Exercise 3 – Communication

3

Now the project is extended by adding CAN communication.

The information about the rear doors and the rear light is exchanged via CAN messages.

RearECU informs MyECU about the state of the rear doors via CAN message.
Based on status of all doors (front and rear) MyECU decides about the resulting state of front and rear light.

Exercise steps

1. Create Ports for Communication
2. Data Mapping
3. BSW configuration for Communication
4. Generate program Runnable and test

Exercise 3 – Communication
Part 1/4: Create Ports for Communication

Open DaVinci Developer, **"Configure SWCs"** using file
_E3_Communications\MyECU.dpa

Port Interfaces **PiDoorState** and **PiLightState** have been used for Intra-ECU communication, now we use them for Inter-ECU communication (over the CAN bus)

Create Receiver Port Prototypes on CtApMySwc for receiving the rear door state:

- **PpDoorStateRearLeft,** port interface: **PiDoorState**, Init value: reference to **CDoorClosed**
- **PpDoorStateRearRight,** port interface: **PiDoorState**, Init value: reference to **CDoorClosed**

Create Sender Port Prototypes on **CtApMySwc** for sending the rear light state

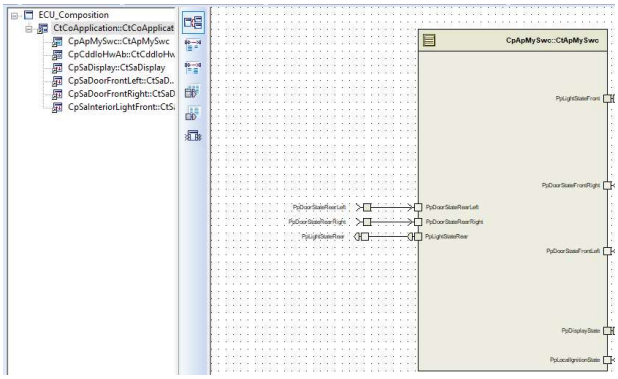
- **PpLightStateRear**, port interface: **PiLightState,** Init value: reference to **CLightOff**

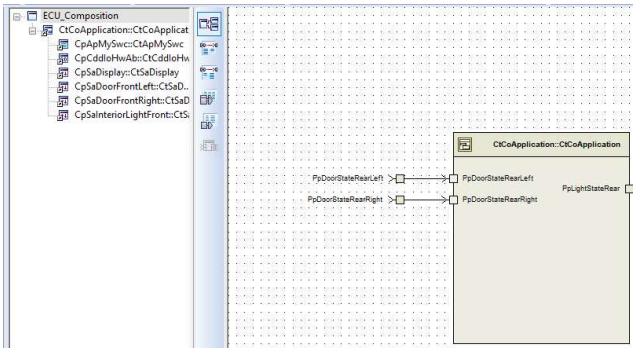
There is a CAN receive message that contains three signals:
- state of rear left door
- state of rear right door
- state of the rear interior light (not used in the exercise)

CAN transmit message contains state of the rear interior light.

Data Mapping is coming later.

