

1. PROJECT

ROOT_OBJECTS

--|examples_AADL/Common\Bus_Properties|--,
--|examples_AADL\EcoSolarTypes|--,
--|examples_AADL\EcoSolar|--

END

1.1. Project Description

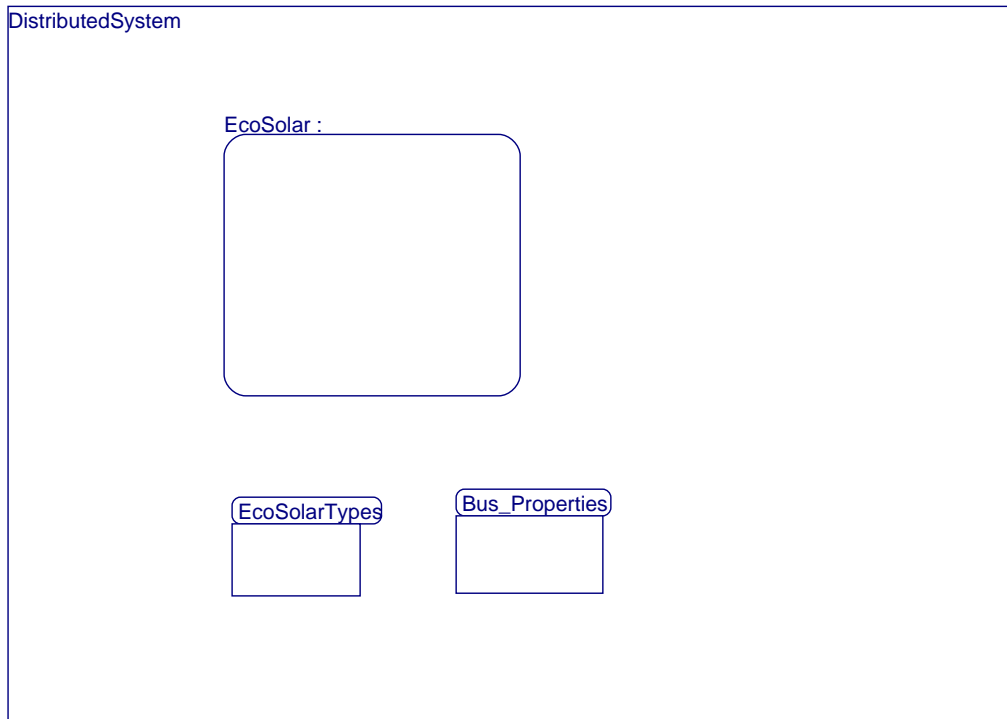
DISTRIBUTED REAL-TIME SYSTEM

This project describes the simplified architecture of the electronics and software of a solar vehicle. Several Electronic Control Units are connected by a CAN bus.

Each node on the CAN network is composed of a processor, a process, and a set of threads. Inter-nodes connections generate messages that are multiplexed on the bus.

The simulation shows how the local scheduling of the sending and receiving threads on each node and the scheduling of the messages on the bus interact together.

