SPACEWIRE/RMAP-BASED DATA ACQUISITION FRAMEWORK FOR SCIENTIFIC INSTRUMENTS: OVERVIEW, APPLICATION, AND RECENT UPDATES

Session: SpaceWire Missions and Applications

Short Paper

Takayuki Yuasa, Wataru Kokuyama, Kazuo Makishima, Kazuhiro Nakazawa,
The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan 113-0033

Hirokazu Odaka, Motohide Kokubun, Takeshi Takashima, Tadayuki Takahashi,
Institute of Space and Astronautical Science (ISAS), Japan Aerospace Exploration Agnency (JAXA), 3-1-1 Yoshinodai, Sagamihara, Kanagawa, Japan 229-8510

Masaharu Nomachi,
Laboratory of Nuclear Studies, Graduate School of Science, Osaka University, 1-1 Machikaneyama, Toyonaka, Osaka 560-0043

Iwao Fujishiro, and Fumio Hodoshima

Shimafuji Electric Incorporated, 8-1-15 Nishikamata, Ota, Tokyo, Japan 144-0051

E-mail (TY) : yuasa@juno.phys.s.u-tokyo.ac.jp

ABSTRACT

We report a data acquisition framework for scientific instruments based on SpaceWire interfaces. The framework consists of computers and front-end electronics which have SpaceWire ports in their small footprints. RMAP protocol stack, portable class library, and a template of hardware logics are provided with these hardwares for users to implement their own applications. The framework has been employed in multiple practical developments mainly for Japanese X-ray satellite. The protocol stack was also utilized in a flight model of the Japanese small demonstration satellite which was launched in 2009, and successfully tested its SpaceWire and RMAP functionality in orbit.

1 SPACEWIRE-BASED DATA ACQUISITION FRAMEWORK

1.1 BUILDING BLOCKS

As shown in Figure 1, our data acquisition framework consists of four principal components.

Scientific instruments - output scientific data as analog and digital signals,

Front-end SpaceWire boards - receive the data and stores them into SDRAM. A typical SpaceWire board has two FPGAs on it; one is dedicated for SpaceWire/RMAP IP core, and users can implement instrument-dependent hardware logic to the other one. The SDRAM and the users FPGA can be accessed via RMAP.

SpaceWire router - connects multiple SpaceWire links constituting a network.

Read-out control computer - executes users' read-out program to transfer the data from the boards via SpaceWire/RMAP. The data can be stored on the computer itself, or on a remote PC.

In laboratory experiments (or non-flight experiments), products by Shimafuji Electric, presented in Figure 2, are utilized as these components. Shimafuji's 6 Port Router, whose size is almost the same as that of SpaceCube, became available recently. In some experiments, an 8-port router from NEC Corp has been used. Footprints of these products are fairy small compared to conventional crates and modules of VME or CAMAC systems, and SpaceWire has network capability not just a data bus. These greatly contribute to make the experimental setup compact and logically well structured. The setup is also highly scalable since we can increase the number of instruments and attached front-end circuit boards by simply introducing new routers.

The link speed of SpaceWire interfaces of the front-end SpaceWire boards and SpaceCube computer is variable, and usually we operate them at 100 MHz.

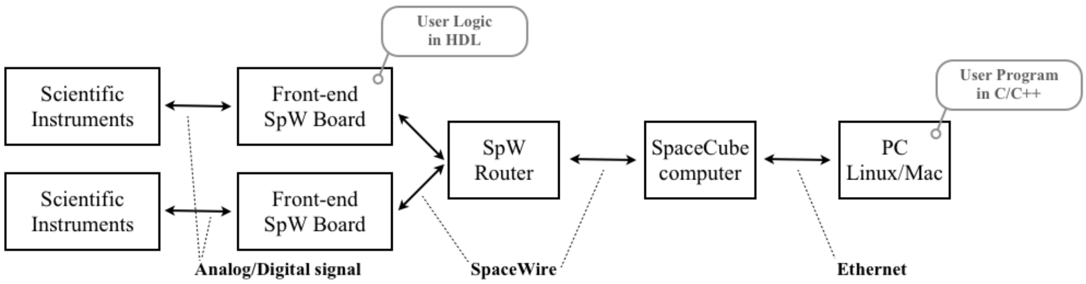

Figure 1 : Components used in the present SpaceWire-base data acquisition framework. User defined logic and program can be implemented on front-end boards and SpaceCube computer, respectively.

