JAXA Status Report Part 2

SpaceWire devices developed in JAXA's SpaceWire R&D activities in the last 3 years (and coming several years).

Takayuki Yuasa, Tadayuki Takahashi
JAXA

Masaharu Nomachi
Osaka University

20th SpaceWire Working Group meeting
at ESTEC

Japan Aerospace Exploration Agency
Institute of Space and Astronautical Science

Background - How to develop SpaceWire components for ASTRO-H

Developments of mission instruments

- SpaceWire was defined as the only data interface between subsystems and the SC bus.
- Multiple institutes(NASA, ESA, SRON, CSA), industries (NEC, MHI, SHI), and universities (>24).

ASTRO-H Onboard Component map

What was necessary:

- 1. Standardized SpaceWire device that can be used to develop and test one's onboard component.
  - SpaceWire-to-GigabitEther (JAXA/Shimafuji)
  - SpaceCube Mk2 (NEC/Shimafuji Electric)
- 2. Test environment for large-scale SpaceWire network.
  - SpaceWire microTCA Backplane and modules (MHI/UoO/JAXA)
- 3. Verification scheme
  - SpaceWire I/F layer, RMAP layer, and Telemetry/Command service layer.
  - Portable end-to-end test system that can be connected to the real SC operation software.

Development of SpaceWire devices mainly for ground testing
in the reliable SpaceWire network project Nov 2010 - Mar 2013.

1. Generic SpaceWire I/F : SpaceWire-to-GigabitEther

Overview

- SpaceWire packet/timecode transfer via TCP/IP.
- No driver software is necessary.
- 4 external SpaceWire ports.

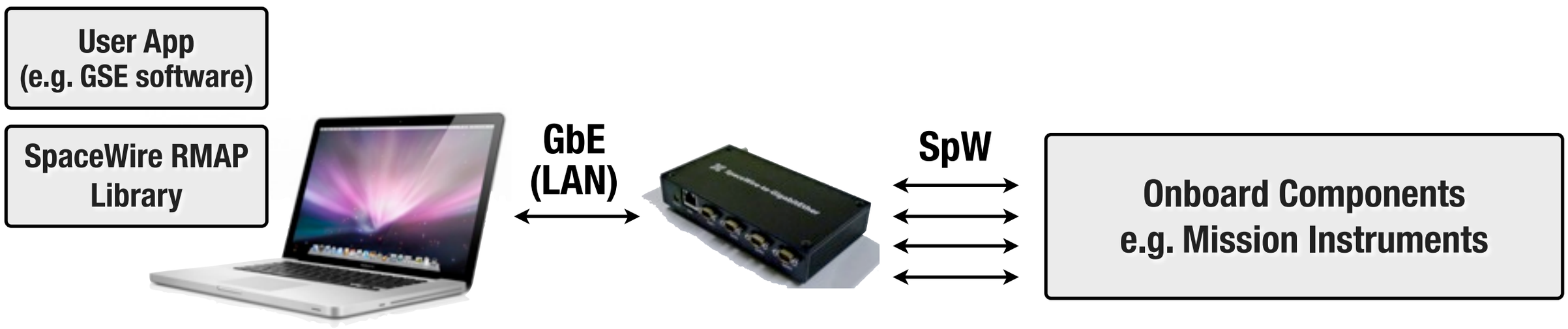
Available from Shimafuji.

Performance

- Max 200 MHz SpaceWire link frequency.
- TCP/IP at max 650 Mbps.

Applications

- GSE software running on ordinary PC.
- Embedded OS programing is not required.
- High throughput data read out of mission instruments.



1. SpaceWire-to-GigabitEther details

Item	Description	Remarks
Interfaces	SpaceWire x 4 GigabitEthernet x 1	Max 200 MHz SpaceWire link frequencies.
Router	1 Internal SpaceWire Router with 9 ports	
TCP Sockets	2 server sockets on TCP 10030 and 10031	
IP Address	192.168.1.100 (default) Modifiable via web browser	
Supported Systems	Linux/Mac/PC	Others with GbE could be used.
Packet Transfer	Packets with EOP/EEP/no-EOP can be transferred	
TimeCode	Supports both emission and reception on PC	
Software	SpaceWire RMAP GUI or User-dependent Applications	SpaceWire RMAP Library is provided for C++ development.
Size and weight		

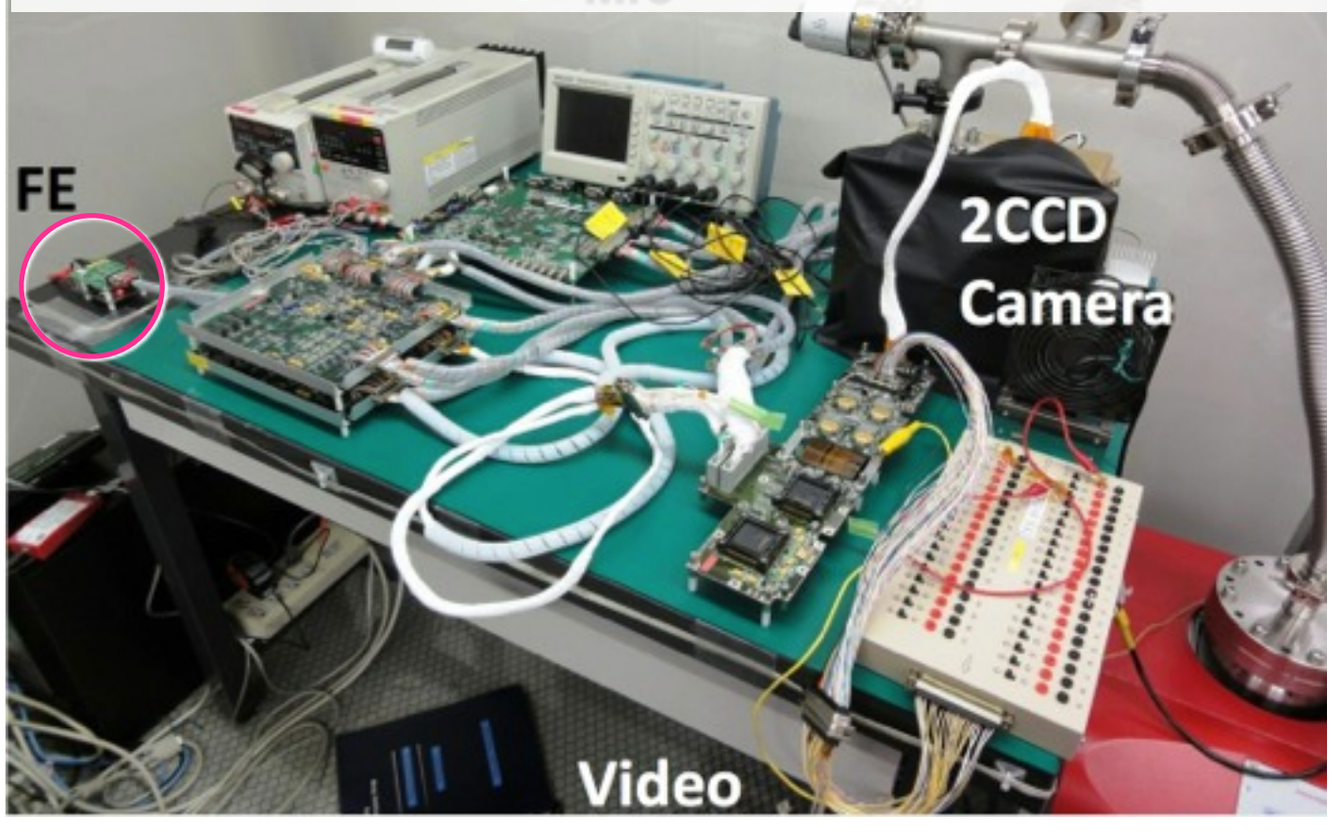
1. SpaceWire-to-GigabitEther example usages

