

Autoware Manual

Completion Date: 2020.12.03

Editor: 홍정훈



Contents

***Autoware?**

***Installation**

1. Install Dual boot Ubuntu 18.04.5 LTS
2. Install Autoware
 1. ROS Melodic Morenia
 2. eigen 3.3.8
 3. autoware.ai
 4. trouble shooting(eigen3)

***Hardware**

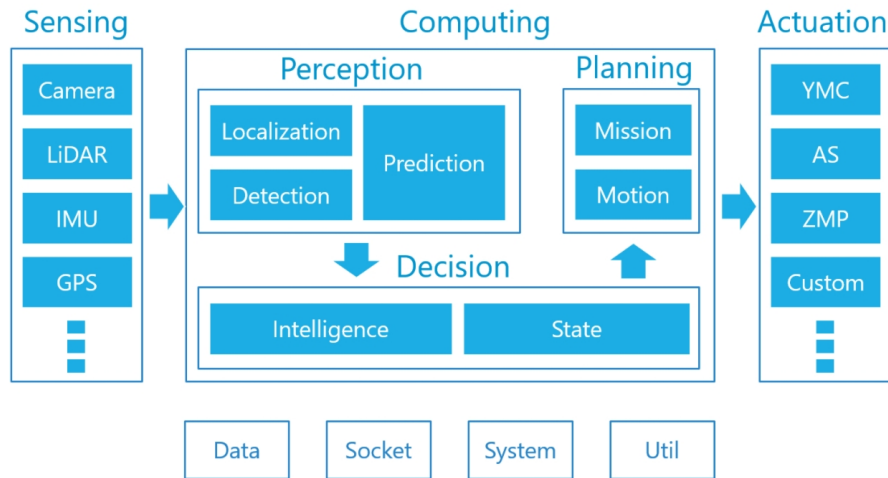
1. CAN
2. VLP 16 FeaturesData packet
 1. ROS Melodic Morenia
 2. eigen 3.3.8

***Program Manual**

1. Runtime Manager
2. Multi lidar Calibration



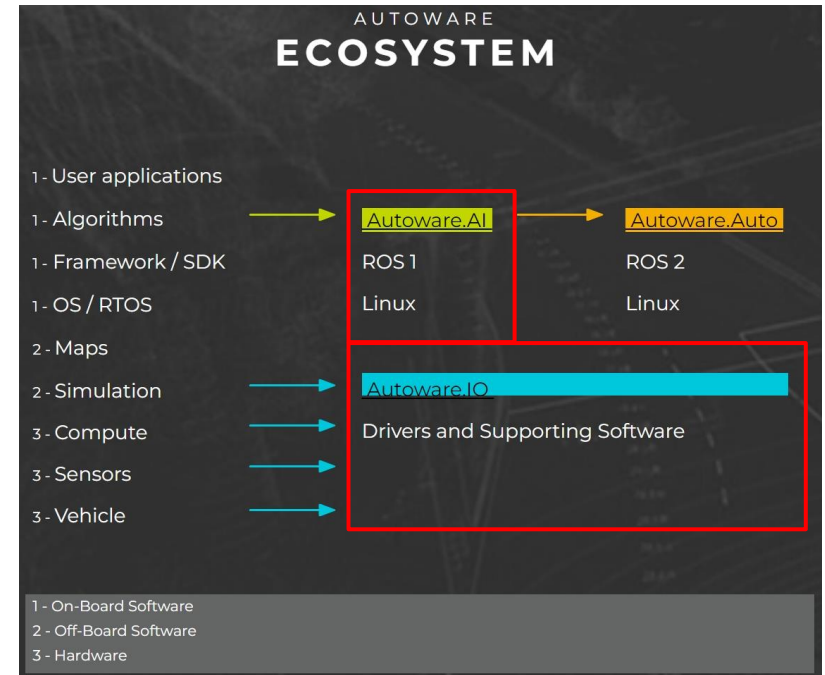
Autoware?



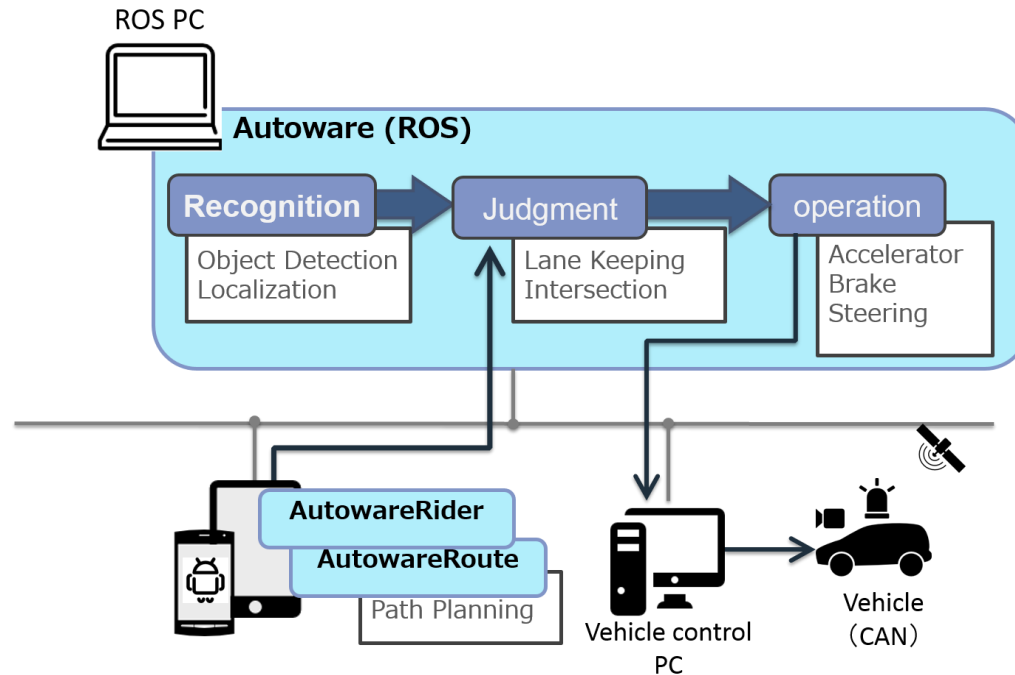
Keywords

- Localization
- Mapping
- Object Detection & Tracking
- Traffic Light Recognition
- Mission & Motion Planning
- Trajectory Generation
- Lane Detection & Selection
- Vehicle Control

- Sensor Fusion
- Cameras, LiDARs
- RADARs
- Deep Learning
- Rule-based System
- Connected Navigation
- Logging
- Virtual Reality



Autoware?



Autoware uses **LIDAR** and **on-vehicle cameras** to localize the ego-car position.

Autoware can detect surrounding objects, such as **pedestrians, vehicles, traffic lights etc.**, by using **LIDAR** and **GNSS**.

Installation

1. Install Dual boot Ubuntu 18.04.5 LTS
2. Install Autoware
 1. ROS Melodic Morenia
 2. eigen 3.3.8
 3. Autoware.ai
 4. Autoware Demo
 5. trouble shooting



1-1. Install Dual boot Ubuntu 18.04.5 LTS

1. Notes before installation

Disk Format : Windows - 디스크 관리 - 디스크 삭제

(BIOS Setting) Secure boot, legacy mode : **Disabled**

Install third-party software for graphics and Wi-Fi hardware and additional media formats : **UNCHECK!**

Something else : **CHECK!**

SWAP partition(RAM's 10%) – Ext4 partition

2.Setting

Setting – Power – Blank screen – Off

Setting – Privacy– Screen Lock – Off

Software Updater – Remind Me Later

3. update & upgrade

```
sudo apt-get update (sudo apt update)
```

```
sudo apt-get upgrade (sudo apt upgrade)
```

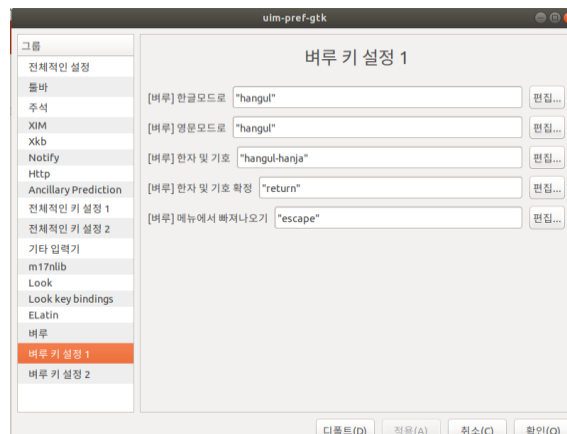
4. Hangul Setting

```
sudo apt install uim
```

Menu - Language Support – Install

툴바 – Display – Never

우측 위 한국어(Hangul CHECK)



1-1. Install Dual boot Ubuntu 18.04.5 LTS

5. Nvidia Driver & CUDA Install

```
sudo apt install gcc sudo rm /etc/apt/sources.list.d/cuda*
```

```
sudo apt update sudo apt remove nvidia-cuda-toolkit
```

```
sudo apt upgrade sudo apt purge nvidia-*
```

```
sudo apt update
```

```
sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
```

```
sudo bash -c 'echo "deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 /" > /etc/apt/sources.list.d/cuda.list'
```

```
sudo apt update
```

```
sudo apt install cuda-10-0
```

```
sudo gedit ~/.bashrc
```

```
export PATH=/usr/local/cuda-10.0/bin${PATH:+:${PATH}}  
Add Bottom line in gedit tool
```

```
sudo reboot
```

```
nvidia-smi
```

```
nvcc --version
```

linux default graphics driver : nouveau

When CUDA is installed,
NVIDIA Drive and CUDA Toolkit are automatically installed.

