

A PORTABLE SPACEWIRE/RMAP CLASS LIBRARY FOR SCIENTIFIC DETECTOR READ OUT SYSTEMS

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

Masaharu Nomachi,

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

Motohide Kokubun, Hirokazu Odaka, Takeshi Takashima, Tadayuki Takahashi

*Department of High Energy Astrophysics, Institute of Space and Astronautical Science (ISAS), Japan
Aerospace Exploration Agency (JAXA), 3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510 Japan*

*E-mail: yuasa@amalthea.phys.s.u-tokyo.ac.jp, kokuyama@granite.phys.s.u-tokyo.ac.jp,
maxima@phys.s.u-tokyo.ac.jp, nakazawa@phys.s.u-tokyo.ac.jp,
nomachi@fn.lns.sci.osaka-u.ac.jp, kokubun@astro.isas.jaxa.jp, odaka@astro.isas.jaxa.jp,
ttakeshi@stp.isas.jaxa.jp, takahasi@astro.isas.jaxa.jp*

ABSTRACT

We developed a C++ class library which provides a unified method to transfer data via SpaceWire and RMAP. It is designed to be highly portable so that users can execute their products based on this library on both POSIX (eg. Linux or MacOS X) and TRON environments. TRON is a real-time operating system that is widely adopted in embedded computers, and will also be used on processors, called SpaceCube, to be onboard Japanese scientific satellites.

An encapsulation of SpaceWire hardware interfaces with an abstract SpaceWireIF class is one of important features of this library. Users develop their programs initially on, for example, accustomed Linux computers using a certain SpaceWire interface. After testing and debugging the program, they can be easily converted to an onboard computer, which has another SpaceWire implementation, without changing the code except for the instantiation of SpaceWireIF child classes. In addition to SpaceWire functionality, this library also provides encapsulating abstract classes for two major functionalities needed by detector read out program; multi threading and IP networking. These classes compensate the differences of system function calls between POSIX and TRON, and simplify the development of read out programs.

The functionality of this library was examined on a Linux computer and SpaceCube. In both these environments, we successfully executed the same sample C++ source codes which use SpaceWire interface, multi threading, and IP networking. The library have already used in the development of detector read out system for the next Japanese X-ray astronomical satellite, NeXT. We will report on applications to an actual laboratory experiment.