SPACEWIRE TEST AND DEMONSTRATION UTILISING THE INTEGRATED PAYLOAD PROCESSING MODULE

Session: SpaceWire Test and Verification

Short Paper

Walter Errico, Pietro Tosi

CAEN AURELIA SPACE srl, Viareggio, Italy

Jørgen Ilstad, David Jameux

European Space Technology Centre, Noordwijk, Netherlands

Riccardo Viviani, Daniele Collantoni

ABSTRAQT srl, Lucca, Italy

E-mail: Jorgen.ilstad@esa.int, david.jameux@esa.int, w.errico@caen.it, p.tosi@caen.it, r.viviani@abstraqt.it, d.collantoni@abstraqt.it

ABSTRACT

A simplified On-Board Data Handling system has been developed by CAEN AURELIA SPACE and ABSTRAQT as PUS[1] over SpaceWire demonstration platform for the Onboard Payload Data Processing laboratory at ESTEC. The system is composed of three Leon2 based IPPM (Integrated Payload Processing Modules) computers that play the roles of Instrument, Payload Data Handling and Satellite Management units. Two PCs complete the test set-up simulating an external Memory Management and the Ground Control units. Communication among units take place primarily through SpaceWire links, RMAP[2] protocol is used for configuration and HK. A limited implementation of ECSS-E-70-41B Packet Utilisation Standard (PUS) over CANbus and MIL1553 has been also realized. The Open Source RTEMS has been running on the IPPM AT697E CPU as RTOS.

IPPM DEMONSTRATION SYSTEM OVERVIEW

The demonstration platform has been designed in order to emulate a simplified On-Board Data Handling System. It consists of three equivalent IPPM and two personal computers. The demonstration architecture is depicted in Figure 1, where a Windows based PC acts as a Ground Control Unit (GCU), while a second PC simulates an external Mass Memory Unit (MMU) and each IPPM module emulates the different components of the on board system:

- An Instrument Unit (IU) plays the role of a scientific instrument that generates data to be processed by the payload data handling unit. The IU simulates an on-board camera and generates raw images data and supports a limited set of PUS services, which allow starting and stopping images data generation.
- A Payload Data Handling Unit (PDHU) manages instrument data incoming from the Instrument Unit. Data are stored by PDHU into a Memory Management Unit (MMU) simulated by a Windows based PC. The PDHU is able to store the data in

both raw and compressed format (JPEG). Compressed data are downloaded to the Ground Control Unit after request.

- The whole IPPM on-board system is managed by a Satellite Management Unit (SMU). This module sends and receives messages to and from the ground control unit and the other IPPM boards. The SMU provide a delayed Telecommands scheduling service and it is in charge to distribute a time synchronization signal to the other boards by means of SpaceWire Time-Codes

