A Comprehensive user guides
Bveeta mini: Model type R

User manual version 1.0

Warning!

This is advanced mobile robot. A proper guide by the professional is required.

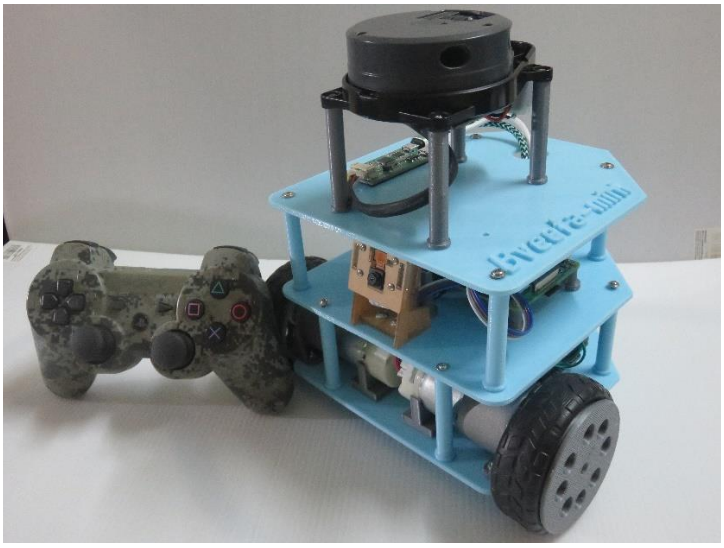
Chapter 1

Introduction to Bveeta mini, assembly process and configuration

What is Bveeta Mini

Bveeta Mini type R is a ROS standard platform robot. Bveeta is based on two wheeled mobile robot, which was driven by the educational computer programming language. Bveeta, which originated from the two wheeled mobile robot, is designed to teach people to use ROS through Bveeta mini type R platform as well as to teach computer programming language. In the early 2021, Bveeta mini has become the standard platform of ROS, which is the most popular platform among developers and students.

There are 2 versions of the Bveeta mini model. The first model was ***Bveeta mini***, and the second model is ***Bveeta mini type R*** as shown in Figure 1.0. Both models have the same capabilities and function.



Bveeta mini

Bveeta mini type R

Figure 1.0 Bveeta mini and Bveeta mini type R.

Bveeta mini was developed by Bizbot Technology, a start-up tech company from Malaysia who has more than 16 years' experience in developing mobile robot platform and has wide experience in electronics control, programming, and mechanical design of mobile robot platform. With the demanding of the open-source platform, thus Bveeta Mini start to revolute its own technology to meet the market needs. In research area, Bveeta mini helps many researchers to accelerate their product design by making this Bveeta mini as an example of product of the shelf and ready to be deploy. The basic components such as its controller, electronics and software are easy to be re-engineered to create new platform.

With a small size, affordable, programmable, and ROS-ready it for use in education, research, hobby, and product prototyping making it popular among the roboticists. The goal of Bveeta mini is to dramatically reduce the size of the platform and lower the price without having to sacrifice its functionality and quality, while at the same time offering expandability. The Bveeta mini can be customized into various ways depending on how you reconstruct the mechanical parts and use optional parts such as the computer and sensor. In addition, Bveeta mini is evolved with cost-effective and small-sized single board computer (SBC) that is suitable for robust embedded system, 360-degree distance sensor and 3D printing technology.

The Bveeta mini's core technology is SLAM, Navigation and Localization, making it suitable for home service robots. It can run SLAM (simultaneous localization and mapping) algorithms to build a map and can drive around your room. Also, it can be controlled remotely from a laptop, joypad or Android-based smart phone. Through this compatibility can compensate for the lack of freedom and can have greater completeness as a service robot with the SLAM and navigation capabilities that the Bveeta mini has.

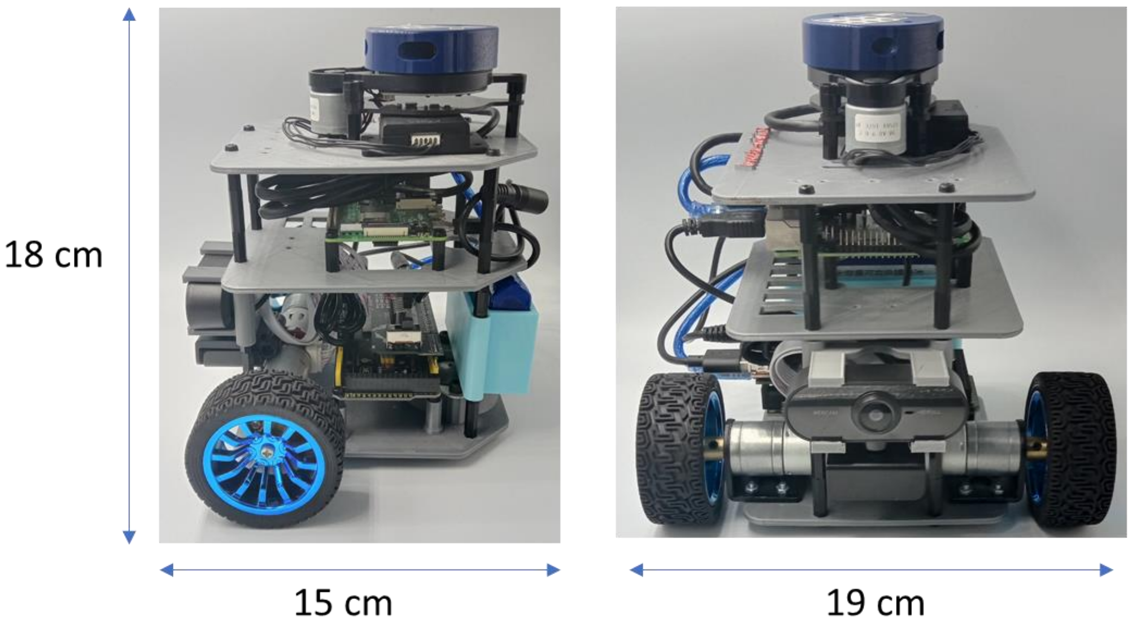
Part list

Bveeta mini type R comes with the components listed as follow:

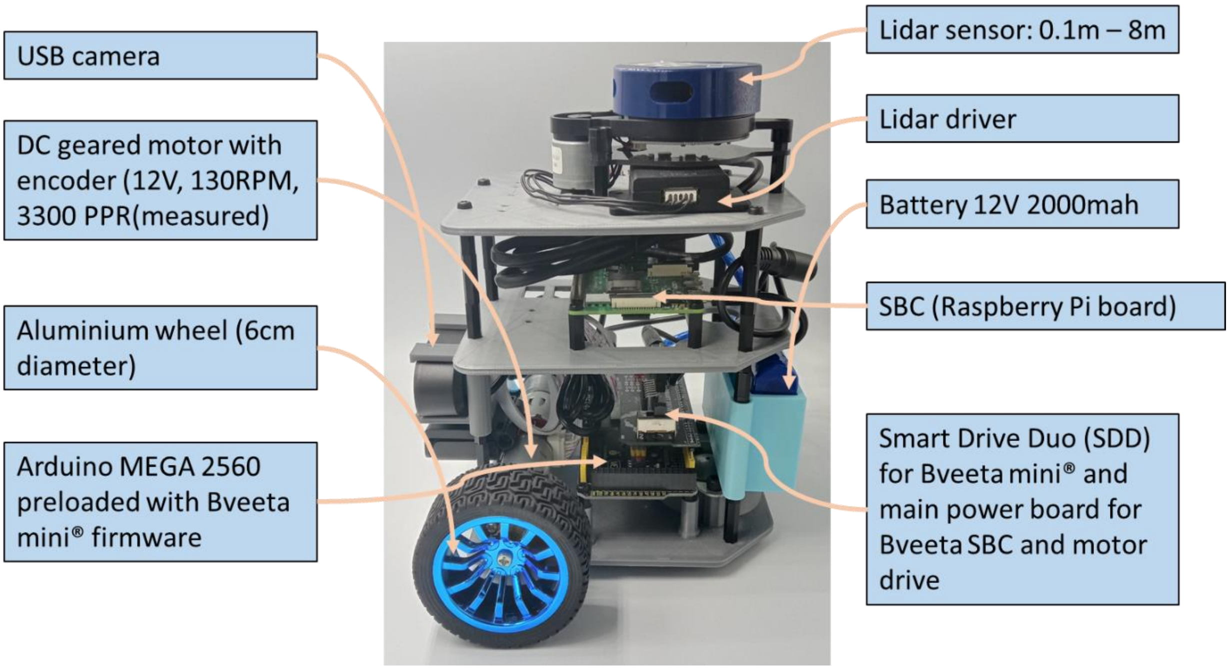
1. Single board computer Raspberry pi 4 Model B (Year 2021 version)-Tested
2. High quality Arduino MEGA main controller board
3. Smart drive DUO with PID control ROS ready specially design for Bveeta mini platform
4. 2 x DC geared motor with 2200 / 3300 CPR (Measured) (Either one)
5. 2 x Metal Wheels with flat tyre for perfect performance on the ground
6. High resolution web camera for image processing purposes
7. 1 x 2000Mah 12V NIMH battery for powering the Bveeta mini electronic and SBC supply (Operating hours 2.5 Hr continuous)
8. 1 x 2D lidar 360^{o} with minimum distance 0.1m up to 8m range.
9. 1 x battery charger
10. set of mechanical structure body created with 3D printing technology

11. Protective hard case for safe storage
12. x SD card preloaded with Ubuntu 18 and ROS melodic (Comes with Bveeta mini ROS code and ready to run)

Dimensions and labels

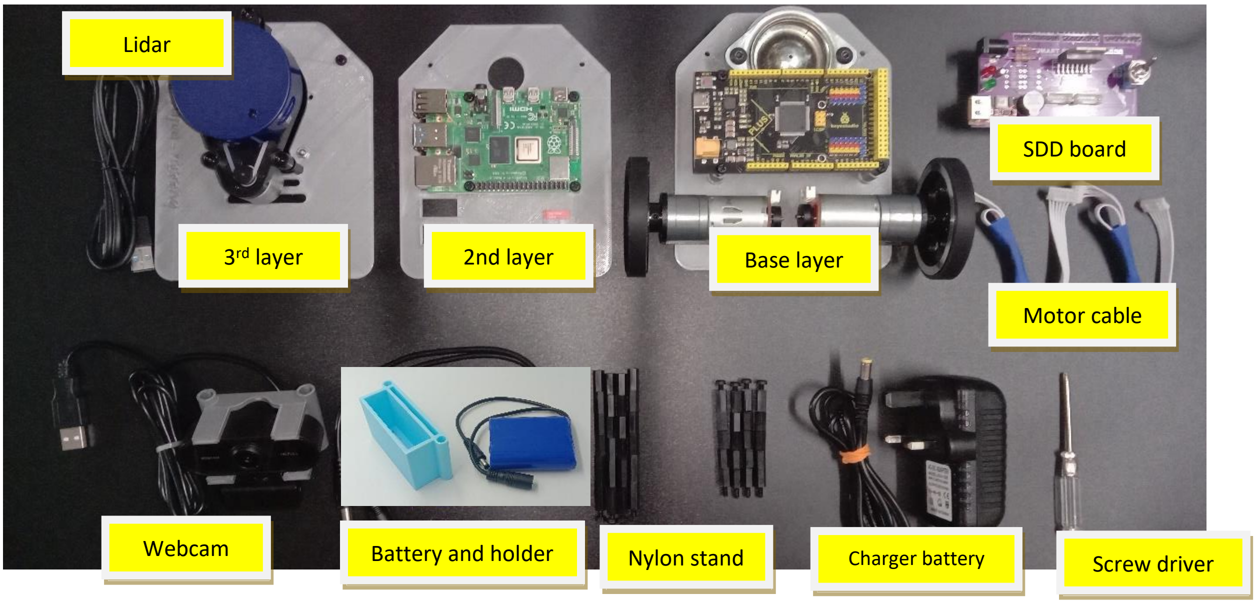


Note: Starting on 1st October 2021, Bveeta Mini Type R model has been replaced with metal wheels



Components details

The components details are shown below:



Assembly instructions

Bveeta mini has 3 layers:

1. Base layer
2. Second Layer
3. Third Layer

Base layers consist of:

1. 2 DC geared motor mounter on the 3D printed plate
2. 1 unit metal ball castor wheel
3. Arduino MEGA 2560
4. Smart drive duo (SDD) Specifically designed for Bveeta Mini
5. Web camera
6. Battery holder
7. 12V battery

Second layer consist of:

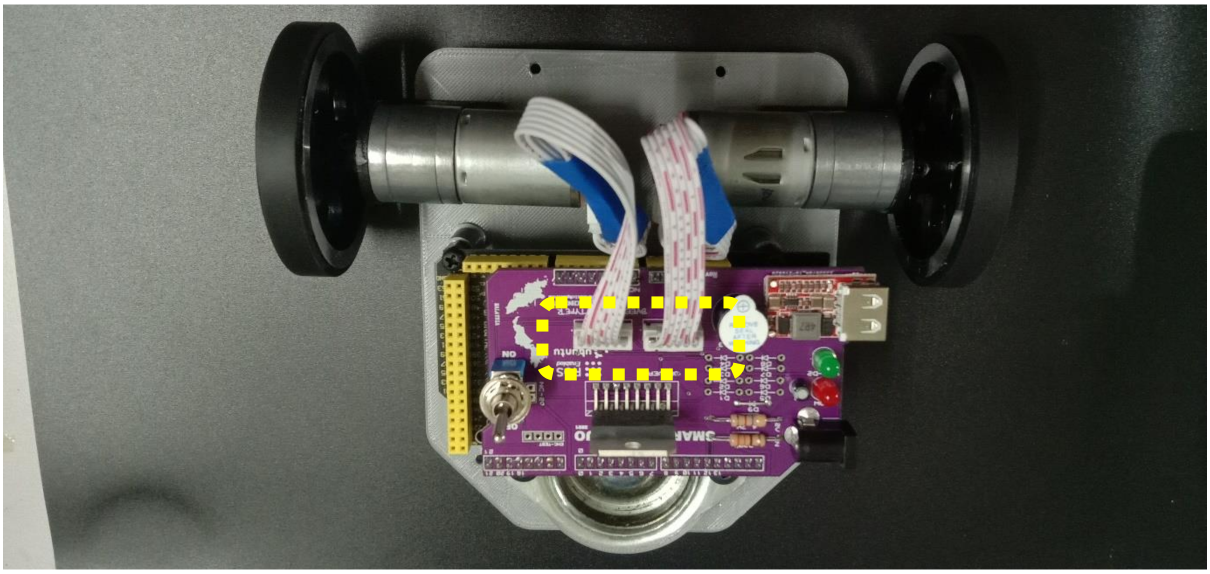
1. Main SBC – Raspberry Pi

Third Layer consist of:

