



Ahmed Mahmoud Taha Abdelwahed

TurtleBot4 Lite -robotti

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tutkinto-ohjelman nimi

Insinöörityö

22.10.2023

Tiivistelmä

Tekijä: Ahmed Mahmoud Taha Abdelwahed
Otsikko: TurtleBot4 Lite -robotti
Sivumäärä: 38 sivua +1 liitettä
Aika: 22.10.2023

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Automaatiotekniikka
Ammatillinen pääaine: Ammatillisen pääaineen nimi
Ohjaajat: Timo Tuominen

Insinöörityössä perehdyttiin ROS2-väliohjelmistoon (Robot Operating System) ja otettiin käyttöön ROS:ia hyödyntävä TurtleBot4-robotti. Tämä robotti otettiin käyttöön Metropolia Ammattikorkeakoulun opetuskäyttöön.

Tavoitteena TurtleBot4 oli esitellä ja selittää TurtleBot4 -robottialustan ominaisuuksia, toimintoja ja käyttötapoja.

Tämä projekti keskittyi TurtleBot4-robotin tuntemiseen ja sen oikeanlaiseen käyttöön. Projektissa suoritin ja noudatin töitä huolellisesti ohjeiden mukaan, kun irrotin ja asensin robotin osat. Oikea oppinen, purkaminen ja kokoaminen on äärimmäisen tärkeää varmistaa, että laite toimii turvallisesti.

Asensin käyttöjärjestelmät ja muut tarvittavat ohjelmistot robotille. Tämä varmisti, että robotti oli valmis suorittamaan sille osoitetut tehtävät. Käytin menestyksellisesti Rviz2-ohjelmaa ja simulointia, mikä osoitti onnistuneen projektin testauksen ja vianetsinnän. Tämä projekti oli mielenkiintoinen haaste. Se vaati tarkkuutta sekä huolellisuutta yksityiskohtiin. Olen ylpeä siitä, miten tehtävä suoritettiin onnistuneesti.

Avainsanat: Robot Operating System, robotti, TurtleBot, SLAM, navigointi, Rviz2

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check-ohjelmalla.

Abstract

Author: Ahmed Mahmoud Taha Abdelwahed
Title: TurtleBot4 Lite –robot
ber of Pages: 38 pages+1 appendices
Date: 22 October 2023

Degree: BachelorofEngineering
Degree Programme: Nameofthedegreeprogramme
Professional Major: Name of the professional major
Supervisors: Timo Tuominen

In the engineering thesis, I delved into the ROS2 middleware (Robot Operating System) and implemented a TurtleBot4 robot that utilizes ROS. This robot was introduced for educational purposes at Metropolia University of Applied Sciences.

The goal with the TurtleBot4 was to introduce and explain the features, functions, and use cases of the TurtleBot4 robot platform.

This project primarily revolved around understanding the TurtleBot4 robot and its proper usage. I meticulously followed instructions during the disassembly and assembly of the robot's components. Proper handling during assembly and disassembly is crucial to ensure the safe operation of the device.

I installed operating systems and other necessary software on the robot, ensuring that it was ready to perform its designated tasks.

I successfully utilized the Rviz2 program and simulation, which proved instrumental in testing and troubleshooting the project. This project presented an intriguing challenge, demanding precision and attention to detail. I take pride in the successful completion of this task.

Keywords: Robot Operating System, robot, TurtleBot, SLAM, navigation, Rviz2

Sisällys

Lyhenteet

Sisältö

1	Johdanto	1
2	TurtleBot4.....	1
2.1	Anturit.....	3
2.2	Yhden piirilevyn tietokone.....	4
3	Laitteen osien purkaminen ja kasaaminen.....	6
3.1	Robotin etuosa	8
3.2	Robotin.....	10
4	Virta pois.....	11
4.1	Wi-Fi-yhteyspisteen luominen.....	12
4.2	Koneen virta päälle.....	15
5	Ladataan päivitystä	19
5.1	Yhdistetään robotti Wi-Fi-verkkoon.....	22
5.2	Wi-Fi yhdistetty	23
6	Aloitetaan ROS2:n testaus	25
6.1	Create3-telakointi	31
7	Rviz2.....	33
	Lähteet	39
	Asennus ROS2 Galactic.....	1
	Creating your first node (Python) GALACTIC	2
	Sensors.....	2
	OAK-D	2
	Launch SLAM	2
	Launch Rviz2	2
	Rviz2 MODEL	2
	View Robot.....	2
	Launching navigation	2
	Launching Ignition Gazebo.....	3

Liitteet

Liite1:Liitteennimi Lii-

te2:Liitteennimi

Lyhenteet

AMCL: Adaptive Monte Carlo localization. Todennäköisyyspohjainen paikannus- menetelmä kaksiulotteisesti liikkuvalla robotilla.

LDS: Laser Distance Sensor. Laserilla toimiva etäisyysanturi.

ROS: Robot Operating System. Avointa lähdekoodia oleva robotiikan väliohjelmisto. Kokoelma ohjelmia robotiikkaohjelmistojen kehittämiseen.

SLAM: Simultaneous Localization and Mapping. Menetelmä, jossa karttaa luodaan arvioimalla nykyistä sijaintia omavalintaisessa tilassa.

SSH: Secure Shell. Salattuun tietoliikenteeseen tarkoitettu protokolla, jonka yleisin käyttötarkoitus on komentopäätteellinen etäyhteyden muodostaminen.

URDF: Unified Robot Description Format. XML-muotoinen tapa kuvata robotin malli.

SLAM: Samanaikainen sijainti ja kartoitus. Robotit ja ajoneuvot avulla tietäen niiden sijainti ympäristössä.

YAML: YAML Ain't Markup Language. Standardi ihmisystävällisen tiedon sarjallistamiselle.

1 Johdanto

Tavoitteena oli ottaa käyttöön TurtleBot4-robotti, jotta sitä voidaan käyttää opetuksessa Metropolia Ammattikorkeakoulussa. Työssä robotti purettiin ja koottiin uudelleen, ja siihen asennettiin tarvittava ohjelmisto. Oli tärkeää päivittää laite oikealla tavalla. Työskentelyyn kuului myös useita kokeiluja robottien kanssa, jotta voitaisiin ymmärtää niiden rajoitukset ja löytää parhaat asetukset niiden tehokkaaseen toimintaan.

2 TurtleBot4

Turtlebot4 Standard ja Lite ovat kaksi eri versiota Turtlebot4-robotista. Standard-versio sisältää kaikki moduulit ja anturit, joita tarvitaan monenlaisiin käyttötarkoituksiin, kuten kartanpiirtoon ja autonomiseen navigointiin.

Lite-versio puolestaan on yksinkertaisempi versio, joka sisältää vähemmän moduuleja ja antureita, mutta se on edullisempi ja helpompi kokoonpanna.

Lite-versio sopii paremmin aloittelijoille käyttöön, jossa tarvitaan vain perustoinnallisuudet. Sekä Standard että Lite voidaan räätälöidä erilaisilla antureilla ja ohjelmistoilla, jotta ne vastaavat tiettyä käyttötarkoitusta (kuva 1).

Työmallini on Turtlebot4 lite.



TurtleBot 4 Standard



TurtleBot 4 Lite

Kuva 1. TurtleBot4 Standard ja Lite

ROS2 on Robotti Operating Systemin seuraava versio, joka on suunniteltu vastaamaan yhä kasvaviin robotiikan ja automaation tarpeisiin.

ROS2:ssa on parannettu useita toiminnallisuuksia, kuten kommunikointia, turvallisuutta, modulaarisuutta ja skaalautuvuutta.

Kommunikointi on yksi ROS2:n päätavoitteista. ROS2:ssa on käytössä uusi kommunikointikerros, joka mahdollistaa erilaisten kommunikointiprotokollien käytön. Tämä mahdollistaa eri robotiikkajärjestelmien ja teknologioiden hallinnan ja yhteensopivuuden.

Turvallisuus on toinen ROS2:n tärkeä ominaisuus. ROS2:ssa on käytössä turvallinen ROS-paketti, joka mahdollistaa laitteiden ja järjestelmien turvallisen hallinnan.

Modulaarisuus on ROS2:n kolmas keskeinen ominaisuus. ROS2 mahdollistaa moduulien käytön, jotka ovat itsenäisiä yksiköitä, joita voidaan yhdistellä tarpeiden mukaan. Tämä mahdollistaa joustavan ja tehokkaan järjestelmän rakentamisen.

Tehokkuus on yksi ROS2:n tärkeimmistä ominaisuuksista. ROS2:ssa on käytössä uusi jakaantunut tiedonhallintajärjestelmä, joka mahdollistaa tehokkaan ja joustavan datan hallinnan.

ROS2:n avulla kehittäjät voivat helposti luoda ja hallita robotti- ja automaatio-sovelluksia. ROS2:n avoimen lähdekoodin tarjoama yhteisö mahdollistaa lisäksi kehittäjien yhteistyön ja hyödyntää muiden kehittäjien kehitystyötä.

ROS2:ssa on useita toiminnallisuuksia, jotka tekevät siitä erittäin hyödyllisen työkalun robotiikan ja automaatio-sovellusten kehityksen. Kommunikointi, turvallisuus, modulaarisuus ja tehokkuus ovat vain muutamia esimerkkejä ROS2:n tarjoamista ominaisuuksista.

TurtleBot4 on ROS-pohjainen robottialusta, joka sisältää useita tärkeitä komponentteja: Sisäänrakennettu tietokone robotin ohjaukseen ja ohjelmointiin.

2D-laserskanneri luo ympäristökartan, jotta robotti voi navigoida.

Samanaikainen sijainti ja kartoitus (SLAM): mahdollistaa paikannuksen ja ympäristökartan luomisen.

Moottorit Mahdollistavat robotin liikkeen. Inertiaalinen mittayksikkö (IMU): mittaa ympäristön liikehdintää robottien liikkeiden hallintaa varten. Kamera: mahdollistaa visuaalisen havainnoinnin ja seurannan. Mikrofoniryhmä: antaa äänipalautetta ja tunnistaa puheen.

Intel RealSense Depth Camera D435i: mahdollistaa syvyyden havainnoinnin, henkilöiden seurannan ja kasvojen tunnistuksen.

Nämä komponentit yhdessä mahdollistavat monipuoliset käyttötilanteet TurtleBot4:lle.

2.1 Anturit

Kuva 2 esittää RPLIDAR A1M8 on 360 asteen laservaloskanneri, joka havainnoi ympäristönsä jopa 12 metrin päästä. Se käyttää laseria valonsäteiden lähettämiseen ympärilleen.

Se kerää heijastumat takaisin, luo tarkan 2D-kartan ympäristöstä. Tämä anturi sopii erinomaisesti roboteille, mikä auttaa niitä tunnistamaan esteet ja liikkumaan turvallisesti.

RPLIDAR A1M8:n suurimmat edut ovat sen nopeus ja tarkkuus. Se voi tuottaa jopa 8000 näytettä sekunnissa, mikä tekee siitä sopivan nopeisiin ympäristöihin ja laajojen alueiden tarkkailuun. Se pystyy myös erottamaan pienimmätkin esteet ja objektit ympärillään.



Kuva 2. RPLIDAR A1M8

Kuva 3 esittelee OAK –D -Lite -kameran, joka on Luxoniksen tuote. Se käyttää 4K IMX214 -värisensoria ja OV7251-stereosensoreita, mikä mahdollistaa korkealaatuisten väri- ja syvyyskuvien tuottamisen. Kamerassa on Myriad X VPU, joka mahdollistaa tietokonenäkösovellusten käytön, objektien seurannan ja te-

koälymallien suorittamisen.

OAK-D-Lite on älykäs kamerajärjestelmä, joka hyödyntää kehittyneitä tietokoneen näköalgoritmeja ympäristön tunnistamiseen ja analysoimiseen. Se tarjoaa monipuolisia ominaisuuksia, kuten erilaisia kuvantamistapoja, liitäntöjä ja prosessointiteknologioita.

Kameran kehittyneet ominaisuudet ja suorituskyky mahdollistavat sen käytön monenlaisissa sovelluksissa, kuten robotiikassa, autonomisessa liikenteessä, teollisissa sovelluksissa ja tiedonkeruussa. OAK-D-Lite voi tunnistaa, analysoida ja seurata reaaliajassa ihmisiä, eläimiä ja muita kohteita hyödyntäen syväoppimista ja tekoälyä.

Kamerassa on monipuolisia liitäntöjä, kuten USB3.0, HDMI ja mikrofonitulo, mikä tekee siitä yhteensopivan erilaisten laitteiden ja järjestelmien kanssa. OAK-D-Liten kehittyneet teknologiat mahdollistavat sen käytön monissa erilaisissa sovelluksissa, kuten lisälaitteena iPadille, henkilökohtaisena robotissa ja älykäänä valvontajärjestelmänä.



Kuva 3. OAK-D-Lite

2.2 Yhden piirilevyn tietokone

RaspberryPi 4 Malli B on Raspberry Pi-perheen neljännen sukupolven yksikorttitietokone (kuva 4). Se on suunniteltu erilaisiin projekteihin ja sovelluksiin.

Sitä voidaan käyttää tietokoneohjelmointiin, elektroniikkaan, mediakeskuksiin ja LOT-sovelluksiin.

Pi 4 Malli B:ssä on huomattavasti tehokkaampi suoritin, enemmän RAM-muistia ja parempi grafiikkasuoritin verrattuna edelliseen sukupolveen.

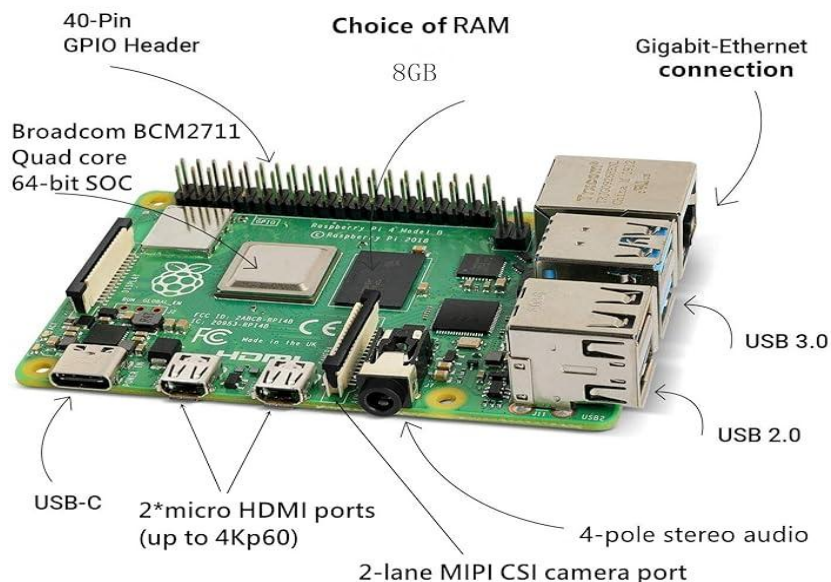
Sen suoritin käyttää neljää ydintä, ja se toimii 1,5 GHz:n taajuudella ja sen tueksi on

saatavilla 1 GB, 2 GB, 4 GB tai 8 GB LP DDR4 RAM-muistia riippuen version valinnasta. Niitä voi käyttää myös median toistamiseen UHD-resoluutiolla 4K:lla.

RaspberryPi4 Malli B: levyssä on runsaasti liitäntöjä mukaan lukien kaksi USB 3.0 -porttia, kaksi USB 2.0 -porttia, kaksi mikro-HDMI-porttia, 3,5 mm:n audioliitäntä, GigabitEthernet- portti ja kaksi mikro-USB-C porttia. Näistä toinen on tarkoitettu virtalähteelle ja toinen Display Port-videolähdölle tai tiedonsiirrolle. Suositellaan käyttämään laadukasta 5V/3A:n virtalähdettä varmistaaksesi sen toimivuuden.

Laite tukee myös langatonta WiFi 802.11ac- ja Bluetooth 5.0 -yhteyttä, mikä mahdollistaa langattoman tiedonsiirron ja laajennusmahdollisuudet. Tämä tekee siitä ihanteellisen valinnan LOT-projekteihin tai sellaisiin sovelluksiin, jotka vaativat langatonta tiedonsiirtoa.

Yksi tärkeä huomio RaspberryPi4 Malli B:ssä on sen suurempi virtatarve verrattuna aiempiin malleihin. Se on myös yhteensopiva useiden käyttöjärjestelmien kanssa, kuten Raspiaan, Ubuntu, OSMC ja Windows 10 IoTCore. Ne auttavat toteuttamaan erilaisia projekteja ja sovelluksia.



Kuva 4. Raspberry Pi 4 -tietokone.

Robotti on varustettu monipuolisella laitteistolla, joka mahdollistaa robotiikka-algoritmien kehittämisen ja testaamisen.

Robotin ohjelmisto perustuu ROS 2:een. Antureiden tuottamat tiedot kulkevat ROS 2 -julkaisujen kautta. Aktuaattoreita ohjataan ROS 2 -palveluiden ja tilaajien avulla.

Robotti tarjoaa valmiita autonomisia toimintoja, kuten telakointia, seinän seuraamista ja esteisiin reagoitua. Kaikki nämä toiminnot voidaan mukauttaa tai suorittaa ROS 2 -toimintojen ja parametrien avulla.

Create3 on suunniteltu ROS 2 -aloittelijoista edistyneisiin käyttäjiin. Siinä on Wi-Fi-, Bluetooth- ja Ethernet-ominaisuudet sekä USB-liitäntä rakentamisen aloittamiseen. Lisäksi siihen voidaan kiinnittää kuormia irrotettavan etulevyn avulla.

