Experiment 2

Name	Hassan Al-Faiz Bin Mohamed Ahmed (HB17031)
Group Number	
Date	18/8/2020

2. Introduction to CANoe

CANoe is a software component for development, testing and analysis of control unit networks. In the automotive industry, this tool is a tool of common use for the simulation of communication networks.

By the means of CANoe, the developer has the possibility to monitor the bus communication between distributed systems, and respectively, for simulating different states.

All the files requested for this experiment are shared online.

2.1 Introduction

The software component CANoe published by the enterprise Vector is available as free download for the work in the lab.

- As a first step, learn how to use the user surface of the program and use the sample configurations as well as the online tutorials (there are tutorials published by Vector available online) for getting familiar with the program after proceeding on this second experiment.
- Get familiar with these notions: CANdb++ editor (signals, messages), trace window, CANoe panel, network node, database, communication matrix.

It is necessary to be well prepared for this experiment!

2.2 Bosch rotation rate sensor

Rotation rate sensors belong to the group of inertial sensors and they measure, besides the lateral acceleration in x, y and z direction, the rotation speed of vehicles in relation to the vehicle's vertical axis (z axis).

The sensor is connected to the working space in the lab and is sending cyclically different signal values, via two CAN messages on the lab's CAN. For observing this, watch the video "VE TI Lab 2.2" available online.

2.3 CANoe configuration

Therefore, consider the video related to experiment 2.3 available online.

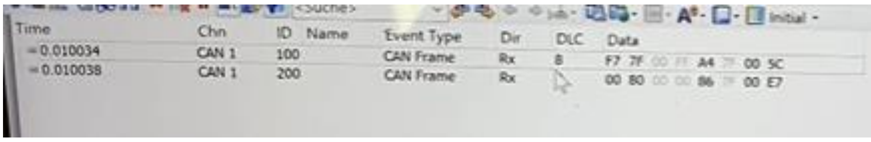
The hardware VN1630 is getting connected to the PC and a new CANoe configuration is getting created. Afterwards, a measurement is being proceeded.

a) How many messages are visible in the Trace window?
   2 CAN messages with various information such as time (cyclic time receive and sent), channel number, ID, Event type, Direction, Data length code (DLC), data (user data)

b) What are their identifiers?
   100 and 200

c) What is the DLC you can see in the messages?
   The data length code is valued at 8 (bytes)

d) What is the cyclic time (Δ t) the messages are send with?
   About 0.01 ms for both messages



2.4 Database

By the means of databases, the different signals contained in the messages can be filtered and decoded. Usually, the signal and message information is defined by the producer of the control unit or the sensor/actuator.

According to the data sheet of the present rotation rate sensor, you can find the following information concerning the messages and the signals:

Drehgeschwindigkeit

Identifier:	0x100
Byte:	0 und 1
Min-Wert:	-163.84 ◦/s = 0x0
Mittel-Wert:	0 ◦/s = 0x8000
Max-Wert:	+163.84 ◦/s = 0xFFFE Quantisierung:
Quantisierung:	0.005 ◦/s/digit

Querbeschleunigung in Y-Richtung

Identifier:	0x100
Byte:	4 und 5
Min-Wert:	-4.1768 g = 0x0
Mittel-Wert:	0 g = 0x8000
Max-Wert:	+4.1768 g = 0xFFFE
Quantisierung:	0.000127465 g/digit