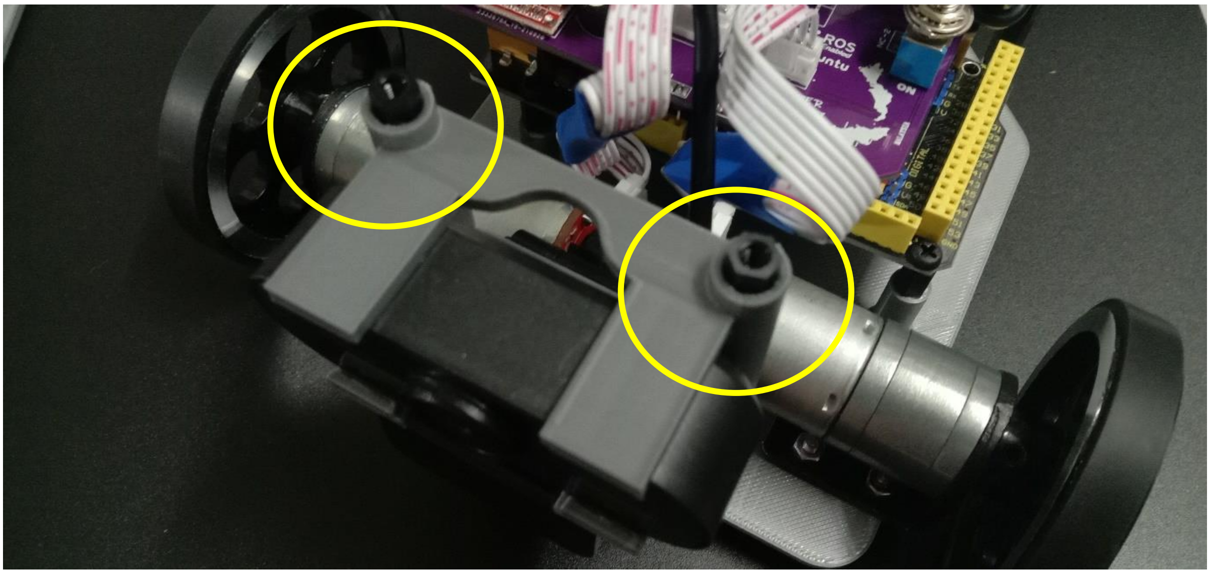
1. 2D lidar
2. Lidar interface board

Base layer assembly process

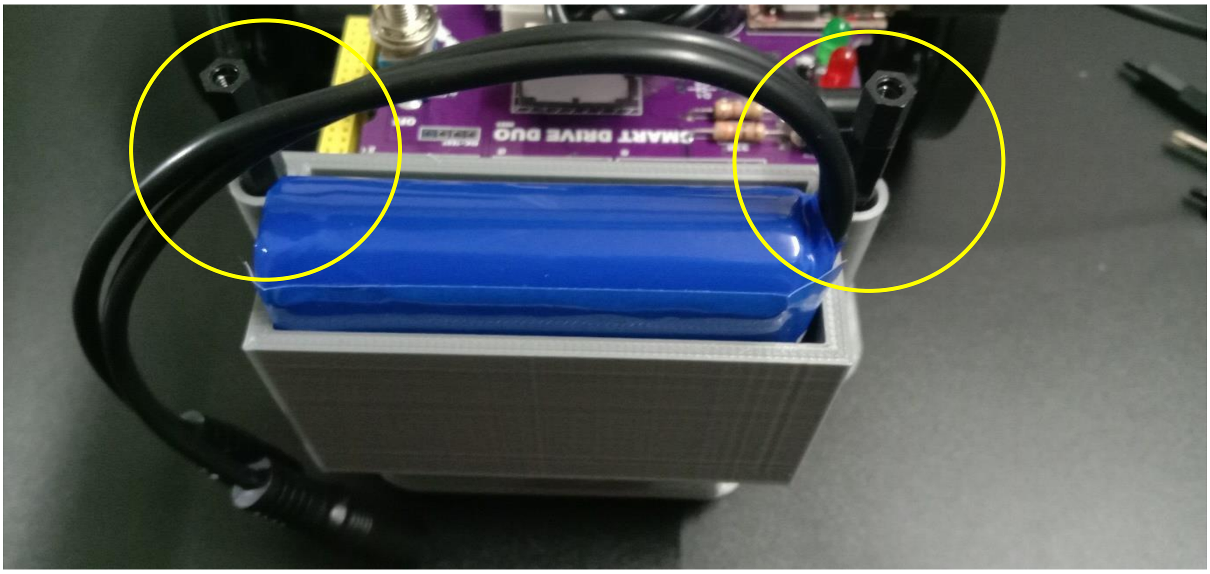
Step 1: Attach the motor connector to the SDD board as shown in the Figure:



Step 2: Slide in the USB camera into the nylon stand:

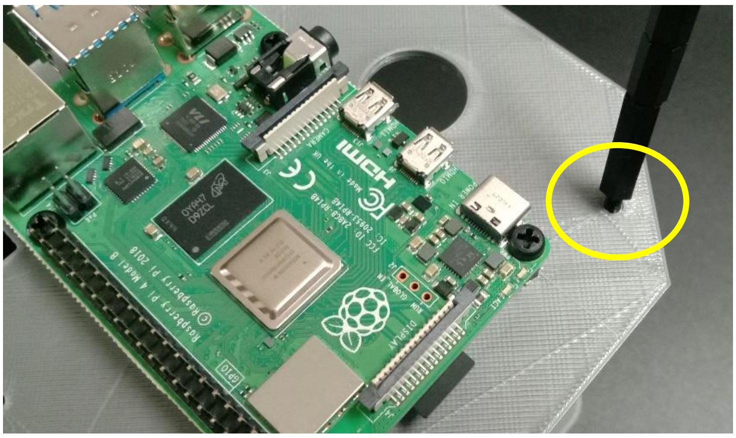


Step 3: Slide in the battery holder into the nylon stand:

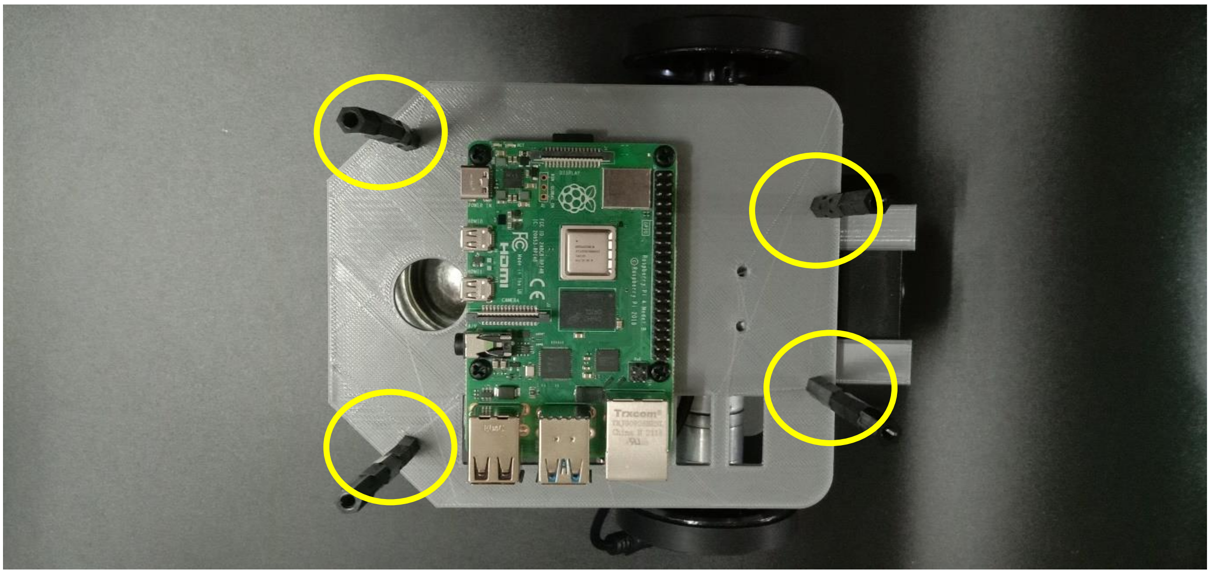


Second layer assembly process

Step 1: Place the second layer plate (That has raspberry pi SBC) on top of the base layer

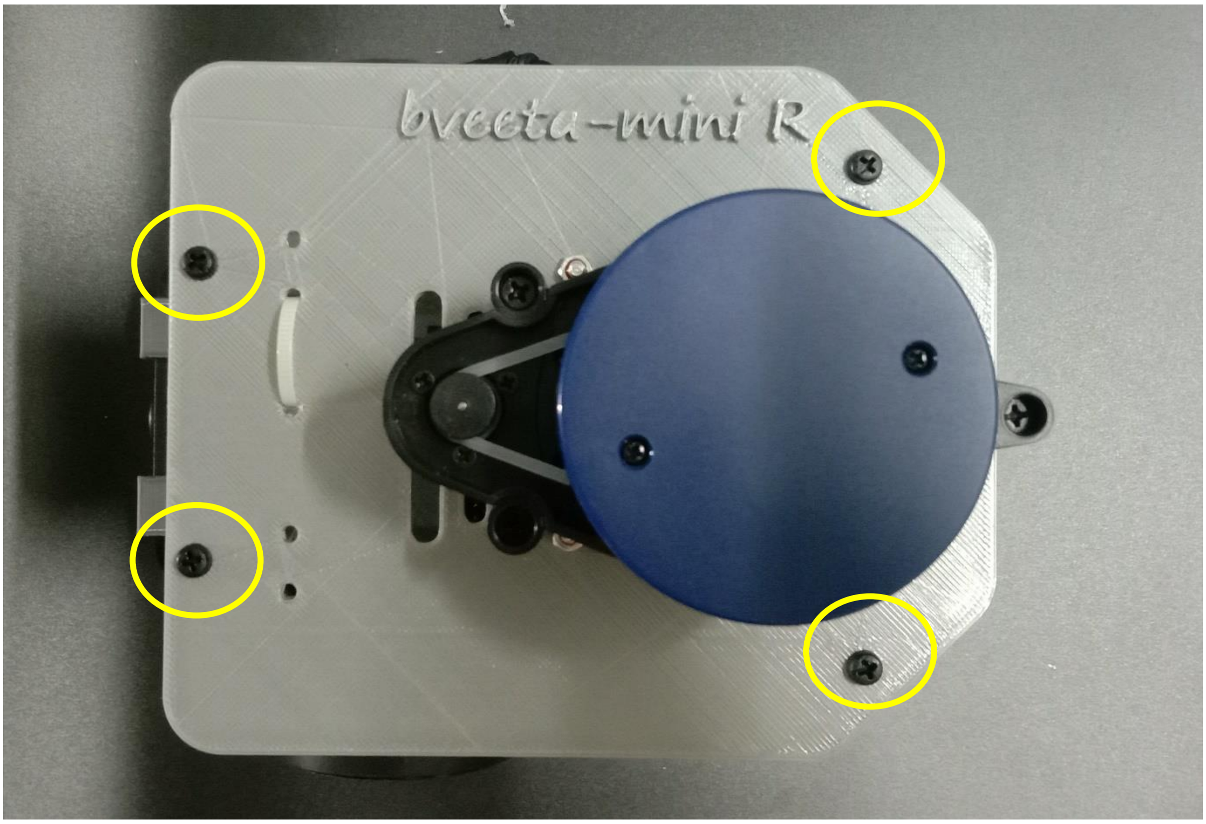


Step 2: Screw all 4 nylon stands but make sure it is not too tighten because it can damaged the nylon screw tread.

