RMAP OVER SPACEWIRE ON THE EXOMARS ROVER FOR DIRECT MEMORY ACCESS BY INSTRUMENTS TO MASS MEMORY

Session: SpaceWire missions and applications

Short Paper

Bryan Dean, Rosalind Warren, Ben Boyes

EADS Astrium, Gunnels Wood Road, Stevenage, SG12AS, UK

E-mail: bryan.dean@astrium.eads.net, rosalind.warren@astrium.eads.net, ben.boyes@astrium.eads.net

ABSTRACT

The ExoMars Rover incorporates a number of high bandwidth instruments including cameras and microscopes which connect to the On Board Computer using SpaceWire technology. We intend to use the Remote Memory Access Protocol (RMAP) to allow the instruments to write large amounts of data to the mass memory within the computer without application software involvement. This will free up much needed processing time for the autonomous navigation algorithms.

Since the file system management tasks will be performed by the application software (ASW) on the processor, the mass memory controller will not have knowledge of the file or data structure in the logical address space. As a result, all write accesses to the mass memory must be controlled by the application software. We are developing a mechanism using RMAP to allow the software to initiate a transfer of data from a specified memory address in an instrument to a specified memory address in the mass memory. Once the data transfer has been initiated, the software can perform other tasks whilst waiting for acknowledgement of a successful transfer.

In order to achieve the high level of processing required for the Rover to navigate and drive autonomously on the surface of Mars, the initial processing of images used for navigation will be performed by dedicated hardware in the On Board Computer. The same mechanism for data transfer from camera to mass memory will be used to send image data directly to this hardware. In this paper we outline the details of this mechanism.

1 DATA HANDLING SYSTEM

The Rover Vehicle Data Handling System is designed with a centralised unit, the On Board Computer (OBC) connected to the instruments and other subsystems. The OBC provides the processing power, memory facilities and overall control. It interfaces to the science equipments and to other subsystems using either SpaceWire or CAN bus links. The use of CAN bus reduces harness size and power by connecting several nodes onto a single bus, at the expense of reduced bandwidth. The CAN bus is used for equipment with low bandwidth requirements but for the optical instruments and for the cameras which generate large data packets that are required promptly for navigation, high speed SpaceWire links are essential.

The processing power is provided through a combination of dedicated hardware in a co-processor and application software running on the processor. The dedicated hardware is used for certain repetitive image processing functions, in order to speed up navigation algorithm execution. The processor provides a platform for software which is more flexible and can be used according to the current tasks being performed by the Rover, for example to execute path-planning algorithms or to compress science data. CPU load is minimised by sending science data directly to the mass memory where possible, without application software involvement.

Within the OBC there are SpaceWire nodes on the mass memory, on the processor and on the hardware image processing within the co-processor. Each instrument with a SpaceWire interface is connected directly to the OBC by a point-to-point link (in some cases cold redundant). SpaceWire routing between the nodes is implemented within the OBC. The SpaceWire links to the cameras are cold redundant; the OBC has the responsibility to select the operational link. The links to the CLUPI and MicrOmega science instruments are single links, with cross-strapping within the OBC.

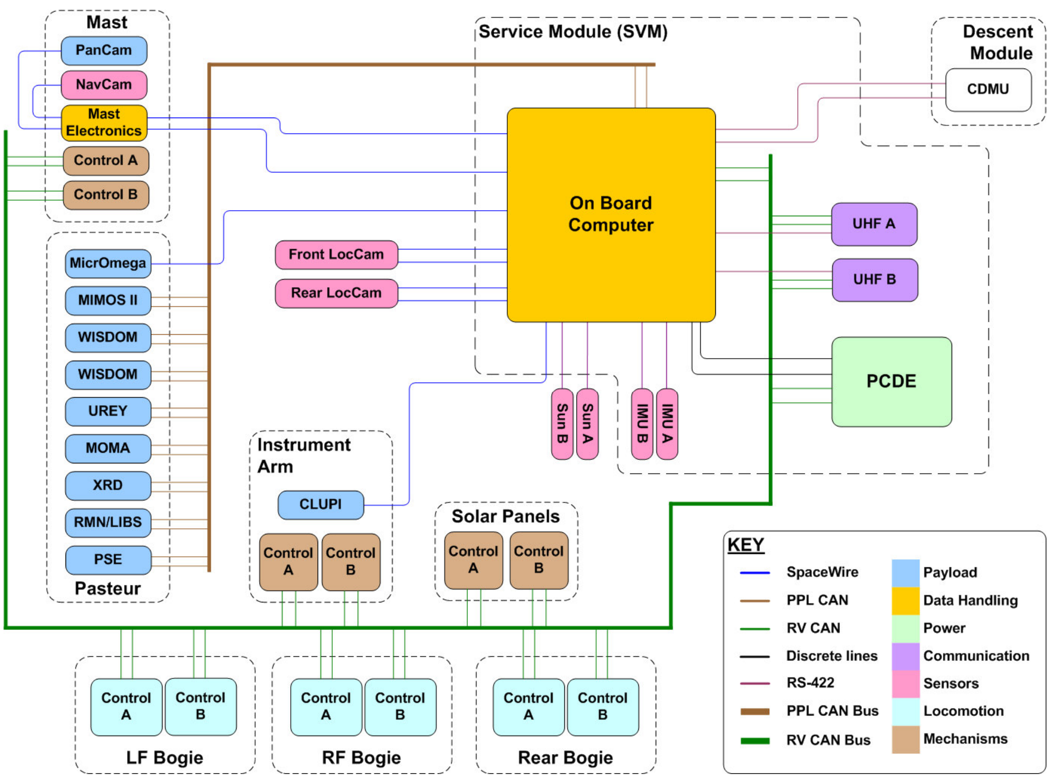

Figure 1: Rover Vehicle Data Handling System

2 DATA TO CO-PROCESSOR

The navigation cameras acquire a pair of stereo images which are used to generate a digital elevation model (DEM) of the surrounding terrain. This DEM is then used to generate a navigation map and plan a path to the specified destination. The Rover will stop every couple of meters to acquire more navigation images and update the navigation map. It is essential that the image processing time during these stops is minimised to enable the Rover to travel a reasonable distance in a Martian day or Sol.

