REAL-TIME SIGNALLING IN NETWORKED EMBEDDED SYSTEMS

Session: Networks and Protocols

Short Paper

Liudmila Koblyakova, Yuriy Sheynin, Dmitry Raszhivin

St.Petersburg University of Aerospace Instrumentation

67 B.Morskaya st., 190 000, St.Petersburg, Russia

E-mail: luda_o@rambler.ru, sheynin@aanet.ru, dmitry.raszhivin@guap.ru

ABSTRACT

The SpaceWire standard includes Time Codes that were designed for implementation of time distribution services. Using time codes the Time Access Service was developed. It provides a consistent application interface to a local time source that is maintained to be synchronised to the onboard time source master. Distributed Interrupt mechanism has been proposed for next SpaceWire standard release. 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 real-time signalling for applications in distributed architectures with SpaceWire interconnections. The received from SpaceWire network Interrupt codes would be transmitted to user applications as real time signals with the standard POSIX Real time signals mechanism. Described services were developed for Linux OS with patches for running in soft real time.

1 SERVICES ARCHITECTURE

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

For chips manufactured by ELVEES with built-in SpaceWire channels software has been developed to work with in the OS Linux environment:

- *Drivers* for SpaceWire channel controllers, which allow to use of SpaceWire channels (links) as regular network devices. Each channel is represented by its network interface with an IP address; so all TCP/IP applications that use BSD POSIX sockets API would work over SpaceWire interconnections without any change.

- The *Time Access Service* (TAS), that provides applications with a consistent interface to a local time source that is source that is maintained to be synchronised to the onboard time source master.. The time values provided by this service might typically be used by applications to schedule some operations, such as the acquisition of an image or to time stamp locally generated telemetry data.

- *Distributed Interrupts Service* (DIS) is a service for real-time signalling with SpaceWire distributed interrupts. Its software interface for applications is the standard POSIX real-time signals interface.

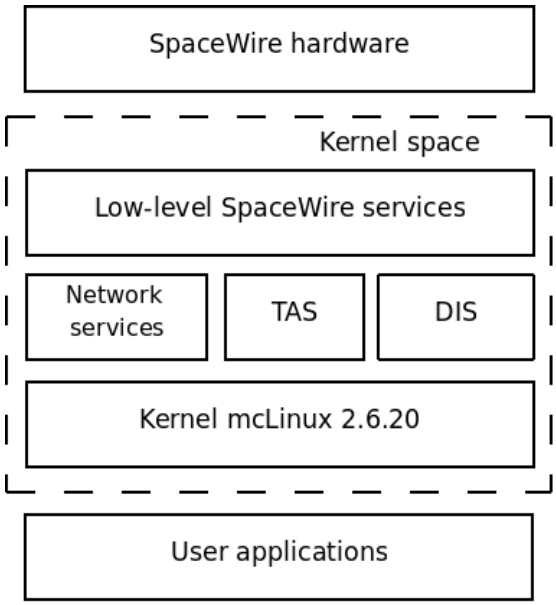

Figure 1: OS Linux architecture with SpaceWire services

2 NETWORK SERVICES

Each SpaceWire channel is represented by the Network Services that work over Low-level SpaceWire services (driver included) as a regular Linux network device with its own IP address; it supports data transmission over TCP/IP. User applications are provided with the standard POSIX socket interface, so a lot of network applications can be used over a SpaceWire interconnection without any change: http, ftp, telnet clients and servers, and a wide range of standard utilities for network configuration and diagnostics, as ifconfig, route, ping, nuttcp, etc.

3 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). The Interrupt-Code is broadcasted to find an Interrupt Handler node. 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.

A signal is a limited form of inter-process communication used in POSIX-compliant operating systems (Linux included). Essentially it is an asynchronous notification sent to a process in order to notify it of an event that has occurred. When a signal is sent to a process, the operating system interrupts the normal flow of program execution. Execution can be interrupted during any non-atomic instruction. If the process has been previously registered as the signal handler its 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 the profile standard (IEEE Std 1003.13-1998). The Real-time amendments to IEEE Std 1003.1-1990 is IEEE Std 1003.1b-1993 Real-time Extension. According to this standard, Linux support 32 real-time signals, ranging from SIGRTMIN to SIGRTMAX that can be used for application-defined purposes.