Mission instrument development

- Used in more than 20 universities/institutes.
- GSE software development in C++ by scientists and engineers in universities and companies.
- RMAP API available in the open-source SpaceWire RMAP Library.
- High speed readout for instrument calibration/test purposes.
  (Max 650Mbps from 4 external SpaceWire links)

Hard X-ray Imager

X-ray CCD Imager



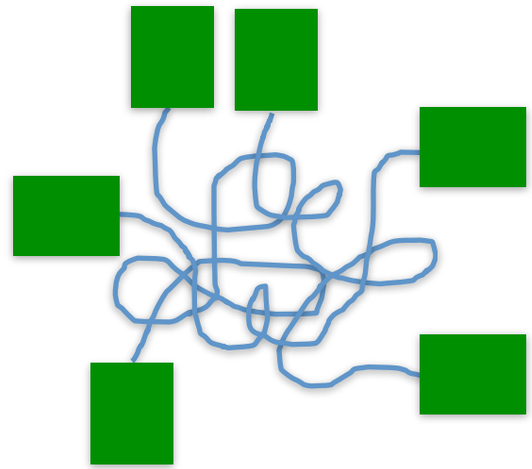
2.1 Test environment for large-scale SpaceWire network

Simplification of development and test

- Test of onboard SpaceWire network of a large satellite can involve more than 20-30 subsystems.
- Construction of test environment (placement, cabling, etc) can be problematic.
- Test bed that allows rapid integration can reduce man-hour needed for tests.

Requirements

- Crate (or subrack) that collects multiple SpaceWire modules, i.e. processor and FPGA boards.
- Rugged mechanical support.
- Good power supply and grounding.
- Good cooling.
- Commercial availability (reduced test cost).
- 100 Ω transmission line for SpaceWire.



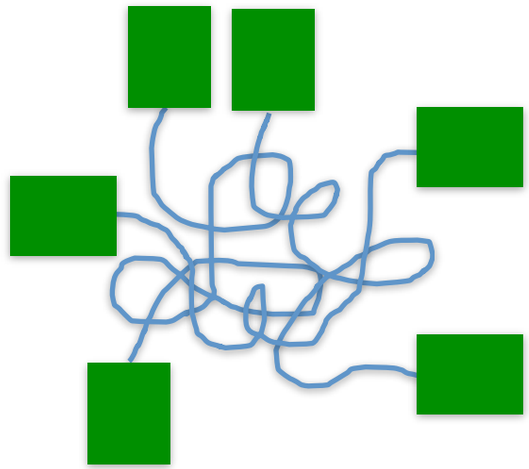
2.1 Test environment for large-scale SpaceWire network

Simplification of development and test

- Test of onboard SpaceWire network of a large satellite can involve more than 20-30 subsystems.
- Construction of test environment (placement, cabling, etc) can be problematic.
- Test bed that allows rapid integration can reduce man-hour needed for tests.

Requirements

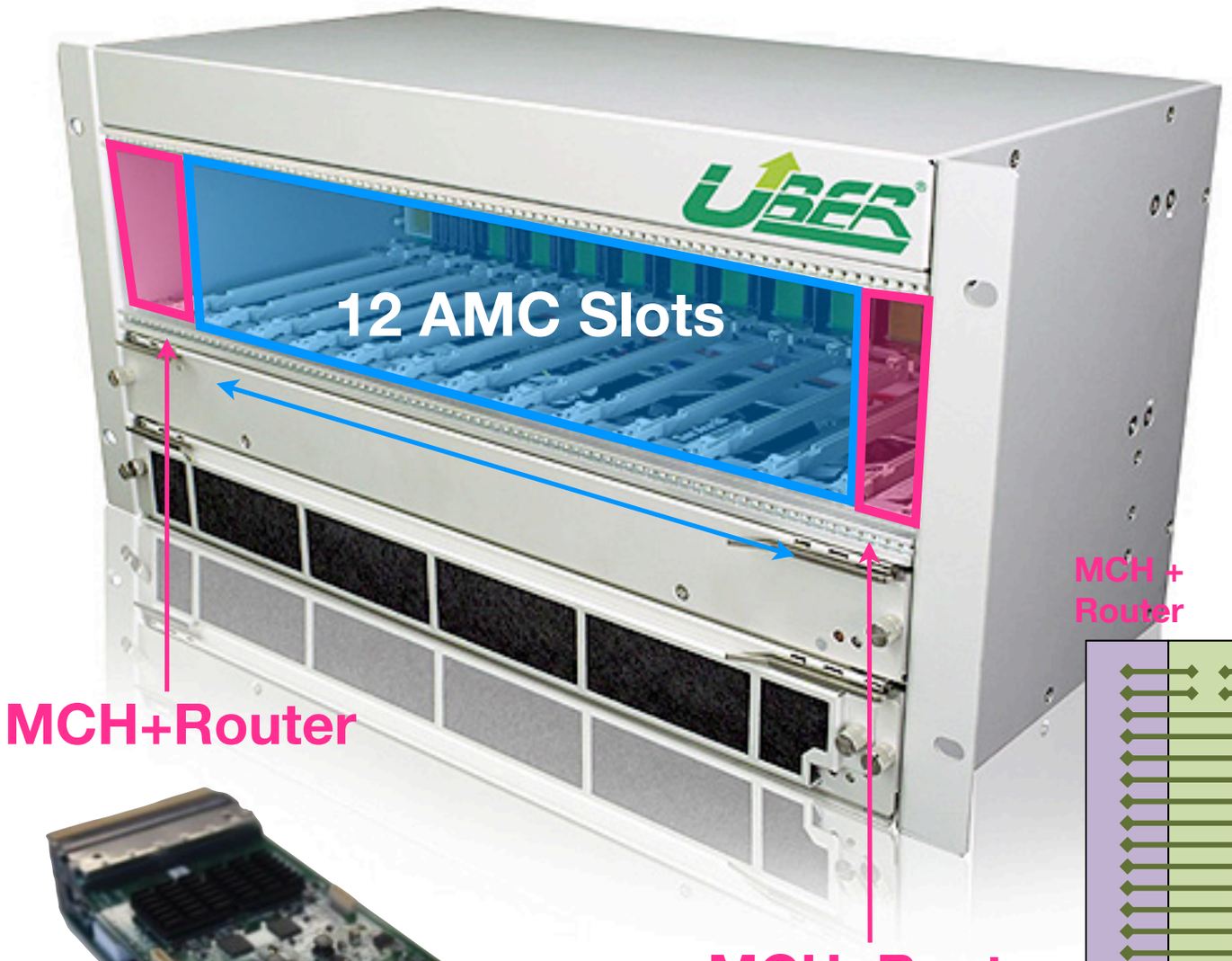
- Crate (or subrack) that collects multiple SpaceWire modules, i.e. processor and FPGA boards.
- Rugged mechanical support.
- Good power supply and grounding.
- Good cooling.
- Commercial availability (reduced test cost).
- 100 Ω transmission line for SpaceWire.

