16031517

University of the West of England

BEng Individual Project

The Implementation of ECSS SpaceWire into the IEEE 802.3 Power over Ethernet Protocol

A report submitted in fulfilment of the requirements for the degree

of Electronic Engineering BEng (Hons)

in the

Engineering Design and Mathematics Department

Faculty of Environment and Technology

&

Support of Thales Alenia Space UK for Project MOSAR

Author: Jack Martin

Supervisor: Yaseen Zaidi

Words: 10634

Pages: 56

Abstract

Engineering Design and Mathematics Department
Faculty of Environment and Technology

Electronic Engineering BEng (Hons)

The Implementation of ECSS SpaceWire into the IEEE 802.3 Power over Ethernet Protocol

By Jack Martin

The operational capability and sustainability of Spacecraft is constrained by their physical system architecture, highly customised monolithic design and the limited or no availability of servicing and maintenance. The research of cost effective, high performing, scalable and flexible solutions is essential to the future of the European Space Agency to spearhead the commercial space industry strategy. Project MOSAR is an EU funded initiative that will aim to raise the degree of modularity of space systems by an order of magnitude with respect to current space industry standards.

The modularisation of space mission systems can be influenced by the use of existing and widely adopted technologies for key functionalities of mission equipment. This report details the feasibility investigation of the implementation of ECSS SpaceWire into the IEEE 802.3 Power over Ethernet protocol, to enable the transmission of data and power between spacecraft sub-systems via a singular architecture. A Literature Review was conducted of the SpaceWire and IEEE standards to ascertain the technical requirements of each protocol; further research of available literature from the space community of interest was conducted to understand the current methods and issues identified for the implementation of a combined full protocol stack.

A proposed full protocol stack was implemented and deployed to a SpaceWire Power over Ethernet (PoE) system, which was subjected to functional testing and theoretical analysis to determine its feasibility and validity for use within the space industry, in support of the aims of Project MOSAR. The SpaceWire PoE system was concluded to be fully compliant to the identified requirements of a combined SpaceWire and Ethernet Protocol Stack. The theoretical analysis identified issues relating to the efficiency of transmission and the degradation of performance in contrast to solely using a SpaceWire Network Architecture. It is recommended that a wider study is envisioned within the space industry community to ascertain whether the use of a SpaceWire PoE Architecture is valid in the context of efficiency savings throughout a spacecrafts lifecycle.

Acknowledgments

I would like to thank my project supervisor Yaseen Zaidi for allowing me the opportunity to conduct this project as apart of my Individual Project Module, and for all the support and guidance he has offered me throughout the duration of my final year in UWE.

I would also like to offer thanks to Thales Alenia Space UK for also offering their encouragement and feedback throughout the duration of the project. The positive feedback I received from the findings presentation has given me confidence in my abilities as a Systems and Electronics Engineer that I will continue to champion throughout my career.

Finally I would like to offer thanks to my co-students Chris, Joshua and Lee for the collaborative and supportive group we have managed to continue through the 5 years at UWE, which undoubtedly has contributed to the achievements and level of professionalism we achieve in our work, displayed through this Dissertation Report.

Disclaimer

The content and work declared within this dissertation report has been composed solely by myself (Jack Martin) and that the work has not been submitted for any other degree or professional qualification. There has been no 3rd Party involvement apart from the feedback and guidance offered by representatives of Thales Alenia Space UK and Project Supervisor.

Contents

Figures

Tables

1. Introduction

1.1 Project Background

The operational capability and sustainability of current spacecraft is constrained by their inherent physical system architecture, highly customised monolithic design and the limited or no availability of servicing and maintenance. The research of cost effective, high performing, scalable and flexible solutions is essential to the future of the European Space Agency to being the primary focus of the commercial space industry strategy.

Project MOSAR is a European Union funded initiative made of an international consortium of 9 partners that will aim to raise the degree of modularity of space systems by an order of magnitude with respect to current space industry standards. The current concept of a modular space missions system is shown in Figure 1 (EJR-Quartz, 2021). Thales Alenia Space UK is a world leader in space systems designs, integration and manufacturing with advanced space engineering facilities located locally to the University of West of England (UWE), acting as the leader on the design of satellite propulsions system, satellite sub-systems and system design studies.

An area of opportunity identified by Thales Alenia UK was the research into a combined architecture that could provide two functionalities: these being mission systems and sub-systems inter-communications and power distribution throughout a spacecraft. If viable, this would result in the potential to produce spacecraft requiring less materials for their construction and operation, to be re-usable due to the ability to reconfigure mission system architectures and less debris being ejected into space. All of these factors and more would contribute towards an intrinsically efficient and sustainable approach to the environmental challenges faced within the space industry.

Current spacecraft mission systems adopt bespoke and/or proprietary methods of deploying communication networks such as MIL-STD 1553B and CAN 2.0 which both present their own issues when considering interoperability with interfacing sub-systems. It is desirable that an existing standardised full protocol stack could provide the underpinning inter-communications between future spacecraft sub-systems as well as provide a common architecture to also distribute power; a solution to support the Project MOSAR objective of modularising spacecraft sub-systems.

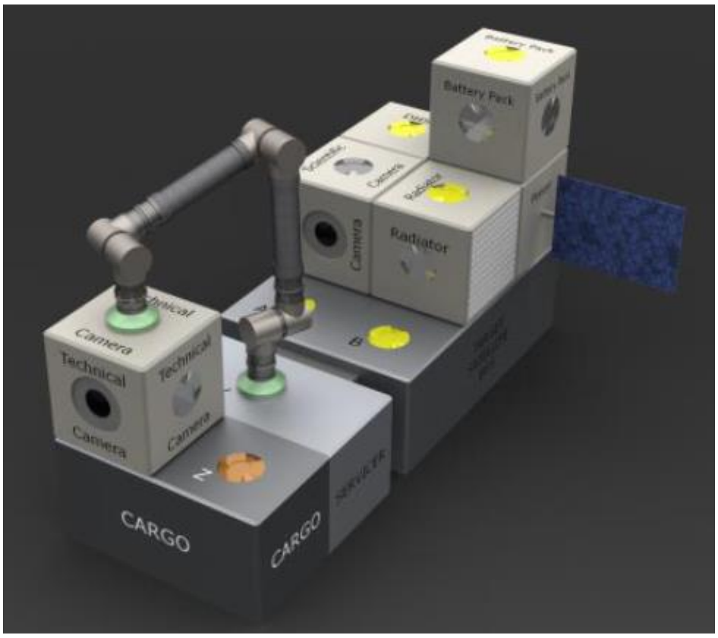


Figure 1 – Project MOSAR Concept of Modular Spacecraft Assembly

1.2 Aims and Objectives

The aim of the project was to conduct a feasibility study of the given topic; "The implementation of ECSS SpaceWire into the IEEE 802.3 Power over Ethernet (PoE) Protocol", theoretically, analytically and through functional testing defined by qualitative criteria. From the described aim, objectives for this project were identified as below;

