


The SpaceWire-PnP Protocol: Status and Future Directions

University of Dundee School of Computing

Peter Mendham

Space Technology Centre University of Dundee

17 September 2009

Agenda

- The plug-and-play concept
- A little history
- Key concepts and terminology
- SpaceWire-PnP services
- One service in detail
- Capability services: data sources and sinks
- Summary
- The way forward from here

Plug-and-Play

- Historically related to user experience
  - The ability to interface two or more devices together without the need for configuration
- Refers to
  - Discovery
  - Configuration
  - (Adaptation)
- of
  - Devices
  - Services

“Devices” could refer to anything from ICs to units to software components to whole systems

Plug-and-Play for Space

- Apply the properties of plug-and-play to space systems
- Central goal is **interoperability**
- Key feature is **discovery**
- Improve:
  - Reuse
  - Development cycles
  - Market for COTS components

Plug-and-Play Scenarios

- Accelerating development
- Automated integration and test
- Failure detection/fault tolerance
- Onboard mode change verification
- Agile spacecraft development
  - Such as ORS

A Little History

- Recent developments (since 2006) spurred on by ORS
  - AFRL, NRL, GSFC
- Contributions from ESA, 4Links, UoD
- New protocol proposed meeting the needs of ORS
  - Implemented and utilised in SPA-S
- UoD propose the use of RMAP packet format to permit reuse of existing IP (proposed late 2006 early 2007)
  - Fitted with existing semantics
- Other drivers for UoD proposals:
  - Cover cases other than ORS; fix bugs; ensure wider applicability; provide mechanisms for using existing hardware and software

Plug-and-Play and Standardisation

- Spectrum of possible approaches to devices:

No standardisation ⟷ Complete standardisation

- Complete standardisation
  - All devices must support a standard interface
  - Device interface must cover all possible device features
  - Standard device driver
  - Could only be qualified once
    - Assumes all device accesses come from software

- No device standardisation
  - Requires no changes to existing devices
  - Devices described by standard data sheet
  - Available in electronic form
  - Permits auto-configuring of device drivers
  - Permits use of devices by standard hardware
  - Software must be re-qualified every time
  - Hardware must be rewritten each time

SpaceWire-PnP: Guiding Principles

- UoD working document: SpaceWire-PnP
- Provide a standard way to **discover** and **configure** the standard features of SpaceWire devices
  - i.e. the features of SpaceWire devices which are identified in the SpaceWire standard
- As interoperable as possible
- Do not *require* any extra SpaceWire features
- Provide hooks for service discovery and configuration
  - But do not implement this: beyond scope
- Consider the application to common use cases

SpaceWire-PnP and Standardisation

- Spectrum of possible approaches to devices:

- Standardisation of standard features
  - Requires devices to support SpaceWire-PnP protocol
  - Documented exceptions for existing devices
  - Standardised mechanisms for discovery
  - Standardised mechanisms for configuring standard features only
  - Provides extensibility but does not require extra features
  - Partners well with electronic data sheets

RMAP Utilisation

- Semantics required for plug-and-play closely match RMAP
- Use a well-defined implementation of RMAP
  - 32-bit wide addressing and alignment
  - Big endian
  - Incrementing addressing
  - Acknowledged, verified writes
  - Pre-defined key
  - RMW implementation (optional) is a conditional write
- Use a different protocol ID
  - To distinguish from generic RMAP traffic
  - E.g. Mass memory device

scisys

Perspective

- PnP views the network like the SpaceWire standard
  - Links
  - Nodes } Devices
  - Routers } Devices
- No topology restrictions
- Both nodes and routers have links
  - Nodes have 1 or more links
  - Routers have 2 or more links
- Every device on the network has a port zero
  - This is the target for PnP transactions

scisys

Levels of Support

- Managed Networks
  - Important role for system designer
  - Competition during discovery process removed by design
  - Competition for configuration of devices removed by design
  - Simplest case

Level 1

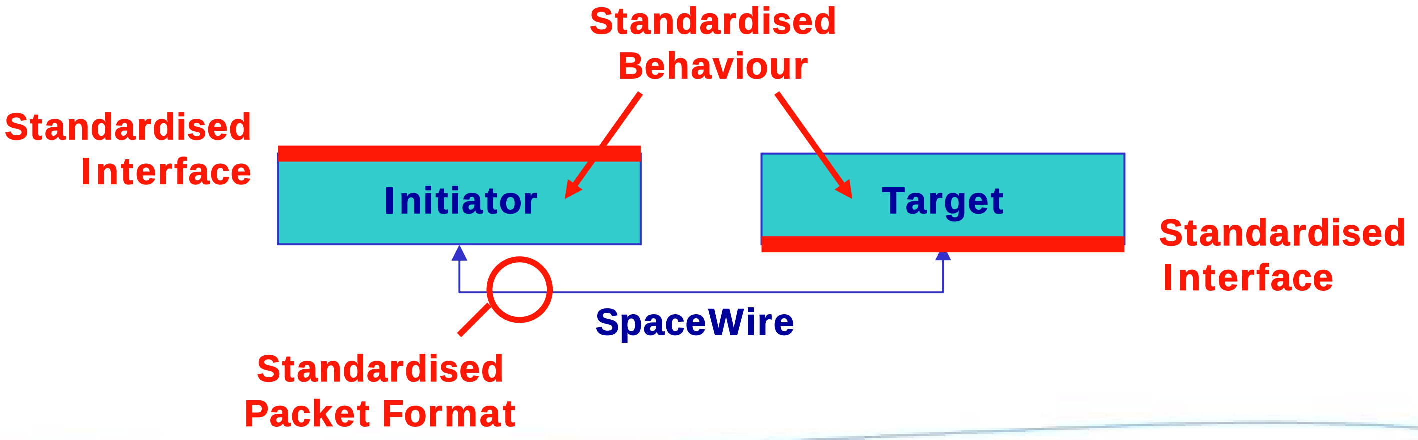
- Open Networks
  - Network handles all competition issues
  - Deals with networks where design is **not** known *a priori*
  - More flexible but more complicated