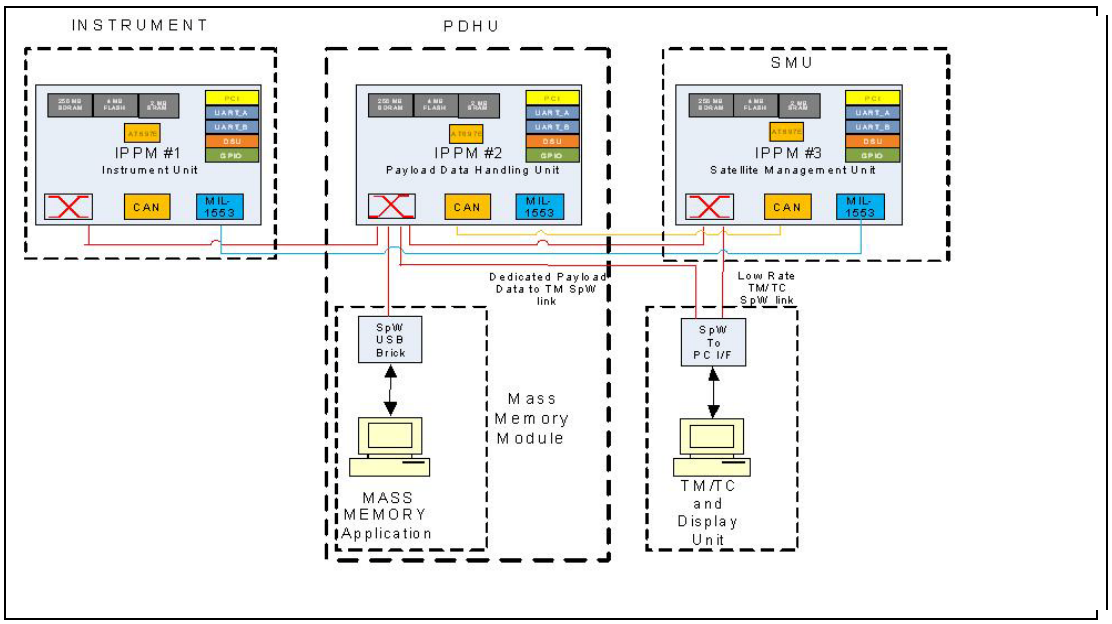

Figure 1: On-Board Data Handling System demonstration architecture

The three IPPM boards involved in the system are custom low power computers based on LEON architecture and running RTEMS as operating system. Each board has a large amount of on-board memory and wide networking resources such as high speed SpaceWire links and, in addition, CAN and MIL-STD-1553 bus interfaces designed to distribute low and medium rate command and control signals.

In the frame of the activity a portable SW library has been developed which supports RMAP transactions and a subset of PUS services. The library provides a platform independent interface to the underlying hardware. All of the three boards run both Housekeeping and Time Synchronization processes: the first process deals with measuring and collecting information about the status of the system, while the second has been designed in order to maintain the clock of both the IU and the PDHU synchronized with the clock of the SMU. The IU runs a camera emulator process that transfers raw image data to the PDHU using PUS Large Data Transfer service. The PDHU stores the received data into the MMU exploiting the RMAP features provided by the software library. An Off-line compression process is in charge to compress the raw images into a dedicated channel of the MMU, while a Data Downlink process download the compressed images to the ground control unit where can be displayed through the GUI. The SMU has a direct SpaceWire link to the ground control unit.

INSIDE THE INTEGRATED PAYLOAD PROCESSING MODULE

The Integrated Payload Processing Module (IPPM) is a single module self-contained computer based on RISC CPU, equipped with a large amount of memory on-board and having a wide inter-networking capability.

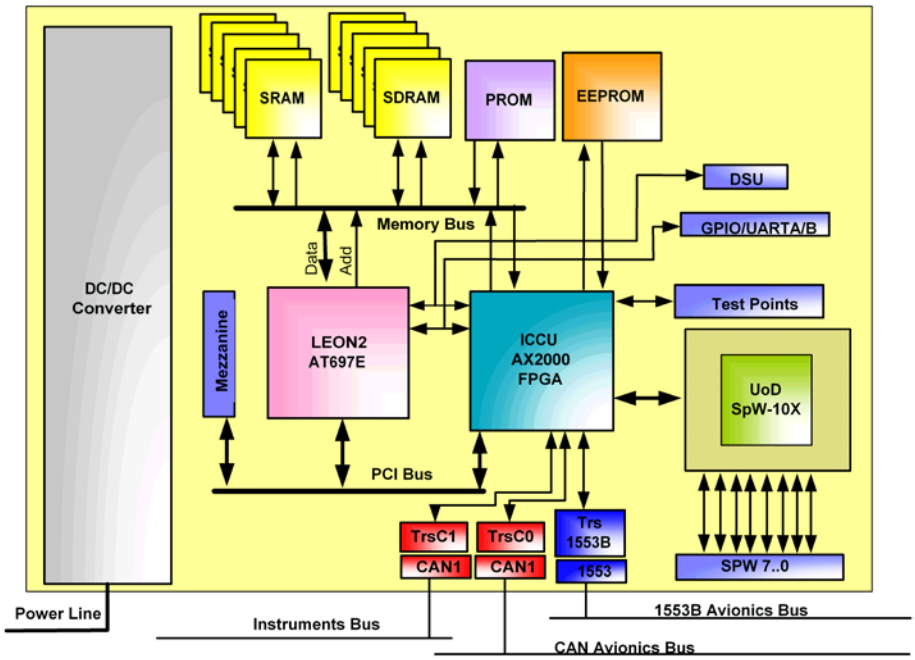

Figure 2: IPPM block diagram

The LEON2 CPU controls the PROM, SRAM and SDRAM components through its memory bus that is also connected to the ICCU FPGA. The ICCU, that internally bridges the memory bus to the FLASH interface, keeps the ownership of the FLASH interface to manage the on board programmability. Data exchange between LEON2 and ICCU takes place on the PCI bus, while ICCU and UoD_SpW-10X router communicate through the data dedicated ports.
The wide IPPM inter-networking is guaranteed by 8 SpW links, connected to the UoD_SpW-10X router, two different CAN buses and a MIL-STD-1553B peripheral block capable of 1553 Bus Controller, Remote Terminal or Bus Monitor functionalities.

ARCHITECTURE OF THE IPPM SOFTWARE LIBRARY

The IPPM software library provides an object-oriented abstraction for both PUS and RMAP standards. The library is organized in different layers to provide a uniform platform-independent interface to the underlying hardware or operating system. The layers are described below, starting from the lower level layer up to the upper level.

