

Autonomous Car Lab

Practice 03. CAN ROS

28th March, 2019
Kichun Jo

Today

■ Follow up Practice

■ Assignment

Follow up Practice

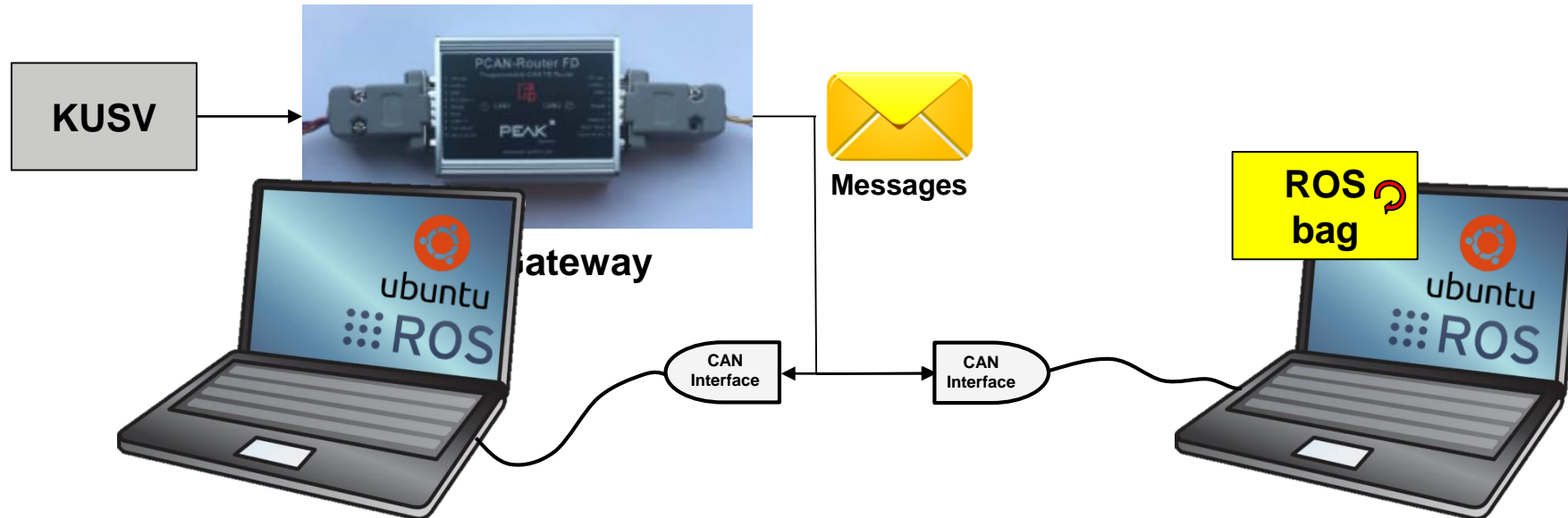
Communicate with CAN in ROS environment

■ Practice 1

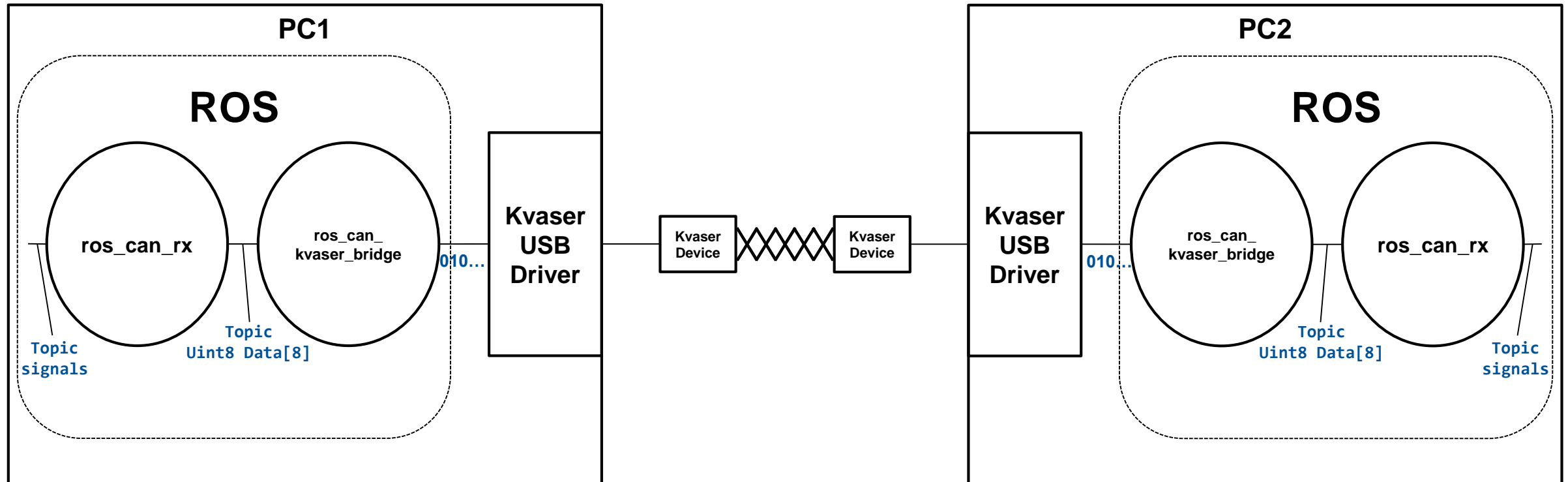
- ▶ PC to PC Communication using simple CAN message