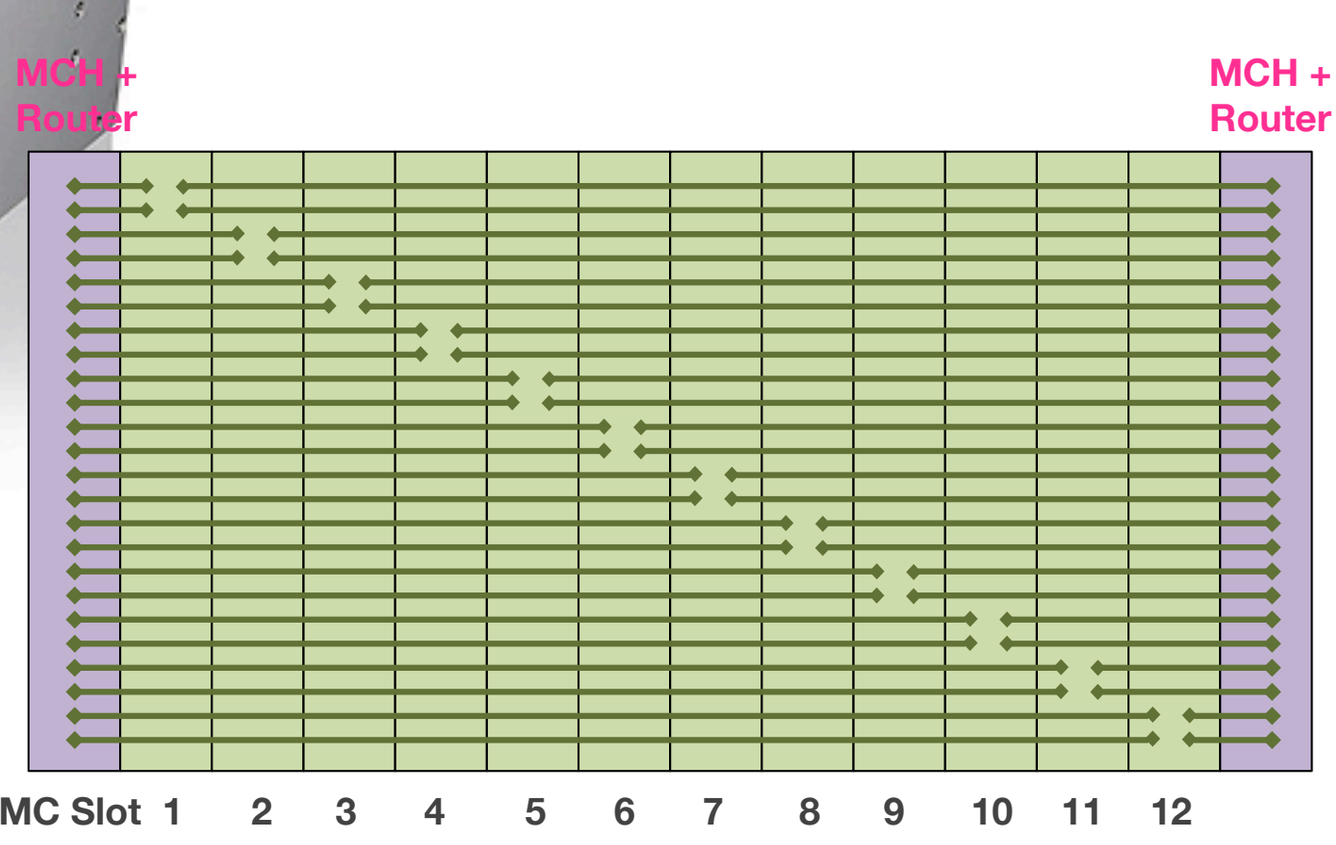


We selected μTCA™ + AdvancedMC™.

2.2 Example of commercial 12-slot microTCA backplane

- Collects 12 AMC modules in a normal microTCA crate.
- Router in a crate-controller module (or MicroTCA Carrier Hub; MCH) interconnects 24 links on the backplane.

