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?

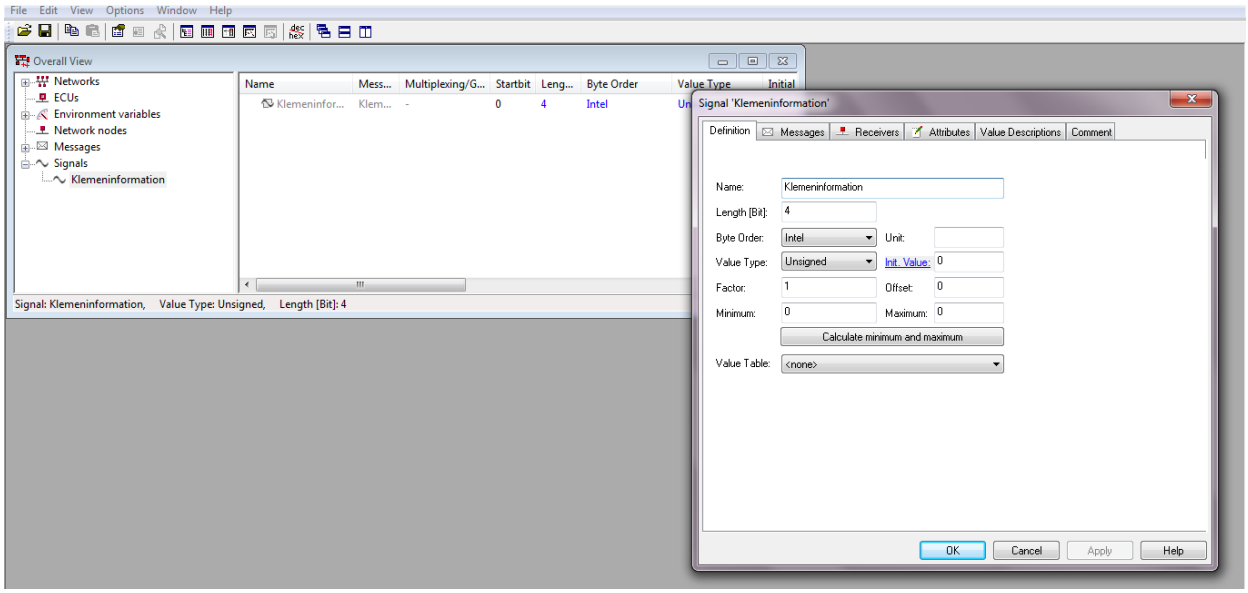
= No.

b) Do you need a terminating resistor for the instrument cluster? If yes, why?

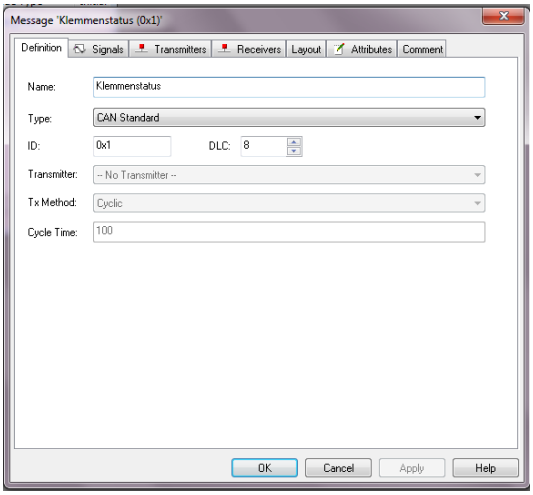
= Yes. A Terminating Resistor is a Signal Quality component. It is used to soak up an AC signal, preventing reflections or ghosts on the line. It takes time for signals to travel along wire, and the energy of the signal cannot be created or destroyed; it has to go somewhere.

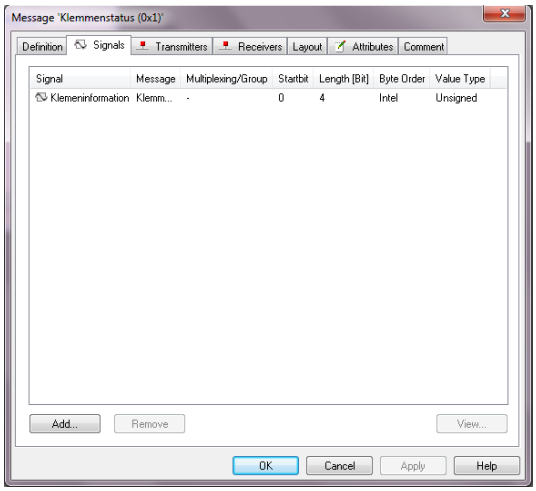
Now, for waking up the instrument cluster from sleep mode, the state of the ignition has to be transmitted by a CAN message. The corresponding signals and transmission properties have to be defined by the means of a database.

- Therefore, open the file *Kombiinstrument_empty.dbc* (available online) in the CANdb++ editor.

