

## 4、 Robot state estimation

### 1、 Program function description

After the program runs, combined with the IMU data and speed vel data read to the ROS expansion board, an ODOM data that integrates IMU and ODOM data is output, which is applied when doing positioning functions.

### 2、 Program code reference path

After entering the docker container, the location of the source code of this function is located at,

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/launch/yahboomcar_bringup_X3_launch.py
```

ekf fusion program code reference path,

```
/root/yahboomcar_ros2_ws/software/library_ws/src/robot_localization/launch/ekf.launch.h.py
```

## 3、 The program starts

### 3.1、 start the command

After entering the docker container, according to the actual model, the terminal input,

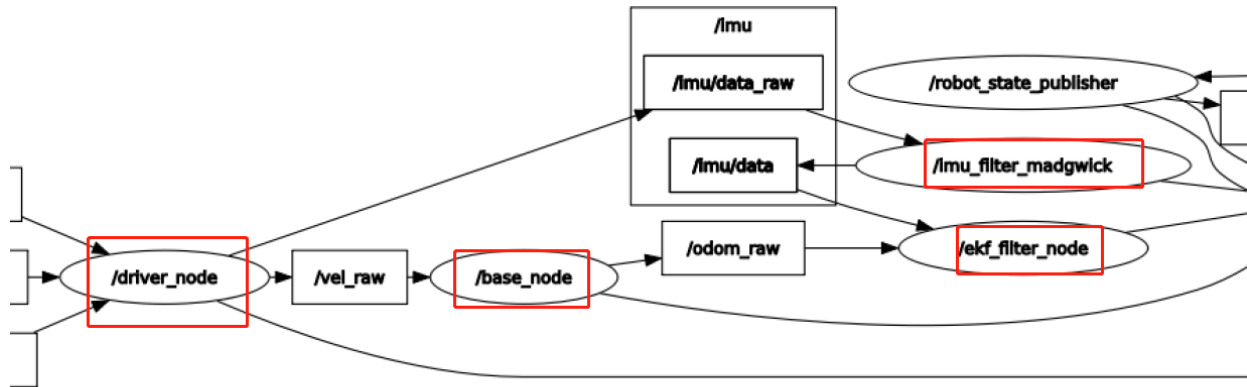
```
ros2 launch yahboomcar_bringup yahboomcar_bringup_X3_launch.py
```

```
root@jetson-desktop:~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/la
unch# ros2 launch yahboomcar_bringup yahboomcar_bringup_X3_launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2023-04-19-09-56-24-819143-jetson-desktop-1768
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [Mcnamu_driver_X3-1]: process started with pid [1771]
[INFO] [base_node_X3-2]: process started with pid [1773]
[INFO] [joint_state_publisher-3]: process started with pid [1775]
[INFO] [robot_state_publisher-4]: process started with pid [1777]
[INFO] [imu_filter_madgwick_node-5]: process started with pid [1779]
[INFO] [ekf_node-6]: process started with pid [1781]
[INFO] [yahboom_joy_X3-7]: process started with pid [1783]
[robot_state_publisher-4] Parsing robot urdf xml string.
[robot_state_publisher-4] Link base_link had 7 children
[robot_state_publisher-4] Link back_left_wheel had 0 children
[robot_state_publisher-4] Link back_right_wheel had 0 children
[robot_state_publisher-4] Link imu_link had 0 children
[robot_state_publisher-4] Link camera_link had 0 children
[robot_state_publisher-4] Link front_left_wheel had 0 children
[robot_state_publisher-4] Link front_right_wheel had 0 children
[robot_state_publisher-4] Link laser_link had 0 children
[robot_state_publisher-4] [INFO] [1681898185.579604218] [robot_state_publisher]: got segment back_left_wheel
[robot_state_publisher-4] [INFO] [1681898185.579911415] [robot_state_publisher]: got segment back_right_wheel
[robot_state_publisher-4] [INFO] [1681898185.579966460] [robot_state_publisher]: got segment base_footprint
[robot_state_publisher-4] [INFO] [1681898185.580066707] [robot_state_publisher]: got segment base_link
[robot_state_publisher-4] [INFO] [1681898185.580107430] [robot_state_publisher]: got segment camera_link
[robot_state_publisher-4] [INFO] [1681898185.580137947] [robot_state_publisher]: got segment front_left_wheel
[robot_state_publisher-4] [INFO] [1681898185.580167006] [robot_state_publisher]: got segment front_right_wheel
[robot_state_publisher-4] [INFO] [1681898185.580196533] [robot_state_publisher]: got segment imu_link
[robot_state_publisher-4] [INFO] [1681898185.580226945] [robot_state_publisher]: got segment laser_link
[imu_filter_madgwick_node-5] [INFO] [1681898185.854631948] [imu_filter_madgwick]: Starting ImuFilter
[imu_filter_madgwick_node-5] [INFO] [1681898185.864917217] [imu_filter_madgwick]: Using dt computed from message headers
[imu_filter_madgwick_node-5] [INFO] [1681898185.869950491] [imu_filter_madgwick]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-5] [INFO] [1681898185.875814538] [imu_filter_madgwick]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-5] [INFO] [1681898185.878382210] [imu_filter_madgwick]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-5] [INFO] [1681898185.880314812] [imu_filter_madgwick]: Magnetometer bias values: 0.000000 0.000000 0.000000
[joint_state_publisher-3] [INFO] [1681898187.746356090] [joint_state_publisher]: Waiting for robot_description topic...
[imu_filter_madgwick_node-5] [INFO] [1681898187.773215259] [imu_filter_madgwick]: First IMU message received.
```

## 3.2, View the node communication graph

Open the terminal and enter the container,