Ignore CUDA Version 11.1 in `nvidia-smi`

Cuda 10.0 in `nvcc --version`

```
cl@scl-TFG277S:~$ nvidia-smi
Tue Dec  1 01:37:40 2020

+-----+
| NVIDIA-SMI 455.45.01    | Driver Version: 455.45.01    | CUDA Version: 11.1    |
+-----+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                      | MIG M. |
+-----+-----+
|  0  GeForce RTX 2070      On          | 00000000:01:00:0 Off |         N/A         |
| N/A   32C    P8         5W /  N/A   |  268MiB /  7982MiB |      4%      Default |
+-----+-----+

scl@scl-TFG277S:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Sat Aug 25 21:08:01 CDT 2018
Cuda compilation tools, release 10.0, V10.0.130

+-----+
|  0  N/A   N/A         1344      G   /usr/bin/gnome-shell | 28MiB |
|  0  N/A   N/A         1574      G   /usr/lib/xorg/Xorg     | 116MiB|
|  0  N/A   N/A         1760      G   /usr/bin/gnome-shell   |  69MiB|
+-----+
U Memory age
```



1-1. Install Dual boot Ubuntu 18.04.5 LTS

5. Nvidia Driver & CUDA Install

```
scl@scl-TFG277S:~$ sudo apt-get -y install cuda
```

패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
몇몇 패키지를 설치할 수 없습니다. 요청한 상황이 불가능할 수도 있고,
불안정 배포판을 사용해서 일부 필요한 패키지를 아직 만들지 않았거나,
아직 Incoming에서 나오지 않은 경우일 수도 있습니다.
이 상황을 해결하는데 다음 정보가 도움이 될 수도 있습니다:

다음 패키지의 의존성이 맞지 않습니다:
cuda : 의존: cuda-11-1 (>= 11.1.1) 하지만 %s 패키지를 설치하지 않을 것입니다
E: 문제를 바로잡을 수 없습니다. 망가진 고정 패키지가 있습니다.

1-2. Install Dual boot Ubuntu 18.04.5 LTS

6. Install Chrome

```
wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
```

```
sudo sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable main" >>  
/etc/apt/sources.list.d/google.list'
```

```
sudo apt-get update
```

```
sudo apt-get install google-chrome-stable
```

```
sudo rm -rf /etc/apt/sources.list.d/google.list
```

```
sudo apt-get clean
```

7. Ubuntu Time Zone

```
timedatectl set-local-rtc 1
```

```
sudo gedit /etc/default/rcS
```

```
UTC=no in gedit tool
```

8. Boot's Loader (Default : Windows Boot Manager)

```
sudo gedit /etc/default/grub
```

```
GRUB_DEFAULT=saved in gedit tool
```

```
sudo update-grub
```

```
sudo grub-set-default 2
```

```
grub-editenv list
```

```
saved_entry=2
```

2. Install Autoware.ai

Autoware Version	Ubuntu 14.04	Ubuntu 16.04	Ubuntu 18.04
v1.14.0			x
v1.13.0			X
v1.12.0		X	X
v1.11.1		X	
v1.11.0		X	
v1.10.0		X	
v1.9.1	X	X	
v1.9.0	X	X	

Product	Ubuntu 14.04	Ubuntu 16.04	Ubuntu 18.04
ROS	Indigo	Kinetic	Melodic
Qt	4.8.6 or higher	5.2.1 or higher	5.9.5 or higher
CUDA (optional)	8.0GA(?)	9.0	10.0
FlyCapture2 (optional)			
Armadillo (optional)			

Install Process

ROS Melodic -> Eigen 3.3.7 -> autoware.ai 1.12.0

Autoware.AI Wiki GitLab : <https://gitlab.com/autowarefoundation/autoware.ai/autoware/-/wikis/Source-Build>

If you install Autoware source build, error will occur.

Error : CUDA, eigen, 의존성, 호환성

Docker 설치 방법은 추후 수정 예정.

2-1. Install Autoware – ROS Melodic Morenia

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
sudo apt-get update
```

```
sudo apt install ros-melodic-desktop-full
```

```
sudo apt-get install python-pip
```

```
sudo -H pip install -U rosdep
```

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

```
source ~/.bashrc
```

```
sudo apt-get install python-roscpp  
sudo apt-get install python-rosinstall-generator  
sudo apt-get install python-wstool build-essential
```

```
sudo apt install python-rosdep
```

```
sudo rosdep init
```

```
rosdep update
```

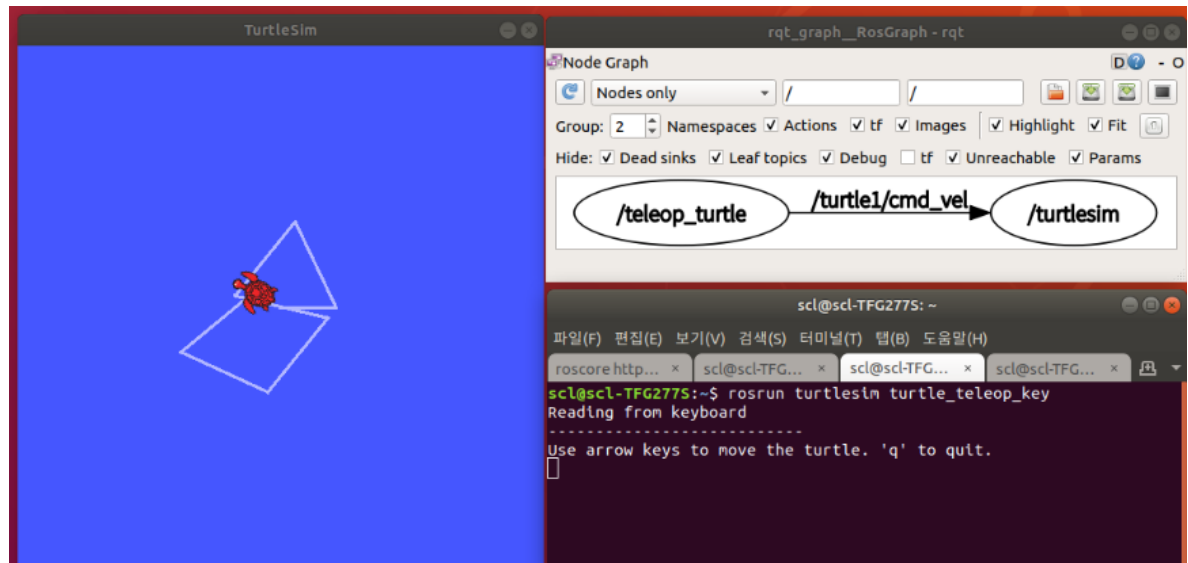
ROS test code

```
roscore
```

```
roslaunch turtlesim turtlesim_node
```

```
roslaunch turtlesim turtle_teleop_key
```

```
rqt_graph
```



2-2. Install eigen 3.3.7

```
cd && wget https://gitlab.com/libeigen/eigen/-/archive/3.3.7/eigen-3.3.7.tar.bz2
```

```
mkdir eigen && tar --strip-components=1 -xzf eigen-3.3.7.tar.bz2 -C eigen
```

```
cd eigen && mkdir build && cd build && cmake .. && make && make install
```

```
cd && rm -rf eigen-3.3.7.tar.bz2 && rm -rf eigen
```

[illegible][illegible]

2-3. Install Autoware.ai

```
mkdir -p autoware.ai/src
```

```
cd autoware.ai
```

```
wget -O autoware.ai.repos  
https://gitlab.com/autowarefoundation/autoware.ai/autoware/raw/1.14.0/autoware.ai.repos?inline=false
```

```
sudo apt-get install -y python3-vcstools
```

```
vcs import src < autoware.ai.repos
```

```
rosdep update
```

```
rosdep install -y --from-paths src --ignore-src --  
rosdistro $ROS_DISTRO
```

```
sudo sh -c 'echo "deb  
http://packages.ros.org/ros/ubuntu  
'lsb_release -  
cs' main" > /etc/apt/sources.list.d/ros-  
latest.list'
```

```
sudo apt-key adv --  
keyserver 'hkp://keyserver.ubuntu.com:80' --  
recv-  
key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
sudo apt update
```

```
sudo apt install python3-colcon-common-  
extensions
```

```
AUTOWARE_COMPILE_WITH_CUDA=1 colcon build --  
cmake-args -DCMAKE_BUILD_TYPE=Release
```

2-4. Autoware Demo

```
wget https://autoware-ai.s3.us-east-2.amazonaws.com/sample_moriyama_data.tar.gz
wget https://autoware-ai.s3.us-east-2.amazonaws.com/sample_moriyama_150324.tar.gz
cd ~
mkdir .autoware
cd .autoware
cp ~/Downloads/sample_moriyama_* .
tar zxfv sample_moriyama_150324.tar.gz
tar zxfv sample_moriyama_data.tar.gz
cd autoware.ai
source install/setup.bash
roslaunch runtime_manager runtime_manager.launch
```

2-4. Autoware Demo

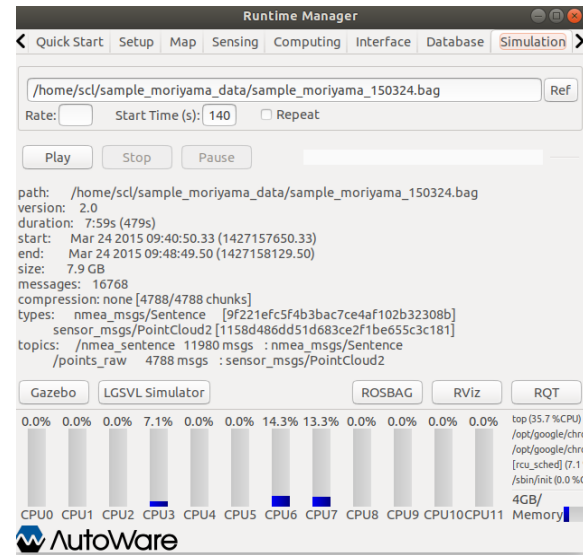


Quick Start Tap

`autoware.ai/src/autoware/documentation/autoware_quickstart_examples/launch/rosbag_demo/ *`

Simulation Tap

Set the start time to 140, click Play and Pause just after it has started playing



Launch RViz through the RViz button in the bottom-right corner of the runtime manager

Ctrl + O

`autoware.ai/src/autoware/documentation/autoware_quickstart_examples/launch/rosbag_demo/default.rviz`

2-5. Trouble shooting

1. CUDA downgrade(latest ver is CUDA 11.1)

GPU support on ROS Melodic requires $\text{CUDA} \leq 10.0$,
CUDA 9.0 is not supported(GCC error)

2. runtime manager WARNING

bash: ../../../../setup.bash: 그런 파일이나 디렉터리가 없습니다

mkdir: `/home/scl/.rviz' 디렉토리를 만들 수 없습니다: 파일이 있습니다

3. Runtime manager GUI WARNING

(runtime_manager_dialog.py:12993): Gtk-WARNING **: 16:38:32.618: Negative content width -15 (allocation 1, extents 8x8) while allocating gadget (node entry, owner GtkEntry)

(runtime_manager_dialog.py:12993): Gtk-WARNING **: 16:38:32.632: for_size smaller than min-size (0 < 3) while measuring gadget (node trough, owner GtkScale)

(runtime_manager_dialog.py:12993): Gtk-CRITICAL **: 16:38:32.632: gtk_box_gadget_distribute: assertion 'size >= 0' failed in GtkScale

4. Simulation ERROR

Failed connect to /tmp/autoware_proc_manager

2-5. Trouble shooting

5. dpkg frontend lock

E /var/lib/dpkg/lock-frontent 잠금파일을 얻을 수 없습니다. -open

E Unable to acquire the dpkg frontend lock