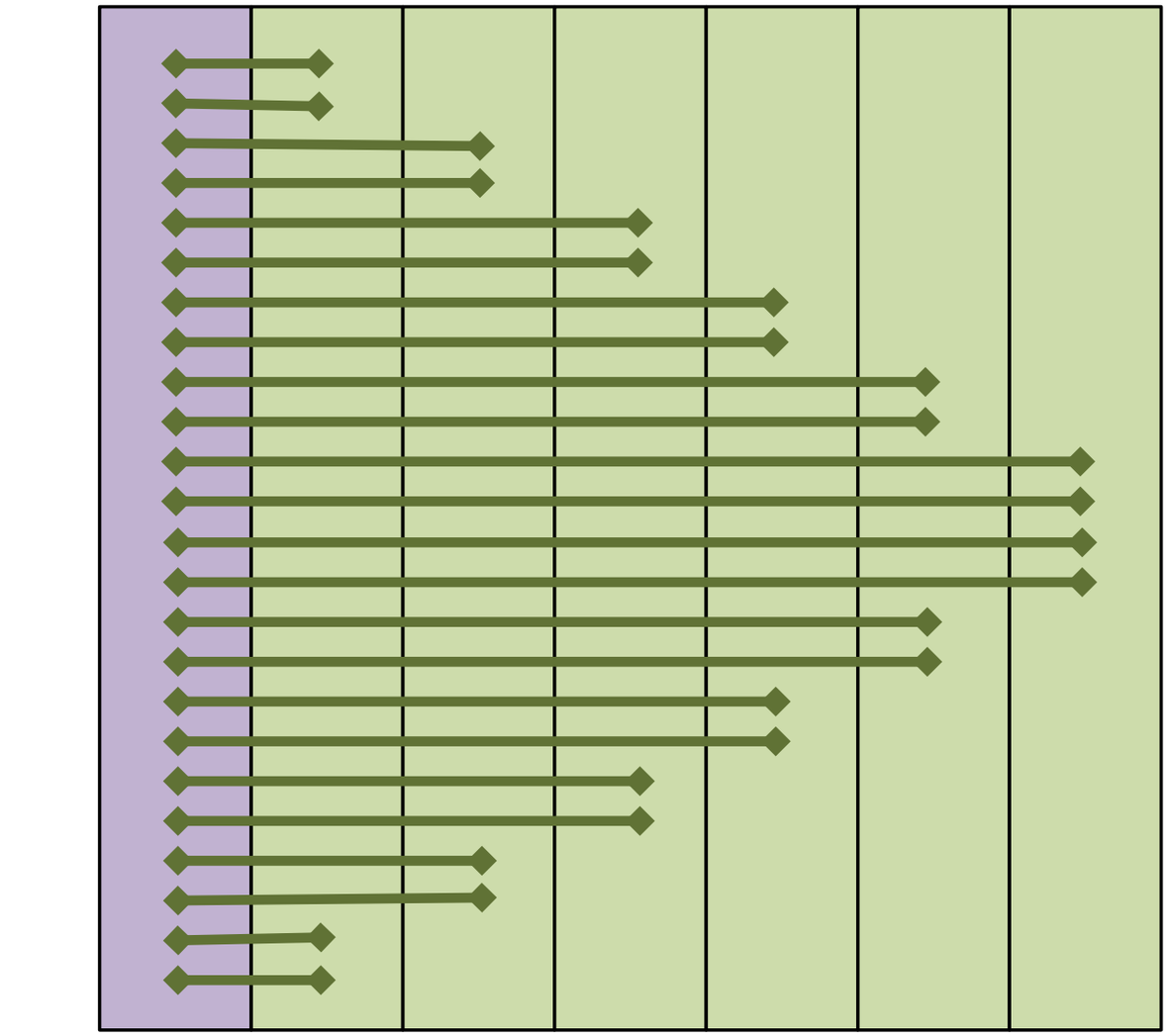


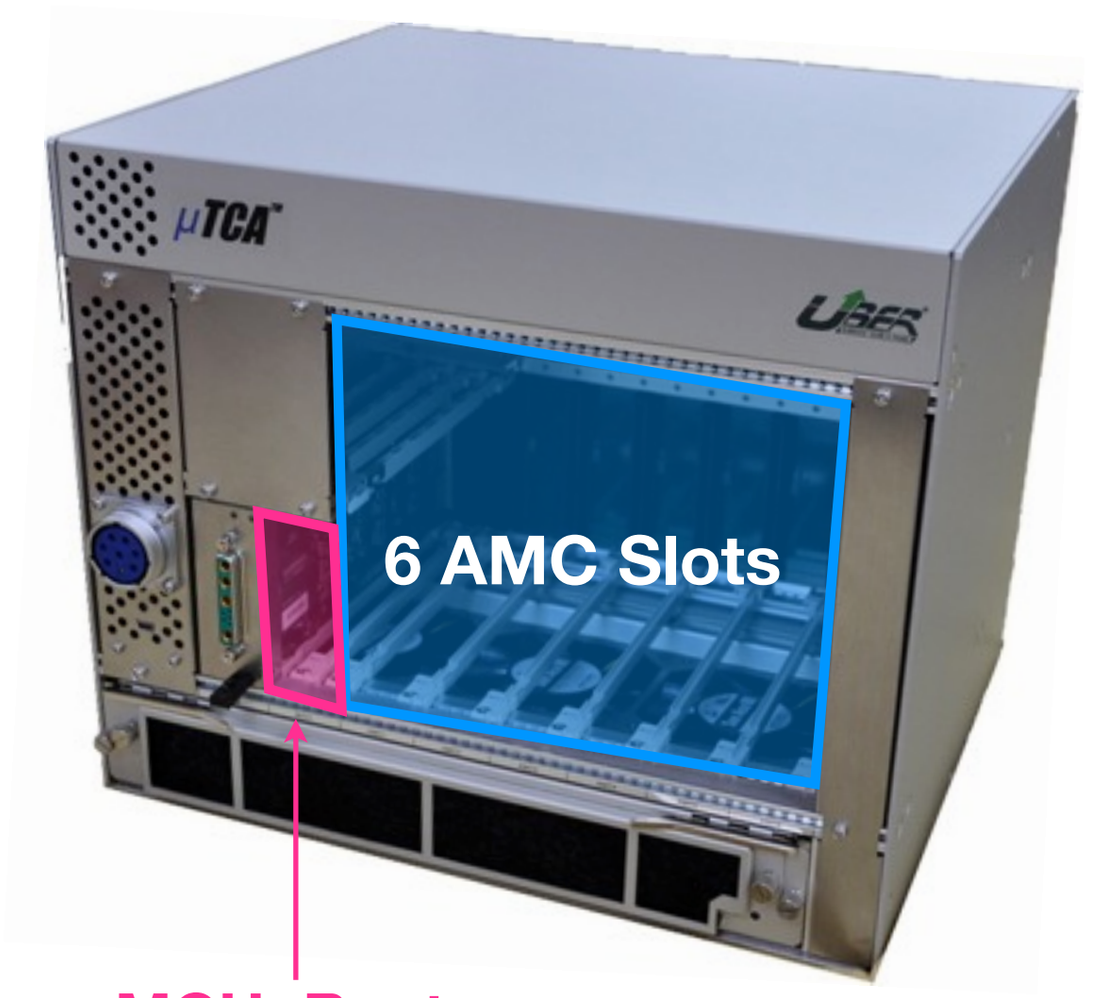


2.4 For smaller systems - Customized 6-AMC-slot crate

- We have developed a modified uTCA-like backplane for smaller systems.
- SpaceWire links to redundant-MCH are flipped to non-redundant-MCH.
  i.e. 1 MCH+Router handles at most 24 SpaceWire links from 6 AMC modules.
- Affordable for small experiments, university groups.
- The same AMC modules can be used with modified routing configuration in the router.

MCH +Router





2.5 Available SpaceWire AMC modules

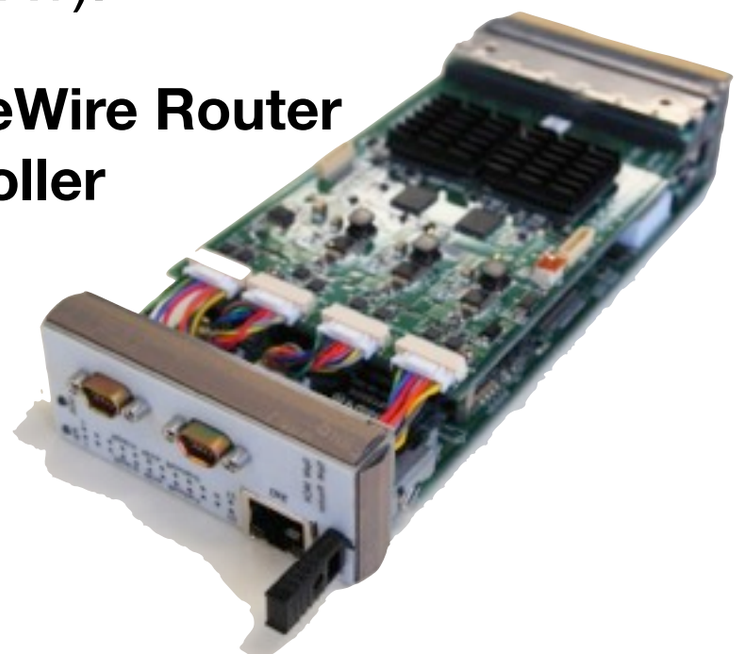
- For SpaceWire network integration test of JAXA's ASTRO-H satellite, Mitsubishi Heavy Industry, Japan, developed CPU board and General-purpose FPGA board based on the SpaceWire AMC concept.

AMC SpaceWire CPU Board

AMC SpaceWire FPGA Board

- Generic SpaceWire I/F Board, SpaceWire-to-GigabitEther, and 28-port Router modules are available from Shimafuji Electric (support from Japan Space Systems/MEXT).

