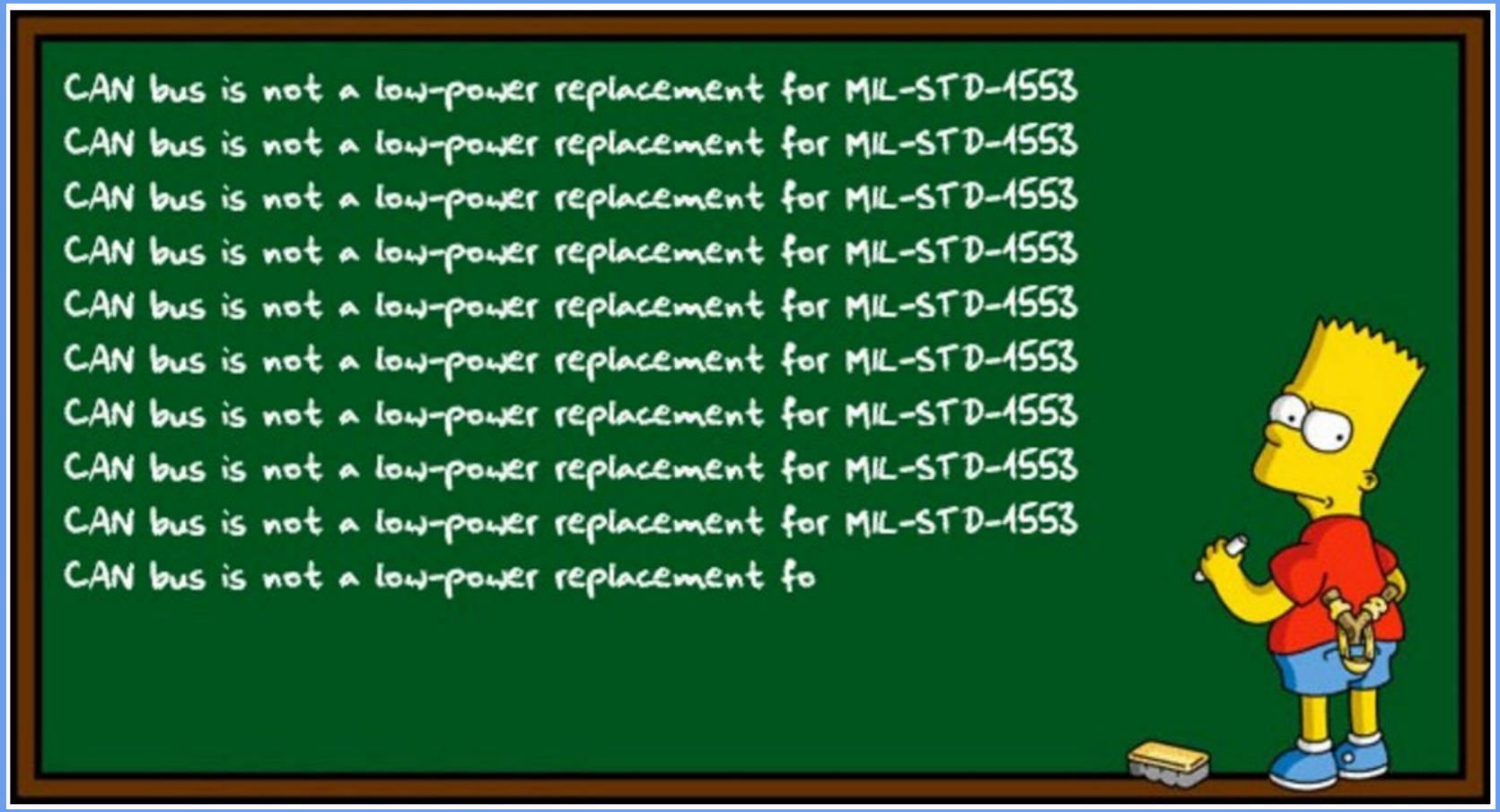
CAN in Space

A small to-do list for the future

Industry is always right

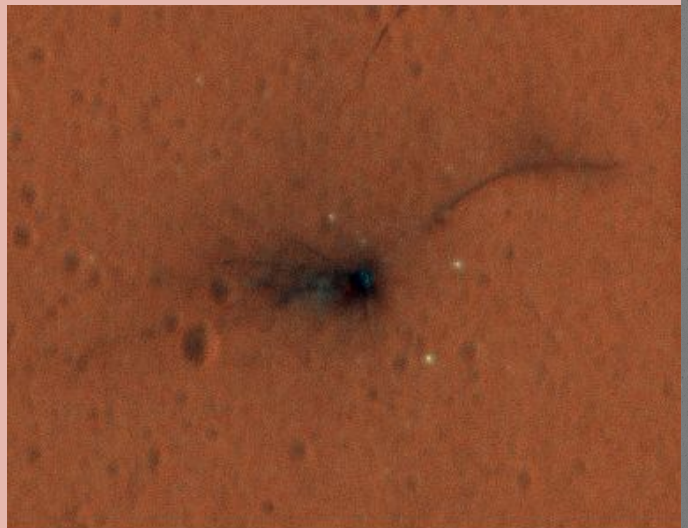


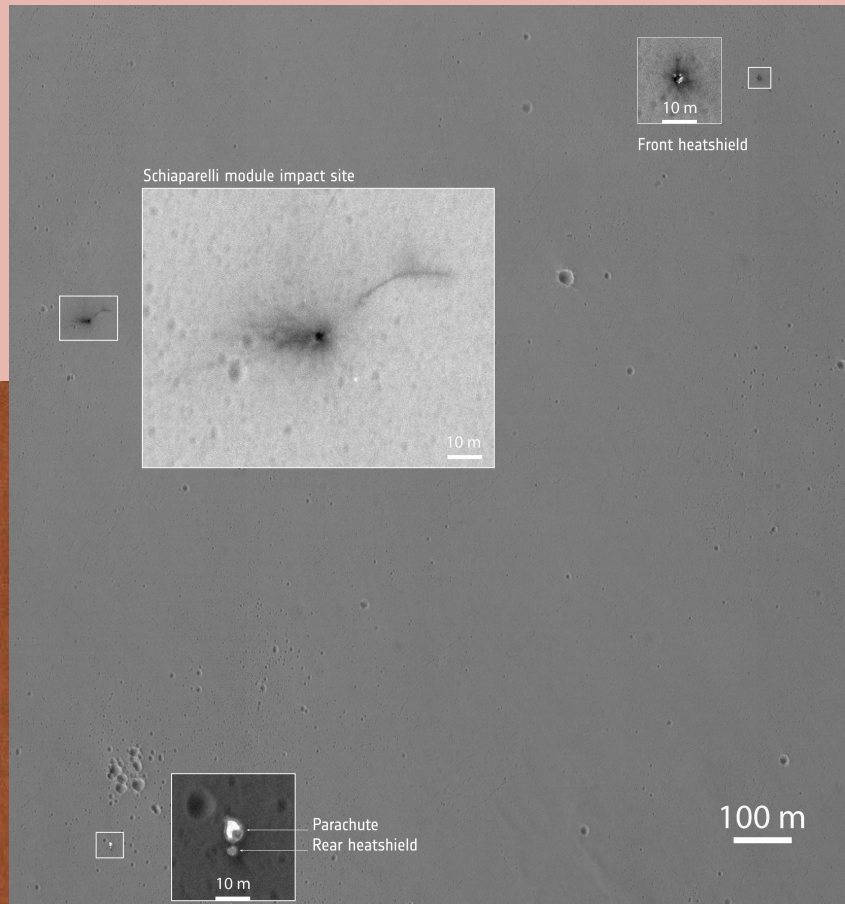
A robust, reliable and capable communication bus

CAN bus on Exomars 2016 EDM worked without issues until the last fateful moment.

The uninterrupted stream of telemetry during the descent phase allowed to understand the root causes of the failure.

The words ‘CAN bus’ and ‘computer’ are not even mentioned in the failure investigation report.





Building Blocks for CAN

ECSS-E-ST-50-15C - CANbus extension protocol (1 May 2015)

Posted on 2015-05-29 by Klaus Ehrlich (as webmaster)

CREATE CR

Scope

This standard is applicable to spacecraft projects that opt to use the CAN Network for spacecraft on-board communications and control. It also defines the optional use of the CANopen standard as an application layer protocol operating in conjunction with the CAN Network data link layer.
This standard does not modify the basic CAN Network specification and complies with ISO 11898-1/-2:2003. This standard does define protocol extensions needed to meet spacecraft specific requirements.
This standard covers the vast majority of the on-board data bus requirements for a broad range of different mission types. However, there can be some cases where a mission has particularly constraining requirements that are not fully in line with those specified in this standard. In those cases this standard is still applicable as the basis for the use of CAN Network, especially for physical layer and redundancy management.