```
sudo rm /var/lib/apt/lists/lock
```

```
sudo rm /var/cache/apt/archives/lock
```

```
sudo rm /var/lib/dpkg/lock*
```

```
sudo dpkg --configure -a
```

```
sudo apt update
```

```
sudo reboot
```

6. Docker Graphics Driver

- > 도커를 설치하면 엔비디아 드라이버, 쿠다를 연결해줄 수 없다는 오류가 나타남
- > 엔비디아, 쿠다가 없으면 colcon build가 안됨.
- > 도커는 나중에 다시 정리

VLP-16

1. Features
2. Data packets



1. CAN

샤시(Chassis)				
엔진	동력전달장치	조향장치	제동장치	현가장치
엔진 본체	클러치	조향 핸들	상용 브레이크	새시 스프링
냉각 장치	변속기	조향 기어	주차 브레이크	쇼크 업서버
윤활 장치	추진축	링크 기구	보조 브레이크	스테빌라이서
과급 장치	차동기	앞 바퀴		
흡배기 장치	차축			
연료 장치	바퀴			

Chassis CAN(샤시캔) :

차량의 클러스터(CLU),
YRS(Yaw Rate Sensor)엔진,
미션, ABS(Anti-lock Breaking System),
ECU(Engine Control Unit),
TCU(Transmission Control Unit)와
같은 고속으로 데이터를 전송하는 용도로 사용되며 통신 속도는 500kbps

1. CAN



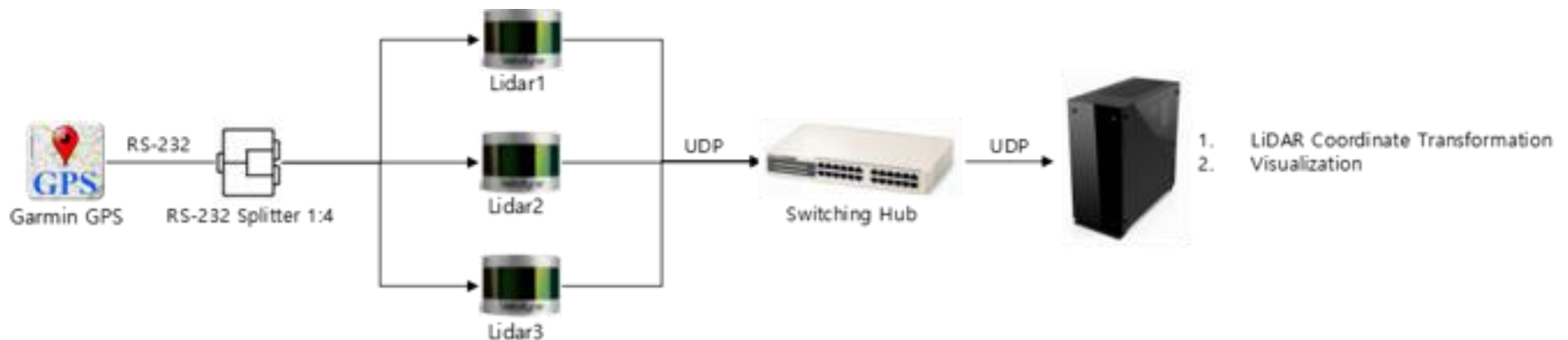
The main features of the VN1630A interface are:

- ▶ 2x CAN high-speed 1051cap transceiver (capacitively decoupled)
- ▶ 2x additional plug-in location for CAN-/LINpiggies
- ▶ Fifth channel for dedicated digital-analog input/output tasks
- ▶ Five LEDs indicating bus activities and status
- ▶ Software sync
- ▶ Hardware sync (via SYNCcableXL)

2-1. Features of VLP-16

Channel	16 channels
Horizontal Field of View(FOV)	360°
Rotational speed	5-20 rotations per second
Vertical Field of View(FOV)	30°
Operating range	up to 100 meters
Angular resolution(vertical)	2°
Angular resolution(horizontal)	0.1° - 0.4°

Velodyne LiDAR Puck



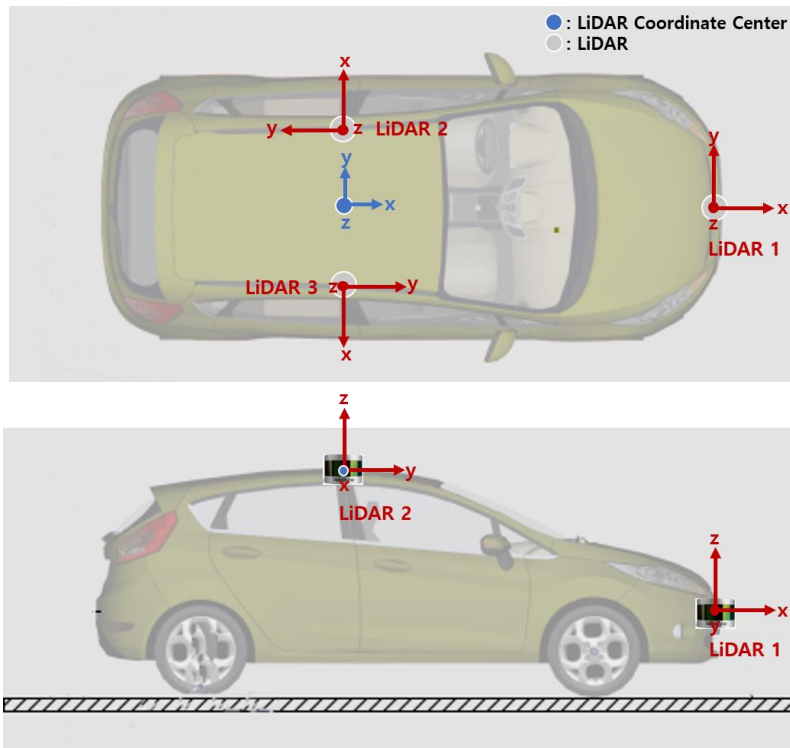
2-1. Features of VLP-16

IP address

Lidar1(Front) : 192.168.1.109

Lidar2(Left) : 192.168.1.110

Lidar3(Right) : 192.168.1.201



Top view Lidar offset from LiDAR coordinate center

	Lidar1 (Front)	Lidar2 (Left)	Lidar3 (Right)
X(mm)	0	-500	500
Y(mm)	2500	0	0
Z(mm)	820	1.77	1.77
Roll(°)	0	0	0
Pitch(°)	0	13	13
Yaw(°)	0	-92	92

2-2. Packet

front.dat

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
00000000	00	72	00	00	00	C6	49	24	C0	02	41	7B	40	56	F2	07	3D	03	B1	6D	CD	49
00000016	01	D5	DF	BF	C1	43	27	7D	41	29	42	A6	3F	3F	B1	6D	CD	49	02	97	F2	3C
0000002C	C0	BB	79	84	40	35	26	25	3D	02	B1	6D	CD	49	03	98	C0	BF	C1	8F	04	7D
00000042	41	A4	71	10	40	4C	B1	6D	CD	49	04	D8	CA	5B	C0	B9	0A	8D	46	33	F1	86
00000058	3D	03	B1	6D	CD	49	05	5A	B0	BF	C1	84	F2	7C	41	C3	F3	4D	40	3C	B1	6D