28-port SpaceWire Router + Shelf Controller

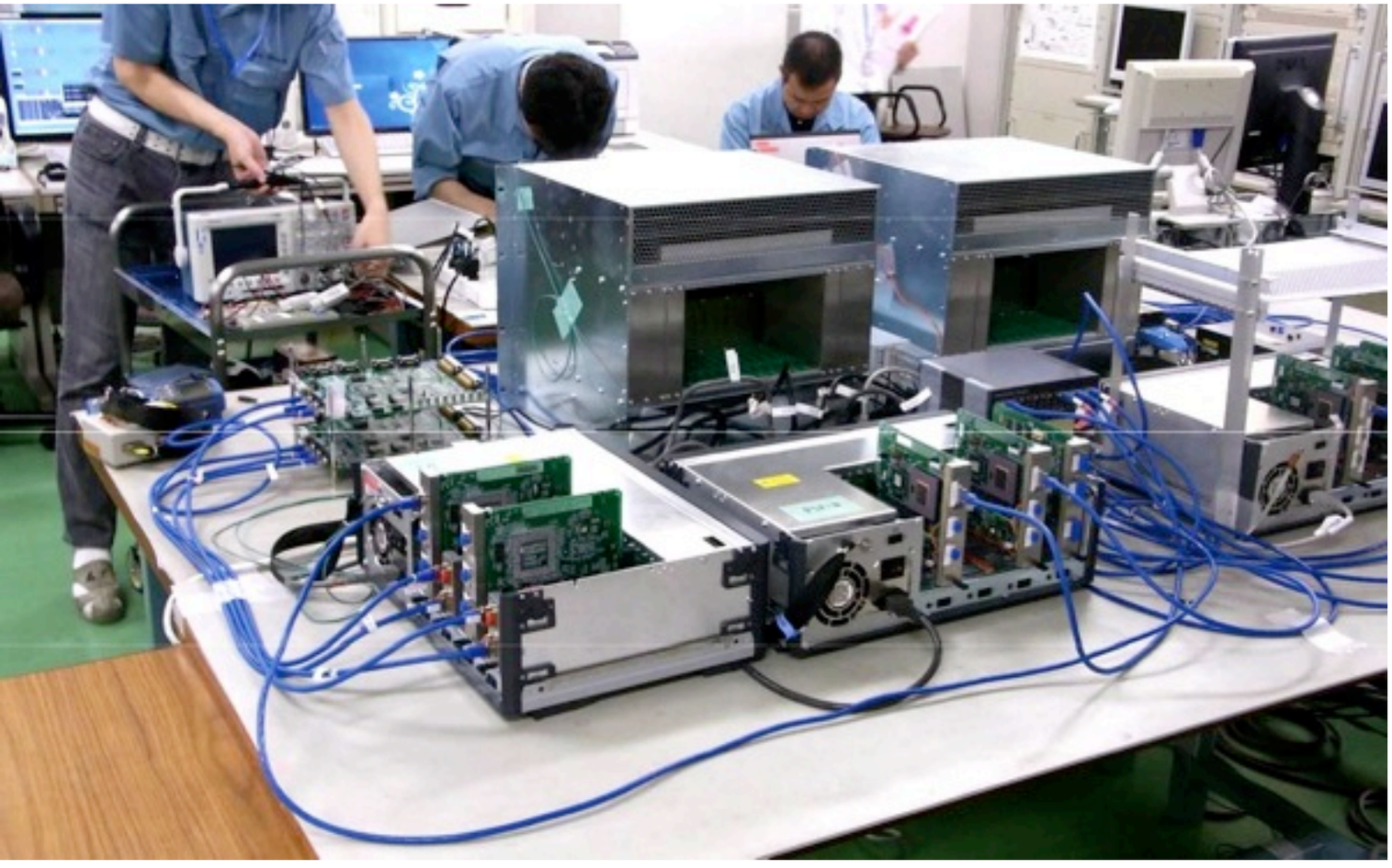


Generic SpaceWire FPGA Module

2.6 Example usage in ASTRO-H

SpaceWire Network Test (2011)

- 5 micro-TCA shelves with 10 processor modules and 10 FPGA modules, simulating mission instrument electronics.
- Thanks to the backplane, we could reduce cables, and therefore complexities in constructing a test setup.
- 1 shelf => 1 mission instrument.
- Interface between SC (SMU + DR) and mission instrument electronics was confirmed.
- Optimum value for router timeout duration was recognized (~2ms for A-H).



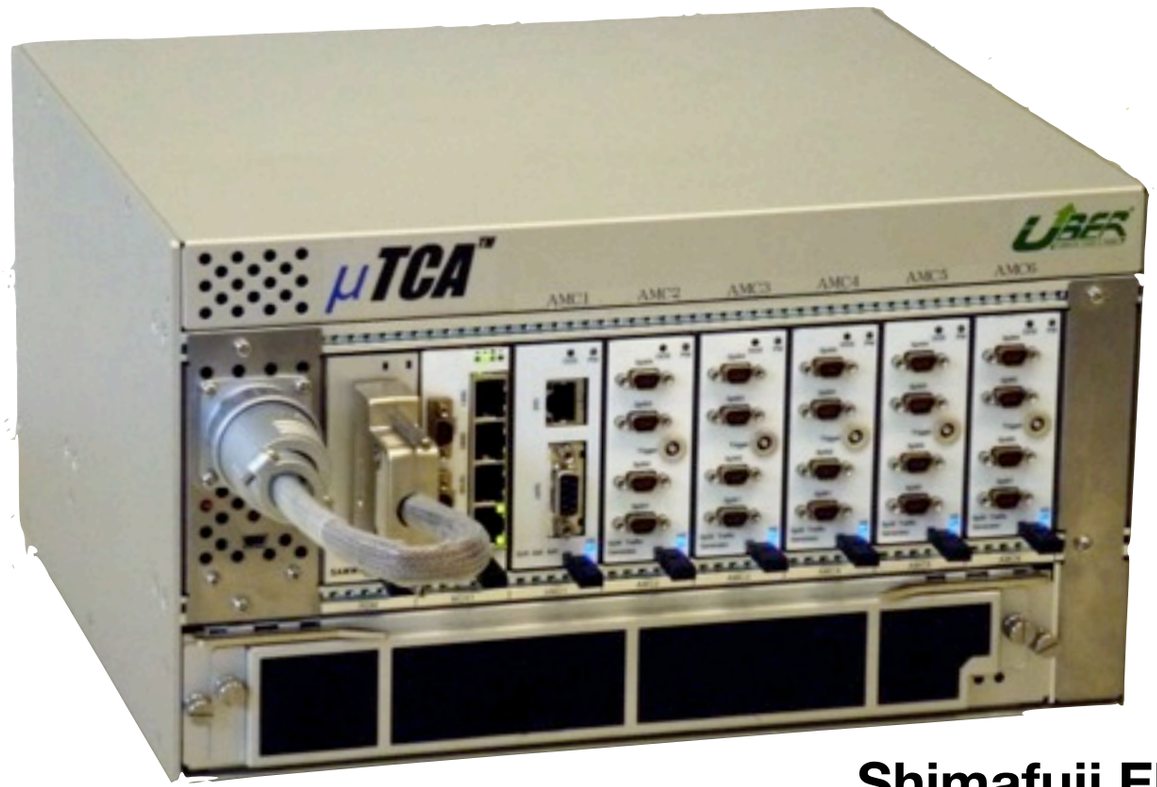
2.7 SpaceWire Traffic Generator - an application of SpW backplane

Overview

- A SpaceWire uTCA Backplane-based SpaceWire packet generator.
- Sends packets according to a given scenario.
- Records received packets.
- Traffic injection to a large network.
- See Yuasa et al. in SpaceWire Conference 2013.

Scalable SpaceWire port number

- A SpaceWire Traffic Generator module has 4 output SpaceWire ports.
- 6-slot shelf => Maximum number of output SpaceWire ports of 24.
- 12-slot shelf => 48



