ROSによるMarvelmindデバイスとの通信
（Robot Operating System）

バージョン 2022.11.20

注：ROS 2との統合については、こちらの別資料をご覧ください。

MarvelmindはROSパッケージ**marvelmind_nav**を提供しています。このパッケージは、モバイルビーコンまたはモデムと通信し、受信した位置データなどを提供できます。パッケージの最新版は、ROS Melodic（Ubuntu 18.04）およびROS Noetic（Ubuntu 20.04）向けにリリースし、テスト済みです。ROSパッケージは、次のリンクのソースリポジトリからも入手できます：

https://bitbucket.org/marvelmind_robotics/ros_marvelmind_package

ROSシステムにソースからパッケージをインストールするには、まず次の説明に従ってcatkinワークスペースを作成します：

http://wiki.ros.org/catkin/Tutorials/create_a_workspace

次に、ターミナルで以下のコマンドを実行して、パッケージ用のディレクトリを作成します：

```
$ cd ~/catkin_ws/src
$ mkdir marvelmind_nav
```

その後、リポジトリからダウンロードしたソースを、作成したディレクトリにコピーします。

パッケージのソフトウェアを実行する前に、'catkin_ws'ディレクトリで次のコマンドを実行してください：

```
$ source devel/setup.bash
```

その後、Marvelmind ROSパッケージをビルドしてインストールできます。'catkin_ws'ディレクトリで次を実行します：

```
$ catkin_make
$ catkin_make install
```

これでソフトウェアを実行する準備が整いましたが、その前にMarvelmindシステムを準備する必要があります。

操作マニュアルの説明に従い、Dashboardソフトウェアをインストールした別のPCでマップを構築します：

http://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf

次に、モバイルビーコン（hedgehog）をUSBケーブルでROSマシンのUSBポートに接続します。ターミナルで次のコマンドを実行し、モバイルビーコンが使用している仮想シリアルポートを確認します：

```
$ ls /dev/ttyACM*
```

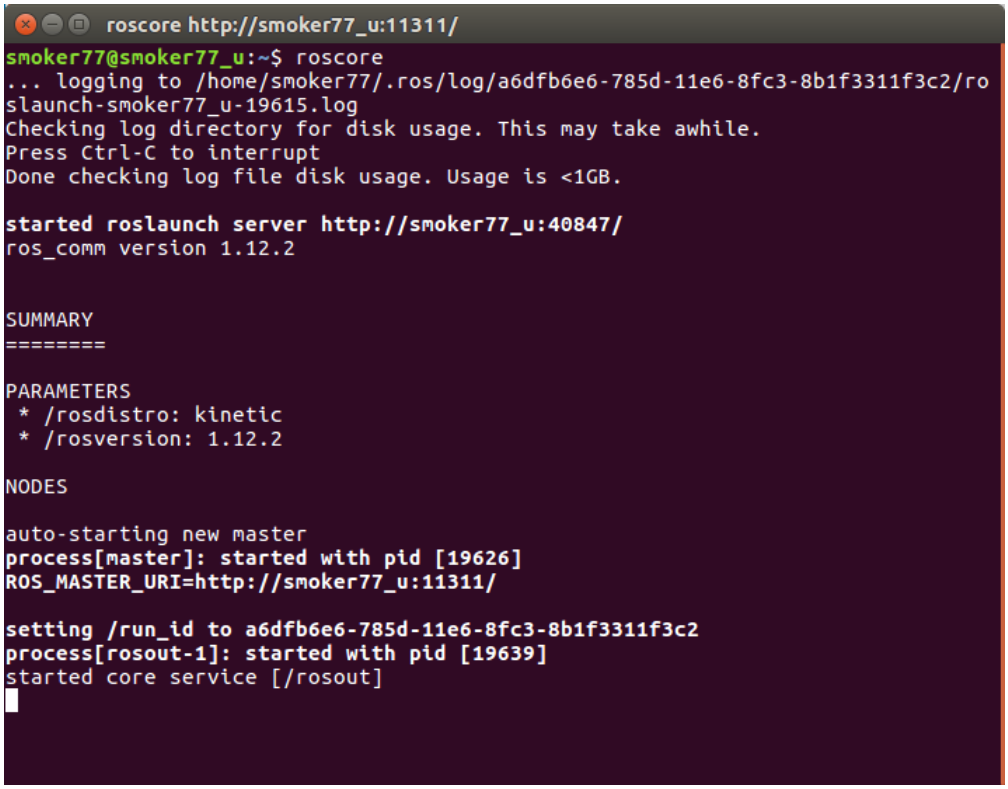
ほとんどの場合、モバイルビーコンは"/dev/ttyACM0"に接続されます。このポートは、Marvelmind ROSソフトウェアで既定で使用されます。このコマンドでポートが見つからない場合は、次のコマンドを試してください：

```
$ ls /dev/ttyUSB*
```

