Real-Time Services in Networked Embedded Systems

L. Koblyakova, Yu. Sheynin, D. Raszhivin
St.Petersburg University of Aerospace Instrumentation
St.Petersburg, Russia
luda_o@rambler.ru, yuriy-sheynin@yandex.ru, dmitry.raszhivin@guap.ru

Abstract

Distributed Interrupt mechanism has been proposed for next SpaceWire standard release. According to specification Interrupt codes and Interrupt_Acknowledge codes are low-latency signalling codes and their distribution does not depend on data flow, that makes it useful for real-time distributed systems interconnections. The distributed interrupts service provides applications ability to receive and to send distributed interrupt codes. Control codes, received from SpaceWire network, would be transmitted to user applications as real time signals, using standard POSIX Real time signals mechanism.

SpaceWire standard includes Time Codes mechanism, it was designed for the implementation of the time distribution service. Using time codes the time access service was developed, which provides a consistent application interface to a local time source that is correlated to some centrally maintained master onboard time source. Described services were developed for Linux OS, which includes patches for running in soft real time.

INDEX TERMS: SPACEWIRE, LINUX, REALTIME.

I. SERVICES ARCHITECTURE

Linux offers embedded designers an inherently modular operating system that can be easily scaled down to compact configurations suitable for embedded design. Plus, Linux is the fastest growing server operating system and is rapidly moving into embedded applications.

For chips manufactured by ELVEES RnD Center with built-in SpaceWire [1] channels has been developed software to work effectively in the OS Linux [3] environment:

- Drivers for SpaceWire channel controllers, which allow the use of SpaceWire channels as usual network devices. Each channel is represented by its network interface with IP address, so all TCP/IP applications, using BSD POSIX sockets API, would work over SpaceWire without any change.
- For use in distributed and parallel systems, implemented a Time Access Service (TAS), that provides applications with a consistent interface to a local time source that is correlated to some centrally maintained master onboard time source. The time values provided by this service might typically be used by the application to schedule some operation, such as the acquisition of an image or to time stamp locally generated telemetry data.
- Distributed interrupts service is a service, meant for operating with SpaceWire distributed interrupts [2]. Interface for applications is a standard POSIX real-time signals interface.

II. NETWORK SERVICES

Each SpaceWire channel is represented as linux network device with its own IP address, which supports data transmission over TCP/IP. User applications are provided with standard POSIX socket interface, so a lot of network applications can be used over SpaceWire without

any change. Among them http, ftp, telnet clients and servers, and a wide range of standard utilities for network configuration and diagnostics, as ifconfig, route, ping, nuttcp etc.

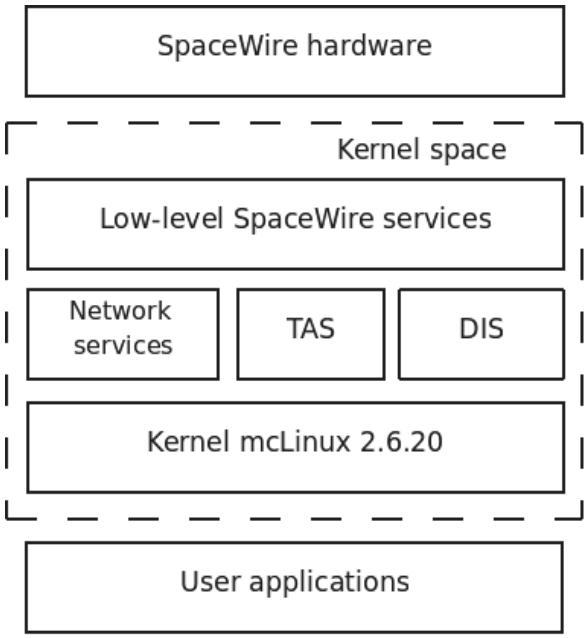

Fig. 1: OS Linux architecture with SpaceWire services

III. Distributed Interrupts Service

Interrupt-Code represents a system signal request. It is issued by a node link that will be considered as the source node for this interrupt (Interrupt Source). It is distributed over the network to all other nodes. An Interrupt-Code should be accepted for handling in some node of the SpaceWire network, which will be called the Interrupt Handler. The host of the node is supposed to implement some interrupt processing routine. One of 32 interrupt request signals (interrupt source identifiers) could be identified by the Interrupt-Code.

Interrupt_Acknowledge-Code represents a confirmation that the Interrupt-Code has reached some Interrupt Handler and has been accepted by it for processing. The Interrupt Handler node should send an Interrupt_Acknowledge-Code with the same five-bit interrupt source identifier as in the accepted Interrupt Code. The Interrupt-Code is broadcasted to find an Interrupt Handler node.

