**MEI-56306 : MIR**

# **Eelis and Romaric on 09.10.2018**

## Connecting to the MIR

Turn MiR on from the blue button (right button in corner).

Connect to MiR’s Wifi, password is **mirex4you**.

After connecting to the MiR’s Wifi, launch the command **ifconfig** in your terminal to see your IP address (should be **http://192.168.56.1/24**)

The IP address of the MIR robot is **192.168.12.20**

Connect to the address 192.168.12.20 in your browser, the identifiers are the following :

* Login name : **Distributor**
* Password : **distributor**

**Always** export your and the MiR’s IP’s when opening a new terminal (fix for this would be nice)

* The command for the master URI is **export ROS\_MASTER\_URI=http://192.168.12.20:11311**.
* The command for exporting your IP to ROS is **export ROS\_IP=192.168.12.246**.

**export ROS\_IP=192.168.12.253**

* + **the one got by ifconfig**

sudo apt-get update && sudo apt-get install --only-upgrade ros-kinetic\*

## Moving MiR with cmd\_vel

The topic for moving the robot is **cmd\_vel**

* Type is **geometry\_msgs/Twist**
* **rostopic echo /cmd\_vel** will show you the cartesian linear and angular velocities – move the robot from the browser in manual mode to see changes in velocity
* Publish with a rate of **at least 11 Hz**, which is MiR’s “refresh rate”
* Move forward and back with **xlin**, turn with **zan**g. We have tried x= +- 0.05 and z = +- 1.2 at maximum, 0.1 seems like a reasonable, slow speed for both
* Turn off manual control in the browser when publishing with cmd\_vel. The browser sends 0 the whole time if it is on, resulting in jerky motion
* Before publishing this command, be sure that you can move the robot without danger of collision!
* rostopic pub /cmd\_vel geometry\_msgs/Twist -r 11 -- '[1,0,0]' '[0,0,0.1]'
  + Only effective X and W\_z

## Install

Install rqt ez publisher in the terminal with **sudo apt-get install ros-kinetic-rqt-ez-publisher**

* Allows for simple publishing
* After installing, launch the GUI with command **rqt**
* The publisher is found in plugins->topics->message publisher

## Publish information

The command for testing the status of the publisher is **rostopic info /cmd\_vel**

## Realsense camera

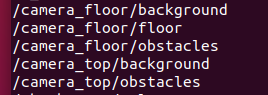
<https://github.com/intel-ros/realsense>

<http://wiki.ros.org/RealSense>

* ROS rqt\_graph image: move\_base\_node.png

15.10.:

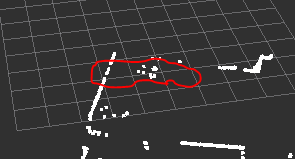
* All camera topics include message of type “PointCloud2”
* There are topics for “obstacles” and “background” for both top and floor cameras, and topic “floor” for camera\_floor



* **17.10:** Camera depth range is +20cm to +10m, but seems to end at about 3m in the lab.

## Laser scanner

* actually quite good range and performance (/b\_raw\_scan in the image)



in the image, the dots in the middle of red area are my legs blocking the view of the wall on the left side of the red area. I was about 3,5 meters from the robot.

## SLAM

<https://cartographer-mir.readthedocs.io/en/latest/>

<https://github.com/googlecartographer/cartographer_mir>

Eelis and Romaric on 13.10.2018

## Working with .py files in ROS

Inside the ROS workspace, code is stored in packages. They must be in the src folder in the workspace (catkin\_ws/src).

1. Go to the folder catkin\_ws/src
2. Make a package with the terminal command **catkin\_create\_pkg [package\_name] [dependencies]**, where dependencies are the (ROS?) libraries you need to import (for example: catkin\_create\_pkg test\_package rospy geometry\_msgs)
3. Go to the **catkin\_ws folder** and build with **catkin\_make**
4. Source built package with **source ~/catkin\_ws/devel/setup.bash**
5. Write .py files
6. Make each .py file executable with **chmod +x [file\_name.py]**
7. Run file with **rosrun [package\_name] [file\_name.py]**

**Note:** In ROS, one file = one node. Multiple subscribers and publishers can be in one file, but the file is only one node.