i. Identify and demonstrate the concept of utilising existing engineering standards and topologies whilst respecting existing space industry standards, for the purpose of providing the key functionality of inter-communications of spacecraft sub-systems, in support of Project MOSAR.

ii. Conduct an initial feasibility study of the project aim through Engineering Standards, Space Industry Standards and Literature Review.

iii. Assess the validity of the feasibility study through the means of developing an Embedded System and conducting Functional Testing and Analysis.

iv. Identify the readiness level of the proposed feasibility study and provide the basis of extended recommended work to further develop the system/capability.

v. Demonstrate evidence against all Project Module Specification (UFMFX8-30-3) Learning Outcomes.

vi. Activities associated with the project are to be identified within a Risk Assessment and mitigated to Tolerable and As Low as Reasonably Practicable (ALARP) levels.

The initial aims and objectives defined within the Interim Research Proposal have not changed in notion, although have been re-worded to better reflect the academic tone and overarching project contribution to the wider goals in relation to Project MOSAR. The aims and objectives were appropriately defined to consider the impacts of the pandemic and mitigate potential constraints that could affect the outputs of the project, therefore no changes have been required to be made. All of the project aims and objectives have been achieved.

1.3 Scope

The technical scope of the project was refined to meet the qualitative criteria defined below, in order to meet the project time and resources constraints internally and externally from the ongoing pandemic;

i. The implementation of only the SpaceWire Network Layer into the Ethernet Protocol Stack onto an embedded system.

ii. Functional Testing shall prove the capability only through qualitative criteria of incorporating the SpaceWire Network Layer into the Ethernet PoE protocol.

iii. Functional Testing shall be conducted with commercially available Commercial Off The Shelf Hardware.

iv. The SpaceWire and Ethernet network shall only consist of an end-to-end link (no routing switches).

1.4 Project Management & Artefacts

The project was managed and controlled effectively, ensuring that the time constraints given to complete the project were planned and implemented appropriately, to conduct the identified activities to achieve the project aims and objectives. There was a degree of uncertainty of how long the solution would take to deploy to an embedded system which was considered within the project schedule; the activity was completed early and therefore allowed for additional activities to be conducted to support the project, as agreed with the project supervisor. The scope of the project was defined appropriately so that the project would not be impacted by pandemic restrictions, this was achieved by constraining the work to predominantly software development and setting qualitive objectives that did not require specialist equipment for testing.

1.4.1 Project Methodology

The project methodology was based upon the Systems Engineering V-Model which is endorsed by the International Council on Systems Engineering (INCOSE, 2007). It stipulates a linear/waterfall approach to systems engineering in relation to the project life-cycle stages and processes, ensuring that the initial user and system requirements identified for the project are verified and validated. The V-Model and how the project's activities aligned with each stage are shown in Figure 2.

Adopting this methodlogy as the project strategy informed the milestones and supported the development of the structure of how to conduct the project activities to achieve the project aim. It also allowed for the project and solution system to bounded suitably, whereas an Agile approach could have allowed the project to be susceptible to requirements creep causing delays and lack of defined technical scope.

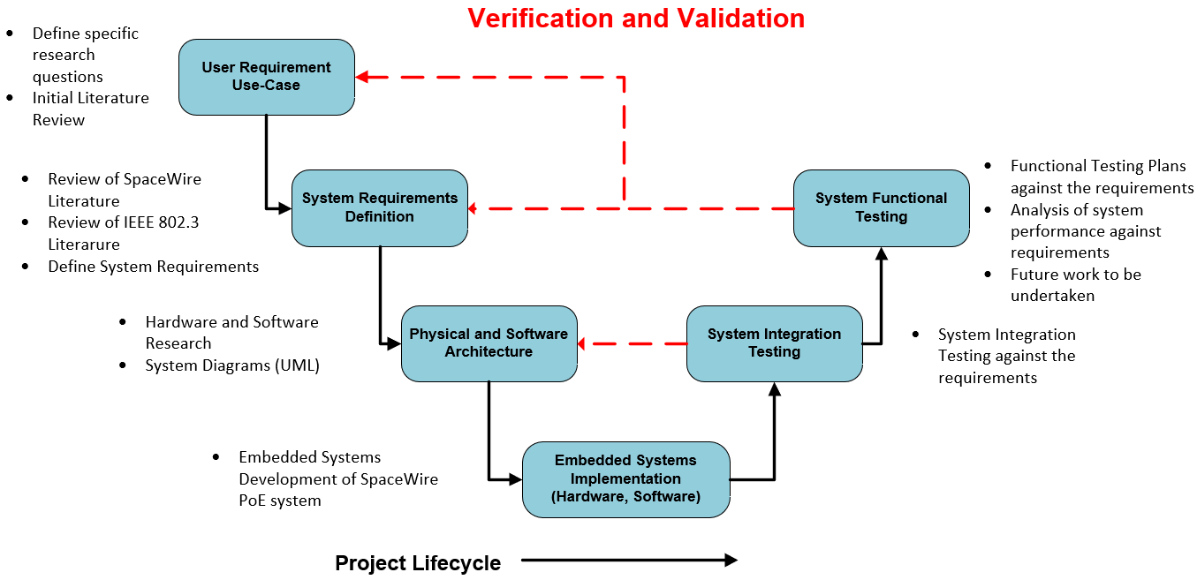


Figure 2 - Project Methodology (V-Model)

1.4.2 Initiation Forms

The Project Initiation forms were completed and authorised by the Supervisor to ensure that the project resources, ethics, risk and feasibility did not present any immediate issues to be rectified. No forms were required to be altered from the initial submission in conjunction with the Interim Research Proposal. Where required the initiation forms have been included in Appendix A and B.

i. Baselined Project Schedule/Gantt Chart

ii. Project Resources Form

iii. Health and Safety Risk Assessment and ALARP Justification

iv. Ethics Checklist

v. Security Awareness Form

1.4.3 Project Gantt Chart

A Baselined Project Schedule was generated at Project Initiation to ensure the key activities, milestones and critical path could be identified, to plan the project accordingly with the given time limitations and resources. The project schedule was updated as the project progressed to ascertain the activities that had been completed and to monitor the progress. Where identified with the Project Supervisor, additional tasks of opportunity to support the project were allocated to the schedule throughout the project.