Shimafuji Electric/JAXA

Backplane

PC

SpW GbE

MDM1
MDM2
MDM3
MDM4

28-Port Router

Router

RMAP

TRIG/ Reset

Up to 24 SpW links

TRIG

Reset

SpaceWire

SpW

UUT

MDM1
MDM2
MDM3
MDM4

Traffic Generator module

Traffic Manager

RMAP

Router

SDRAM 16 MB

SDRAMC

Descriptor Manager

RMAP

RMAP

SpaceWire

MDM1
MDM2
MDM3
MDM4

Traffic Generator module

Traffic Manager

RMAP

Router

SDRAM 16 MB

SDRAMC

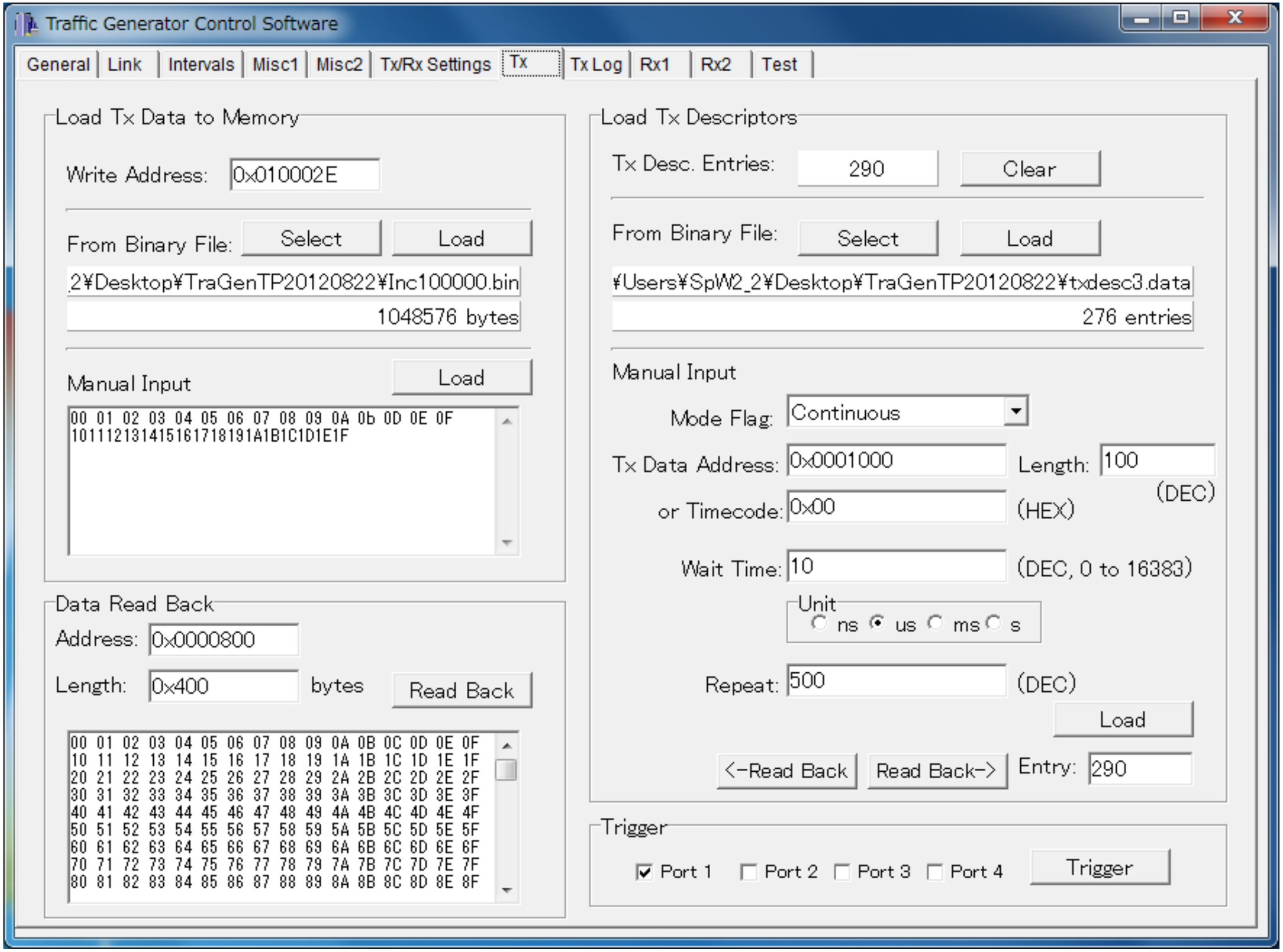
Descriptor Manager

RMAP

RMAP

SpaceWire

Traffic Generator Control software (Shimafuji)



3.1 Verification scheme

Purposes

- To reduce risks of system-level electrical interface test by front-loading end-to-end Telemetry/Command tests at subsystem level.
- Standard verification scheme was defined for ASTRO-H.
- This was possible because SpaceWire is the only data interface (no RS422, no MIL-1553B, nor dedicated interface between SC and subsystems).

3 steps of subsystem-level verification

Step1

SpaceWire I/F layer test
SpaceWire Conformance Tester (STAR-Dundee)

Step2

RMAP I/F layer test
RMAP Tester (STAR-Dundee)
+ Test cases for ASTRO-H Standard RMAP Memory Map

Step3

Telemetry/Command I/F
SMU Simulator Light (NEC)
+ Satellite Operation Software (JAXA/NEC)

System-level I&T in Japan

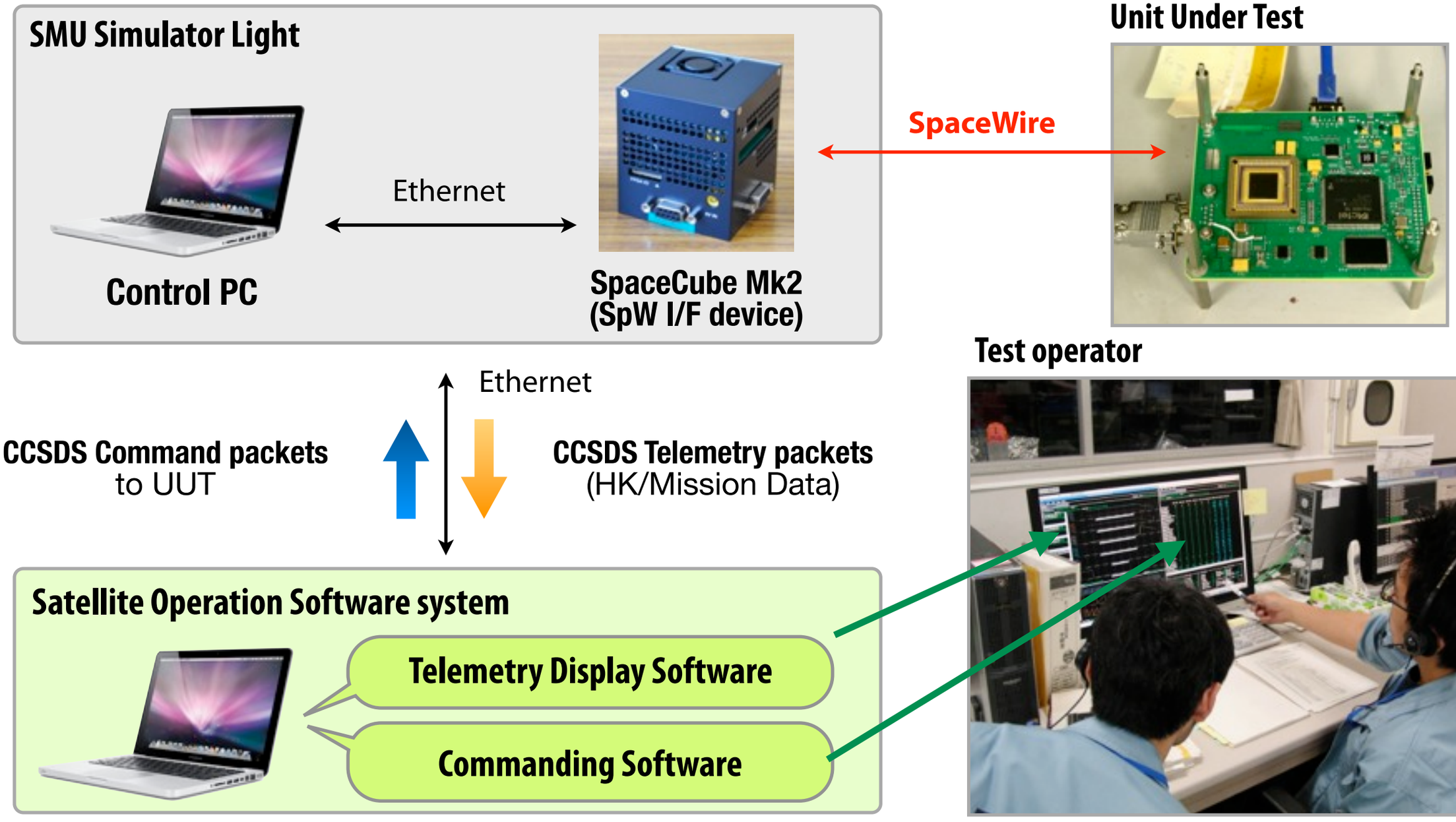
SpW Conformance Tester (STAR-Dundee)

SMU Sim Light (NEC)

3.2 Satellite Management Unit (SMU) Simulator (NEC/JAXA)

Overview

- Implements SMU's Telemetry/Command services on the SpaceCube Mk2 computer.
- Interfaces the real satellite operation software (GSTOS).
- Test scenarios/scripts are directly transferred to FM tests and post-launch operations.
- The simulator was distributed to each company/university.
- Helped to find interface problems of subsystems well before the system I&T.