```
ros2 run rqt_graph rqt_graph
```



Mainly look at the node input and output of the red box in the figure above, you can see that **/ekf\_filter\_node** receive **odom\_raw** data and **imu\_data** data for fusion, and finally the output publishes an ODOM data, we can view through the **ros2 node tool**, terminal input,

```
ros2 node info /ekf_filter_node
```

```
root@jetson-desktop:~# ros2 node info /ekf_filter_node
/ekf_filter_node
Subscribers:
  /example/odom2: nav_msgs/msg/Odometry
  /example/pose: geometry_msgs/msg/PoseWithCovarianceStamped
  /example/twist: geometry_msgs/msg/TwistWithCovarianceStamped
  /imu/data: sensor_msgs/msg/Imu
  /odom_raw: nav_msgs/msg/Odometry
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /set_pose: geometry_msgs/msg/PoseWithCovarianceStamped
Publishers:
  /diagnostics: diagnostic_msgs/msg/DiagnosticArray
  /odom: nav_msgs/msg/Odometry
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /rosout: rcl_interfaces/msg/Log
  /tf: tf2_msgs/msg/TFMessage
Service Servers:
  /ekf_filter_node/describe_parameters: rcl_interfaces/srv/DescribeParameters
  /ekf_filter_node/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
  /ekf_filter_node/get_parameters: rcl_interfaces/srv/GetParameters
  /ekf_filter_node/list_parameters: rcl_interfaces/srv/ListParameters
  /ekf_filter_node/set_parameters: rcl_interfaces/srv/SetParameters
  /ekf_filter_node/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  /enable: std_srvs/srv/Empty
  /set_pose: robot_localization/srv/SetPose
  /toggle: robot_localization/srv/ToggleFilterProcessing
Service Clients:

Action Servers:

Action Clients:
```

## 4、Launch file parsing

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

- /driver\_node: Start the chassis of the trolley, obtain the speed vel data of the wheels, publish to the /base\_node node, obtain IMU data, and publish to the /imu\_filter\_madgwick node;
- /base\_node: Receive vel data, convert it into odom\_raw data through calculation, and publish it to /ekf\_filter\_node nodes;
- /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 the /ekf\_filter\_node node;
- /ekf\_filter\_node: Receive the ODOM data published by the /base\_node node and the IMU/Data data released by the /imu\_filter\_madgwick, and release the ODOM data after fusing it through its own algorithm.