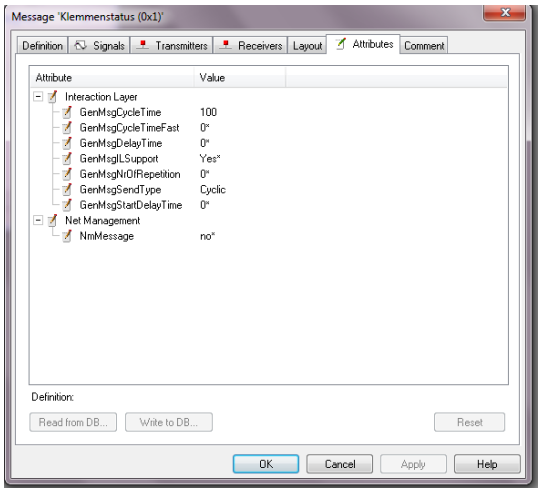


- Afterwards, create a signal called "*Klemmeninformation*" and position it at the correct place in the corresponding message called "*Klemmenstatus*".





- 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

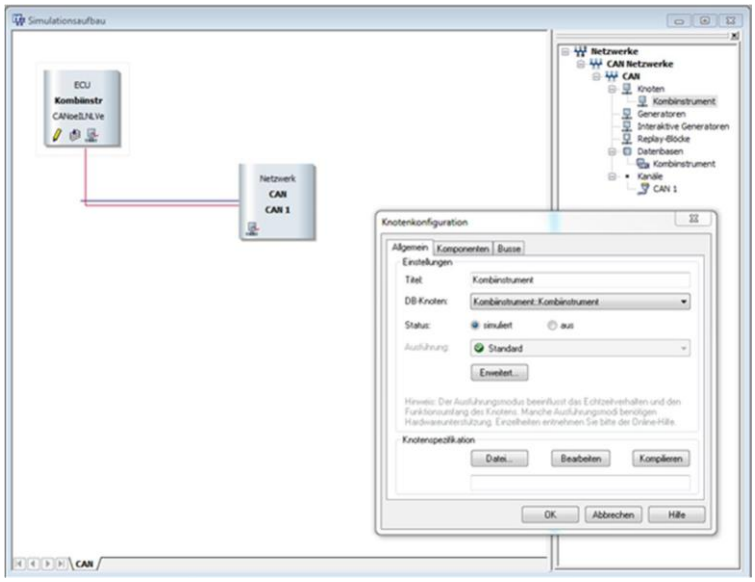
- Afterwards, create a new network node called "*Kombiinstrument*" that represents a simulated control unit for transmitting the messages. Assign the message "*Klemmenstatus*" to this new network node in the tab *Tx messages* in the network node settings.

- 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.

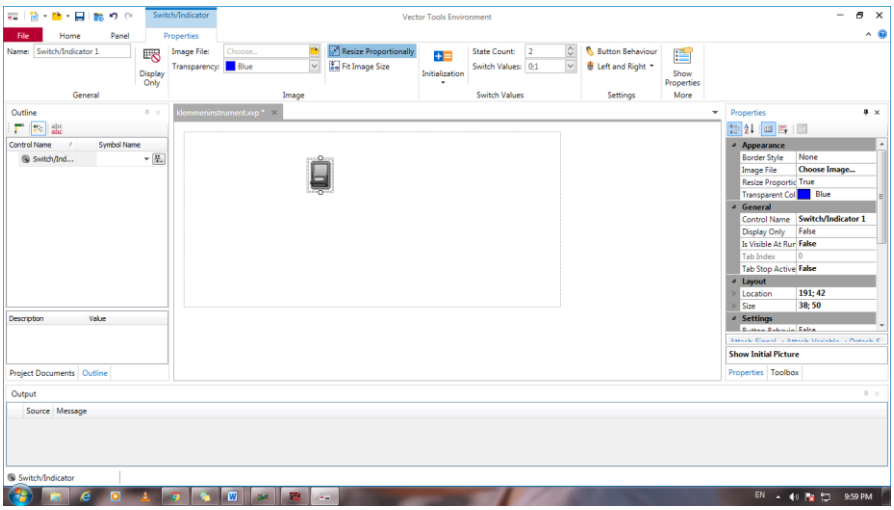
- Upload your database.

