THE ADAPTATION AND IMPLEMENTATION OF SPACEWIRE-RT FOR THE MARC PROJECT

Session: SpaceWire Networks and Protocols

Short Paper

Stuart Fowell, Patricia Lopez-Cueva

SciSys UK Ltd, Clothier Road, Bristol, BS4 5SS, UK

Alan Senior

Systems Engineering and Assessment (SEA) Ltd, Somerset, BA11 6TA, UK

Omar Emam

EADS Astrium, Stevenage, Hertfordshire, SG1 2AS, UK

Wahida Gasti

ESA/ESTEC, 2200 AG Noordwijk ZH, The Netherlands

E-mail: stuart.fowell@scisys.co.uk, patricia.lopez-cueva@scisys.co.uk, alan.senior@sea.co.uk, omar.emam@astrium.eads.net, wahida.gasti@esa.int

ABSTRACT

The Modular Architecture for Robust Computing (MARC) is an ESA GSTP mini-project being undertaken by SciSys, Astrium UK and SEA. MARC is developing a decentralised onboard computer using SpaceWire and SOIS as a communication backbone with a hierarchical FDIR mechanism.

The hardware architecture is closely coupled to the software aims of the Generic Fault-tolerant Software Architecture using SOIS (GenFAS) software framework, developed by SciSys. This provides a PUS-based Data Handling Services, communication functions using SOIS, FDIR management and a software deployment and upgrade mechanism.

A crucial part of the SOIS software stack for the MARC project is the provision of a suitable SpaceWire service guaranteeing timely delivery of data. To achieve this, SciSys applied the proposed SpaceWire-RT protocol. The protocol aims to ensure timeliness by utilising SpaceWire time-codes to divide the available network bandwidth in pre-allocated slots and specifies facilities for reliability and redundancy management.

This paper describes the application and implementation of the SpaceWire-RT protocol in the MARC project: a practical scenario, utilising representative flight hardware and next-generation network architectures. A relevant subset of the protocol was selected and implemented entirely in software and was adapted to communicate with hardware not specifically designed for a SpaceWire-RT system. Additionally, the context of a representative flight software system raised issues such as synchronisation which were not considered by SpaceWire-RT.

The paper closes by summarising the lessons for timely and reliable use of SpaceWire that can be drawn from this detailed project, considering the complete communications stack from subnetwork to application interface.