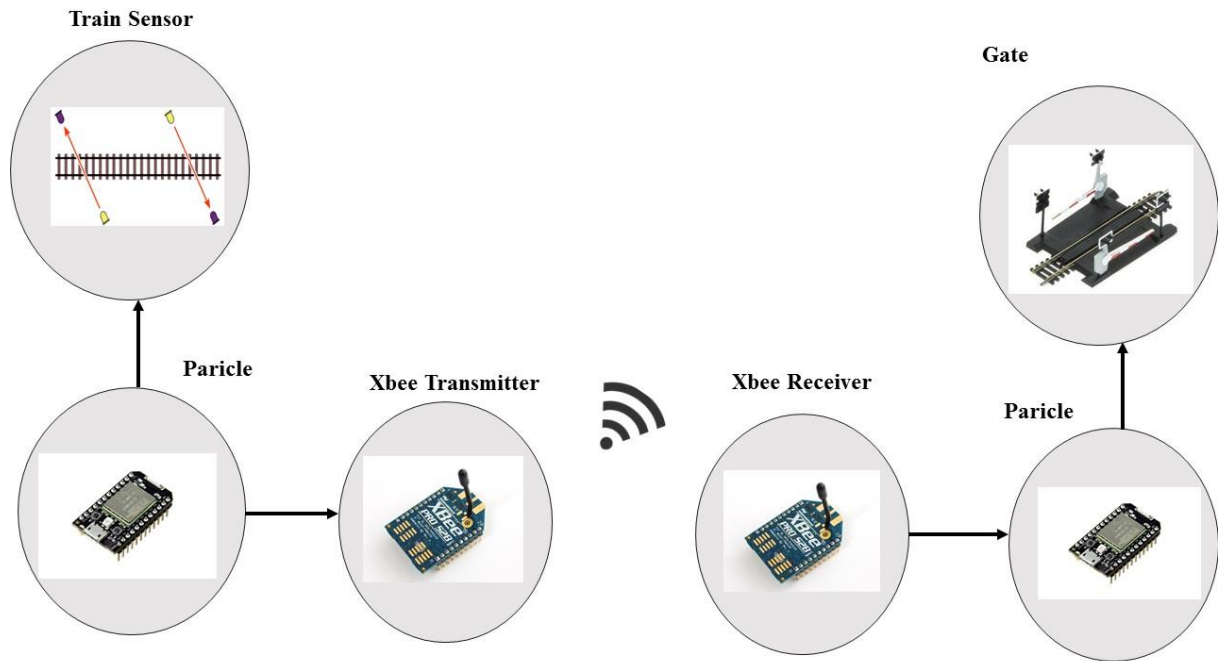
```

//Luuakse kirja saatmise funktsiooni
void TKDRequest()
{
    //Peatatakse kõik eelnevaid tegu
    TKDClient.stop();

    //Tehakse kontroll, kas klient on õigesse serverisse
    ühendatud.
    If ( TKDClient.connect(TKDServer, 9001) )
    {
        //Kui klient on saanud ühendust õige serveriga, siis
        saadetakse kirja nii serveri, kui ka kliendi
        //jadapordisse töötaja nimega, kes on ühendust saanud.
        Serial.println ("<" + TootajaNimi + "-CONNECTED>");
        TKDClient.println ("<" + TootajaNimi + "-CONNECTED>");
    }
}

```

Lisa 4- Skeem Turvalisus raudteel



Lisa 5 – KOMKON SDM seadme pilt