■ Practice 2

- ▶ Receiving CAN Gateway messages from ROS bag file



Internal configuration



Install ROS CAN (I)

■ Check your device

- ▶ Connect Kvaser interface device and type
 - lsusb

■ Download “Practice_03_CAN_with_ROS” and extract

- ▶ Two files are available
 - linuxcan.tar.gz
 - ROS_CAN_PRACTICE.tar.gz

```
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon
visualization_msgs /opt/ros/kinetic/share/visualization_msgs
webkit_dependency /opt/ros/kinetic/share/webkit_dependency
xacro /opt/ros/kinetic/share/xacro
xmlrpcpp /opt/ros/kinetic/share/xmlrpcpp
skros@skros-ThinkPad-X230:~/ws_ros_can_sangkwon$ rospack list | grep test
roctest /opt/ros/kinetic/share/roctest
self_test /opt/ros/kinetic/share/self_test
test_publisher /home/skros/ws_ros_can_sangkwon/src/test_publisher
skros@skros-ThinkPad-X230:~/ws_ros_can_sangkwon$ lsusb
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 04f2:b2ea Chicony Electronics Co., Ltd Integrated Camera [ThinkPad]
Bus 001 Device 004: ID 0a5c:21e6 Broadcom Corp. BCM20702 Bluetooth 4.0 [ThinkPad]
Bus 001 Device 003: ID 147e:2020 Upek TouchChip Fingerprint Coprocessor (WBF advanced mode)
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 005: ID 0bfd:010b Kvaser AB
Bus 003 Device 003: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
skros@skros-ThinkPad-X230:~/ws_ros_can_sangkwon$
```

Install ROS CAN (II)

■ Install the linuxcan package

- ▶ `sudo cp ~/Downloads/Practice_03_CAN_with_ROS/linuxcan.tar.gz /usr/src`
- ▶ `cd /usr/src`
- ▶ `sudo tar -xvzf linuxcan.tar.gz`
- ▶ `cd linuxcan`
- ▶ `make`
- ▶ `sudo make install`

```
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon
install -m 644 libcanlib.so.1.5.0 /usr/lib/
ln -sf libcanlib.so.1.5.0 /usr/lib/libcanlib.so
ln -sf libcanlib.so.1.5.0 /usr/lib/libcanlib.so.1
/sbin/ldconfig
install -m 644 ../include/canlib.h /usr/include
install -m 644 ../include/canstat.h /usr/include
install -m 644 ../include/obsolete.h /usr/include
mkdir -p /usr/doc/canlib
cp -r ../doc/HTMLhelp /usr/doc/canlib
make[1]: Leaving directory '/usr/src/linuxcan/canlib'
make -C linlib install
make[1]: Entering directory '/usr/src/linuxcan/linlib'
rm -f /usr/lib/liblinlib.so
rm -f /usr/lib/liblinlib.so.1
install -m 644 liblinlib.so.1.5.0 /usr/lib/
ln -sf liblinlib.so.1.5.0 /usr/lib/liblinlib.so
ln -sf liblinlib.so.1.5.0 /usr/lib/liblinlib.so.1
/sbin/ldconfig -X
install -m 644 ../include/linlib.h /usr/include
make[1]: Leaving directory '/usr/src/linuxcan/linlib'
skros@skros-ThinkPad-X230:/usr/src/linuxcan$ la
10-kvaser.rules  COPYING      include  mhydra  pciids  pciids
canlib          COPYING.BSD  test    moduleinfo.txt  README
common          COPYING.GPL  liblin  pciids  pciids  pciids
```