The **Communication D evice Layer** directly manages the communication devices. This layer is designed to send and receive data using a specific communication device (SpaceWire, CAN, MIL-1553, etc) and, in principle, makes the upper layers independent with respect to the communication device used to transfer the data.During the development phase various tests concluded that transferring large data sets using CAN/1553 became impractical, thus these buses have been used for transferring HK parameters to the SMU.

The **CoDec Layer** is in charge to encode outgoing messages from the routing layer to the communication layer, and decode incoming messages from the communication layer to the routing layer.

The **Routing Layer** makes the upper layer independent of routing strategies and dispatching methods. While static configuration tables have been utilized for CANbus/1553 topologies, the logic addressing with the 10X-Router adaptive routing capabilities have been exploited for the SpaceWire links.

The **Message Layer** deals with data transformation: application data and requests are encapsulated into objects which reflect the structure of a standard message, such as Telecommand and Telemetry messages of the Packet Utilization Standard protocol (PUS). Telemetry and Telecommand messages are implemented according to the On-Board Software Framework [3], with additional PUS services and RMAP features.

The **Application Layer** decouples a generic Application Processes from the lower layer providing them a standard communication interface.

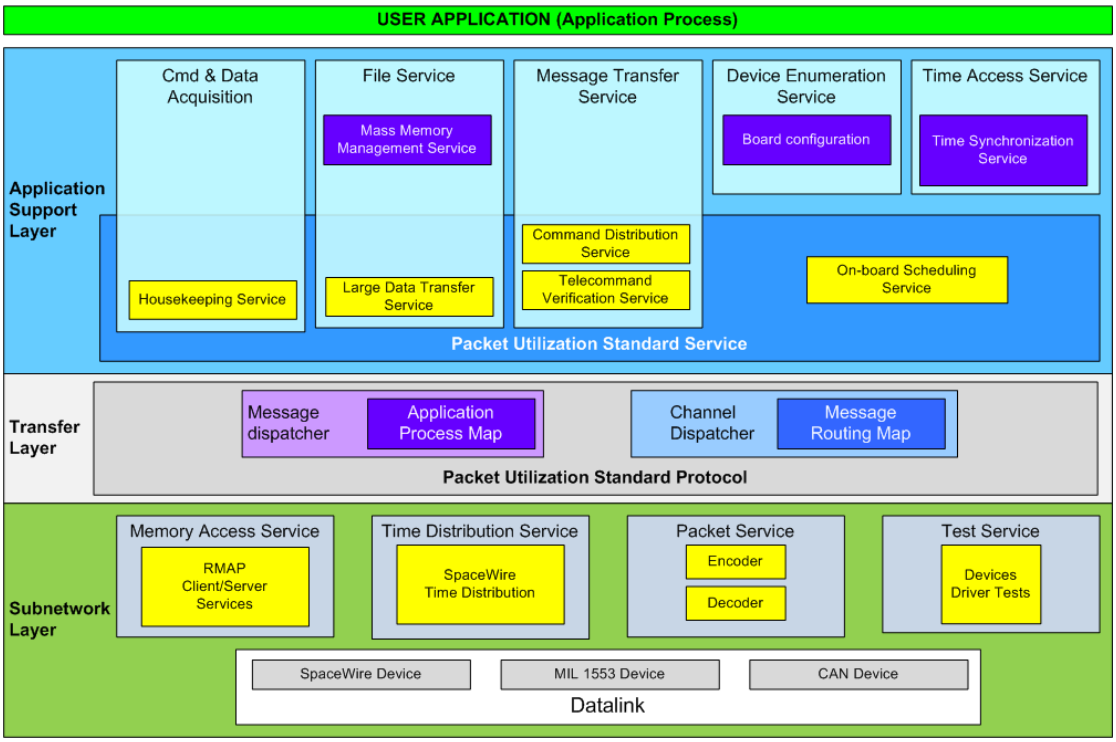

Figure 3: Main modules of the IPPM software library

The current version of the software library is organized in modules, as depicted in Figure 3, and supports Telecommand Verification, Housekeeping, Large Data Service, Command Distribution and On Board Scheduling standard PUS services together with RMAP capabilities; while SpaceWire, MIL-1553 and CAN are the currently supported communication devices.

REFERENCES

1. ECSS, E-70-41 Telemetry & Telecommand Packet Utilization Standard.

2. ECSS, E-ST-50-52C SpaceWire - Remote Memory Access Protocol (normative).

3. Alessandro Pasetti and Vaclav Cechticky, Automatic Control Laboratory of ETH-Zurich, *On-Board Software Framework*, http://www.pnp-software.com/ObsFramework.