

## 4、 Robot state estimation

By using the IMU module on the ROS expansion board and the encoder of the wheels, the current position and posture of the car can be estimated, which plays an important role in map building and navigation.

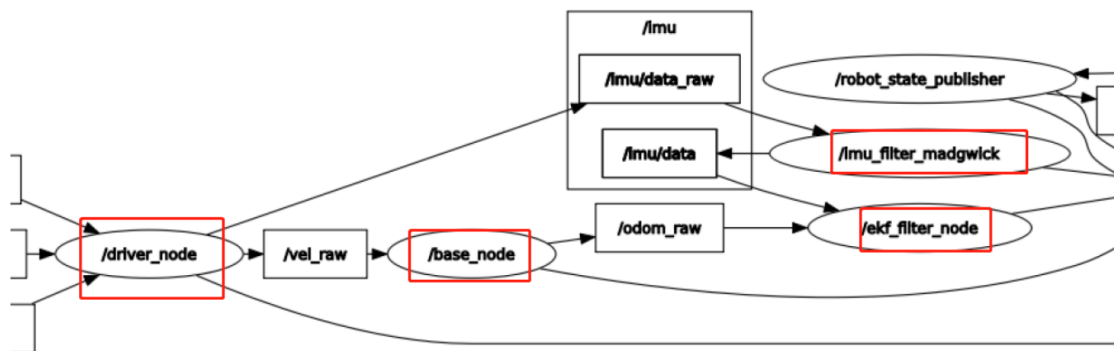
### 1、 Run sample code

Taking our company's Rosmaster-X3 as an example, the terminal inputs commands,

```
ros2 launch yahboomcar_bringup yahboomcar_bringup_X3_launch.py
```

### 2、 View the node communication diagram

```
ros2 run rqt_graph rqt_graph
```



Mainly by looking at the input and output of the nodes in the red box in the above figure, it can be seen that **/ekf\_Filter\_Node** receiving **odom\_Raw** data and **imu\_The data** is fused, and finally an **ODOM** data is output and published. We can view it through the ROS2 node tool, with terminal input,

```
ros2 node info /ekf_filter_node
```

### 3、 Launch file parsing

Let's take a look at the main related nodes of the launch file

- **/driver\_node**: Start the car chassis, obtain the speed **vel** data of the wheels, and publish it to **/base\_Node** node, obtain IMU data, publish to **/IMU\_Filter\_Madgwick** node;
- **/base\_node**: Receive **vel** data, calculate and convert it into **odom\_Raw** data, published to **/ekf\_Filter\_Node**;
- **/Imu\_filter\_madgwick**: Receive the **imu** data released by the chassis, filter it through its own algorithm, and publish the filtered **imu/data** data to **/ekf\_Filter\_Node**;

- /ekf\_filter\_node: Receive/base\_odom data and/Imu published by node nodes\_Filter\_Madgwick publishes imu/data data data through its own algorithm, and after fusion, publishes odom data.