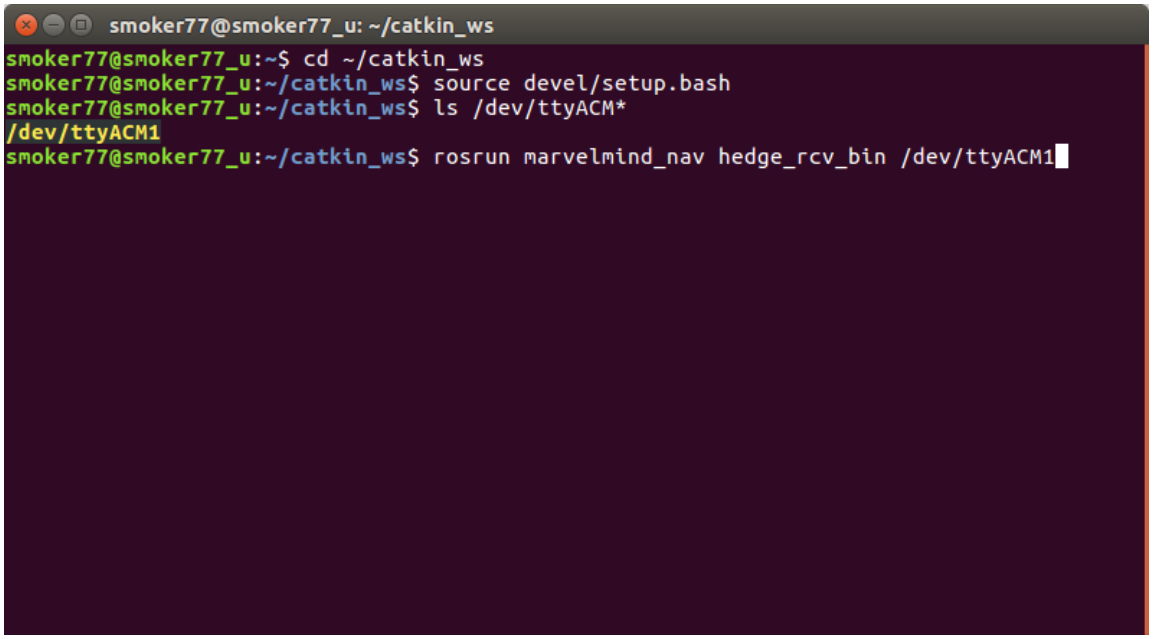
Marvelmind robotics

Marvelmind ROSソフトウェアを実行する前に、別のターミナルでROSサーバーを起動します：

$ roscore



次に、以下のスクリーンショットのように、モバイルビーコンからデータを受信するノード'**hedge_rcv_bin**'を実行します。実行するプログラムのパラメーターは'/dev/ttyACM1'で、これは前のコマンドで検出した仮想シリアルポートの名前です。ポートが'/dev/ttyACM0'の場合、このパラメーターは省略できます。