3 Laitteen osien purkaminen ja kasaaminen

TurtleBot4 Lite perustuu iRobot Create3 -oppimisalustaan, jossa on helppo muokattava integrointilevy tarvikkeiden kiinnittämiseen perusyksikköön lähelle OAK- kameraa ja RPLIDAR.

Tämä tila on olemassa reikälevylle käytettävissä tarvikkeiden kiinnittämiseen. Näiden 03.5 mm reikien väli on 10 mm. Vaihtoehtoisesti reikiä voidaan porata poistamalla Create3-integrointilevy. Robottien 3D-mallit ovat saatavilla GitHubissa. Ne voivat auttaa suunnittelussa.

Create3-integrointilevy voidaan poistaa seuraamalla alla olevia vaiheita. Irrotetaan USB- kaapelit, jotka on liitetty OAK- D-kameraan ja RPLIDAR: iin. Syötään nämä kaapelit robotin takaosassa olevasta aukosta. Kuvan 5 mukaisesti otettiin kaapelit irti kamerasta ja RPLIDAR:sta sekä levystä.



Kuva 5. USB-kaapelit

Etulevy poistetaan pyörittämällä sitä robotin keskiakselin ympäri peukalon tukien avulla (kuva 6). Etulevyn asentamiseen ja pitämiseen käytetään neljää yläkannen kohdistustappia.

Kun 3-levyirrotetaan, asetetaan peukalo käänntökielekkeelle.

Etulevyä työnnetään ja käännetään vastapäivään kuvan osoittamalla tavalla.

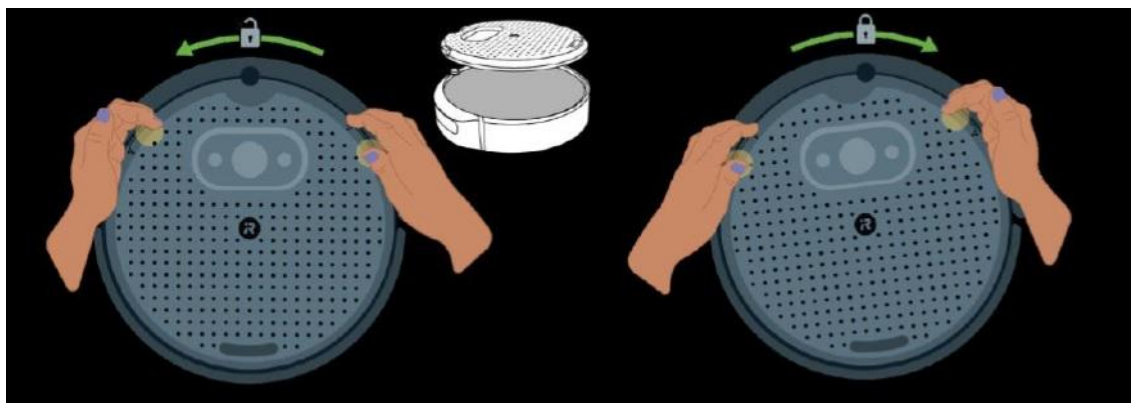
Asetetaan etulevy kevyesti paikalleen käännettynä noin 5 astetta keskustasta kuvan mukaisesti.

Asetetaan peukalo käänntökielekkeille ja painetaan käännettäessä myötäpäivään.

Nyt pitäisi tuntee etulevyn napsahtavan paikalleen keskitettynä.

Jos on vaikeuksia irrottaa etulevy, ota avuksi tussinpää, jolla voidaan napauttaa etulevyä. On mahdollista, että se on laskeutunut lähetyksen aikana.

Jos näin on, käytetään kumivasaraa tai vastaavaa esinettä napauttamalla peukalon kielekkeitä löysätäkseen.



Kuva 6. Etulevyn irrottaminen

Avasin lokeron ja irrotin kaapelit. Etulevy poistettiin pyörittämällä sitä robotin keskiakselin ympäri peukalon tukien avulla (kuva 7). Etulevyn asentamiseen ja pitämiseen käytettiin neljää yläkannen kohdistustappia.



Kuva 7. Etulevy poistettiin

3.1 Robotin etuosa

Robotin etupuolella on monivyöhykkeinen törmäysanturi, joka sisältää seitsemän paria IR-infrapunatutka-antureita. Nämä anturit mahdollistavat esteiden havaitsemisen.

Robotin yläosassa on kolme painiketta, jotka voidaan kaikki mukauttaa ROS 2 -sovelluksen avulla. iRobot Coding -sovelluksessa voi mukauttaa painikkeita. Virtapainike sisältää ympyrän, jossa on kuusi RGB-LED-valoa ilmaisua varten (kuva 8).



Kuva 8. Robotin etuosa

Robotin etulevy ja kuormatila sisältävät säännöllisen reikäkuviointimallin kuormien kiinnittämiseksi. Ne voidaan irrottaa ilman työkaluja nopeaa prototyypausta varten. Siinä on kaksi kaapelien läpiviennin kohtaa: yksi kuormalokeron yläreunassa, mikä on kätevää nopeaa prototyypausta varten.

Toinen, joka kulkee läpi yläkannen ja etulevyn, auttaa pitämään johdot robotin toiminta-alueella. Kun etulevy on poistettu, näkyvillä on myös sovitinlevy, jota käytetään liitettäessä ulkoisiin tietokoneisiin joko Bluetooth1:n tai USB-C 2:n kautta (kuva 9 ja 10).



Kuva 9. Laitteen osien paikat.



Kuva 10. Laitteen osat purettuna.

Kuva 11 esittää Robotin alapuolella komponentit, jotka ovat jyrkänteiden neljä tunnistinta ja pitävät robotin vakaalla kiinteällä alustalla. Etutukipyörä on oletusarvoisesti robotin painopiste.

Keskiakselin etupuolella on latauskoskettimet, kaksi pyörää, joissa on virta-anturit ja kooderit sekä optinen matkanmittausanturi. Robotin IMU ei näy ulospäin, mutta sitä käytetään yhdessä optisen matkanmittausanturin ja pyöräanturien kanssa sulautetun matkanmittausarvion luomiseen.

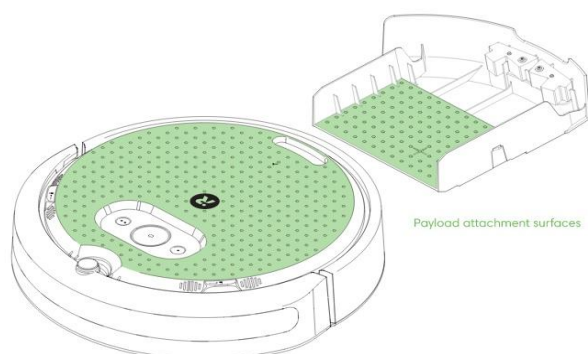


Kuva 11. Robotin pohjakomponentin nimet

3.2 Robotin

Create3 on differentiaalivetoisesti toimiva robotti, joka on suunniteltu liikkumaan suhteellisen tasaisissa sisätiloissa (kuva 12). Ajopyörät on varustettu itsenäisillä jousituksilla, jotka auttavat robottia selviytymään tavallisista matoista tai ovien kynnyksistä.

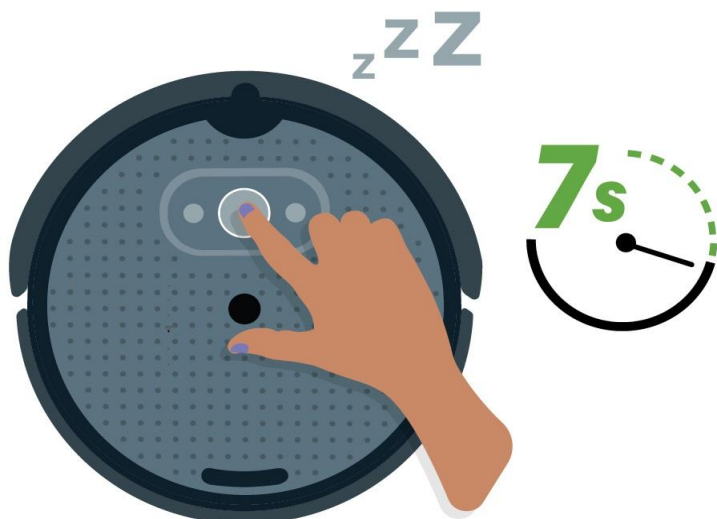
Antureita tai lisävarusteita voidaan kiinnittää etulevyyn tai sisäiseen kuormalokeroon. Molemmissa on säännöllinen 12 mm:n ruudukko, jossa on 3 mm:n halkaisijaltaan olevat kiinnitysreiät. Suurin suositeltu kuormapaino (ilman kiihdytysrajoitusten muuttamista) on 9 kg suoraan rungon painopisteen yläpuolella. Robotti voi kantaa enemmän painoa pienentämällä kiihdytysrajoituksia tai huolellisesti hallinnoimalla kuormaa.



Kuva 12. Etulevyn ja kuormalokero

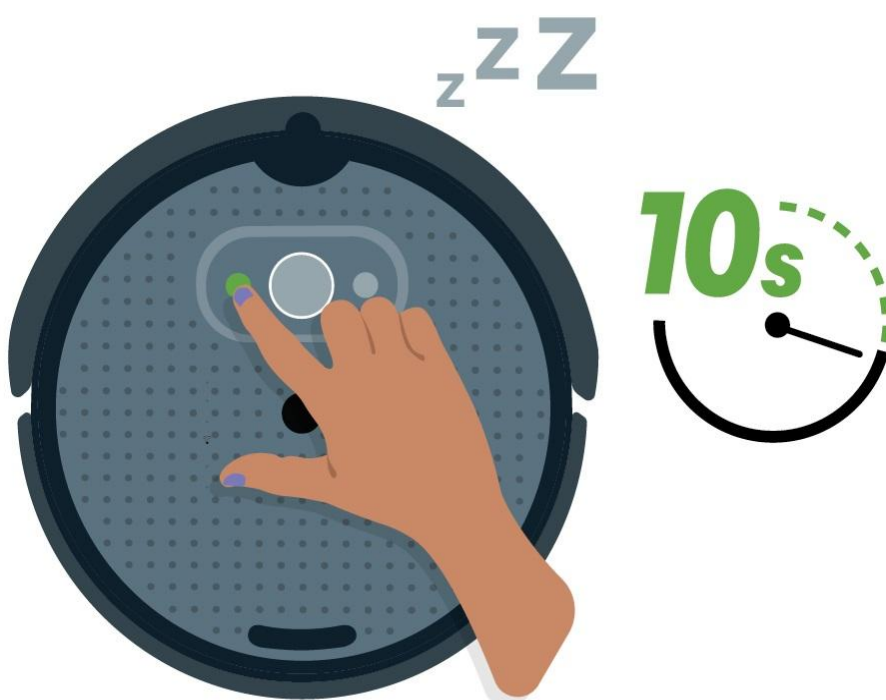
4 Virta pois

Sammutetaan robotti ja painetaan sekä pidetään keskipainiketta seitsemän (7) sekunnin ajan. Sen valorengas välähtää kirkkaan valkoisena kolme (3) kertaa ja toistaa sitten virrankatkaisun sammutusäänen. Valorenkaan pitäisi sammua 10 sekunnin kuluttua (kuva 13).



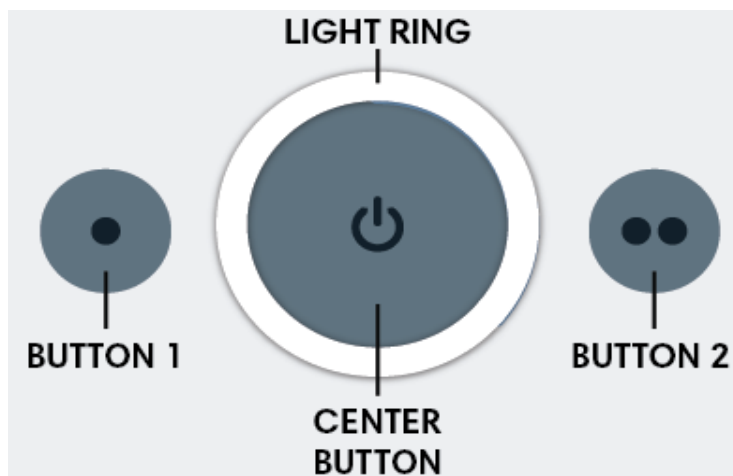
Kuva 13. Kone sammuu

Virransäätötila: Robotti laitetaan vähävirtaiselle tilalle, painetaan ja pidetään painettuna painiketta 1 kymmenen (10) sekunnin ajan (kuva 14). Kun robotti on vähävirtaisessa tilassa, sen kuormitusvirta pysyy päällä ja se latautuu. Se ei reagoi Wi-Fiin, USB:iin tai Bluetoothiin. Poistuttaessa vähävirtaisesta tilasta painetaan keskipainiketta ja pidetään sitä painettuna yhden (1) sekunnin ajan.



Kuva 14. Painike 1 valmiustila

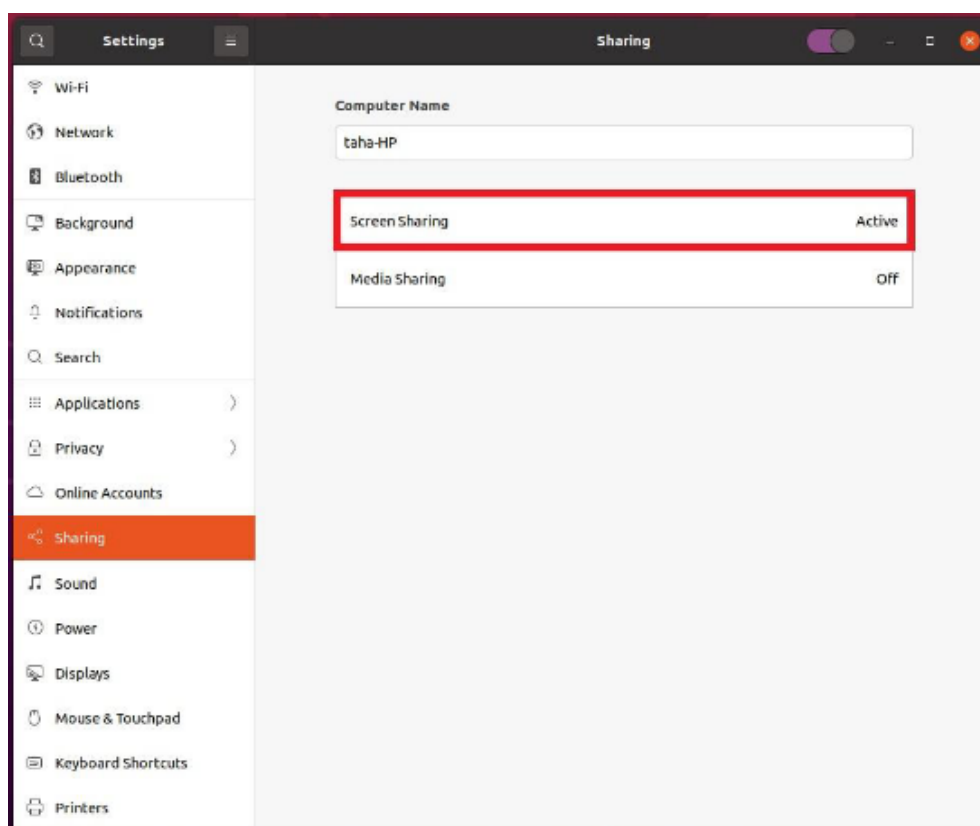
Painike 2 (keskeltä oikealle) on merkitty kahdella pisteellä (**). Se käynnistyy painamalla painiketta sekä ROS 2 -tilassa että iRobot Coding / Bluetooth -tilassa (kuva 15).



Kuva 15. Painike 2

4.1 Wi-Fi-yhteyspisteen luominen

Näytön jakaminen: Pitäisi ottaa käyttöön jaettavan näytön jakaminen. Kun jakaminen on aktivoitu, se voidaan langattomasti liittää tietokoneen muihin laitteisiin (kuva 16).



Kuva 16. Näytön jakaminen

SSH (Secure Shell) on protokolla, joka mahdollistaa tietokoneen etähallinnan ja tie-

dostojen siirron turvallisesti verkon yli. Se on erittäin tärkeä työkalu monille tietokoneiden käyttäjille, erityisesti järjestelmänvalvojille.

SSH:n avulla käyttäjät voivat muodostaa salatun yhteyden tietokoneeseen, joka mahdollistaa tietokoneen hallinnan ja komentosarjojen suorittamisen. SSH:n ansiosta tiedonsiirto verkon yli on lisäksi salattu ja turvallinen. Tämä tekee SSH:sta erittäin tärkeän työkalun tietoturvallisuuden kannalta, sillä se estää tietojen pääsyn väärin käsiin tai tietojen häviämisen matkalla verkon yli.

SSH:n käytön aloittaminen edellyttää SSH-palvelimen asentamista tietokoneelle ja SSH-asiakasohjelman käynnistämistä etäyhteyden muodostamiseksi. Kun SSH-asiakasohjelma on käynnissä, käyttäjä voi kirjautua palvelimelle antamalla käyttäjätunnuksensa ja salasanansa.

Tämän jälkeen tietokonetta voi hallita ja käyttää tarvittavia komentoja.

SSH on erittäin yleinen ja hyödyllinen työkalu, kun halutaan käyttää esimerkiksi verkon yli toimivia palvelimia, kuten web-palvelimia, tietokantapalvelimia tai verkkolaitteita. SSH-protokollalla voi myös siirtää tiedostoja tietokoneen ja verkon muiden solmujen välillä. Se tekee tiedostojen jakamisesta ja siirrosta erittäin helppoa ja turvallista.

Päivitys ja SSH asennus

Komennolla "sudo apt update" päivitetään tietokoneen pakettivarastoja, jolloin saatavilla olevat ohjelmat ja ohjelmistopäivitykset ovat ajan tasalla.