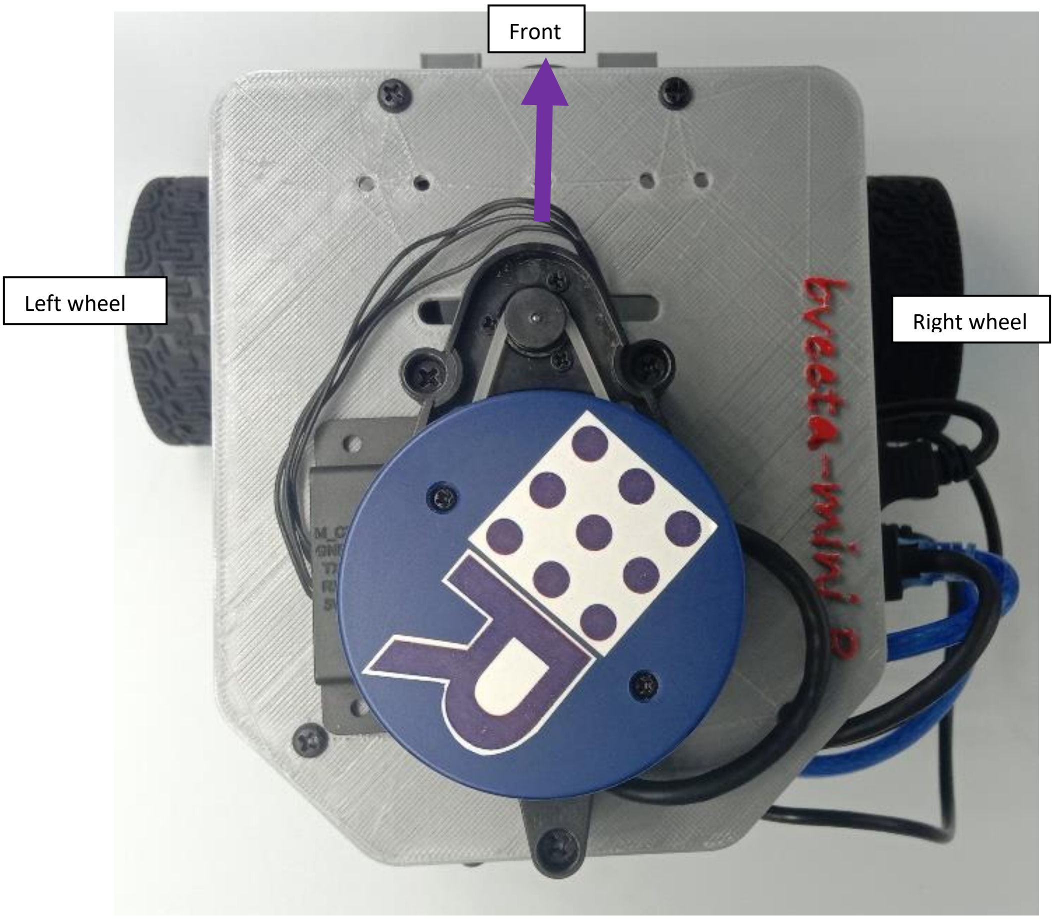


Third layer assembly process

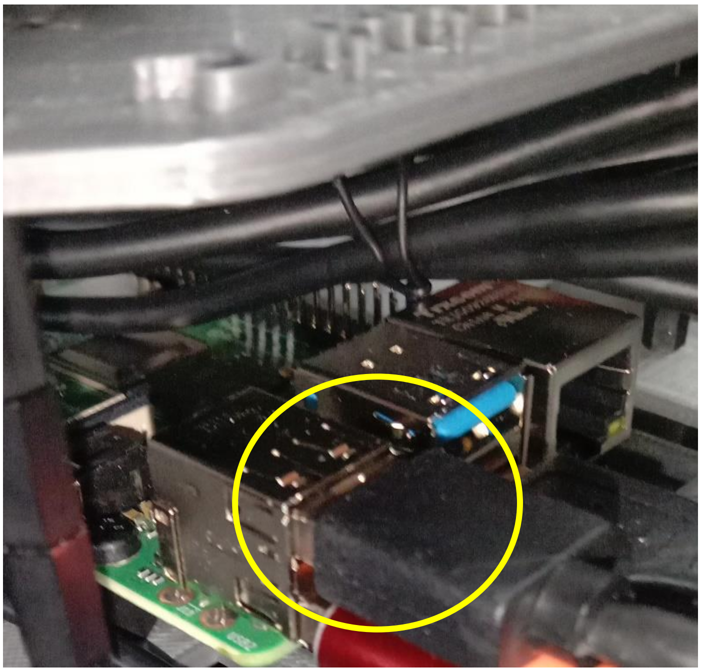
Step 1: Place the 3rd layer on top of the second layer and screw all 4-nylon



Step 2: Make sure all layers are aligned as shown:

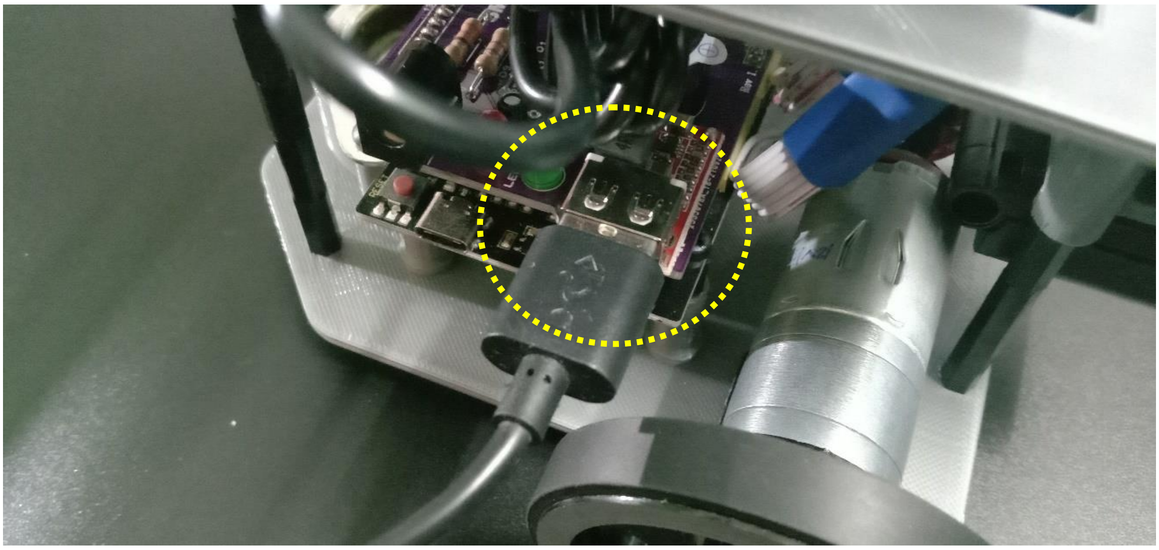


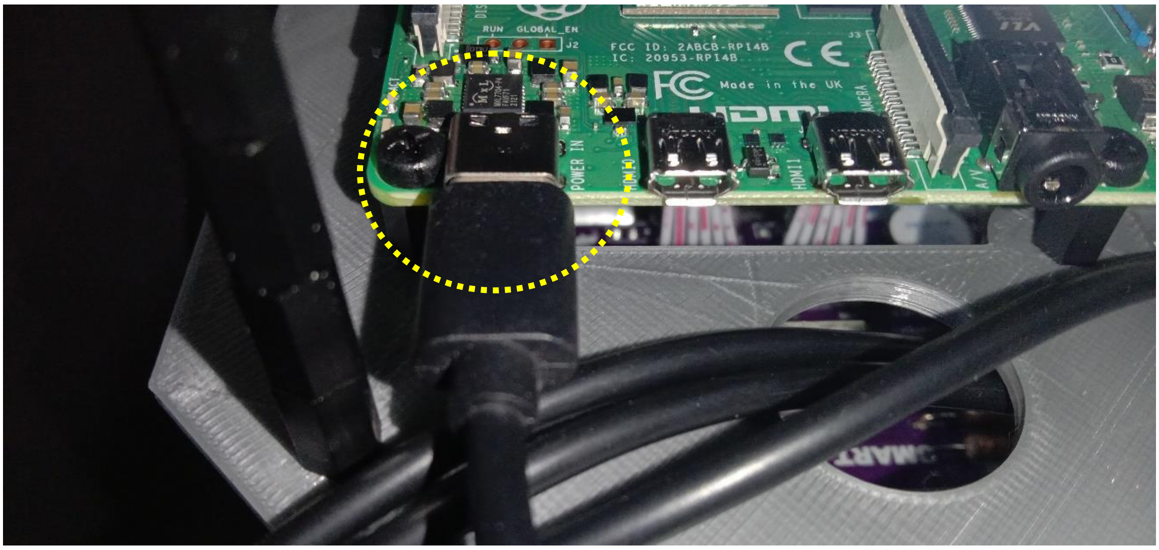
Step 3: Connect the USB cable lidar to the raspi USB port as shown.



Step 4: Connect USB camera cable to raspi USB port

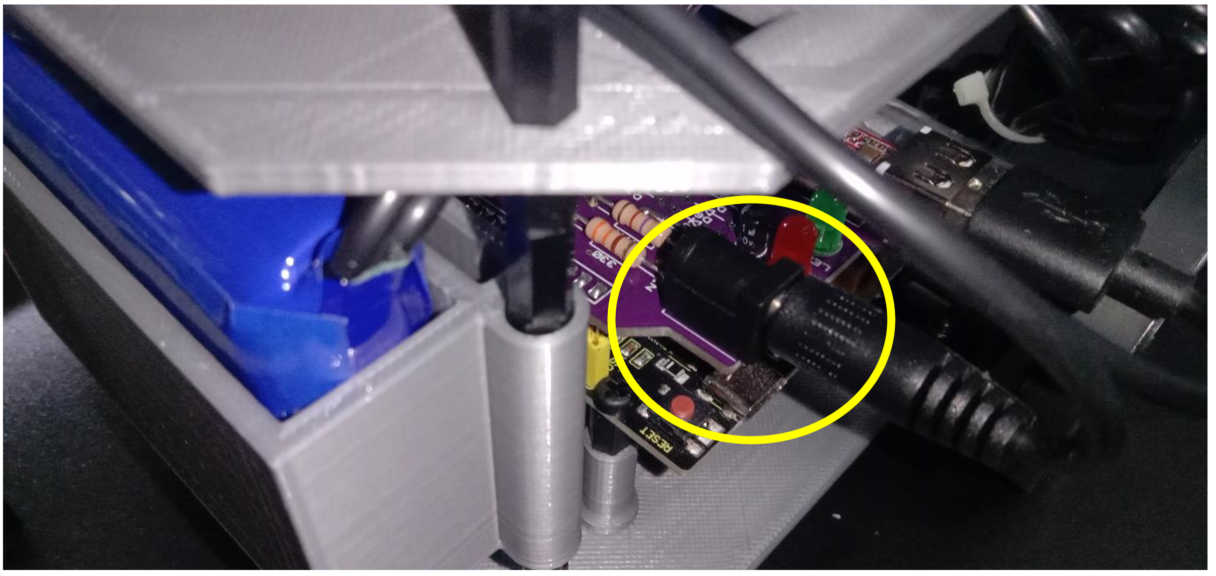
Step 5: Connect the USB type C from raspi to power the USB power outlet on the SDD as shown.





Step 6: Connect the battery to the SDD. (Make sure the power switch on the SDD is turn off)

Please make sure to connect the battery power jack to SDD and not the ARDUINO MEGA



Charging the battery

When robot is automatically tuned off while its runs, that means the battery need to be recharged. It is advisable to save your coding regularly when developing your project. At this moment, Bveeta mini series does provide any battery level indicators since it is power by the Lithium Ion Battery. To charge the battery, connect the 12V Charger adaptor to the battery female DC jack shown:

The battery is full charged after **3hours charging**.

User may disengage the battery and connect the charger only directly to the SDD board if the development requires longer time before deployment and doesn't require the battery.

Caution!

Do not use other power adaptor to supply power to the SDD as this can damaged the SDD board permanently. Use the only provided adaptor to charge the battery and powering the SDD board.

Connect to Bveeta Mini Type R

Bveeta mini by default is connected via ethernet connection (LAN) through the manual settings of its IP address which is default to:

Bveeta mini ethernet ip address: 169.254.1.10

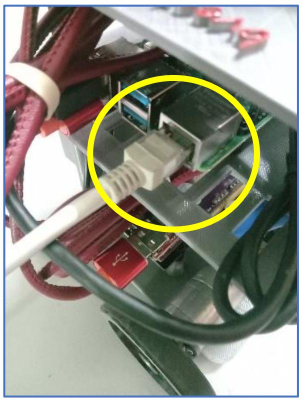
For the first time connections, user must connect Bveeta mini through the LAN cable provided and set your computer ethernet connection to static ip 169.254.1.11

There are 2 methods can be used to connect to Bveeta Mini.

