Autonomous Car Lab

Practice 03. CAN ROS

28th March, 2019

Kichun Jo

KU KONKUK UNIVERSITY

Today

- Follow up Practice
- Assignment

Follow up Practice

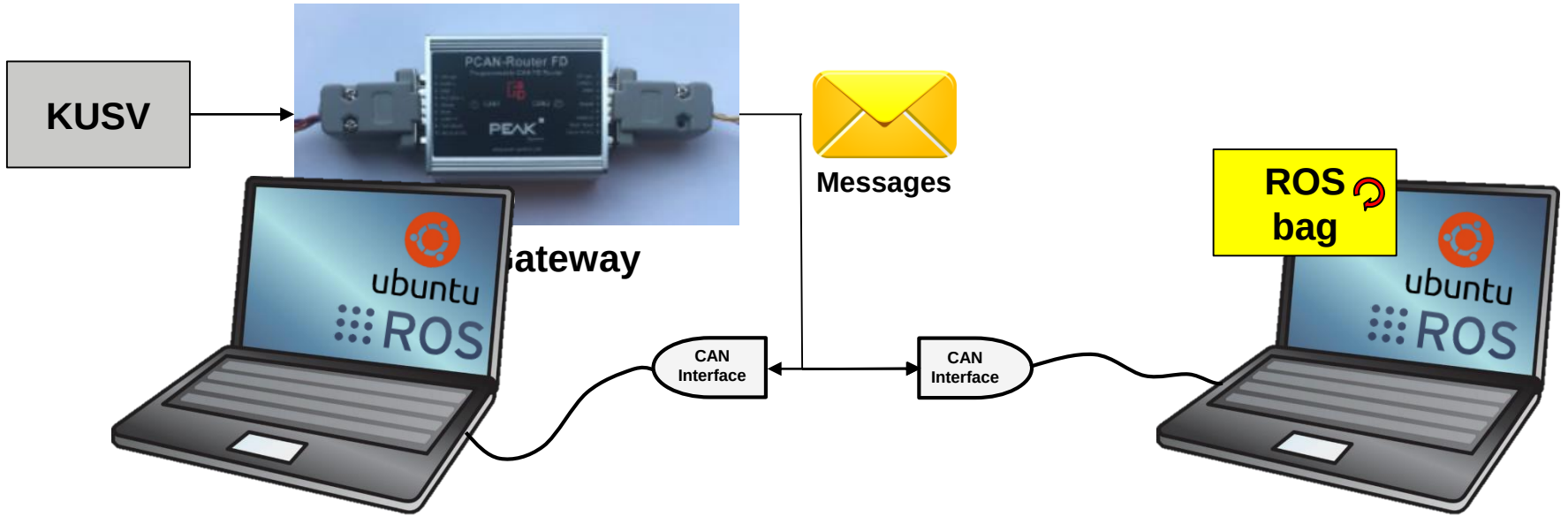
Communicate with CAN in ROS environment

Practice 1

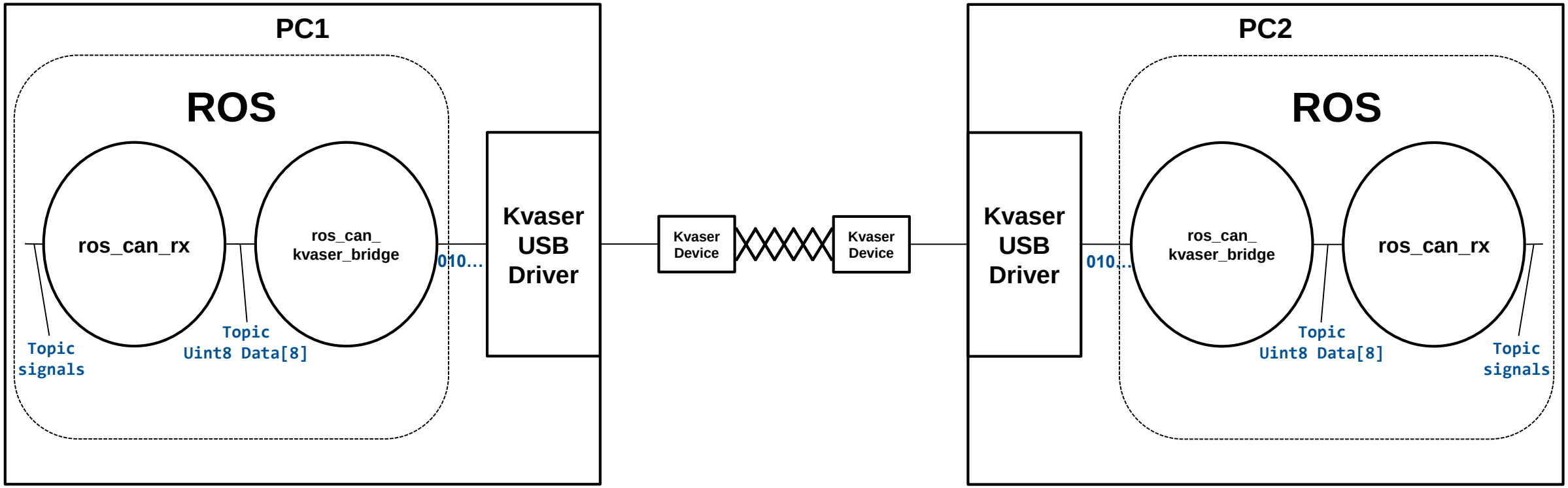
- PC to PC Communication using simple CAN message