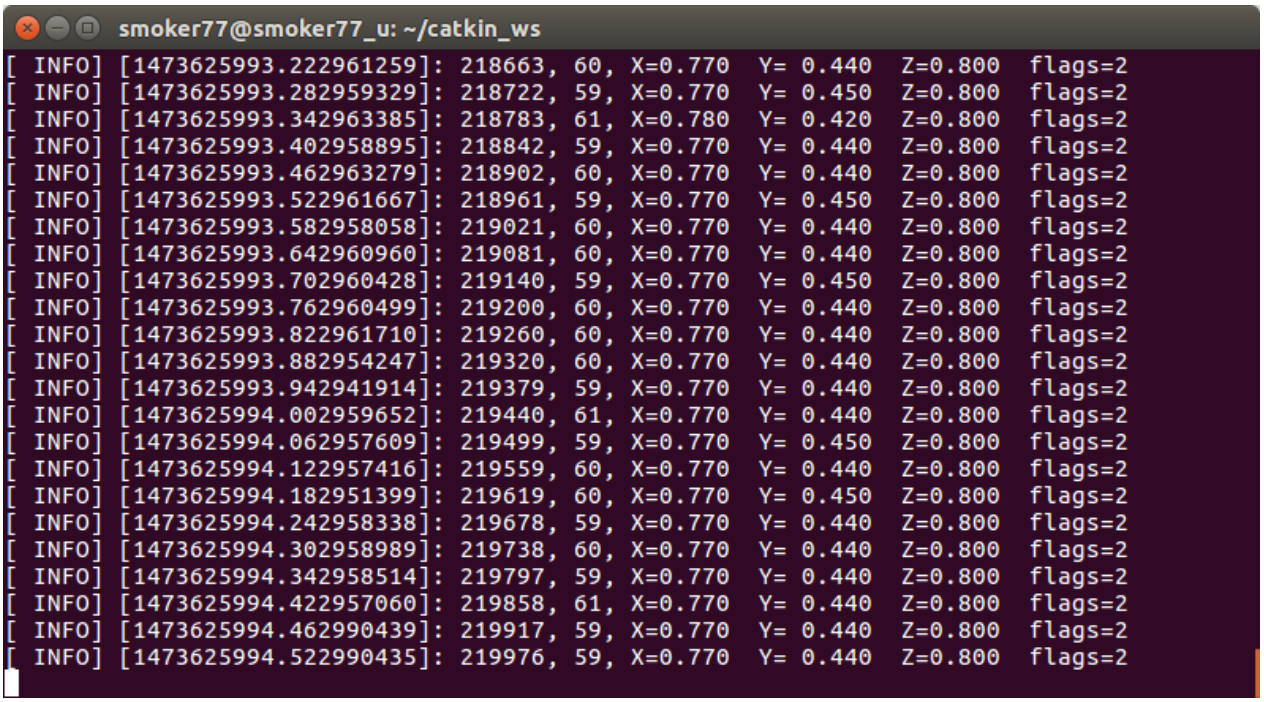


Marvelmind robotics

2番目のコマンドラインパラメーターでは、シリアルポートのボーレートを指定できます。例：**hedge_rcv_bin /dev/ttyACM1 115200**。USB接続では意味を持ちませんが、UART接続では、モバイルビーコン（またはモデム）のボーレート設定と一致させる必要があります。

シリアルポートが存在していても、'unable to open serial connection'のようなメッセージが表示されることがあります。これは、このポートへのアクセス権限がないことを意味している場合があります。コマンド'**sudo chmod 0777 /dev/ttyACM0**'ですべての権限を取得できますが、次回の再起動後には権限が失われます。恒久的な権限を得るには、次の説明に従ってユーザーを**dialout**グループに追加します：
https://askubuntu.com/questions/58119/changing-permissions-on-serial-port

ノードがモバイルビーコンからデータを正常に受信すると、スクリーンショットのように、位置データがターミナルに出力されます：



角括弧内の最初の値はROSのタイムスタンプで、続いてモバイルビーコンのタイムスタンプ（ミリ秒）、位置サンプル間の時間（ミリ秒）、X、Y、Z座標（メートル）、フラグのバイトです。

ノード'**hedge_rcv_bin**'はROSのパブリッシャーとして動作し、位置データを含むメッセージを、'**hedge_pos**'、'**hedge_pos_a**'、'**hedge_pos_ang**'という名前のトピックに送信します。
'**hedge_pos_ang**'は、このトピックの最新バージョンです。モバイルビーコンのアドレスと、ペアードビーコンの向きの角度が含まれます。