Point Array Size: 29184

Laser ID: 0

Position X: 3.22m

Position Y: 1.08m

Position Z: 1.02m

Time Stamp: 1238199729µs
(1238199729µs past the hour based on the current UTC provided by GPS)

Intensity: 3

Position Calculation:

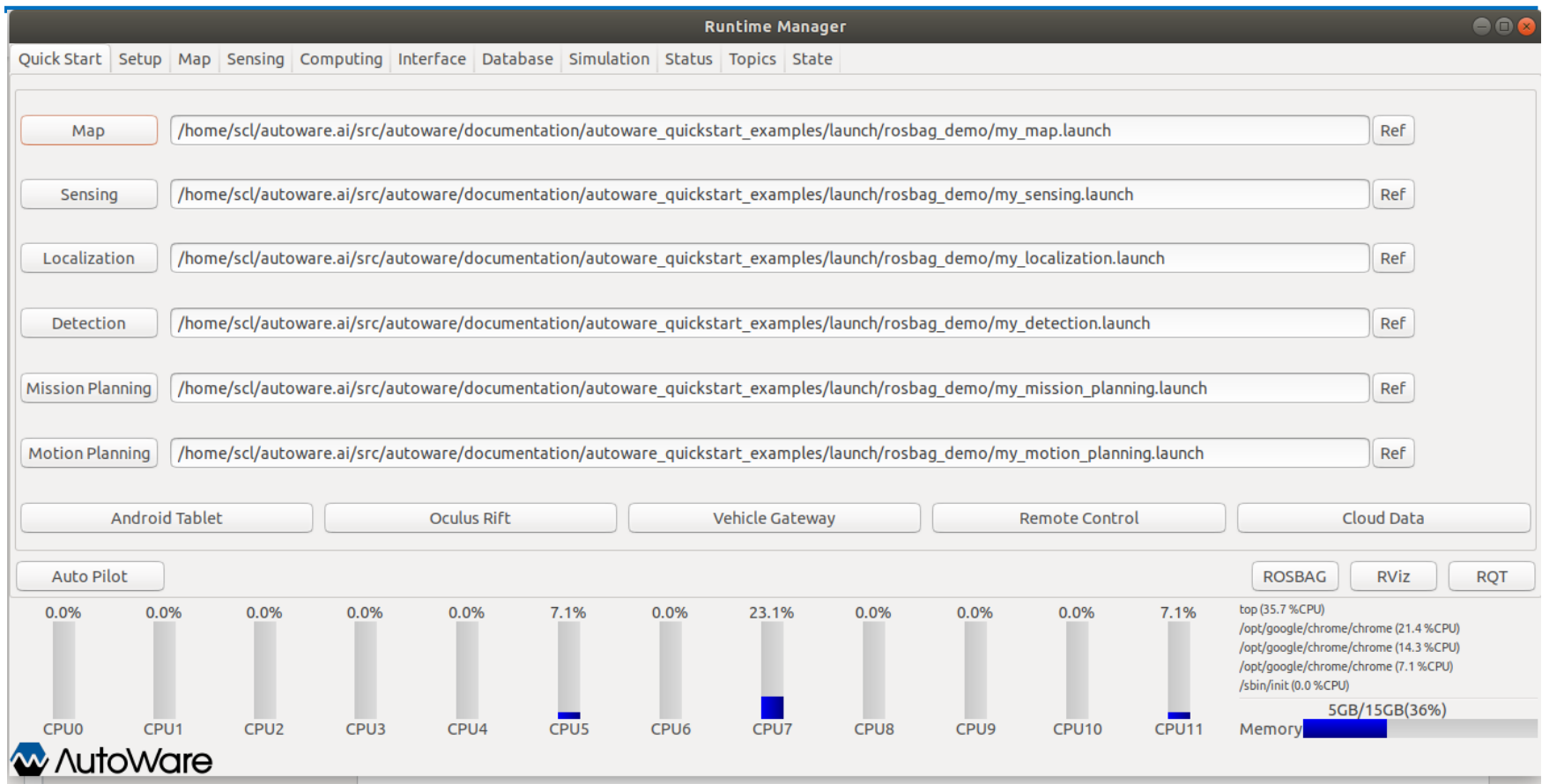
- 1) Get Position Values: C6 49 24 C0
- 2) Reverse the bytes: C0 24 49 C6
- 3) Combine the bytes: 0xC02449C6
- 4) Convert to float number: +3.22m

Program Manual

1. Runtime Manager
2. Multi Lidar Calibrator



1. Runtime Manager

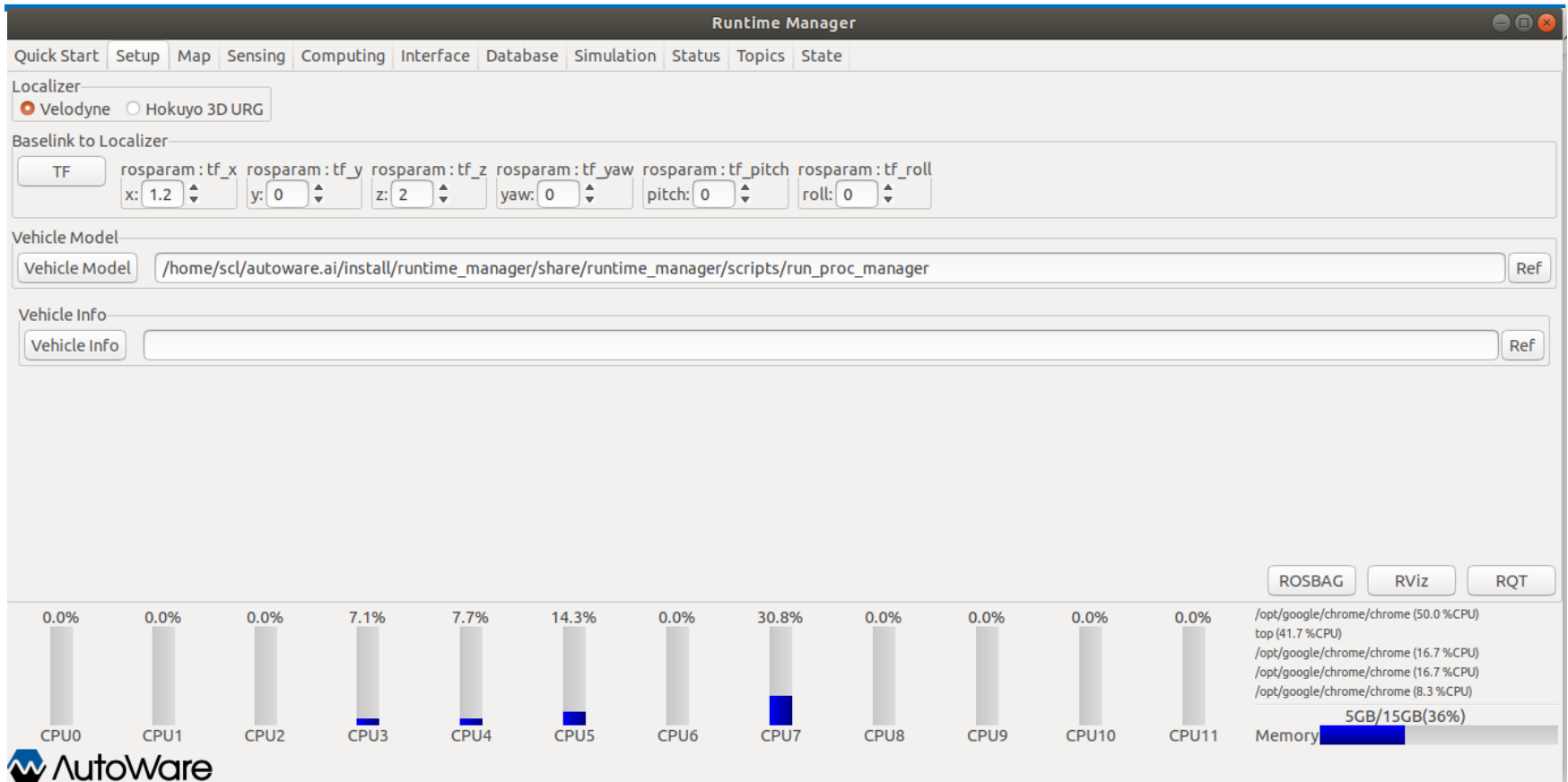


Map : TF, Point Cloud, Vector Map

Sensing : calibration file path, HDL-32e

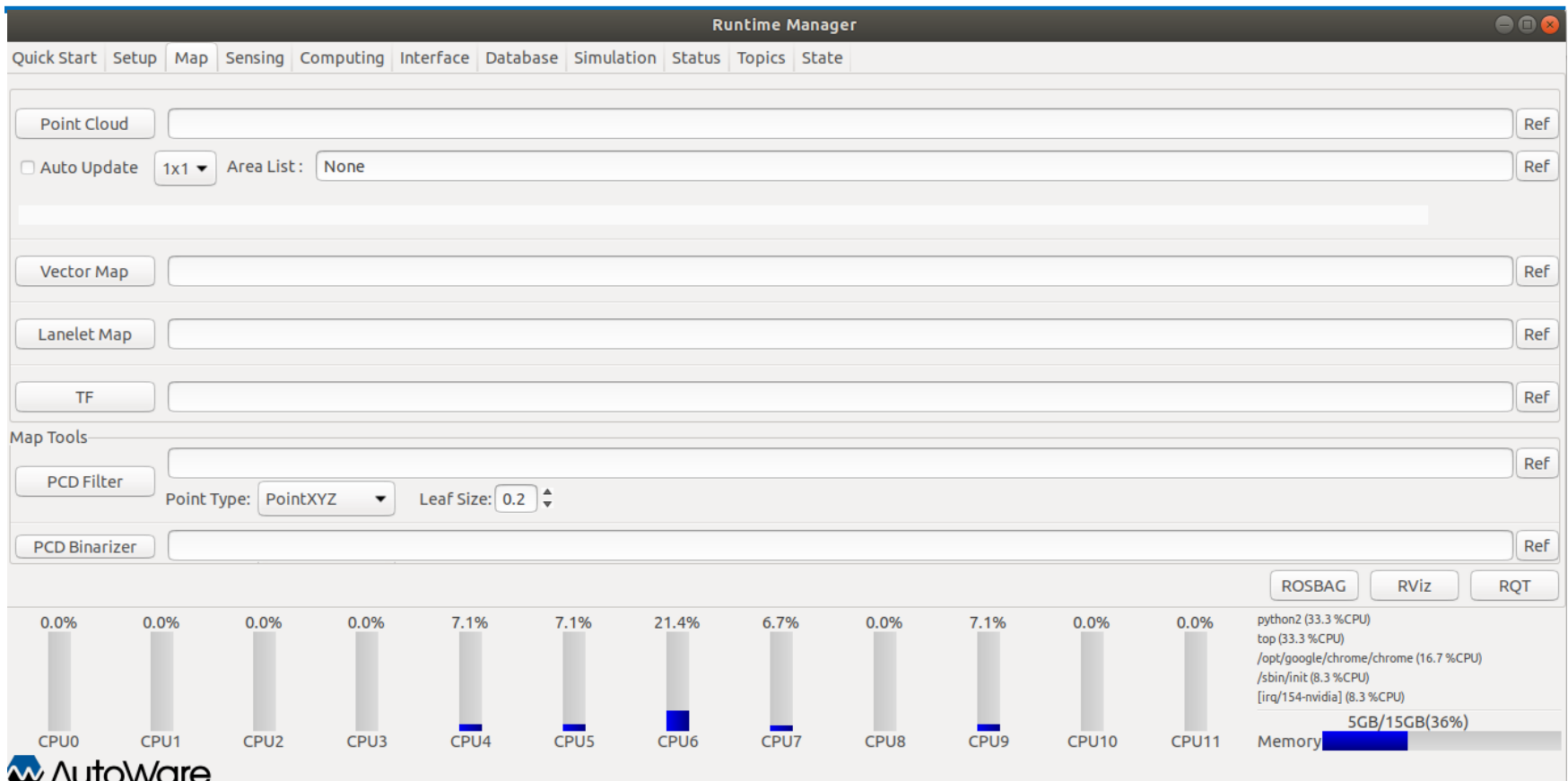
Localization : setting path parameter, Setup, points downsampler, nmea2tfpose, ndt_matching

1. Runtime Manager



TF : base_link → Velodyne position(velodyne)
 x, y, z, yaw, pitch, roll : Enter the relative position of velodyne to base_link
 Vehicle Model : *Autoware/ros/src/.config/model/default.urdf*

1. Runtime Manager



TF : ~/.autoware/data/tf/tf.launch

나머지는 정보가 없음 추후 업데이트 예정

1. Runtime Manager

Quick Start Setup Map Sensing Computing Interface Database Simulation Status Topics State

Drivers

CAN

- ☐ can_converter
- ☐ can_draw
- ☐ can_listener [config]

Cameras

- ☐ PointGrey Grasshopper 3 (USB1) [config]
- ☐ PointGrey Generic
- ☐ FLIR ADK [config]
- ☐ PointGrey LadyBug 5 [config]
- ☐ PointGrey Spinnaker [config]
- ☐ USB Generic
- ☐ IEEE1394
- ☐ Baumer VLG-22
- ☐ IDS UI-3060CP
- ☐ Sekonix 3322/3323 GMSLCamera
- ☐ AVT Vimba Mako [config]

GNSS

- ☐ Javad Delta 3 (TTY1) [config]
- ☐ Garmin GPS 18x LVC
- ☐ Serial GNSS [config]

IMU

- ☐ Memsic VG440 [config]
- ☐ Xsens MTi-300 [config]
- ☐ MicroStrain 3DM-GX5-15 [config]
- ☐ Analog Devices ADIS16470 [config]

LIDARs

- ☐ Velodyne HDL-64E-S2 [config]
- ☐ Velodyne HDL-64E-S3 [config]
- ☐ Velodyne HDL-32E [config]
- ☐ Velodyne VLP-32C [config]
- ☐ Velodyne VLP-16 [config]
- ☐ Velodyne VLP-16 Hi-Res [config]
- ☐ Hokuyo 2D URG [config]
- ☐ Hokuyo 3D-URG [config]
- ☐ Sick LMS511 [config]
- ☐ Sick LD-MRS (IBEO Lux) [config]
- ☐ Robosense RS32
- ☐ Robosense RS16
- ☐ Ouster OS1 [config]

Points Downsample

- ☐ voxel_grid_filter [sys] [app]
- ☐ ring_filter [sys] [app]
- ☐ distance_filter [sys] [app]
