Team members:
Alessandro De Luca 10676114
Andrea Ceruti 10557963
Samuele Panzeri 10525892

Files:
gmapping.launch - launch file used for the creation of the map, holds the instruction to start gmapping, use sim time and the estimated tf between laser and scout/base_link (95 degrees over z-axis)

gmapping.launch.xml - configuration file that holds the parameters used in gmapping

amcl.launch - launch file used in the localization part of the project, holds all the static tf, the imu_filter node, the ekf node and the amcl node

amcl.launch.xml - configuration file that holds the parameters used in amcl

PS: all the launch files are expected to run on the remapped bags, not the old ones

TF tree:
The final TF tree has

map -> odom, created by amcl node
odom -> scout/odom / camera_odom_frame, static tf to correct some warnings
odom -> base_link, created by the extended kalman filter node
base_link -> laser, a static tf of a rotation of 95 degrees

The static tf for the camera frame taken from the old bags:
camera_pose_frame -> camera_link
camera_link -> camera_gyro_frame / camera_accel_frame
camera_gyro_frame -> camera_gyro_optical_frame
camera_accel_frame -> camera_accel_optical_frame

The complete TF tree can be seen in the frames.png file in the package

The map was computed using the first bag and robot localization was executed on bag two and three.

How to use the nodes:
To use the gmapping launch file write “roslaunch project2 gmapping.launch” on your command line having both gmapping.launch and gmapping.launch.xml in the project2/launch folder and then start bag number one (the one used in gmapping).

To use the amcl launch file write “roslaunch project2 amcl.launch” on your command line having both amcl.launch and amcl.launch.xml in the project2/launch folder and then start either bag number two or bag number three.

PS: all the launch files are expected to run on the remapped bags, not the old ones

Sensors used as EKF sources:
odom0 = /odom : odometry of the manufacturer, we used linear and angular velocity and also orientation theta.

odom1 = /camera/odom/sample : odometry of the camera, we used linear and angular velocity so that the kalman filter had two sources of odometry.

For the odometries we have chosen to use only the linear velocity in x and the angular velocity in yaw since other data are redundant. In fact, as we have seen in the first project, the position of the robot (x,y) is calculated by integrating the velocities in the sampling interval. So it is safer to use just the velocity and not the position, For the odometry of the manufacturer we have considered also the yaw to true because we have noticed that with this configuration the best performances are achieved.

imu0 = /imu/data : data gathered from the imu filter applied to mavros/imu/data_raw at first we wanted to use all of the data gathered from the imu filter meaning orientation, angular velocity and linear acceleration, but when using the orientation we noticed that the localization went bad pretty soon so we settled on using only angular velocity and linear acceleration.

imu1 = /camera/accel/sample : accelerometer of the camera, thus we set the linear acceleration to true.

imu0 = /camera/gyro/sample : gyroscope of the camera, thus we set the angular velocity to true.

We decided to use this configuration with the sensors because after a lot of fine tuning and experiments we have noticed that the kalman filter could compute the best transformation possible between odom and base_link.