

Autoware

[Introduction](#)

[Localization](#)

[Detection](#)

[Tracking and Prediction](#)

[Planning](#)

[Tutorial](#)

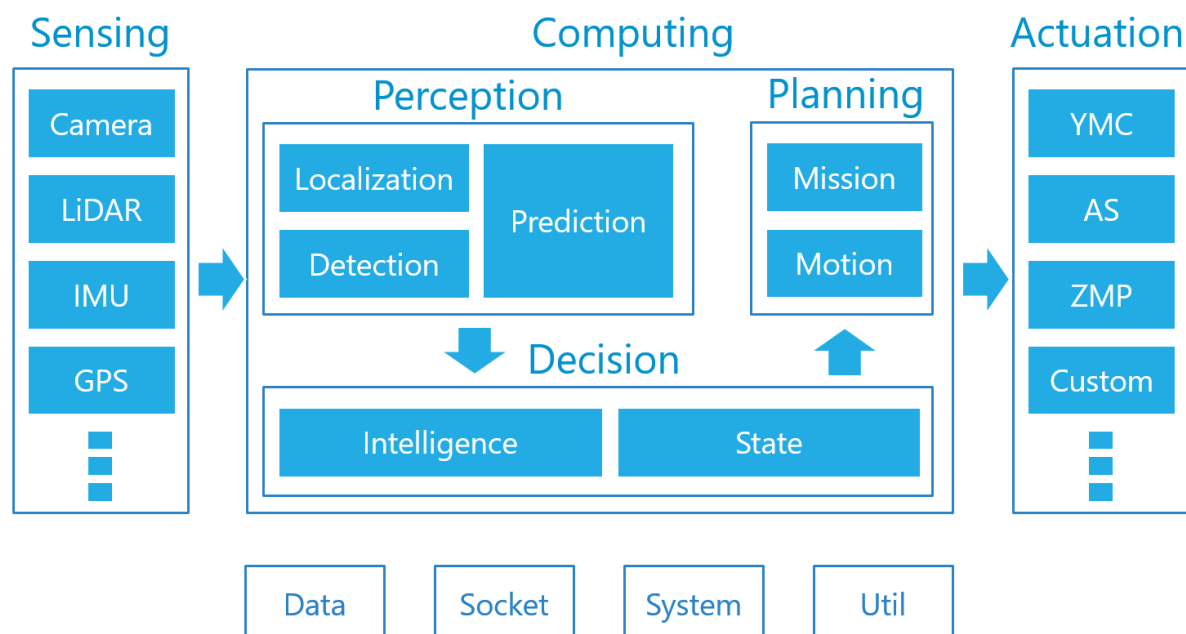
[Test localization](#)

Introduction

Autoware.AI is **ROS-based open-source software**, enabling self-driving vehicles to be tested in private areas, urban roads, and highways. Another variant of Autoware, *a.k.a.*, [Autoware.Auto](#), is also under development for the purpose of commercial deployment of self-driving vehicles with functional safety capabilities.

[Refer to Autoware Wiki for Users Guide and Developers Guide.](#)

The current version of Autoware.AI provides the functional modules described below.



Localization

Localization depends on **3D high-definition map** data and the **NDT** algorithm. The result of *Localization* is complemented by the **Kalman Filter** algorithm, using odometry information obtained from **CAN** messages and **GNSS/IMU** sensors.

Detection

Detection is empowered by **camera** and **LiDAR** devices in combination with **3D high-definition map** data. The *Detection* module uses **deep learning** and **sensor fusion** approaches.

Tracking and Prediction

Tracking and *Prediction* are realized with the **Kalman Filter** algorithm and the lane network information provided by **3D high-definition map** data.

Planning

Planning is based on **probabilistic robotics** and **rule-based systems**, partly using **deep learning** approaches as well.

Control defines motion of the vehicle with a twist of **velocity** and **angle** (also **curvature**). The *Control* module falls into both the Autoware-side stack and the vehicle-side interface (**PID** variants).

Tutorial

There are two ways to install Autoware: Docker or build from source. I recommend building from source which is more straightforward and no need some external drivers.

<https://github.com/Autoware-AI/autoware.ai/wiki/Source-Build>

System Variables:

- ☐ Autoware version: 1.12.0
- ☐ ROS Version: Melodic 18.04
- ☐ Cuba: 10.0 (important**,only support ≤ 10.0)

☐ Autoware working directory: /home/autoware.ai

```
# check nvidia system/cuda version:
nvidia-smi
```

Launch Autoware (Runtime Manager)

```
$ cd autoware.ai
$ source install/setup.bash
$ roslaunch runtime_manager runtime_manager.launch
```

Follow the steps to test Rosbag in autoware :