Komento "sudo apt install openssh-server" asentaa tietokoneelle SSH-palvelimen, mikä mahdollistaa etäyhteyden muodostamisen tietokoneeseen ja tiedostojen siirron turvallisesti verkon yli (kuva 17).

```

taha@taha-HP:~$ sudo apt update
[sudo] password for taha:
Hit:1 http://fi.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://fi.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://fi.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
taha@taha-HP:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 libfprint-2-tod1 libfwupdplugin1 liblvm9 libxmlb1
  ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 688 kB of archives.
After this operation, 6 018 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://fi.archive.ubuntu.com/ubuntu focal-updates/main amd64 ncurses-term all 6.2-0ubuntu2.1 [249 kB]
Get:2 http://fi.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-sftp-server amd64 1:8.2p1-4ubuntu0.9 [51,7 kB]
Get:3 http://fi.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-server amd64 1:8.2p1-4ubuntu0.9 [377 kB]
Get:4 http://fi.archive.ubuntu.com/ubuntu focal/main amd64 ssh-import-id all 5.10-0ubuntu1 [10,0 kB]
Fetched 688 kB in 0s (1 837 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ncurses-term.
(Reading database ... 181746 files and directories currently installed.)
Preparing to unpack .../ncurses-term_6.2-0ubuntu2.1_all.deb ...
Unpacking ncurses-term (6.2-0ubuntu2.1) ...
Selecting previously unselected package openssh-sftp-server.
Preparing to unpack .../openssh-sftp-server_1%3a8.2p1-4ubuntu0.9_amd64.deb ...
Unpacking openssh-sftp-server (1:8.2p1-4ubuntu0.9) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.2p1-4ubuntu0.9_amd64.deb ...
Unpacking openssh-server (1:8.2p1-4ubuntu0.9) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../ssh-import-id_5.10-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.10-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.2p1-4ubuntu0.9) ...
Setting up openssh-server (1:8.2p1-4ubuntu0.9) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:c4NIcR22W70AyyLPNzbemepetKSeoGawdbox9Z/RmIw root@taha-HP (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:ehB7znISoeVla0dxaDlPiU4+kcodY0iWl3XoIdJt4q root@taha-HP (ECDSA)

```

Kuva 17. ssh-asennus

Tarkistetaan SSH-käyttöä ja porttia 22: Komennolla "sudo systemctl status SSH" tarkastellaan SSH-palvelimen tilaa System-palvelunohjauksen avulla. Tämä komento kertoo palvelimen nykyisen tilan ja mahdolliset virheviestit. Jos SSH-palvelin on käynnissä, tulosteessa näkyy tieto "active". Jos palvelin ei ole käynnissä, tulosteessa näytetään "inactive" ja viestitään mahdolliset virheet tai hälytykset.

Komennolla "sudo systemctl status SSH" voidaan myös tarkastella SSH-palvelun tiedot, kuten sen käyttäjätiliä. Siitä ohjattuja tiedostoja tai kansiota käytetään SSH-asetuksissa. Tieto siitä, milloin SSH-palvelin on viimeksi käsitellyt olemassa olevaa pyyntöä sekä muita palvelun hallintaan liittyviä tietoja. Kun käsitellään palvelinympäristöä, on suositeltavaa käyttää systemctl-komentoa kun seurataan palvelinten tilaa ja sen toimivuutta.

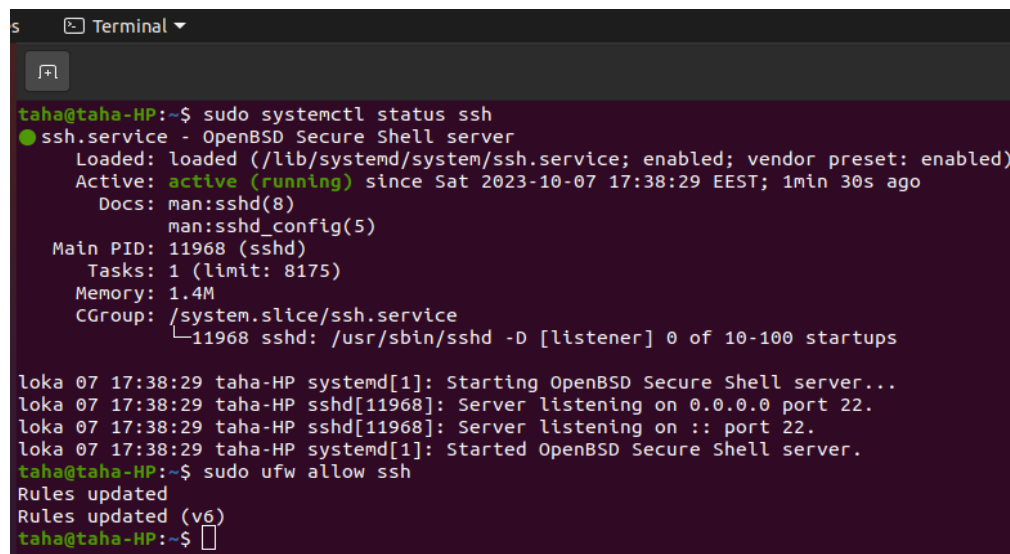
Komennolla "sudo ufw allow SSH" annetaan lupa SSH-yhteyksille Ubuntu Linux – palvelimessa käyttäen yksinkertaista palomuuria (UFW).

SSH turvallinen kuori on protokolla, jota käytetään etäyhteyden luomiseen tietoko-

neeseen tai palvelimeen turvallisesti. Komennolla "sudo ufw allow SSH" otetaan siis yksinkertaista palomuurista käyttöön käyttäjän SSH-yhteydet, jolloin hän voi muodostaa SSH-yhteyksiä Ubuntu-palvelimeen.

UFW mahdollistaa näiden yhteyksien valvomisen ja rajoittamisen tarvittaessa.

Avattiin Terminal ja käytettiin komentoja "sudo systemctl status SSH" ja "sudo ufw allow SSH" sen varmistamiseksi, että SSH ja portti 22 toimivat onnistuneesti kuvassa 18 nähdään activen toiminta SSH-palvelussa.



```
taha@taha-HP:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-10-07 17:38:29 EEST; 1min 30s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 11968 (sshd)
    Tasks: 1 (limit: 8175)
   Memory: 1.4M
    CGroup: /system.slice/ssh.service
            └─11968 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

loka 07 17:38:29 taha-HP systemd[1]: Starting OpenBSD Secure Shell server...
loka 07 17:38:29 taha-HP sshd[11968]: Server listening on 0.0.0.0 port 22.
loka 07 17:38:29 taha-HP sshd[11968]: Server listening on :: port 22.
loka 07 17:38:29 taha-HP systemd[1]: Started OpenBSD Secure Shell server.
taha@taha-HP:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
taha@taha-HP:~$
```

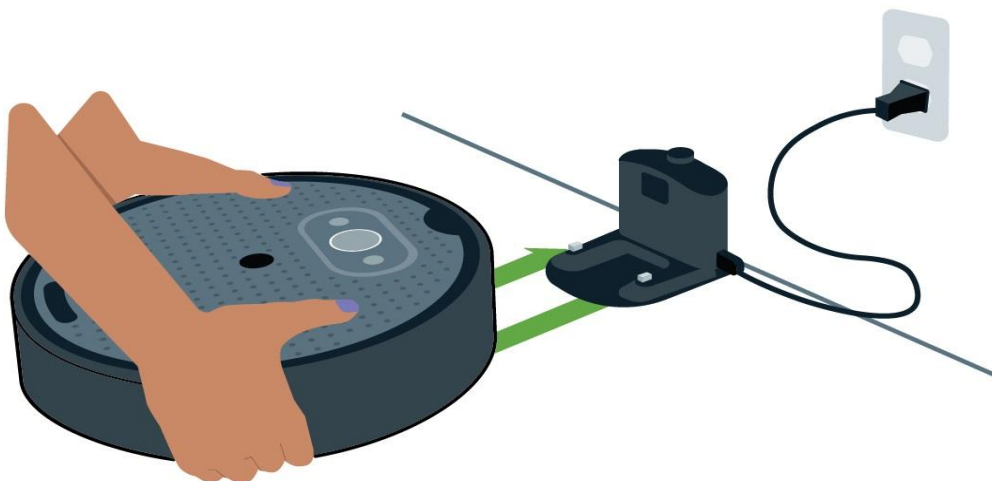
Kuva 18. SSH-palvelu toimii

4.2 Koneen virta päälle

Aluksi liiityttiin telakointiasema, joka yhdistettiin sähköverkkoon. Kytettiin robotti päälle asettamalla se lataustelakkaan. Telakan vihreä LED-valo syttyi vihreänä osoittaen onnistuneen yhteyden muodostumisen.

Robotin valorengas loisti kirkkaalla pyörivällä valolla. Valorengaan pyörimistä jatkettiin ja robotti käynnistettiin. Robotin käynnistysprosessi on valmis, siitä kuultiin äänimerkki, joka kertoi sen valmiudesta.

Määriteltiin Internet-verkko. Sen jälkeen sekä Raspberry Pi että Create3 WiFi-verkko alkoi toimimaan. Valorengaan pyörimisliike hidastuu, se muuttuu hitaasti pyöriväksi valkoiseksi valoksi. Robotin ollessa edelleen lataustelakassa valorengasta tuli tasainen valkoinen valo. Valorengaan valo muuttui tasaiseksi valkoiseksi valoksi. Se poistettiin lataustelakasta (kuva 19).



Kuva 19. Koneen virta päälle

Verkon määrittäminen: Tärkein ensimmäinen toimenpide on verkkoasetukset ja yhdistäminen sekä RaspberryPi:n että Create3:n Wi-Fi-verkkoihin (kuva 20). Tämä tehtiin avaamalla dotbashrc-tiedosto etätietokoneen komentokehotteessa. Kirjoitettiin ensin komento "nano ~/.bashrc"

Terminaalissa aukaistiin bashrc-tiedosto. bashrc -tiedosto avattiin. Sen jälkeen lisättiin koodin viimeiseen riviin "export CYCLONEDDS_URI='<Cyclone DDS><Domain><General><DontRoute>true</></></></>'" bashrc -tiedostonloppuun. Tämän jälkeen tallennettiin painamalla Ctrl + s ja poistutaan tiedostosta painamalla Ctrl + x.

```

# nano: /root/.bashrc
# enable color support of ls and also use handy aliases
if [ -e /usr/bin/dircolors ]; then
    case "$OSTYPE" in s390* | sysv*) alias ls='ls --color=none';; *) alias ls='ls --color=auto';; esac
    alias du='du --color=auto'
    alias vdu='vdu --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'

# colored GCC warnings and errors
export GCC_COLORS='error=01;31:warn=01;35:note=01;36:caret=01;32:locus=01;quote=01'

# some more to aliases
alias ll='ls -lF'
alias la='ls -A'
alias l='ls -CF'

# add an "alert" alias for long running commands. Use like so:
# sleep 30; alert
alias alert='notify-send --urgency=low -t "${SHELL}" "Alert: $(history | tail -n1 | sed -e "s/^ *//")"'

# Alias definitions
# You may want to put all your aliases into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile)
source /etc/bash.bashrc

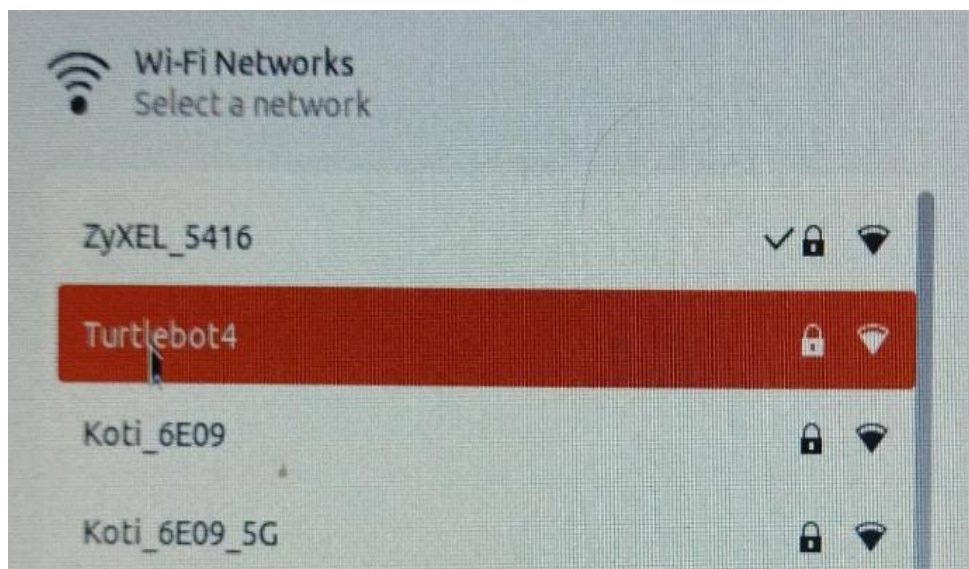
if [ -d /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
else
    . /etc/bash_completion
fi

export CYCLONEDDS_URI='<Cyclone DDS><Domain><General><DontRoute>true</></></></>'

```

Kuva 20. bashrc-tiedosto avattu

Raspberry-verkon määrittäminen: Aloitettiin Raspberry Pi:stä, avattiin etätietokoneen asetukset ja yhdistettiin se Turtlebot4 Wi-Fi-verkkoon, jonka salasana on "turtlebot4" (kuva 21).



Kuva 21. Turtlebot4-verkko

Testetaan ssh [ubuntu@10.42.0.1](https://10.42.0.1). Etätietokoneen on muodostanut yhteyden Turtle-Bot4-verkkoon. Siirryttiin Raspberry Pi:n WiFi-konfigurointiin. Avattiin etätietokoneen terminaali ja kirjoitettiin seuraava komento: `ssh ubuntu@10.42.0.1`
Komento toimi kuvan 22 mukaisesti.

```

es/ Terminal
[~]
teha@Taha-HP: $ ssh ubuntu@10.42.0.1
ubuntu@10.42.0.1's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-1069-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Apr 21 12:58:10 UTC 2022

System load:  1.55          Processes:            162
Usage of /:   18.9% of 28.94GB  Users logged in:     0
Memory usage: 10%          IPv4 address for usb0: 192.168.186.3
Swap usage:   0%           IPv4 address for wlan0: 10.42.0.1
Temperature: 52.6 C

10 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr 21 13:40:26 2022 from 10.42.0.74

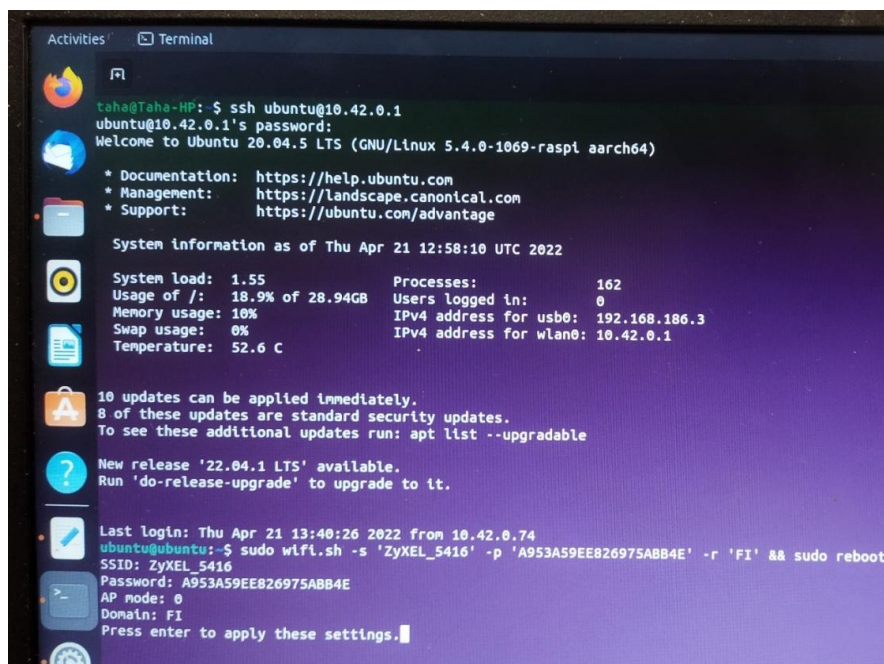
```