A signal is a limited form of inter-process communication used in POSIX-compliant operating systems (as Linux). Essentially it is an asynchronous notification sent to a process in order to notify it of an event that occurred. When a signal is sent to a process, the operating system interrupts the process's normal flow of execution. Execution can be interrupted during any non-atomic instruction. If the process has previously registered a signal handler, that routine is executed.

The PASC Real-time System Services Working Group (SSWG-RT) has developed a series of standards that amend IEEE Std 1003.1-1990 and a profile standard (IEEE Std 1003.13-1998). The Real-time amendments to IEEE Std 1003.1-1990 is IEEE Std 1003.1b-1993 Realtime Extension. According to this standard, Linux support 32 real-time signals, ranging from SIGRTMIN to SIGRTMAX, that can be used for application-defined purposes.

Distributed interrupts service uses one real-time signal to inform user applications about received interrupts. Applications have to register at service and define interrupt handler to obtain required signal.

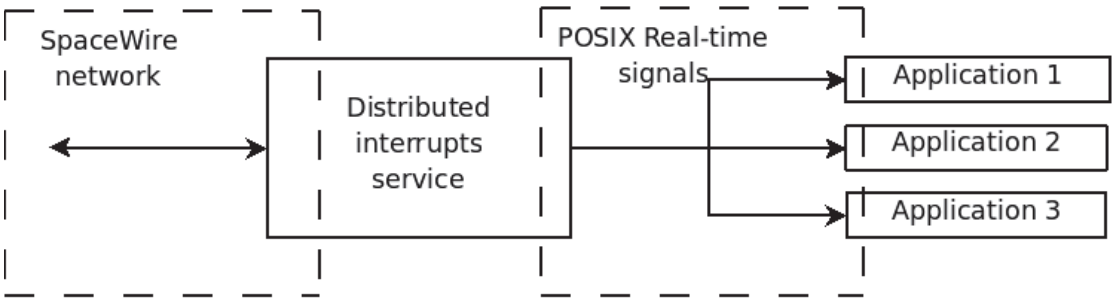

Fig. 2: Distributed interrupts service architecture

IV. Time Access Service

Organization “The Consultative Committee for Space Data System” (CCSDS) develops standards for space systems. They proposed a draft standard SOIS version CCSDS 872.0-R-0.3, which defines the requirements for the network subsystem, presented in the form of services over the network driver.

One of proposed services is time access service (872.0-R-0.3). This service allows many hosts to work with the same time. This achieves synchronous and planned execution of programs, which is important for on-board real-time systems.

The SOIS Time Access Service provides applications with a consistent interface to a local time source that is correlated to some centrally maintained master onboard time source. The time values provided by this service might typically be used by the application to schedule some operation, such as the acquisition of an image or to time stamp locally generated telemetry data.

The standard SpaceWire does not suggest a mechanism of time synchronization, however, there is a tool for the implementation of the time distribution service - time codes. The work reviewed existing algorithms of time distribution, and unified time service was realized, according to developed algorithm. It corresponds to the standard CCSDS SIOS «Time access service», and uses SpaceWire times codes for time distribution.

A typical architectural scenario is shown in Fig. 3. An onboard time system architecture consists of local and master onboard time sources implemented in hardware.

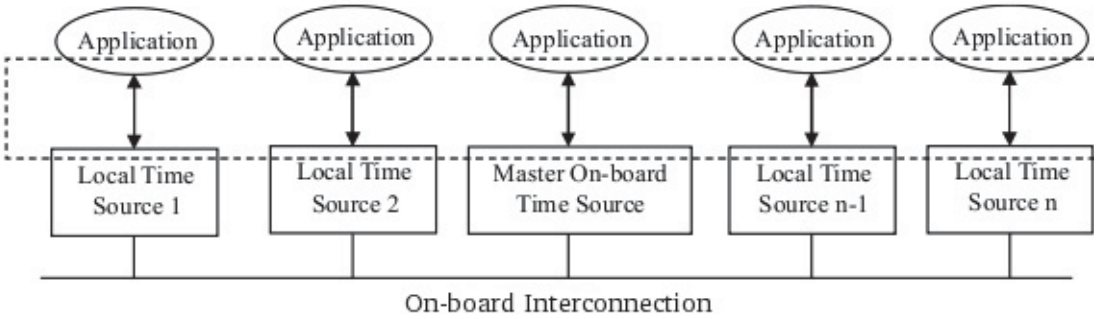

Fig. 3: Typical Onboard Time System Architecture

REFERENCES

1. ECSS-E-50-12A “SpaceWire - Links, nodes, routers and networks”, European Cooperation for Space Standardization (ECSS), 2003, 124 p.
2. Liudmila Onishchenko, Artur Eganyan, Irina Lavrovskaya. Distributed interrupts mechanism verification and investigation by modelling on SDL and SystemC. International SpaceWire Conference, Nara 2008. Conference Proceedings. ISBN: 978-0-9557196-1-5
3. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers, Third Edition - O'Reilly Media, Inc., 2007