Level 2

scisys

Services

- A set of parameters on the target
  - This is a standardised RMAP address space
- A service interface at the initiator
- A description of how the initiator and target will both behave



Target Parameters

- Follow a regular form
- Parameters are made up of *fields*
  - Each field is 32-bit
- Optionally, a parameter may have multiple *entries*
  - This is to permit tables, such as routing tables
  - The *root entry* has one set of fields
  - Every other *non-root entry* has a different but identical set of fields
- For example, the port configuration parameter
  - Has a root entry with one field giving the number of ports
  - Has a non-root entry for each port, each of which has the same fields

Core Services

- Four core services defined
  - Device Identification
  - Network Management
  - Link Configuration
  - Router Configuration (routers only)
- Optionally, there is also a time-code source

Device Identification and Status

- Identifies the device
  - Vendor ID and Product ID (like PCI, USB etc.)
  - Type (node/router)
  - Number of ports
  - Optional static device ID
  - Optional vendor and product strings
- Provides current status
  - Active ports
  - Device ID (non-static)
  - Return port

scisys

Example Parameter Fields

Table 5-3: Device Information Parameter Fields		
ID	**Name**	**Summary**
0	Vendor ID/ Product ID	Contains 16-bit vendor and product IDs
1	Region/Number of Ports	Indicates preferred device region gives port count
2	Static Device ID High	High 32 bits of the 64-bit static device ID (if present)
3	Static Device ID Low	Low 32 bits of the 64-bit static device ID (if present)
4	Version/Instance ID	Version and System instance of this device type
5	Operation/String Lengths	Length of the vendor a product strings (can be zero)
6-31	*Reserved*	*Reserved for future use*

Example Initiator Primitive

- DIDS_READ_INFO.request
  - RMAP_Parameters
- DIDS_READ_INFO.indication
  - Result
  - Vendor_ID
  - Product_ID
  - Preferred_Region
  - Router_Node
  - Support_Level
  - Port_Count
  - Device_ID
  - Version
  - Instance_ID

scisys

Capabilities

- Device can provide a list of *capabilities*
- Capabilities based on protocol ID
  - A protocol which is supported
  - Optionally transported over another protocol
  - Supports nesting of transports
- Each capability can define a service
  - Just like the core services
  - Defines target parameters, initiator primitives etc.
  - Flexible and extensible
- An example protocol would be RMAP
  - Predefined capability services to permit use of RMAP
  - Data source service and data sink service

scisys

Data Source and Sink Capability Services

- Permit description of existing RMAP address spaces
- Utilise previously documented RMAP mechanisms
  - Such as delayed response read
- Also provides an interface to an initiator
  - Permits configuration of an initiator
- Each source/sink defines its data type
  - A few standard ones, most left open
- Provides a great deal of flexibility

scisys

Uses for the Data Source

- Electronic data sheets
  - Standard type for (e.g. xTEDS) data sheets
  - Describes where to read for the data sheet
  - Responds immediately
- Router status change notification
  - Standard type for router status
  - Either delayed response read
  - Or initiated write
  - Both completely configurable

Summarising SpaceWire-PnP

- Protocol utilising packet format and semantics of RMAP
- Defines
  - Target parameters
  - Initiator primitives (service interface)
  - Behaviours (algorithms) where necessary
- Simple
  - Does not require extra feature support
- Flexible and extensible
  - Can use capability services to extend support

scisys

The Way Forward

- This is one possible approach
  - Has involved complete documentation, research and simulation of key principles
- Various decisions were made
  - Including trade-offs
- Each decision should be considered in turn by a group of people
  - This can be used to guide future developments

Some Decisions

- Is plug-and-play a useful concept for SpaceWire?
  - I assume "yes"
- Core objectives of plug-and-play for SpaceWire?
  - Network discovery
  - Device configuration?
  - Interoperability? (how much?)
- How much should be standardised?
  - Nothing?
  - More than I have suggested?
  - Less?

Technical Specifics

- Should we use RMAP?
  - To what extent?
  - What about the use of protocol IDs?
- Is it OK for a leading zero to indicate discovery/configuration information?
  - I.e. every device has a port zero
- Vendor IDs etc.
  - How standardised?
  - Controlled by SpaceWire WG?

Questions?
Discussion?