<https://github.com/Autoware-AI/autoware.ai/wiki/ROSBAG-Demo>

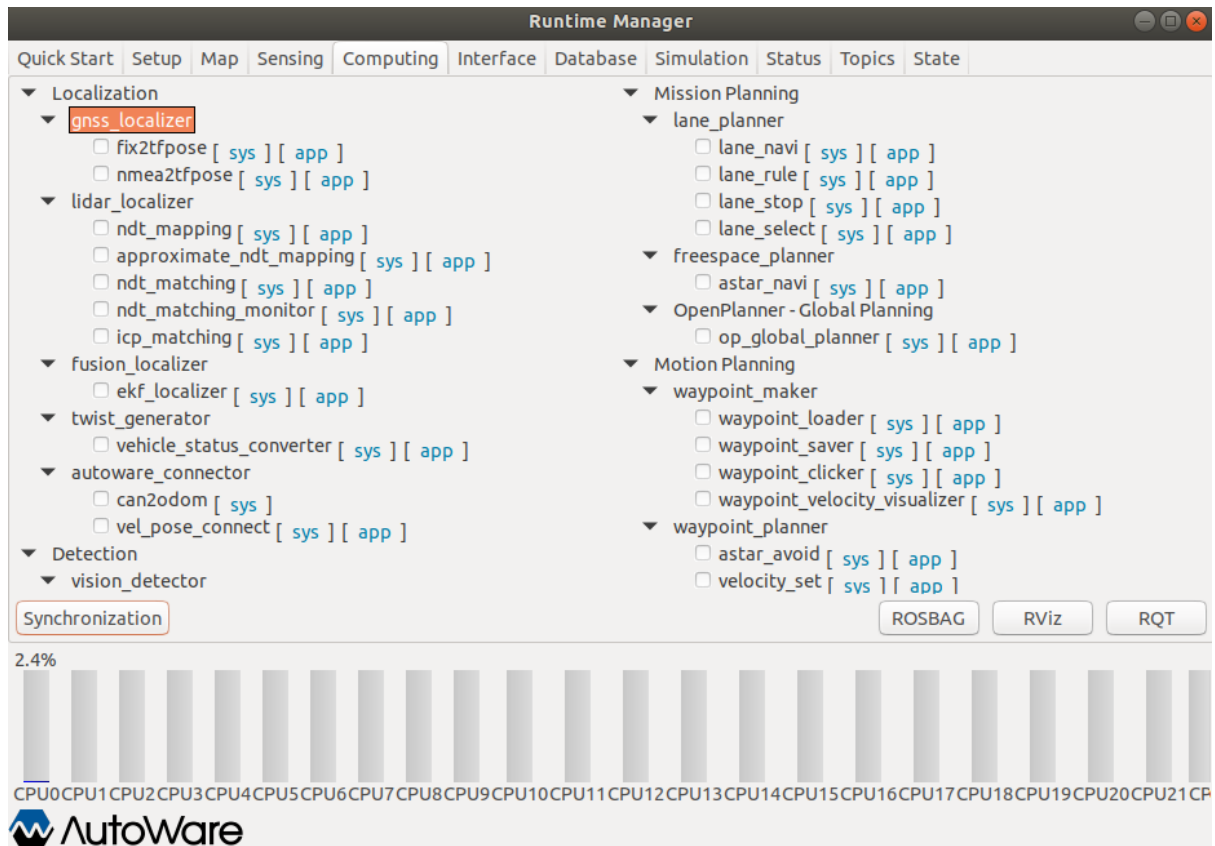
*As the runtime manger has overlapping problem,we use terminal to directly access autoware.

Run Autoware from terminal commands

```
$ roscore
# Open another terminal
$ cd autoware.ai
$ source install/setup.bash
# Run specific packages
$ roslaunch lidar_localizer ndt_matching.launch
# Run GUI rViz
$ rviz
# Run specific packages and see avaiable nodes
$ roslaunch /home/erian/autoware.ai/src/(go to that package)
```

Even through we cannot use Runtime manager to directly interface, we still can make use of it as references.

For example, the package is gnss_localizer and there are two nodes under it: fix2tfpose & nmea2tfpose.



Set bash file

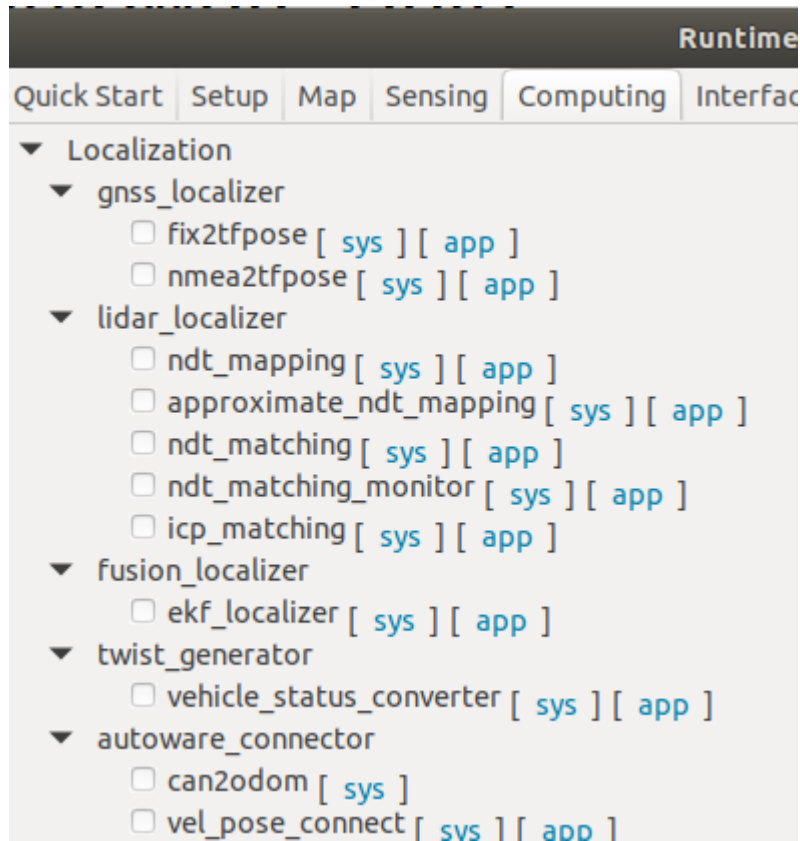
```

#( bash file will automatically run for very new terminal)
$ export EDITOR="gedit"
$ cd home
$ gedit ~/.bashrc

```



Test localization



```
# Run specific packages
$ roslaunch lidar_localizer ndt_matching.launch
# Check perception
$ rqt_graph
```