Kuva 22. Turtlebot4-verkkotoiminot

Lisätään oman verkon tiedot: Siirryttiin SSH Raspberry Pi:lle. Seuraavaksi määrättiin WIFI-verkko Raspberry Pi -laitteeseen. Raspberry Pi kytkettiin verkkoon. Syötin verkkotietoni ja anoin verkon nimen ja salasanan kuvan 23 mukaisesti.
Parhaan suorituskyvyn saavuttamiseksi yhdistin Raspberry Pi 5Ghz:n WIFI- verk-

koon. SSH-istunnossa kirjoitettiin seuraava komento:

```
sudo wifi.sh -s '<WIFI_SSID>' -p '<WIFI_PASSWORD>' -r <REGULATORY_DOMAIN>&&sudo reboot
```



```

taha@Taha-HP: ~$ ssh ubuntu@10.42.0.1
ubuntu@10.42.0.1's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-1069-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Apr 21 12:58:10 UTC 2022

System load:  1.55          Processes:           162
Usage of /:   18.9% of 28.94GB    Users logged in:    0
Memory usage: 10%           IPv4 address for usb0: 192.168.186.3
Swap usage:  0%             IPv4 address for wlan0: 10.42.0.1
Temperature: 52.6 C

10 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

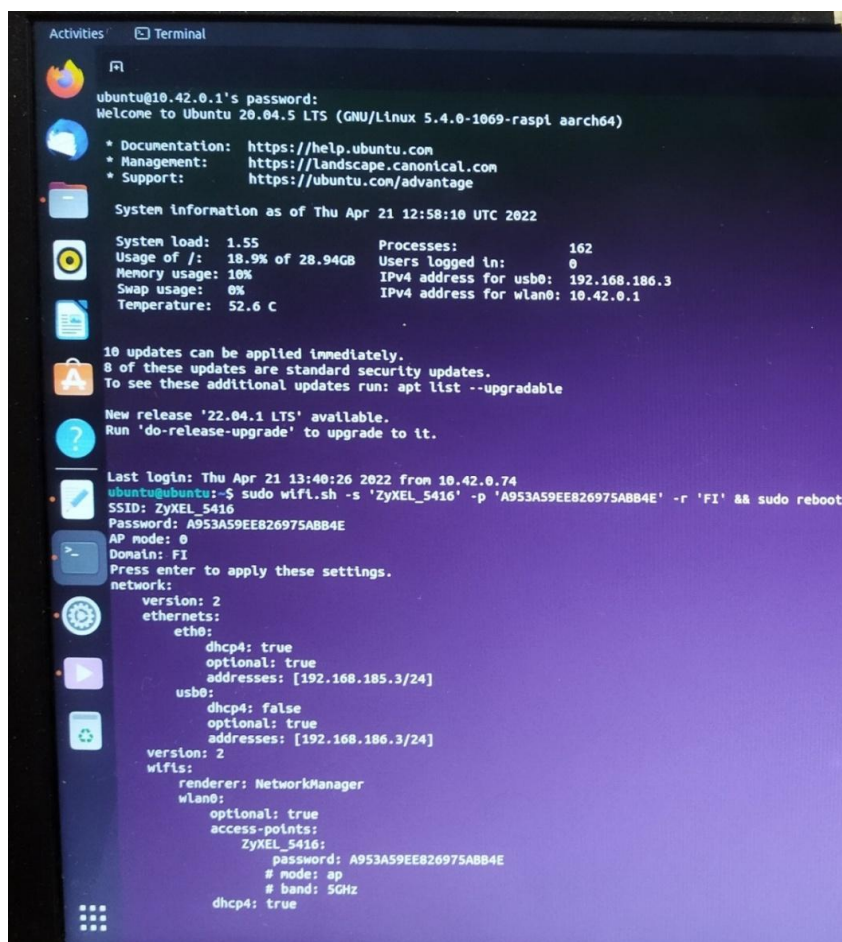
New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr 21 13:40:26 2022 from 10.42.0.74
ubuntu@ubuntu:~$ sudo wifi.sh -s 'ZyXEL_5416' -p 'A953A59EE826975ABB4E' -r 'FI' && sudo reboot
SSID: ZyXEL_5416
Password: A953A59EE826975ABB4E
AP mode: 0
Domain: FI
Press enter to apply these settings.

```

Kuva 23. Verkon määrittäminen

Verkkotiedot: Olen painanut enteriä, ja se on antanut kaikki tiedot verkosta ja asetuksista, kuten on esitetty kuvassa 24.



```

ubuntu@10.42.0.1's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-1069-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Apr 21 12:58:10 UTC 2022

System load:  1.55          Processes:           162
Usage of /:   18.9% of 28.94GB    Users logged in:    0
Memory usage: 10%           IPv4 address for usb0: 192.168.186.3
Swap usage:  0%             IPv4 address for wlan0: 10.42.0.1
Temperature: 52.6 C

10 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr 21 13:40:26 2022 from 10.42.0.74
ubuntu@ubuntu:~$ sudo wifi.sh -s 'ZyXEL_5416' -p 'A953A59EE826975ABB4E' -r 'FI' && sudo reboot
SSID: ZyXEL_5416
Password: A953A59EE826975ABB4E
AP mode: 0
Domain: FI
Press enter to apply these settings.
network:
  version: 2
  ethernet:
    eth0:
      dhcp4: true
      optional: true
      addresses: [192.168.185.3/24]
  usb0:
    dhcp4: false
    optional: true
    addresses: [192.168.186.3/24]
  version: 2
  wifis:
    renderer: NetworkManager
    wlan0:
      optional: true
      access-points:
        ZyXEL_5416:
          Password: A953A59EE826975ABB4E
          # mode: ap
          # band: 5GHz
      dhcp4: true

```

Kuva 24. Oman verkon tiedot ja nopeus

Koneen testaus: Tämän jälkeen suoritettiin robottitestaus. Painettiin nappia kaksi, jolloin robotti liikkui taaksepäin ulos telakasta ja kääntyi sitten 180 astetta. Tämän jälkeen painettiin nappia yksi, jolloin robotti liikkui eteenpäin ja kääntyi sitten 180 astetta.

Robotti etsi ensin telakkaa ympäröivästä ympäristöstään. Sen jälkeen robotti linjasi itsensä telakan suhteen ja ajoi varovasti telakan päälle. Tämän seurauksena telakan valo syttyi vihreäksi, mikä osoittaa, että robotti on oikeassa asennossa ja on valmis toimintaan.

5 Ladataan päivitystä

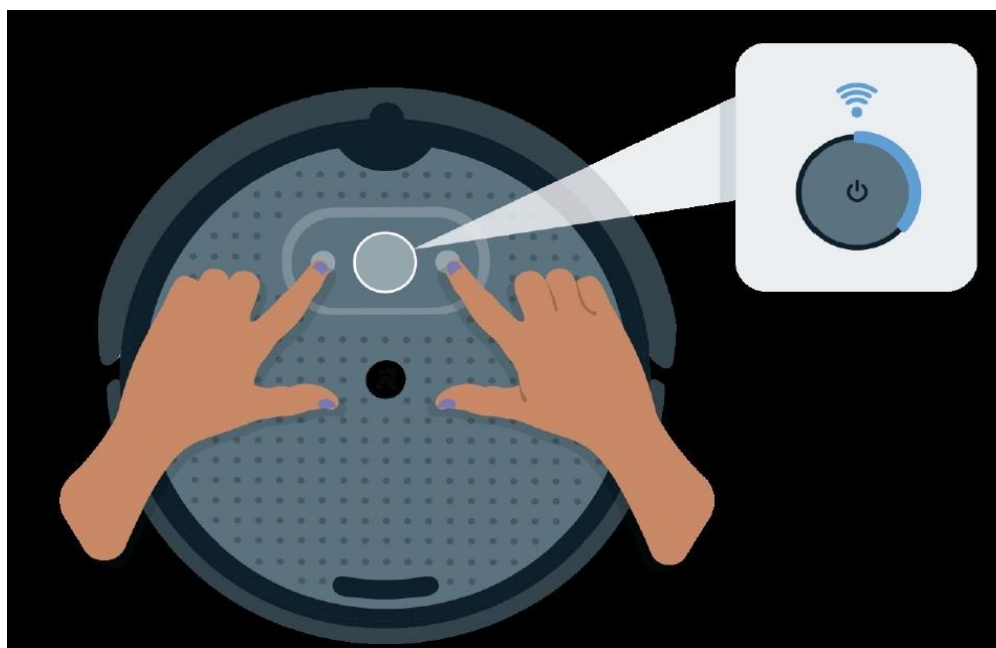
Päivitetään robotin laiteohjelmisto ensin, kun virta on kytketty ja lataus on suoritettu. Ennen kuin aloitetaan robotin koodaus, ladataan uusin laiteohjelmistopaketti osoitteesta <http://edu.irobot.com/create3-latest-fw>.

Ladattiin uuden päivityksen ennen koodin syöttämistä, koska tämä auttoi komentojen syöttämisessä laitteeseen. Kun siirryttiin linkkiin suoraan, se tunnistaa koneen mallin ja lataa uuden version laiteohjelmispäivityksen (firmware update) automaattisesti.

Laitetaan Wi-Fi päälle: Seuraavaksi yhdistetään create3:n WIFI-verkkoon.

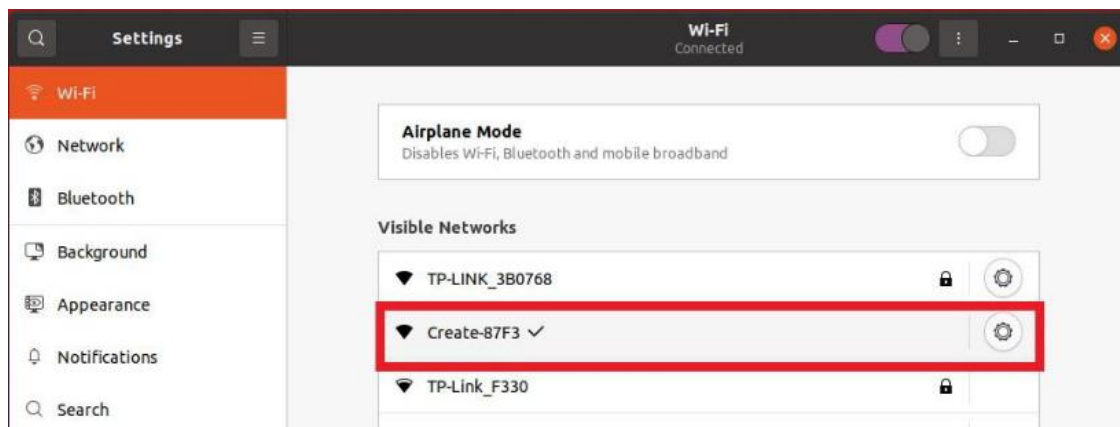
Aktivoidaan robotin Hotspot pitämällä painikkeita 1 ja 2 painettuna samaan aikaan.

Create3:n WIFI onnistuu, jos robotin valorenkaassa näkyy pyörivä sininen valo (kuva 25).



Kuva 25. Verkko päälle

Create3-verkko: Yhdistettiin robotti WIFI-verkkoon painamalla 1 ja 2 painiketta samanaikaisesti, jotta aktivoitiin robotin tukiaseman. Valorengas hehkui sinisenä. Avattiin etätietokoneen WiFi-luettelon napsauttamalla robottiverkkoa muodostaaksesi yhteyden siihen [Create-87F3]. Robottiverkoston nimi on kuvan 26 mukainen.



Kuva 26. Create3 yhdistetty

Nyt etätietokone yhdistettiin robottiin, ja valorengas hehkui tasaisesti syaanin värisenä (kuva 27). Etätietokone irrotettiin Internetistä etätietokoneen ja robotin ollessa kytkettynä.

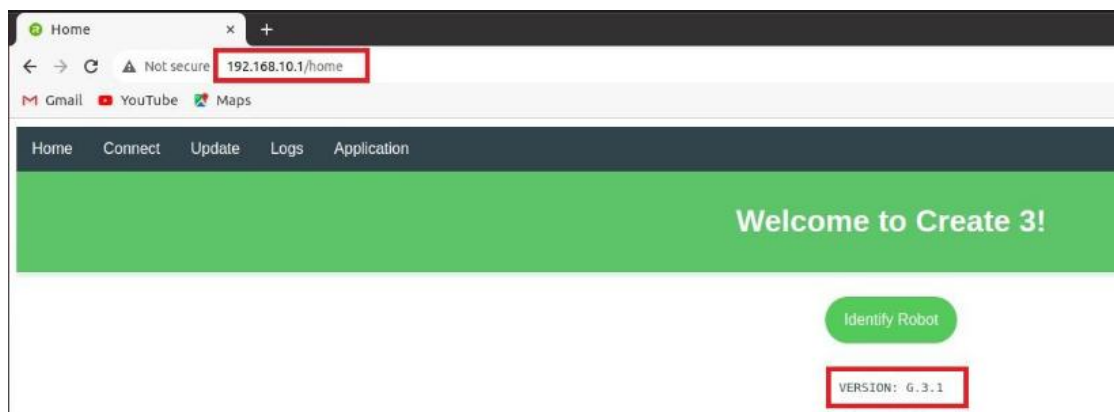


Kuva 27. Verkko onnistunut

Etätietokone yhdistettiin robotin WIFI-verkkoon ja sen jälkeen avattiin uusi internet-selainikkuna. Osoite, johon siirryttiin, oli 192.168.10.1.

Tätä sivua isännöi Create3-robotti, joten sen avaaminen ei vaadi Internet-yhteyttä.

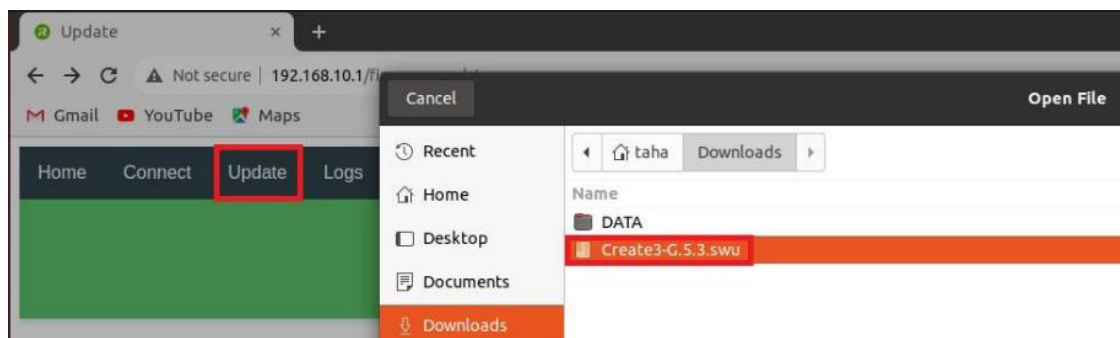
Tällä sivulla nähtiin saatavilla olevan laiteohjelmiston versio, joka on G3.1, kuten kuvassa numero 28. Tältä sivulta pystyttiin päivittämään laitte ja yhdistämään Create3 Wi-Fi-verkkooni.



Kuva 28. Näkyy robotin versio

Siirryttiin vaiheeseen 1 ja ladattiin tiedoston sivustolta <http://edu.irobot.com/create3-latest-fw>

Napsautettiin latausta ja aloitettiin robotin päivitys. Valorengas hehkui sinisenä päivityksen aikana, kuten kuvassa numero 29.

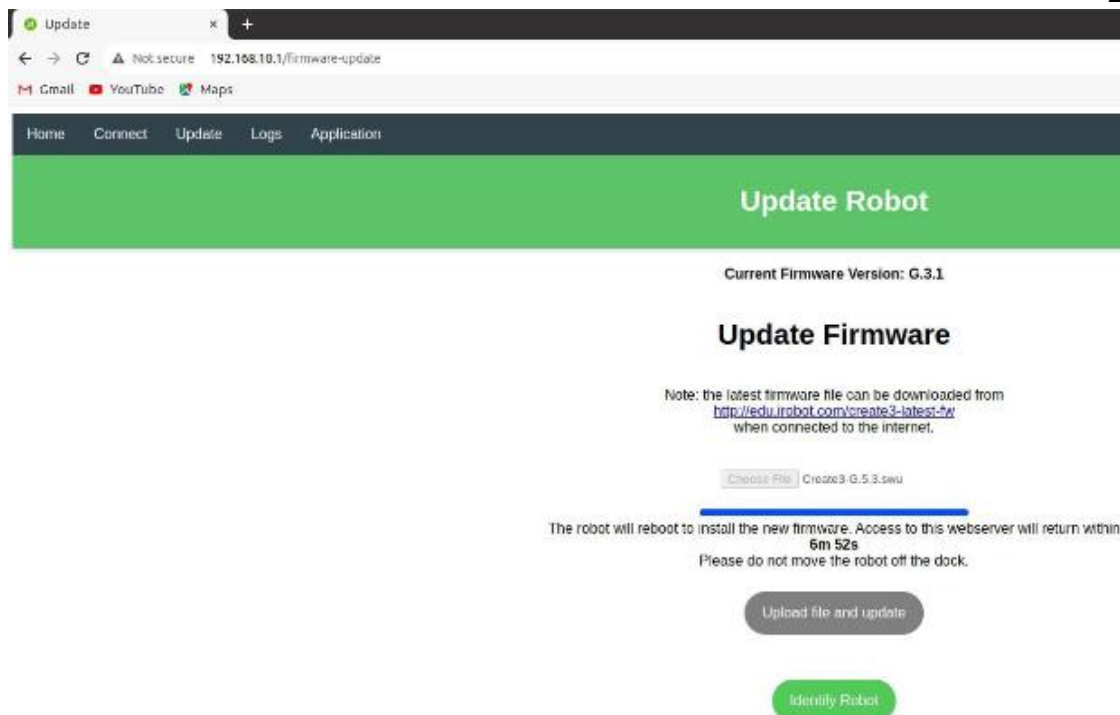


Kuva 29. Uusi versio valittu

Tämä prosessi kesti noin 6 minuuttia, kuten kuvassa numero 30.

Muutaman minuutin kuluttua robotti irrotettiin etätietokoneesta ja alettiin käynnistys uudelleen. Tämän tapahtuessa valorengas muuttui valkoiseksi.