1.2. AADL Diagram



2. SYSTEM EcoSolar IS

2.1. DESCRIPTION

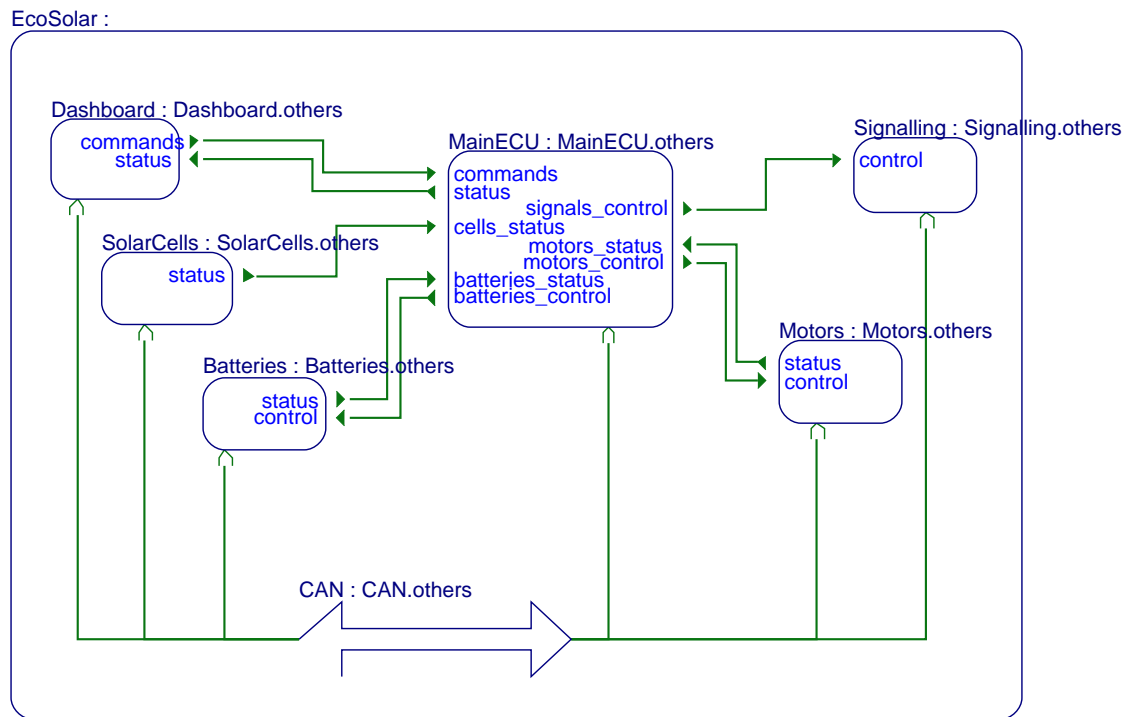
2.1.1. PROBLEM

2.1.1.1. Statement of the Problem (text)

This distributed system is composed of six ECUs connected by a CAN bus.

- MainECU: implements the centralized control command of the vehicule.
- Dashboard: represents the local controler within the car dashboard.
- Motors: describes the electrical engines control subsystem.
- SolarCells: solar cell regulation software (not implemented).
- Batteries: batteries load and voltage control (not implemented).
- Signalling: control software for lights and other equipments (not implemented).