- ☐ random_filter [sys] [app]

Points Preprocessor

- ☐ ring_ground_filter [sys] [app]
- ☐ ray_ground_filter [sys] [app]
- ☐ points_concat_filter [sys] [app]
- ☐ cloud_transformer [sys] [app]
- ☐ compare_map_filter [sys] [app]

Fusion

- ☐ multi_lidar_calibrator [sys] [app]

Calibration Publisher

Points Image

Virtual Scan Image

ROS BAG RViz RQT

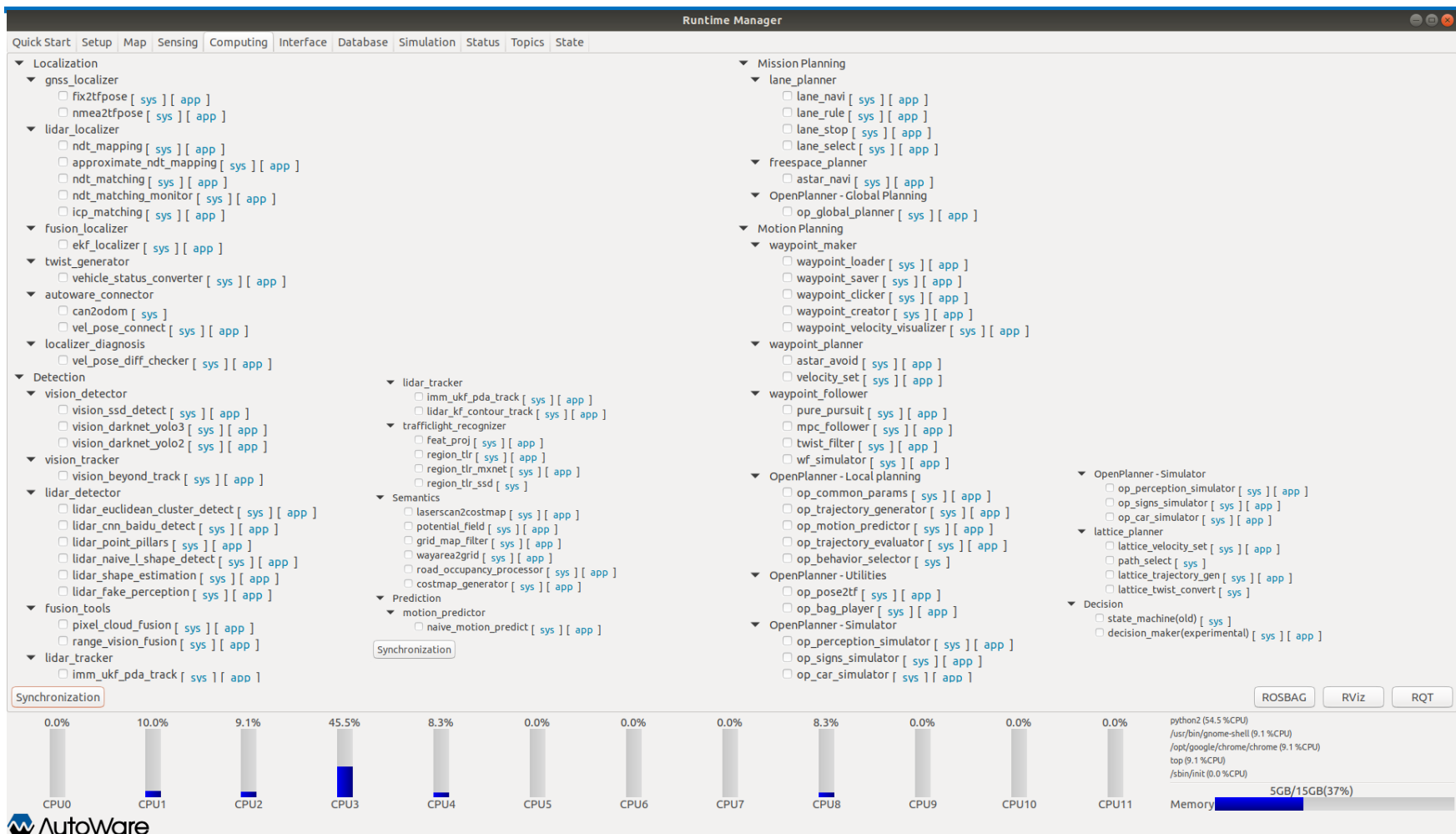
python2 (40.0 %CPU)
top (26.7 %CPU)
/opt/google/chrome/chrome (20.0 %CPU)
/opt/google/chrome/chrome (13.3 %CPU)
/usr/lib/xorg/Xorg (6.7 %CPU)

5GB/15GB(37%)

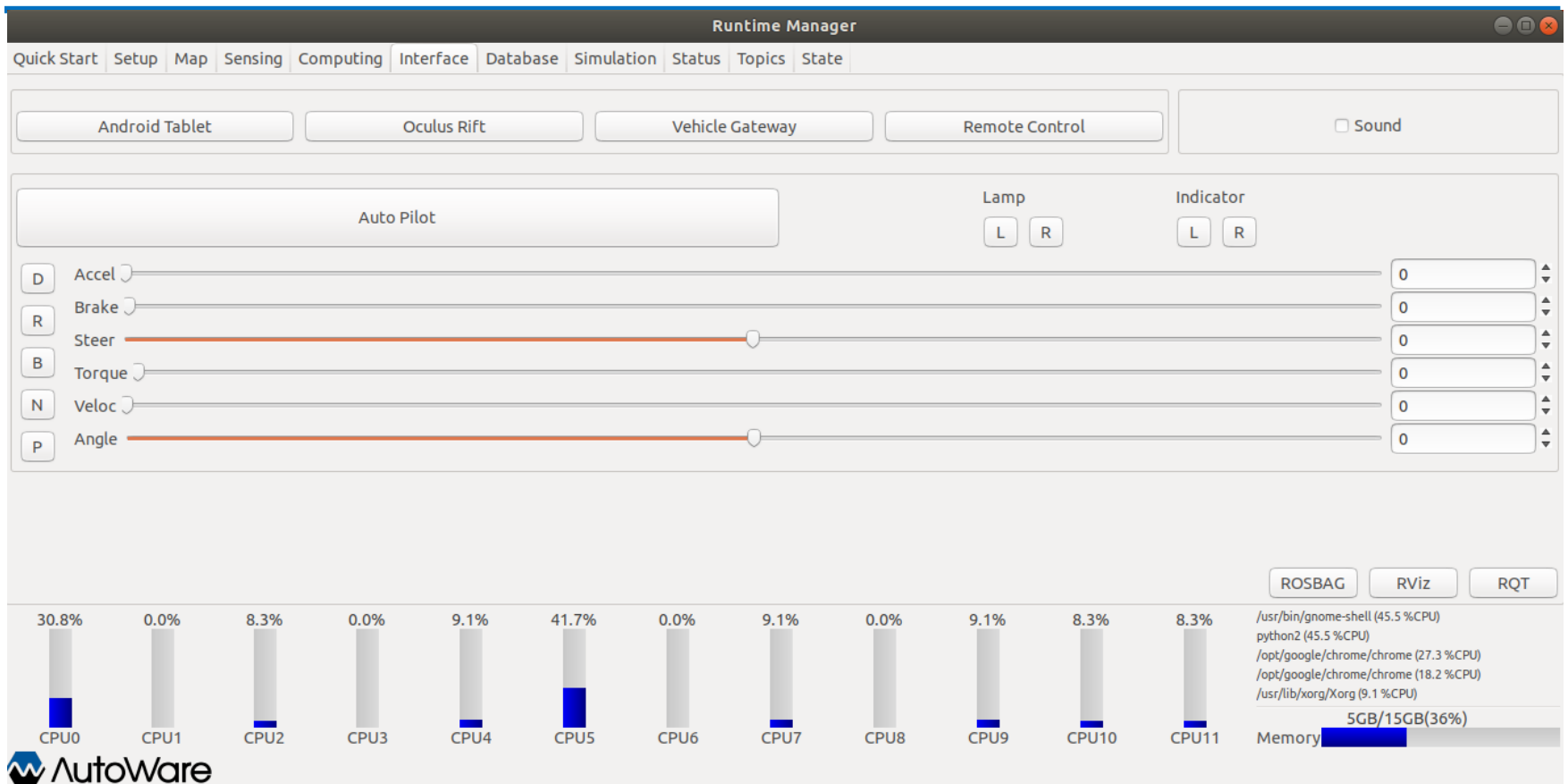
AutoWare

CPU0 5.9% CPU1 6.2% CPU2 0.0% CPU3 7.1% CPU4 0.0% CPU5 0.0% CPU6 20.0% CPU7 0.0% CPU8 35.7% CPU9 0.0% CPU10 7.1% CPU11 7.1%

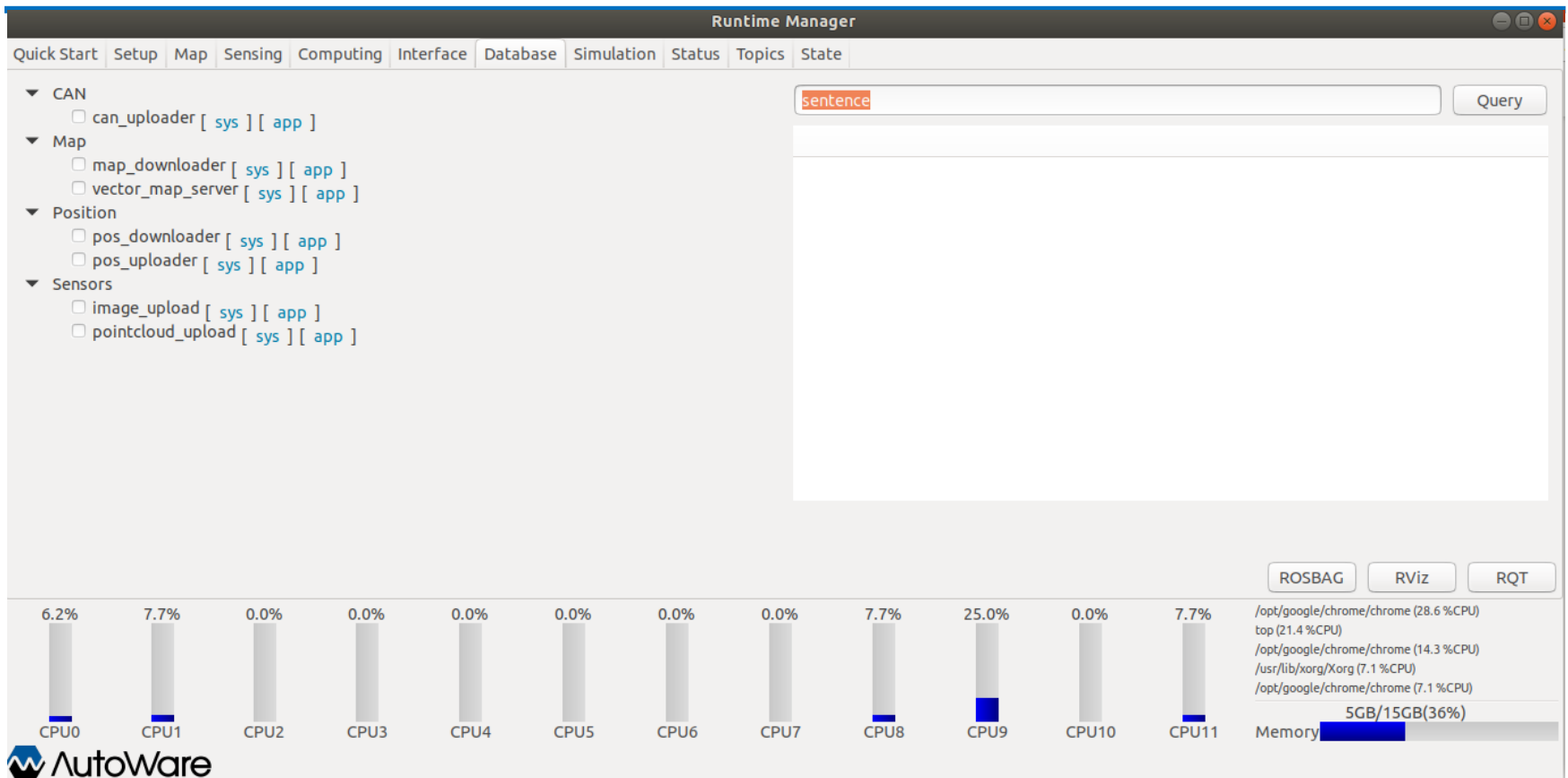
1. Runtime Manager



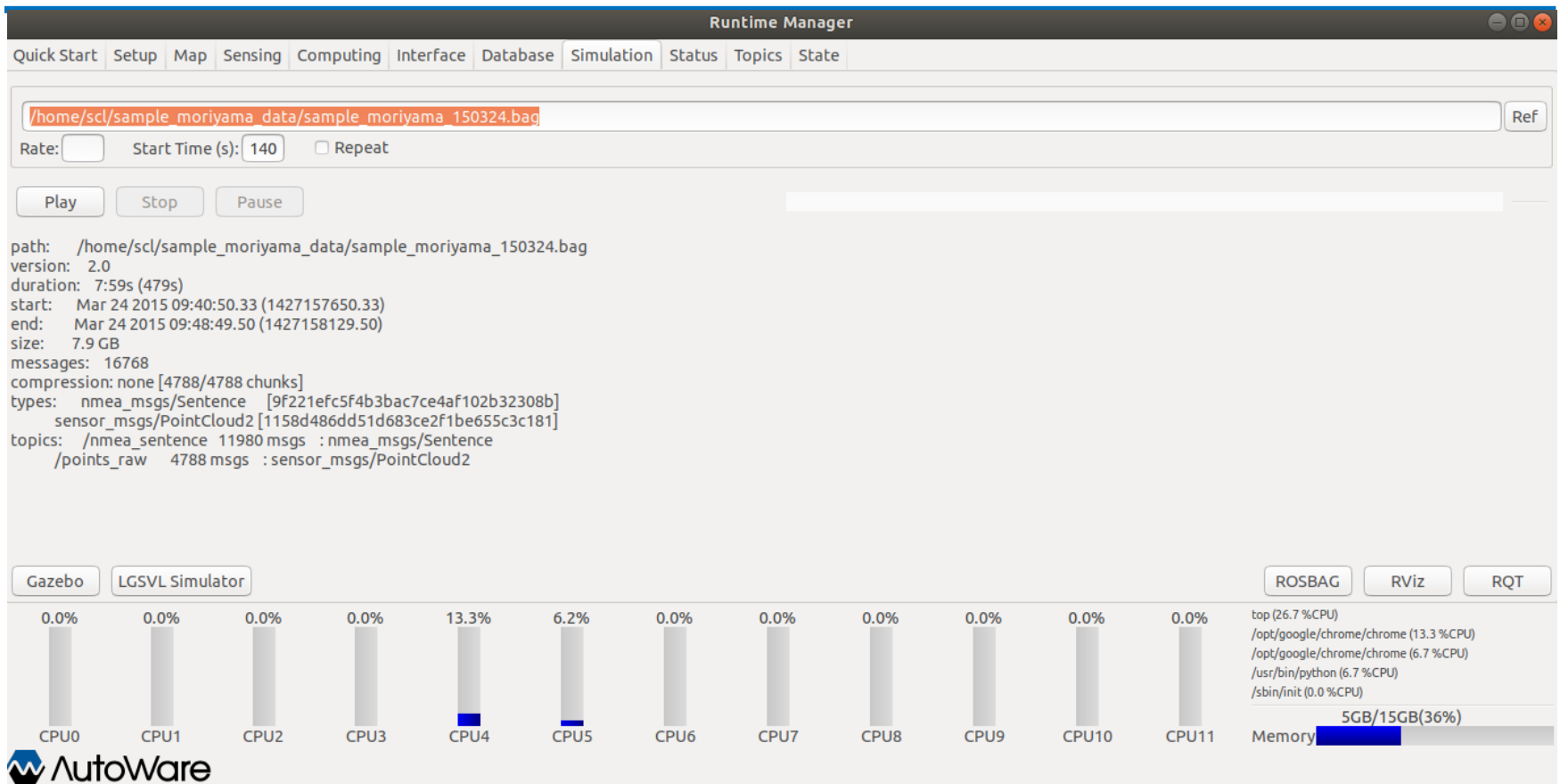
1. Runtime Manager



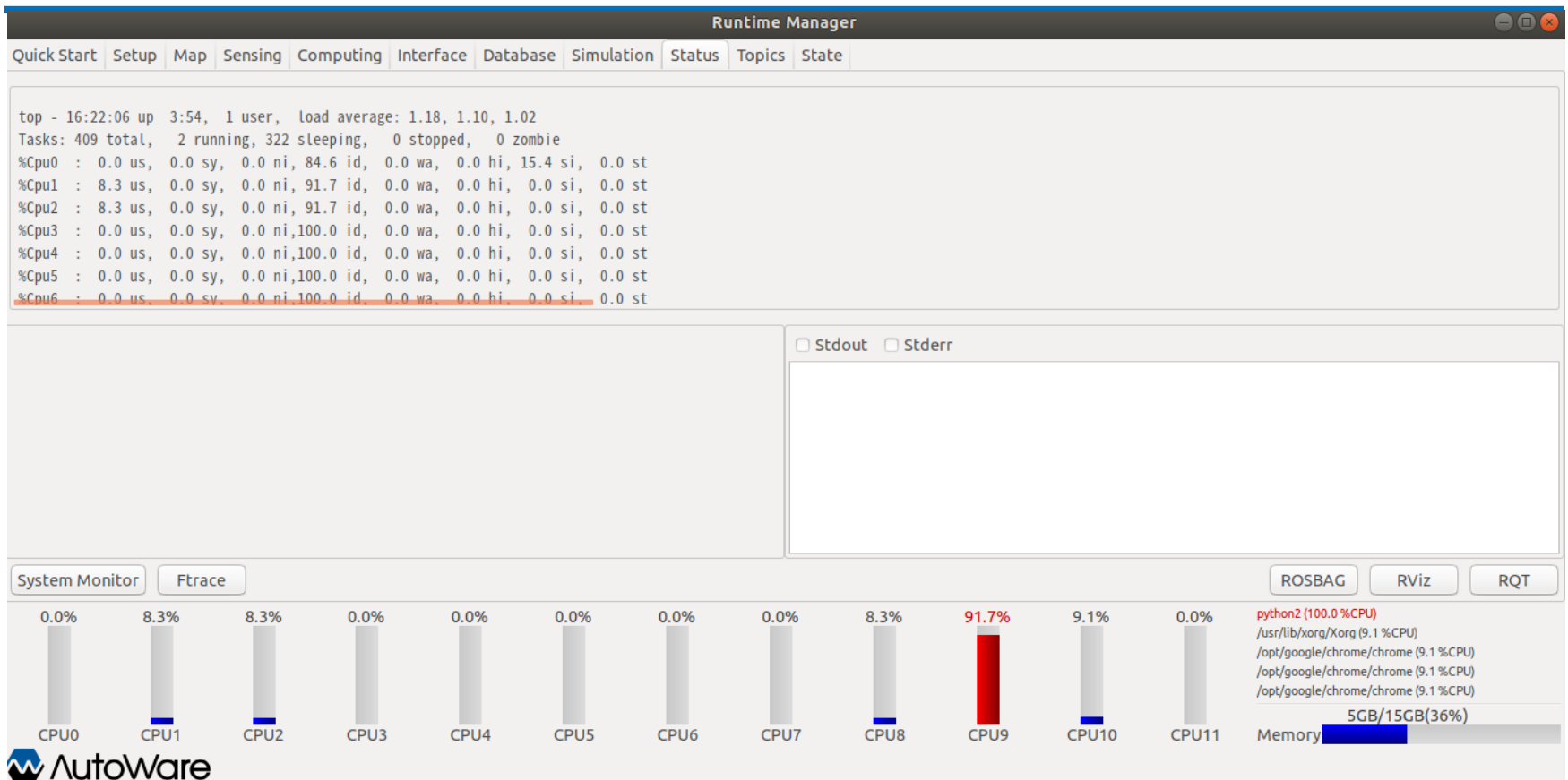
1. Runtime Manager



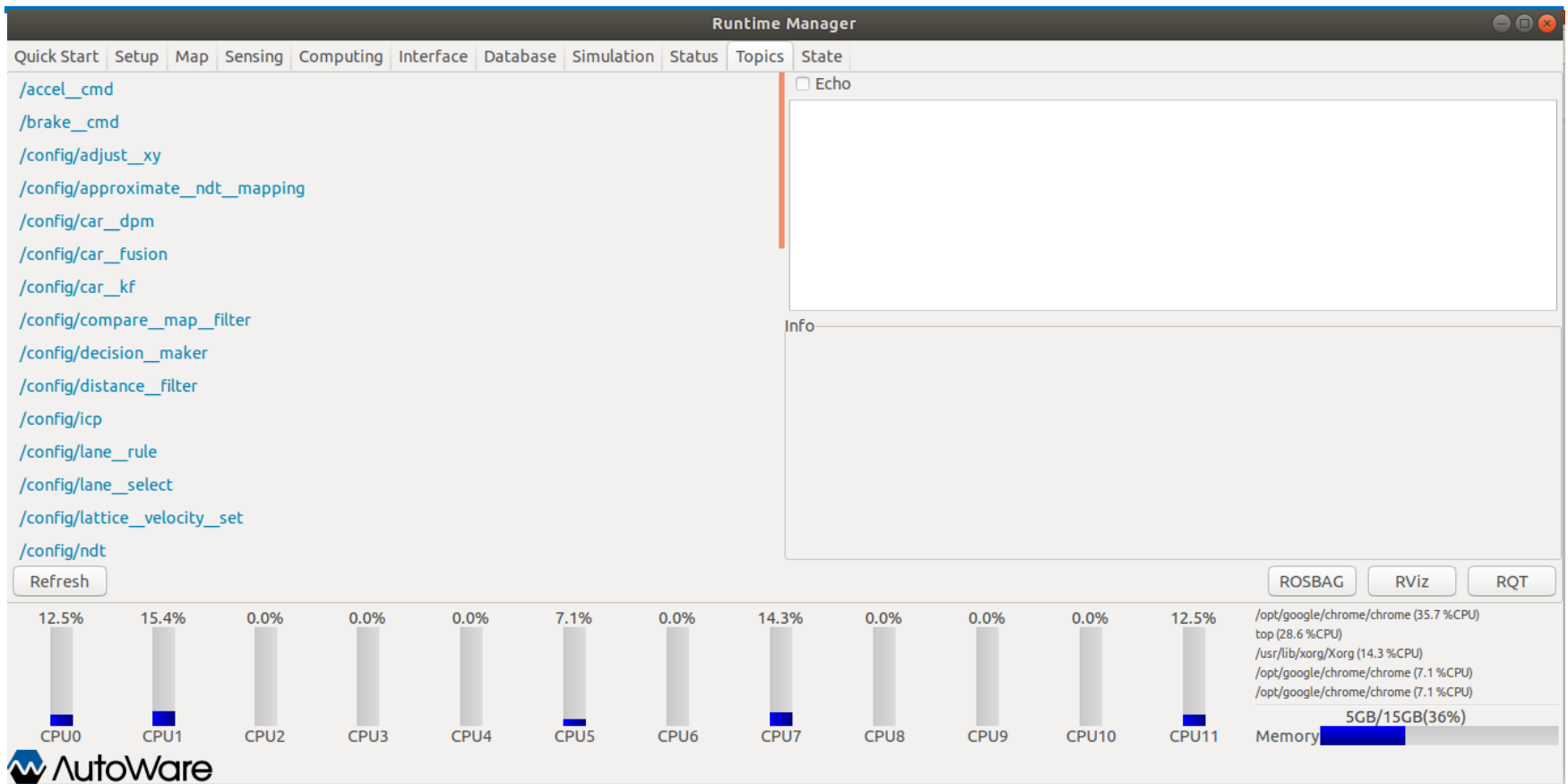
1. Runtime Manager



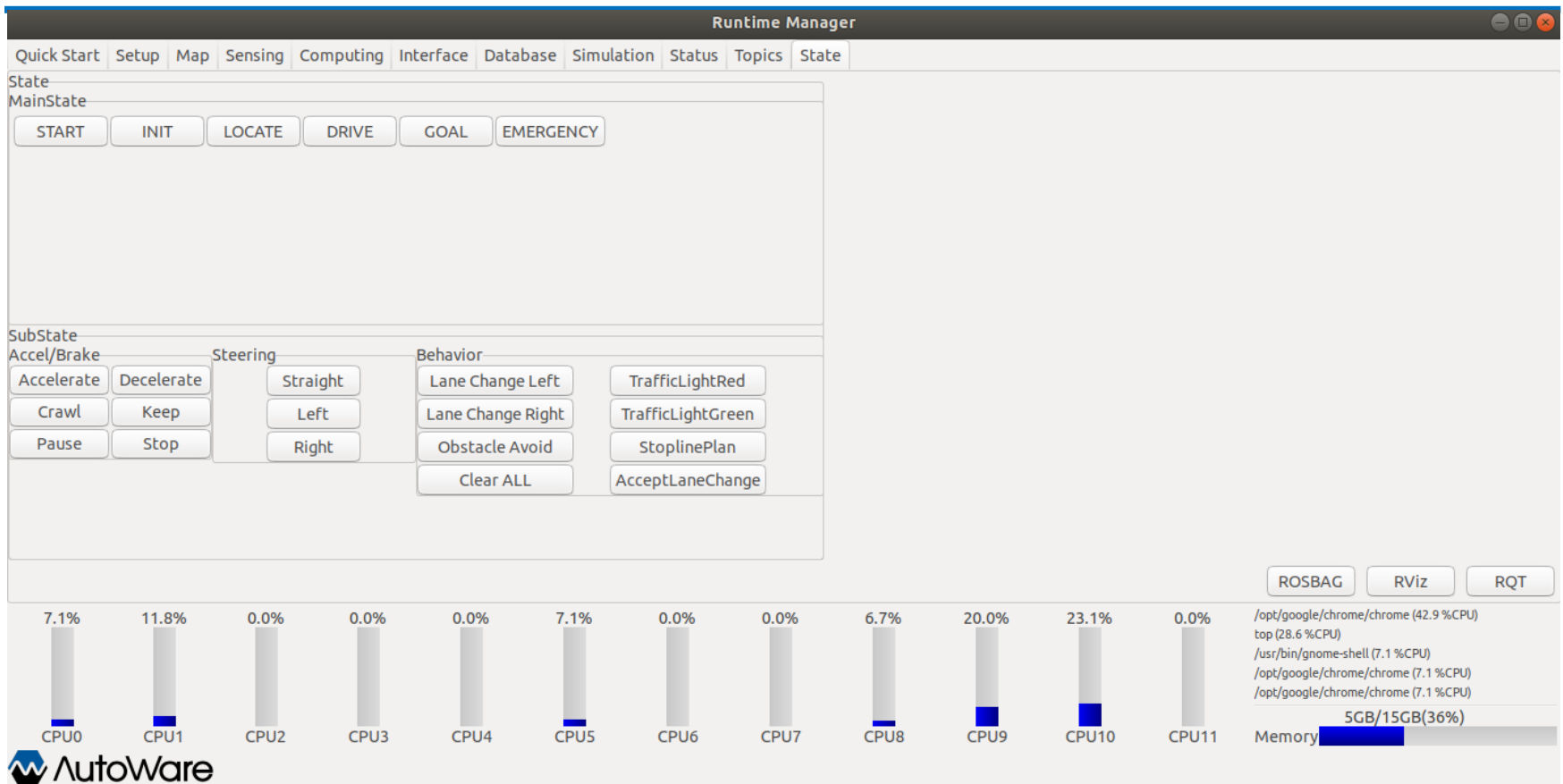
1. Runtime Manager



1. Runtime Manager



1. Runtime Manager



2. NDT(Normal Distribution Transform)

1. NDT mapping (Map generation)

Transform the LiDAR point cloud into a piecewise continuous and differentiable probability density (NDT).

The probability density contains a set of the normal distributions where each point in point cloud is assigned to a voxel.

A voxel is a 3D lattice cube to which points are assigned depending upon their coordinate value.

The Point cloud is divided into k ND voxels clouds and are combined together , and also the voxel grid filter is used to decrease the computation cost and to reduce the noise from the 3D map

2. NDT matching (Localization)

A search problem where we have to find a transform that maximizes NDT sum to match the different point clouds, a variety of minimization functions can be used for this. Newton nonlinear optimizer is used to find the best 6-DOF pose.

3. Multi Lidar Calibrator

Q . SLAM in AUTOWARE using 3 VLP16

- A. Unfortunately, if Autoware.AI does not currently support this feature we cannot provide support for it. Additionally, Autoware.AI has stopped accepting new features and is now in "maintenance mode" (only accepting bug fixes and clean-ups). See [this discourse post](#) for more details.

