CAN bus status on SPACEBUS NEO Telecom satellite

THALES ALENIA SPACE

STEPHANE CAEL – DATA HANDLING SYSTEM

CAN bus at the heart of the avionics architecture:

- **Architecture of CAN bus in THALES ALENIA SPACE design.**
- **Validation strategy of the CAN bus.**
- **CAN validation results:**
  - Electrical tests
  - Functional tests

Introduction of CAN bus in Telecom S/C

Current SB4000 telecom avionics architecture makes extensive use of digital data buses which have shown their flight ability and high performance.

- OBDH RS485 digital data bus that is THALES ALENIA SPACE specific adapted from ESA OBDH standard
- MIL-std-1553B data bus.

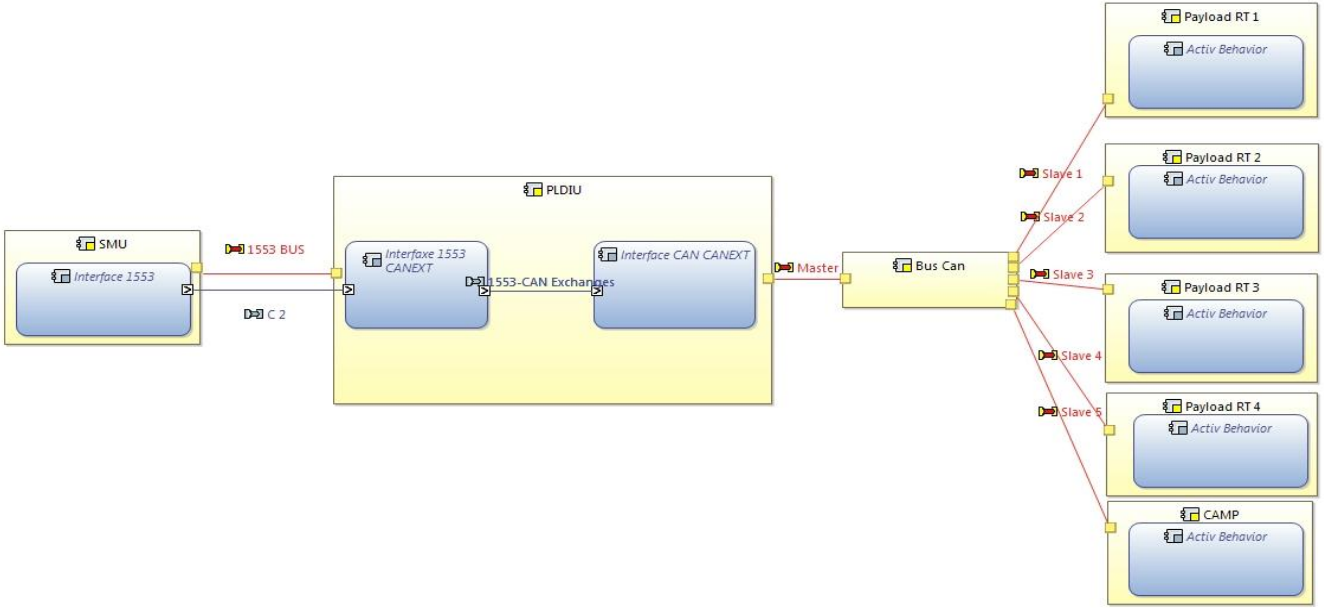
Main drivers for introducing the CAN bus in Telecom spacecraft:

- Widely used in industrial world: many components, IPs and test tools available. Therefore, brings significant cost reduction.
- Performance : data rate, length… well suited to spacecraft bus requirements.
- Communication protocol more advanced than OBDH, but light enough to be implemented into a programmable component, ASIC or FPGA.
- Multi master capability offers interesting opportunities for Payload ground testing.

  ➔ direct connection of a ground test bench to the bus (as a secondary bus master) allows to ease and accelerate the Payload tests while keeping the telemetry through the nominal bus master.
- Ability to plug higher number of remote terminals per bus.
- Use of standard twisted pair offering significant cost reduction and easier integration on satellite.

Telecom S/C Avionics Architecture using CAN

- CAN bus at the heart of avionics architecture on SPACEBUS NEO and other platforms.
- Overall higher CAN protocol is based on Master-slaves architecture.
- Compliant to common NEOSAT Applicable Document AD919 issue03.



THALES ALENIA SPACE INTERNAL

CAN development plan

Main steps of the CAN development:

- **Development and qualification of THALES ALENIA SPACE units:**
  - Payload units embedding ASIC MEGA.
  - PLDIU unit with the DPC embedding CAN IP.