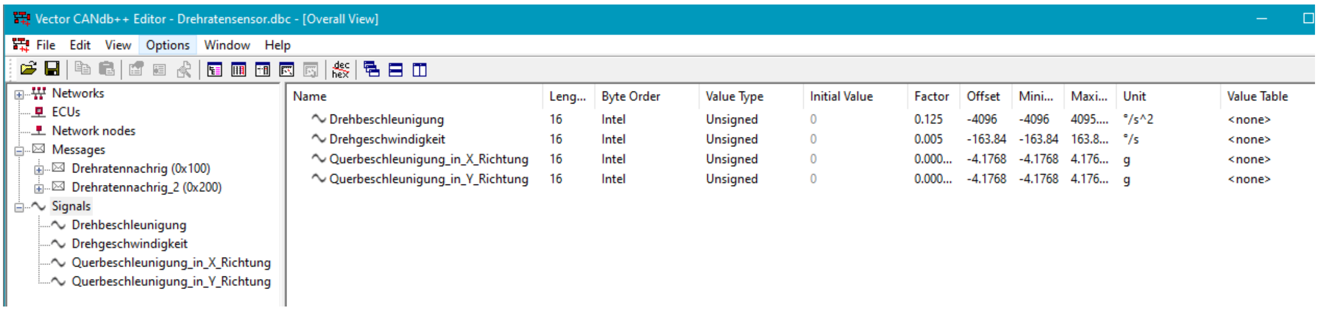
Drehbeschleunigung

Identifier:	0x200
Byte:	0 und 1
Min-Wert:	-4096 $\circ/s^2$ = 0x0
Mittel-Wert:	0 $\circ/s^2$ = 0x8000
Max-Wert:	4095 $\circ/s^2$ = 0xFFFE
Quantisierung:	0.125 $\circ/s^2$/digit

Querbeschleunigung in X-Richtung

Identifier:	0x200
Byte:	4 und 5
Min-Wert:	-4.1768 g = 0x0
Mittel-Wert:	0 g = 0x8000
Max-Wert:	+4.1768 g = 0xFFFE
Quantisierung:	0.000127465 g/digit

Result:



Pay attention: all the messages are sent in intel format.

- Watch the 3 corresponding videos available online.
- Open the database editor by *Tools -> CANdb++ Editor*
- Create a new database by a CANoe template and save the file using the name Drehratensensor.dbc
- On the left hand of the editor, you can add networks, control units, messages and signals etc. Create the four measurands seen above (in the tab “signals”, by using right-hand click).
  Take into consideration the correct *transformation formula*, as well as the units. *(Note: in the tab definition -> inertial value)*
- By using the tab Layout in the message properties, you can connect the signals already defined to the databytes of the CAN message, according to the information in the data sheet. Save all the settings.

Now add the database you created before to the CANoe simulation setup (simulation -> simulation setup). On the right hand (figure 2.1) you can find a network overview where you can add databases by right-hand clicking on databases -> add). You can see the running simulation in the video.

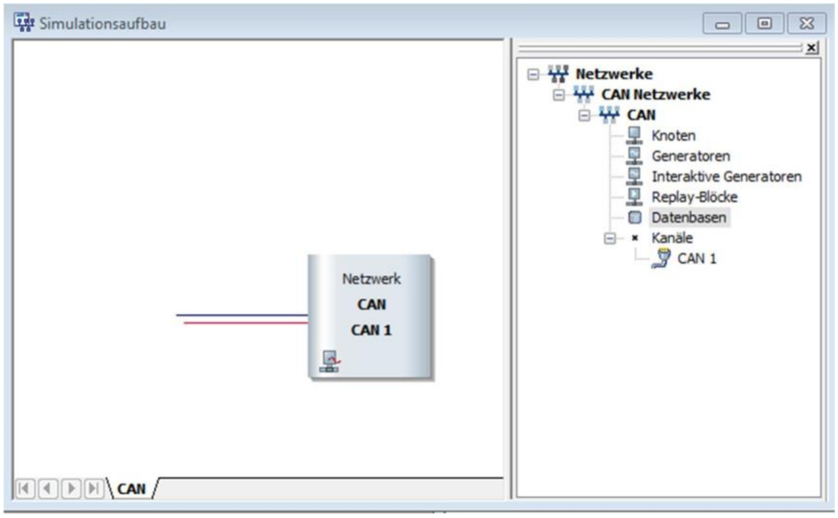
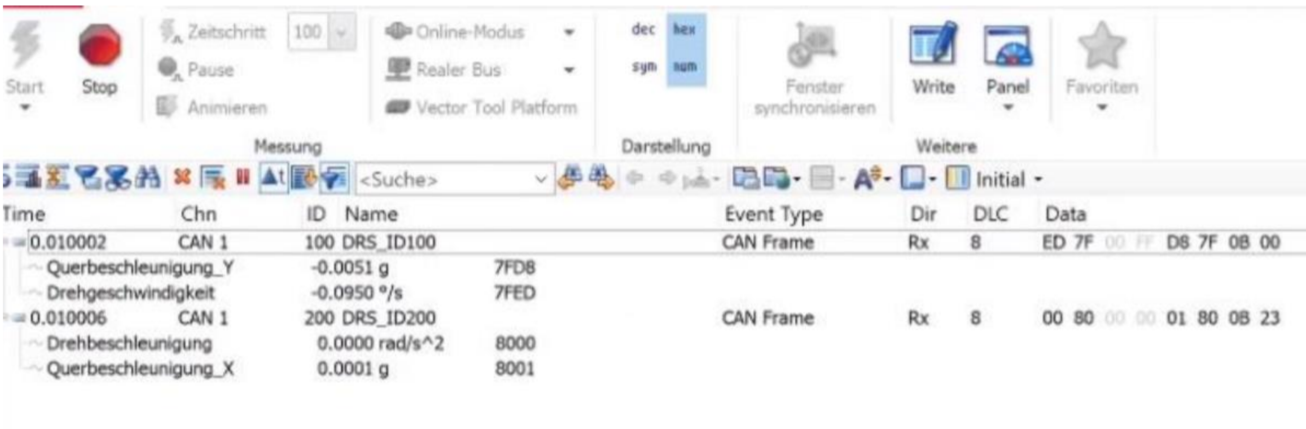