This standard may be tailored for the specific characteristic and constrains of a space project in conformance with ECSS-S-ST-00.

Attachments:

- ECSS-E-ST-50-15C(1May2015).pdf
- ECSS-E-ST-50-15C(1May2015).pdf

Md5 checksum .doc file = CC4EFCF915D1AA0CF3E3C971875812BD
Md5 checksum .pdf file = 8B8FAF8BC8069A603321213D687E464A

Posted in Active Standards, Standard Tagged Active Standard, Engineering, published, Standard permalink

Are we ready for distributed intelligence ?

FPGA vs. SW cost:

In many companies there are FPGA development teams with 100s of years of combined experience: very efficient & reliable development methods.

For uC ‘firmware’, needs to be booted up.

Moreover (send a thank you note to ESA), documentation volume is about 2.5 times heavier for software and does not take into account reuse in an efficient way.

Some process improvement efforts focused on automating regression tests & automatic document generation are needed to get back to equal development efforts vs VHDL in FPGA.

Are we ready for distributed intelligence ?

There is a necessity to find some metrics to compare use of uC with respect to FPGAs in avionic systems.

Nowadays it is cheaper (in terms of total cost of ownership) to write 10k lines of VHDL than 1k lines of bare-metal C code for a uC !
I believe that this is not reflecting the reality and needs to change.
A proper evaluation of the different design methods needs a coordinated effort to review PA/QA processes to pave the way towards a greater decentralisation of avionic systems.

I’m not blaming the software PAs here.


To-Do-List
1. make
2. things
3. better