1.4.4 Risk

A Health and Safety Risk Assessment was conducted at the initiation of the project and regularly updated as appropriate to account for changes in project activities and equipment. The hazards identified and the associated risk were assessed as Tolerable and ALARP not requiring any further control measures to reduce the risk to the exposed parties. No changes or additions were required to be made to the risk assessment during the project.

1.4.5 Ethics

It was agreed between the Student and Project Supervisor that no ethical issues could arise from the area of research and activities associated with this project. The on-line Ethics Checklist was completed and supported this judgement.

1.4.6 Meeting Minutes and Contact Register

Regular contact was held between the student and Project Supervisor when required throughout the project, of which Project Meeting Minutes were generated, capturing a Record of Decisions, Discussion, Attendees and Actions. These were captured on-line within Microsoft OneNote with a supporting Action Tracker (Appendix B) which was monitored throughout the project.

1.4.7 Dependencies

Only one dependency was identified for the project, which was for the resources (hardware) to be provided by UWE to enable the embedded systems development and subsequent functional testing activities. This dependency was mitigated early on within the project as the hardware was acquired early.

1.4.8 Configuration Management

Configuration Management for the project was essential for the control of source code that was to be written for the hardware devices, to ensure that the latest versions of software were appropriately stored and could not be mistakenly altered which would prohibit progress. The chosen tool for source-code control throughout the project was GitHub; a Github Repository can be created within a Project File Folder that tracks the changes and provides version control.

1.4.9 Resources

Where identified, the resources required to achieve the project which incur additional processes or procurement were identified within the Project Resources Form (Appendix B). The following resources were utilised throughout the project which were already readily available;

i. Microsoft Visual Studio Code

ii. Microsoft Office Software

iii. Github

iv. UWE Library (Standards)

v. Modelio

2. Literature and Documentation Review

2.1 Research Strategy

From the aim of the project and the technical scope being defined, it was clear that the project was considered to be an engineering design study. The most appropriate research method would be to collate and assess the current information that informs the solution to the given engineering problem. Therefore the use of research questions was utilised, of which the following six research questions were identified that would progress the project to the stage of being able to design, implement and functionally test a solution system.

i. What are the requirements of the SpaceWire and IEEE 802.3 Ethernet Protocols?

ii. Can the SpaceWire protocol be implemented into the IEEE 802.3 Ethernet Protocol to create a full stack?

iii. What are the difficulties and advantages of implementing SpaceWire into the Ethernet Protocol Stack?

iv. What does IEEE 802.3 Power over Ethernet deliver and how can it be exploited?

v. What use-cases would benefit from a combined power and communications architecture?

vi. What is the future vision of SpaceWire by the Academic/Space Industry Community?

The objectives of the project were defined such that the potential effects of the pandemic were mitigated, and thus the research methodology was not required to be constrained or altered. It was recognised early in the project that most of the research phase would consist of desk-level analysis of open-source documentation, therefore no other methodologies were required to be considered. The identified research questions were answered by identifying and reviewing the relevant engineering standards, space industry standards, and current academic literature on the given topic.

2.2 Project Context and Existing Academic Literature

2.2.1 SpaceWire Community of Interest

Within the community of interest for SpaceWire applications, there has been the study of topics on SpaceWire and how it can be progressed to become more efficient and interoperable with multiple missions-systems and technologies. In particular the International SpaceWire Conference (Conference, 2016) is a collation of many SpaceWire Studies, of which includes the following;

i. The distribution of SpaceWire Time-Codes for Spacecraft synchronisation (Susan, Mazen and Angkasa, 2016).

ii. The synchronisation of SpaceWire Interrupt Codes delivery in aerospace onboard networks (Liudmila, Elena and Yuriy, 2016).

iii. SpaceWire network management and discovery (Krzysztof *et al.*, 2016).

From the given example papers published it can be observed the attitude to move towards a further integrated spacecraft architecture that is harmonious in both architecture and function is the strategy for the space industry. The general attitude of moving towards a system-level of cohesiveness can be realised using a SpaceWire Power over Ethernet Protocol and directly complements the modular and efficiency focused culture within the space industry that is embodied through Project MOSAR.

2.2.2 SpaceWire and Ethernet Literature Review

The implementation of transmitting SpaceWire Packets over Ethernet has been investigated in literature within the aerospace and space community (Rozanov and Yablokov, 2014); identifying the methods, limitations, and issues that arise.

The overarching issue is the limitation of the format of the ethernet frame containing a fixed sized header, payload and error checking sections that in some scenarios are inefficient to transmit. SpaceWire packets in principle can have infinitely long payload sizes whereas Ethernet can only transmit a fixed length of between 46 – 1500 bytes, therefore it is able to be recognised that for SpaceWire payloads larger than 1500 bytes that further processing within the network layer of the full protocol stack is required to re-build segmented payloads.

The constraint of Ethernet Frames also requires further processing to determine the SpaceWire data-type that is contained within the Ethernet payload, including N-Char Data and Broadcast Codes which consist of time-codes and interrupt codes. An interpretation of how this can be implemented is shown in Figure 4 (Vinogradov, Yablokov and Yachnaya, 2019). In this interpretation the necessary artificial headers are inserted to allow the identification and size of the SpaceWire data being sent, and subsequently how this is handled within the receiving element via further processing.