End-of-Life dates for Autoware.AI

■ Autoware



gbiggs Regular

1 Apr 21

As most Autoware users now know, the Autoware Foundation is working on the next generation of Autoware, known as [Autoware.Auto](#).

In order to focus our development resources on Autoware.Auto, the Autoware Foundation's Technical Steering Committee has taken a decision on the end-of-life date for Autoware.AI (the current generation of Autoware).

The dates have been set as follows:

- **From now:** Only minor releases will be made (1.14, 1.15, 1.16, ...) with the number of releases to be determined by the maintainers based on having something to release.
 - This means that API-breaking features cannot be released.
- **End of 2020:** Autoware.AI will enter maintenance mode. This means that:
 - No new features will be added
 - Bug fixes will be made if necessary (e.g. critical bugs, safety-relevant bugs)
 - Only patch releases (1.15.1, 1.15.2, etc.) will be made
- **End of 2022:** End of life. No releases will be made and no merge requests accepted. The source will still be available if someone wishes to fork and maintain it themselves, but the Autoware Foundation will not commit any resources to maintaining Autoware.AI beyond this date.

We currently expect to reach feature parity between Autoware.Auto and Autoware.AI by the end of 2020. Users of Autoware.AI are encouraged to begin planning their transition to Autoware.Auto this year so that they can be fully transitioned by the end of 2022.

We also encourage any current and future contributors to consider making their contributions to Autoware.Auto rather than Autoware.AI.

3. Multi Lidar Calibrator

This package allows to obtain the extrinsic calibration between two PointClouds with the help of the NDT algorithm.

The **multi_lidar_calibrator** node receives two PointCloud2 messages (parent and child), and an initialization pose. If possible, the transformation required to transform the child to the parent point cloud is calculated, and output to the terminal.

How to launch

1. You'll need to provide an initial guess, otherwise the transformation won't converge.
2. In a sourced terminal:

Using rosrn