Figure 1: simulation setup

e) What is different when looking at the message in the Trace window now?
- In CANdb++ Editor, the number of signals per message depends on their respective length.
- Length of one bit usually won't be enough so we need to concider more data (between 8 to 16 bits)
- In the trace window, we can see the different signal and messages that added to the CAN Bus, additional data when starting the simulation.



2.4.1 Creating a CANoe panel

Panels are used for visualization, monitoring as well as for controlling the content of the signals and the CAN messages.

By linking the panels and the database, you can connect signals directly to pushbuttons, control instruments or displays.

- Open the Panel Designer by *Tools -> Panel Designer*
- Create a panel for visualizing the different sensor signals of the rotation rate sensor.
- On the following figure 2.2, you can find an example of how to visualize the signals. The different elements can be selected by the toolbox on the right hand and fixed on the panel's surface by drag and drop.
- By right-hand clicking on the corresponding element, you can connect it to signals contained in the database you created before.

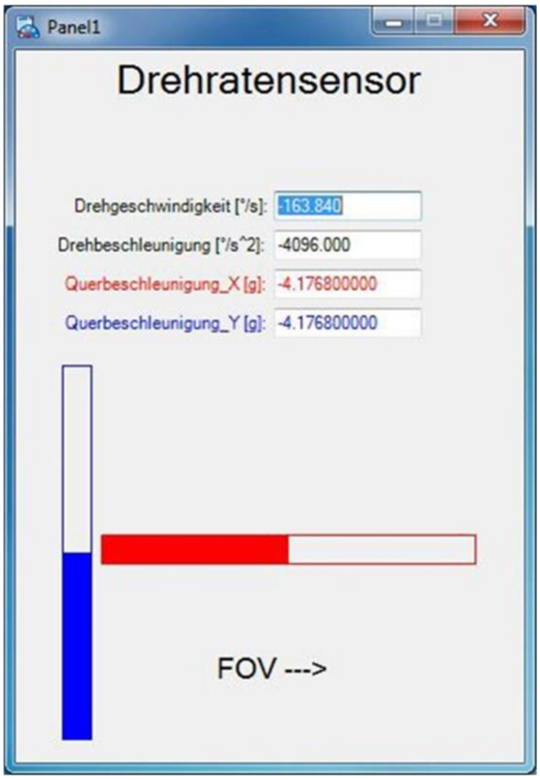


figure 2.2: example of a CANoe panel

- Save the panel you have created. If ever it disappears on your screen, you can find it again in the tab *Home -> Panel*.