Practice 2

- Receiving CAN Gateway messages from ROS bag file



Internal configuration



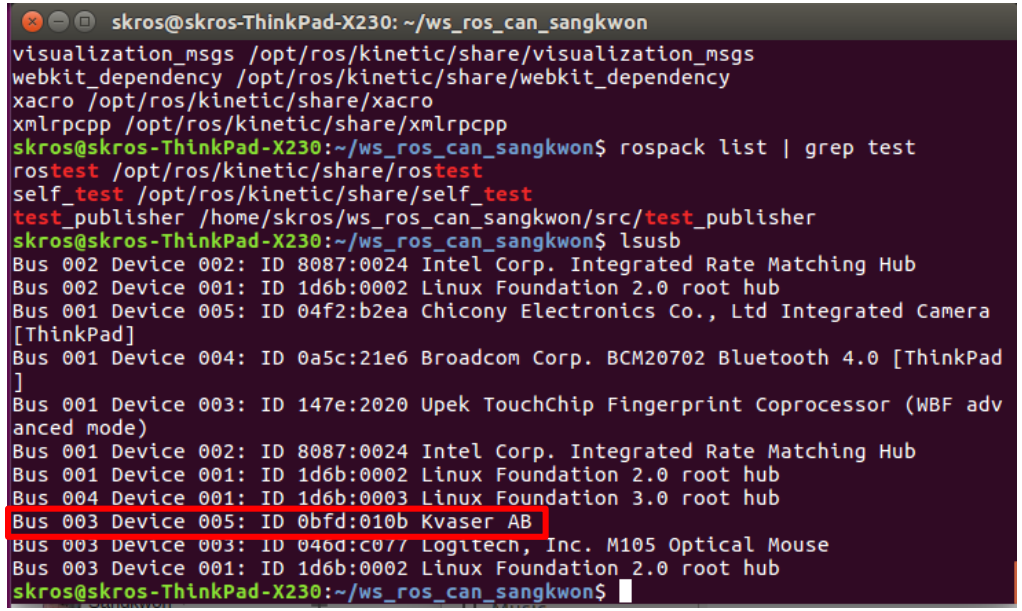
Install ROS CAN (I)

Check your device

- Connect Kvaser interface device and type
  - lsusb

Download “Practice_03_CAN_with_ROS” and extract

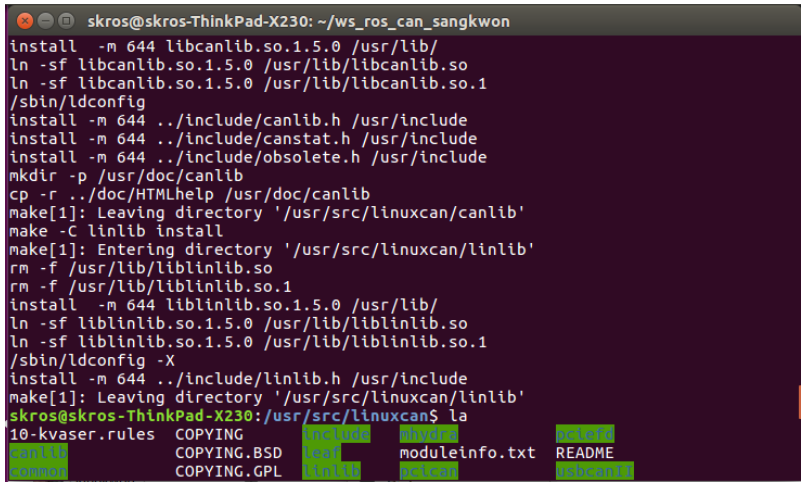
- Two files are available
  - linuxcan.tar.gz
  - ROS_CAN_PRACTICE.tar.gz