```
roslaunch multi_lidar_calibrator multi_lidar_calibrator _points_child_src:=/lidar_child/points_raw _points_parent_src:=/lidar_parent/points_raw _x:=0.0 _y:=0.0 _z:=0.0 _roll:=0.0 _pitch:=0.0 _yaw:=0.0
```

Using roslaunch

```
roslaunch multi_lidar_calibrator multi_lidar_calibrator points_child_src:=/lidar_child/points_raw points_parent_src:=/lidar_parent/points_raw x:=0.0 y:=0.0 z:=0.0 roll:=0.0 pitch:=0.0 yaw:=0.0
```

1. Play a rosbag with both lidar data /lidar_child/points_raw and /lidar_parent/points_raw
2. The resulting transformation will be shown in the terminal as shown in the Output section.
3. Open RViz and set the fixed frame to the Parent
4. Add both point cloud /lidar_parent/points_raw and /points_calibrated
5. If the algorithm converged, both PointClouds will be shown in rviz.

Input topics

Output

1. Child Point cloud transformed to the Parent frame and published in /points_calibrated.
2. Output in the terminal showing the X,Y,Z,Yaw,Pitch,Roll transformation between child and parent. These values can be used later with the static_transform_publisher.

3. Multi Lidar Calibrator

Error list

Runtime manager – multi lidar calibrator