- **Analyses and trade off on the CAN architecture:**
  - CAN transceivers selection (ISO transceivers Rad hard models supplied in 3.3V).
  - Application protocol (minimum subset of CAN Open objects)
  - Topology, performance and data rate needs.
  - Robustness and failure tolerance (CAN bus redundancy).

- **Validation of the CAN architecture based on mock-up and modelling.**

CAN validation objectives:

- **Early verification of the optimum functioning point of the Physical layer:**
  - number of users,
  - harness definition,
  - Baud-rate,
  - CAN protocol configuration (sampling point location, Tf/Tr…).

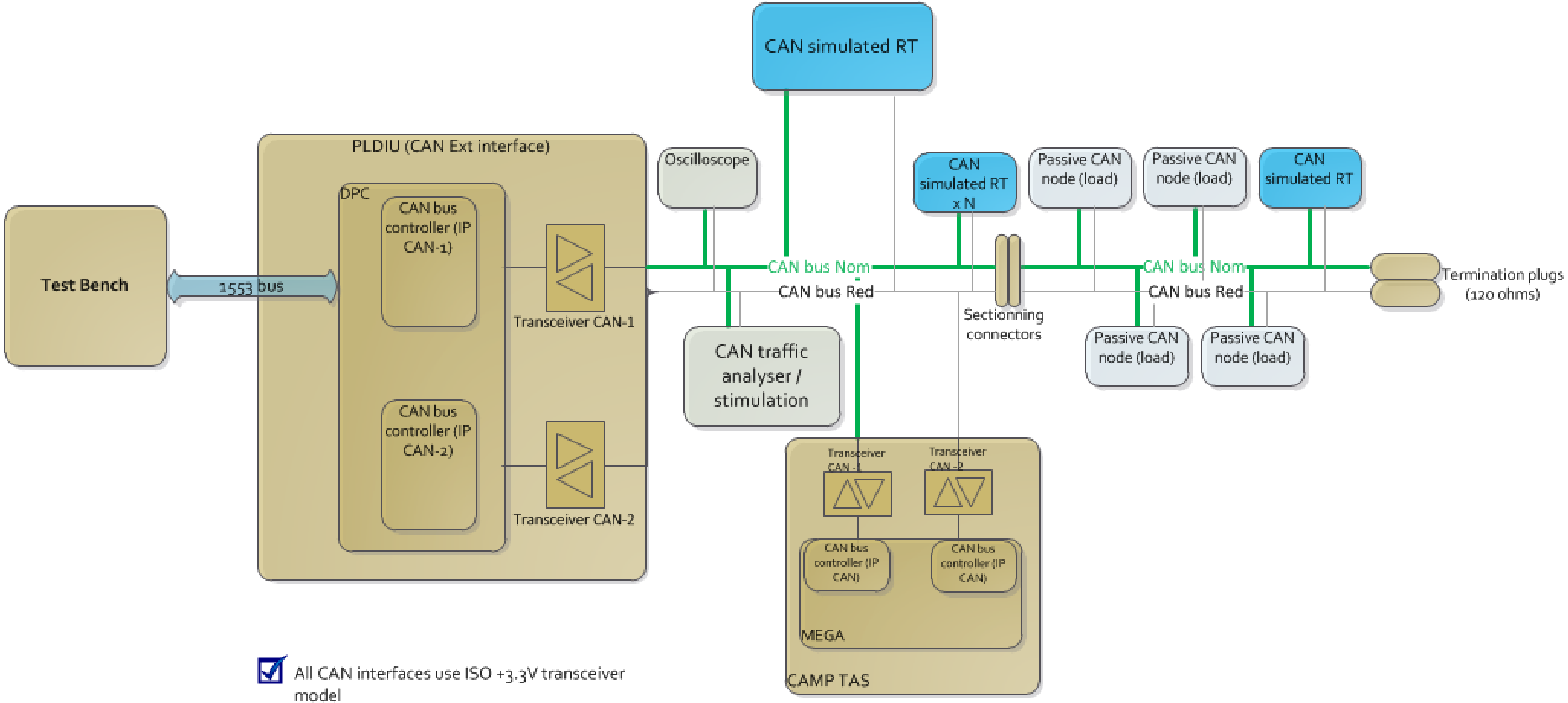
- **Confirm bus performance in real environment:**
  - Integration of real and simulated hardware (PLDIU & PL units) on Avionics test bench.
  - CAN harness configured in worst case bus topology,
  - Real time test with bus control SW,
  - Error messages rate monitoring,
  - Error simulation & management.

