Distributed testing of the satellite's components with the use of the Internet on the example of the VirtualCAN

MSc Eng. Marcin Stolarski
Warsaw University of Technology
Faculty of Electronics and Information Technology
Institute of Radioelectronics
Nowowiejska Street 15/17, 00-665 Warsaw, POLAND
M.Stolarski@elka.pw.edu.pl

Abstract: *Modern Satellites are built by international teams with all the components being created in different parts of the world. Common communication protocols are established by the designers to ensure the compatibility of all the components. Despite of all the detailed arrangements differences may arise that can lead to communication errors. To ensure proper communication between all the satellite's components tests have to be carried out. The methods used nowadays are expensive as the simulators and laboratory models need to be built. The article will describe the ways used to test satellites on the example of the Integral Satellite. It will also present an Author's project of a virtual bus that allows testing of the satellite's apparatus and software during the construction stage of the prototype. Its main advantage is the fact of using the Internet as a way of communication to test the components when they are in there parent laboratories all over the world. VirtualCAN bus is one of the suggestions for other teams building the students' satellite named ESEO within the European Space Agency project (ESA SSETI ESEO).*

1. Introduction

Modern Satellites are built by international teams with all the components being created in different parts of the world. Common communication protocols are established by the designers to ensure the compatibility of all the components. Despite of all the detailed arrangements differences may arise that can lead to communication errors. To ensure proper communication between all the satellites's components tests have to be carried out. The methods used nowadays are expensive as the simulators and laboratory models need to be built. Although noticed problems are solved immediately additional tests based on the simulators are to be carried out which results in extending the building time of the satellite. To reduce it the Author suggests introducing communication tests already in the project phase of creating the laboratory model in order to check the compatibility of all parts.

2. Testing methods used nowadays.

During the construction of the satellite's components several stages can be distinguished [1]:

- **Phase O (Mission/Function):** mission analysis, needs identification. During this stage general mission profile, its objectives, initial working environment of the satellite, employed solutions and preliminary costs of the mission are established.
- **Phase A (Requirements):** feasibility study. This stage is to specify any critical elements of the project, set methods of financing it and evaluate the risk of chosen technology. This phase should end with the selection of the project conception and the initial work distribution.

- **Phase B (Definition & Justification):** preliminary definition of project. This stage consists of the following: selecting technology solutions, system requirements, making decisions concerning what to buy and what to build, creating and modeling prototypes of the critical project elements, distributing technical tasks, creating the organization scheme and the promises to the plan of reliability.
- **Phase C (Verification incl. Qualification):** detail project defining, building instruments' models. During this phase such steps can be observed: detailed project assumptions concerning the whole project as well as all the elements are set, qualification and testing of the subsystems and instruments, beginning of the ordering procedure, preparation of the Interface Control Documents [2], building first models (sometimes it is a qualification model with tests being carried out on it) This stage ends with a Critical Design Review being issued. This document is needed for entering the next phase - construction of a flying model.
- **Phase D (Production):** building the flight models, satellites integration. This stage consists of: making qualifications of the instruments, building flying and back-up models, testing and accepting flying models, integrating the devices with the satellite, carrying out ground tests, building Operating Center, time scheduling for usage of instruments. It ends with reaching the decision of launching, making final satellite tests, integrating it with the rocket and launching.
- **Phase E (Utilization):** (instrument usage)
  - **E1** - cruise phase – approaching phase (in case of long distance missions);
  - **E2** - commissioning phase – testing and calibrating the instruments on the orbit;
  - **E3** – operating the on board instruments;
- **Phase F (Disposal):** ending of the mission.

During stage B, C, and D the following component models are created:

- **Laboratory Model:** is built to check the basic instrument assumptions;
- **Structural Thermal Model (STM):** is build for the Prime Constructor and is used for the mechanical and thermal tests of the satellite.
- **Engineering Model (EM):** is built to examine the full operation of the instruments. This model in the final test phase is delivered to the Prime Constructor and then is integrated with the first operating satellite's model.
- **Qualification Model (QM):** is designed to test the response of the instruments in all environmental conditions. It is assumed that this model is the same as the flying one. It then goes through tests of exploitation which are more detailed and extreme than the normal premised circumstances.
- **Flight Model (FM) and Flight Spare Model (FS):** the flying model is tested, integrated with the satellite and launched. The flight spare model is on stand-by and can by swap with the flying one even in the last moment. The second model is used after the launch on the ground as a simulator of the instruments. It is sometimes also used as a flying model in next missions.