2.5 Signal simulation by a real instrument cluster

The instrument cluster (taken from a Mercedes Benz S class vehicle) used here is a central element for visualizing important information in and about the vehicle.

2.5.1 Activating the instrument cluster

The instrument cluster is one out of many control units used in a Mercedes Benz S class vehicle. It is operated only by using the CAN bus with a speed of 500 kbits/s. Besides the CAN signals, the instrument cluster needs an operating voltage of 13.8V (terminal 30 and 31 in the real vehicle). Just after connecting it to the operating voltage, you cannot see any messages sent by the instrument cluster yet. Only by sending the message *Klemmenstatus* via CANoe, the instrument cluster will wake up from sleep mode.

The instrument cluster is getting connected now and a new configuration is getting created, afterwards, the simulation is started (see the video 2.5 available online).

a) What can you observe in the trace window (in the video)? Are there any messages that are sent from the instrument cluster?
   <u>In the trace window, we can see that there are messages sent from instrument cluster with 2 different ID (klemmenstatus = 0x1 and Wheel_rpm = 0x203), with each containing 8 data length code. Both messages sent in cyclic Tx method. Klemmenstatus with 100 cyclic time while Wheel_rpm with 20 cyclic time.</u>

b) Do you need a terminating resistor for the instrument cluster? If yes, why?
   <u>Yes, because it will eliminate the reflective signal. Neglecting the terminating resistor will create noise therefore no communication would happen such as bit error and other communications will stop</u>.

Now, for waking up the instrument cluster from sleep mode, the state of the ignition has to be transmitted by a CAN message. The corresponding signals and transmission properties have to be defined by the means of a database.

- Therefore, open the file *Kombiinstrument_empty.dbc* (available online) in the CANdb++ editor.
- Afterwards, create a signal called "*Klemmeninformation*" and position it at the correct place in the corresponding message called "*Klemmenstatus*".
- The behavior of this message, during its transmission, has to be adapted in the tab Attributes in the message settings, according to the information below (*GenMsgCycleTime* -> Cyclic time has to be in *ms*, *GenMsgSendType* -> has to be *cyclic*).

Software-Klemmen	
Botschaftsname:	Klemenstatus
Identifier:	0x1
Byteanzahl:	8
Zyklus:	100 ms
Signalname:	Klemmeninformation
Byte:	0
Bit:	0 bis 3
Wertetabelle:	
IGN_OFF	0x1
IGN_ON	0x4

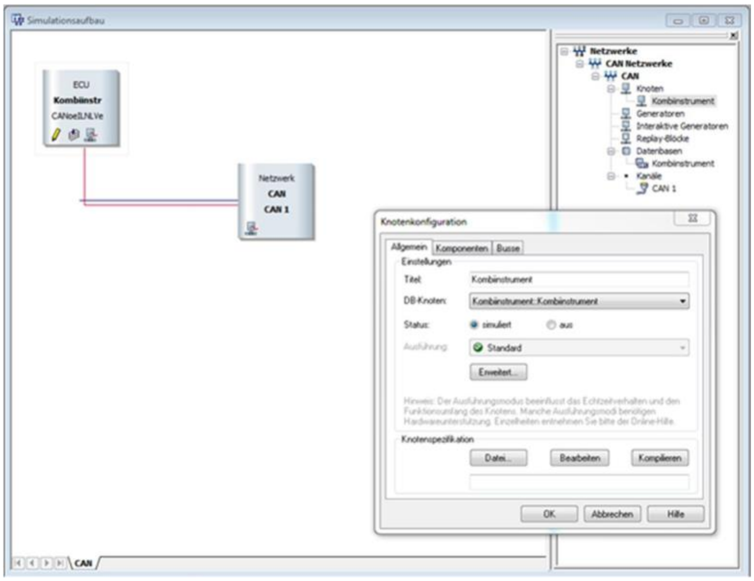
- Afterwards, create a new network node called “*Kombiinstrument*” that represents a simulated control unit for transmitting the messages. Assign the message “*Klemmenstatus*” to this new network node in the tab *Tx messages* in the network node settings.

