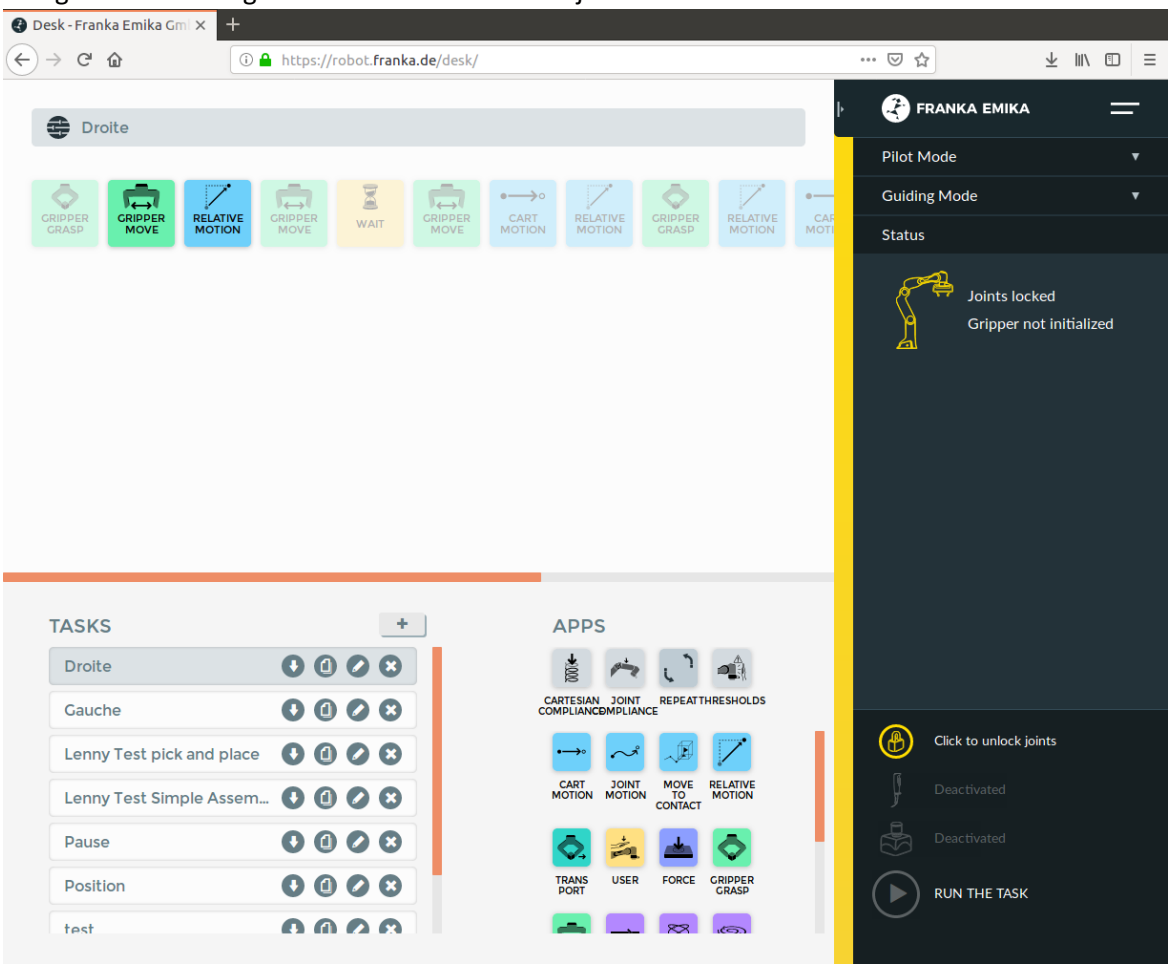
Start Franka desktop with Ubuntu 16 installed on it.

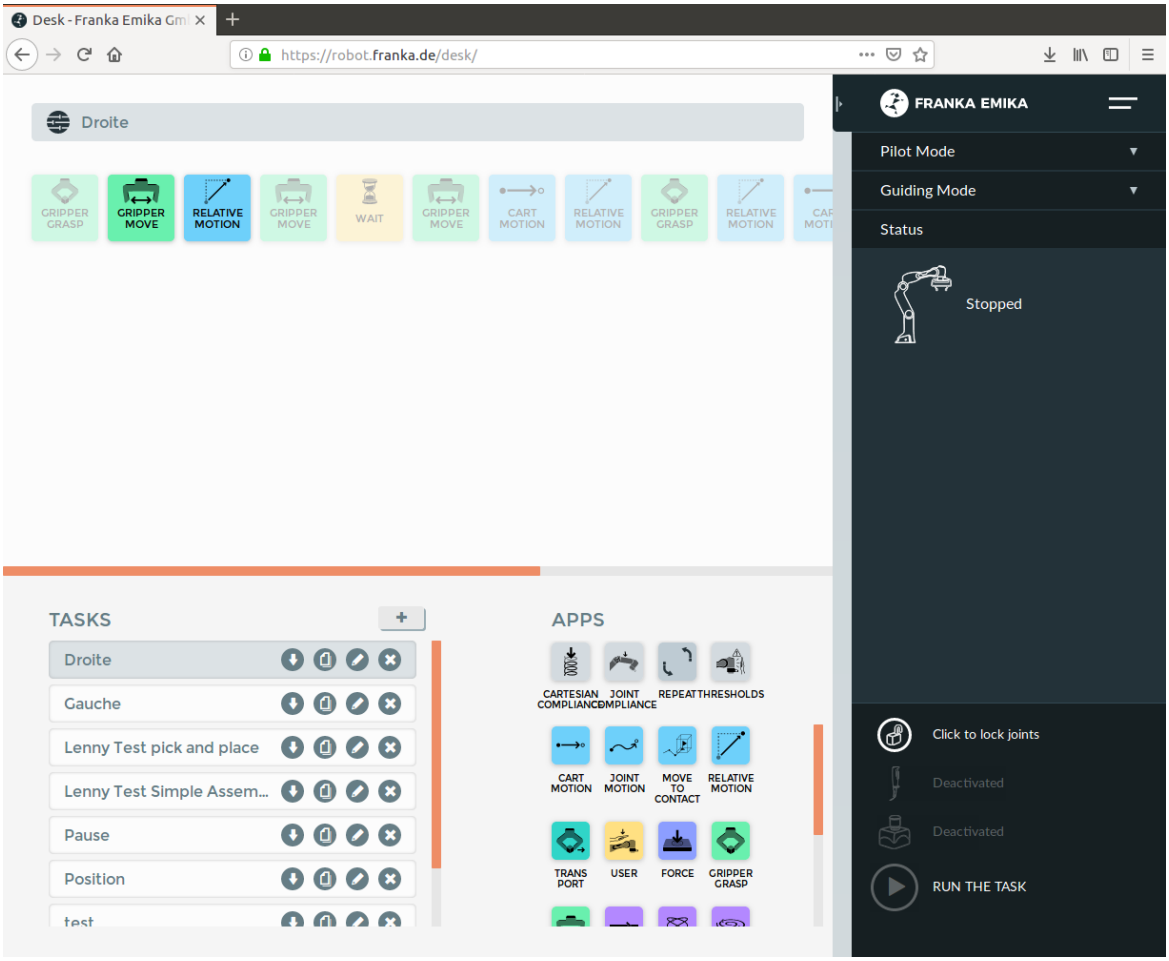
1) When starting the computer, press “shift” button till you get a menu where you can choose which Ubuntu version you want to run.
2) Choose “Advanced options for Ubuntu”.
3) Choose the first “rt” (real time kernel) version: “Ubuntu, with Linux 4.16.18-rt12”.
4) The password of this computer is “franka123”, its IP address “192.168.2.105”.

Start the Franka Panda robot via Desk.

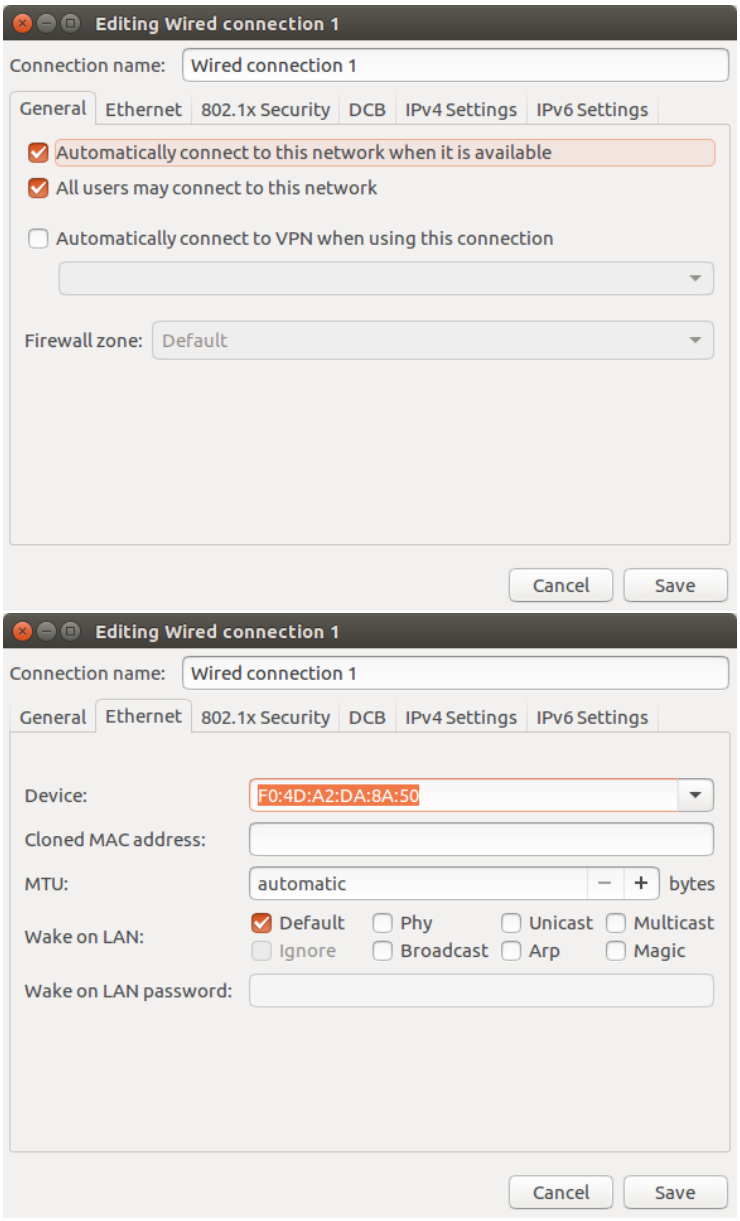
1) Turn on the Franka Control (black box).
2) Use an ethernet cable to connect the Franka Base and the Franka desktop.
3) Choose “Wired connection 1”. See below for specifications of this network.
4) Open Firefox Web Browser.
5) Go to https://robot.franka.de/desk/
6) The username is “admin”, the password is “franka123”.
7) You get the following window when the robot’s joints are locked.

