Communication with Marvelmind devices using ROS 2 (Robot Operating System).

Version 2022.11.30

Note: for integration with ROS 1 see this separate document.

Marvelmind supplies ROS 2 package **marvelmind_ros2** (with separate messages package **marvelmind_ros2_msgs**), which is able to communicate with mobile beacon or modem and provide received location and other data. We have tested latest version of the package for ROS 2 Humble (tested under Ubuntu 22.04 and Windows 10). The ROS 2 packages source are available in repositories by links:
https://github.com/MarvelmindRobotics/marvelmind_ros2_upstream
https://github.com/MarvelmindRobotics/marvelmind_ros2_msgs_upstream

Before installation the Marvelmind ROS 2 package, please install ROS 2 as described here (if you have not installed it previously): https://docs.ros.org/en/humble/Installation.html

Also, before using the ROS 2 package you will need to prepare the Marvelmind system.
Use another PC with dashboard software to build map as described in the operating manual:
http://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf
You should get a tracking in the dashboard. If you have a tracking, the location data should be available through the ROS 2 package.

To install the package in the ROS system from source, at first create a workspace folder, for example '**ros2_ws**', and place there two folders with the packages source: folders '**marvelmind_ros2**' and '**marvelmind_ros2_msgs**'.

1.1. Installation and running under Linux

Open the terminal.
Type in the terminal:

```
cd ~/ros2_ws
source /opt/ros/humble/setup.bash
```

Connect mobile beacon or modem to the computer via USB.
Execute command in terminal to find the virtual serial port used by the modem of the mobile beacon:

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

In most cases, the hedgehog connects to "/dev/ttyACM0", this port is used by default in the Marvelmind ROS software. If no ports found by this command, try another one:

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

Make sure you have permissions to access this port. You can get all permissions by command '**sudo chmod 0777 /dev/ttyACM0**'. But you will lose the permissions after next reboot. For permanent permissions you can add user to **dialout** group as described here: https://askubuntu.com/questions/58119/changing-permissions-on-serial-port
When you got the permissions, set the serial port setting 'marvelmind_tty_filename' in the configuration file: /marvelmind_ros2/config/marvelmind_ros2_config.yaml
After the installation, you will be able to modify this setting via ROS 2 parameters commands. Then type in the terminal

```
sudo apt install python3-colcon-common-extensions
```

```
colcon build
```

‘Colcon build’ should report successful build of the packages. Then type in terminal:

```
. install/setup.bash
ros2 launch marvelmind_ros2 marvelmind_ros2.launch.py
```

Don’t forget a space between a dot “.” and “install/setup.bash”.

‘Hedgehog is running’ message should appear. This means ROS2 package is connected to the hedgehog (or modem).

Then open another terminal in the workspace directory and type:

```
. install/setup.bash
source /opt/ros/humble/setup.bash
ros2 topic echo /hedgehog_pos_ang
```

You should get location data of the active mobile beacon from the specified topic

1.2. Installation and running under Windows 10.

Open command prompt from Visual Studio ‘Tools/Command Line’ menu.

Connect mobile beacon or modem to the computer via USB.

Check the serial port name used by the mobile beacon or modem. You can run the dashboard, see the serial port name in the left bottom angle and then close the dashboard.

Set the serial port setting ‘marvelmind_tty_filename’ in the configuration file: /marvelmind_ros2/config/marvelmind_ros2_config.yaml

After the installation, you will be able to modify this setting via ROS 2 parameters commands.

If your ROS2 installation path is ‘c:/dev/ros2_humble’ and ROS2 workspace path is ‘c:/ros2_ws’, then type in the command shell:

```
cd c:/ros2_ws
call c:/dev/ros2_humble/local_setup.bat
colcon build --merge-install
```

‘Colcon build’ should report successful build of the packages. Then type in the command shell:

```
call install/local_setup.bat
ros2 launch marvelmind_ros2 marvelmind_ros2.launch.py
```

‘Hedgehog is running’ message should appear. This means ROS2 package is connected to the hedgehog (or modem).

Then open another command shell from Visual Studio ‘Tools/Command Line’ menu and type:

```
cd c:/ros2_ws
call c:/dev/ros2_humble/local_setup.bat
call install/local_setup.bat
 ros2 topic echo /hedgehog_pos_ang
```

You should get location data of the active mobile beacon from the specified topic

Marvelmind
robotics

Following table lists all topics and data available via these topics:

Topic	Message field	Type	Description
hedge_pos_ang	address	uint8	Address of mobile beacon
	timestamp_ms	uint32	Timestamp of location, milliseconds
	x_m	float64	X coordinate, meters
	y_m	float64	Y coordinate, meters
	z_m	float64	Z coordinate, meters
	flags	uint8	flags of location
	angle	float64	Orientation angle of paired beacons, degrees
beacon_pos_a	address	uint8	Address of stationary beacon
	x_m	float64	X coordinate, meters
	y_m	float64	Y coordinate, meters
	z_m	float64	Z coordinate, meters
beacon_distance	address_hedge	uint8	Address of mobile beacon
	address_beacon	uint8	Address of stationary beacon
	distance_m	float64	Raw distance from mobile to stationary beacon, meters
hedge_imu_fusion	timestamp_ms	int64	Timestamp of IMU fusion data, milliseconds
	x_m	float64	(X,Y,Z) coordinates of mobile beacon by IMU fusion. meters.
	y_m	float64	
	z_m	float64	
	qw	float64	Orientation quaternion of mobile beacon (qw,qx,qy,qz). Normalized $(qw^2+qx^2+qy^2+qz^2=1)$
	qx	float64	
	qy	float64	
	qz	float64	
	vx	float64	(vx, vy, vz) – speed vector of mobile beacon calculated by IMU fusion, meters/s
	vy	float64	
	vz	float64	
	ax	float64	(ax, ay, az) – acceleration of mobile beacon meters/s^2
	ay	float64	
	az	float64	
hedge_imu_raw	timestamp_ms	int64	Timestamp of raw IMU data, milliseconds
	acc_x	int16	(acc_x, acc_y, acc_z) – raw accelerometer data, 1 mg/LSB
	acc_y	int16	
	acc_z	int16	
	gyro_x	int16	(gyro_x, gyro_y, gyro_z) – raw gyroscope data, 0.0175 dps/LSB
	gyro_y	int16	
	gyro_z	int16	
	compass_x	int16	

	compass_y	int16	(compass_x, compass_y, compass_z) – raw compass data (only for HW4.9 beacons). X,Y: 1100 LSB/Gauss Z: 980 LSB/Gauss
	compass_z	int16	
hedge_quality	address	uint8	Address of the mobile beacon beacon
	quality_percents	uint8	Quality of location, percents
hedge_telemetry	battery_voltage	float64	Battery voltage of the mobile beacon, volts
	rssi_dbm	int8	RSSI (radio signal strength), dBm
marvelmind_waypoint	total_items	uint8	Total number of waypoint program items (N)
	item_index	uint8	Index of this waypoint item (0…N-1)
	movement_type	uint8	Type of action (6 = move to specified point)
	param1	int16	Parameter 1 (depends from movement_type) X coordinate of waypoint, cm if type= 6
	param2	int16	Parameter 2 (depends from movement_type) Y coordinate of waypoint, cm if type= 6
	param3	int16	Parameter 3 (depends from movement_type) Z coordinate of waypoint, cm if type= 6