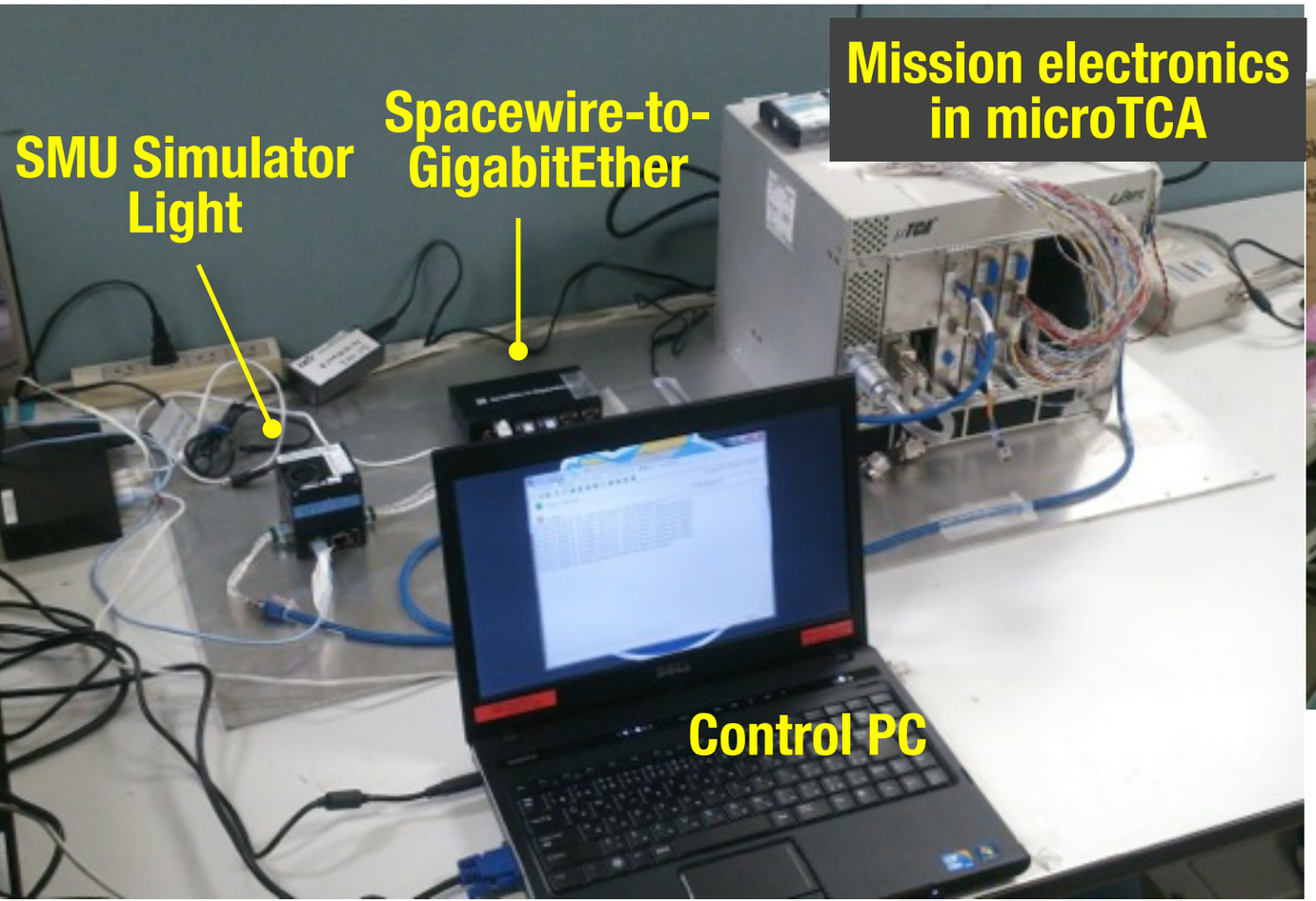
3.3 Services implemented in the SMU Simulator

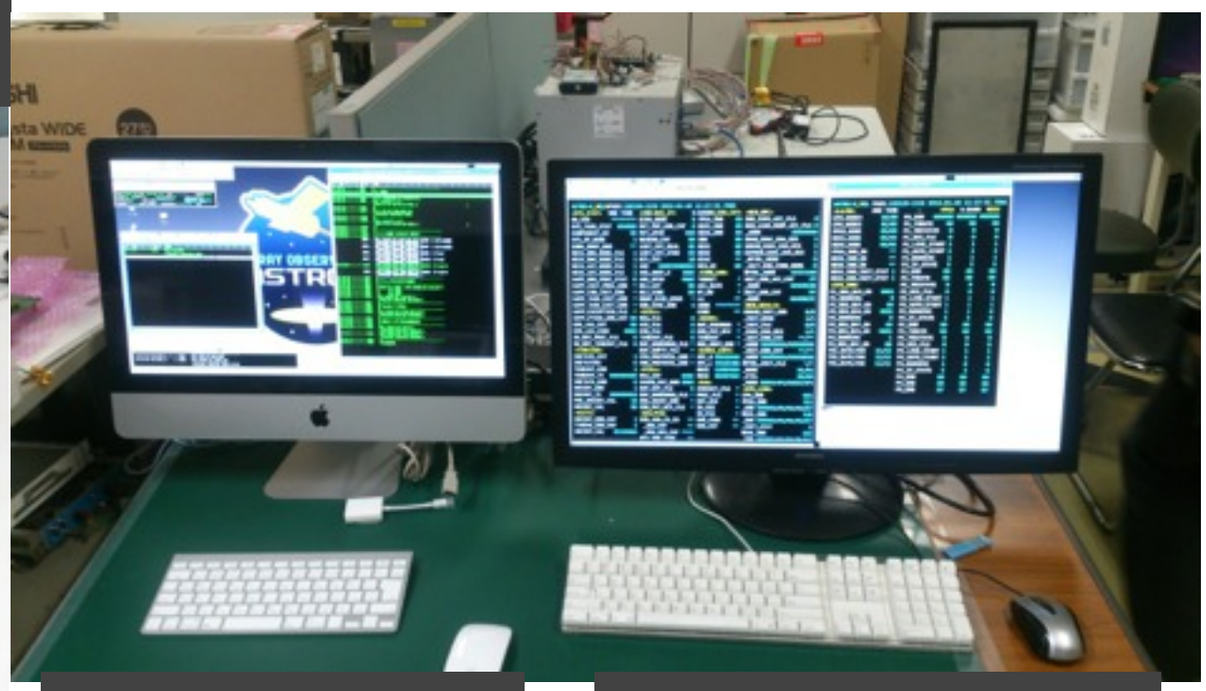
Communication services

- Command write. --- Single RMAP Write
- HK collection. --- Single RMAP Read
- Request polling and response. --- Polling, then single RMAP Read
- Mission data collection. --- Polling, then multiple RMAP Read

Test procedure

- Basic communication test of each services.
- Check all defined commands work as expected.
- Check behavior against failure cases.
- Scripted test of command sequence that will be used in the flight operation.
- Telemetry monitoring while commands are sent.