■ Check your device information

- ▶ `cd /usr/src/linuxcan/canlib/examples`
- ▶ `./listChannels`

```
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon
ln -sf liblinlib.so.1.5.0 /usr/lib/liblinlib.so.1
/sbin/ldconfig -X
install -m 644 ../include/linlib.h /usr/include
make[1]: Leaving directory '/usr/src/linuxcan/linlib'
skros@skros-ThinkPad-X230:/usr/src/linuxcan$ la
10-kvaser.rules  COPYING      include  mhydra  pciids  pciids
canlib          COPYING.BSD  test    moduleinfo.txt  README
common          COPYING.GPL  liblin  pciids  pciids  pciids
config.mak      Makefile    Makefile  pciids  pciids  pciids
skros@skros-ThinkPad-X230:/usr/src/linuxcan$ cd canlib/examples/
skros@skros-ThinkPad-X230:/usr/src/linuxcan/canlib/examples$ ./listChannels
CANlib version 5.27
Found 0 channel(s).
skros@skros-ThinkPad-X230:/usr/src/linuxcan/canlib/examples$ ./listChannels
CANlib version 5.27
Found 0 channel(s).
skros@skros-ThinkPad-X230:/usr/src/linuxcan/canlib/examples$ ./listChannels
CANlib version 5.27
Found 2 channel(s).
ch 0: Kvaser Hybrid 2xCAN/LIN 73-30130-00965-3, s/n 10203, v3.9.467 (mhydra v
8.27.776)
ch 1: Kvaser Hybrid 2xCAN/LIN 73-30130-00965-3, s/n 10203, v3.9.467 (mhydra v
8.27.776)
skros@skros-ThinkPad-X230:/usr/src/linuxcan/canlib/examples$ cd ~/
skros@skros-ThinkPad-X230:~$ sudo apt-get install ros-kinetic-ros-canopen
```

■ Install the ros-canopen library

- ▶ `sudo apt-get install ros-kinetic-ros-canopen`
- ▶ `sudo apt-get update`

■ Make your workspace for practice

- ▶ `cd ~/`
- ▶ `mkdir ws_ros_can_NAME`

■ Install CAN practice package

- ▶ Move 'ROS_CAN_PRACTICE.tar.gz' to workspace
- ▶ `tar -xvzf ROS_CAN_PRACTICE.tar.gz`
- ▶ `cd src`
- ▶ `catkin_init_workspace`
- ▶ `cd .. && catkin_make`
- ▶ Source `devel/setup.bash`

PC to PC Communication (I)

■ Prepare for Kvaser driver

- ▶ `cd ~/ws_ros_can_NAME/src/ros_can/launch`
- ▶ Modify 'kvaser_can_bridge.launch'
 - `gedit kvaser_can_bridge.launch`
 - `can_hardware_id`: kvaser id (S/N)
 - `can_circuit_id`: CAN channel
 - » CAN 1 = 0
 - » CAN 2 = 1
 - `can_bit_rate`: 500000 or ...
- ▶ Execute 'kvaser_can_bridge.launch' file
 - `roslaunch ros_can kvaser_can_bridge.launch`

```
can_bridge.launch (~ws_ros_can_sangkwon/src/ros_can/launch) - gedit
Open  [?]
<?xml version="1.0"?>
<launch>
  <arg name="can_hardware_id" default="10203" />
  <arg name="can_circuit_id" default="0" />
  <arg name="can_bit_rate" default="500000" />

  <arg name="can_hardware_id2" default="10203" />
  <arg name="can_circuit_id2" default="1" />
  <arg name="can_bit_rate2" default="500000" />

  <node pkg="ros_can" type="ros_can_kvaser_bridge" name="ros_can_kvaser_bridge">
    <param name="can_hardware_id" value="$(arg can_hardware_id)" />
    <param name="can_circuit_id" value="$(arg can_circuit_id)" />
    <param name="can_bit_rate" value="$(arg can_bit_rate)" />

    <param name="can_hardware_id2" value="$(arg can_hardware_id2)" />
    <param name="can_circuit_id2" value="$(arg can_circuit_id2)" />
    <param name="can_bit_rate2" value="$(arg can_bit_rate2)" />
  </node>
</launch>
```

```
skros@skros-ThinkPad-X230:~/ws_ros_can_sangkwon/src/ros_can/launch$ roslaunch ros_can kvaser_can_bridge.launch
... logging to /home/skros/.ros/log/eea70116-50bc-11e9-85b2-606720c30370/roslaunch-skros-ThinkPad-X230-23623.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://skros-ThinkPad-X230:33601/

SUMMARY
=====

PARAMETERS
* /ros_can_kvaser_bridge/can_bit_rate2: 500000
* /ros_can_kvaser_bridge/can_bit_rate: 500000
* /ros_can_kvaser_bridge/can_circuit_id2: 1
* /ros_can_kvaser_bridge/can_circuit_id: 0
* /ros_can_kvaser_bridge/can_hardware_id2: 10203
* /ros_can_kvaser_bridge/can_hardware_id: 10203
* /roscdistro: kinetic
* /rosversion: 1.12.14

NODES
/
  ros_can_kvaser_bridge (ros_can/ros_can_kvaser_bridge)

auto-starting new master
process[master]: started with pid [23633]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to eea70116-50bc-11e9-85b2-606720c30370
process[rosout-1]: started with pid [23646]
started core service [/rosout]
process[ros_can_kvaser_bridge-2]: started with pid [23660]
```