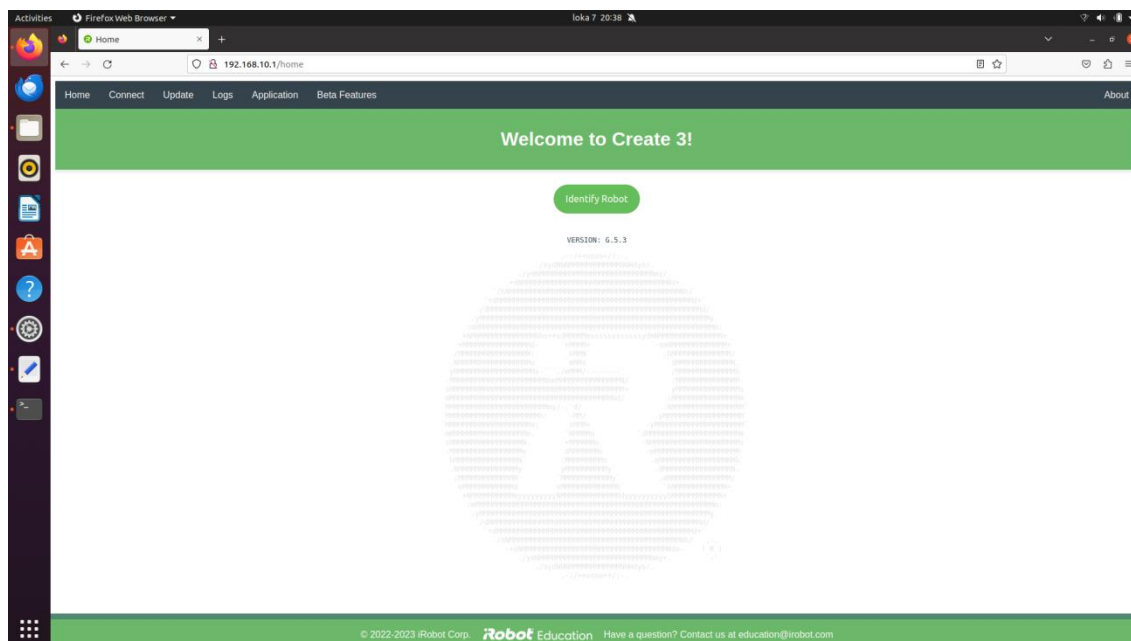
Päivitys on valmis, kun valorengas paloi latausanimaatioon ja robotti antoi merkkiä. Varmistettiin, että etätietokone yhdistettiin oletusarvoiseen Internet-yhteyteen. Kun päivitys on valmis, sen jälkeen aloitettiin koodaus Create3-robotin kanssa.



Kuva 30. Päivitys kestää noin 6 min

Kuva 31 esitetään uusi versio.

Vanha versio oli G 3.1 ja uusi versio G 5.3.



Kuva 31. Uusi versio

5.1 Yhdistetään robotti Wi-Fi-verkkoon

iRobot Create3 mahdollistaa nopean yhteyden Wi-Fi-verkkoon 2,4 GHz.

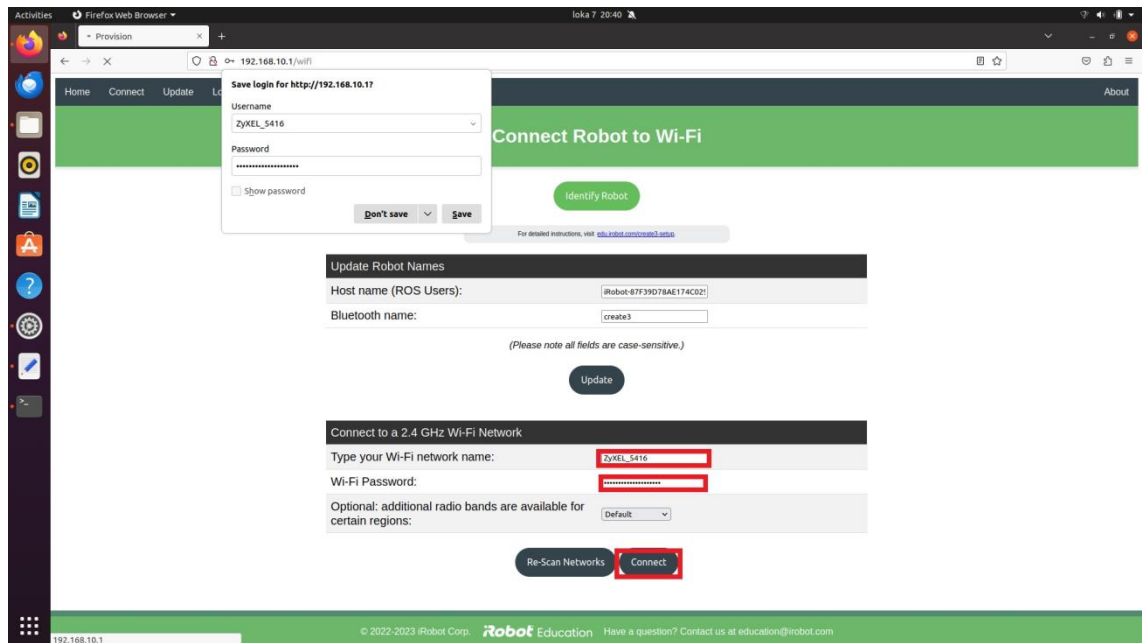
Create3:n verkkopalvelun sivu mahdollistaa robotin nimen muuttamisen ja sitä käytetään myös robotin määrittämiseen 2,4 GHz:n WIFI-verkkoon.

Siirrytään yhdiste-välilehdelle napsauttamalla WIFI-verkon nimen vieressä olevaa kenttää, kun avataan skannattu luettelo käytettävissä olevista WIFI-verkoista. Huomataan, että robotti voi muodostaa yhteyden vain 2,4 GHz:n verkkoon, jos verkon

nimeä ei näy valikossa.

Napsautetaan sivun alareunassa olevaa oranssia skannaa verkot uudelleen – painiketta Seuraavaksi kirjoitetaan WIFI-salasana. Pohjois-Amerikassa tai Japanissa voidaan valita sijainti, kun määritetään muut saatavilla olevat radiotaajuudet.

Määritetään robotille aakkosnumeerinen isäntänimi käytettäväksi myöhemmin koodausprojekteissa. Kun kaikki yllä olevat kentät on täytetty, napsautetaan Yhdistä-painiketta. Valittiin verkko ja syötettiin verkkosalasana kuvan 32 ohjeiden mukaisesti.



Kuva 32. Yhdistää verkon

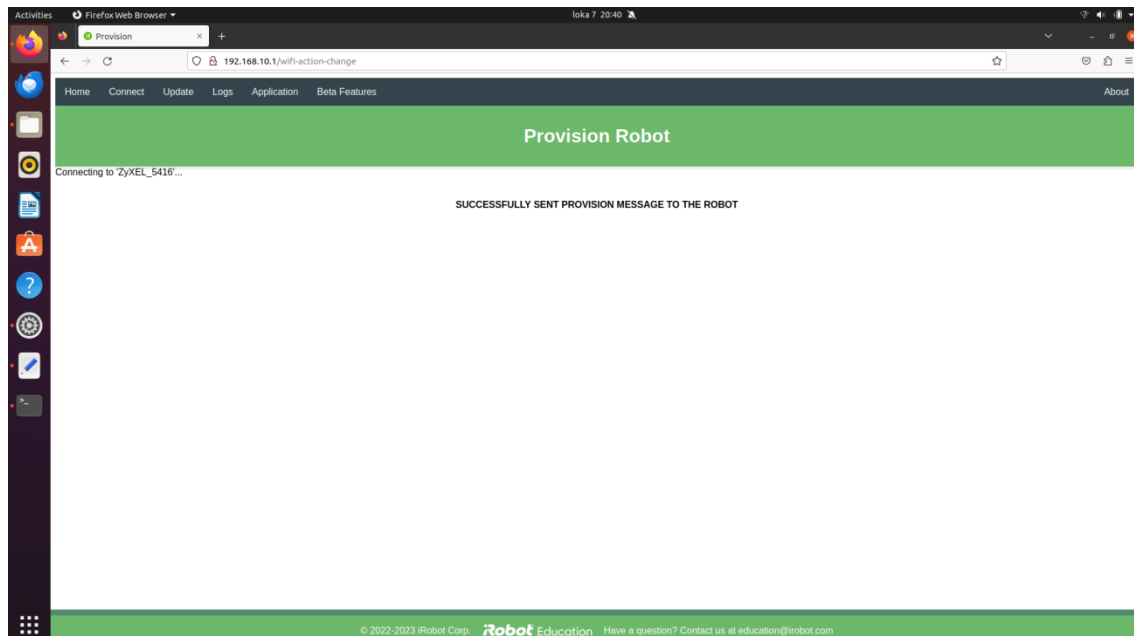
5.2 Wi-Fi yhdistetty

Sivu muutettiin ja vahvistettiin, ja WIFI-tunnus on lähetettiin robotille.

Robotin valorengas paloi sinisenä, kun se yritti muodostaa yhteyden WIFI-verkkoon.

Kun yhteys oli valmis, valorengas muutettiin tasaisen valkoiseksi, ja robotti antoi äänimerkin. Robotti ja verkko yhdistettiin onnistuneesti.

Tämän jälkeen robotti siirtyi omalle verkolle automaattisesti, kuten kuvassa 33.



Kuva 33. Yhdistäminen onnistuu

Mallikone Galactic sopii käyttöjärjestelmään Ubuntu 20.04.

Huomio: Mallikone Humble sopii käyttöjärjestelmään Ubuntu 22.04.

G = Se tarkoittaa Galactic

H = Se tarkoittaa Humble

Asennetaan ROS 2

Etätietokoneella pitää olla asennettu Ubuntu 20.04. Jos sitä ei ole vielä asennettu, lataa ja asenna Ubuntu Server 20.04 64-bittinen versio etätietokoneeseen. Asennettiin ROS 2 Galactic etätietokoneelle noudattamalla näitä ohjeita.

Avattiin terminaali ja kirjoitettiin seuraavia komentoja:

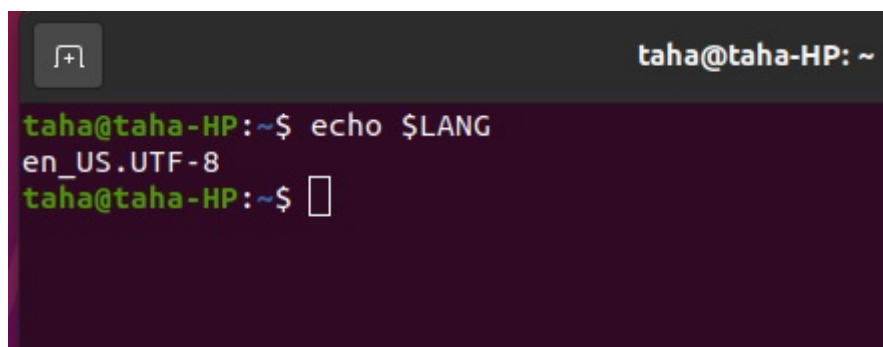
Taulukko 1. Tarvittavat ROS2-paketti

ROS 2 asennus	Galactic on Ubuntu 20.04
UTF-8-kieltä	echo \$LANG
Asentaakseen ROS 2 suorittamallakomento	sudo apt update &&sudo apt install -y curl gnupg2 lsb-release build-essential gitcmake
	sudo curl -ksSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key - o /usr/share/keyrings/ros-archive-keyring.gpg
	echo "deb [arch=\$(dpkg --print-architecture) signed- by=/usr/share/keyrings/ros-archive-keyring.gpg]"

	<code>http://packages.ros.org/ros2/ubuntu \$(lsb_release -cs) main" sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null</code>
	<code>sudo apt update &&sudo apt install -y ros-galactic-ros-base py- thon3-colcon-common-extensions python3-rosdep ros-galactic- rmw-fastrtps-cppros-galactic-rmw-cyclonedds-cppros-galactic- irobot-create-msgs</code>
	<code>echo "source /usr/share/colcon_cd/function/colcon_cd.sh" >> ~/.bashrc</code>
	<code>echo "export _colcon_cd_root=/opt/ros/galactic/" >> ~/.bashrc</code>
	<code>echo "source /usr/share/colcon_argcomplete/hook/colcon- argcomplete.bash" >> ~/.bashrc</code>
	<code>echo "source /opt/ros/galactic/setup.bash" >> ~/.bashrc</code>
	<code>echo "export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp" >> ~/.bashrc</code>
	<code>source ~/.bashrc</code>

Kirjaututtiin sisään, varmistettiin, että käytetään UTF-8-merkistökoodia kirjoittamalla komento (kuva 34).

`echo $LANG`



```
taha@taha-HP: ~$ echo $LANG
en_US.UTF-8
taha@taha-HP: ~$
```

Kuva 34. Käytetään UTF-8-merkistökoodia

6 Aloitetaan ROS2:n testaus

Etätietokone ja robotti ovat samassa verkossa. Ne pitäisi nyt pystyä testaamaan asioita komennolla `ros2 topic list`. Jos tämä ei toimi, tutustutaan ROS 2 -verkon lisätietojen määrittämiseen varten.

Create3-robotti perustuu ROS 2:een, sellaisenaan se paljastaa kaikki käyttäjälle suunnatut API-liittymänsä ROS 2 -olioiden kautta (aiheet, palvelut, toiminnot ja parametrit).

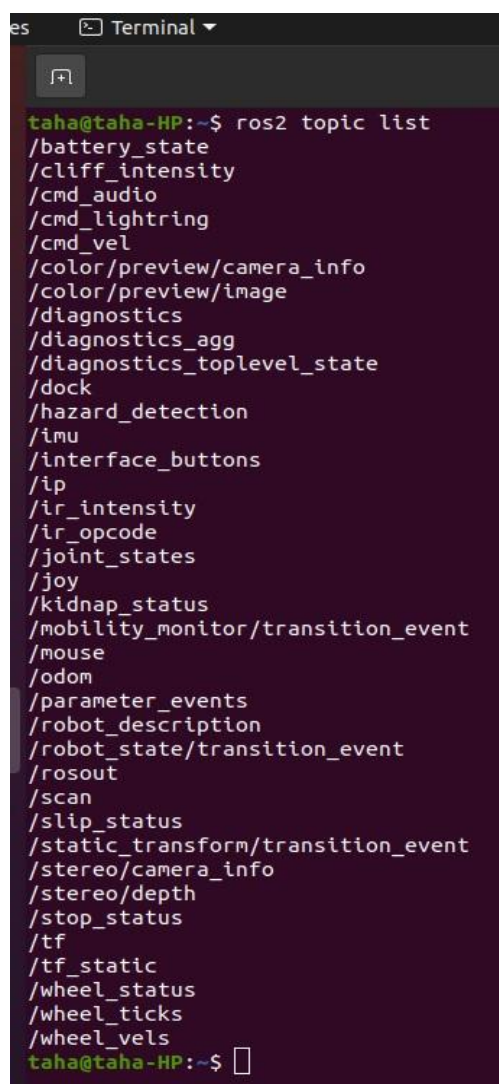
Näiden komentojen tarkoitus on tarjota yleiskuva ROS 2 -sovellusliittymistä.

Robotti käyttää tavallisia ROS 2 -viestejä, kun niitä on saatavilla ja toteuttaa mukautettuja viestejä irobot_create_msgs-tiedostossa tiedoille, joita vakioviestit eivät edusta. Voidaan nähdä ROS 2 -aiheet, jotka Create3-robotti paljastaa Ros2-aiheluettelokomentoa suorittava taulukko näyttää Ros2-testikomennot (taulukko 1).

Taulukko 2. Näyttää Ros2-testikomennot

Verkkotestaus	ROS 2-komento
	ros2 topic list
	ros2 topic list -t
	ros2 service list -t
	ros2 action list -t
	ros2 param list
	ros2 topic echo /tf

ROS 2 -aihelista tarkoittaa kaikkia niitä aiheita, joita käytettiin viestintään ROS 2 -järjestelmän eri osien välillä. käytettiin komentoa "ros2 topic list", joka oletusarvoisesti listaa kaikki käytössä olevat aiheet ROS 2 -järjestelmässä (kuva 35).



```

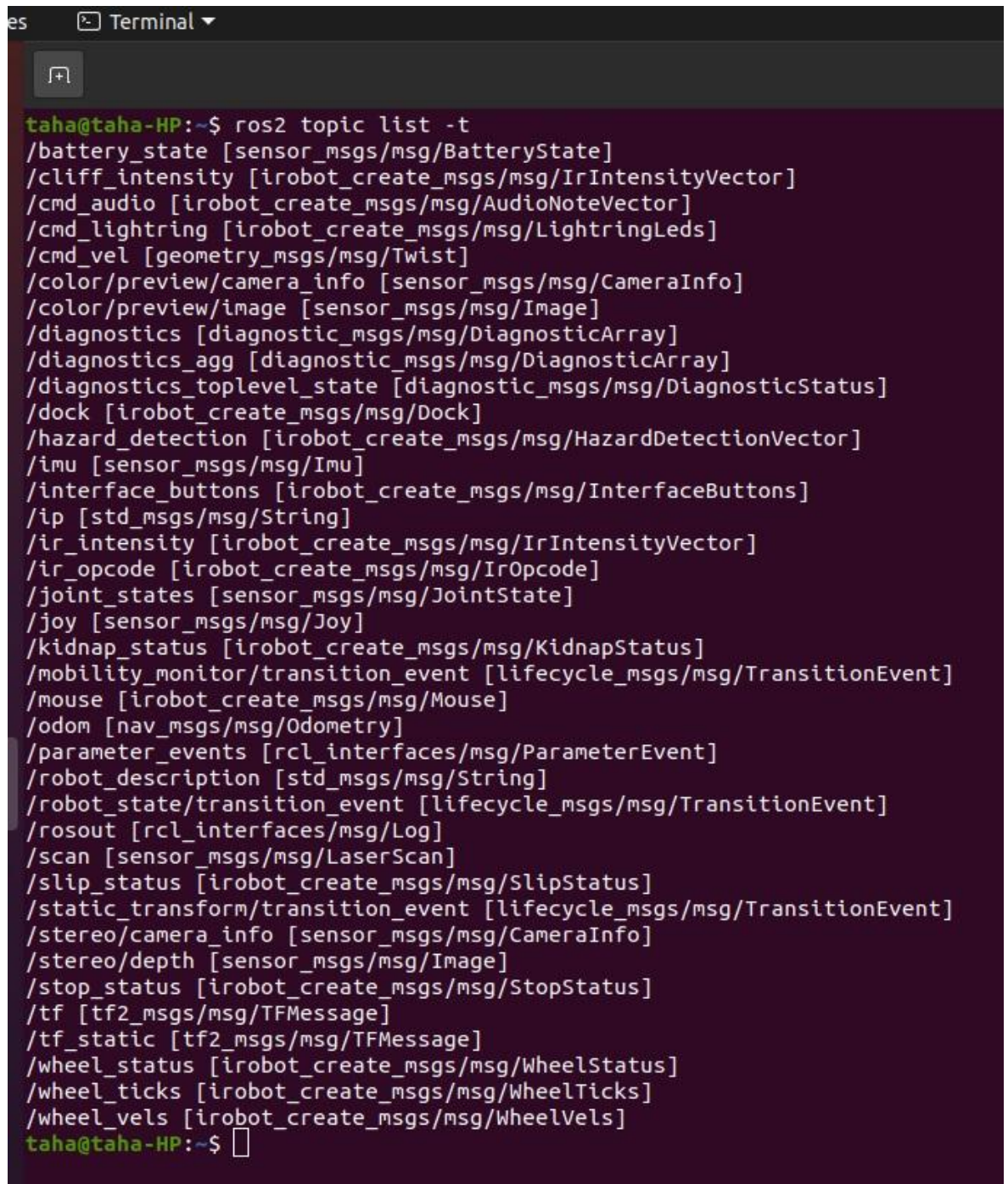
taha@taha-HP:~$ ros2 topic list
/battery_state
/cliff_intensity
/cmd_audio
/cmd_lightring
/cmd_vel
/color/preview/camera_info
/color/preview/image
/diagnostics
/diagnostics_agg
/diagnostics_toplevel_state
/dock
/hazard_detection
/imu
/interface_buttons
/ip
/ir_intensity
/ir_opcode
/joint_states
/joy
/kidnap_status
/mobility_monitor/transition_event
/mouse
/odom
/parameter_events
/robot_description
/robot_state/transition_event
/rosout
/scan
/slip_status
/static_transform/transition_event
/stereo/camera_info
/stereo/depth
/stop_status
/tf
/tf_static
/wheel_status
/wheel_ticks
/wheel_vels
taha@taha-HP:~$