# Notes on orientation and moving

* **Coordinate system**:
  + +x direction is from charging station toward Franka Panda
  + +y direction is from outer windows toward inner windows
  + +z direction is (presumably) up
  + origin might be the charging station
  + W in odom topic might be cosine between MiR pose and the charging station pose
* Odom and odom\_enc are similar. Odom cannot be published to
* Move\_base\_node/current\_goal is empty when moving manually
* To save data in ROS: **rosbag record [topic\_name]**
  + MiR has a topic which has all nodes as a publisher, for easily recording all data from MiR

## Modify your bash file

It is possible to modify your bash file to don’t have to type the export commands everytime you connect to the MiR. To do it, open a terminal (CTRL+ALT+T) and type the command :

**sudo nano ~/.bashrc**

Then, the bash file is shown, you just have add the following lines in the end of the file, please change the IP address corresponding to your IP address obtained with the command **ifconfig** :

source /opt/ros/indigo/setup.bash

**source ~/catkin\_ws/devel/setup.bash**

**#ROS config for MiR**

**SSID=$(iwgetid -r)**

**if [[ $SSID = MiR\_R853 ]]; then**

**export ROS\_MASTER\_URI=http://192.168.12.20:11311**

**export ROS\_IP=192.168.12.252**

**fi**

## Using rosbag

**rosbag record [topic\_name(s)]** will subscribe to topics and record them into a .bag file. Afterwards, these topics can be played with **rosbag play [bag\_name.bag]**. When playing, they appear as if they were being subscribed to at that moment. For example, visualization is possible with rviz as usual.

File *2018-10-23-14-41-42\_subset\_of\_all\_msgs.bag* contains many topics published by MiR. MiR’s movement while recording is shown in *VID\_20181023\_144236.mp4* (times don’t match exactly)*.* To see which topics were recorded, do **rosbag info [bag\_name.bag]**. Some topics might show lag, not sure if it’s avoidable. At least camera color image is very laggy.

Without launcher

/camera\_floor/background

/camera\_floor/camera\_info

/camera\_floor/driver/color/camera\_info

/camera\_floor/driver/color/image\_raw

/camera\_floor/driver/color/image\_raw/compressed

/camera\_floor/driver/color/image\_raw/compressed/parameter\_descriptions

/camera\_floor/driver/color/image\_raw/compressed/parameter\_updates

/camera\_floor/driver/depth/camera\_info

/camera\_floor/driver/depth/color/points

/camera\_floor/driver/depth/image\_rect\_raw

/camera\_floor/driver/depth/image\_rect\_raw/compressed

/camera\_floor/driver/depth/image\_rect\_raw/compressed/parameter\_descriptions

/camera\_floor/driver/depth/image\_rect\_raw/compressed/parameter\_updates

/camera\_floor/driver/extrinsics/depth\_to\_color

/camera\_floor/driver/parameter\_descriptions

/camera\_floor/driver/parameter\_updates

/camera\_floor/filter/parameter\_descriptions

/camera\_floor/filter/parameter\_updates

/camera\_floor/filter/visualization\_marker

/camera\_floor/floor

/camera\_floor/obstacles

/camera\_floor/transform/parameter\_descriptions

/camera\_floor/transform/parameter\_updates

/camera\_top/background

/camera\_top/obstacles

Add to .bashrc :

source ~/catkin\_ws/devel/setup.bash

Run mir\_driver launcher:

roslaunch mir\_driver mir.launch disable\_map:=true

rosrun gmapping slam\_gmapping scan:=/robot0/laser\_0 \_base\_frame:="/robot0" map:=/gmapping/map

rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py

roscd …; rosrun map\_server map\_saver map:=/map\_2 -f name\_of\_the\_map

PEGAS SOLUCIONADAS

TF OLD DATA

rosrun tf tf\_monitor

A la hora de sincronizar, tarda 1 seg en enviar la señal al robot el click

Cada vez que se mueva el robot, resetear mapa config map de la web interface al reiniciar algoritmo

En costmap\_common\_params.yaml:

footprint: [[0.506,-0.32],[0.506,0.32],[-0.454,0.32],[-0.454,-0.32]]

footprint\_padding: 0.0

inflation\_radius: 0.55

1º Actualizar hora (la frecuencia de reloj del robot es mas rapida que la realidad)