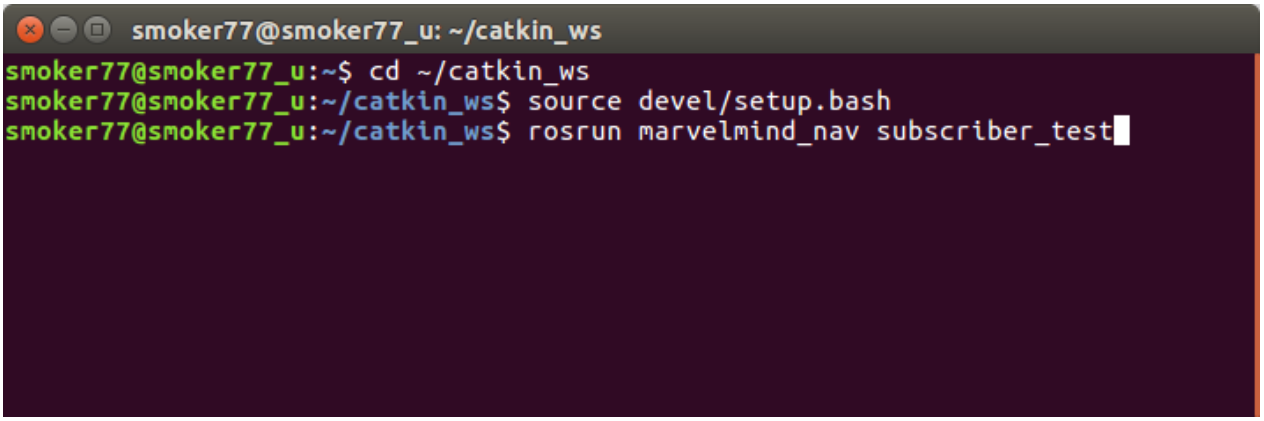
次の表に、すべてのトピックと、それらのトピックで取得できるデータを示します：

トピック	メッセージフィールド	型	説明
hedge_pos_ang	address	uint8	モバイルビーコンのアドレス
	timestamp_ms	uint32	位置のタイムスタンプ、ミリ秒
	x_m	float64	X座標、メートル
	y_m	float64	Y座標、メートル
	z_m	float64	Z座標、メートル