The requirement of further SpaceWire headers to be inserted within the Ethernet Frame resultantly has a detrimental impact on the efficiency of the communications link. Figure 3 shows the relationship between the number of bytes sent with respect to the payload for both the Ethernet and SpaceWire protocols. It shows that the viability of the Ethernet Protocol is only applicable where payloads of more than 30-40 Bytes are transmitted per packet, therefore for the implementation of the combined SpaceWire and Ethernet protocol this characteristic is to be considered for the intended use-case of the spacecraft mission system.

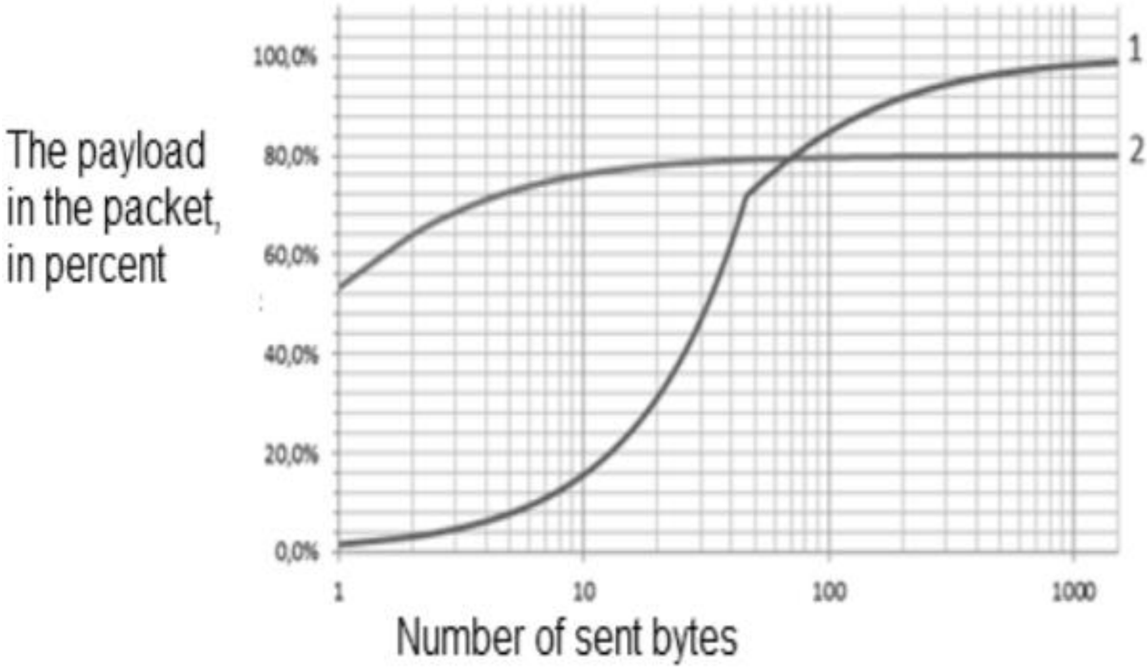


Figure 3 - Ethernet (1) and SpaceWire (2) Packet vs Payload Analysis

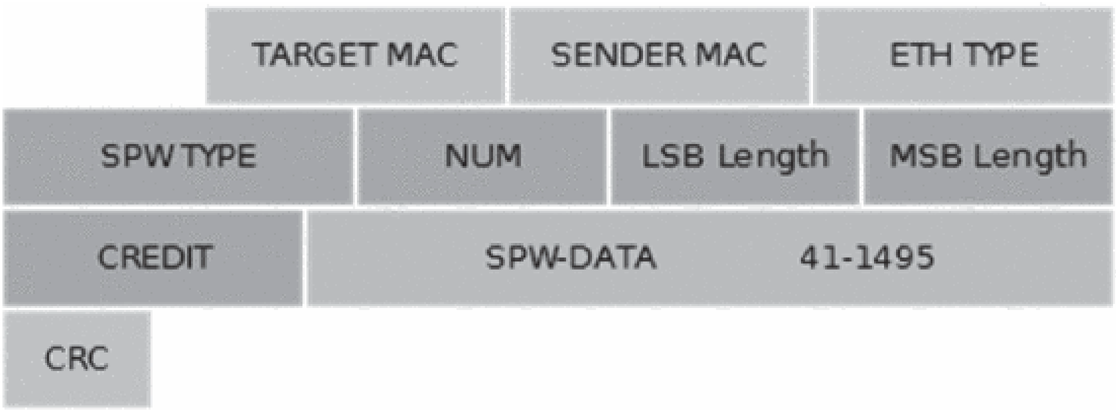


Figure 4 - SpaceWire Ethernet Frame Interpretation

2.3 OSI Concept Model

To understand how to design and implement a communications system, it was required to research the theoretical concept of how communication systems are designed, in reference to professional engineering standards. A common topology used to describe a communications system is the full protocol stack. The Open Systems interconnection (OSI) Protocol Stack is defined within Reference (OSI, 1994), which sections the theoretical full protocol stack in seven abstract layers, each one stacked up upon the last showing how data is compartmentalised and handled at each stage. The seven layers of a communications protocol are defined by the International Organisation of Standardisation are shown below in Figure 5 (Forcepoint, 2020).