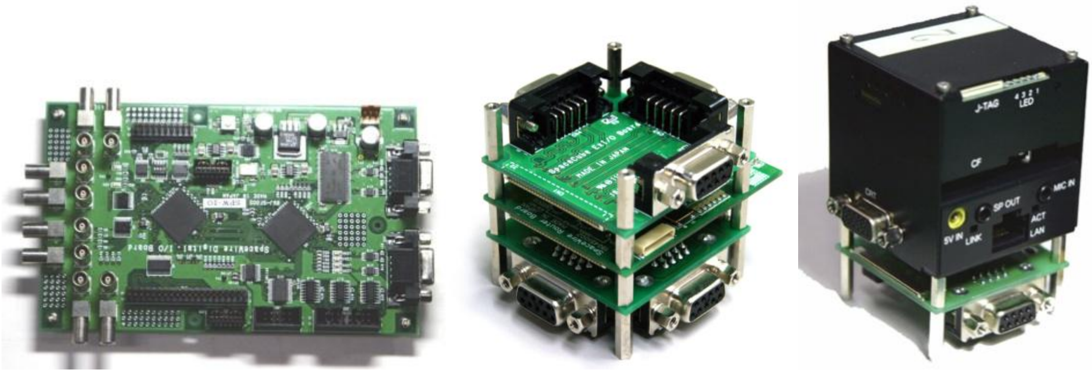

Figure 2 : An example of a front-end SpaceWire board, SpaceWire Digital IO (left), SpaceWire 6 Port Router (center), and SpaceCube computer (right) by Shimafuji Electric. For specs, see e.g. [1].

1.2 PROVIDED SOFTWARES AND THEIR FEATURES

An RMAP protocl stack and related class libraries for user programs on SpaceCube is available in C++ language (SpaceWire/RMAP Library [2]). They are designed to be portable, i.e. independent from hardware types and operating systems, and implementations of the system dependent layers for TRON-base realtime operating system (SpaceCube) and POSIX (Linux and Macintosh) are included. Therefore, once a user writes a program along with the library, its source code can be used to produce binaries for both of SpaceCube and ordinary PCs by just re-compiling.

If there is no SpaceWire interface available on an ordinary PC, SpaceCube can be a virtual SpaceWire interface via TCP/IP (Ethernet) as presented in Figure 3. A protocol converter provided with the library should be started on SpaceCube, and it seamlessly transmits SpaceWire packets to a user program running on a PC over TCP/IP connections. The basic idea is describe in detail in [3] and references therein. With this protocol converter, a development and debugging of a user program become easier since most of work can be done on a PC. Therefore, if an additional PC (its size and cost) is acceptable in an experiment, it is worth considering to execute a user program on a PC not on SpaceCube. Since this is a software protocol converter, information of the character level cannot be reconstructed; e.g. NULL does not appear in a TCP/IP end.

We have also been providing a template VHDL files for a hardware logic for an FPGA on a front-end circuit board [1]. The template includes several typical functionalities, needed to construct instrument-dependent logics, such as an on-chip data bus, a bus adapter, and a skeleton for registers and user original modules. RMAP accesses to a front-end board are translated into local bus access to the FPGA. The FPGA, or user logic on it, can respond to an RMAP Read and Write accesses by returning data to the local bus and by updating register value with data passed via the local bus, respectively.

From 2008, the source code of the SpaceWire IP core and the RMAP IP core used in Shimafuji's products is publicly released via their web site so that users can implement it into their own circuits.