PC to PC Communication (II)

■ Send test message

- ▶ `roslaunch test_publisher test_publisher_node ID DLC DATA`
- ▶ After 'source devel/setup.bash' in your workspace
 - `rostopic echo /can_tx`

```
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can home/skros/ws_ros_can_sangkwon/src/ros_can/launch/kvaser_can_bridge.launch http://
SUMMARY
=====
PARAMETERS
* /ros_can_kvaser_bridge/can_bit_rate2: 500000
* /ros_can_kvaser_bridge/can_bit_rate: 500000
* /ros_can_kvaser_bridge/can_circuit_id2: 1
* /ros_can_kvaser_bridge/can_circuit_id: 0
* /ros_can_kvaser_bridge/can_hardware_id2: 10203
* /ros_can_kvaser_bridge/can_hardware_id: 10203
* /roscpp: kinetic
* /rosversion: 1.12.14

NODES
/
  ros_can_kvaser_bridge (ros_can/ros_can_kvaser_bridge)

auto-starting new master
process[master]: started with pid [23633]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to eea70116-50bc-11e9-85b2-606720c30370
process[rosout-1]: started with pid [23646]
started core service [/rosout]
process[ros_can_kvaser_bridge-2]: started with pid [23660]
^C[rosout-1] killing on exit
[rosout-1] killing on exit
[master] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can/la
..
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can$ r
Segmentation fault (core dumped)
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can$ r
t_publisher test_publisher_node 256 8 11
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can$
t_publisher test_publisher_node 256
```

```
S
an_kvaser_bridge/can_bit_rate2: 500000
an_kvaser_bridge/can_bit_rate: 500000
an_kvaser_bridge/can_circuit_id2: 1
an_kvaser_bridge/can_circuit_id: 0
an_kvaser_bridge/can_hardware_id2: 10203
an_kvaser_bridge/can_hardware_id: 10203
stro: kinetic
rsion: 1.12.14

an_kvaser_bridge (ros_can/ros_can_kvaser_bridge)

ting new master

skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon
secs: 1553711616
nsecs: 611295605
frame_id: "0"
id: 256
is_rtr: False
is_extended: False
is_error: False
dlc: 8
data: [11, 11, 11, 11, 11, 11, 11, 11]

header:
seq: 14633
stamp:
secs: 1553711616
nsecs: 631357373
frame_id: "0"
id: 256
is_rtr: False
is_extended: False
is_error: False
dlc: 8
data: [11, 11, 11, 11, 11, 11, 11, 11]
---
```

Receiving messages from ROS bag file (I)