2.1.1.2. AADL Diagram



3. SYSTEM Dashboard IS

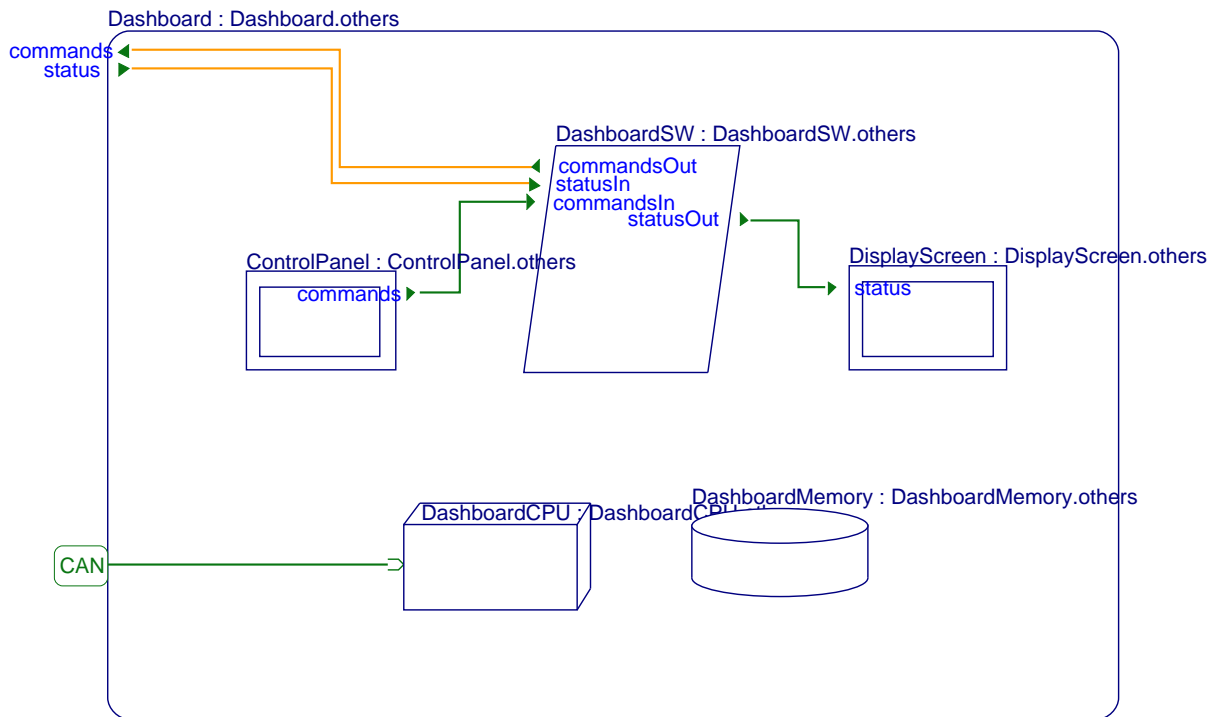
3.1. DESCRIPTION

3.1.1. PROBLEM

3.1.1.1. Statement of the Problem (text)

The Dashboard subsystem is composed of a single software executable running on a dedicated microcontroller and associated memory. Input controls and output displays of the dashboard are represented by two AADL devices.

3.1.1.2. AADL Diagram



4. PROCESS DashboardSW IS

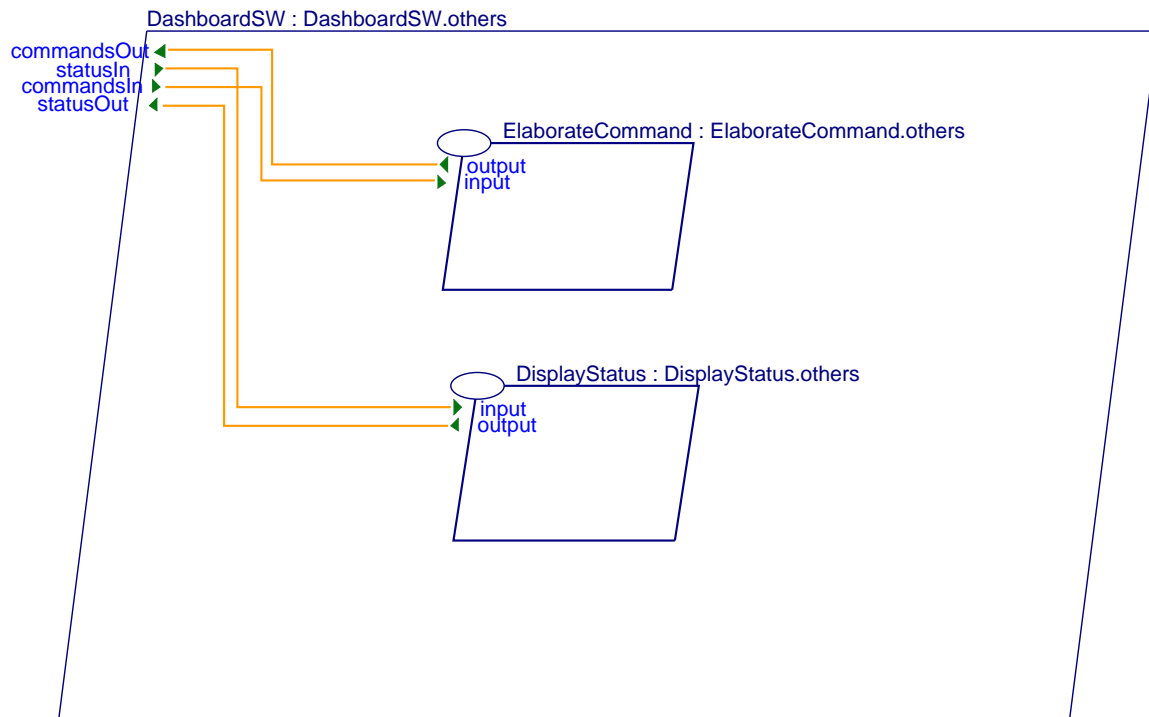
4.1. DESCRIPTION

4.1.1. PROBLEM

4.1.1.1. Statement of the Problem (text)

The dashboard software is composed of two threads that act as device drivers. The ElaborateCommands thread converts data coming from the control panel into appropriate commands for the main ECU. The DisplayStatus thread receives status data from the main ECU and convert it into a proper format to be sent to the display screen.