CAN bus validation campaign is based on the following:

- **A CAN mock-up embedding real hardware in the loop with:**
  - « Flight-like » models of PLDIU and Payload RT,
  - Simulated active subscribers,
  - Use of testing tools for frames recording, analysis and stimulation,
  - Use of passive nodes simulating the real impedance on the CAN bus,
  - Use of CAN harness flight representative in terms of length, physical characteristics, topology, connecting…

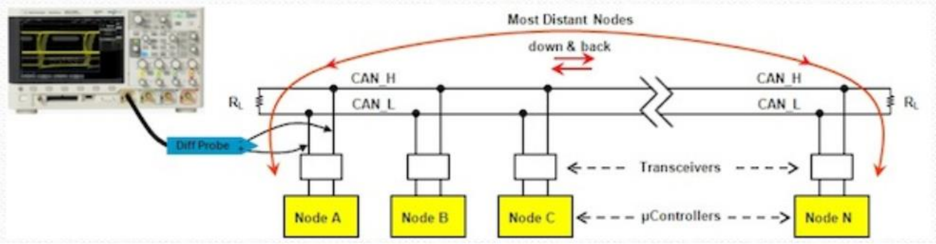
- **CAN bus simulation allowing to quickly assess the bus margins in different mission configurations (number of units, harness length and topologies).**
  - Allow to adapt to different bus configurations without re-qualifying the CAN bus.

CAN development and validation – Mock-up

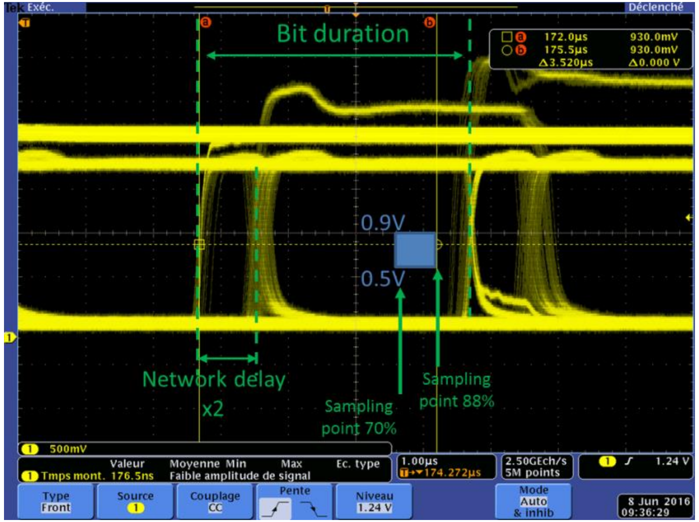


- ☑ All CAN interfaces use ISO +3.3V transceiver model
- ☑ CAN bus controller (IP CAST).

Electrical tests successfully conducted:



- **On a representative mock-up:**
  - Harness in a worst case flight configuration.
  - Rad-Hard flight representative transceivers power supplied in +3,3V.
  - 80 nodes (68 passives + 12 actives).
  - PLDIU unit and CAMP unit.



- **Success criteria:**
  - Electrical characteristics of CAN signals (differential and common voltage levels, timing values, …),
  - Error frames monitoring,
  - Telemetry analysis on CAN network.

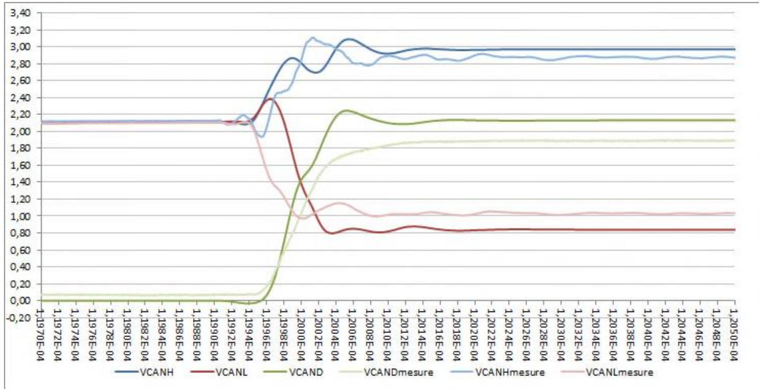
CAN Electrical tests.