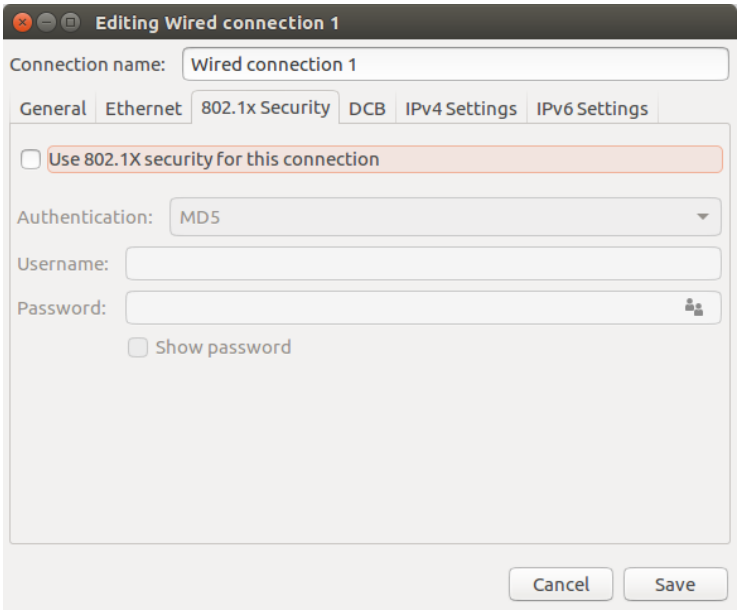


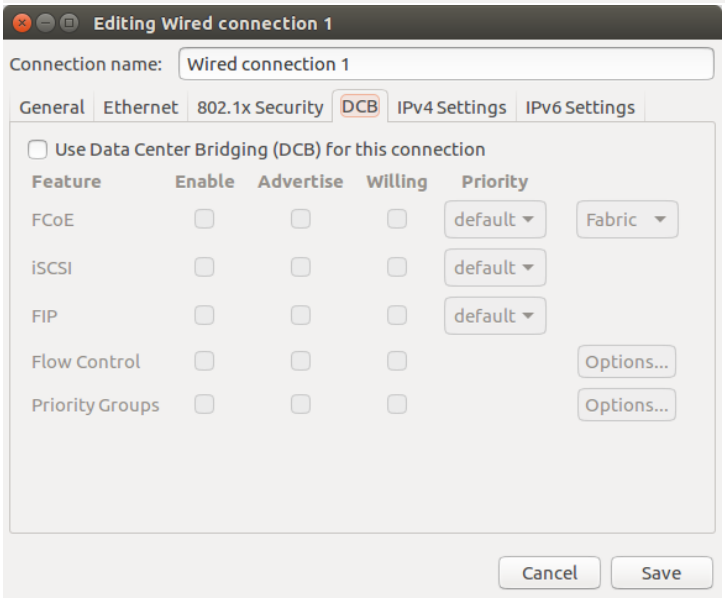
8) To unlock the joints, click on “click to unlock the joints”. You will hear 7 clicks (the robot will move a bit) and you will get the following window.

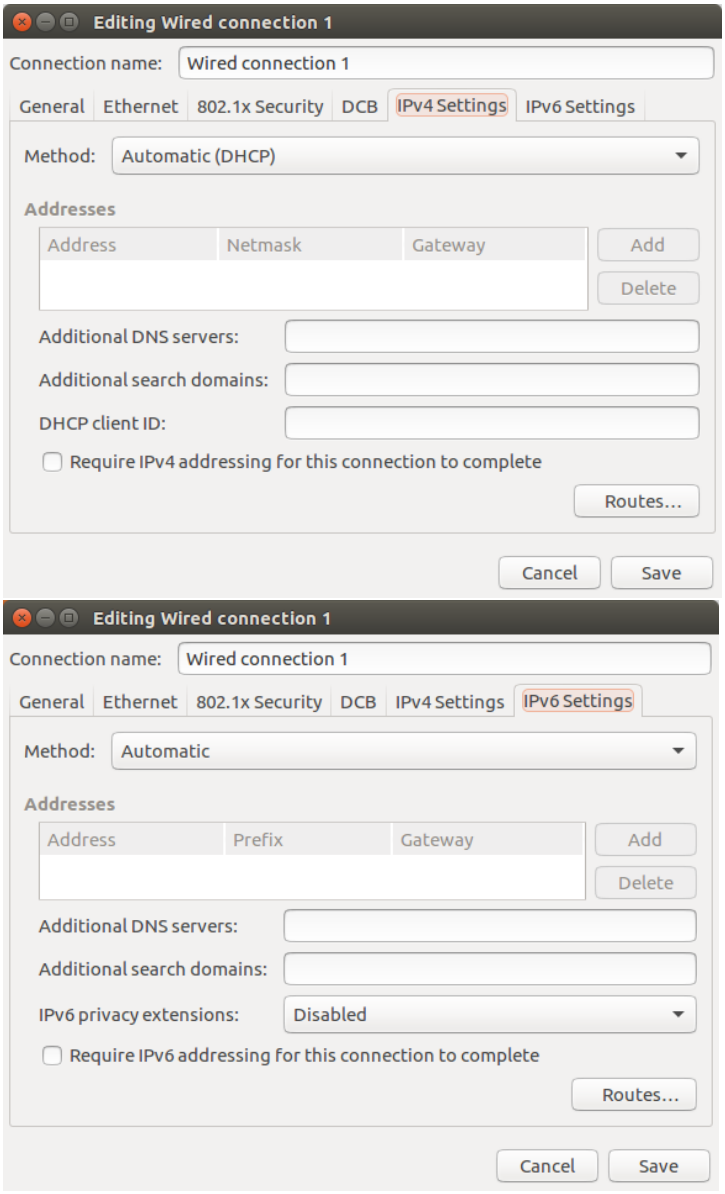


Specifications of “Wired connection 1”




Editing Wired connection 1
Connection name:
Wired connection 1
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Use 802.1X security for this connection
Authentication:
MD5
Username:
Password:
Show password
Cancel
Save


Editing Wired connection 1
Connection name:
Wired connection 1
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Use Data Center Bridging (DCB) for this connection
Feature
Enable
Advertise
Willing
Priority
FCoE
default
Fabric
iSCSI
default
FIP
default
Flow Control
Options...
Priority Groups
Options...
Cancel
Save


Editing Wired connection 1
Connection name: Wired connection 1
General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings
Method: Automatic (DHCP)
Addresses
Address Netmask Gateway
Add
Delete
Additional DNS servers:
Additional search domains:
DHCP client ID:
Require IPv4 addressing for this connection to complete
Routes...
Cancel Save
Editing Wired connection 1
Connection name: Wired connection 1
General Ethernet 802.1x Security DCB IPv4 Settings IPv6 Settings
Method: Automatic
Addresses
Address Prefix Gateway
Add
Delete
Additional DNS servers:
Additional search domains:
IPv6 privacy extensions: Disabled
Require IPv6 addressing for this connection to complete
Routes...
Cancel Save

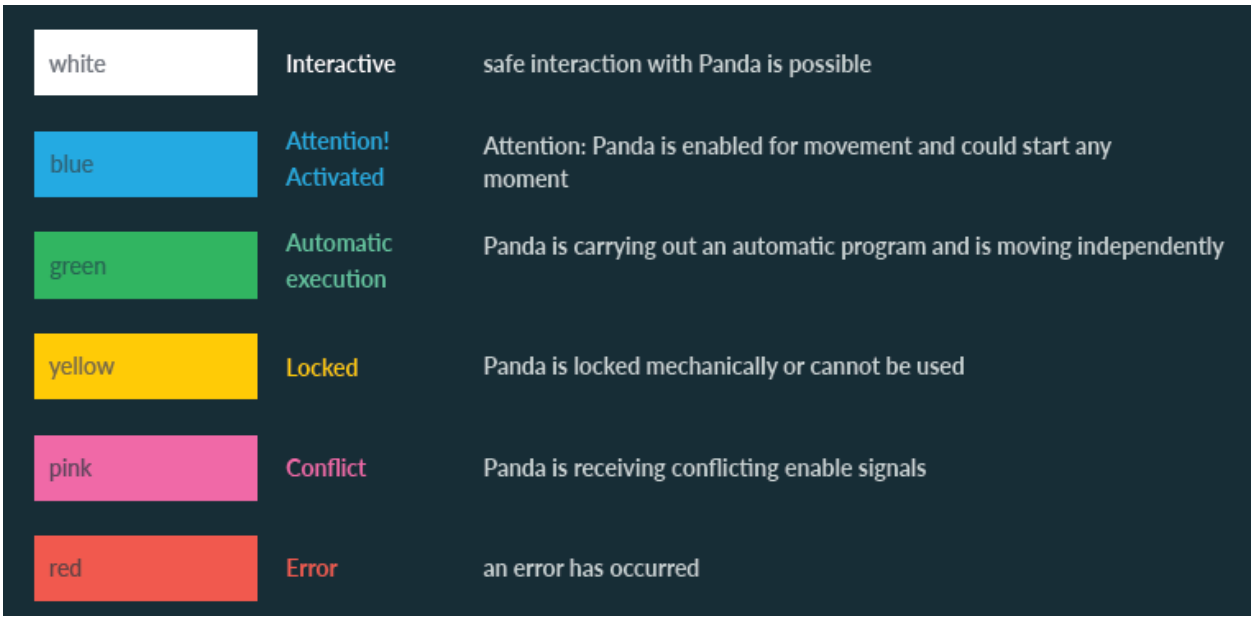
Control the Franka Panda robot via Franka Control Interface (FCI).

1) Disconnect the Franka Base from the Franka desktop. Use the same ethernet cable to connect the Franka Control with the Franka desktop.
2) Choose “Control”. See below for specifications of this network.
3) Check that you can receive data by for example using the echo code.

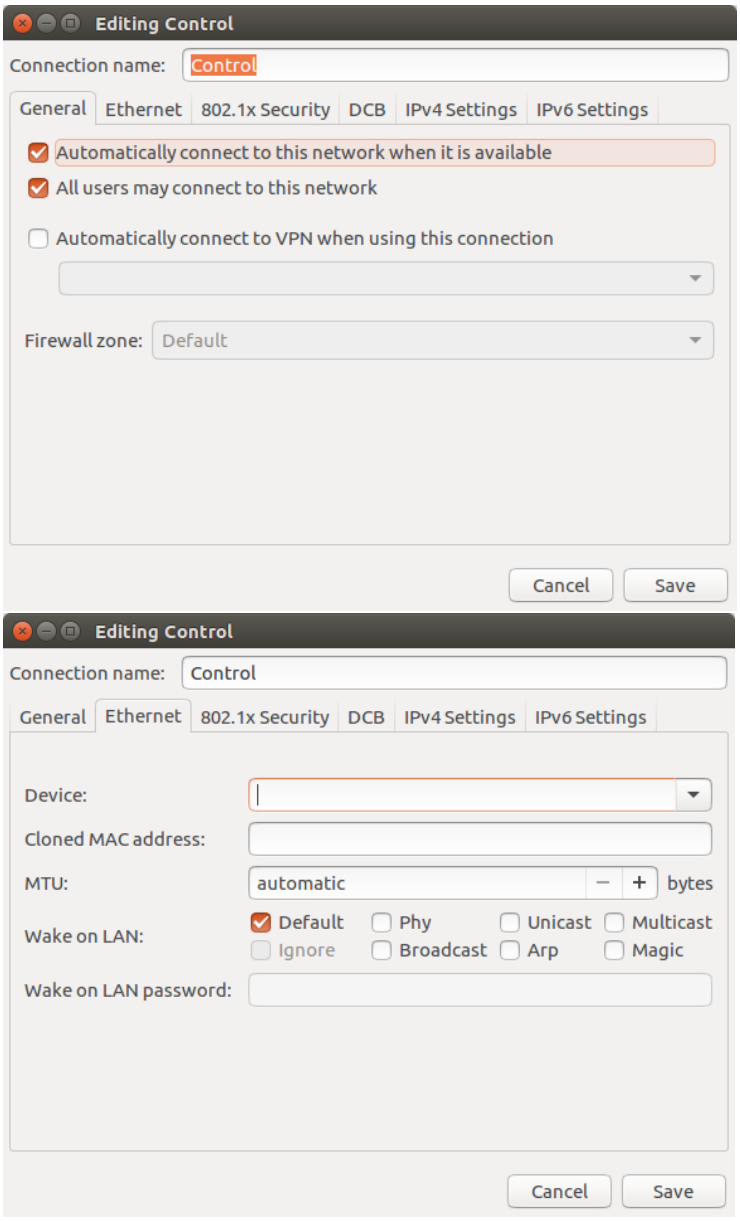
Run a code

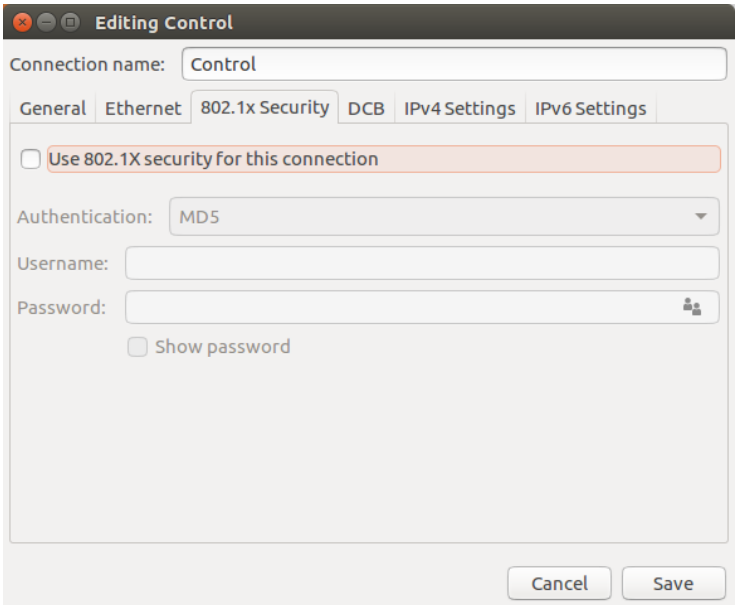
My programs and Harry’s programs can be found in the examples folder in Libfranka.