4.1.1.2. AADL Diagram



5. SYSTEM Motors IS

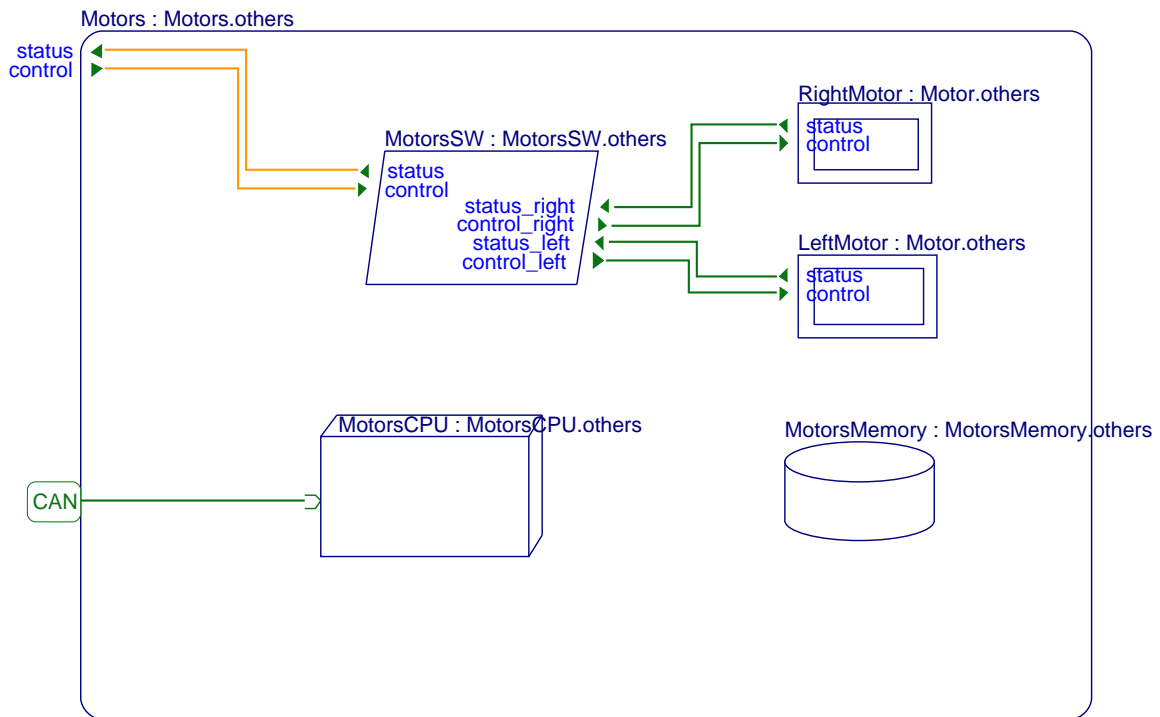
5.1. DESCRIPTION

5.1.1. PROBLEM

5.1.1.1. Statement of the Problem (text)

The Motors subsystem contains the motor control software and its execution platform. The MotorSW interacts with the motor controllers represented by two AADL devices, one for the right rear wheel and one for the left rear wheel.