■ Execute ros_can_rx

- ▶ export ROSCONSOLE_FORMAT="{message}"
- ▶ rosrun ros_can ros_can_rx

■ Play ROS bag

- ▶ rosbag play ~/ws_ros_can_NAME/src/konkuk_rosbag_2019-03-26-22-26-00.bag

```
skros@skros-ThinkPad-X230: ~/ws_ros_can_sangkwon/src/ros_can home/skros/ws_ros_can_sangkwon/src/ros_can/launch/kvaser_ca
[RUNNING] Bag Time: 1553606814.207113 Duration: 53.200691 / 299.189
[RUNNING] Bag Time: 1553606814.215770 Duration: 53.209348 / 299.189 S
[RUNNING] Bag Time: 1553606814.216331 Duration: 53.209908 / 299.189 an_kvaser_bridge/can_bit_rate2: 500000
[RUNNING] Bag Time: 1553606814.216676 Duration: 53.210254 / 299.189 an_kvaser_bridge/can_bit_rate: 500000
[RUNNING] Bag Time: 1553606814.216696 Duration: 53.210274 / 299.189 an_kvaser_bridge/can_circuit_id2: 1
[RUNNING] Bag Time: 1553606814.222689 Duration: 53.216267 / 299.189 an_kvaser_bridge/can_circuit_id: 0
[RUNNING] Bag Time: 1553606814.222802 Duration: 53.216380 / 299.189 an_kvaser_bridge/can_hardware_id2: 10203
[RUNNING] Bag Time: 1553606814.226174 Duration: 53.219752 / 299.189 an_kvaser_bridge/can_hardware_id: 10203
[RUNNING] Bag Time: 1553606814.227470 Duration: 53.221048 / 299.189 stro: kinetic
[RUNNING] Bag Time: 1553606814.227584 Duration: 53.221161 / 299.189 rsion: 1.12.14
[RUNNING] Bag Time: 1553606814.227842 Duration: 53.221420 / 299.189
[RUNNING] Bag Time: 1553606814.227866 Duration: 53.221444 / 299.189
[RUNNING] Bag Time: 1553606814.228005 Duration: 53.221583 / 299.189
[RUNNING] Bag Time: 1553606814.228319 Duration: 53.221897 / 299.189 an_kvaser_bridge (ros_can/ros_can_kvaser_bridge)
[RUNNING] Bag Time: 1553606814.228896 Duration: 53.222474 / 299.189
[RUNNING] Bag Time: 1553606814.228976 Duration: 53.222554 / 299.189 ting new master
[RUNNING] Bag Time: 1553606814.229160 Duration: 53.222737 / 299.189
[RUNNING] Bag Time: 1553606814.236406 Duration: 53.229984 / 299.189
[RUNNING] Bag Time: 1553606814.236518 Duration: 53.230096 / 299.189 [98895] ID: 0x102
[RUNNING] Bag Time: 1553606814.242238 Duration: 53.235815 / 299.189 [I INFO] [1553712440.425379529]:
[RUNNING] Bag Time: 1553606814.242600 Duration: 53.236238 / 299.189 [98896] ID: 0x103
[RUNNING] Bag Time: 1553606814.246025 Duration: 53.239602 / 299.189 [I INFO] [1553712440.425427257]:
[RUNNING] Bag Time: 1553606814.246139 Duration: 53.239717 / 299.189 [98897] ID: 0x100
[RUNNING] Bag Time: 1553606814.247151 Duration: 53.240725 / 299.189 [I INFO] [1553712440.425456953]: GW1:B3 2 B4 82 B4 C2 B7 C2
[RUNNING] Bag Time: 1553606814.247671 Duration: 53.241246 / 299.189 [I INFO] [1553712440.425506596]: FR: 21.593750
[RUNNING] Bag Time: 1553606814.247761 Duration: 53.241336 / 299.189 / RL: 21.625000
[RUNNING] Bag Time: 1553606814.247792 Duration: 53.241376 / 299.189 / RR: 21.625000
[RUNNING] Bag Time: 1553606814.247889 Duration: 53.241466 / 299.189 / FL: 21.718750
[RUNNING] Bag Time: 1553606814.247907 Duration: 53.241485 / 299.189 [I INFO] [1553712440.434000512]:
[RUNNING] Bag Time: 1553606814.256221 Duration: 53.249798 / 299.189 [98898] ID: 0x101
[RUNNING] Bag Time: 1553606814.257299 Duration: 53.250877 / 299.189 [I INFO] [1553712440.435148527]:
[RUNNING] Bag Time: 1553606814.257652 Duration: 53.251230 / 299.189 [98899] ID: 0x102
[RUNNING] Bag Time: 1553606814.257799 Duration: 53.251376 / 299.189 [I INFO] [1553712440.435192083]:
[RUNNING] Bag Time: 1553606814.262537 Duration: 53.256114 / 299.189 [98900] ID: 0x103
[RUNNING] Bag Time: 1553606814.262722 Duration: 53.256299 / 299.189 [I INFO] [1553712440.436010562]:
[RUNNING] Bag Time: 1553606814.266136 Duration: 53.259714 / 299.189 [98903] ID: 0x100
[RUNNING] Bag Time: 1553606814.266464 Duration: 53.260047 / 299.189 [I INFO] [1553712440.436039877]: GW1:B0 42 A3 2 AE C2 B5 2
[RUNNING] Bag Time: 1553606814.266783 Duration: 53.260366 / 299.189 [I INFO] [1553712440.436062961]: FR: 21.500000
[RUNNING] Bag Time: 1553606814.267323 Duration: 53.260906 / 299.189 / RL: 21.093750
[RUNNING] Bag Time: 1553606814.267545 Duration: 53.261127 / 299.189 / RR: 21.437500
[RUNNING] Bag Time: 1553606814.267824 Duration: 53.261402 / 299.189 / FL: 21.656250
```