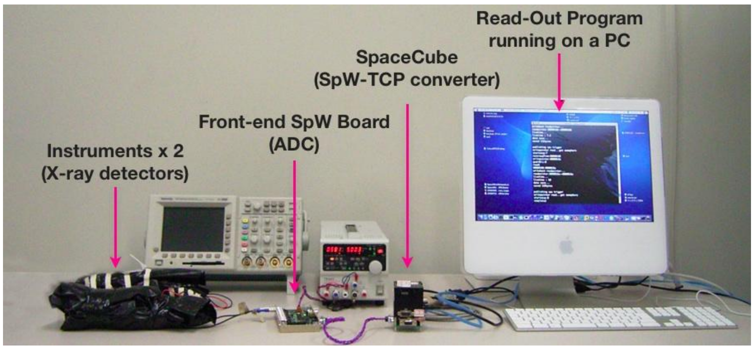

Figure 3 : An example experiment which utilizes SpaceWire-to-TCP/IP converter on SpaceCube. A user program runs on a PC, and transfers SpaceWire packets via a SpaceWire interface of SpaceCube.

1.3 PERFORMANCE

When a link is operated at 100 MHz, SpaceCube transfers SpaceWire packets at ~30 Mbps without any packet processing. If the software RMAP implementation is used in a user program on SpaceCube, data transfer rate drops to ~8 Mbps because of emerging CPU load. The SpaceWire-to-TCP/IP converter does not interpret packet content but just transfers them to/from a SpaceWire interface and a TCP socket. Therefore, in this case, the transfer speed is limited by the maximum transfer rate, ~10 Mbps, of a TCP protocol stack used in SpaceCube over its 100 Mbps Ethernet interface.

2 PRACTICAL EXAMPLES

2.1 APPLICATIONS IN GROUND EXPERIMTNS

We have been utilizing this data acquisition framework in many real experiments. Since our background is X-ray and Gamma-ray astrophysics, examples include development and test of a new X-ray microcalorimeter by JAXA, a balloon borne hard X-ray mission by Hiroshima and PoGO experiment group, and other radioactive measurements. As presented by [4,5], the framework has been put into commission in developments of scientific instruments for the Japanese next X-ray observatory satellite, ASTRO-H, which is scheduled to be launched in 2014.

2.2 TEST IN ORBIT

Japanese Small Demonstration Satellite 1 which has onboard SpaceWire Interface test Module (SWIM) together with other test components, was launched in January 2009. SWIM consists of two sub-modules flight model of SpaceCube computer (SpaceCube2 by NEC and JAXA, [6]) and a scientific instrument module SWIMμν with SpaceWire interfaces (Mitsubishi Heavy Industrial Corp, University of Tokyo, and JAXA). A subset of the RMAP protocol stack provided in the class library was implemented on SpaceCube2, and a user program developed using the protocol stack successfully performed scientific data transfer between SpaceCube2 (RMAP master) and SWIMμν module (RMAP target) in orbit. This proved that the concept of the SpaceWire data acquisition framework works well even in a flight model of a satellite.

3 FUTURE WORK

To improve the transfer speed of the SpaceWire-to-TCP/IP converter, we may have to use GbEther and a TCP/IP stack implemented on a dedicated hardware, e.g. ZestET1 by Orange Tree Tech, so as to increase the bandwidth and to reduce the CPU load for TCP. Since the development of ASTRO-H instruments are becoming very active, it is useful for the class library to support breadboard model and flight model of onboard computers.

4 REFFERENCES

1. Yuasa et al., “Development of a SpW/RMAP-based Data Acquisition Framework for Scientific Detector Applications”, SpaceWire Conference, Dundee, September 2007
2. Yuasa et al., ”A Portable SpaceWire/RMAP Class Library for Scientific Detector Read Out Systems”, SpaceWire Conference, Nara, Japan, November 2008
3. S. Mills and S. Parks, “The SpaceWire internet tunnel and the advantages it provides for spacecraft”, SpaceWire Conference, Nara, Japan, November 2008
4. T. Kouzu et al., “Verification of high resolution timing system with SpaceWire network onboard ASTRO-H”, this conference
5. T. Fujinaga et al., “Development of SpaceWire based data acquisition system for the X-ray CCD camera on board ASTRO-H”, this conference
6. T. Takahashi et al., “SpaceCube 2 -- An Onboard Computer Based on SpaceCube Architecture”, SpaceWire Conference, Dundee, September 2007