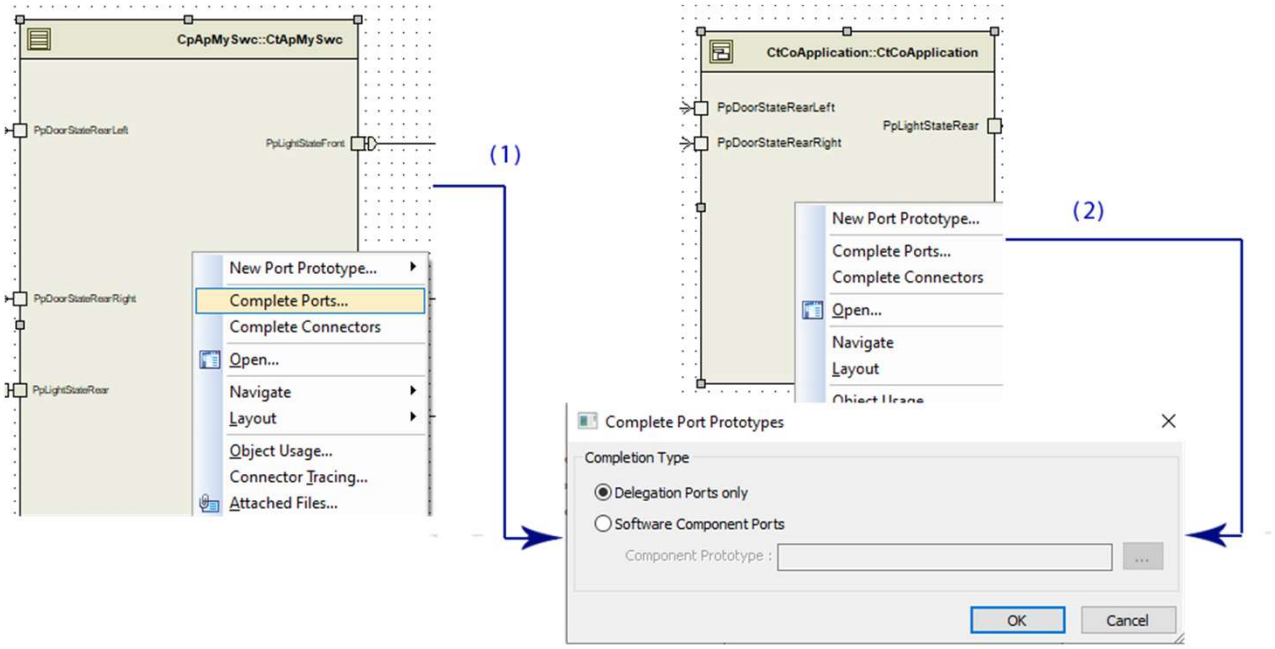
Exercise 3 – Communication
Part 1/4: Create Ports for Communication





Create Delegation Ports at the level inside **CpCoApplication** and a second time at the level **ECU_Composition** (this is the part for data mapping).

Exercise 3 – Communication
Part 1/4: Create Ports for Communication

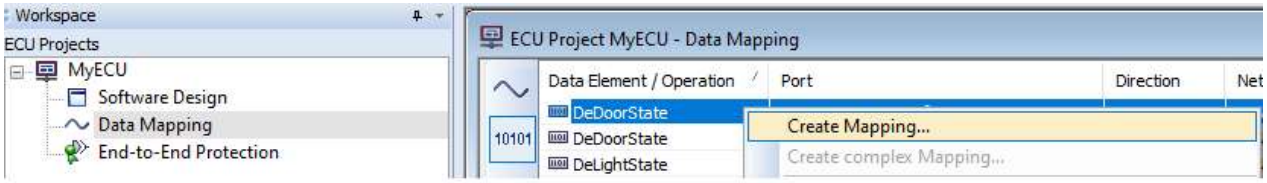


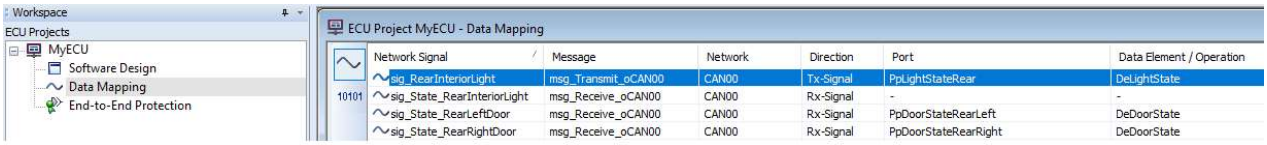
How to create the Delegation Ports (see image below):

1. right-click on the Component and select "Complete Ports"
2. select **"Delegation ports only"**

This will create the Delegation Ports for the outer interface of the ECU Composition.

Exercise 3 – Communication
Part 2/4: Data Mapping





Perform the data mapping, right-click on each data element, select Create Mapping and choose the respective network signal until all data elements have a data mapping.