5.1.1.2. AADL Diagram



6. PROCESS MotorsSW IS

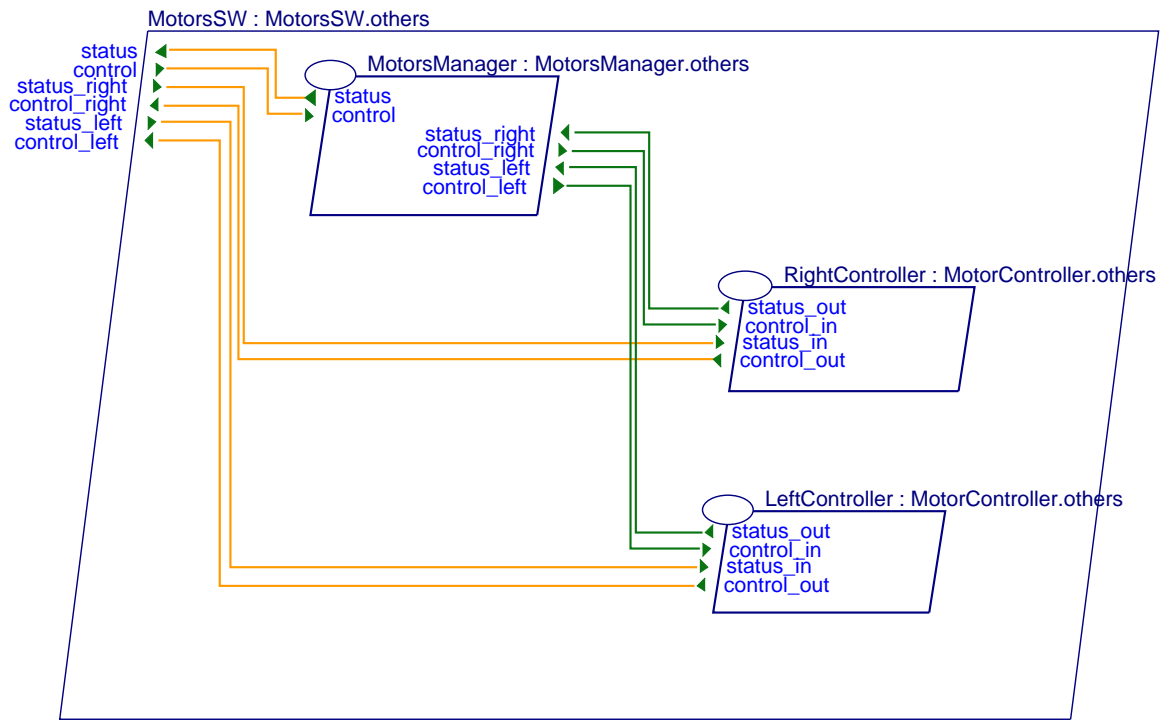
6.1. DESCRIPTION

6.1.1. PROBLEM

6.1.1.1. Statement of the Problem (text)

The MotorSW application is implemented by three threads. The MotorsManager thread is in charge of receiving commands from the main ECU and to dispatch them to the right and left motor controller software drivers (RightController and LeftController threads).

6.1.1.2. AADL Diagram



7. SYSTEM MainECU IS

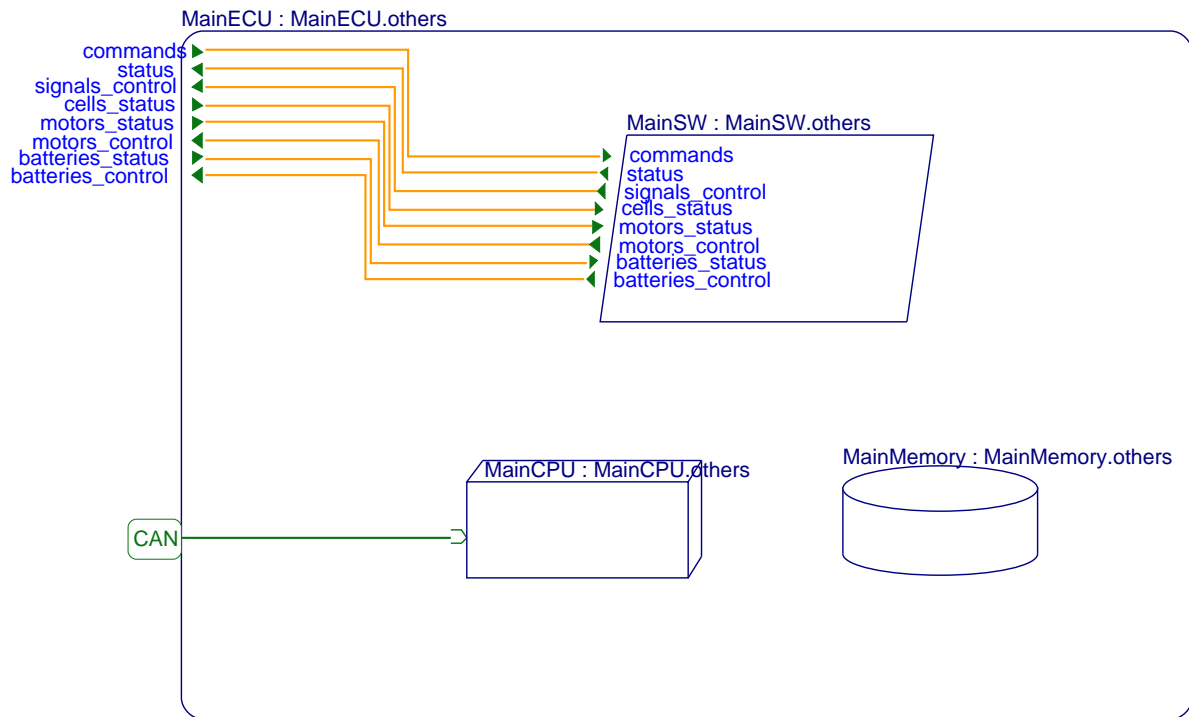
7.1. DESCRIPTION

7.1.1. PROBLEM

7.1.1.1. Statement of the Problem (text)

The MainECU subsystem consists of a single executable (AADL process) running on a single core microcontroller (AADL processor) and located on a volatile on the main board memory (AADL memory).