Assignment

■ Design the database of CAN gateway

- ▶ Open 'CAN_rx.cpp' file and fill //TODO//
 - 세부 내용 설명~~~~~
- ▶ Submit a compressed file of your workspace directory 'ws_ros_can_NAME'
 - NAME.zip, ...
- ▶ Due date: ~4/4

```
uint8_t Gway_Wheel_Velocity_RL_H : 6;
uint8_t reserve2 : 2;
uint8_t Gway_Wheel_Velocity_RR_L;
uint8_t Gway_Wheel_Velocity_RR_H : 6;
uint8_t reserve3 : 2;
uint8_t Gway_Wheel_Velocity_FL_L;
uint8_t Gway_Wheel_Velocity_FL_H : 6;
uint8_t reserve4 : 2;
}Str;
}CAN_MSG_GWAY1;

typedef union _CAN_MSG_GWAY2_
{
//*****TODO*****//
}CAN_MSG_GWAY2;

typedef union _CAN_MSG_GWAY3_
{
//*****TODO*****//
}CAN_MSG_GWAY3;

typedef union _CAN_MSG_GWAY4_
{
//*****TODO*****//
}CAN_MSG_GWAY4;

typedef union _CAN_MSG_GWAY5_
{
//*****TODO*****//
}CAN_MSG_GWAY5;
```

```
case CAN_ID_GWAY1:
memcpy(gway1.CAN_GWAY1_Data, &L_msg->data[0], sizeof(test_buf));
FL = (float)((int16_t)((gway1.Str.Gway_Wheel_Velocity_FL_H << 8) + gway1.Str.Gway_Wheel_Velocity_FL_L));
FL = FL * (0.03125);
FR = (float)((gway1.Str.Gway_Wheel_Velocity_FR_H << 8) + gway1.Str.Gway_Wheel_Velocity_FR_L);
FR = FR * (0.03125);
RL = (float)((gway1.Str.Gway_Wheel_Velocity_RL_H << 8) + gway1.Str.Gway_Wheel_Velocity_RL_L);
RL = RL * (0.03125);
RR = (float)((gway1.Str.Gway_Wheel_Velocity_RR_H << 8) + gway1.Str.Gway_Wheel_Velocity_RR_L);
RR = RR * (0.03125);

ROS_INFO("GM1: %X %X %X %X %X %X", gway1.CAN_GWAY1_Data[0], gway1.CAN_GWAY1_Data[1], gway1.CAN_GWAY1_Data[2], gway1.CAN_GWAY1_Data[3], gway1.CAN_GWAY1_Data[4], gway1.CAN_GWAY1_Data[5], gway1.CAN_GWAY1_Data[6], gway1.CAN_GWAY1_Data[7]);
ROS_INFO("FR: %f\n RL: %f\n RR: %f\n FL: %f", FR, RL, RR, FL);
break;

case CAN_ID_GWAY2:
//memcpy(gway2.CAN_GWAY2_Data, &L_msg->data[0], sizeof(test_buf));
//*****TODO*****//

break;

case CAN_ID_GWAY3:
//memcpy(gway3.CAN_GWAY3_Data, &L_msg->data[0], sizeof(test_buf));
//*****TODO*****//

break;

case CAN_ID_GWAY4:
//memcpy(gway4.CAN_GWAY4_Data, &L_msg->data[0], sizeof(test_buf));
//*****TODO*****//

break;

case CAN_ID_GWAY5:
//memcpy(gway5.CAN_GWAY5_Data, &L_msg->data[0], sizeof(test_buf));
//*****TODO*****//
```

Thank you