```
['roslaunch', 'multi_lidar_calibrator', 'multi_lidar_calibrator.launch', 'points_parent_src:=lidar0/points_raw', 'points_child_src:=lidar1/points_raw', 'ndt_epsilon:=0.01', 'ndt_step_size:=0.1', 'ndt_resolution:=1', 'ndt_iterations:=400', 'x:=0', 'y:=0', 'z:=0', 'roll:=0', 'pitch:=0', 'yaw:=0'] pid=12463 sched_policy=OTHER prio=0  
Failed connect to /tmp/autoware_proc_manager
```

Terminal

```
roslaunch multi_lidar_calibrator multi_lidar_calibrator _points_child_src:=/lidar_child/points_raw _points_parent_src:=/lidar_parent/points_raw _x:=0.0 _y:=0.0 _z:=0.0 _roll:=0.0 _pitch:=0.0 _yaw:=0.0
```

```
[ INFO] [1606969888.585903549]: [multi_lidar_calibrator] points_parent_src: /lidar_parent/points_raw  
[ INFO] [1606969888.586813307]: [multi_lidar_calibrator] points_child_src: /lidar_child/points_raw  
[ INFO] [1606969888.587108717]: [multi_lidar_calibrator] ndt_epsilon: 0.10  
[ INFO] [1606969888.587329276]: [multi_lidar_calibrator] voxel_size: 0.01  
[ INFO] [1606969888.587565770]: [multi_lidar_calibrator] ndt_step_size: 0.10  
[ INFO] [1606969888.587790699]: [multi_lidar_calibrator] ndt_resolution: 1.00  
[ INFO] [1606969888.587967571]: [multi_lidar_calibrator] ndt_iterations: 400  
[ INFO] [1606969888.589968550]: [multi_lidar_calibrator] Initialization Transform x: 0.00 y: 0.00 z: 0.00 roll: 0.00 pitch: 0.00 yaw: 0.00  
[ INFO] [1606969888.590926706]: [multi_lidar_calibrator] Subscribing to... /lidar_parent/points_raw  
[ INFO] [1606969888.591959543]: [multi_lidar_calibrator] Subscribing to... /lidar_child/points_raw  
[ INFO] [1606969888.592214832]: [multi_lidar_calibrator] Publishing PointCloud to... /points_calibrated  
[ INFO] [1606969888.592253502]: [multi_lidar_calibrator] Ready. Waiting for data...
```

3. Multi Lidar Calibrator

static_transform_publisher

static_transform_publisher x y z yaw pitch roll frame_id child_frame_id period_in_ms

Publish a static coordinate transform to tf using an x/y/z offset in meters and yaw/pitch/roll in radians

The period, in milliseconds, specifies how often to send a transform. 100ms (10hz) is a good value.

static_transform_publisher is designed both as a command-line tool for manual use, as well as for use within roslaunch files for setting static transforms.

For example:

<launch>

```
<node pkg="tf" type="static_transform_publisher" name="link1_broadcaster" args="1 0 0 0 0 0 1 link1_parent link1 100" />
```

</launch>



3. Multi Lidar Calibrator

How to use two VLP16(velodyne) with just one computer at a same time

<https://github.com/ros-drivers/velodyne/issues/108>

Evaluation of multiple lidar placement on a self-driving car in Autoware

https://iseauto.ttu.ee/wp-content/uploads/2018/06/Mihkel_Vali_Multiple_lidar_analysis.pdf

Setting up and calibrating multiple LiDAR sensors

<https://wowelec.wordpress.com/2019/06/18/setting-up-and-calibrating-multiple-lidar-sensors/>

Welcome to Autoware's documentation!

https://autoware.readthedocs.io/en/feature-documentation_rtd/index.html

https://github.com/AbangLZU/multi_lidar_calibration/tree/master/include

<https://github.com/themathgeek13>

https://github.com/JeongJae0815/Multi_Velodyne