```

Kuva 35. Ros2-aiheluettelo

Create3-robotin altistamat ROS 2 -aiheet ajamalla voi nähdä komennon ros2 topic

list -t. Jos halutaan nähdä vain tietyn aiheen tiedot, voidaan käyttää komentoa "ros2 topic list <aiheen_nimi> -t", joka näyttää yksityiskohtaisemmin kyseisen aiheen tietoja. Näiden aiheiden tietojen avulla voidaan seurata kommunikointia järjestelmän eri moduulien välillä ja tarvittaessa korjata mahdolliset ongelmat (kuva 36).



```

taha@taha-HP:~$ ros2 topic list -t
/battery_state [sensor_msgs/msg/BatteryState]
/cliff_intensity [irobot_create_msgs/msg/IrIntensityVector]
/cmd_audio [irobot_create_msgs/msg/AudioNoteVector]
/cmd_lightring [irobot_create_msgs/msg/LightringLeds]
/cmd_vel [geometry_msgs/msg/Twist]
/color/preview/camera_info [sensor_msgs/msg/CameraInfo]
/color/preview/image [sensor_msgs/msg/Image]
/diagnostics [diagnostic_msgs/msg/DiagnosticArray]
/diagnostics_agg [diagnostic_msgs/msg/DiagnosticArray]
/diagnostics_toplevel_state [diagnostic_msgs/msg/DiagnosticStatus]
/dock [irobot_create_msgs/msg/Dock]
/hazard_detection [irobot_create_msgs/msg/HazardDetectionVector]
/imu [sensor_msgs/msg/Imu]
/interface_buttons [irobot_create_msgs/msg/InterfaceButtons]
/ip [std_msgs/msg/String]
/ir_intensity [irobot_create_msgs/msg/IrIntensityVector]
/ir_opcode [irobot_create_msgs/msg/IrOpcode]
/joint_states [sensor_msgs/msg/JointState]
/joy [sensor_msgs/msg/Joy]
/kidnap_status [irobot_create_msgs/msg/KidnapStatus]
/mobility_monitor/transition_event [lifecycle_msgs/msg/TransitionEvent]
/mouse [irobot_create_msgs/msg/Mouse]
/odom [nav_msgs/msg/Odometry]
/parameter_events [rcl_interfaces/msg/ParameterEvent]
/robot_description [std_msgs/msg/String]
/robot_state/transition_event [lifecycle_msgs/msg/TransitionEvent]
/rosout [rcl_interfaces/msg/Log]
/scan [sensor_msgs/msg/LaserScan]
/slip_status [irobot_create_msgs/msg/SlipStatus]
/static_transform/transition_event [lifecycle_msgs/msg/TransitionEvent]
/stereo/camera_info [sensor_msgs/msg/CameraInfo]
/stereo/depth [sensor_msgs/msg/Image]
/stop_status [irobot_create_msgs/msg/StopStatus]
/tf [tf2_msgs/msg/TFMessage]
/tf_static [tf2_msgs/msg/TFMessage]
/wheel_status [irobot_create_msgs/msg/WheelStatus]
/wheel_ticks [irobot_create_msgs/msg/WheelTicks]
/wheel_vels [irobot_create_msgs/msg/WheelVels]
taha@taha-HP:~$

```

Kuva 36. Ros2-aihelista -t

Create3 -robotin altistamat ROS2 -palvelimet ajamalla voitiin nähdä komennon ros2 service list (kuva 37). ROS2 service list -t puolestaan on komentorivikomento, joka listaa kaikki ROS2-järjestelmässä käytössä olevat palvelut.

Palvelut ovat toinen ROS2:n kommunikointimuoto, jonka avulla voidaan siirtää eri-

laisia tietoja robotin ja sen ympäristön välillä. Palvelujen avulla voi esimerkiksi hakea tietoja robotin sensorilaitteilta, asettaa parametreja tai muuttaa robotin toimintoja.

```

Terminal
taha@taha-HP:~$ ros2 service list -t
/analyzers/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/analyzers/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/analyzers/get_parameters [rcl_interfaces/srv/GetParameters]
/analyzers/list_parameters [rcl_interfaces/srv/ListParameters]
/analyzers/set_parameters [rcl_interfaces/srv/SetParameters]
/analyzers/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/e_stop [irobot_create_msgs/srv/EStop]
/mobility_monitor/change_state [lifecycle_msgs/srv/ChangeState]
/mobility_monitor/get_available_states [lifecycle_msgs/srv/GetAvailableStates]
/mobility_monitor/get_available_transitions [lifecycle_msgs/srv/GetAvailableTransitions]
/mobility_monitor/get_state [lifecycle_msgs/srv/GetState]
/mobility_monitor/get_transition_graph [lifecycle_msgs/srv/GetAvailableTransitions]
/motion_control/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/motion_control/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/motion_control/get_parameters [rcl_interfaces/srv/GetParameters]
/motion_control/list_parameters [rcl_interfaces/srv/ListParameters]
/motion_control/set_parameters [rcl_interfaces/srv/SetParameters]
/motion_control/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/reset_pose [irobot_create_msgs/srv/ResetPose]
/rgb_stereo_node/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/rgb_stereo_node/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/rgb_stereo_node/get_parameters [rcl_interfaces/srv/GetParameters]
/rgb_stereo_node/list_parameters [rcl_interfaces/srv/ListParameters]
/rgb_stereo_node/set_parameters [rcl_interfaces/srv/SetParameters]
/rgb_stereo_node/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/robot_power [irobot_create_msgs/srv/RobotPower]
/robot_state/change_state [lifecycle_msgs/srv/ChangeState]
/robot_state/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/robot_state/get_available_states [lifecycle_msgs/srv/GetAvailableStates]
/robot_state/get_available_transitions [lifecycle_msgs/srv/GetAvailableTransitions]
/robot_state/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/robot_state/get_parameters [rcl_interfaces/srv/GetParameters]
/robot_state/get_state [lifecycle_msgs/srv/GetState]
/robot_state/get_transition_graph [lifecycle_msgs/srv/GetAvailableTransitions]
/robot_state/list_parameters [rcl_interfaces/srv/ListParameters]
/robot_state/set_parameters [rcl_interfaces/srv/SetParameters]
/robot_state/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/robot_state_publisher/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/robot_state_publisher/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/robot_state_publisher/get_parameters [rcl_interfaces/srv/GetParameters]
/robot_state_publisher/list_parameters [rcl_interfaces/srv/ListParameters]
/robot_state_publisher/set_parameters [rcl_interfaces/srv/SetParameters]
/robot_state_publisher/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/rplidar_composition/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/rplidar_composition/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/rplidar_composition/get_parameters [rcl_interfaces/srv/GetParameters]
/rplidar_composition/list_parameters [rcl_interfaces/srv/ListParameters]
/rplidar_composition/set_parameters [rcl_interfaces/srv/SetParameters]
/rplidar_composition/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/set_camera_info [sensor_msgs/srv/SetCameraInfo]
/start_motor [std_srvs/srv/Empty]
/static_transform/change_state [lifecycle_msgs/srv/ChangeState]
/static_transform/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/static_transform/get_available_states [lifecycle_msgs/srv/GetAvailableStates]
/static_transform/get available transitions [lifecycle_msgs/srv/GetAvailableTransitions]

```

Kuva 37. Ros2 service list

ROS2 action list -t on komentorivikomento, joka listaa kaikki ROS2-järjestelmässä käytettävät toimintopalvelut (kuva 38). Toimintopalvelut ovat ROS2:n eräänlaisia kommunikointipalveluita, jotka mahdollistavat robotin suorittamaan monimutkaisempia tehtäviä. Toimintopalveluiden avulla robotin toiminnan ohjaaminen ja seuranta on helppoa ja tehokasta.

```

Terminal
taha@taha-HP:~$ ros2 action list -t
/audio_note_sequence [irobot_create_msgs/action/AudioNoteSequence]
/dock [irobot_create_msgs/action/DockServo]
/drive_arc [irobot_create_msgs/action/DriveArc]
/drive_distance [irobot_create_msgs/action/DriveDistance]
/led_animation [irobot_create_msgs/action/LedAnimation]
/navigate_to_position [irobot_create_msgs/action/NavigateToPosition]
/rotate_angle [irobot_create_msgs/action/RotateAngle]
/undock [irobot_create_msgs/action/Undock]
/wall_follow [irobot_create_msgs/action/WallFollow]
taha@taha-HP:~$

```

Kuva 38. ros2-toimintoluettelo -t

ROS 2 -parametrista tarkoittaa kaikkia niitä muuttujia, joita käytettiin parametrien tallentamiseen ROS 2 -järjestelmässä. Käytettiin komentoa "ros2 param list", joka listaa kaikki käytettävissä olevat parametrit ROS 2 -järjestelmässä. Näiden parametrien avulla voitiin hallita erilaisia järjestelmä- ja robottilogistiikan asetuksia, kuten esimerkiksi alustan nopeus, herkkyys ja tarkkuus (kuva 39).

```

Terminal
taha@taha-HP:~$ ros2 param list
/analyzers:
camera.contains
camera.path
camera.type
create3.analyzers.battery.contains
create3.analyzers.battery.path
create3.analyzers.battery.type
create3.analyzers.docked.contains
create3.analyzers.docked.path
create3.analyzers.docked.type
create3.analyzers.hazards.contains
create3.analyzers.hazards.path
create3.analyzers.hazards.type
create3.analyzers.imu.contains
create3.analyzers.imu.path
create3.analyzers.imu.type
create3.analyzers.mouse.contains
create3.analyzers.mouse.path
create3.analyzers.mouse.type
create3.analyzers.wheels.contains
create3.analyzers.wheels.path
create3.analyzers.wheels.type
create3.path
create3.type
lidar.contains
lidar.path
lidar.type
path
qos_overrides./parameter_events.publisher.depth
qos_overrides./parameter_events.publisher.durability
qos_overrides./parameter_events.publisher.history
qos_overrides./parameter_events.publisher.reliability
use_sim_time
/motion_control:
max_speed
min_speed
qos_overrides./parameter_events.publisher.depth
qos_overrides./parameter_events.publisher.durability
qos_overrides./parameter_events.publisher.history
qos_overrides./parameter_events.publisher.reliability
reflexes.REFLEX_BUMP
reflexes.REFLEX_CLIFF
reflexes.REFLEX_DOCK_AVOID
reflexes.REFLEX_GYRO_CAL
reflexes.REFLEX_PANIC
reflexes.REFLEX_PROXIMITY_SLOWDOWN
reflexes.REFLEX_STUCK
reflexes.REFLEX_VIRTUAL_WALL
reflexes.REFLEX_WHEEL_DROP
reflexes_enabled
safety_override
use_sim_time
wheel_accel_limit
/rgb_stereo_node:

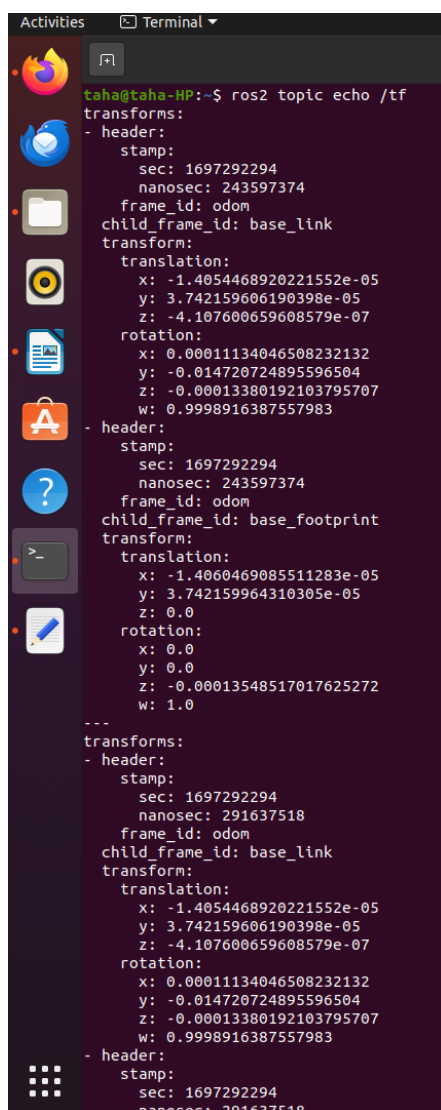
```

Kuva 39. ros2-parametriluettelo

ROS 2 –Koordinaatisto: Create3-robotti tuotettiin fuusioituneen odometrian, joka yhdistettiin sen pyörän kooderin, IMU- laitteen ja maan optisen virtausanturin. Robotin koordinaatisto on oikeakätinen, jossa x on eteenpäin, y vasemmalle ja z ylöspäin. Se paljastaa tämän koordinaatiston sekä tf-puun että /odom-julkaisun kautta.

ROS2 topic echo /tf on komento, jota käytettiin ROS2-ympäristössä kuuntelemaan muutoksia (tf)-viesteissä. Muuntaa-viestit sisältävät tietoa koordinaattijärjestelmien välisistä muunnoksista robotiikan sovelluksissa.

Komento "ros2 topic echo /tf" tulostaa kaikki /tf-topiikkiin julkaistut viestit terminaaliin. Viestien sisältö voi sisältää aikaleiman, kahden koordinaattijärjestelmän välisen muunnoksen sekä muunnoksen matriisimuodossa. Tämä mahdollistaa robotin eri osien ja antureiden paikallistamisen ja seurannan robotin koordinaattijärjestelmään (kuva 40).



```

Activities Terminal
taha@taha-HP:~$ ros2 topic echo /tf
transforms:
- header:
  stamp:
    sec: 1697292294
    nanosec: 243597374
  frame_id: odom
  child_frame_id: base_link
  transform:
    translation:
      x: -1.4054468920221552e-05
      y: 3.742159606190398e-05
      z: -4.107600659608579e-07
    rotation:
      x: 0.00011134046508232132
      y: -0.014720724895596504
      z: -0.00013380192103795707
      w: 0.9998916387557983
- header:
  stamp:
    sec: 1697292294
    nanosec: 243597374
  frame_id: odom
  child_frame_id: base_footprint
  transform:
    translation:
      x: -1.4060469085511283e-05
      y: 3.742159964310305e-05
      z: 0.0
    rotation:
      x: 0.0
      y: 0.0
      z: -0.00013548517017625272
      w: 1.0
---
transforms:
- header:
  stamp:
    sec: 1697292294
    nanosec: 291637518
  frame_id: odom
  child_frame_id: base_link
  transform:
    translation:
      x: -1.4054468920221552e-05
      y: 3.742159606190398e-05
      z: -4.107600659608579e-07
    rotation:
      x: 0.00011134046508232132
      y: -0.014720724895596504
      z: -0.00013380192103795707
      w: 0.9998916387557983
- header:
  stamp:
    sec: 1697292294
    nanosec: 291637518

```

Kuva 40. ROS 2 -Koordinaatisto

6.1 Create3-telakointi

Create3-robotissa on telakointiasema, jolla se ladataan kokeiden välillä (kuva 41). ROS 2 -sovellusliittymien kautta voidaan ohjata itsenäisiä telakointi- ja irrotustoimintoja.

Huomataan, että telakointitoiminnon ja anturin aihe vaihtui Galacticin ja Humblen välillä.

Jotta robotti havaitsee telakan, määrittää sen sijainnin ja ymmärtää, milloin se on telakoitu onnistuneesti. Telakointiaseman on oltava kytkettynä virtalähteeseen.

Telakan irrottaminen (Undocking).