Install ROS CAN (II)

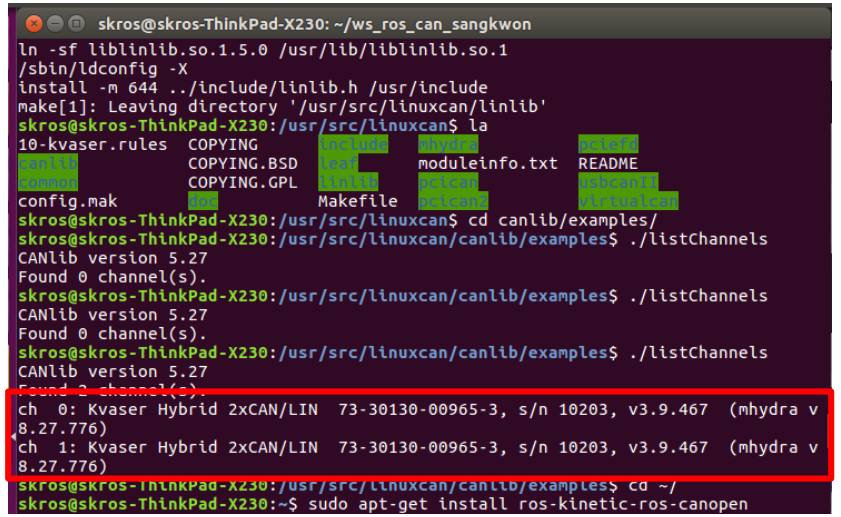
Install the linuxcan package

- 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



Check your device information

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



KU KONKUK UNIVERSITY

Install ROS CAN (III)