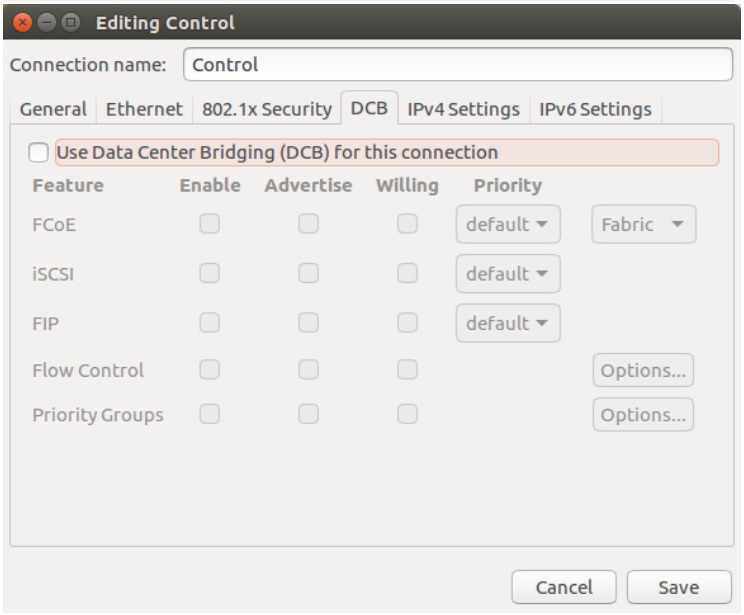
1) Open a terminal and build
    a. cd libfranka/build
    b. cmake -DCMAKE_BUILD_TYPE=Release ..
    c. cmake --build . (you have to do this every time you change something in your code)
2) Open a new terminal and run your code
    a. cd libfranka/build/examples/Kelly (the folder where you can find your program)
    b. ./echo_robot_state_Kelly
3) When you get the error “command not possible in the current mode!”, you probably have to open the emergency-stop device (such that lights are blue).
    a. Emergency-stop device pushed = lights of the robot are white
    b. Emergency-stop device open = lights of the robot are blue

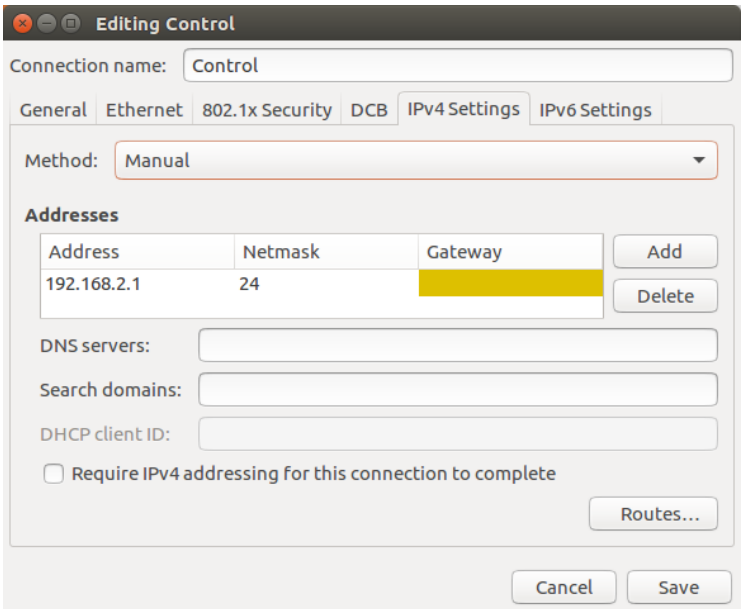


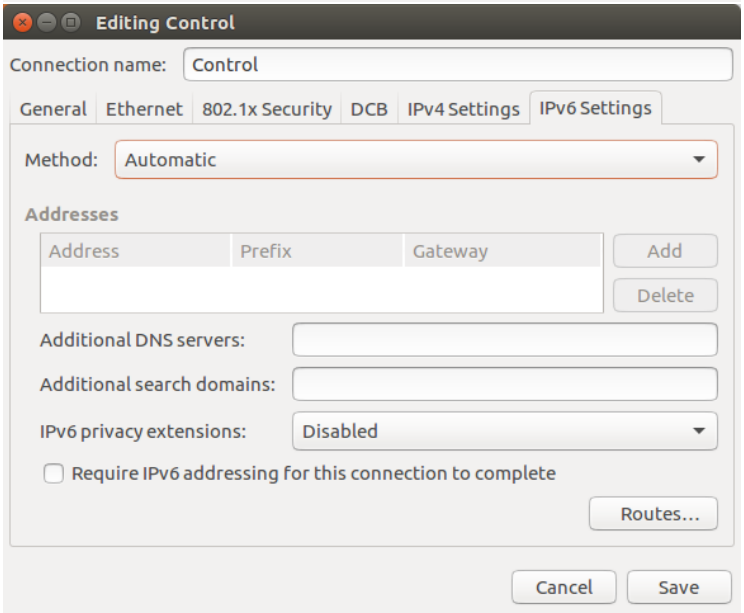
Specifications of “Control”



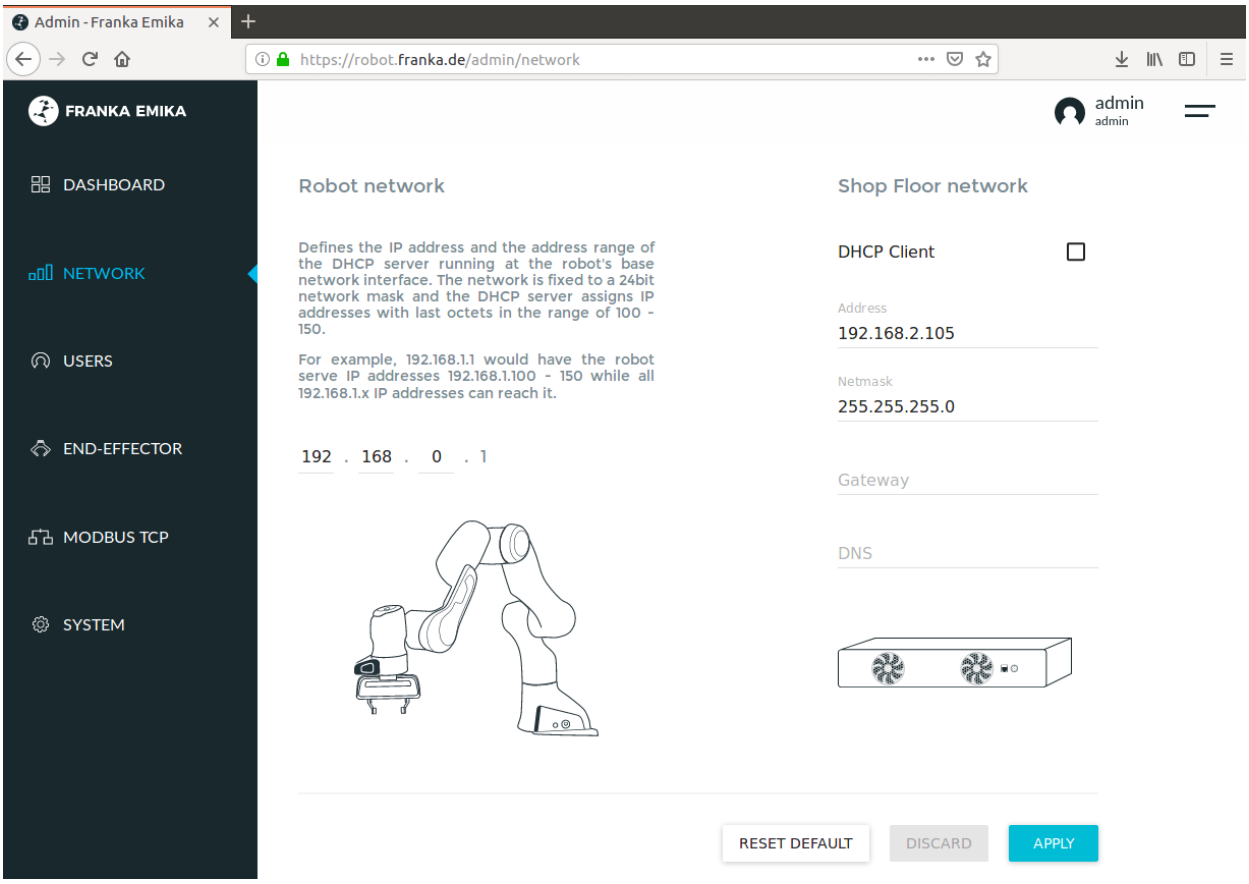

Editing Control
Connection name:
Control
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Use 802.1X security for this connection
Authentication:
MD5
Username:
Password:
Show password
Cancel
Save


Editing Control
Connection name:
Control
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Use Data Center Bridging (DCB) for this connection
Feature
Enable
Advertise
Willing
Priority
FCoE
default
Fabric
iSCSI
default
FIP
default
Flow Control
Options...
Priority Groups
Options...
Cancel
Save


Editing Control
Connection name: Control
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Method: Manual
Addresses
Address
Netmask
Gateway
Add
192.168.2.1
24
Delete
DNS servers:
Search domains:
DHCP client ID:
Require IPv4 addressing for this connection to complete
Routes...
Cancel
Save


Editing Control
Connection name: Control
General
Ethernet
802.1x Security
DCB
IPv4 Settings
IPv6 Settings
Method: Automatic
Addresses
Address
Prefix
Gateway
Add
Delete
Additional DNS servers:
Additional search domains:
IPv6 privacy extensions: Disabled
Require IPv6 addressing for this connection to complete
Routes...
Cancel
Save

The address in IPv4 Settings is chosen as “192.168.2.1” because the robot’s address is “192.168.2.105”, see screenshot below.