Voidaan käskää robotin irrottamaan telakoinnin seuraavalla ROS 2 -toiminnolla

```
ros2 action send_goal /undock irobot_create_msgs/action/Undock "{}"
```

Robotti liikkuu taaksepäin ulos telakasta ja kääntyy sitten 180 astetta.

Tämä toiminto epäonnistuu, jos robotti on jo irrotettu.

Telakointi (Docking)

Voidaan käskää robotin telakoitumaan ROS 2 -toiminnolla.

```
ros2action send_goal /dock irobot_create_msgs/action/DockServo "{}"
```

Robotti liikkuu eteenpäin ja kääntyy sitten 180 astetta.

Robotti etsii ensin telakan välittömästä ympäristöstään.

Huomaa, että toiminto epäonnistuu, jos robotti on liian kaukana telakasta.

Sitten robotti linjaa telakan ja ajaa varovasti sen päälle.

Toiminto suoritettiin onnistuneesti kuvan 41 osoittamalla tavalla.

```

taha@taha-HP:~$ ros2 action send_goal /undock irobot_create_msgs/action/Undock "{}"
Waiting for an action server to become available...
Sending goal:
{}

Goal accepted with ID: 358affd4ef154d9eab73dbb24384636e

Result:
  is_docked: false

Goal finished with status: SUCCEEDED
taha@taha-HP:~$ ros2 action send_goal /dock irobot_create_msgs/action/DockServo "{}"
Waiting for an action server to become available...
Sending goal:
{}

Goal accepted with ID: 4a0aab34b86448c385c38e9433b710db

Result:
  is_docked: true

Goal finished with status: SUCCEEDED
taha@taha-HP:~$

```

Kuva 41. Poistu ja palauta laite lataustelakkaan

Robottia voi etäohjata esimerkiksi näppäimistöllä tai peliohjaimella, jos tarvittavat Ros-paketti on asennettu, yksinkertaisin tapa saada robotti ajamaan on käyttää etätietokoneen näppäimistösovellusta.

Voidaan asentaa teleop_twist_keyboard-paketin etätietokoneeseen suorittamalla seuraavat komennot:

```
sudo apt update
```

```
sudo apt install ros-galactic-teleop-twist-keyboard
```

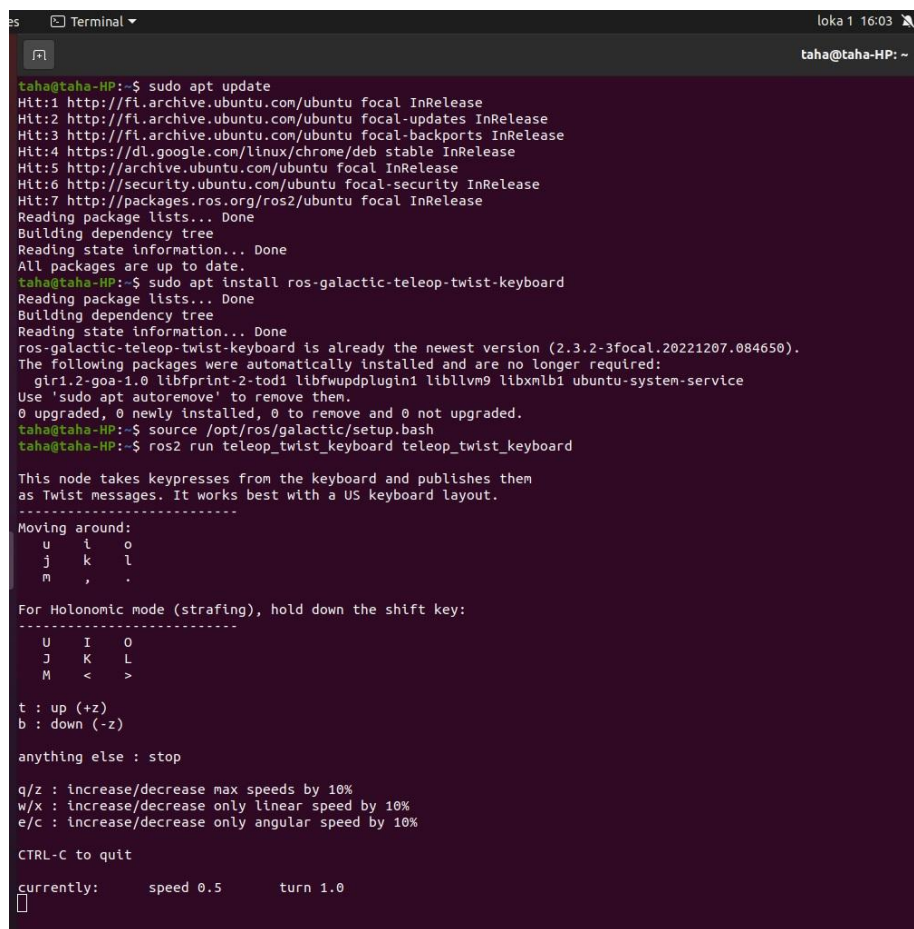
Asennuksen jälkeen suorita solmu soittamalla:

```
source /opt/ros/galactic/setup.bash
```

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

TurtleBotia ohjatakseen täytyy kyseisen komentopäätteikkunan olla aktiivisena.

Aktiiviseksi sen saa esimerkiksi klikkaamalla sitä. Näppäimillä I ja muutetaan lineaarista nopeutta. I ja näppäimillä muutetaan lineaarista nopeutta. J ja L näppäimillä muutetaan kulmanopeutta. Näppäimellä K pysäytetään robotti. Kuvassa 42 on kuva-kaappaus näppäimistöohjauksesta, kun robotilla on vain pyörity paikallaan ja pysäytetty sitä useita kertoja.



```
taha@taha-HP:~$ sudo apt update
Hit:1 http://fl.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://fl.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://fl.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:7 http://packages.ros.org/ros2/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
taha@taha-HP:~$ sudo apt install ros-galactic-teleop-twist-keyboard
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-galactic-teleop-twist-keyboard is already the newest version (2.3.2-3focal.20221207.084650).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 libfprint-2-tod1 libfwupdplugin1 liblvm9 libxmlb ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
taha@taha-HP:~$ source /opt/ros/galactic/setup.bash
taha@taha-HP:~$ ros2 run teleop_twist_keyboard teleop_twist_keyboard

This node takes keypresses from the keyboard and publishes them
as Twist messages. It works best with a US keyboard layout.
-----
Moving around:
  u   l   o
  j   k   l
  m   ,   .

For Holonomic mode (strafing), hold down the shift key:
-----
  U   I   O
  J   K   L
  M   <   >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:   speed 0.5      turn 1.0
□
```

Kuva 42. Näppäimistö

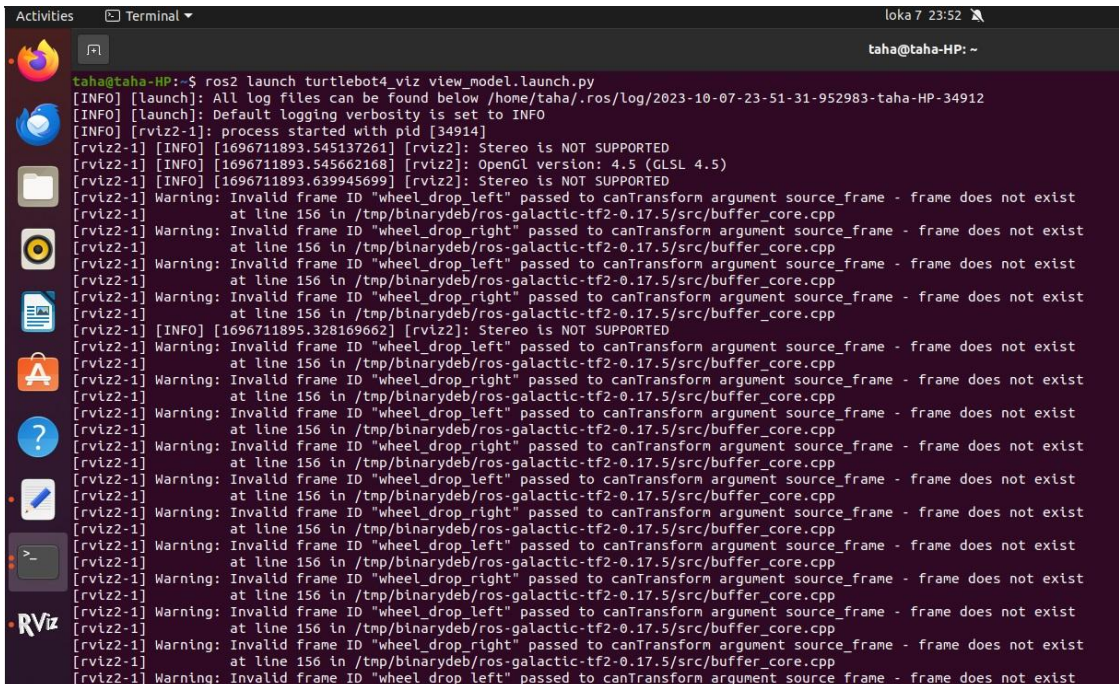
7 Rviz2

Rviz2 on ROS 2:lle portattu Rvizin versio, joka tarjoaa graafisen käyttöliittymän, jonka avulla voidaan tarkastella robotin anturidataa, karttoja ja muuta vastaavaa dataa (kuva 43 ja 44). Se on asennettu ROS 2:een oletuksena ja sen käytön edellytyksenä on Ubuntu desktop -versio.

Turtlebot4 Desktop -paketti sisältää käynnistystiedostoja ja asetuksia TurtleBot4:n tarkastelemiseen Rviz2:lla. Mallin tarkasteluun ja anturidatan tarkastelemiseen voi suorittaa komennon "ros2 launch turtlebot4_viz view_model.launch.py".

Kun tämä komento käynnistettiin, silloin nähtiin robotin malli ja sen ympäristö visuaalisesti. Visualisointiohjelma näyttää esimerkiksi robotin sijainnin, kameran kuvan ja laserskannerin havainnot. Tällainen visualisointi on tärkeä keino seurata robotin toimintaa ja saada parempi ymmärrys sen liikkeistä ja ympäristössä tapahtuvista tapahtumista.

Tämä komento toimii vain, kun TurtleBot4-ohjelman asennettu ja sen yhteydessä tarvittavat ohjelmistot. Sen tulee myös kytkeä TurtleBot4-simulaattori tai robotti varusteineen ohjattavaksi.

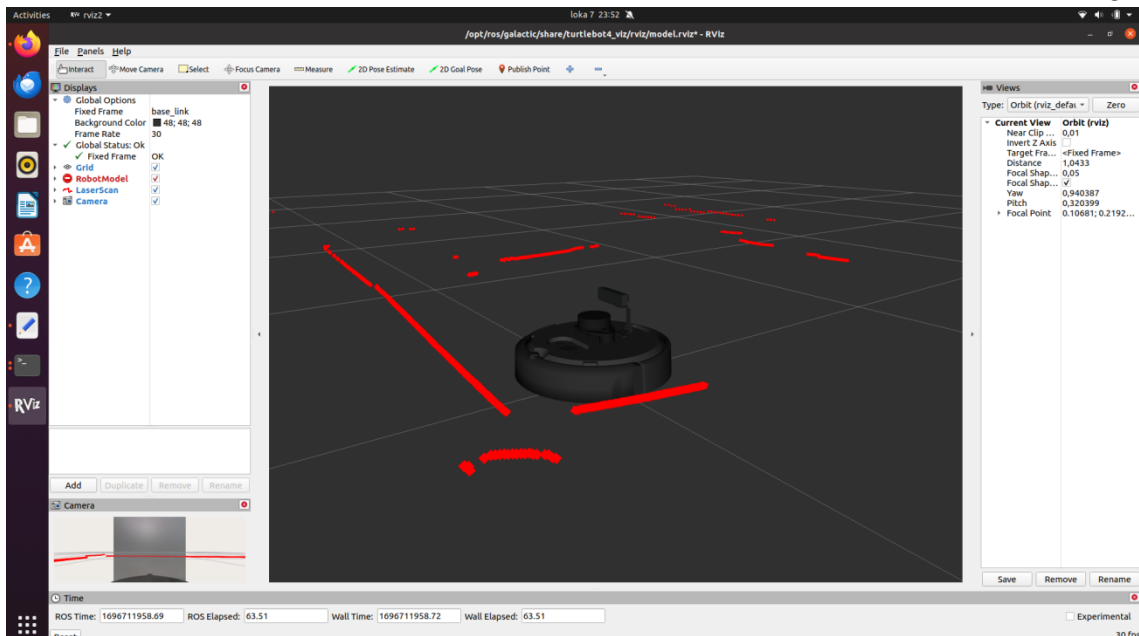


```

taha@taha-HP:~$ ros2 launch turtlebot4_viz view_model.launch.py
[INFO] [launch]: ALL log files can be found below /home/taha/.ros/log/2023-10-07-23-51-31-952983-taha-HP-34912
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [rviz2-1]: process started with pid [34914]
[rviz2-1] [INFO] [1696711893.545137261] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-1] [INFO] [1696711893.545662168] [rviz2]: OpenGL version: 4.5 (GLSL 4.5)
[rviz2-1] [INFO] [1696711893.639945699] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] [INFO] [1696711895.328169662] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_right" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp
[rviz2-1] Warning: Invalid frame ID "wheel_drop_left" passed to canTransform argument source_frame - frame does not exist
[rviz2-1] at line 156 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer_core.cpp

```

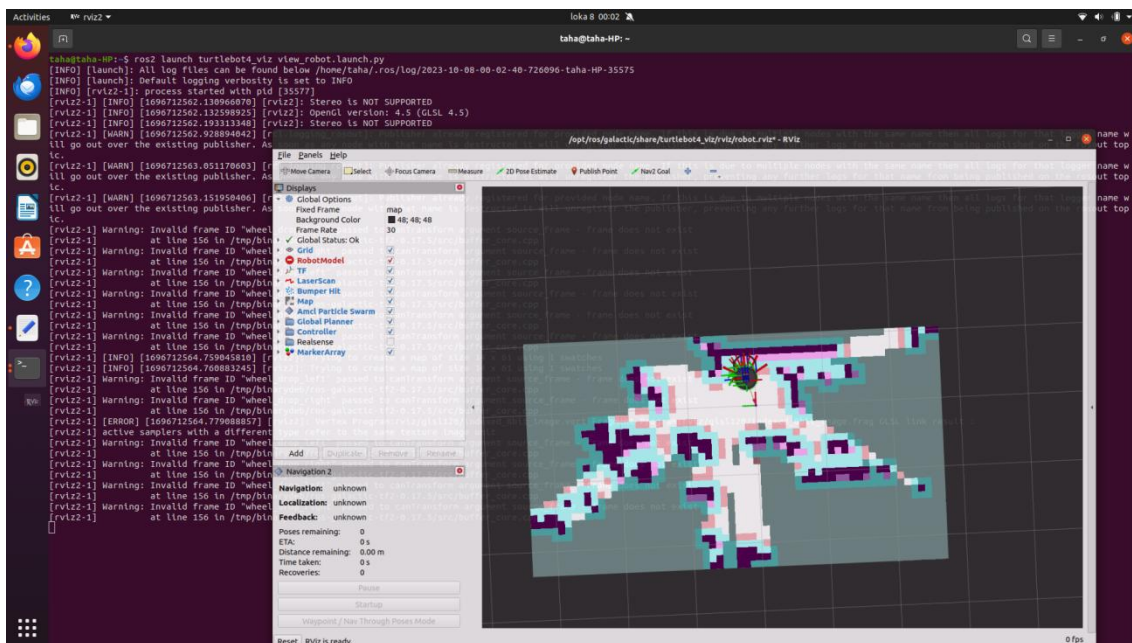
Kuva 43. Komento käynnistetään, jolloin näkee robotin mallin



Kuva 44. Laitteen malli

Navigointi: Kun halutaan tarkastella robottia ylhäältä päin sen ympäristössä, voidaan suorittaa komento: "ros2 launch turtlebot4_viz view_robot.launch.py". Tämä on hyödyllistä silloin, kun robotti liikkuu ympäristössä ja halutaan karttaa sen liikkeitä, tai navigoida sen kanssa.

Näin voidaan helposti seurata robottia, sen meneillään olevia liikkeitä ja reittiä ympäristössä, jossa se liikkuu. Tämä parantaa käyttäjän kokemusta robotin hallinnasta ja auttaa robottia toimimaan tehokkaammin ja turvallisemmin (kuva 45).