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

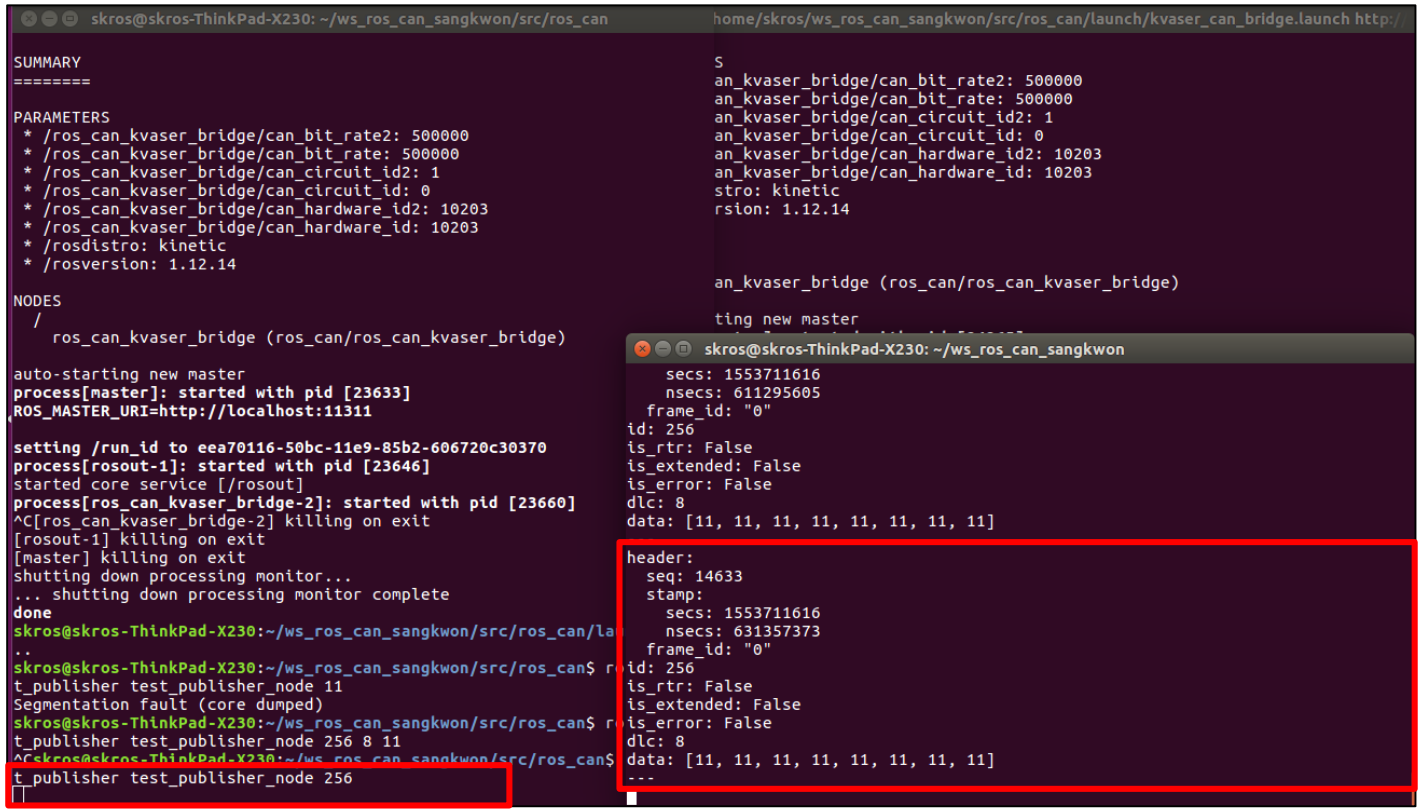
```
<?xml version="1.0"?>
<launch>
  <arg name="can_hardware_id" default="10203" />
  <arg name="can_circuit_id" default="0" />
  <arg name="can_bit_rate" default="500000" />

  <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$ ros
launch ros_can kvaser_can_bridge.launch
... logging to /home/skros/.ros/log/eea70116-50bc-11e9-85b2-606720c3037
0/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
 * /rosdistro: 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

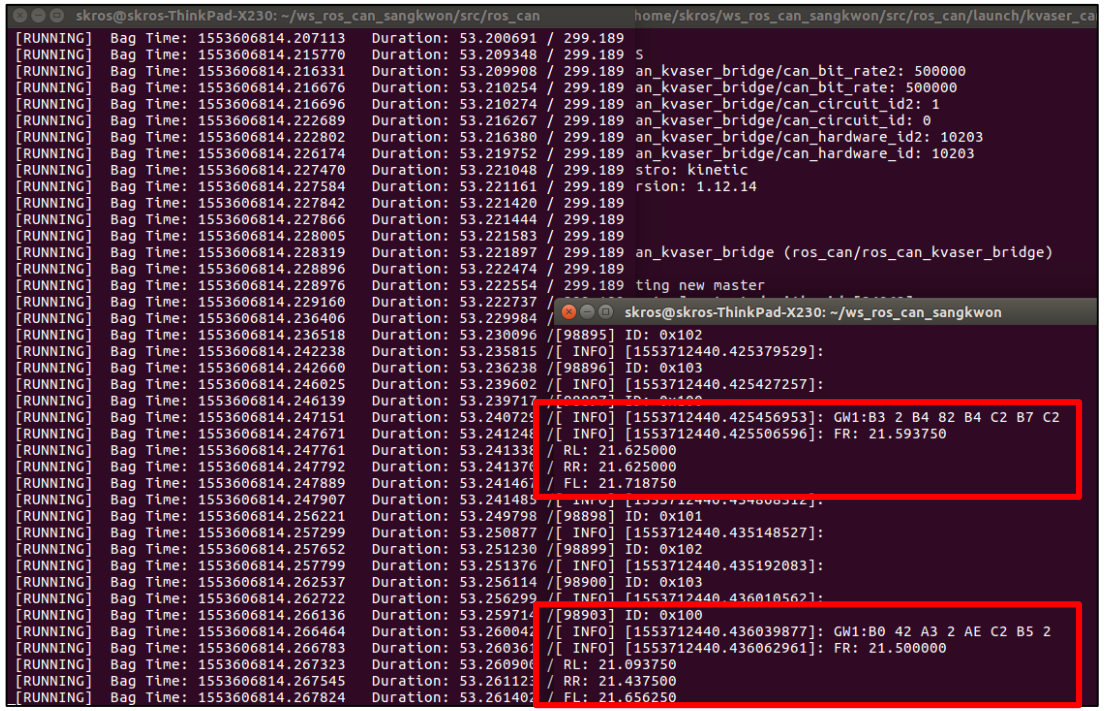
- rosrun test_publisher test_publisher_node ID DLC DATA
- After ‘source devel/setup.bash’ in your workspace
  - rostopic echo /can_tx



KU KONKUK UNIVERSITY

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



Assignment

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;
```

C++ Tab Width: 8 Ln 20, Col 18 INS

```
                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("GW1:%X %X %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**************//
```

C++ Tab Width: 8 Ln 20, Col 18 INS

Thank you

KU KONKUK UNIVERSITY