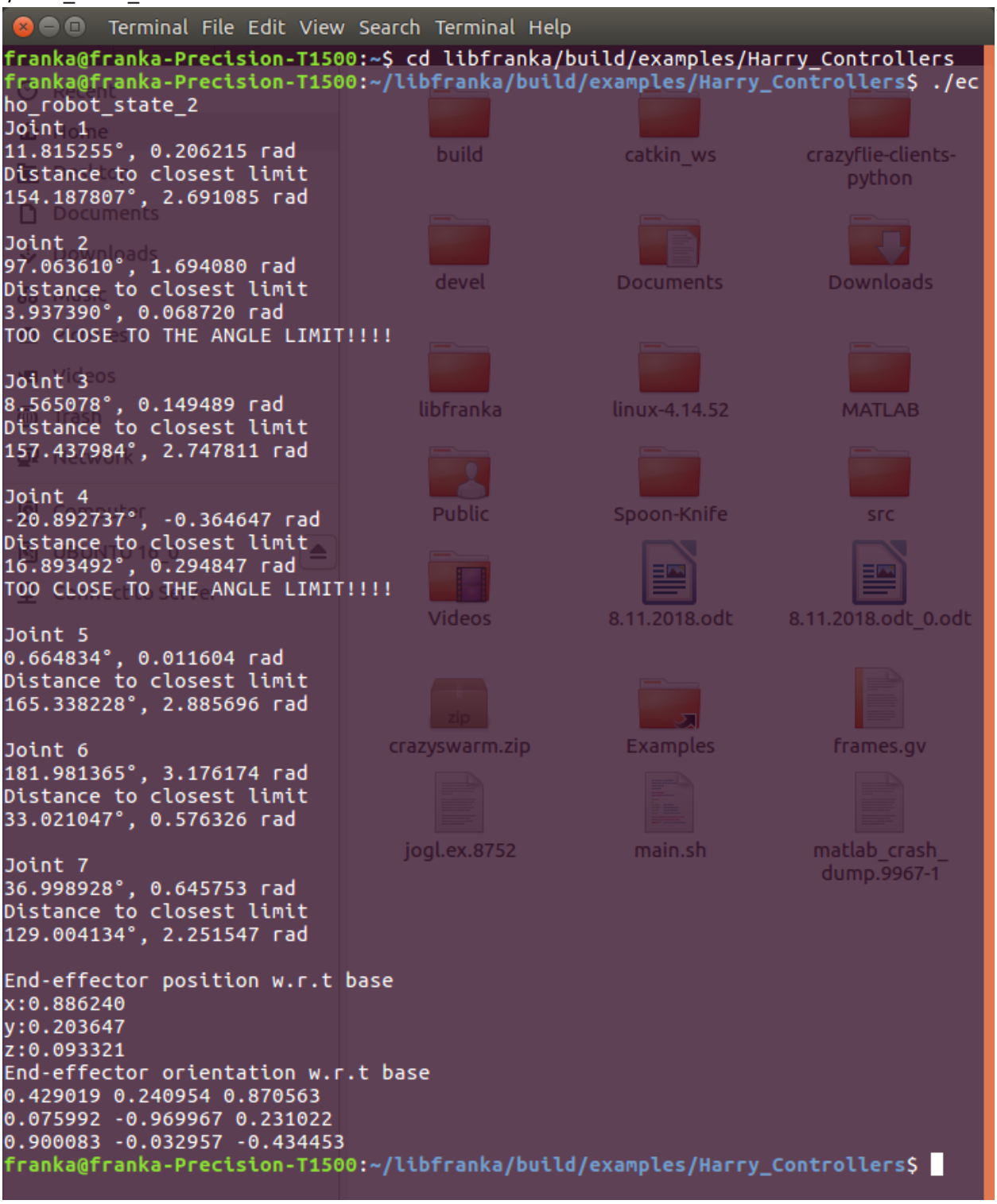


Check that the connection works.

New echo robot state code can be found in folder “Kelly”

cd libfranka/build/examples/Kelly

./echo_robot_state



Get Vicon data on the Franka desktop

1) Keep the ethernet connection between the Franka desktop and the Franka control. Since the Franka desktop has only one ethernet port, the connection between the Franka desktop and the Vicon (Windows) desktop has to be done wireless.
2) Make and start an Ad-Hoc network on the Vicon desktop.
    a. Press “Windows + x”, select “Windows PowerShell (administrator)”, and set hostednetwork called “adhocname” and key “password”.
    You don’t always have to change/set the hostednetwork name and password. In case you want to use the already created hostednetwork, go to step c/d.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> netsh wlan set hostednetwork mode=allow ssid=adhocname key=password
```

    b. You get a message that the hostednetwork you just created is set.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> netsh wlan set hostednetwork mode=allow ssid=adhocname key=password
The hosted network mode has been set to allow.
The SSID of the hosted network has been successfully changed.
The user key passphrase of the hosted network has been successfully changed.

PS C:\WINDOWS\system32>
```

    c. Show more information about this hostednetwork. Now you can see it is not yet started.

```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork

Hosted network settings
-----------------------
    Mode                   : Allowed
    SSID name              : "adhocname"
    Max number of clients  : 100
    Authentication         : WPA2-Personal
    Cipher                 : CCMP

Hosted network status
---------------------
    Status                 : Not started
```

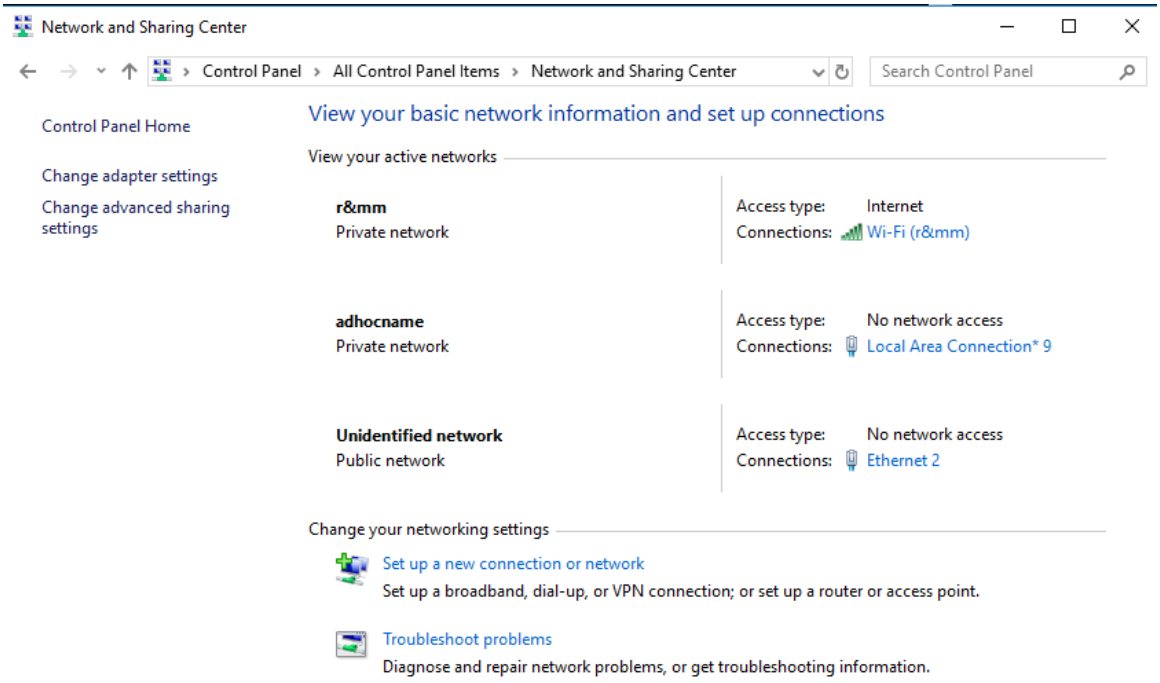
    d. Start the hostednetwork.

```
PS C:\WINDOWS\system32> netsh wlan start hostednetwork
The hosted network started.

PS C:\WINDOWS\system32>
```

e. Show more information about this hostednetwork. Now you can see it is started and that it has 0 clients.

```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork

Hosted network settings
-----------------------
    Mode                   : Allowed
    SSID name              : "adhocname"
    Max number of clients  : 100
    Authentication         : WPA2-Personal
    Cipher                 : CCMP

Hosted network status
---------------------
    Status                 : Started
    BSSID                  : 00:1b:11:be:0b:a6
    Radio type             : 802.11g
    Channel                : 2
    Number of clients      : 0

PS C:\WINDOWS\system32>
```

3) Check in “Network & Sharing Center” the created network. You can see it under the name “adhocname” and with connection type “Local Area Connection*9”.

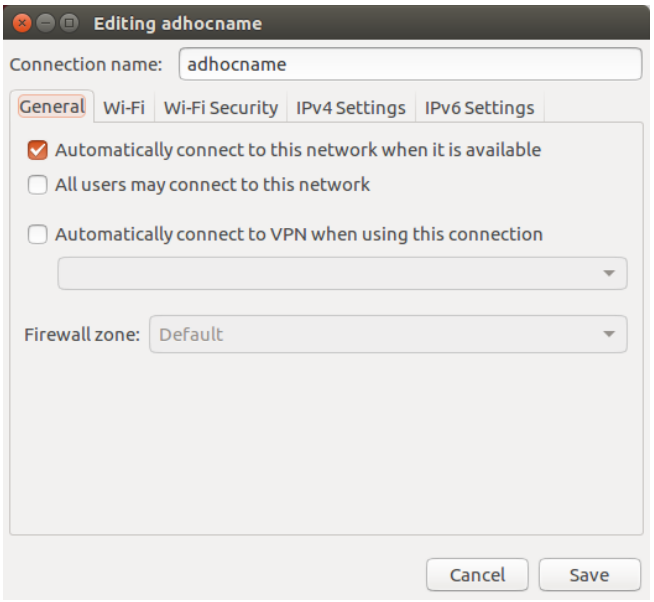


4) Check the IP address of the created network.
   a. Open a command prompt.
   b. Type “ipconfig”.
   c. Under “Wireless LAN adapter Local Area Connection* 9” you can see that the IP address of the created network is “192.168.137.1”.