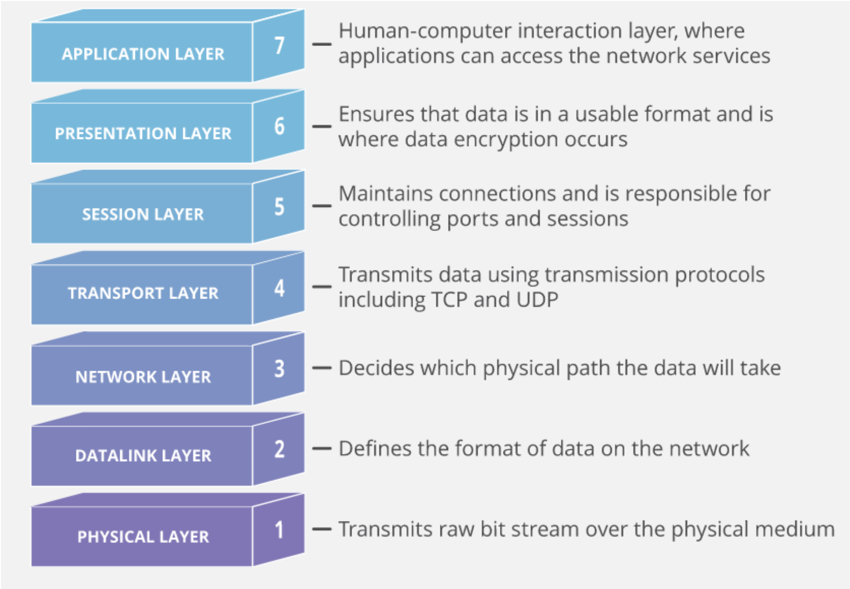


Figure 5 – OSI Concept Model

To achieve the aim of this project, it was necessary to recognise the requirements of what is regarded as a full communications protocol stack (OSI Model), to allow the transmission and receiving of information between communications systems.

2.4 ECSS SpaceWire

2.4.1 ECSS SpaceWire Overview

The European Cooperation for Space Standardization (ECSS) SpaceWire is defined within the ECSS-E-ST-50-12C Rev.1 standard (European Cooperation for Space Standardisation - ESA Requirements and Standards Division, 2019). The rationale behind SpaceWire was to standardise the proprietary and ad-hoc approach adopted by space equipment manufacturers for interunit communications. This led to several different types of communication links being used on spacecraft, which subsequently increased the cost and time required for spacecraft integration and testing.

The use of the SpaceWire standard ensures that all equipment used on spacecraft is compatible at both the component and sub-system levels, allowing greater re-usability and reliability whilst also reducing development costs. The typical application of a SpaceWire based inter-communications network is shown in Figure 6.

SpaceWire provides a 2 – 200 Mbps, bi-directional, full-duplex data link that connects together SpaceWire enabled equipment. The SpaceWire implementation can either be a point to point link or the use of a network routing switch. The typical SpaceWire application is shown in Figure 6, the inter-connection of several SpaceWire enabled equipment is achieved through the use of SpaceWire compliant routing switches.

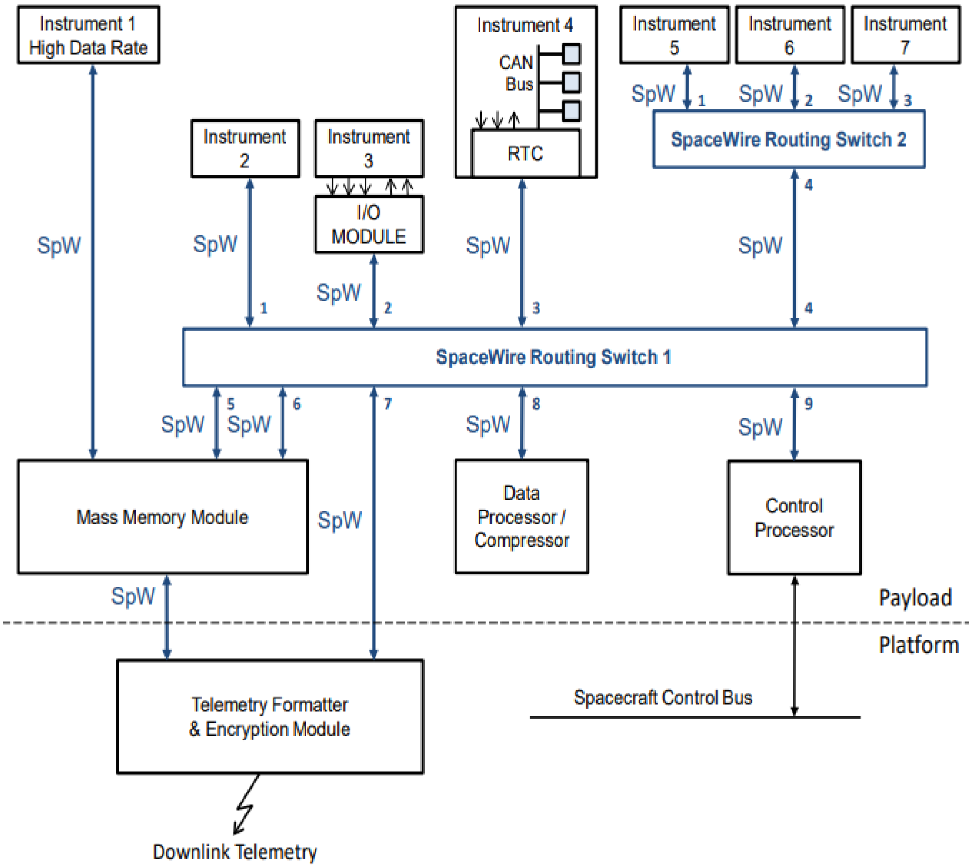


Figure 6 - Example SpaceWire Application

2.4.2 ECSS SpaceWire Protocol Stack

The SpaceWire Protocol Stack can be described in reference to the OSI Concept Model (Figure 5) to establish the methodology for network communications between sub-systems and components within a spacecraft mission system. SpaceWire mandates the use of four layers of the OSI concept model, notably being the Physical, Data Link, Network and Application Layer. SpaceWire also specifies the use of a Management Information Base that configures and determines the status of the SpaceWire network. The SpaceWire Protocol Stack is shown diagrammatically in reference to the OSI Concept Model in Figure 7 and Figure 8.