The Distributed Interrupts Service (DIS) uses a real-time signal to inform user applications about the interrupt or exception that has been set somewhere in the distributed system. Applications have to register at the DIS service and define the interrupt handler to receive the particular real-time signal.

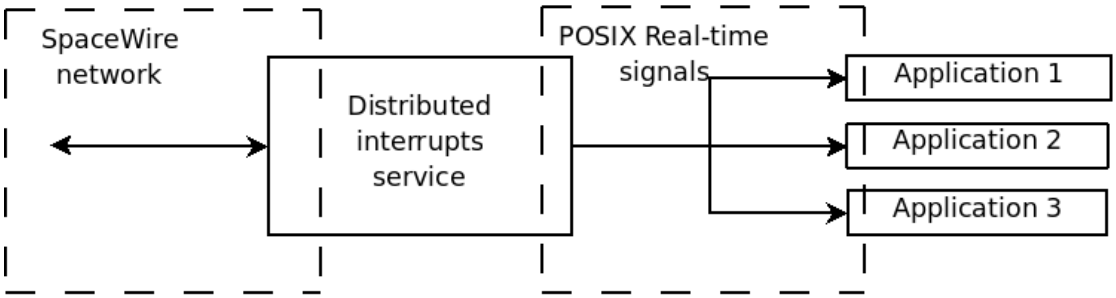

Figure 2: Distributed interrupts service architecture

4 TIME ACCESS SERVICE

The CCSDS (The Consultative Committee for Space Data System) develops standards for space systems. It proposed a draft standard SOIS version CCSDS 872.0-R-0.3, which defines the requirements for the network subsystem; it is specified in the form of services over the network.

One of the proposed services is time access service. This service allows many hosts to work with the same time. It supports synchronous and time scheduled execution of programs, which is important for onboard real-time systems.

The SOIS Time Access Service provides applications with a consistent application interface to the local time source that is maintained to be synchronised to the onboard time source master.. The time values provided by this service might typically be used

by the application to schedule some operations, such as the acquisition of an image or to time stamp locally generated telemetry data.

The SpaceWire standard has Time Codes – a tool that could be used for time distribution service implementation, though it does not suggest a ready-maid mechanism for local times synchronization in distributed systems.

In our research we have reviewed existing algorithms of time distribution, designed and realized the unified time service according to the developed algorithm. It corresponds to the standard CCSDS SIOS «Time access service», and uses SpaceWire times codes for time marks distribution.

A typical architectural scenario is shown at the Figure 3. The onboard time system architecture consists of local and master onboard time sources implemented in hardware.

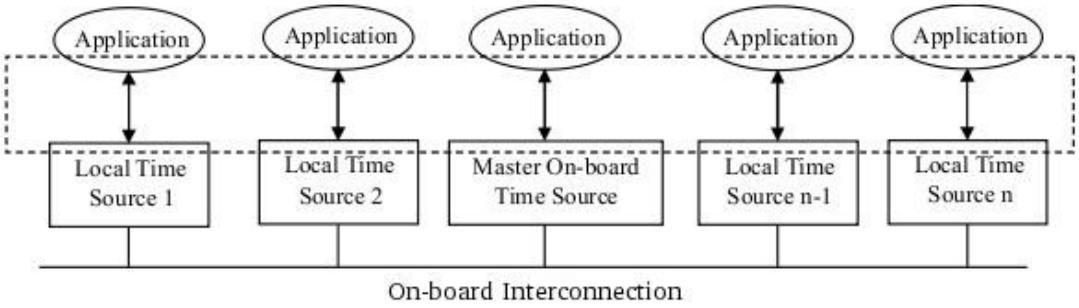

Figure 3: Typical Onboard Time System Architecture

REFERENCES

1. ECSS-E-50-12A “SpaceWire - Links, nodes, routers and networks”, European Cooperation for Space Standardization (ECSS), 2003, 124 p.

2. Sheynin Yu., Gorbatchev S., Onishchenko L., “Real-Time Signalling in SpaceWire Networks”. International SpaceWire Conference. Conference Proceedings. Space Technology Centre, University of Dundee, Dundee, 2007. ISBN: 978-0-9557196-0-8..

3. Onishchenko L., Eganyan A., Lavrovskaya I. 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

4. Corbet J., Rubini A., Kroah-Hartman G. Linux Device Drivers, Third Edition - O'Reilly Media, Inc., 2007.