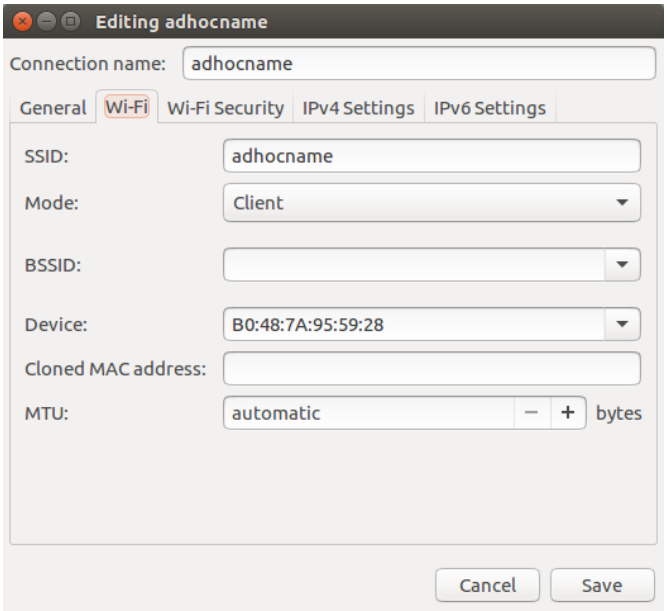
```
C:\Users\VICON>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 4:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 5:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.10.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4d71:3348:dc0f:4d7b%7
   IPv4 Address. . . . . . . . . . . : 192.168.1.176
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Wireless LAN adapter Local Area Connection* 9:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::90da:3919:7b88:91e%19
   IPv4 Address. . . . . . . . . . . : 192.168.137.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

5) Use a Wi-Fi dongle to have Wi-Fi on the Franka desktop (Ubuntu) while having an ethernet connection.

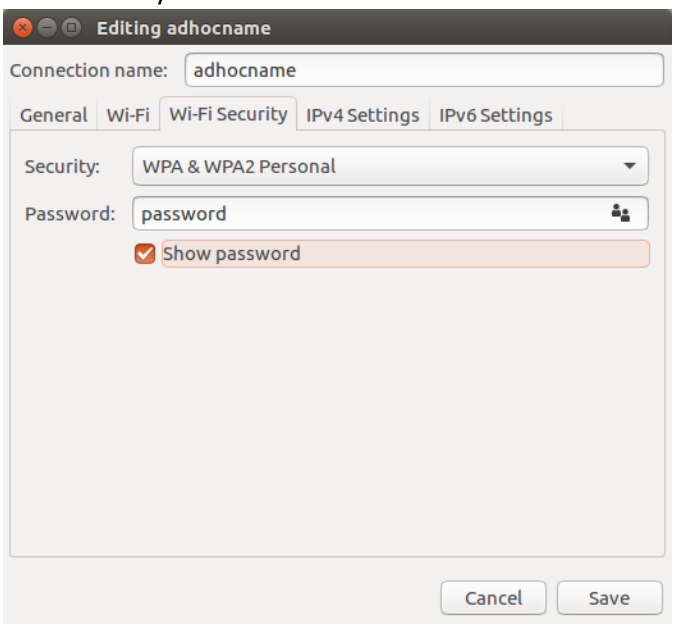
6) Make a new wireless network on the Franka desktop with following properties. Go to step 7) if you already created this wireless network.
    a. General



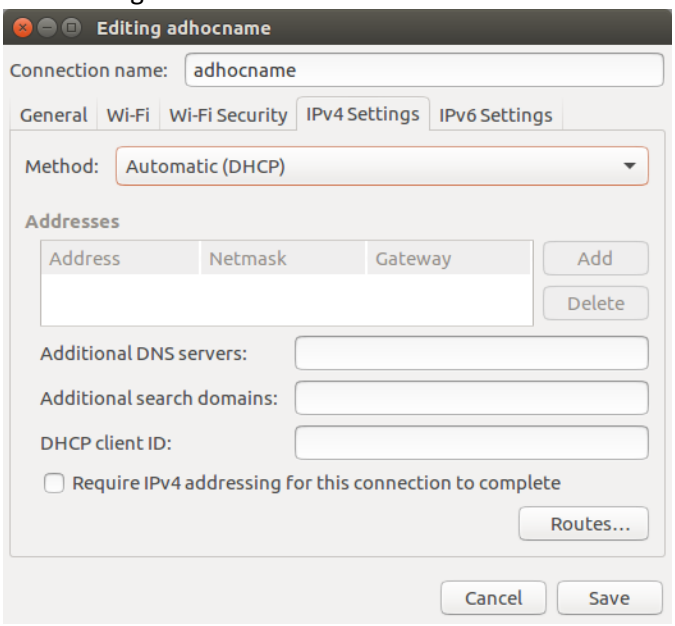
    b. Wi-Fi



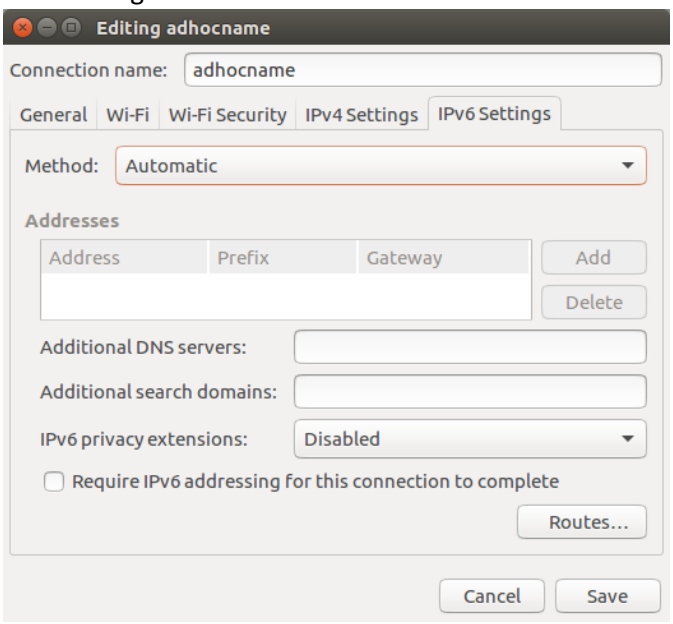
c. Wi-Fi Security



d. IPv4 Settings

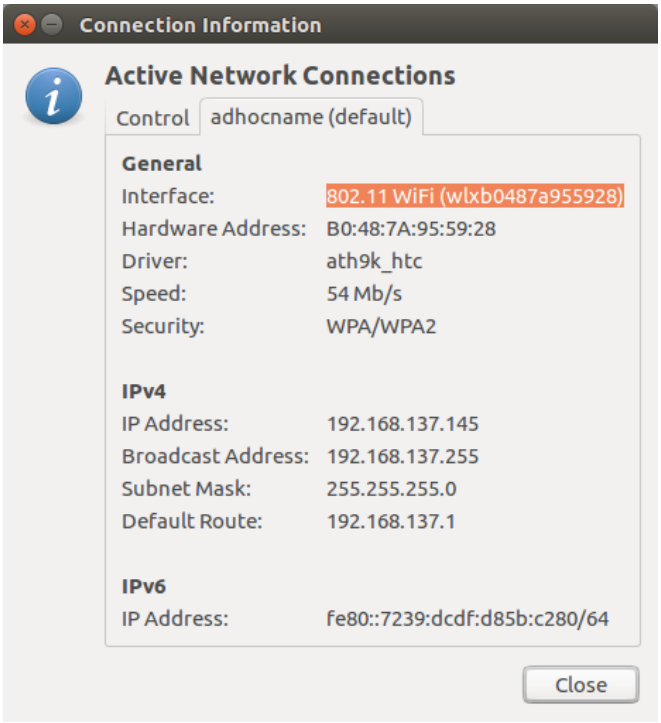


e. IPv6 Settings



7) The wireless network with the name “adhocname” will show up in the list of wireless networks. If not, exert and insert the Wi-Fi dongle again. Connect to this network.
8) Check the connection information on the Franka desktop.
   a. Ethernet connection with the name “Control” to Franka Panda Control.

b. Wireless connection with the name “adhocname” to Vicon desktop.
Here you can see the IP address of the Vicon desktop, i.e. Default Route: 192.168.137.1, and of the Franka desktop, i.e. IP Address: 192.168.137.145.



9) Check that the Franka desktop can send and receive packages to and from the Vicon desktop. Open a terminal (Ctrl+Alt+t), type “ping 192.168.137.1”.

```
franka@franka-Precision-T1500:~$ ping 192.168.137.1
PING 192.168.137.1 (192.168.137.1) 56(84) bytes of data.
64 bytes from 192.168.137.1: icmp_seq=1 ttl=128 time=1.30 ms
64 bytes from 192.168.137.1: icmp_seq=2 ttl=128 time=1.39 ms
64 bytes from 192.168.137.1: icmp_seq=3 ttl=128 time=0.969 ms
64 bytes from 192.168.137.1: icmp_seq=4 ttl=128 time=1.20 ms
64 bytes from 192.168.137.1: icmp_seq=5 ttl=128 time=1.87 ms
64 bytes from 192.168.137.1: icmp_seq=6 ttl=128 time=0.882 ms
64 bytes from 192.168.137.1: icmp_seq=7 ttl=128 time=1.38 ms
64 bytes from 192.168.137.1: icmp_seq=8 ttl=128 time=1.36 ms
64 bytes from 192.168.137.1: icmp_seq=9 ttl=128 time=3.10 ms
64 bytes from 192.168.137.1: icmp_seq=10 ttl=128 time=1.38 ms
64 bytes from 192.168.137.1: icmp_seq=11 ttl=128 time=1.55 ms
64 bytes from 192.168.137.1: icmp_seq=12 ttl=128 time=1.55 ms
64 bytes from 192.168.137.1: icmp_seq=13 ttl=128 time=4.93 ms
64 bytes from 192.168.137.1: icmp_seq=14 ttl=128 time=1.00 ms
64 bytes from 192.168.137.1: icmp_seq=15 ttl=128 time=1.76 ms
64 bytes from 192.168.137.1: icmp_seq=16 ttl=128 time=5.27 ms
64 bytes from 192.168.137.1: icmp_seq=17 ttl=128 time=1.52 ms
^C
--- 192.168.137.1 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16085ms
rtt min/avg/max/mdev = 0.882/1.910/5.275/1.262 ms
```

10) Check that the Vicon desktop can send and receive packages to and from the Franka desktop. Open a command prompt, type “ping 192.168.137.145”.

```
C:\Users\VICON>ping 192.168.137.145

Pinging 192.168.137.145 with 32 bytes of data:
Reply from 192.168.137.145: bytes=32 time=2ms TTL=64
Reply from 192.168.137.145: bytes=32 time=2ms TTL=64
Reply from 192.168.137.145: bytes=32 time=1ms TTL=64
Reply from 192.168.137.145: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.137.145:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

11) Show more information about the hostednetwork in the Windows PowerShell (Administrator). Now you can see the created Ad-Hoc network has 1 client.

```
PS C:\WINDOWS\system32> netsh wlan show hostednetwork

Hosted network settings
-----------------------
    Mode                   : Allowed
    SSID name              : "adhocname"
    Max number of clients  : 100
    Authentication         : WPA2-Personal
    Cipher                 : CCMP

Hosted network status
---------------------
    Status                 : Started
    BSSID                  : 00:1b:11:be:0b:a6
    Radio type             : 802.11b
    Channel                : 2
    Number of clients      : 1
        b0:48:7a:95:59:28        Authenticated
```

12) Launch the vicon_bridge (bridge between Vicon and ROS) on the Franka desktop.
    a. Go to the “vicon.launch” file in the folder: catkin_ws/src/vicon_bridge/launch
    b. Change the IP address at the 5th line to the IP address of the Windows pc. In this example 192.168.137.1. Save this file.
    c. Open a new terminal (Ctrl+Alt+t), type “cd catkin_ws/devel”, “source setup.bash”, “cd ..”, and “roslaunch vicon_bridge vicon.launch”.
    d. If it works, you will get the following window. Objects that are set visible in the Vicon software will be created and will be advertised. In case the object is occluded, a warning will show up.

```
franka@franka-Precision-T1500:~/catkin_ws$ cd devel/
franka@franka-Precision-T1500:~/catkin_ws/devel$ source setup.bash
franka@franka-Precision-T1500:~/catkin_ws/devel$ cd ..
franka@franka-Precision-T1500:~/catkin_ws$ roslaunch vicon_bridge vicon.launch
... logging to /home/franka/.ros/log/0c066c04-136b-11e9-8cbe-f04da2da8a50/roslau
nch-franka-Precision-T1500-4400.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://franka-Precision-T1500:45709/

SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.14
 * /vicon/datastream_hostport: 192.168.137.1:801
 * /vicon/stream_mode: ClientPull
 * /vicon/tf_ref_frame_id: /world

NODES
  /
    vicon (vicon_bridge/vicon_bridge)

auto-starting new master
process[master]: started with pid [4410]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 0c066c04-136b-11e9-8cbe-f04da2da8a50
process[rosout-1]: started with pid [4423]
started core service [/rosout]
process[vicon-2]: started with pid [4426]
[ INFO] [1546968611.259579825]: Connecting to Vicon DataStream SDK at 192.168.13
7.1:801 ...
[ INFO] [1546968611.261424521]: .
[ INFO] [1546968612.261728647]: ... connected!
[ INFO] [1546968612.261844835]: Setting Stream Mode to ClientPull: Success
[ INFO] [1546968612.261882539]: Axis Mapping: X-Forward Y-Left Z-Up
[ INFO] [1546968612.261920818]: Version: 1.3.0
[ INFO] [1546968612.261950744]: setting up grab_vicon_pose service server ...
[ INFO] [1546968612.263205411]: setting up segment calibration service server ..
.
[ WARN] [1546968612.265299264]: grab frame returned false
[ WARN] [1546968612.268935340]: Wand occluded, not publishing...
[ INFO] [1546968612.280461397]: creating new object Wand/seg ...
[ WARN] [1546968612.282750878]: unable to load zero pose for Wand/seg
[ INFO] [1546968612.282799675]: ... done, advertised as " vicon/Wand/seg"
[ WARN] [1546968624.440503362]: Wand occluded, not publishing...
[ WARN] [1546968627.689481120]: Wand occluded, not publishing...
[ WARN] [1546968629.837934180]: Wand occluded, not publishing...
```