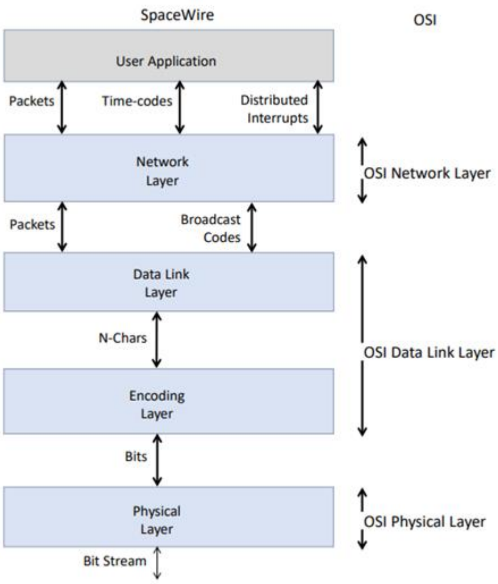


Figure 7 - SpaceWire Protocol Stack compared to OSI Concept Model Layers

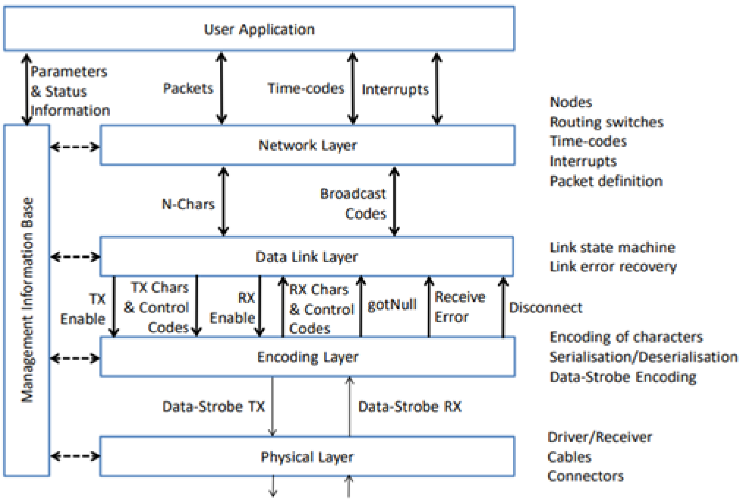


Figure 8 - SpaceWire Protocol Stack

2.4.3 Physical Layer Overview

The SpaceWire Physical Layer defines the cables, connectors and line drivers for transmitting and receiving SpaceWire data and strobe signals over the physical medium. The physical connector consists of a nine-pin micro-miniature D Type Plug which encapsulates 4 twisted pairs of differential signalling wires. The SpaceWire Type A connector is shown in

Figure 9.

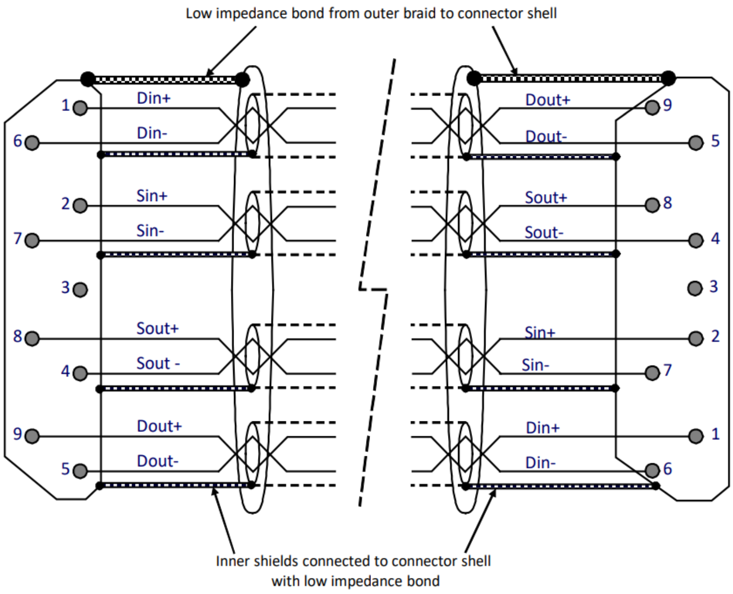


Figure 9 – SpaceWire Type A Connector

Figure 9 shows the use of 4 differential signal pairs that allows bi-directional data transmission. The signalling wires provide the physical medium for Low Voltage Differential Signalling (LVDS) for SpaceWire Data and Data Strobing Signals. The strobing signal is used to encode and recover the transmission clock signal which is simply recovered by XORing the Data and Strobe Lines together. The coding scheme is shown in Figure 10, illustrating how the transmission clock signal is encoded into the data and strobe signals.

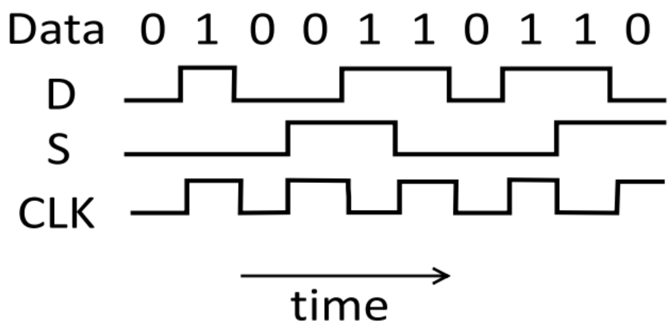


Figure 10 - Data Strobing Coding Scheme

2.4.4 Data Link Layer Overview

From Figure 7 the Data-Link and Encoding Layer within the SpaceWire protocol stack are to be considered just the Data-Link Layer in reference to the OSI Concept Model (Figure 5). The Data-Link Layer within the SpaceWire standard specifies how communication is established across a link, how the information flow is controlled, how N-chars and broadcast codes are sent and received, and how communications is re-established across the link after an error has occurred. The Encoding Layer within the SpaceWire standard specifies the encoding/decoding of characters into symbols, the serialisation/de-serialisation of the encoded symbols into a bit stream, and the data-strobe encoding/decoding of the serial bit stream.

SpaceWire has two types of characters which are the data and control characters. Data characters hold 8-bit values (char/byte) and also a parity bit and data-control flag, therefore containing 10 bits per character. Control characters hold a 2-bit control code in addition to a parity bit and data-control flag. In addition to this control codes are also used for Link Control and Time-Code distribution over a SpaceWire network. SpaceWire data characters, control characters and control codes are shown in Figure 11. The only further encoding that occurs is the Data Strobing that is driven over the physical layer as described in 2.4.3.