7.1.1.2. AADL Diagram



8. PROCESS MainSW IS

8.1. DESCRIPTION

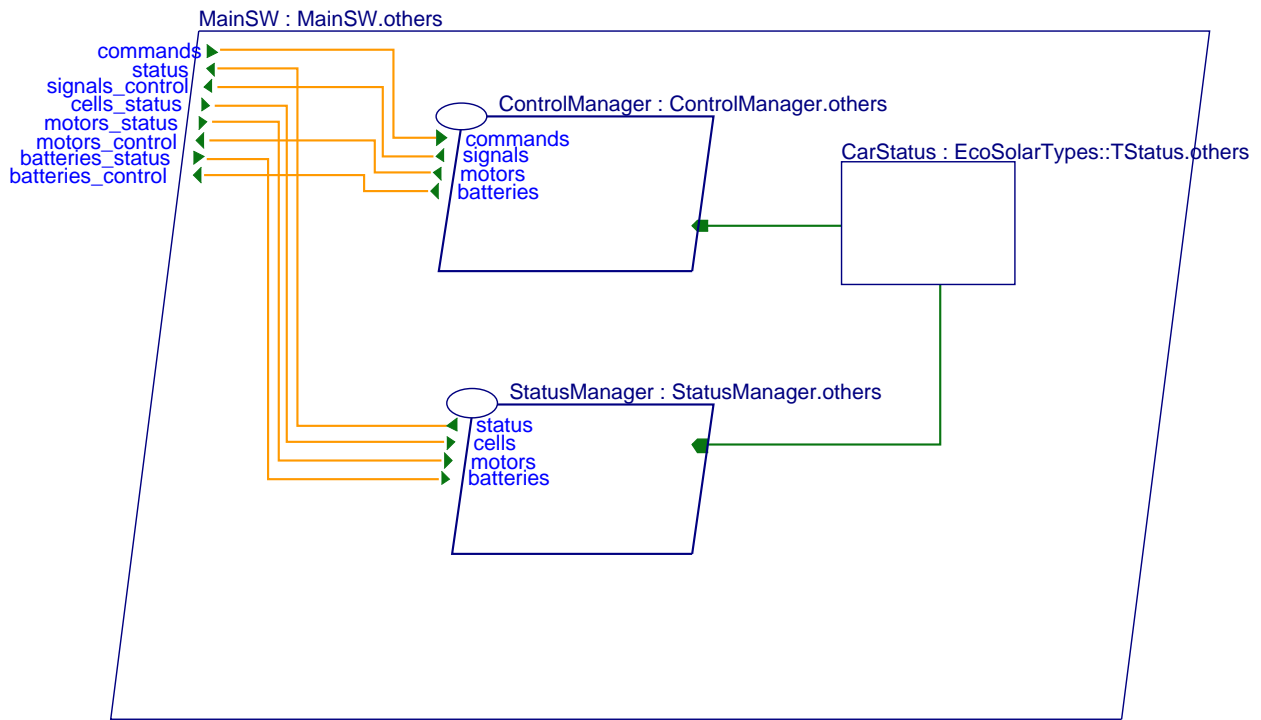
8.1.1. PROBLEM

8.1.1.1. Statement of the Problem (text)

The main software is implemented by two periodic threads that share the current vehicle status:

- ControlManager: elaborates and sends commands to the other ECUs according to the current vehicle status.
- StatusManager: gets feedback from the other ECUs to update the vehicle status accordingly.

8.1.1.2. AADL Diagram



9. BUS CAN IS

9.1. DESCRIPTION

9.1.1. PROBLEM

9.1.1.1. Statement of the Problem (text)

This component is an abstraction of a physical CAN network on which the logical connections between the ECUs are bound. It is implemented as a non preemptive highest priority first bus message scheduler.

Bus messages are produced by the out data port of each bound connection and received by the corresponding remote in data port.