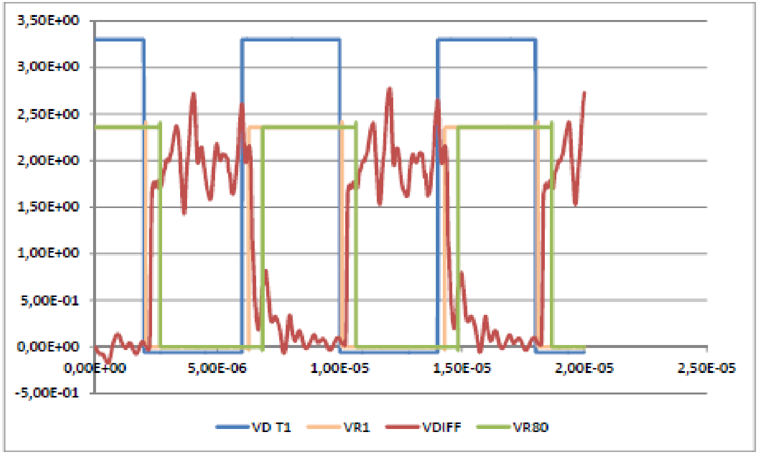
- **Objective to characterize the electrical performances of the CAN bus in several configurations.**
- **Several test cases:**
  - CAN bus state at <u>**power ON transition**</u>.
  - Variability of <u>**Bus termination**</u> impedance (120 to 500 ohms) ➔ de-adaptation of CAN bus.
  - Variability on the <u>**number of active RT**</u>.
  - Variability on the <u>**harness length**</u> and RT location <u>**density**</u>.
  - Variability on the <u>**sampling point**</u> location (from 70% to 88%).
  - Variability on the <u>**slope**</u> control configuration (Rs).
  - Different electrical <u>**failure cases**</u> (CC to GND, CC to +5V, line interruption, disconnection).

CAN Test campaign results

Simulation at physical layer.

- **Time correlation between simulation and real measures.**
- **Test cases:**
  - Up to 80 nodes connected,
  - Different bus terminations (standard and splitted),
  - Number of powered RT,
  - Different Rs.
- **Test measures:**
  - Rise time / fall time
  - Total propagation time
- **EMC simulation (CS test):**
  - Injection of differential noise from 10kHz to 30MHz.





THALES ALENIA SPACE FRANCE

CAN Functional tests successfully completed:

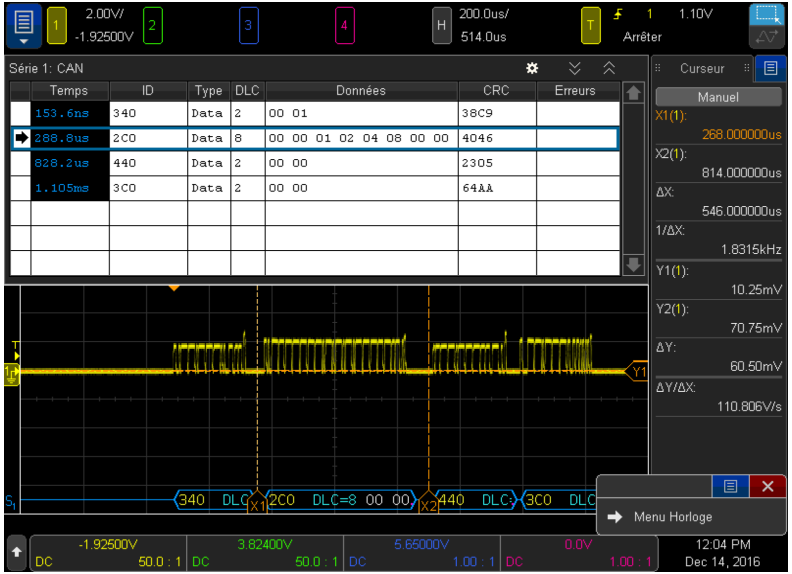
- **Consolidate the preliminary results on the flight representative mock-up integrating ISO transceivers supplied in +3.3V and flight representative hardware.**
  - By measuring the quality of signal at any point of the network and the quality of service.
  - By validating the CAN exchange protocol with **real data traffic**, and injecting **faults**.

- **Integrating the mock-up on the Avionics Test Bench and check the CAN bus functional behavior in its complete HW and SW environment.**

CAN Test campaign

Functional Tests:

- **Objective to characterize the functional behaviour in __nominal__ and __degraded__ cases over the CAN bus.**
- **Several test cases:**
  - __Nominal case__ of acquisitions and commands over CAN bus.
  - Test with different __number of active nodes__ (allowing to simulate different mission life phases).
  - Bus error test case 1 ➔ __no response__ of RT node.
  - Bus error test case 2 ➔ __bad configuration__ of one RT (sampling point, data rate…).
  - Redundancy test ➔ Can __bus reconfiguration__ and automatic bus selection.



Conclusion and way forward:

- Positive CAN tests results at physical and data link layers.
- Step forward ➔Integration of the Applicable Document AD919 in ECSS.
- Further FDIR management:
  - Improve CAN failure monitoring (identification of the failed unit).
  - Improve CAN failure auto isolation.