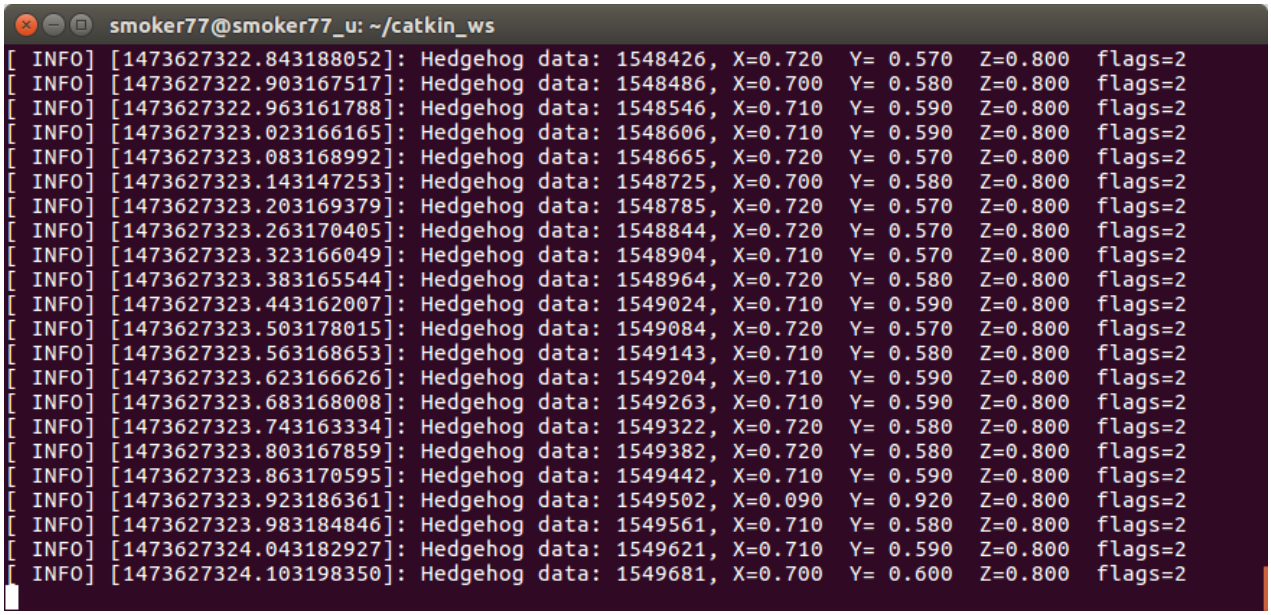
	flags	uint8	位置のフラグ
	angle	float64	ペアードビーコンの向きの角度、度
beacon_pos_a	address	uint8	固定ビーコンのアドレス
	x_m	float64	X座標、メートル
	y_m	float64	Y座標、メートル
	z_m	float64	Z座標、メートル
beacon_distance	address_hedge	uint8	モバイルビーコンのアドレス
	address_beacon	uint8	固定ビーコンのアドレス
	distance_m	float64	モバイルビーコンから固定ビーコンまでの生の距離、メートル
hedge_imu_fusion	timestamp_ms	int64	IMUフュージョンデータのタイムスタンプ、ミリ秒
	x_m	float64	IMUフュージョンによるモバイルビーコンの(X,Y,Z)座標、メートル。
	y_m	float64	
	z_m	float64	
	qw	float64	モバイルビーコンの姿勢クォータニオン(qw,qx,qy,qz)。正規化済み（$qw^2+qx^2+qy^2+qz^2=1$）
	qx	float64	
	qy	float64	
	qz	float64	
	vx	float64	(vx, vy, vz) – IMUフュージョンで計算したモバイルビーコンの速度ベクトル、メートル/秒
	vy	float64	
	vz	float64	
	ax	float64	(ax, ay, az) – モバイルビーコンの加速度、メートル/秒^2
	ay	float64	
	az	float64	
hedge_imu_raw	timestamp_ms	int64	IMU生データのタイムスタンプ、ミリ秒
	acc_x	int16	(acc_x, acc_y, acc_z) – 加速度センサーの生データ、1 mg/LSB
	acc_y	int16	
	acc_z	int16	
	gyro_x	int16	(gyro_x, gyro_y, gyro_z) – ジャイロスコープの生データ、0.0175 dps/LSB
	gyro_y	int16	
	gyro_z	int16	
	compass_x	int16	(compass_x, compass_y, compass_z) – コンパスの生データ（HW4.9のビーコンのみ）。 X,Y: 1100 LSB/Gauss Z: 980 LSB/Gauss
	compass_y	int16	
	compass_z	int16	
hedge_quality	address	uint8	モバイルビーコンのアドレス
	quality_percents	uint8	位置の品質、パーセント
hedge_telemetry	battery_voltage	float64	モバイルビーコンのバッテリー電圧、ボルト
	rssi_dbm	int8	RSSI（無線信号の強度）、dBm

marvelmind_waypoint	total_items	uint8	ウェイポイントプログラムの項目の総数（N）
	item_index	uint8	このウェイポイント項目のインデックス（0...N-1）
	movement_type	uint8	動作の種類（6 = 指定したポイントへ移動）
	param1	int16	パラメーター1（movement_typeによって異なる） type= 6の場合：ウェイポイントのX座標、cm
	param2	int16	パラメーター2（movement_typeによって異なる） type= 6の場合：ウェイポイントのY座標、cm
	param3	int16	パラメーター3（movement_typeによって異なる） type= 6の場合：ウェイポイントのZ座標、cm

このパッケージには、もう1つのノード'**subscriber_test**'も含まれています。このノードはROSのサブスクライバーとして動作し、すべてのトピックからデータを受信します。テスト用として、またユーザーソフトウェアのベースとして使用できます。
以下のスクリーンショットのように、別のターミナルで'**subscriber_test**'ノードを実行します：



