

## 4. Robot state estimation

### 1. Program Function Description

After the program runs, it combines the imu data and speedvel data read from the ROS expansion board, and outputs an odom data which combines the imu and odom data, and this data is applied when doing the positioning function.

### 2. Program Code Reference Path

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

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/launch/yahboomcar_bringup_R2_launch.py
```

ekf fusion program code reference path.

```
/root/yahboomcar_ros2_ws/software/library_ws/src/robot_localization/launch/ekf.1aunch.py
```

### 3. Program startup

#### 3.1. Startup commands

After entering the docker container, depending on the actual model, terminal input, the

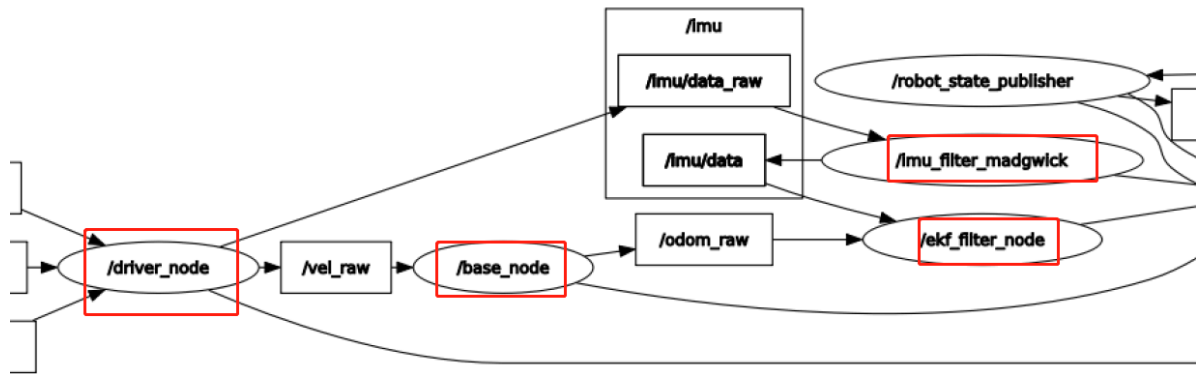
```
ros2 launch yahboomcar_bringup yahboomcar_bringup_R2_launch.py
```

```
root@jetson-desktop:~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/la
nch$ ros2 launch yahboomcar_bringup yahboomcar_bringup_R2_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] [1681898185.580066797] [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 to be published on the robot_description topic...
[imu_filter_madgwick_node-5] [INFO] [1681898187.773215259] [imu_filter_madgwick]: First IMU message received.
```

#### 3.2. Viewing the node communication graph

Open the terminal and enter the container and type,

```
ros2 run rqt_graph rqt_graph
```



The main thing is to look at the node inputs and outputs in the red box of the above figure, as you can see, /ekf\_filter\_node receives the odom\_raw data and the imu\_data data for fusion, and finally outputs the release of an odom data, which we can look at through the ros2 node tool with the terminal inputs,

```
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. Parsing the launch file

Let's look at the main relevant nodes of the launch file.

- /driver\_node: start the chassis of the cart, get the velocity vel data of the wheel, publish it to /base\_node node, get the imu data, publish it to /Imu\_filter\_madgwick node; /base\_node: receive the vel data, through calculation, convert it to odom\_raw data, publish it to /Imu\_filter\_madgwick node.
- /base\_node: receive vel data, through calculation, convert to odom\_raw data, publish to /ekf\_filter\_node node;

- /Imu\_filter\_madgwick: receive imu data posted by chassis, filter it through its own algorithm, and post the filtered imu/data data to /ekf\_filter\_node node; /ekf\_filter\_node.
- /ekf\_filter\_node: receive odom data released by /base\_node node and imu/data data released by /Imu\_filter\_madgwick, through its own algorithm, after fusion, release odom data.