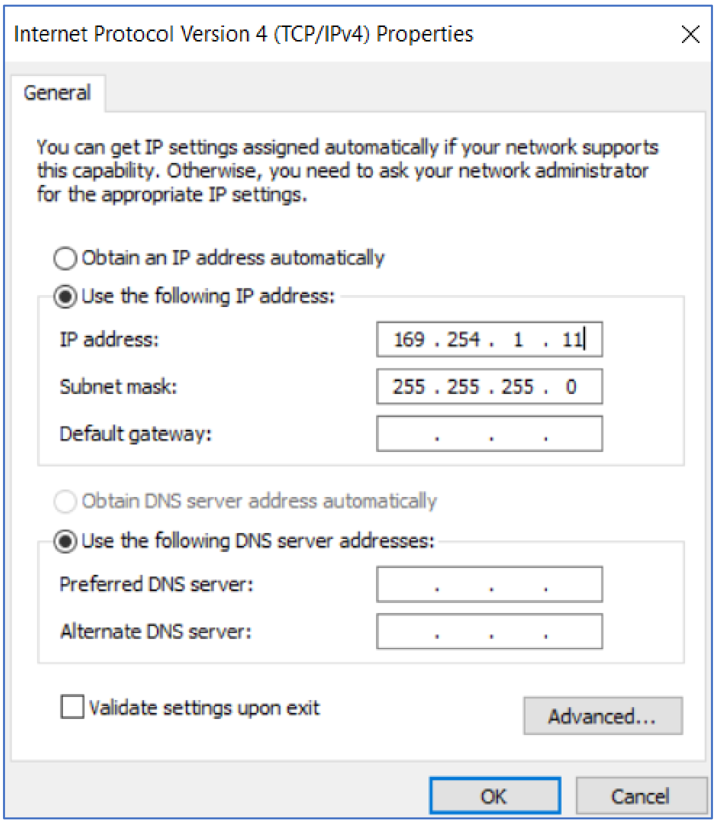
1. Remote desktop connections with Ethernet cable
2. Remote desktop connections with Wifi connections (Access Point)

Method 1: Remote desktop connections (Windows OS user)

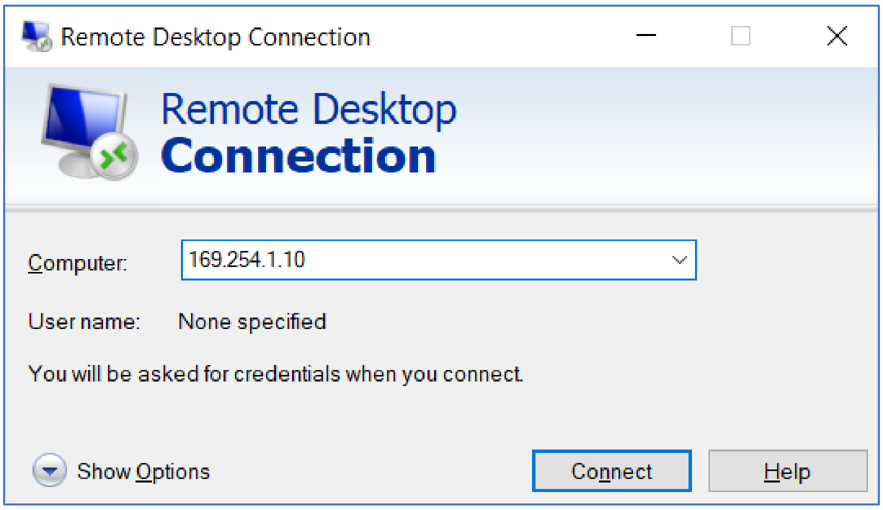
1. Connect the LAN cable that is provided from your Remote computer to Bveeta mini raspi SBC as shown in figure below.

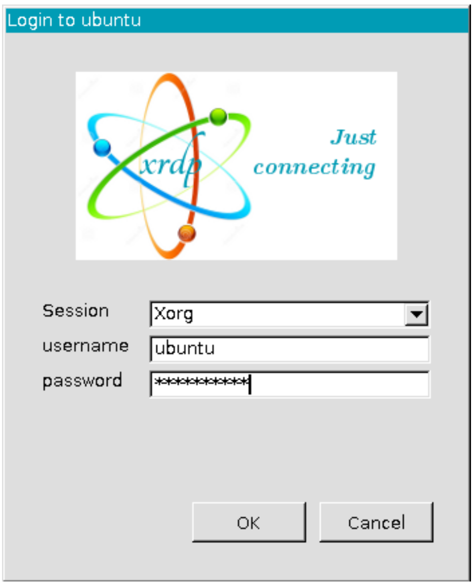


2. In your PC environment, open the "networks and internet settings". Select ethernet options and open the "Change adaptor options"
3. Select the "Ethernet" and open its properties.
4. In the properties windows, select the TCP/ipV4 and select its properties.
5. In the ipv4 properties change the following accordingly as shown:



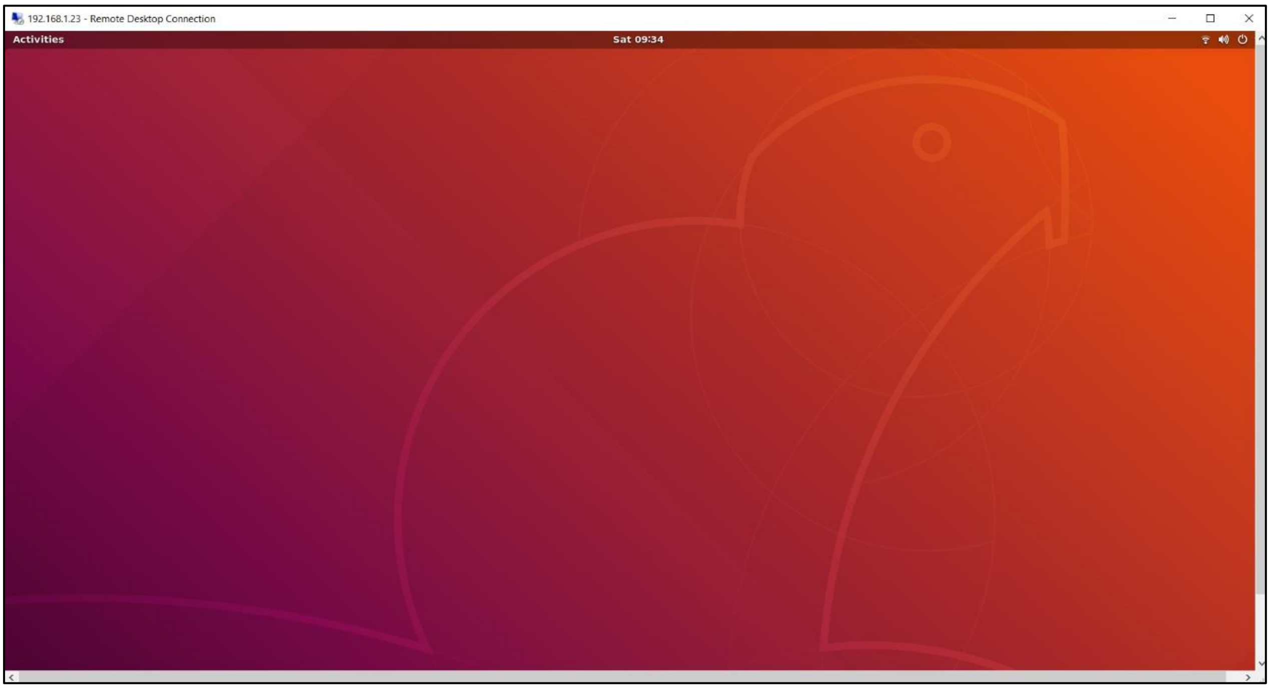
6. Save by click the ok button.
7. Next, open the "Remote Desktop" connections and set as follow:





8. Click connect and key in as follow:
    a. Username: **ubuntu**
    b. Password: **bveeta2021**

9. Once ok, you should see the display exactly as shown in Figure below.



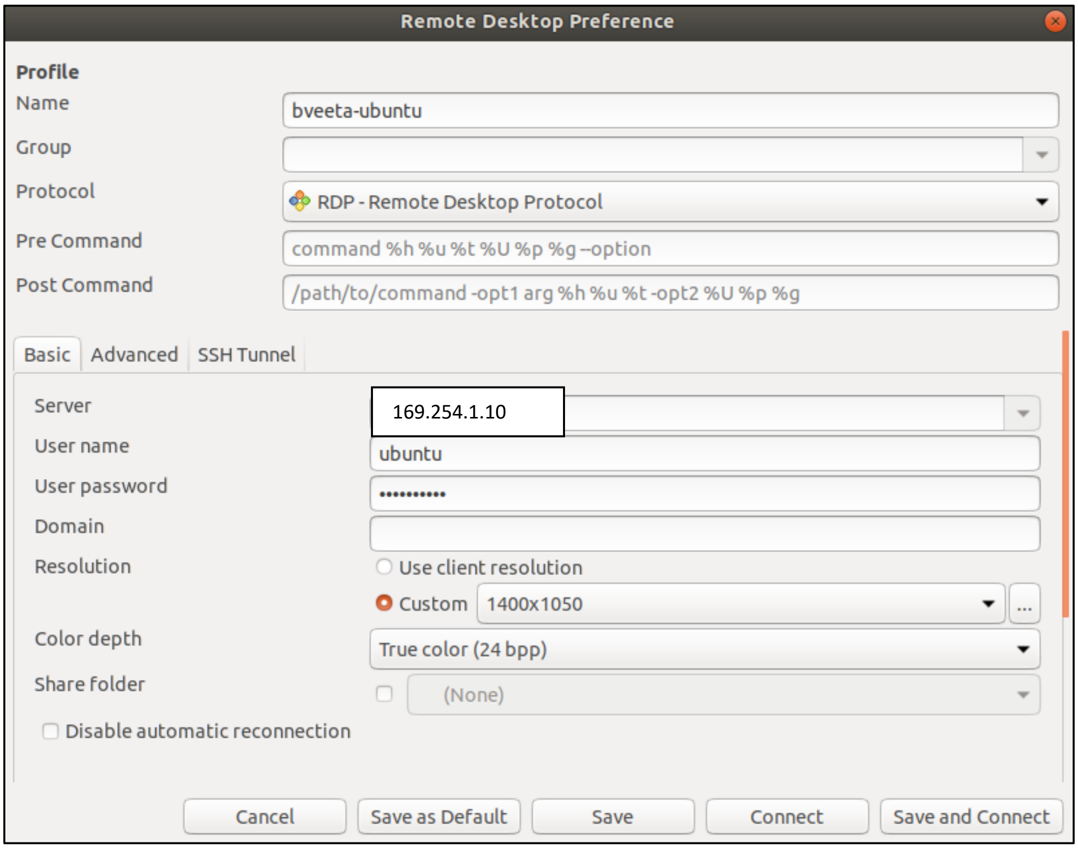
If the connection refused, please double check your Firewall and Anti-Virus settings, and don't forget to enable the Remote Desktop connections settings. More on Remote Desktop settings please refer to Microsoft.com.

10. For the first time user, it is recommended to connect Bveeta mini with your own wireless internet connection either through Wi-Fi or hotspot connectivity. By default, Bveeta mini wifi connection is set to manual **ip address 192.168.1.23.** User may change this ip address accordingly.

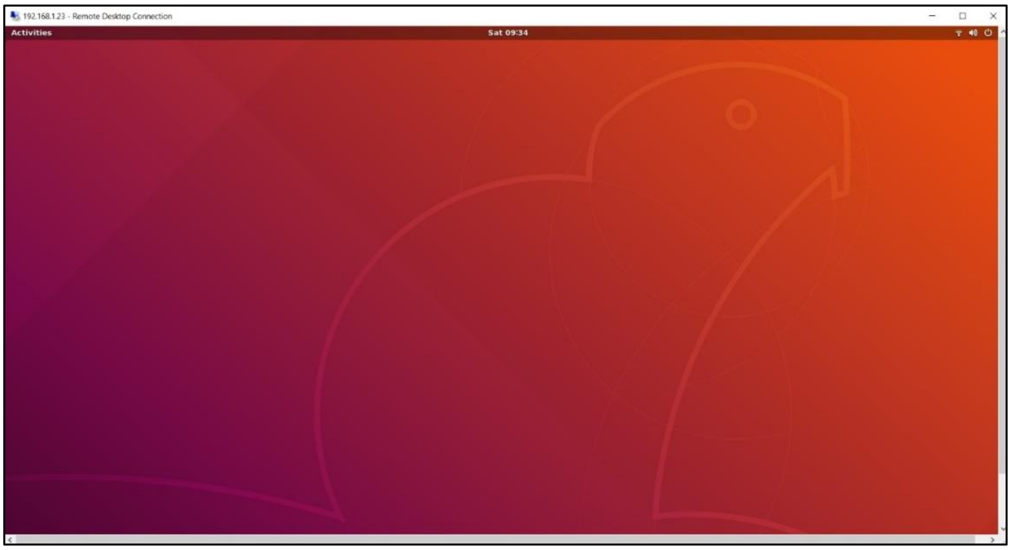
Method 2: Remote desktop connections (Ubuntu 18.04 user)

For UNIX system, Bveeta Mini has been tested successfully on Ubuntu 18.04 version. Please install "Remmina" software from the Ubuntu software online or can manually browse to https://remmina.org/

1. To get start, open Remmina software.
2. Set the properties accordingly:
3. Click connect and key in as follow:
   a. Username: **ubuntu**
   b. Password: **bveeta2021**



4. If everything is fine, you should see same as below:



Bveeta mini program structure

By default Bveeta mini comes with comprehensive example of applications and program such as:

- Autonomous navigation
- Simultaneous Localization and Mapping (SLAM)

Autonomous navigation

Bveeta mini has been program with autonomous navigation capabilities based on DWA local planner, TEB local planner and Trajectory local planner. More details on this local planner can be found in http://ros.org

Simultaneous Localization and Mapping (SLAM)

SLAM is the ability of a robot to map it unknown environment while it keeps tracking its position on a global map. This technique is called localization technique. Bveeta mini has been equipped with few example of SLAM method such as Gmapping SLAM and Hector SLAM. Before going further with SLAM, it is advisable for user to go through the fundamental of ROS and understand how the ***tf*** (transform) in ROS works.

In the next chapter, we going to start with the basic examples before going further in SLAM.

Method 3: Connect Bveeta mini with SSH (ROS MASTER)

ROS is designed with distributed computing in mind. A well-written node makes no assumptions about where in the network it runs, allowing computation to be relocated at run-time to match the available resources (there are exceptions; for example, a driver node that communicate with a piece of hardware must run on the machine to which the hardware is physically connected). Deploying a ROS system across multiple machines is easy. Keep the following things in mind:

You only need one master. Select one machine to run it on. All nodes must be configured to use the same master, via ROS_MASTER_URI. There must be complete, bi-directional connectivity between all pairs of machines, on all ports (see http://wiki.ros.org/ROS/NetworkSetup). Each machine must advertise itself by a name that all other machines can resolve (see ROS/NetworkSetup). Bveeta mini has been tested with full performance based on the SSH connections. Therefore, it is recommended for user to connect the robot with SSH method. To get started please follow steps below:

1. On user computer, make sure it has been installed with ROS. It is recommended to use ROS MELODIC (Ubuntu 18.04) as Bveeta mini has been program based on ROS MELODIC. For installation on ROS MELODIC, user need to have Ubuntu 18.04 on their computer. Another version of ROS also can be use but has not been tested yet.
2. Once your ROS is ready on your PC, now you can follow the next steps
3. On your computer, open new terminal
   Step 1. Do the basic check: ifconfig (get your ip address)
   Step 2. You can check for basic connectivity with ping 192.168.1.10
   Step 3. If you get the response, that means the connections is good.
4. With the same terminal type the following:

```
export ROS_MASTER_URI=http://192.168.1.10:11311  ← Bveeta IP address
export ROS_IP=192.168.1.123  ← Your computer IP address (the last
digit might be different from this example)
```

5. Next, open another terminal and enter SSH

```
ssh ubuntu@192.168.1.10
```

   enter Bveeta password: **bveeta2021**

6. Now you already log into the Bveeta mini.
7. The ROS_MASTER configuration can also be written in ~/.bashrc
   To edit this file, simply enter "$*sudo nano ~/.bashrc*" in the terminal and add these 2 lines below (Please change the ip address according to your access point ip address)

```
export ROS_MASTER_URI=http://192.168.1.10:11311  ← Bveeta IP address
export ROS_IP=192.168.1.123   ← Bveeta ip address
```

8. Close all the terminal and reopen for the changes takes effect.

Note: For the new user in ubuntu environment applying Bveeta connection with ROS_MASTER is not recommended.