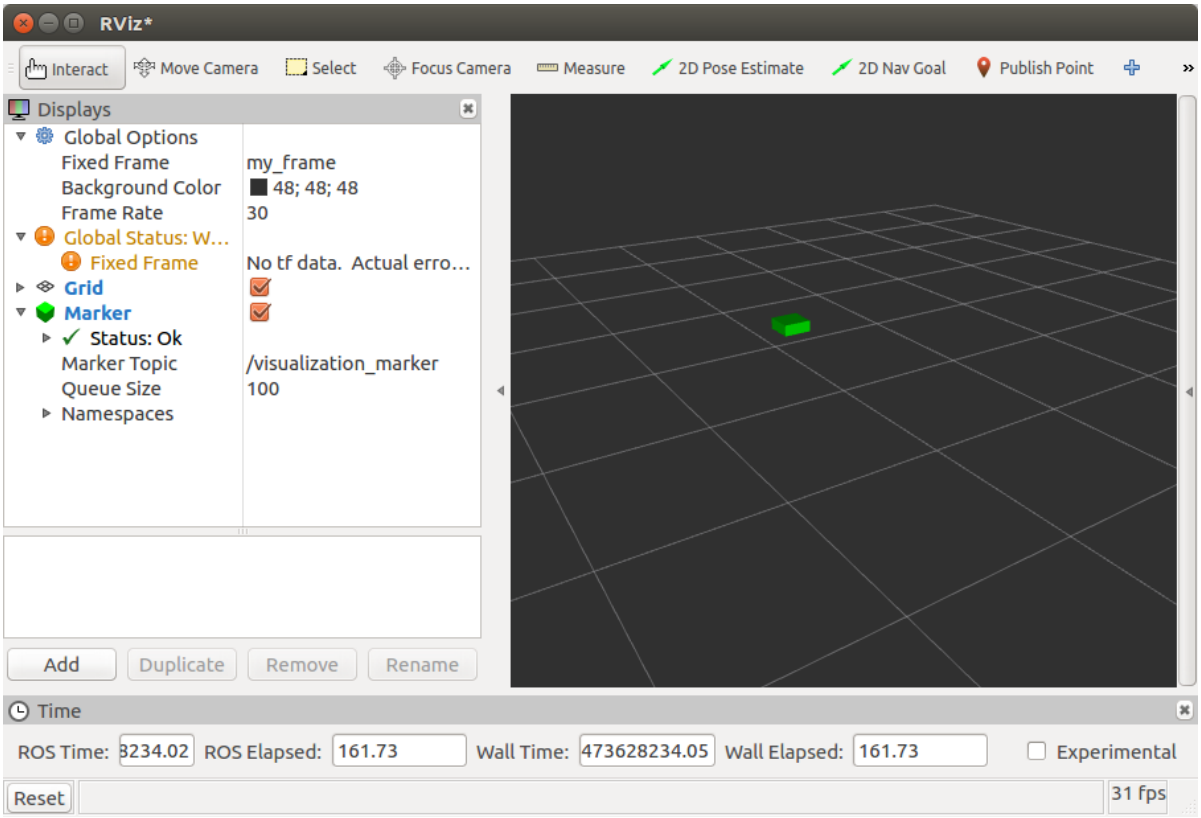
実行中の'**subscriber_test**'ノードは、次のスクリーンショットのように、トピック'**/hedge_pos**'から受信した位置データをターミナルに出力します：