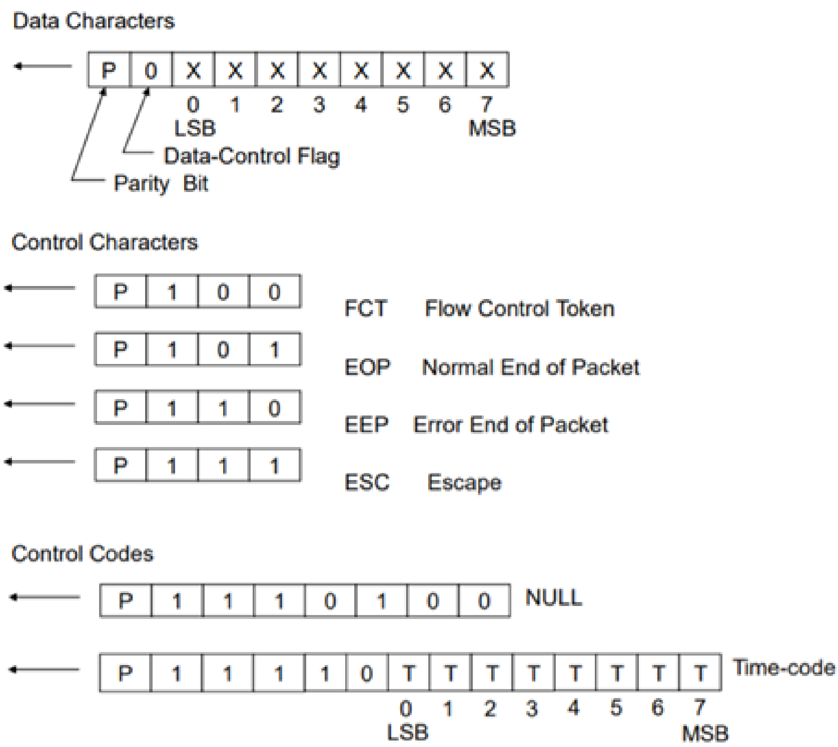


Figure 11 - SpaceWire Data and Control Characters and Control Codes

2.4.5 Network Layer Overview

The Network Layer specifies how SpaceWire Packets, time-codes and distributed interrupts are transferred over a SpaceWire network. SpaceWire Networks are built from links, nodes and routers which are connected together so they can exchange information and work together to perform a desired function. The Network Layer is defined to provide the three principal services listed below.

i. A packet service which sends and receives packets over a SpaceWire network.
ii. A time-code service which sends and receives time-codes over a SpaceWire network.
iii. A distributed interrupt service which sends and receives distributed interrupts over a SpaceWire Network.

Within a SpaceWire Network the nodes are required to act as the following three requirements;

i. A source of packets and broadcast codes sent over a SpaceWire network.
ii. A destination for packets and broadcast codes received over a SpaceWire network.
iii. Both a source and destination of packets and broadcast codes.

2.4.6 SpaceWire Packets

Information is distributed across the network through the use of SpaceWire Packets; a SpaceWire Packet consists of the format shown in Figure 12. A SpaceWire Packet is made up of three elements;

i. **Destination Address**. The first part of the packet to be sent which is a list of data characters that represent the path a packet should take through the SpaceWire network to reach the destination node. If the link is a point to point link between two nodes, the Destination Address is not required.

ii. **Cargo**. The data that is sent following the Destination Address. The length of N-Chars/Bytes that can be transferred in the cargo section is not limited to any number, although it is advised that data is split into appropriate frames to reduce likelihood of errors and need for re-transmissions causing delay.

iii. **End of Packet (EoP)/Error in Packet.** The last section of the packet that indicates the current packet of data has ended. The data character following an EoP is the start of the next packet. Where an error has occurred in the link, an EEP will be inserted at the end of the packet.

Figure 12 - SpaceWire Packet Format

2.4.7 SpaceWire Packet Addressing

The Destination Address at the front of a SpaceWire Packet is used to direct the packet through the SpaceWire Network. The two forms of packet addressing are defined as Path Addressing and Logical Addressing. The main difference lies in how the packets are forwarded to the correct ports from each routing switch to reach the desired destination node; Path Addressing has each path (output port) listed at the front of the packet which is processed by the routing switch, Logical Addressing just has the desired destination node at the front of the packet and the routing switch has pre-defined "directions" (lookup table) on the correct path to take to the desired destination node.

2.4.8 Broadcast and Time-Codes

SpaceWire supports the use of optional network layer broadcast codes which are used to for the distribution of time-codes and interrupt-codes within a SpaceWire Network. The time-codes allow a network to have a consistent view of time and synchronising sub-systems. The interrupt-codes allow network level interrupts to be distributed across a SpaceWire network for uses such as power-down/up controls, with up to 32 use definable interrupt signals.

The SpaceWire time-code comprises of the SpaceWire ESC character followed by a single 8-bit character, the data character contains two control-flags and a six-bit time count. The SpaceWire time-code format is shown in Figure 13. The time-codes values are stored in a dedicated time-code register by the nodes.

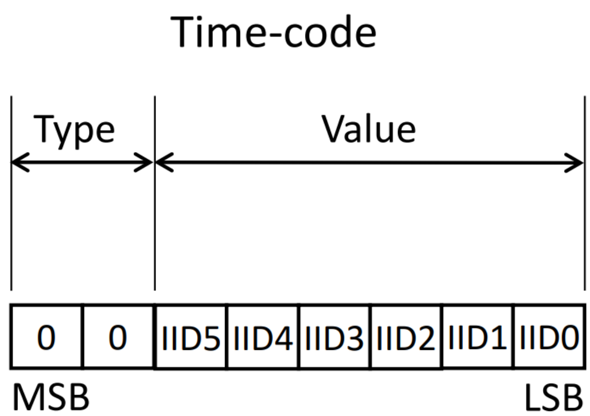


Figure 13 - SpaceWire Time-Code

The SpaceWire interrupt-code utilises the same 8-bit format as the time-code in Figure 13, except Bit 5 is reserved for interrupt acknowledgment.

2.5 IEEE 802.3 Ethernet

2.5.1 IEEE 802.3 Overview

The Institute of Electrical and Electronic Engineers (IEEE) 802.3 Standard (IEEE Computer Society, 2018) is an international standard for Local and Metropolitan Area Networks, employing Carrier Sense Multiple Access/Collision Detection (CSMA/CD) and the Ethernet Protocol (IEEE 802.3) for data communication. In reference to the OSI Conceptual model (Figure 5) the Ethernet Protocol implements the Physical and Data-Link Layer for a data communications. The sub-layers Media Access Control (MAC) and Logical Link Controller (LLC) make up the Data-Link Layer of the OSI Model. The representation of IEEE 802.3 Ethernet Protocol in reference to the OSI Model is shown in Figure 14, with the variances in the physical layer depending on the transmission speeds available within the protocol.