"vicon.launch" selected

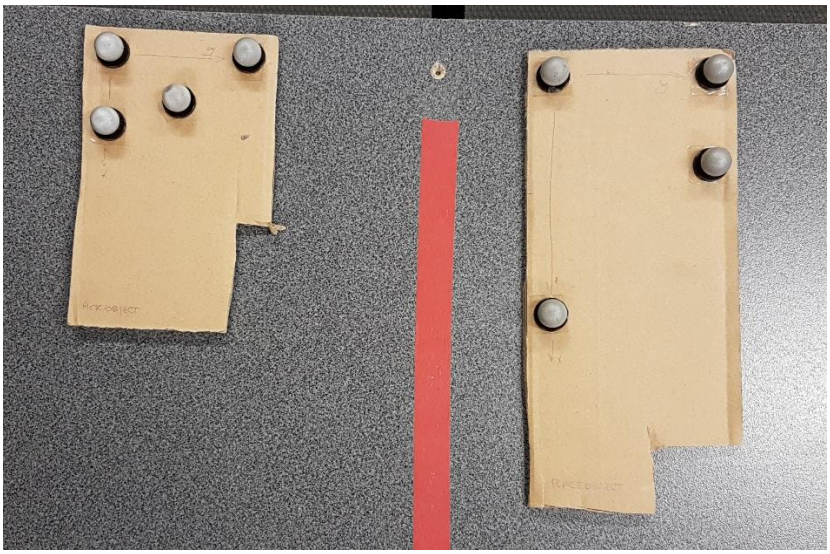
13) Check if you can get data from an object created and visible in Vicon on the Franka desktop.
Open a new terminal (Ctrl+Alt+t), type “rostopic echo /vicon/markers”. The name “markers” is in this case Wand/seg. In general it’s the object name / the segment name.
For my objects I had to write “rostopic echo /vicon/pickobject/pickobject”

Make objects in Vicon Nexus

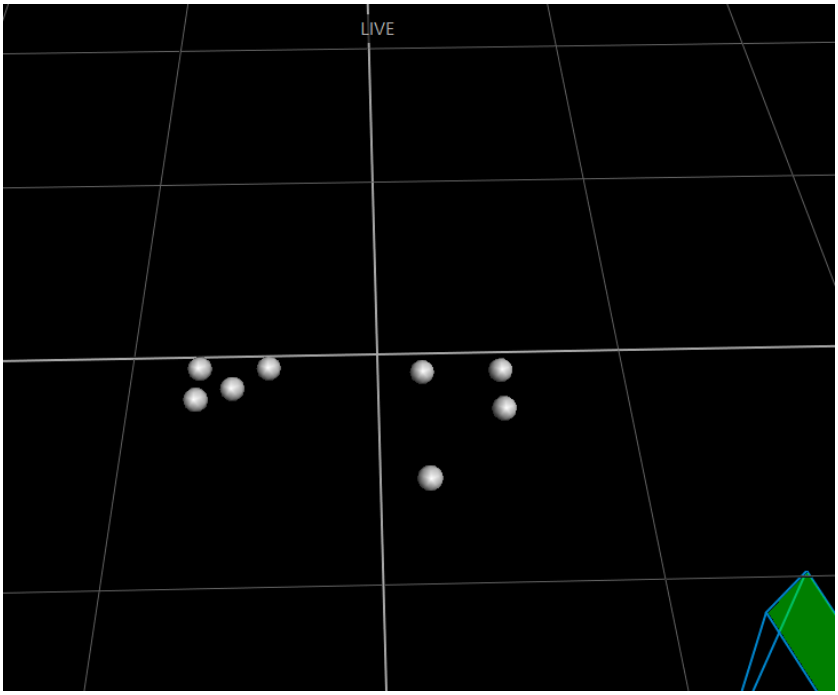
Look to Louis' report and to Harry's second report.

Make objects in Vicon Tracker

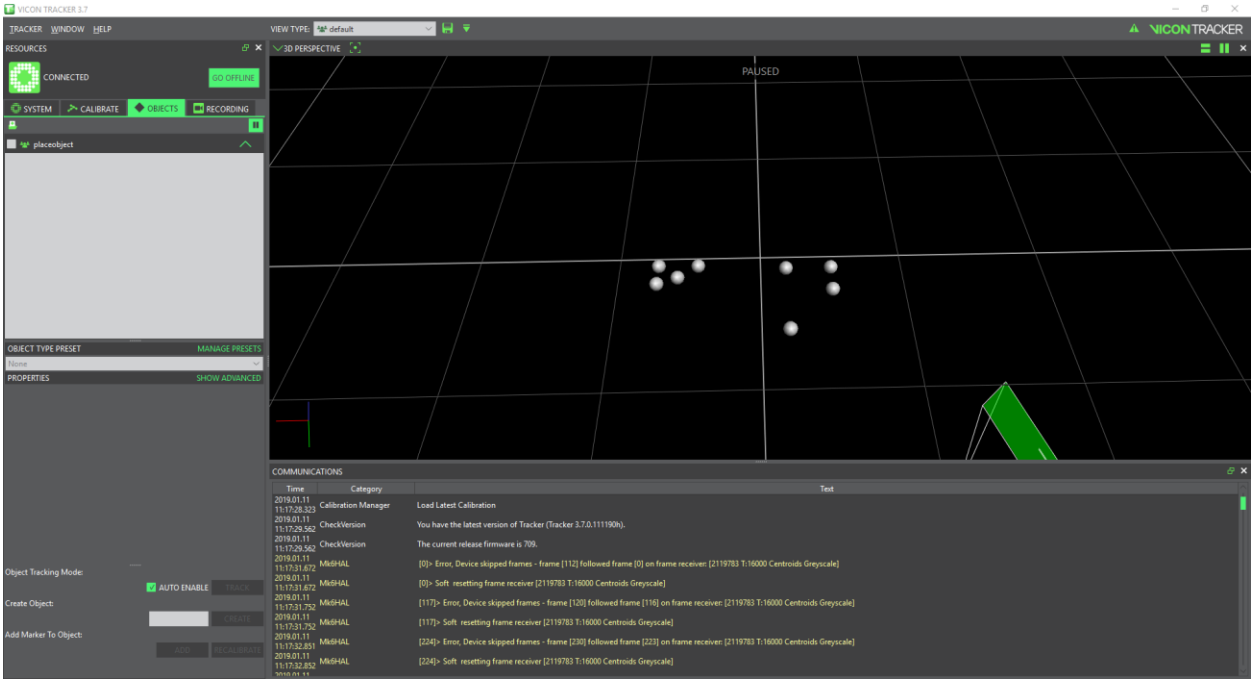
1. Put markers on your object.
   Here you can see the "pickobject" and the "placeobject", 4 reflective markers are placed on both.



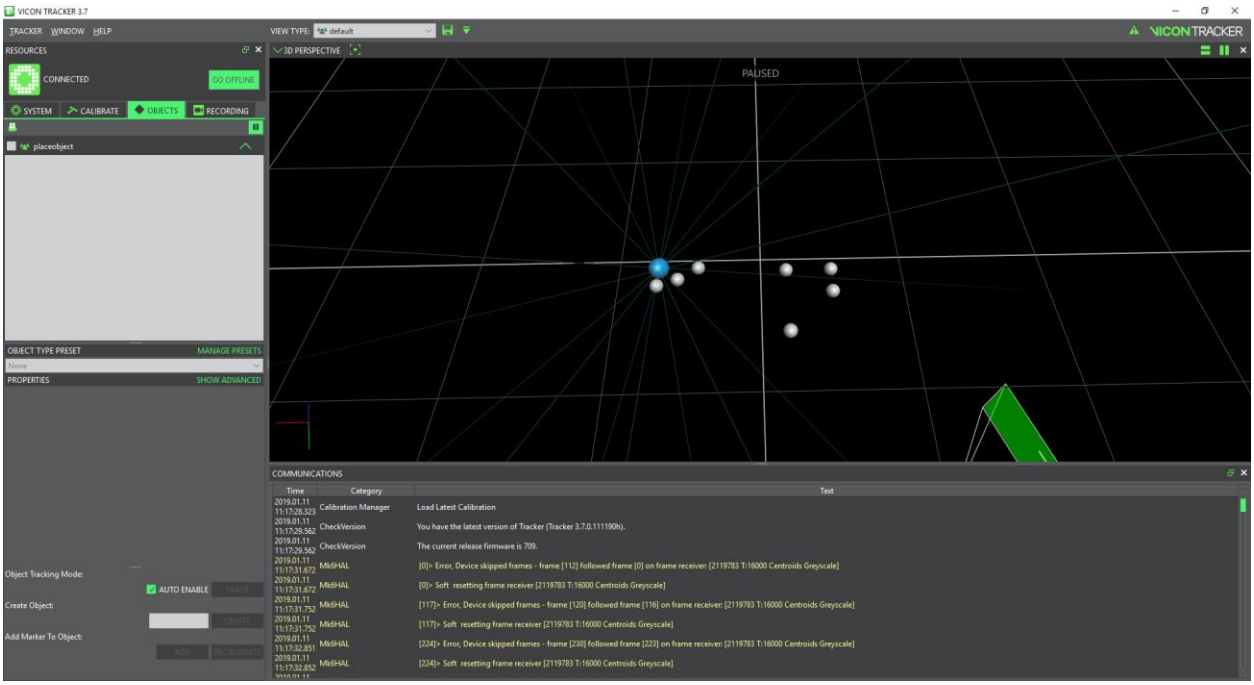
2. You can see them in Vicon Tracker as separated markers when no object is selected.