In the signal view, not all signals need to be mapped to a data element, we do not need a mapping for Rx signal **sig_State_RearInteriorLight.**

Add Port Access for following Runnable:

CtApMySwc. RCtApMySwcCode

- **implicit read** for **PpDoorStateRearLeft.DeDoorState**
- **explicit read by argument** for **PpDoorStateRearRight.DeDoorState**
- **explicit write** for **PpLightStateRear.DeLightState**

Add an On Data Reception Trigger for the rear door data elements

Check Workspace, save and close DaVinci Developer

Exercise 3 – Communication
Part 3/4: Watch BSW configuration for Communication

Open DaVinci Configurator, **"Configure BSW"** using file _E3_Communication\MyECU.dpa

Execute the auto-solving action for the
RTE59000 (Synchronize the System Description)

If Configurator was left open, you can choose “Synchronize Configuration now" from a pop-up dialog.

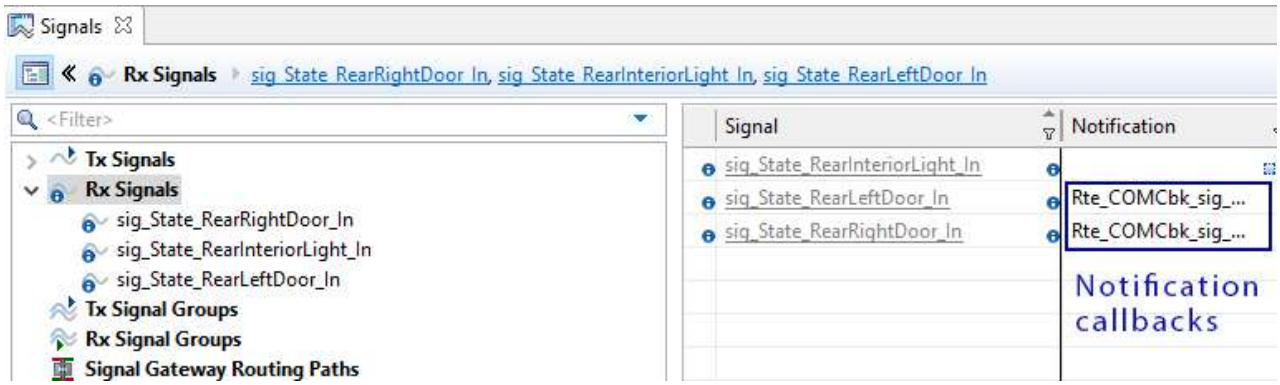
New Data Receive trigger for rear doors need to be mapped to a task. DaVinci Configurator already suggests to map them to **My_Task.** Choose the appropriate solving actions.

Validate and Generate BSW

Press F7 in order to update the SWC templates

Note: By executing the auto-solving action, the task position for the new Data Receive trigger is automatically set to the value "0".

Exercise 3 – Communication
Part 3/4: Watch BSW configuration for Communication



As a consequence, from configured Data Receive Events at SWC R-Ports in the RTE, the Configurator activates the corresponding COM callbacks to trigger these events. The RTE has to provide the implementation of the corresponding callbacks.

For this exercise, no further actions are necessary in Configurator.

Become familiar with the default settings and other possible settings in Configurator. Look for settings of transmission modes of the I-PDUs…

Look at the **sig_State_Rear*** Rx Signals
- Notification function has been added by RTE Validator in Configurator because a "Receive data...“ trigger has been set.
- Search also for the generated **Rte_COMCbk** implementations in **Rte.c.**

Close Configurator.

Exercise 3 – Communication
Part 4/4: Generation, program Runnables and test

Open Visual Studio Project by double-clicking on file
_E3_Communication\Solution\MyECU.sln

Extend **CtApMySwc** functionality

- Read the state of rear doors in CtApMySwc.c
- If any door is open, turn ON front and rear interior light, else turn them OFF

Control algorithm already exists, you only have to extend the 'if' condition and add the rear doors and light handling.

Build ECU code.

Start CANoe, switch back to Visual Studio, then press F5.

Switch back to CANoe, then press F9.

Test your code.