When building the models it is essential to test the communication among all the satellite's components. One of the methods to achieve that is to create simulators. The other one used to ensure proper working of all the elements is the communication tests carried out during the

integration of the satellite. The first satellite's functional model is created after assembling the engineering models.
For critical elements simulators are designed earlier to facilitate the examinations and tests of it already during the laboratory model stage.

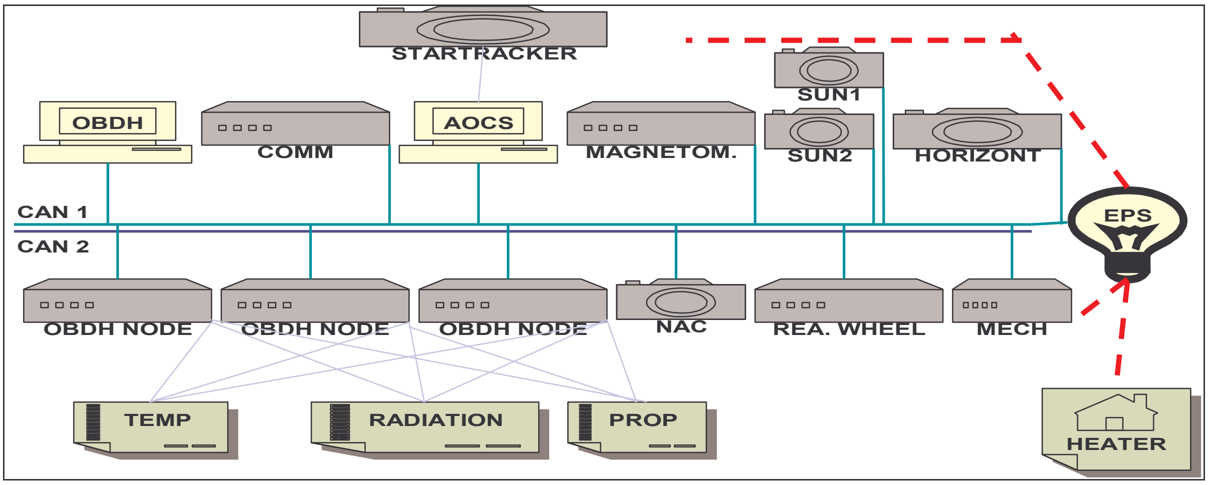
3. ESEO Satellite

ESEO Satellite (European Student Earth Orbiter) [3, 4] is a simple satellite model which is going to be placed on the geo-stationary transfer orbit (GTO). It is designed within the SSETI (Student Space Exploration & Technology) [4] project at ESA (European Space Agency). Its total mass will not exceed 120 kg.

The main objectives of its mission are:

- Failure-free operation in the Universe for a minimum time of 1 month;
- Taking pictures of the Earth;
- Masking failure of one of the subsystems;
- Testing advanced communication protocols between the satellite and the ground station;
- Testing all the satellite's components in the Universe to determine its possible application for other missions.

The indirect goal is to show the possibility of mutual cooperation among students in designing and building the satellite using different Internet methods of communication (IRC, WWW, FTP, NEWS).



Pic 1. ESEO satellite components.

Following components will be installed on a satellite [Pic. 1]

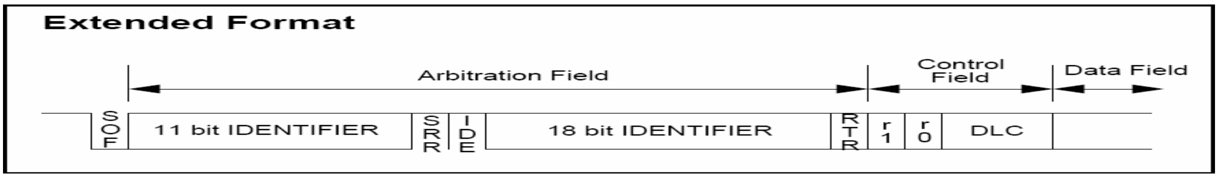
- OBDH – On Bard Data Handling System, main computer
- AOCS – Attitude and Orbit Control System, flight computer
- OBDH_NODE – data and analog interfaces for sensors and execute systems
- MAGNETOMETER – magnetometer sensor
- SUN – two solar position sensors
- HORIZON – Earth horizon position sensor

- STARTRACKER – star position sensor
- PROP – propulsion system with cold-gas
- REACTION WHEEL – reaction wheel for satellite stabilization
- COMM – communication system