- Save the database.

- Now, create a new node in your CANoe *simulation setup*, in the column on the right hand: *Networks* -> *CAN networks* -> *CAN* (right-hand clicking on the node -> add network node) and, in the corresponding settings, select the *CANdb++ node* you just created (see figure 2.3). By right-hand clicking on the network node you just created, you can open the node configuration.

- Upload your database.

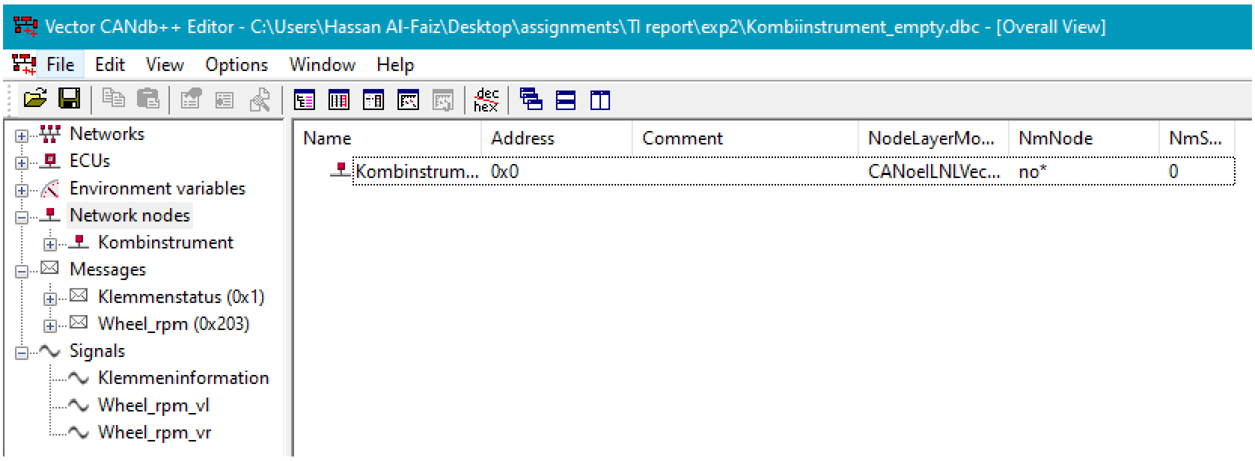
a) What can you observe in the trace window (in the video)?



2figure 2.3: assigning the database node to the simulated node

- Now, connect the signal “*Klemmeninformation*” to a switch element in a new CANoe panel that you create on this purpose.
  Assign the values seen above for the states *IGN_OFF and IGN_ON* to this switch.

- Upload your modified database so that we can test it. The instrument cluster should wake up from sleep mode when the switch is activated, as seen in the video.



2.5.2 Displaying the vehicle's speed

For displaying the vehicle's speed on the instrument cluster, the speed pulses of at least two wheels are needed.

- In the manufacturer's communication matrix, you can find the following information:

Raddrehzahlen

Botschaftsname:	Wheel_rpm
Identifier:	0x203h
Byteanzahl:	8
Zyklus:	20 ms
Signalname:	Wheel_rpm_vl
Byte:	0
Bit:	0 bis 5
Byte:	1
Bit:	0 bis 7
Signalname:	Wheel_rpm_vr
Byte:	2
Bit:	0 bis 5
Byte:	3
Bit:	0 bis 7

Pay attention: all the messages are sent in Motorola format.

- Create the signals in the CANdb++ editor and position them at the correct place in the corresponding message.

- The behaviour of the message when transmitting has to be adapted, in the tab *Attributes* in the message settings, according to the information of the communication matrix (*GenMsgCycleTime*, *GenMsgSendType*).

- Assign the message "*Wheel_rpm*" to the network node already created, in the tab *Tx messages* in the network node settings.

2.5.3 Extending the panel

Extend the panel by two track bars for the speed pulse signals of the two wheels. The value range should be between 0 and 4000. Upload your panel. Also, make sure that all the data you previously uploaded is complete.