Chapter 2

Hardware check and basic runs

2.1 Power up LIDAR

Bveeta mini is equipped with 2D lidar. To power up the lidar, simply connect the usb cable to the Raspi usb port.

Next, open two terminals in the Bveeta mini and type the following:

In the first terminal, type:

roscore

```
ubuntu@ubuntu:~$ roscore
```

and press enter

On the second terminal enter:

roslaunch ydlidar_ros X2L.launch

```
ubuntu@ubuntu:~$ roslaunch ydlidar_ros X2L.launch
```

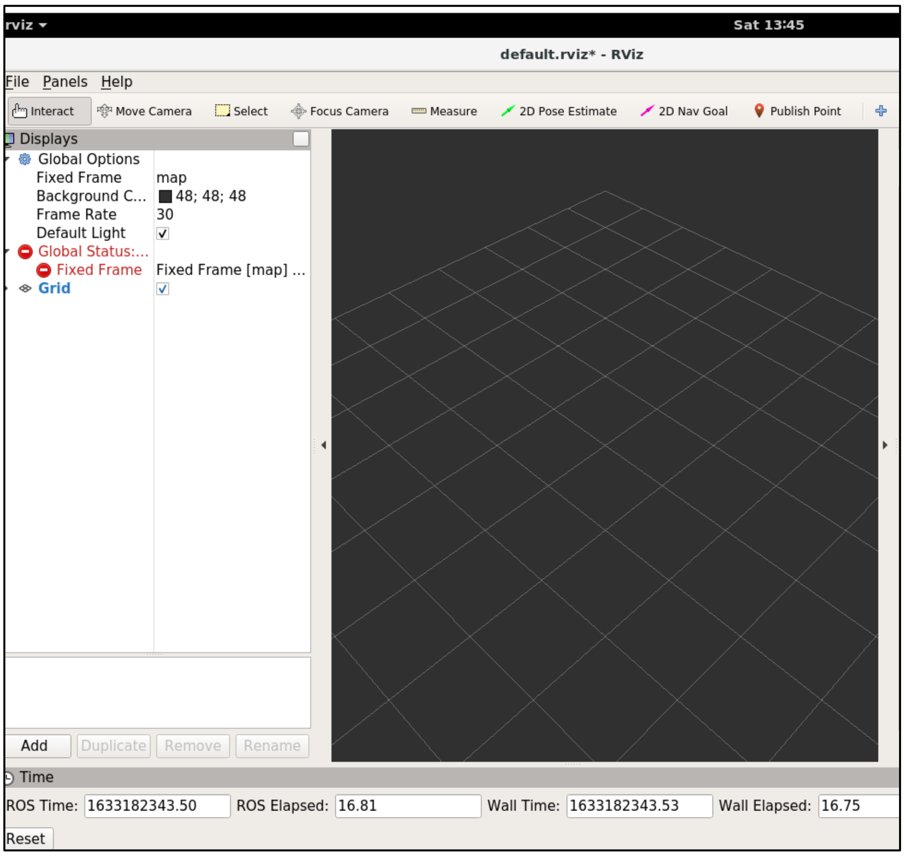
You should see the output as follow:

```
SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.11
 * /ydlidar_node/angle_max: 180.0
 * /ydlidar_node/angle_min: -180.0
 * /ydlidar_node/auto_reconnect: True
 * /ydlidar_node/baudrate: 115200
 * /ydlidar_node/frame_id: scan
 * /ydlidar_node/frequency: 7.0
 * /ydlidar_node/ignore_array: 0.012
 * /ydlidar_node/isSingleChannel: True
 * /ydlidar_node/port: /dev/ttyUSB0
 * /ydlidar_node/range_max: 8.0
 * /ydlidar_node/range_min: 0.1
 * /ydlidar_node/resolution_fixed: True
 * /ydlidar_node/reversion: False
 * /ydlidar_node/samp_rate: 3

NODES
  /
    camera_link (tf/static_transform_publisher)
    scan (tf/static_transform_publisher)
    ydlidar_node (ydlidar_ros/ydlidar_node)

ROS_MASTER_URI=http://192.168.1.23:11311

process[ydlidar_node-1]: started with pid [13963]

process[scan-2]: started with pid [13964]
process[camera_link-3]: started with pid [13970]
[ERROR] [1633182070.676644190]: ignore array is odd need be even
[ INFO] [1633182070.705198920]: [YDLIDAR INFO] Now YDLIDAR ROS SDK VERSION:1.4.6 .......
YDLidar SDK initializing
YDLidar SDK has been initialized
[YDLIDAR]:SDK Version: 1.4.7
LiDAR successfully connected
[YDLIDAR]:Lidar running correctly ! The health status: good
LiDAR init success!
[YDLIDAR]:Fixed Size: 500
[YDLIDAR]:Sample Rate: 4K
[YDLIDAR INFO] Current Sampling Rate : 4K
[YDLIDAR INFO] Now YDLIDAR is scanning ......
```

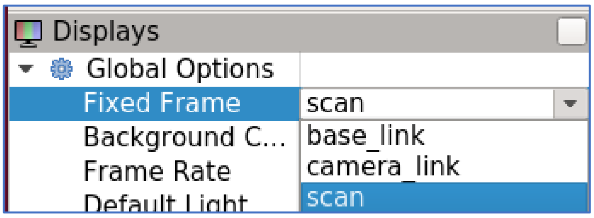
To visualize the Lidar output, user can use Rviz simulation tools to display the environment. Open another terminal and enter the following:

```
ubuntu@ubuntu:~$ rosrun rviz rviz
```

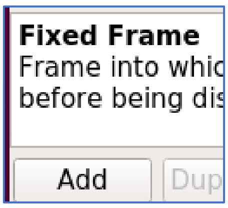
You should see the Rviz windows as follow:



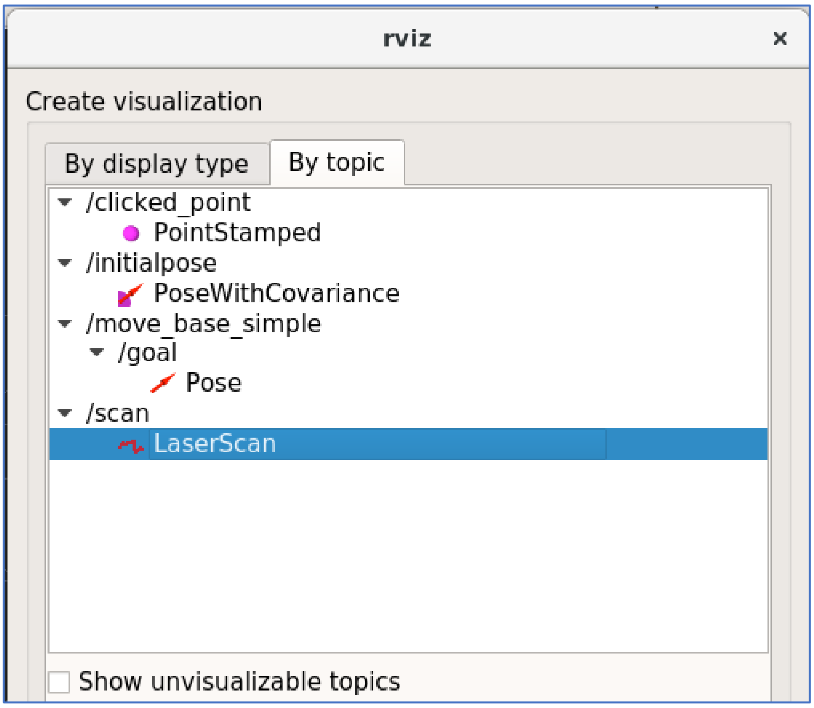
In the display panel, click the dropdown button and choose scan.



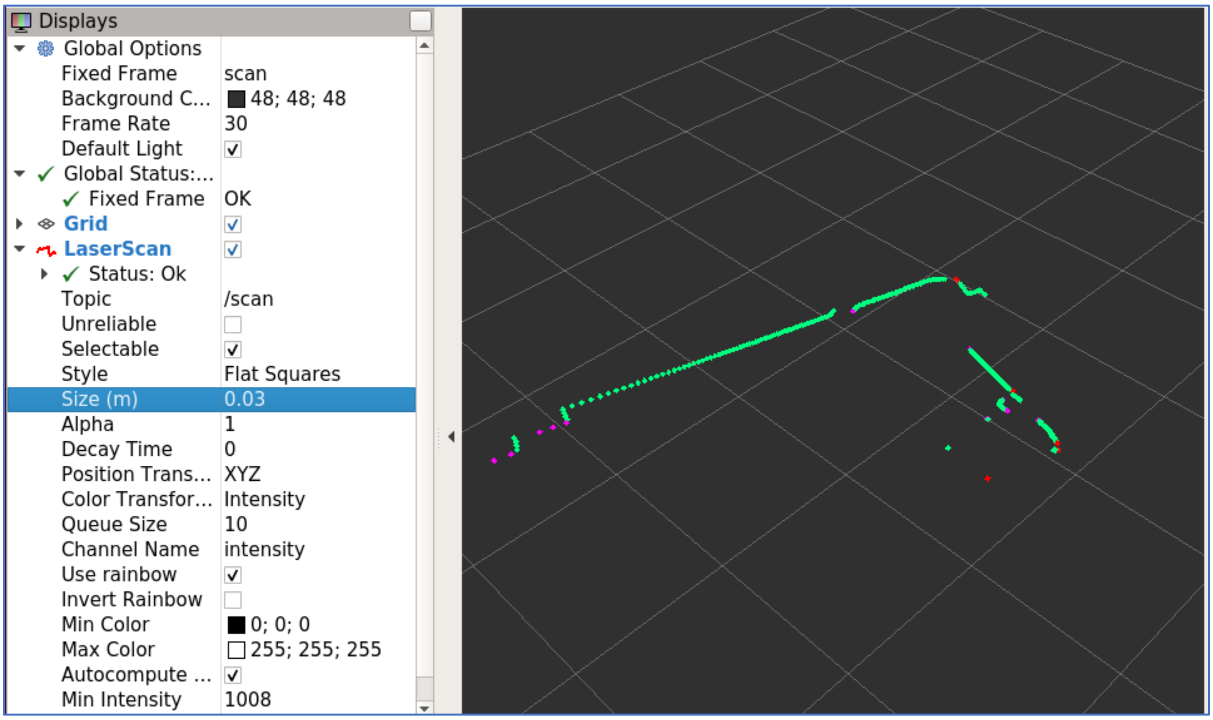
To display the laser point in the Rviz, Click the Add button below.



In the Create visualization windows, select *by topic*, and select the *laser scan*.



You should see the laser output in the Rviz as shown in figure below:



Change the Size (m) to 0.03 so you can see the laser dot clearly. You can play around with the settings also. Congratulations on your first start with ROS! The next step we going to control Bveeta mini motion and movement using keyboard.

2.2 Bveeta motion control with teleop twist keyboard

Teleop twist keyboard is one of the method for user to control the Bveeta mini manually using the computer keyboard. The twist msg that is produced by the teleop_twist_keyboard will send to the Arduino MEGA to drive the robot forward, backward, turns left or right and with adjustable speed through the Bveeta Mini SDD® controller board. Figure below shows the TF tree and Node graph for teleop_twist_keyboard for Bveeta mini.

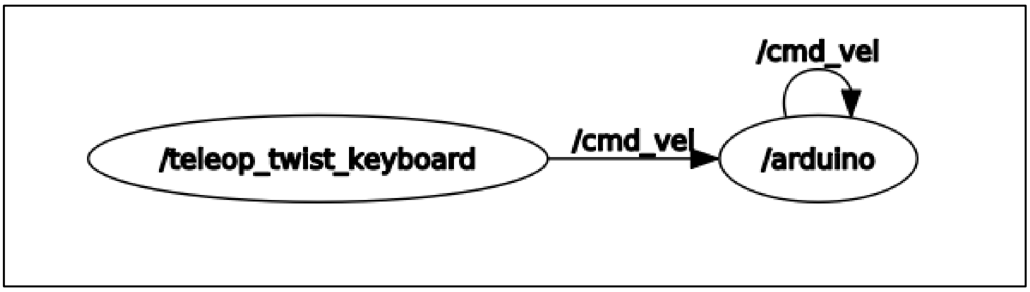


Figure: Node graph for teleop_twist_keyboard msg.