The communication among the satellite’s systems will be realized by two CAN buses (the main and reserved one), or through direct connections. As mentioned above the systems are to be designed in such way that failure of any satellite’s components will not affect the full operational of the satellite. To achieve that back-up systems are used: (redundancy of the systems or redundancy of components in the systems)

4. CAN

CAN-bus is one of the most frequently used logical buses in cars [5]. Its popularity was probably the main reason to use it in many students’ satellites projects [6] as one of the best communication solutions. Its main features are: simple wiring system, high reliability due to a built-in transmission correction system which ensures the non-collision and automatic repetition of the packets in case of a CRC sum incompatibility. It also permits granting a priority bus access to devices according to the voting method on the basis of a 29-bit identifier [Pic 2]. In case of a simultaneous transmission of several devices the access is granted to the one which sends longer the low state (connected with the identifier) on the bus. The next advantage of a CAN-bus is the speed of passing the messages as only short ones (max. 8 bytes of length) are sent. Its capacity (up to 2Mbps) is not big but is sufficient for tasks that it was designed for, that is the device control.

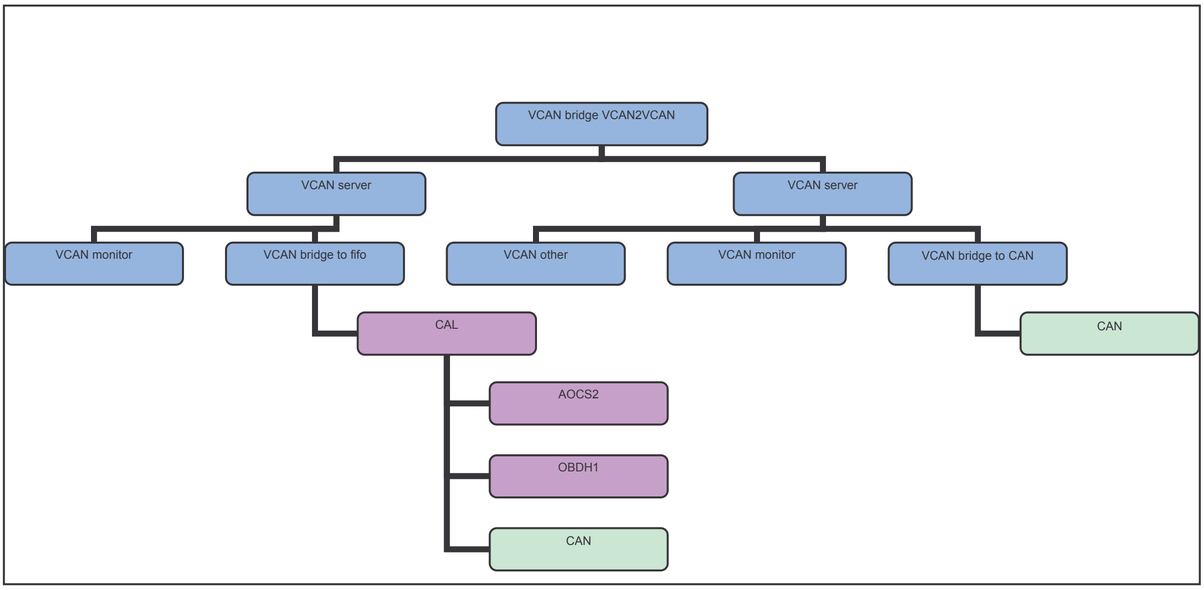


Pic 2. CAN data packet.

5. VirtualCAN

When working on the ESEO Satellite a problem of too many tasks between two computers AOCS and OBDH has emerged. For the present moment both computers are to be built on the same hardware platform (PC104) with an extended version of a Linux system for a RTAI (Real-time Application Interface) running on them. Two processes AOCS and OBDH are going to be executed on the computers that will communicate [6] with the real CAN bus by the process called CAL (CAN Abstraction Layer). One of the earlier propositions for the communication of the AOCS and OBDH with the CAL process was the use of the BSD socket. Additionally, this solution allowed to communicate among the processes when they were working on different computers. Currently the CAL process uses the Real-Time FIFO (RT FIFO) to communicate, whereas the use of the BSD socket is more likely to be applied in creation of the VirtualCAN.