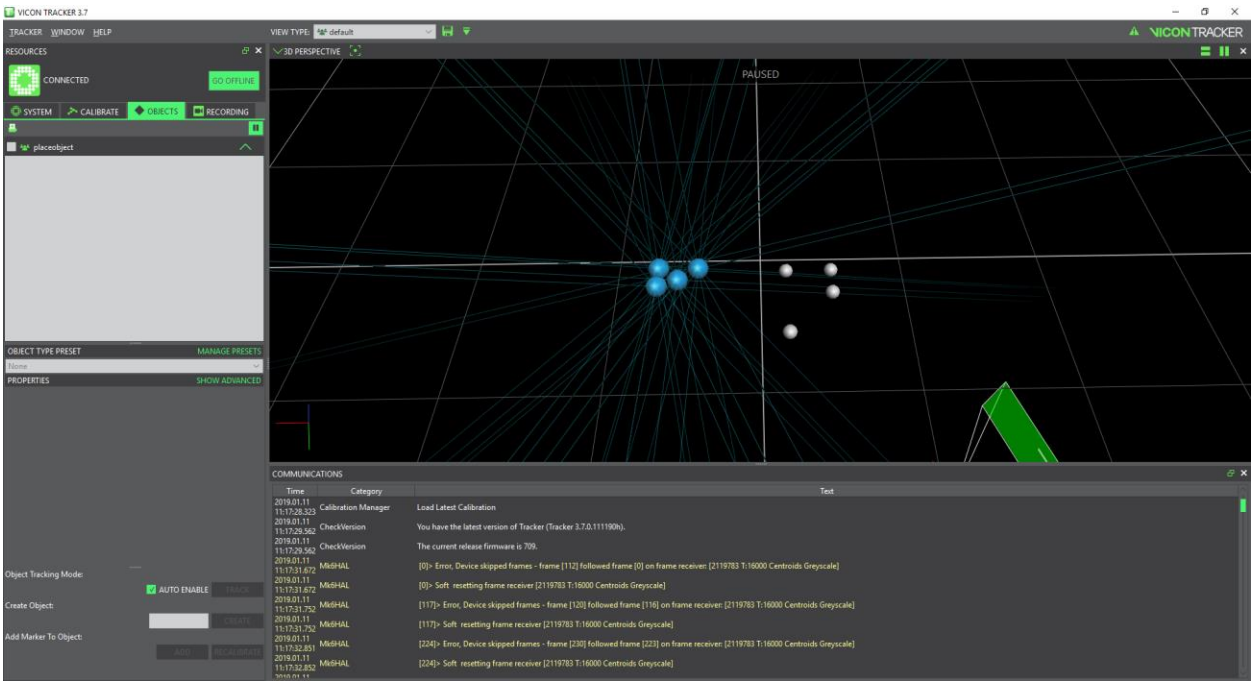


3. Pause the Vicon Tracker screen.

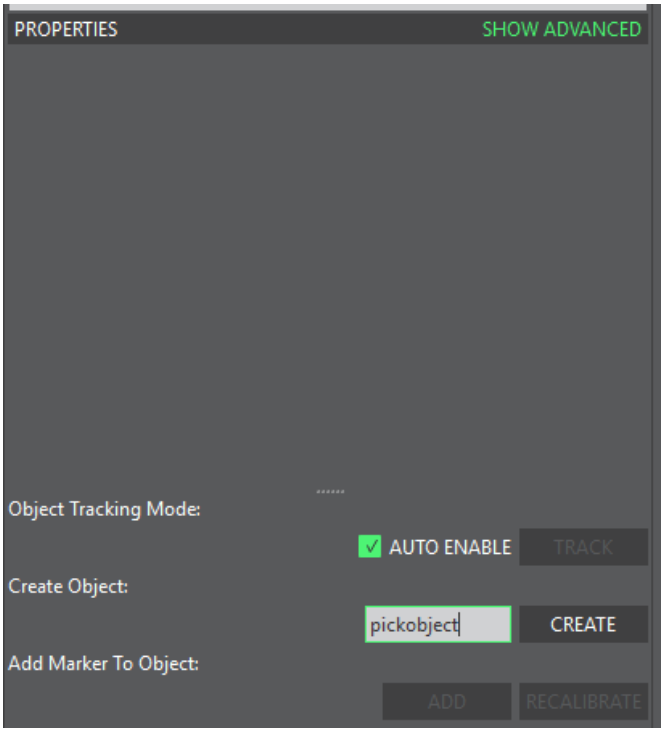


4. Select marker per marker, use the Ctrl key.

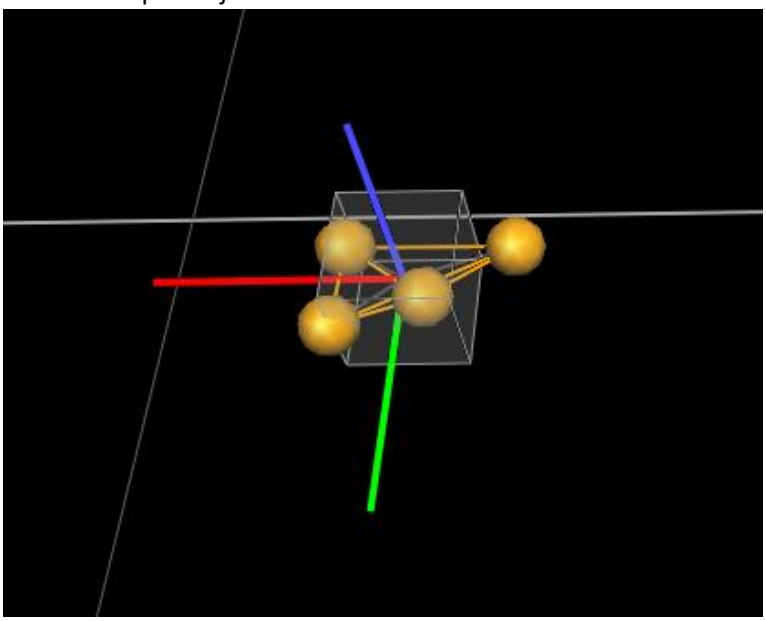




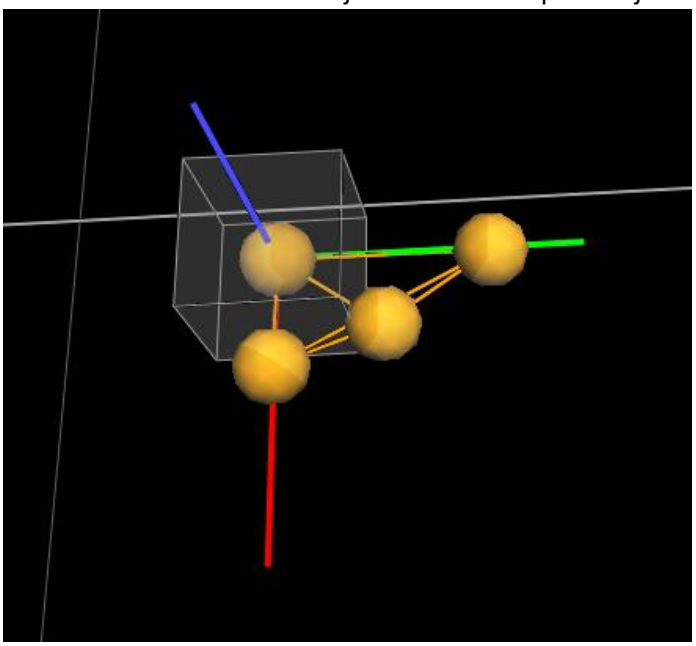
5. Create an object.



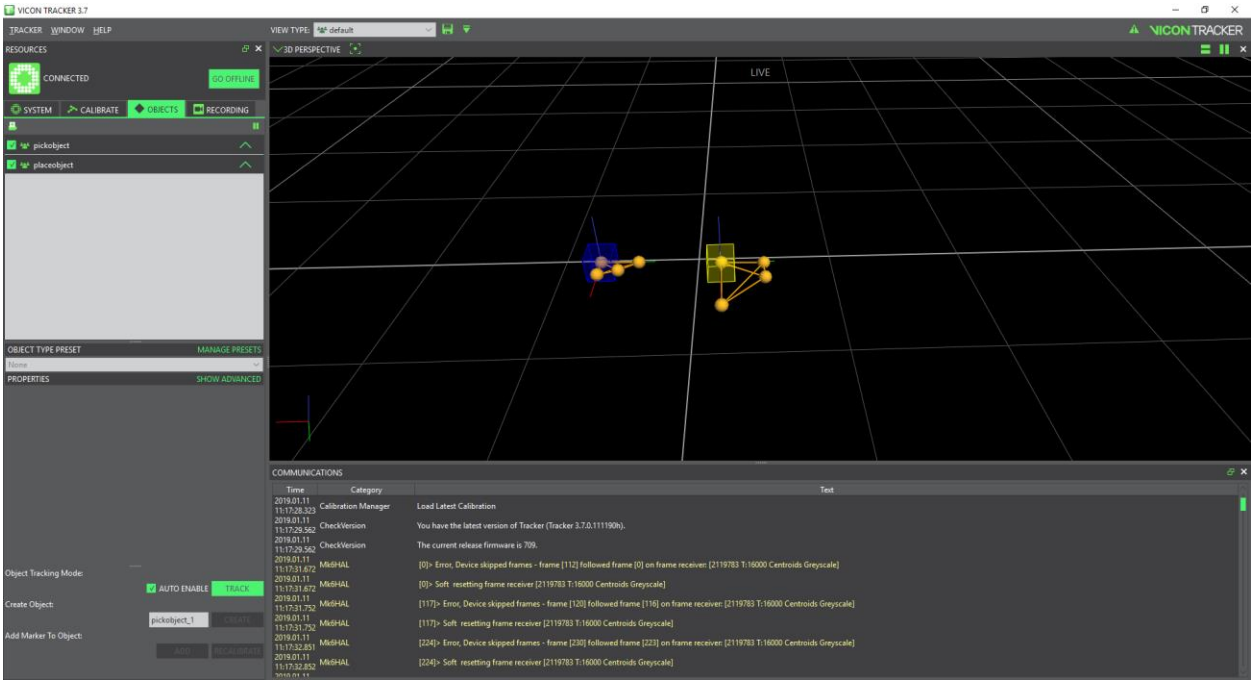
6. You will see its x-axis (red), y-axis (green), and z-axis (blue). You can change the position and orientation of this frame.
   This is the “pickobject”.



7. Do the same for a second object. This is the “placeobject”.

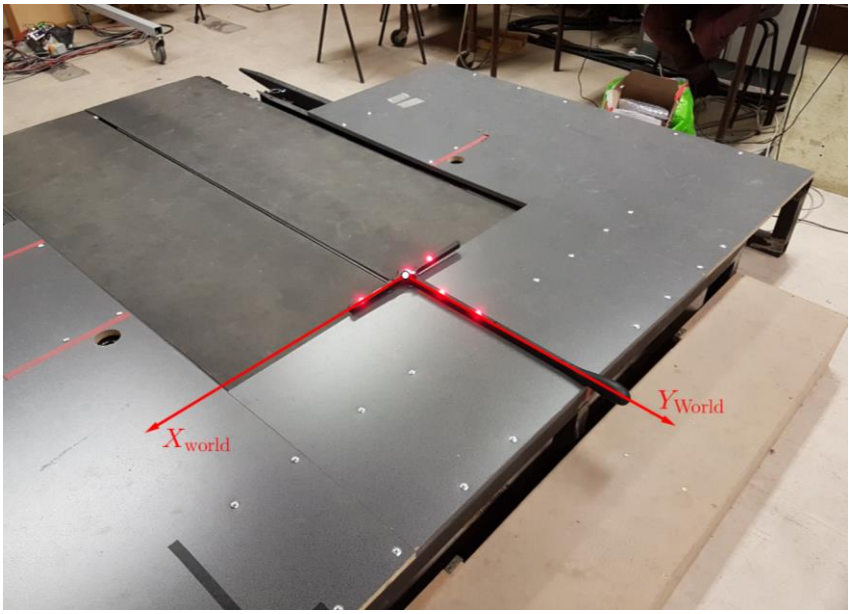


8. The two objects with their center (positioned and oriented myself) in blue (pick) and yellow (place).



Start of “Vicon_” codes

1. In Vicon we have set the world frame (origin + xyz axes) with the Vicon wand.



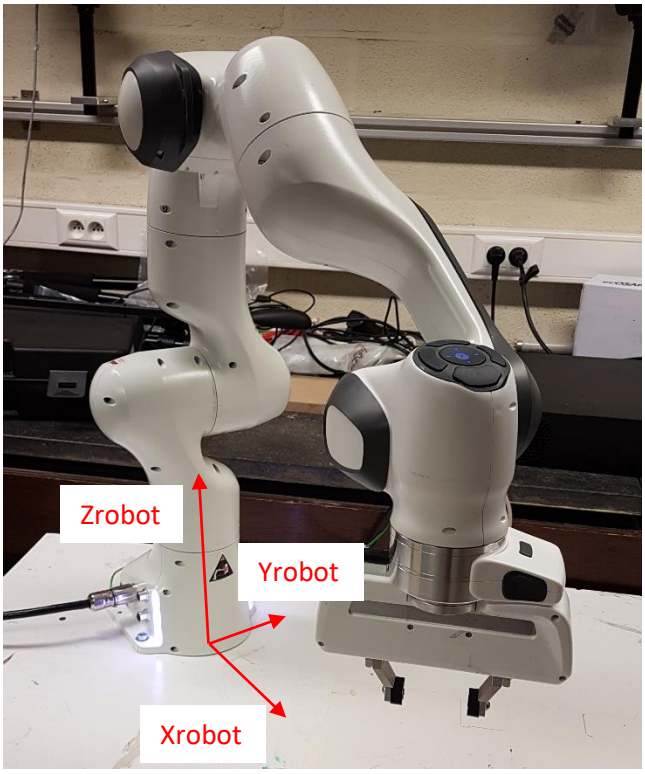
2. We put three markers on the robot to know the translation of the robot with respect to the world frame in x-, y-, and z-direction . In the code, we call this the robot_wrt_world frame.


```
robot_wrt_world(0) = msg->markers[8].translation.x; // [mm]
robot_wrt_world(1) = msg->markers[8].translation.y; // [mm]
robot_wrt_world(2) = msg->markers[8].translation.z; // [mm]
```