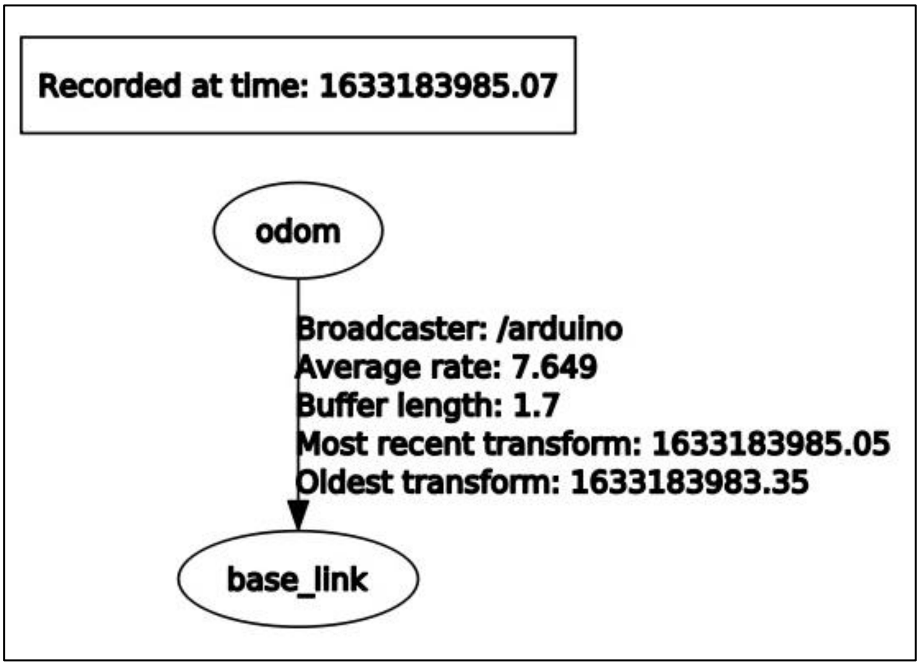
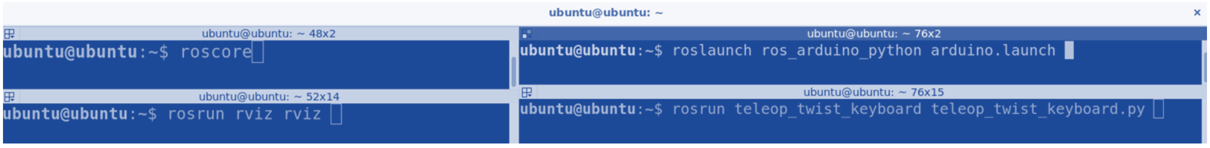


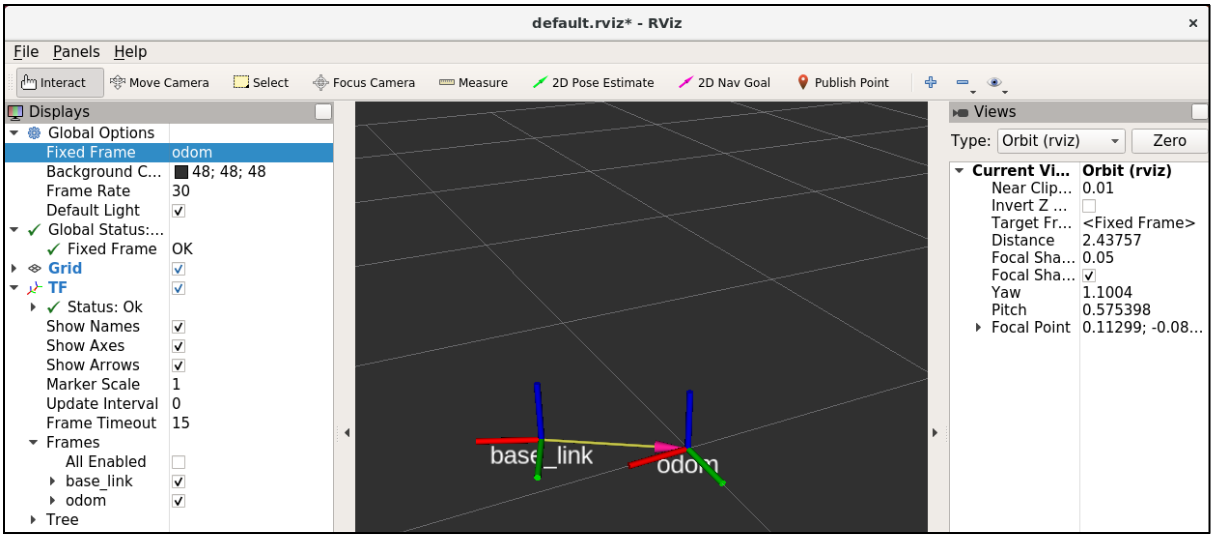
Figure: TF tree for Bveeta mini with Odom and base_link.

From figure above, you can see that the Arduino (MEGA) is subscribe to the topic /cmd_vel published by the teleop_twist_keyboard. This topic is then processed by the Arduino MEGA and translate it to linear motion based on the odometry (odom) input coming from the motor encoder. The odom is connected to the base_link which in this case is the Bveeta Mini. The Arduino MEGA used for Bveeta mini is hardcoded with the PID control so it can process the encoder and produce the output properly. The odom topic is published by the Arduino MEGA.

To use the teleop twist, open 3 new terminals. Type in the terminal as shown in figure below:



In the Rviz windows, add the tf in the menu and change the fixed frame to Odom. See figure below:



To move your robot with your keyboard, follow the table given below:

Key	Function	Key	Function	Key	Function
I	Move forward	W	Increase linear speed	U	Circular motion CCW
,	Move backward	X	Decrease linear speed	M	Circular motion CW reverse
L	Rotate CW direction	Q	Increase angular speed	.	Circular motion CCW reverse
J	Rotate CCW direction	Z	Decrease angular speed		
K	Stop robot	O	Circular motion CW		

Now that's complete the basic part of Bveeta mini. The next chapter we going to build a map that Bveeta mini can navigate later. Congratulations.

Chapter 3

Simultaneous Localization and Mapping (SLAM)

3.1 Simultaneous localization and Mapping (SLAM)

Simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. While this initially appears to be a chicken-and-egg problem there are several algorithms known for solving it, at least approximately, in tractable time for certain environments. Popular approximate solution methods include the particle filter, extended Kalman filter, covariance intersection, and GraphSLAM. SLAM algorithms are based on concepts in computational geometry and computer vision, and are used in robot navigation, robotic mapping and odometry for virtual reality or augmented reality (Wikipedia).

3.2 SLAM with Bveeta Mini

There are two methods used by Bveeta mini to do SLAM.

a. Hector SLAM
b. Gmapping SLAM

For more information details on the above, please visit https://ros.org

3.3 Bveeta mini with Hector SLAM

To run Hector SLAM on Bveeta mini, please follow steps below:

1. Open new 4 terminals
2. First terminal, enter:

```
roscore
```

3. Second terminal enter:

```
roslaunch hectormapping bveeta_hector_slam.launch
```

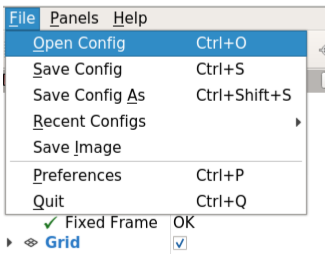
4. Third terminal enter:

```
rosrun rviz rviz
```

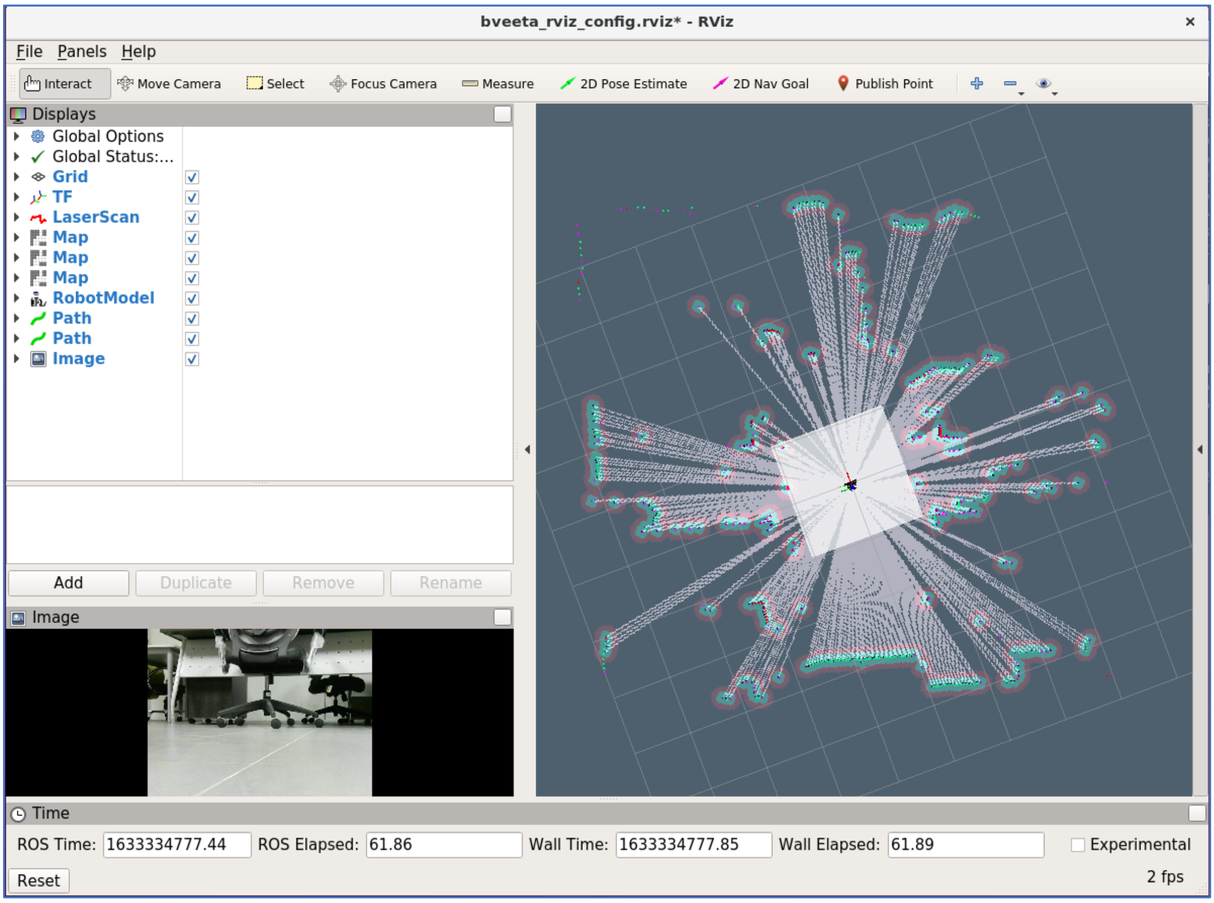
5. 4th terminal enter:

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

6. In the Rviz window, open the config files name in Home directory > *"bveeta_rviz_config.rviz"*



7. The Rviz layout should appear like Figure below:



8. Set your robot angular and linear speed to low by pressing X or Z key few times. The reason is when doing SLAM, it requires high capacity of the SBC RAM which will slow down the calculation and processing power.
9. Runs your robot until it creates enough map features for your requirement.
10. Once the map is satisfied, save the map created so that it can be use for navigation later. To save the map, open new terminal, and enter the following:

```
rosrun map_server map_saver -f myMap
```

Warning!

When doing mapping with Hector SLAM through the remote Desktop connection, you will face an issue where the map generated is very bad. This is due to the lack capacity of Graphical processor Memory and limited resolutions of Lidar. Please consider connecting your Bveeta mini through the ROS MASTER so it can perform well for Hector SLAM.

3.4 Bveeta mini with Gmapping SLAM

To run Gmapping SLAM on Bveeta mini, please follow steps below:

1. Open new 4 terminals
2. First terminal, enter roscore

```
roscore
```

3. Second terminal enter the following:

```
roslaunch gmapping bveeta_gmapping.launch
```

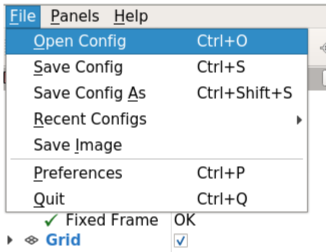
4. Third terminal enter

```
rosrun rviz rviz
```