In addition the localisation cameras acquire images whilst driving and the OBC identifies features which are used to track the progress along the planned path and avoid collisions. The speed at which the Rover can drive is limited by the time taken to process the images.

The dedicated hardware in the co-processor reduces this processing time considerably but requires that the images can be sent directly from the cameras to the dedicated hardware (not via the processor). SpaceWire routing allows images from the navigation cameras, the front localisation cameras or the rear localisation cameras to be sent directly to the dedicated image processing hardware in the co-processor. RMAP is used on the Rover to enable the processor to initiate a download of the images directly to the co-processor.

3 DATA TO MASS MEMORY

The memory for science and housekeeping data storage is provided by a mass memory function with memory controller. The majority of the memory is non-volatile but a small volatile area is provided for temporary storage, for example of science data prior to compression. Science data can be written directly to the mass memory, using RMAP for data arriving over the SpaceWire link. The memory controller includes logical to physical address mapping as well as error correction.

The mass memory is essential to provide immediate storage of science data for later processing by software and to provide storage of processed scientific data between the limited downlink windows. The memory is accessed by the application software using logical addressing.

Data received on the SpaceWire interface can be written directly to the volatile part of the mass memory without software intervention. RMAP is used to provide the memory addressing. The ASW enables and disables the access to the mass memory for each individual instrument, identified by the source address in the header of the SpaceWire packet.

Following storage in the volatile memory, the data can be processed by the ASW when there is available CPU capacity. Once this data is in the format required for transmission to Ground, the ASW writes the data to the file system in the non-volatile memory to wait for availability of return-link bandwidth.

4 RMAP OVER SPACEWIRE

Since RMAP is used to write data from the ASW to the mass memory, RMAP is also used to write commands to the instruments and to receive data back. All commands initiated by the OBC are initiated by the processor but responses may be required to be sent to the processor, the hardware image processing in the co-processor or the mass memory. When the processor requests data to be written to the mass memory, it supplies the RMAP header in a write command sent to the instrument, to enable the instrument to reply directly to the mass memory. The instrument then initiates a RMAP write command to send the data to the mass memory or co-processor using the RMAP header provided by the processor.

Each acquisition command to an instrument results in a write command from the instrument back to the OBC. For simplicity, the first 12 bytes of the RMAP header for the write command back from the instrument are provided in the cargo of the write command to the instrument. The instrument does not need to check all the contents of these 12 bytes, nor use the contents of all of these bytes instead of values known locally (e.g. Response Protocol Identifier). This is provided simply to allow for flexibility of implementation in the instrument. The transaction identifiers used will differ depending on the choice made.

OBC Command byte	Response Destination Logical Address	Response Protocol Identifier	Response Packet Type / Command / Source Path Addr Len
Response Destination Key	Response Source Logical Address	Response Transaction Identifier (MS)	Response Transaction Identifier (LS)
Response Extended Write Address	Response Write Address (MS)	Response Write Address	Response Write Address
Response Write Address (LS)	Data CRC	EOP	
	Last byte transmitted		

Figure 2: Cargo - Instrument Acquisition Write Command

If the data is required to be returned directly to the processor, a simple RMAP read command to the instrument will suffice, although additional write commands are required to trigger the data acquisition.

The exchange is illustrated Figure 3 from the point of view of a single instrument, showing a RMAP write to the instrument requesting a further RMAP write to mass memory.

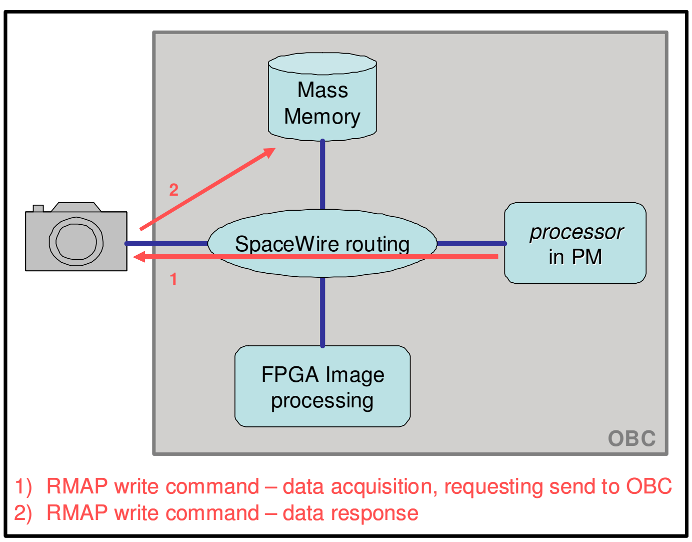

Figure 3: Data acquisition across SpW from an instrument to the mass memory