3. We also put three markers in the corner of the table where the robot is mounted on to know the translation of the table with respect to the world frame in x-, y-, and z-direction . In the code, we call this the table_wrt_world frame.

```
table_wrt_world(0) = msg->markers[11].translation.x; // [mm]
table_wrt_world(1) = msg->markers[11].translation.y; // [mm]
table_wrt_world(2) = msg->markers[11].translation.z; // [mm]
```

4. We want to define the “real” robot origin with respect to the world frame set by the Vicon wand. The real robot frame, as set by Franka, is a frame of which the origin is positioned in the base of the robot and oriented as shown on the figure below.
   For the real x- and real y-position, we use the x- and y-coordinates of the robot_wrt_world frame. For the real z-position, we use the z-coordinate of the table_wrt_world frame.



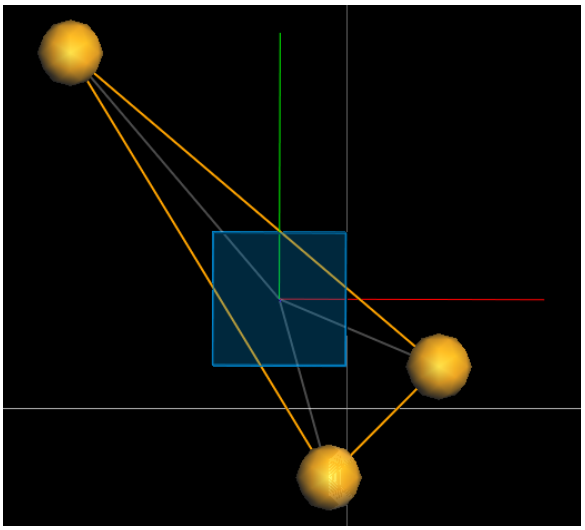
These positions have to be filled in manually.

```
robot_origin(0) = -81.8/1000; // = robot_wrt_world(0) [m]
robot_origin(1) = 2011.3/1000; // = robot_wrt_world(1) [m]
robot_origin(2) = 462.7/1000; // = table_wrt_world(2) [m]
```

Every time you place the robot and table back in the Vicon room, you have to repeat this process, because the position can vary. If you don’t move the table, then you don’t have to do this.

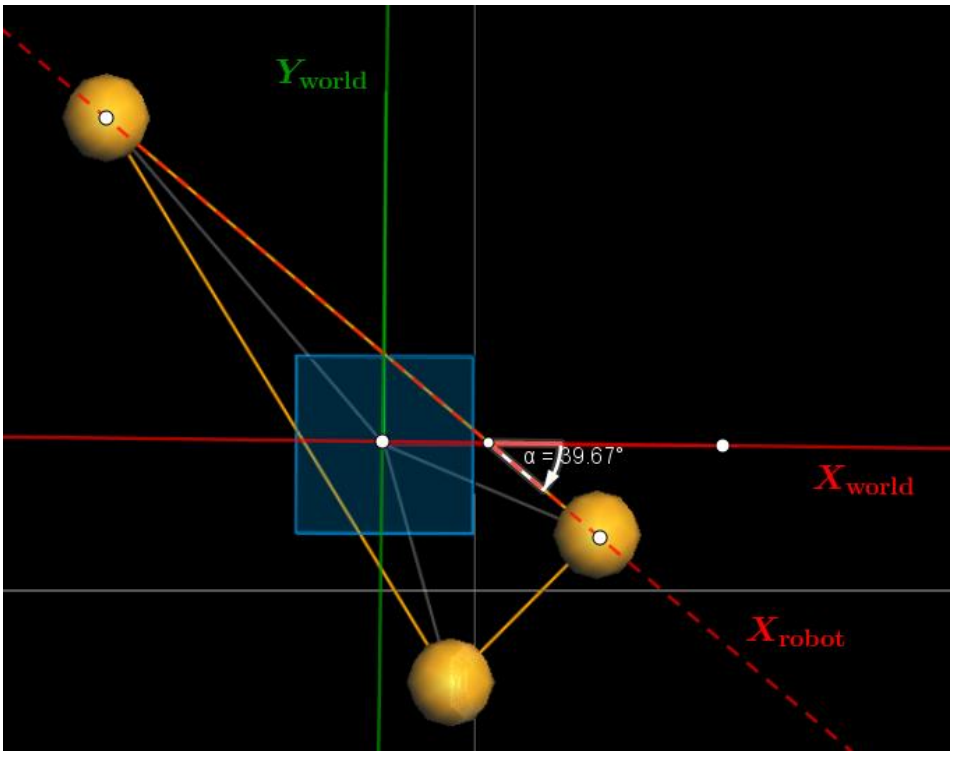
```
robot_origin(0) = -465.0/1000; // = robot_wrt_world(0) [m]
robot_origin(1) = 2015.7/1000; // = robot_wrt_world(1) [m]
robot_origin(2) = 447.6/1000; // = table_wrt_world(2) [m]
```

5. For the orientation of the robot with respect to the orientation of the world frame set by Vicon, we look again to the three markers we placed on the robot.
   The long segment is in the x-direction of the robot and the short segment in the y-direction of the robot. The red line denotes the x-axis and the green line denotes the y-axis of the world frame set by the Vicon wand.



From the positive Xworld axis to the positive Xrobot axis, we need to do an rotation with an angle alpha=40° about the negative Z axis, so with an angle of -40° about the positive z-axis.

$$\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} = \begin{bmatrix} \cos(-40°) & \sin(-40°) & 0 \\ -\sin(-40°) & \cos(-40°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix}$$

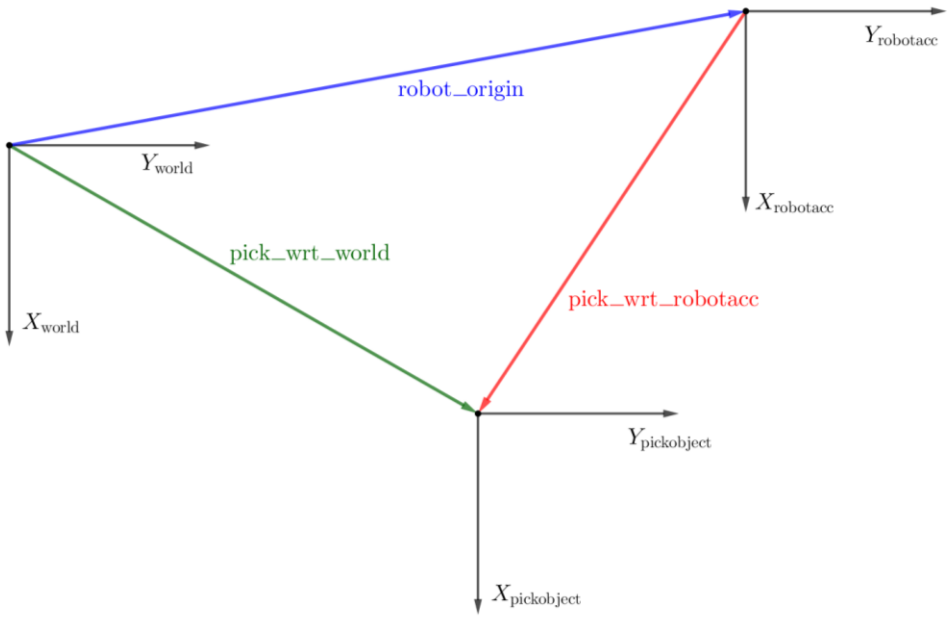


6. The “pickobject” and “placeobject” made in Vicon Tracker software with respect to the world frame (=set by the Vicon wand) can be received by the following commands.
   The marker number you use depends on the sequence you added the markers and the objects in Vicon Tracker.
   a) When creating a new object in Vicon Tracker, you have to select all the markers placed on the object. The sequence you use for selecting the markers “fixes” the number that marker gets.
   b) Selecting objects in Vicon Tracker also “fixes” the number that the markers get.
   c) In the pick and place example, the “origin” of each object was selected first and each object had four markers. When starting the experiment, I first selected my “pickobject” and afterwards my “placeobject”. That’s why I have to call markers[0] for the origin of my “pickobject” and markers[4] for the origin of my “placeobject”.

```
// coordinates of pick and place frame w.r.t. Vicon frame (Vicon frame = world frame)
// (check marker number)
pick_wrt_world(0) = msg->markers[0].translation.x; // [mm]
pick_wrt_world(1) = msg->markers[0].translation.y; // [mm]
pick_wrt_world(2) = msg->markers[0].translation.z; // [mm]
place_wrt_world(0) = msg->markers[4].translation.x; // [mm]
place_wrt_world(1) = msg->markers[4].translation.y; // [mm]
place_wrt_world(2) = msg->markers[4].translation.z; // [mm]
```

7. We don’t want the positions of our objects with respect to the world frame, but with respect to the robot frame. Therefore, we do the following transformations.

```
// coordinates of pick and place frame w.r.t. robot frame
pick_wrt_robotacc(0)=pick_wrt_world(0)/1000-robot_origin(0); // [m]
pick_wrt_robotacc(1)=pick_wrt_world(1)/1000-robot_origin(1); // [m]
pick_wrt_robotacc(2)=pick_wrt_world(2)/1000-robot_origin(2); // [m]
place_wrt_robotacc(0)=place_wrt_world(0)/1000-robot_origin(0); // [m]
place_wrt_robotacc(1)=place_wrt_world(1)/1000-robot_origin(1); // [m]
place_wrt_robotacc(2)=place_wrt_world(2)/1000-robot_origin(2); // [m]
```



And take into account the rotation about the z-axis with an angle alpha = -40°, see explanation in point 5, with “robotacc = world” (when only looking to orientation) and “robot = robot” to have the orientation of our object with respect to the real robot frame fixed by Franka.
https://frankaemika.github.io/docs/control_parameters.html#denavithartenberg-parameters

```
// take into account rotation: world frame -> rotation about z with -42° -> robot frame
pick_wrt_robot(0)=cos(alpha)*pick_wrt_robotacc(0)+sin(alpha)*pick_wrt_robotacc(1); // [m]
pick_wrt_robot(1)=-sin(alpha)*pick_wrt_robotacc(0)+cos(alpha)*pick_wrt_robotacc(1); // [m]
pick_wrt_robot(2)=pick_wrt_robotacc(2)+z_offset; // [m]
place_wrt_robot(0)=cos(alpha)*place_wrt_robotacc(0)+sin(alpha)*place_wrt_robotacc(1); // [m]
place_wrt_robot(1)=-sin(alpha)*place_wrt_robotacc(0)+cos(alpha)*place_wrt_robotacc(1); // [m]
place_wrt_robot(2)=place_wrt_robotacc(2)+z_offset; // [m]
```