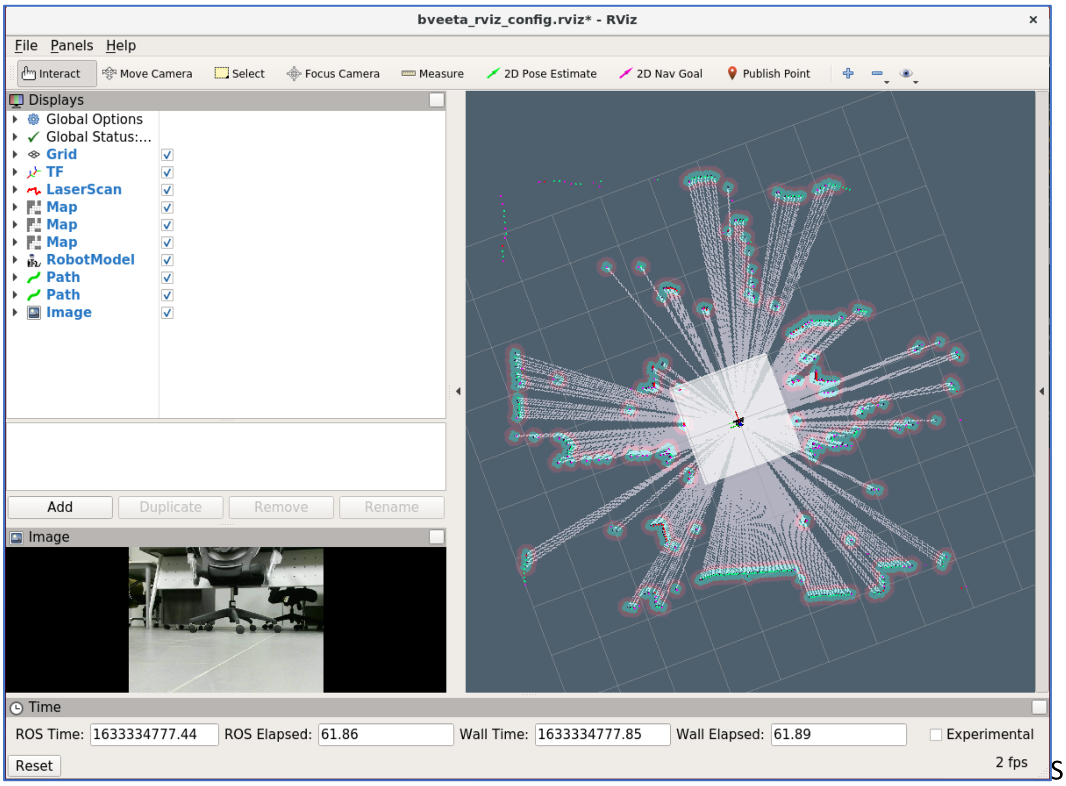
5. 4th terminal enter

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

6. In the Rviz window, open the config files name in Home directory > "*bveeta_rviz_config.rviz*"



7. The Rviz layout should appear like Figure below:



8. Set your robot angular and linear speed to low by pressing X or Z key few times. The reason is when doing SLAM, it requires high capacity of the SBC RAM which will slow down the calculation and processing power.
9. Runs your robot until it creates enough map features for your requirement.
10. Once the map is satisfied, save the map created so that it can be use for navigation later. To save the map, open new terminal, and enter the following:

```
rosrun map_server map_saver -f myMap
```

Chapter 4

Autonomous navigation

4.1 Autonomous navigatin with Bveeta mini mobile robot

A navigation system is a program that provides graphical maps, coordinates or directions to a destination. Now autonomous navigation system of a mobile robot involves self-steering of a robot from one place to another based on computational resources on-board the robot. Bveeta mini has been programmed with 3 different of autonomous navigation algorithm which are:

1. TEB local planner
2. DWA local planner
3. Base local planner

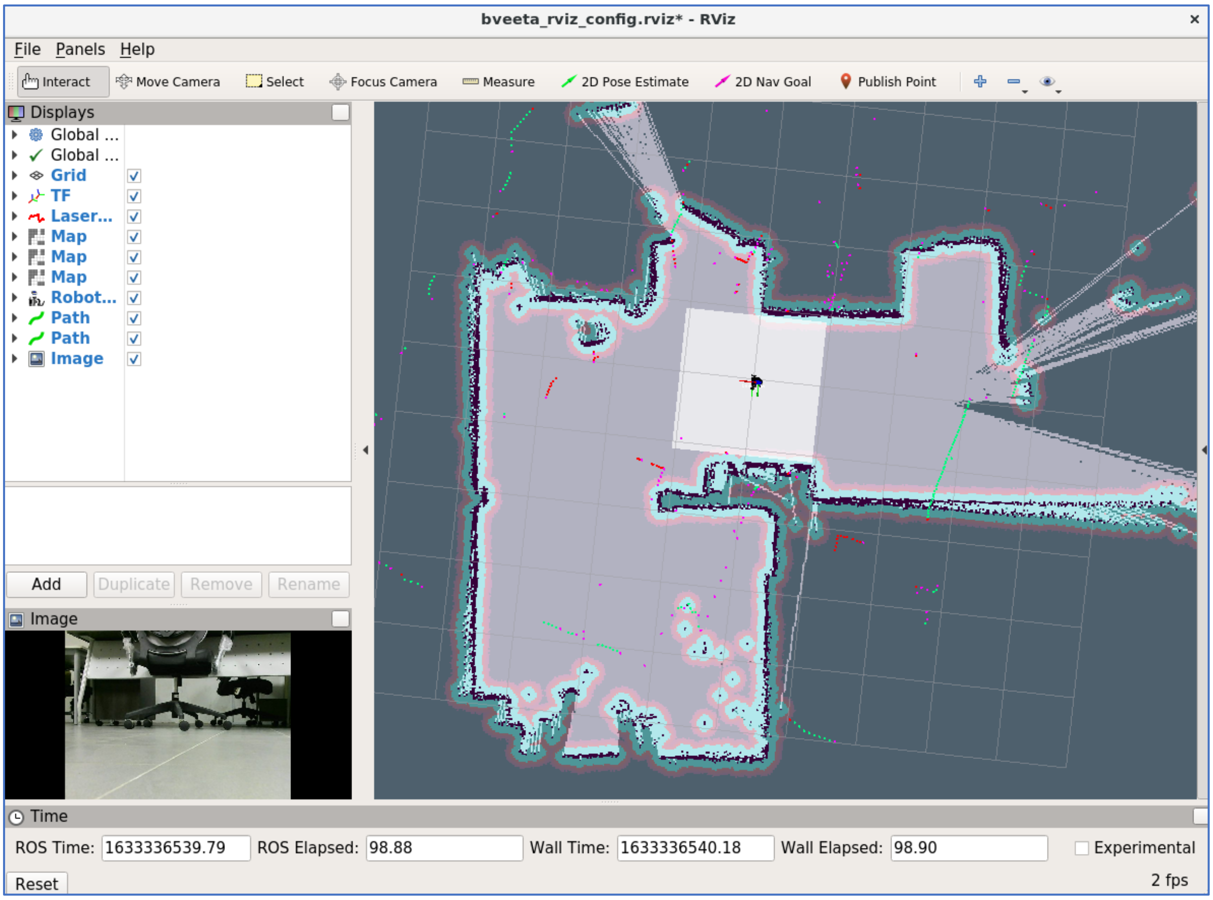
4.2 TEB local planner autonomous navigation system

TEB local planner implements an online optimal local trajectory planner for navigation and control of mobile robots. The local planner is able to switch to the current globally optimal trajectory among the candidate set. Distinctive topologies are obtained by utilizing the concept of homology / homotropy classes. More details on TEB local planner can be found at http://wiki.ros.org/teb_local_planner

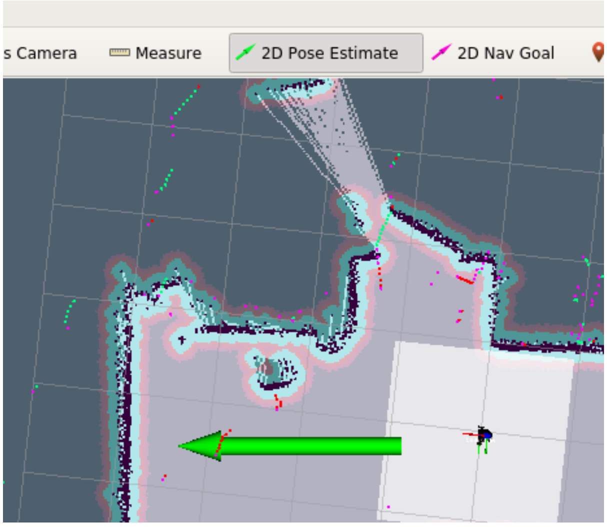
To run TEB local planner navigation, the previous map generated based on SLAM method must be use. To get started, please follow the steps below:

1. Open new 4 terminals
2. First terminal, enter:
   ```
   roscore
   ```
3. Second terminal enter:
   ```
   rosrun map_server map_server myMap
   ```
4. Third terminal enter:
   ```
   rosrun rviz rviz
   ```
5. 4th terminal enter:
   ```
   roslaunch bveeta_navigation bveeta_teb_local_planner.launch
   ```
6. In the Rviz window, open the config files name in Home directory > *"bveeta_rviz_config.rviz"*

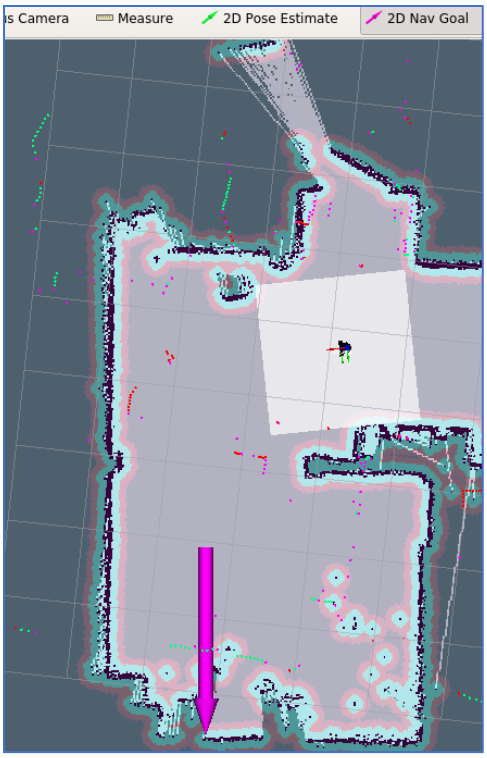
7. The Rviz layout should appear as shown in the Figure below:



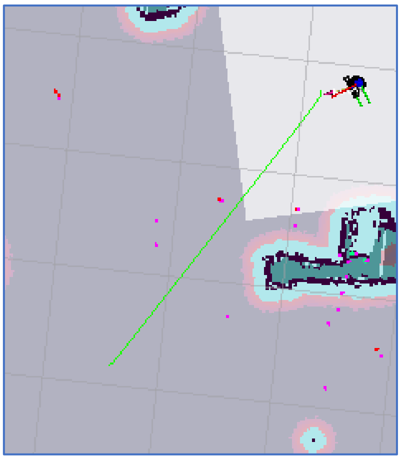
8. To move your robot, set the position of the robot on the map so that the laser dots is aligned closely with the pre generated map. To set the position click the 2D Pose Estimate button and locate the robot on the map according its current position. This will help to localize the robot correctly before its navigate to its goal.



9. Once you satisfied with the current position on the map, next you can start to point its goal. To perform this, click the  button and click anywhere on the map or the position you want the robot to go.



10. You will see the green line showing the global path that drives the robot to its goal as shown in Figure below. Once the goal is reached, the robot will stop and wait for another instruction from user. Congratulations on your first autonomous navigation with the Bveeta mini.



4.3 Dynamic windows approach (DWA) local planner autonomous navigation system

This package provides an implementation of the Dynamic Window Approach to local robot navigation on a plane. Given a global plan to follow and a costmap, the local planner produces velocity commands to send to a mobile base. This package supports any robot who's footprint can be represented as a convex polygon or cicrle, and exposes its configuration as ROS parameters that can be set in a launch file. The parameters for this planner are also dynamically reconfigurable. This package's ROS wrapper adheres to the BaseLocalPlanner interface specified in the nav_core package. More details on TEB local planner can be found at http://wiki.ros.org/dwa_local_planner

To run DWA local planner navigation, the previous map generated based on SLAM method must be use. To get started, please follow the steps below:

1. Open new 4 terminals
2. First terminal, enter:

```
roscore
```

3. Second terminal enter:

```
rosrun map_server map_server myMap
```

4. Third terminal enter:

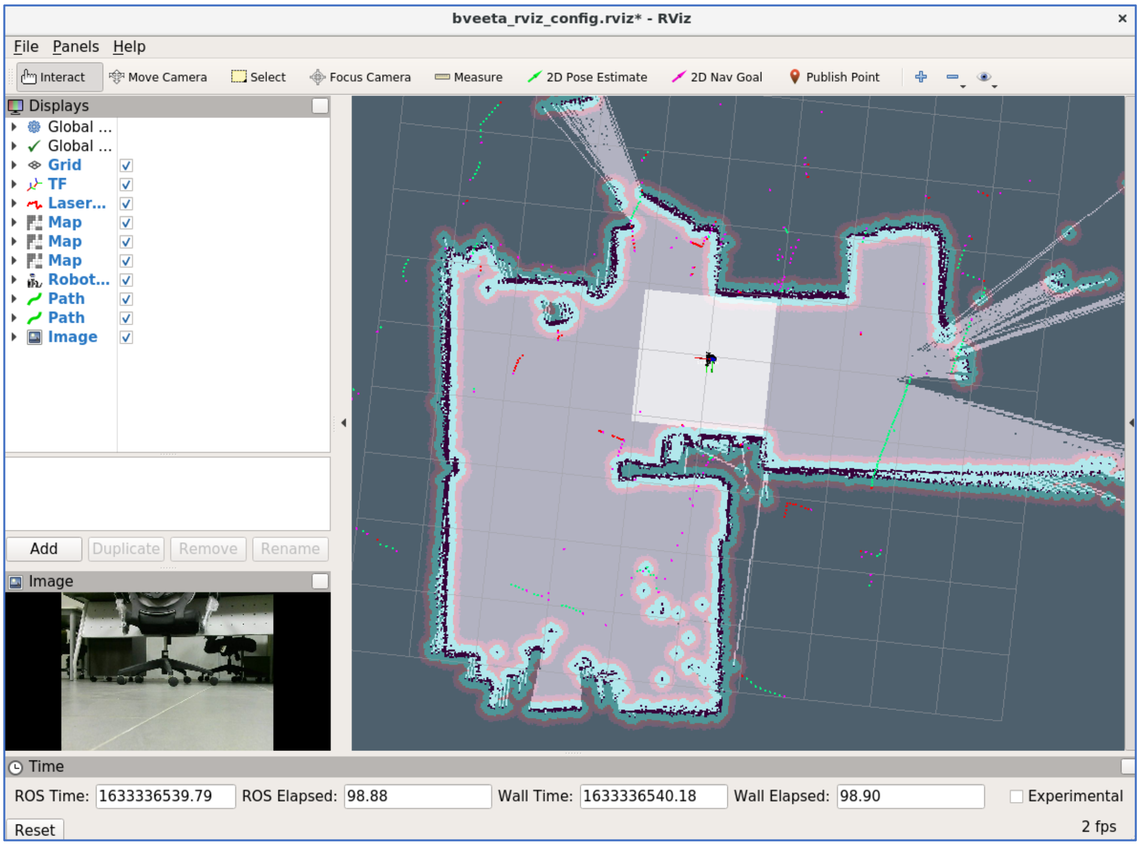
```
rosrun rviz rviz
```

5. 4th terminal enter:

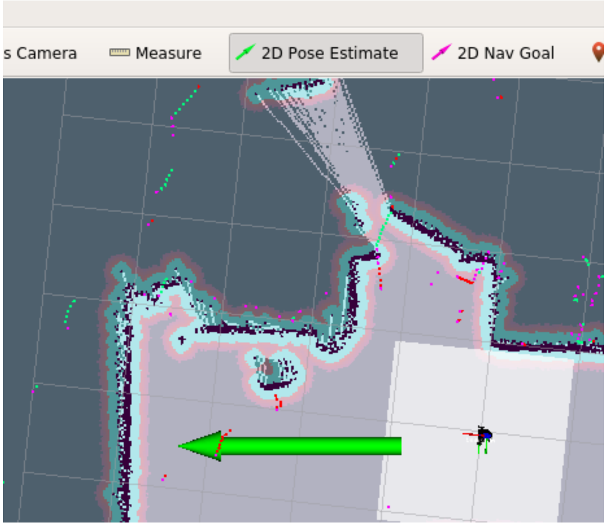
```
roslaunch bveeta_navigation bveeta_dwa_local_planner.launch
```

6. In the Rviz window, open the config files name in Home directory > *"bveeta_rviz_config.rviz"*

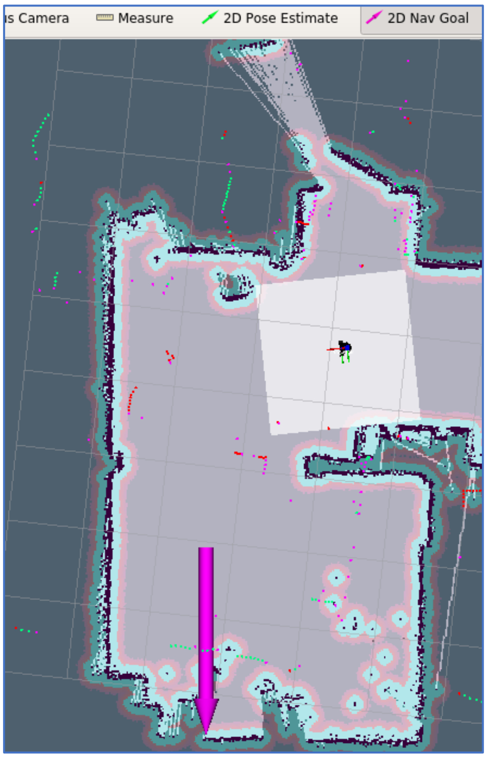
7. The Rviz layout should appear like Figure below:



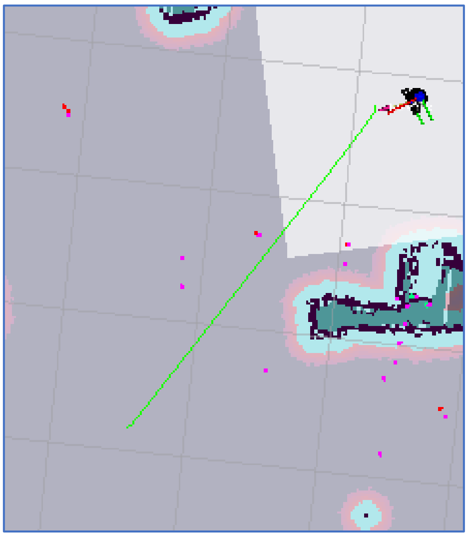
8. To move your robot, set the position of the robot on the map so that the laser dots is aligned closely with the pre generated map. To set the position click the 2D Pose Estimate button and locate the robot on the map according its current position. This will help to localize the robot correctly before its navigate to its goal.



9. Once you satisfied with the current position on the map, next you can start to point its goal. To perform this, click the  button and click anywhere on the map or the position you want the robot to go.



10. You will see the green line showing the global path that drives the robot to its goal as shown in Figure below. Once the goal is reached, the robot will stop and wait for another instruction from user. Congratulations on your DWA local planner autonomous navigation with the Bveeta mini.



4.4 Trajectory local planner (Known as Base local planner) autonomous navigation system

The trajectory local planner also known as base_local_planner package provides a controller that drives a mobile base in the plane. This controller serves to connect the path planner to the robot. Using a map, the planner creates a kinematic trajectory for the robot to get from a start to a goal location. Along the way, the planner creates, at least locally around the robot, a value function, represented as a grid map. This value function encodes the costs of traversing through the grid cells. The controller's job is to use this value function to determine dx, dy, dθ velocities to send to the robot.

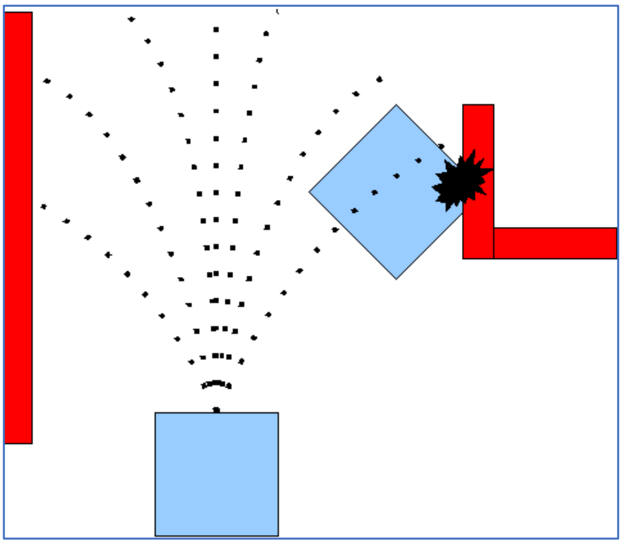


Image source: http://ros.org

More details on trajectory local planner can be found here
http://wiki.ros.org/dwa_local_planner

To run DWA local planner navigation, the previous map generated based on SLAM method must be use. To get started, please follow the steps below:

1. Open new 4 terminals
2. First terminal, enter:

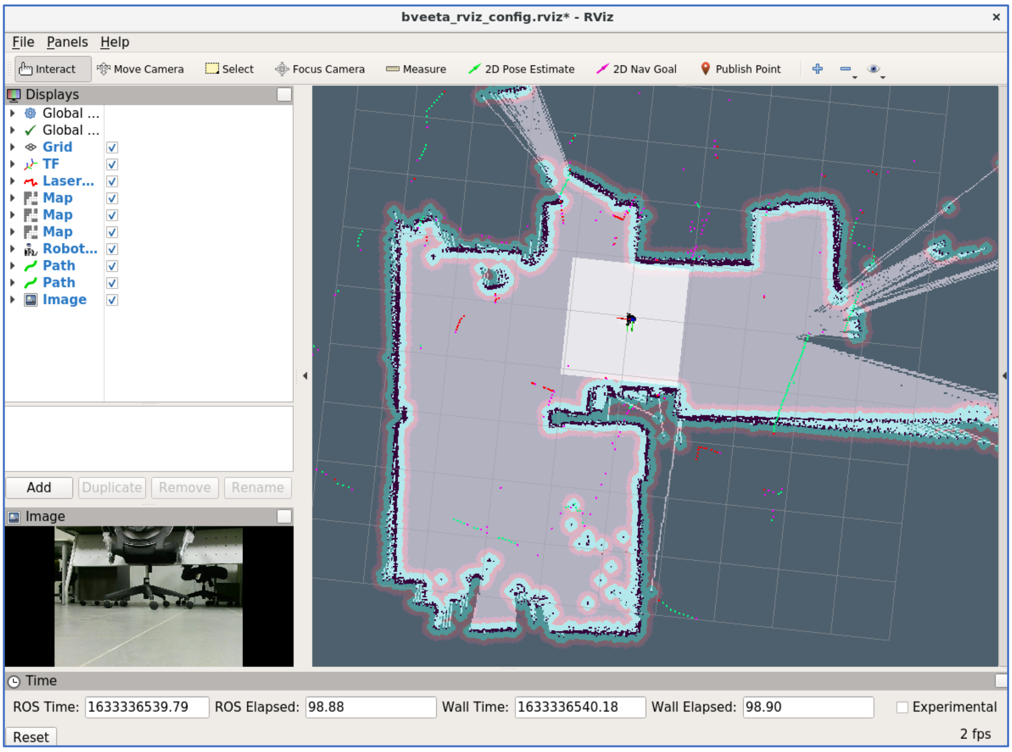
   ```
   roscore
   ```
3. Second terminal enter:

   ```
   rosrun map_server map_server myMap
   ```
4. Third terminal enter:

   ```
   rosrun rviz rviz
   ```
5. 4th terminal enter:

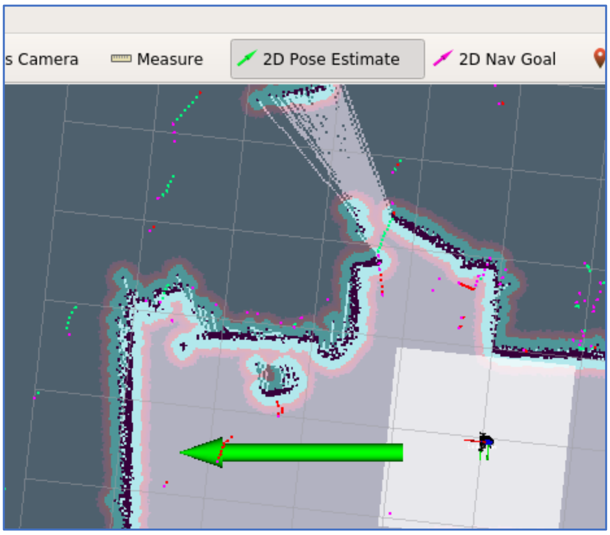
   ```
   roslaunch bveeta_navigation bveeta_trajectory_local_planner.launch
   ```
6. In the Rviz window, open the config files name in Home directory > "*bveeta_rviz_config.rviz*"

7. The Rviz layout should appear like Figure below:



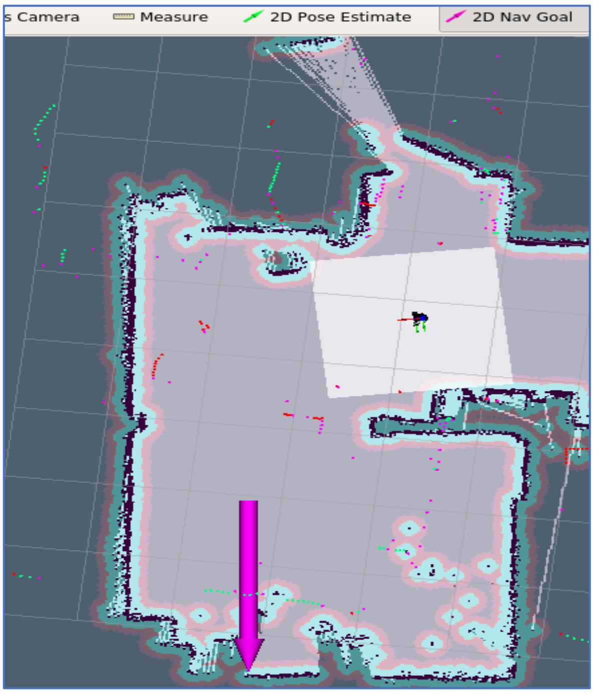
8. To move your robot, set the position of the robot on the map so that the laser dots is aligned closely with the pre generated map. To set the position click the 

button and locate the robot on the map according its current position. This will help to localize the robot correctly before its navigate to its goal.

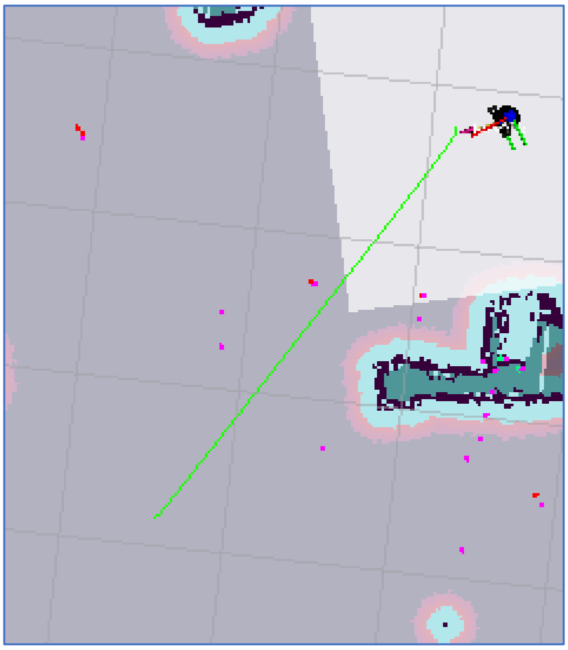


9. Once you satisfied with the current position on the map, next you can start to point its goal. To perform this, click the 

button and click anywhere on the map or the position you want the robot to go.



10. You will see the green line showing the global path that drives the robot to its goal as shown in Figure below. Once the goal is reached, the robot will stop and wait for another instruction from user. Congratulations on your trajectory local planner autonomous navigation with the Bveeta mini.



Bizbot Technology(
14 Jalan 6/6C, Seksyen 6, Bandar Rinching,
43500 Semenyih, Selangor
Malaysia
https://bzonemy101.wixsite.com/mysite
c4robotic@gmail.com