2º Poner mapa Configuration map desde la interfaz web

3º roslaunch mir\_driver mir.launch

4º roslaunch rrt\_exploration gmapping\_christen.launch

5º roslaunch rrt\_exploration simple.launch

# ROSTOPIC LIST

/LightCtrl/us\_list

/MC/currents

/MC/encoders

/MissionController/CheckArea/visualization\_marker

/MissionController/prompt\_user

/SickPLC/parameter\_descriptions

/SickPLC/parameter\_updates

/amcl\_pose

/b\_raw\_scan

/b\_scan

/camera\_floor/background

/camera\_floor/floor

/camera\_floor/obstacles

/camera\_top/background

/camera\_top/obstacles

/check\_area/polygon

/cmd\_vel

/diagnostics

/diagnostics\_agg

/diagnostics\_toplevel\_state

/event\_cmd

/f\_raw\_scan

/f\_scan

/hook/status

/imu\_data

/initialpose

/joystick\_vel

/laser\_back/driver/parameter\_descriptions

/laser\_back/driver/parameter\_updates

/laser\_back/transform/parameter\_descriptions

/laser\_back/transform/parameter\_updates

/laser\_front/driver/parameter\_descriptions

/laser\_front/driver/parameter\_updates

/laser\_front/transform/parameter\_descriptions

/laser\_front/transform/parameter\_updates

/light\_cmd

/map

/map\_metadata

/mir\_amcl/parameter\_descriptions

/mir\_amcl/parameter\_updates

/mir\_amcl/selected\_points

/mir\_cmd

/mir\_log

/mir\_sound

/mir\_status\_msg

/mirspawn/node\_events

/mirwebapp/grid\_map\_metadata

/mirwebapp/laser\_map\_metadata

/mirwebapp/web\_path

/move\_base/cancel

/move\_base/feedback

/move\_base/goal

/move\_base/result

/move\_base/status

/move\_base\_node/MIRPlannerROS/cost\_cloud

/move\_base\_node/MIRPlannerROS/global\_plan

/move\_base\_node/MIRPlannerROS/len\_to\_goal

/move\_base\_node/MIRPlannerROS/local\_plan

/move\_base\_node/MIRPlannerROS/parameter\_descriptions

/move\_base\_node/MIRPlannerROS/parameter\_updates

/move\_base\_node/MIRPlannerROS/visualization\_marker

/move\_base\_node/SBPLLatticePlanner/plan

/move\_base\_node/SBPLLatticePlanner/sbpl\_lattice\_planner\_stats

/move\_base\_node/SBPLLatticePlanner/visualization\_marker

/move\_base\_node/current\_goal

/move\_base\_node/global\_costmap/forbidden\_area

/move\_base\_node/global\_costmap/inflated\_obstacles

/move\_base\_node/global\_costmap/obstacles

/move\_base\_node/global\_costmap/parameter\_descriptions

/move\_base\_node/global\_costmap/parameter\_updates

/move\_base\_node/global\_costmap/robot\_footprint

/move\_base\_node/global\_costmap/unknown\_space

/move\_base\_node/global\_plan

/move\_base\_node/local\_costmap/forbidden\_area

/move\_base\_node/local\_costmap/inflated\_obstacles

/move\_base\_node/local\_costmap/obstacles

/move\_base\_node/local\_costmap/parameter\_descriptions

/move\_base\_node/local\_costmap/parameter\_updates

/move\_base\_node/local\_costmap/robot\_footprint

/move\_base\_node/local\_costmap/unknown\_space

/move\_base\_node/mir\_escape\_recovery/visualization\_marker

/move\_base\_node/parameter\_descriptions

/move\_base\_node/parameter\_updates

/move\_base\_node/traffic\_costmap/forbidden\_area

/move\_base\_node/traffic\_costmap/inflated\_obstacles

/move\_base\_node/traffic\_costmap/obstacles

/move\_base\_node/traffic\_costmap/parameter\_descriptions

/move\_base\_node/traffic\_costmap/parameter\_updates

/move\_base\_node/traffic\_costmap/robot\_footprint

/move\_base\_node/traffic\_costmap/unknown\_space

/move\_base\_simple/goal

/odom

/odom\_enc

/one\_way\_map

/param\_update

/particlecloud

/relative\_move\_action/cancel

/relative\_move\_action/feedback

/relative\_move\_action/goal

/relative\_move\_action/result

/relative\_move\_action/status

/relative\_move\_node/parameter\_descriptions

/relative\_move\_node/parameter\_updates

/relative\_move\_node/time\_to\_coll

/relative\_move\_node/visualization\_marker

/robot\_mode

/robot\_pose

/robot\_state

/robot\_status

/robot\_tracker/external\_robots

/robot\_tracker/robots\_pointcloud

/robot\_tracker/tracked\_robots

/rosout

/rosout\_agg

/scan

/scan\_filter/parameter\_descriptions

/scan\_filter/parameter\_updates

/scan\_filter/visualization\_marker

/tf

/tf\_static

/traffic\_map

/transform\_footprint/parameter\_descriptions

/transform\_footprint/parameter\_updates

/transform\_imu/parameter\_descriptions

/transform\_imu/parameter\_updates