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



1figure 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

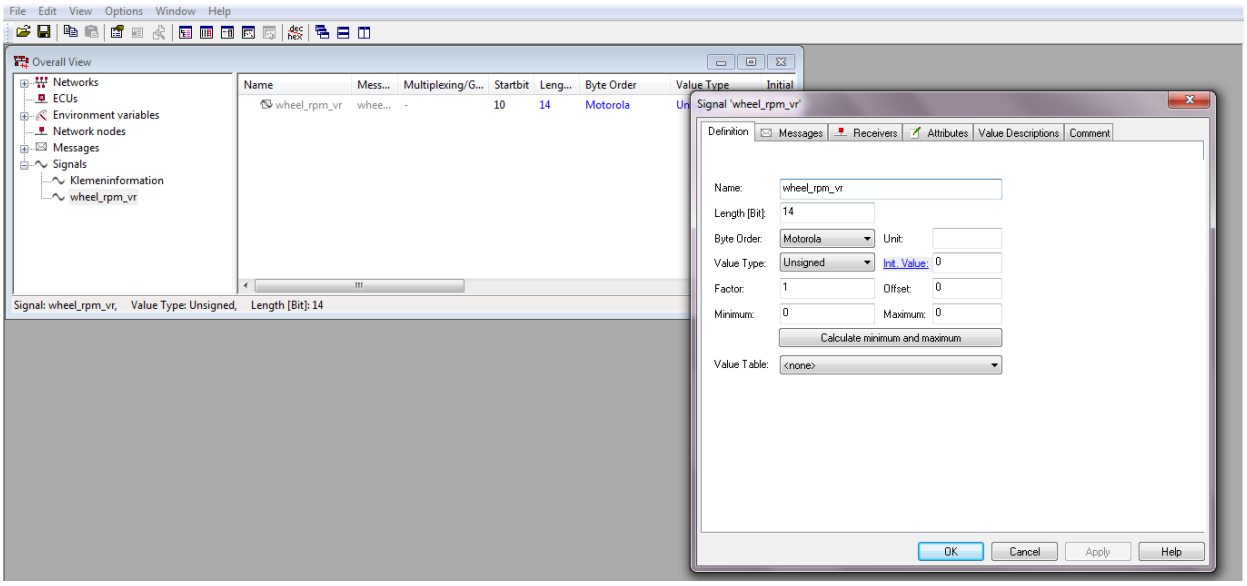
For displaying the vehicle's speed on the instrument cluster, the speed pulses of at least two wheels are needed.

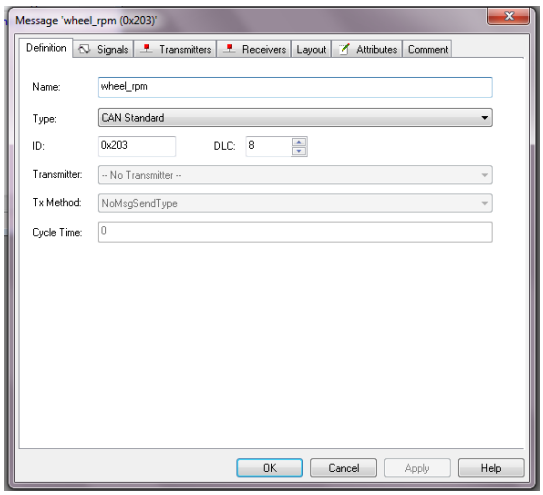
- In the producer's communication matrix, you can find the following information:

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

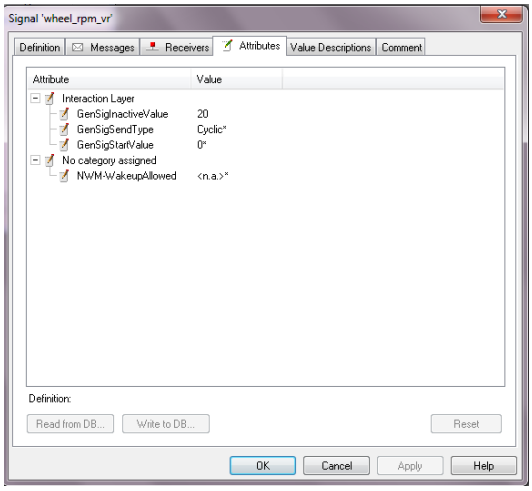
Pay attention: all the messages are sent in Motorola format.

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

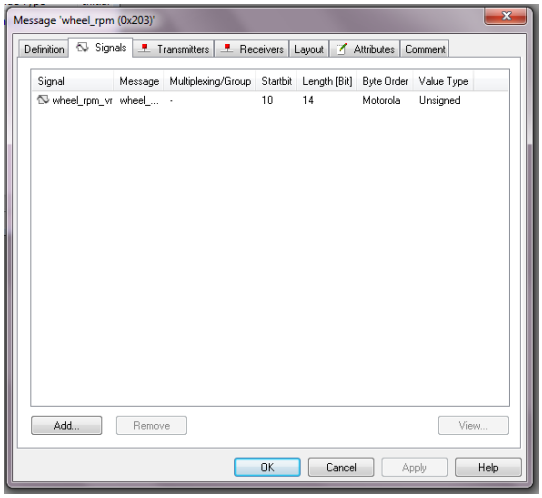




- The behaviour of the message when transmitting has to be adapted, in the tab *Attributes* in the message settings, according to the information of the communication matrix (*GenMsgCycleTime*, *GenMsgSendType*).



- 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.