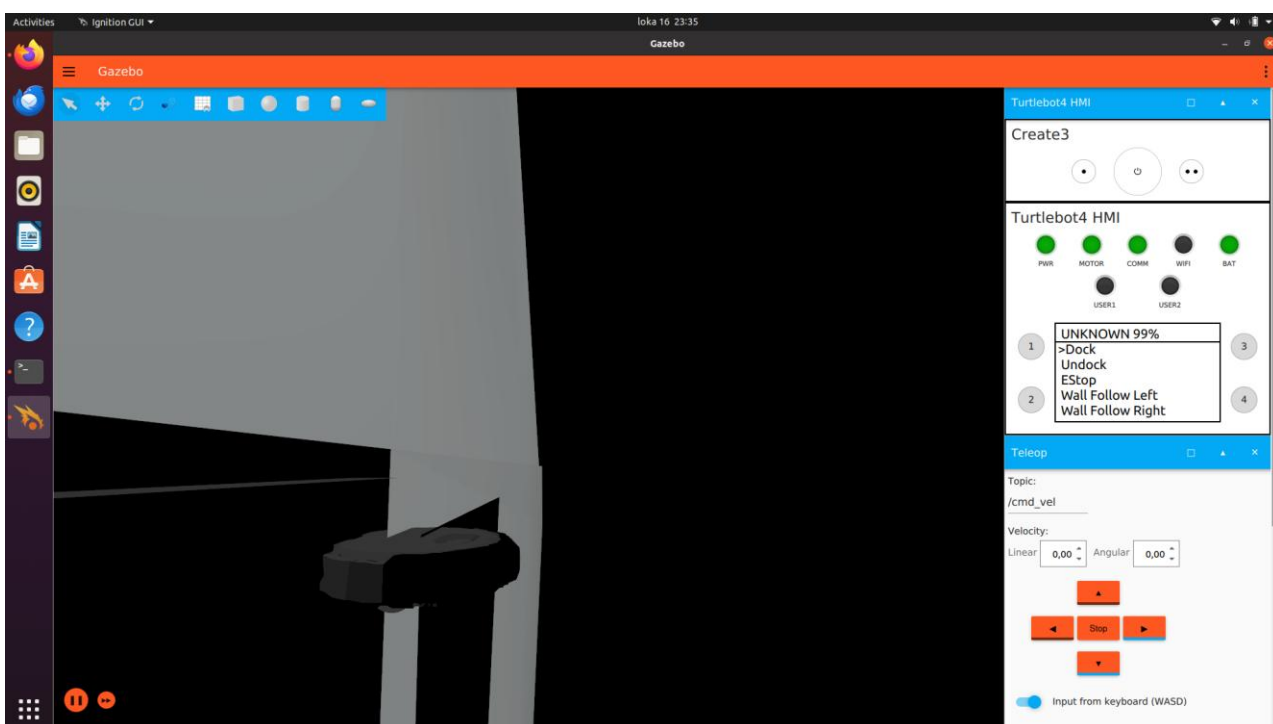


Kuva 45. Tarkastellaan robottia ylhäältä päin sen ympäristössä

Simulaattori: on ohjelmisto tai järjestelmä, joka mahdollistaa reaaliaikaisen tai lähellä reaaliaikaista simulaation ympäristöstä, ilmiöistä tai prosesseista. Simulaattorit ovat hyödyllisiä monissa eri konteksteissa, mukaan lukien koulutus, testaus, kehitys ja viihde. Simulaattorit voivat jäljitellä monenlaisia skenaarioita ja ympäristöjä. Niitä käytetään usein tilanteissa, joissa todellinen toiminta voisi olla vaarallista, kallista tai muuten hankalaa.

TurtleBot4-simulaattori liittyy todennäköisesti robotiikkaan ja ROS 2:een (Robot Operating System 2). Tämä simulaattori mahdollistaa robotin käyttöön liittyvien tehtävien ja algoritmien testaamisen virtuaalisessa ympäristössä ennen kuin niitä käytetään oikeassa fyysisessä robotissa. Simulaattori voi sisältää simuloidun version robotista ja sen ympäristöstä, jotta kehittäjät voivat testata ja optimoida ohjelmistojaan ennen kuin ne otetaan käyttöön todellisessa robotissa.

Kun komento `ros2 launch turtlebot4_ignition_bringup ignition.launch.py` suoritetaan, se käynnistää TurtleBot4-simulaattorin oletusasetuksilla. Tämä tarkoittaa, että virtuaalinen TurtleBot4-robotti ja sen ympäristö ovat käytettävissä simulaattorissa, ja voi suorittaa erilaisia kokeita ja testejä kyseisessä simulaatioympäristössä. Simulaattori on hyödyllinen työkalu robotiikan kehittäjille, jotta he voivat hienosäätää ja testata robotin ohjausjärjestelmiä ennen kuin siirrytään fyysisen robotin käyttöönottoon (kuva 46).



Kuva 46. Simulaattori

SLAM ja Nav2: Synkroninen SLAM ja Nav2 tarkoittaa sitä, että käytetään samanaikaista paikannusta ja kartanmuodostusta Navigointipino (Nav2) -ohjausjärjestelmän kanssa. Tämä on tärkeää autonomisissa robottisovelluksissa.

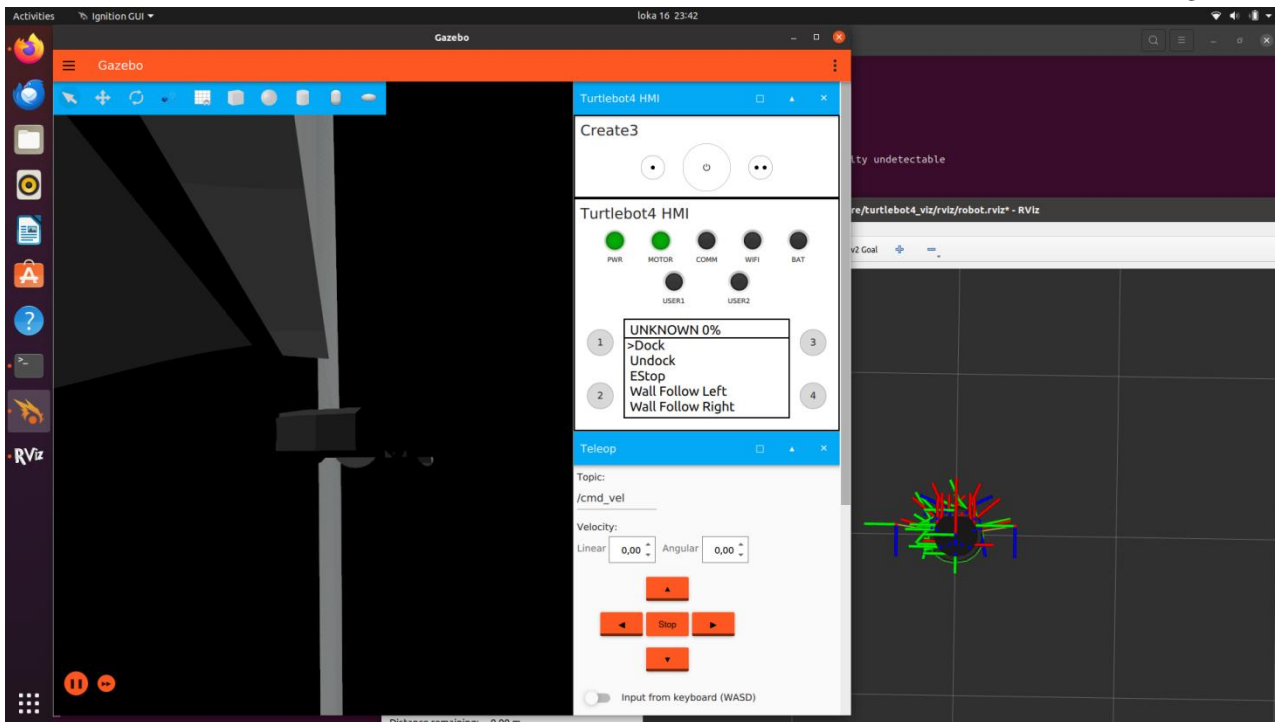
Synkroninen SLAM Samanaikainen paikannus ja kartanmuodostus tarkoittaa, että robotti paikantaa itsensä ja luo karttaa ympäristöstään samanaikaisesti. Toisin sanoen robotti tietää, missä se on suhteessa ympäristöön ja voi samalla päivittää ympäristön karttaa.

Nav2: Nav2 on ROS 2 -ohjelmisto tarjoaa navigointiin liittyviä toimintoja roboteille. Se käyttää erilaisia antureita ja ohjausjärjestelmiä paikannukseen ja kartanmuodostukseen.

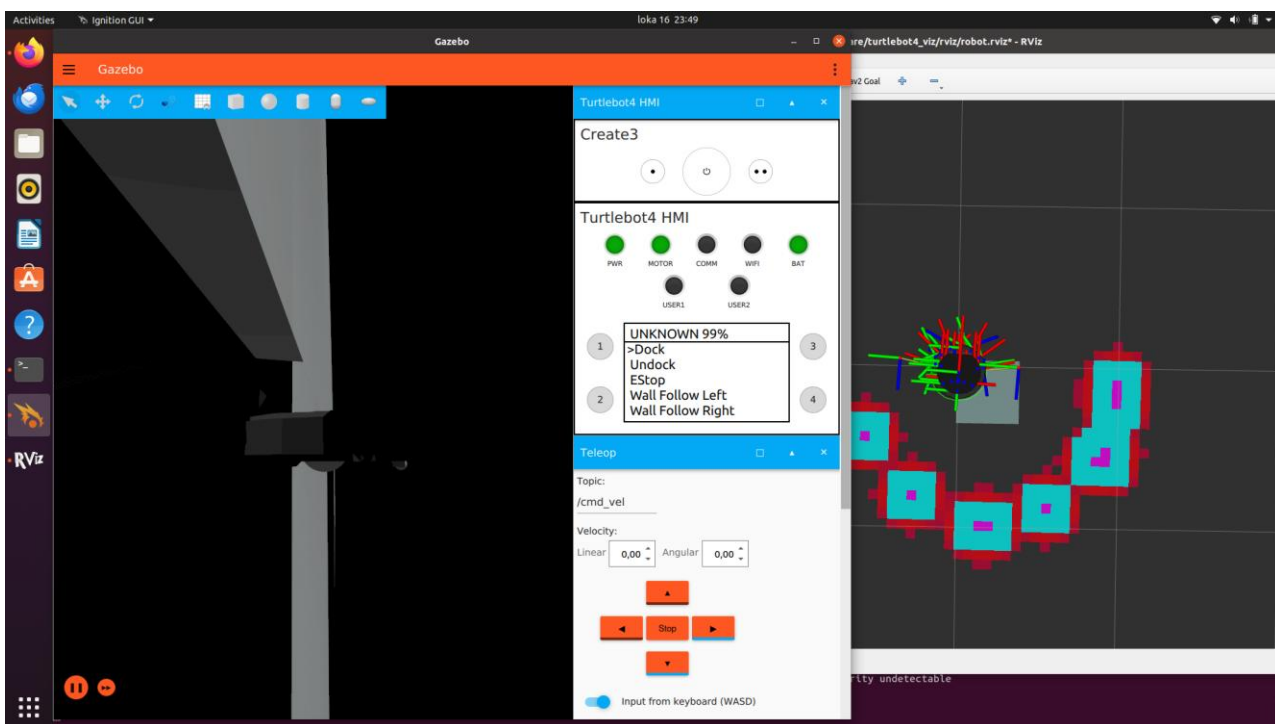
Komennon `ros2 launch turtlebot4_ignition_bringup ignition.launch.pyslam:=sync nav2:=true rviz:=true` avulla käynnistetään TurtleBot 4 -robottisimulaattori samanaikaisella SLAM:lla Nav2:n ja RVizin kanssa:

`slam:=sync` määrää simulaattorin suorittamaan samanaikaista paikannusta ja kartanmuodostusta.`nav2:=true` ottaa käyttöön Nav2-ohjausjärjestelmän, jotta robotti voi suorittaa navigointia.`rviz:=true` avaa RViz-ohjelman, joka mahdollistaa simulaattorin tilan ja kartan visualisoinnin.

Tämä komento käynnistää siis simulaattorin, joka mahdollistaa robotin samanaikaisen paikannuksen, kartanmuodostuksen ja navigoinnin testaamisen virtuaaliympäristössä. Se myös tarjoaa visuaalisen käyttöliittymän simulaation seuraamiseen (kuva 47 ja 48).



Kuva 47. RobottiSimulaattori



Kuva 48. SLAM Nav2:lla

Insinööriyön tavoitteena oli valmistella TurtleBot4-robotti opetuskäyttöön ja samalla oppia ROS:in käytäntöä, erityisesti karttojen luomista roboteilla. Hyviä karttoja saatiin luotua, mutta se vaati aikaa erityisesti, kun otettiin huomioon ympäristön laajuus ja monimutkaisuus. Parhaimmat tulokset saavutettiin yksinkertaisissa ja pienissä ympäristöissä.

Laitteen osien purkaminen ja asentaminen ei ollut vaikeaa, mutta keskeistä oli varmistaa laitteen osien eheys säilyttämällä.

Työskennellessäni opin paljon uutta ROS-järjestelmästä, aikapalvelimista ja Linux-käyttöjärjestelmästä. Työn aikana sain kertyneestä kokemuksesta paljon hyötyä ja opin lukuisia asioita robottien parissa työskentelystä. Tämä työ oli erittäin hyödyllistä ja olen nyt erittäin tyytyväinen sen tuloksiin.

Lähteet

- 1 ROS. Verkkodokumentti. <<https://www.ros.org/>> . Luettu 24.9.2023.
- 2 iRobot Video TurtleBot4-laitteistosta ja -ohjelmistosta. Verkkodokumentti. <<https://edu.irobot.com/create3-setup>>. Luettu 26.9.2023.
- 3 Tietoja laitteistosta. Verkkodokumentti. <https://iroboteducation.github.io/create3_docs/hw/overview/> . Luettu 27.9.2023.
- 4 Turtlebot4 PäivitysLaiteohjelmiston lataus automaattisesti. Verkkodokumentti. <<http://edu.irobot.com/create3-latest-fw>>. Luettu 2.10.2023.
- 5 TurtleBot4 Valitse koneen malli ja päivitys. Verkkodokumentti. <https://iroboteducation.github.io/create3_docs/releases/overview/> . Luettu 2.10.2023.
- 6 ROS2 Asennus. Verkkodokumentti. <https://iroboteducation.github.io/create3_docs/setup/ubuntu2004/> . Luettu 2.10.2023.
- 7 Tietoja Create3. Verkkodokumentti. <<https://edu.irobot.com/learning-library/create-3-getting-started>>. Luettu 3.10.2023.
- 8 TurtleBot4. Verkkoasetukset. Verkkodokumentti. <<https://www.generationrobots.com/blog/en/quick-start-guide-for-turtlebot-4-mobile-robot/>> . Luettu 4.10.2023.
- 9 Create3. Verkkodokumentti. <<https://edu.irobot.com/support/create-3-support>> Luettu 5.10.2023.
- 10 Tuotteen yksityiskohdat. Verkkodokumentti. <<https://edu.irobot.com/what-we-offer/create3>> . Luettu 6.10.2023.
- 11 Create3-etulevyn poistaminen. Verkkodokumentti. <https://players.brightcove.net/6055873617001/rwlsGjJMH_default/index.html?videoId=6306280352112> . Luettu 8.10.2023.
- 12 Video työstäni. Verkkodokumentti. <https://drive.google.com/drive/folders/19QvN37_yXxRWNmhjcBmKDvYMWIVxk4Sd> . Luettu 20.10.2023.

Asennus ROS2 Galactic

```
echo $LANG
```

```
sudo apt update && sudo apt install -y curl gnupg2 lsb-release build-essential git  
cmake
```

```
sudo curl -ksSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o  
/usr/share/keyrings/ros-archive-keyring.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-  
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" |
```

```
sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

```
sudo apt update && sudo apt install -y ros-galactic-ros-base python3-colcon-  
common-extensions python3-rosdep ros-galactic-rmw-fastrtps-cpp ros-galactic-rmw-  
cyclonedds-cpp ros-galactic-irobot-create-msgs
```

```
echo "source /usr/share/colcon_cd/function/colcon_cd.sh" >> ~/.bashrc
```

```
echo "export _colcon_cd_root=/opt/ros/galactic/" >> ~/.bashrc
```

```
echo "source /usr/share/colcon_argcomplete/hook/colcon_argcomplete.bash" >>  
~/.bashrc
```

```
echo "source /opt/ros/galactic/setup.bash" >> ~/.bashrc
```

```
echo "export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp" >> ~/.bashrc
```

```
source ~/.bashrc
```

```
ros2 topic list
```

```
ros2 topic list -t
```

```
ros2 service list -t
```

```
ros2 action list -t
```

```
ros2 param list
```

```
ros2 topic echo /tf
```

Creating your first node (Python) GALACTIC

```
mkdir ~/turtlebot4_ws/src -p
source /opt/ros/galactic/setup.bash
cd ~/turtlebot4_ws/src
ros2 pkg create --build-type ament_python --node-name turtle-
bot4_first_python_node turtlebot4_python_tutorials
ros2 topic info /interface_buttons --verbose
cd ~/turtlebot4_ws
colcon build --symlink-install --packages-select turtlebot4_python_tutorials
source install/local_setup.bash
ros2 run turtlebot4_python_tutorials turtlebot4_first_python_node
```

Sensors

```
sudo apt install ros-galactic-rplidar-ros
ros2 launch turtlebot4_bringup rplidar.launch.py
ros2 service call /stop_motor std_srvs/srv/Empty {}
ros2 service call /start_motor std_srvs/srv/Empty {}
```

OAK-D

```
sudo apt install ros-galactic-depthai-ros
ros2 launch turtlebot4_bringup oakd.launch.py
ros2 launch depthai_examples mobile_publisher.launch.py
```

Launch SLAM

```
sudo apt install ros-galactic-turtlebot4-navigation
ros2 launch turtlebot4_navigation slam_sync.launch.py
ros2 launch turtlebot4_navigation slam_async.launch.py
```

Launch Rviz2

```
ros2 launch turtlebot4_viz view_robot.launch.py
```

Rviz2 MODEL

```
ros2 launch turtlebot4_viz view_model.launch.py
```

View Robot

```
ros2 launch turtlebot4_viz view_robot.launch.py
```

Launching navigation

```
ros2 launch turtlebot4_navigation nav_bringup.launch.py slam:=off localization:=true
map:=office.yaml
ros2 launch turtlebot4_ignition_bringup ignition.launch.py nav2:=true slam:=off loca-
lization:=true
```

Launching Ignition Gazebo

```
ros2 launch turtlebot4_ignition_bringup ignition.launch.py model:=lite
```

```
ros2 launch turtlebot4_viz view_model.launch.py
```

```
ros2 launch turtlebot4_viz view_robot.launch.py
```

```
ros2 launch turtlebot4_ignition_bringup ignition.launch.py
```