Commanding Software

Telemetry Display Software

3.3 STEP3 tests in various places (with various SpaceWire/RMAP IP cores)

Test records

Component	Place	IP Core type	
X-ray instruments	Japan (MHI/Univ.)	MHI	2011-
Cooler driver	Japan (SHI)	NEC	2011-
ADR cooler driver	NASA/GSFC	NASA	Nov. 2011, Jun. 2012
Filter wheel system	Swiss	ESA	Sep. 2011, Mar. 2012
Laser metrology system	Canada	ESA	Jul. 2011, May-Jun. 2013

- Not only in ASTRO-H, the same SMU Simulator has been used in other JAXA missions such as SPRINT-A space observatory (2013) and HAYABUSA-2 sample return mission (2014).

Lessons learned

- Define assumed failure cases clearly/precisely in the design rule document to get coherent response to failures (e.g. response to non-RMAP Protocol ID; incorrect RMAP IP core usage could block further packet reception).
- Portable test environment made oversea tests easy. In the on-site tests, verification of maker's own GSE were also possible based on devices from other suppliers.
- Single data interface design in both of SC bus and mission seems very effective to reduce cost of tests and interface adjustment. (TC over SpaceWire is promising)

3.3 STEP3 tests in various places (with various SpaceWire/RMAP IP cores)

Test records

Component	Place	IP Core type	
X-ray instruments	Japan (MHI/Univ.)	MHI	2011-
Cooler driver	Japan (SHI)	NEC	2011-
ADR cooler driver	NASA/GSFC	NASA	Nov. 2011, Jun. 2012
Filter wheel system	Swiss	ESA	Sep. 2011, Mar. 2012
Laser metrology system	Canada	ESA	Jul. 2011, May-Jun. 2013

- Not only in ASTRO-H, the same SMU Simulator has been used in other JAXA missions such as SPRINT-A space observatory (2013) and HAYABUSA-2 sample return mission (2014).

Lessons learned

- Define assumed failure cases clearly/precisely in the design rule document to get coherent response to failures (e.g. response to non-RMAP Protocol ID; incorrect RMAP IP core usage could block further packet reception).
- Portable test environment made oversea tests easy. In the on-site tests, verification of maker's own GSE were also possible based on devices from other suppliers.
- Single data interface design in both of SC bus and mission seems very effective to reduce cost of tests and interface adjustment. (TC over SpaceWire is promising)

4. On-going and near future activities

New R&D activities from Q2 2013

- SpaceFibre codec IP developments.
- Non-blocking SpaceWire Router based on framing/segmentation.

Activities done by Japan SpaceWire Users Group members

- SpaceFibre developments (NEC/NTSpace, MELCO)
- SpaceWire Middleware porting to Leon CPU. (Nagoya U./Shimafuji Electric/MHI)
- 24-port Packet Recorder (Shimafuji Electric)
- Lightweight SpaceWire cable (ITTCanon/Junkosha)

Summary

- Driven by ASTRO-H, JAXA, together with NEC, Shimafuji Electric, and other companies, has been developing SpaceWire devices for ground testing of satellite components.
- Resulting products are heavily used in on-going projects.
- Some of the products will be internationally available.
- New R&D on SpaceFibre and non-blocking SpaceWire router will start shortly.
- Other players in Japan also actively studying/developing SpaceWire-related components/software.

Thank you very much for your attention.
Welcome comments/requests/advices on JAXA SpaceWire R&D activities.

Consideration on non-blocking SpaceWire Router

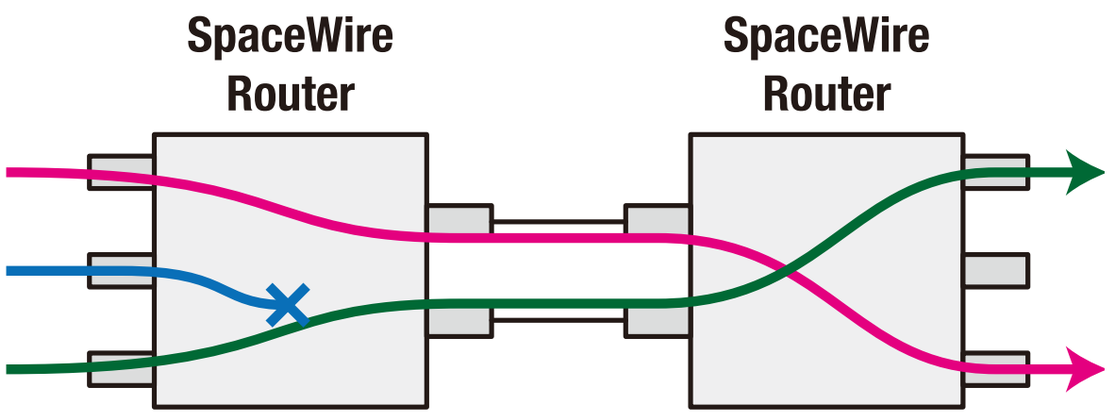
Requirements

- When applying SpaceWire to the spacecraft bus, determinism, timeliness, reliability are strongly required (as has been discusses repeatedly).
- SpaceWire-D will be a baseline for deterministic design.
- QoS of a network should be guaranteed even when a node fails.
  - SpaceWire Router developed based on the current standard can block packets when a destination port is occupied by earlier packet (worm-hole routing).
  - Bubbling-idiot node can block important routers.
  - Links connecting two routers can be a bottle neck in terms of blocking.
- Solution for the blocking should have minimum impact on existing end nodes.

Background

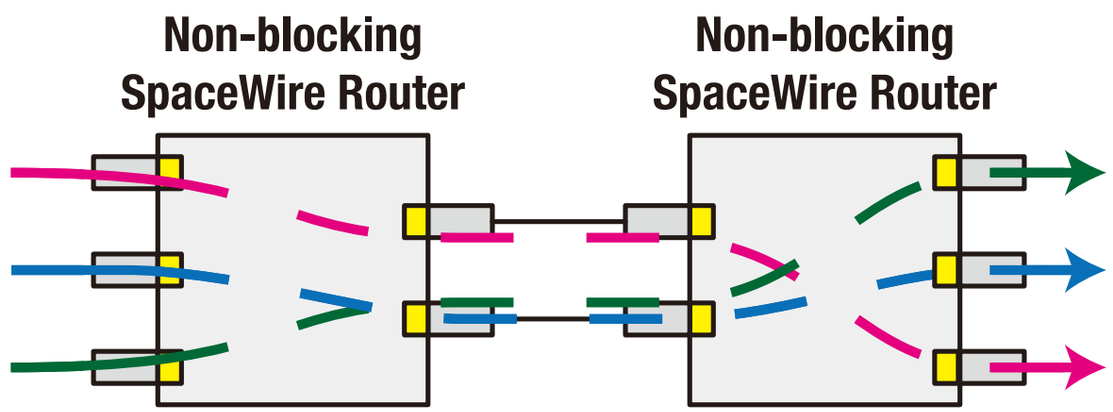
- SpaceFibre has a virtual channel capability, and this could be reimported to SpaceWire Router.

Simple worm-hole routing



Blue packet will be blocked until Green packet is completely transferred through the router.

Non-blocking router based on segmentation



Packets are chopped into segments among the routers, and latency caused by blocking can be reduced.

Tentative structure of non-blocking router

- Based on Framing, Retry, and Multi-lane layers of SpaceWire. (see previous WG discussions)
- May require additional configuration to open virtual channel between router ports.
- Prototype implementation from Q2-3 2013.