The main objective of the VirtualCAN is to create virtual connections among all the ESEO satellite’s components that would resemble the one used between the CAL with the AOCS and OBDH. Such connection would allow testing communication methods among components already at the Laboratory Model Stage without the necessity of organizing meetings of different working teams in one laboratory. Picture 3 presents an example of designing such network. The main components of the network are the VirtualCAN Servers. They are working similarly to IRC servers because they are awaiting at a specific port for a connection. After receiving a client's request for a connection to the network, the connection is transferred to a different port and at the same time the server creates a new port in place of the old one for another client's connection. When one of the clients sends a data packet it is distributed to other connected clients apart from the one that sent it. This mechanism is essential to eliminate duplication of the packets in the network. Additionally, the VirtualCAN packets have a TTL (Time To Live) counter installed which ensures that in case of network's configuration errors packet multiplication is going to be blocked. The simplest network client is a "VCAN2VCAN" which is a bridge that ensures exchange of information between two VirtualCAN Servers.



Pic 3. Example of VirtualCAN structure.

The next network client is a "VCAN monitor". Its main role is the monitoring of the VirtualCAN Network. It also allows sniffing on the current transmission and introducing specially designed packets in order to simulate the operation of components or different network behaviors.

"VCAN bridge to RT FIFO" is another network client. The role of this device is to connect the CAL, AOCS or OBDH processes to the network.

On picture 3 two more clients can be seen: "VCAN other" and "VCAN bridge to CAN". The first one is a simulator of any of the ESEO satellite's components. It can be used as a simple simulator that answers only to specific network packets or a highly advanced one which permits testing the real algorithms used in the satellite.

The second client is the direct bridge between the VirtualCAN Network and a real CAN-bus. This client will probably be distributed as a package of a specially designed bootable CD Linux version and a USB connected CAN Interface. When run on any computer it will allow a connection of a real satellite's component model to the VirtualCAN Network. If the CAN Server is run on a computer that has an access to the Internet, a possibility of designing satellite's components in different laboratories around the world with a constant and simultaneous logical compatibility control with other components will be possible. You can also test all kinds of mission scenarios and in case of finding errors repairs can be introduced immediately to the satellite's software.

Network created in such way has one disadvantage – it is not true "Real-time" bus. When designing the TCP/IP Network the Real-time events were not taken into consideration. Apart from that the Internet can also introduce additional delays. However, this can be minimized by using a Dial-up connection. Another solution to resolve this problem is to mark the packets with the time stamp but for the Real-time communication tests integration of all the satellite's component models which need such tests is essential. The Author wrote exemplary program [7] that creates the VirtualCAN Network. The VCAN Server is also running (visible in the Internet) to which example clients can be connected. The connection to the "VCAN monitor" is also possible using the SSH Connection. The first tests show that such solution is useful and the further software development is planned to implement in the next generation of satellite systems (such as switching between primary and secondary CAN buses).

6. Summary

When realizing projects which include designing different components in different laboratories mutual communication tests among subsystems play an important role. In students' projects the communication problems which occur during integration of the satellite are the most common and time consuming ones. That is why moving the testing process to the earliest designing phase without raising the costs of the project is such an attractive proposal for the satellites designers. The Author's suggestion to take advantage of the Internet is what suits best this objective. This solution allows to test all the satellite's components during the Laboratory Phase at low hardware costs. The use of the Internet does not limit only to the tunneling of the CAN bus, but can also simulate other buses such as RS485 or Space Wire ones. VirtualCAN software has a module structure and only a few changes are needed to adapt it to a different bus type. The problem might be with designing and writing the software for the USB Interface for a particular bus. It is a good idea when building satellite's components to create such interface as it facilitates the testing process of the satellite's components in the laboratory. Communication tests can be carried out with the use of the virtual bus simultaneously with the laboratory tests.

References:

[1] Piotr Orleański, "Podstawy budowy aparatury pomiarowej", materials for lecture "Technologie Kosmiczne", CBK PAN 2005.

[2] K. Clausen and others, „Integral experiment interface document, part A for instruments", ESA 1998.

[3] Documents for internal use only:

- Presentation "General_SSETI_Presentation".
- Presentation „SSETI_ESEO_Presentation".
- Project documentation ESA SSETI ESEO.

[4] WWW SSETI, http://www.sseti.net .

[5] Robert Bosch, "CAN Specification, Version 2.0", BOSCH 1991.

[6] M. Olak, M. Stolarski, R. Graczyk, "CAN Summary Proposal for ESEO WS8", ESA 2005.

[7] WWW VirtualCAN, http://ktpd.ire.pw.edu.pl/sseti/vcan/ .