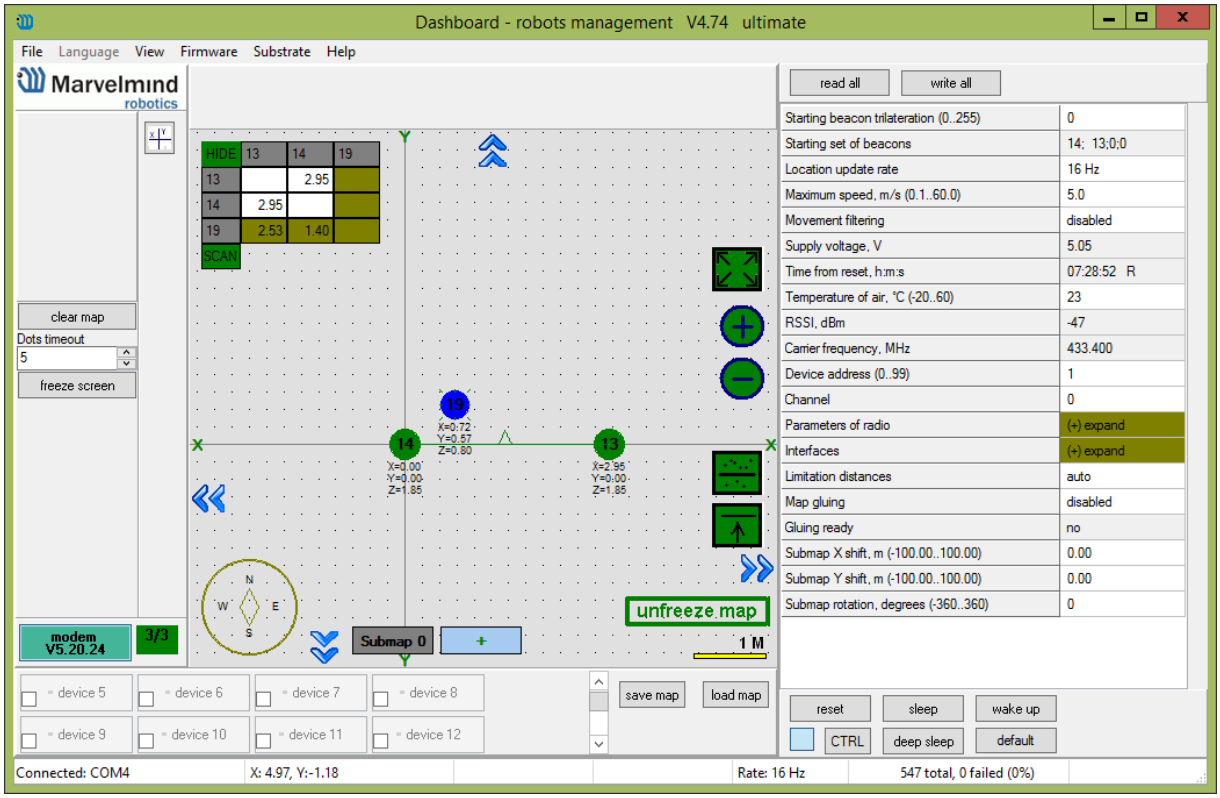
さらに、このノードはパブリッシャーとしても動作し、トピック"**visualization_marker**"にデータを送信します。これにより、標準のROSソフトウェア'**rviz**'で位置を表示できます。
スクリーンショットのように、別のターミナルで'rviz'を実行します：



次のスクリーンショットに示すGUIウィンドウが表示されます。
モバイルビーコンを表示するには、'Fixed frame'パラメーターの値が'my_frame'であり、マーカー'visualization marker'が接続されていることを確認してください。



次のスクリーンショットは、上記のROSのテストに使用したシステムを接続した別のコンピューター上の、Dashboardのウィンドウです。


Dashboard - robots management V4.74 ultimate
File Language View Firmware Substrate Help
Marvelmind robotics
read all
write all
Starting beacon trilateration (0..255) 0
Starting set of beacons 14; 13;0;0
Location update rate 16 Hz
Maximum speed, m/s (0.1..60.0) 5.0
Movement filtering disabled
Supply voltage, V 5.05
Time from reset, h:m:s 07:28:52 R
Temperature of air, °C (-20..60) 23
RSSI, dBm -47
Carrier frequency, MHz 433.400
Device address (0..99) 1
Channel 0
Parameters of radio (+) expand
Interfaces (+) expand
Limitation distances auto
Map gluing disabled
Gluing ready no
Submap X shift, m (-100.00..100.00) 0.00
Submap Y shift, m (-100.00..100.00) 0.00
Submap rotation, degrees (-360..360) 0
HIDE 13 14 19
13 2.95
14 2.95
19 2.53 1.40
SCAN
clear map
Dots timeout
5
freeze screen
modem V5.20.24
3/3
Submap 0
unfreeze map
1 M
save map
load map
reset
sleep
wake up
CTRL
deep sleep
default
device 5
device 6
device 7
device 8
device 9
device 10
device 11
device 12
Connected: COM4
X: 4.97, Y:-1.18
Rate: 16 Hz
547 total, 0 failed (0%)