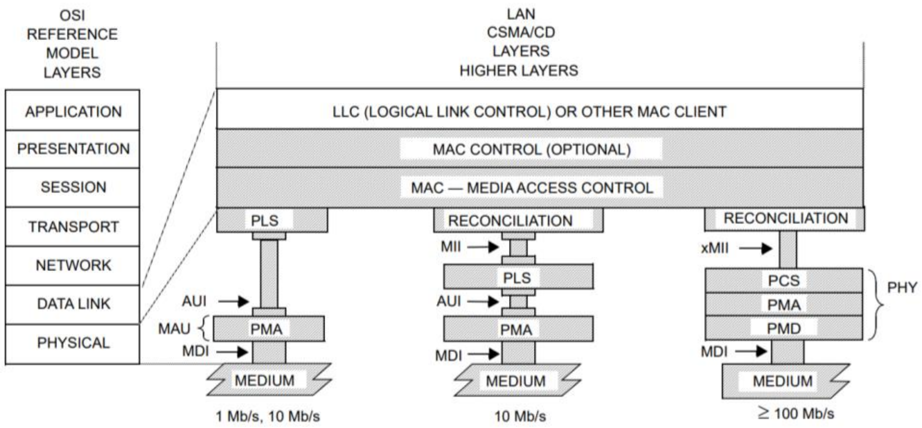


Figure 14 - 802.3 Ethernet OSI Concept Model Comparison

2.5.2 Physical Layer Overview

The Physical Layer states the connectors denoted as the Medium Dependant Interfaces (MDI) for the IEEE 802.3 Ethernet Protocol. The implementation of the twisted pair topology are realised by the RJ45 connector as defined in IEC 60603-7:1990. The 8-pins on the connector and physical design are shown in Figure 15, consisting of 4 twisted-pairs. The physical layer provides the interface to allow the signals defined within 802.3 Protocol to be transmitted and received by devices through the use of differential signalling wires.

Contact	MDI signal
1	TD+
2	TD–
3	RD+
4	Not used by 10BASE-T
5	Not used by 10BASE-T
6	RD–
7	Not used by 10BASE-T
8	Not used by 10BASE-T

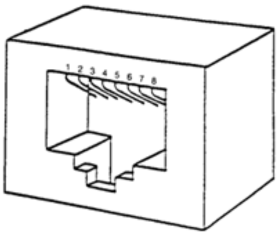


Figure 14–21—MAU MDI connect

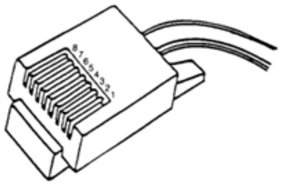


Figure 14–22—Twisted-pair link segment connector

Figure 15 - IEEE 802.3 RJ45 Twisted Pair Connector and Pinout

2.5.3 Data-Link Layer Overview

The IEEE 802.3 standard specifies the use of the two sub-layers MAC and LLC for the partitioning of data into frames and ethernet packets that are to be interfaced with the physical layer for transmission. These two sub-layers in unison achieve the below in Figure 16 (Devices, 2005);

a) Data encapsulation (transmit and receive)
   1) Framing (frame boundary delimitation, frame synchronization)
   2) Addressing (handling of source and destination addresses)
   3) Error detection (detection of physical medium transmission errors)

b) Media Access Management
   1) Medium allocation (collision avoidance)
   2) Contention resolution (collision handling)

Figure 16 - IEEE 802.3 Ethernet Data-Link Layer Functions

The MAC packet and frame format specifications are shown in Figure 17. The MAC packet and frame allows the synchronisation, addressing, client data and error checking of the data to be transmitted and received across the network.

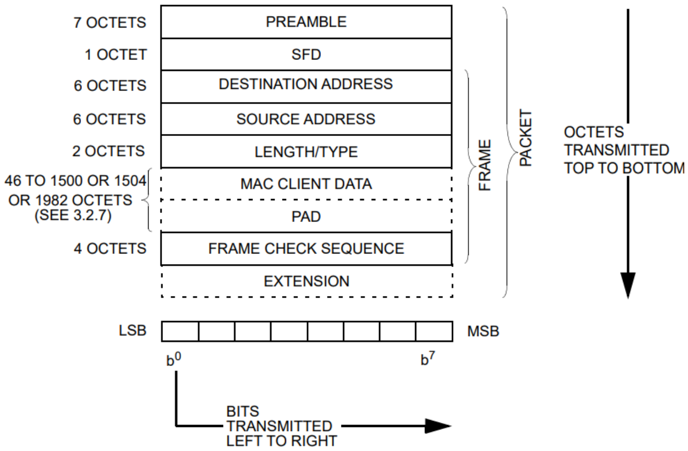


Figure 17 - 802.3 Ethernet MAC Frame and Packet Specification

The method of encoding for Frames and Ethernet Packets depends on the bit rate of the carrier signal. For 10Mbits/s data rate uses the Manchester Data Encoding Scheme when transmitted at the physical layer. Manchester Encoding uses the transition from Hi and Lo states to signal a '1' or '0' bit-symbol per clock cycle, therefore this allows a 10MHz baud rate. An Example of the Manchester Encoding Scheme is shown below in Figure 18 (Devices, 2005) for a bit sequence of '10110'.

Fast Ethernet or 100BASE-TX is defined by IEEE 802.3u-1995 (IEEE, 1995) for 100Mbits/s data rate which utilises the 4B/5B block encoding scheme and transmitted via Non-Return Zero Inversion (NRZ-I). The 4B/5B encoding scheme takes data nibbles (4-bits) to/from five-bit code groups, although only 16 of the 32 codewords are used to allow for atleast 2 signal transitions (two 1-bit values) due to the use of NRZ-I to improve receiver synchronisation. An example of the NRZ-I encoding scheme is shown in Figure 19 and five 4B/5B code-words in Figure 20 .

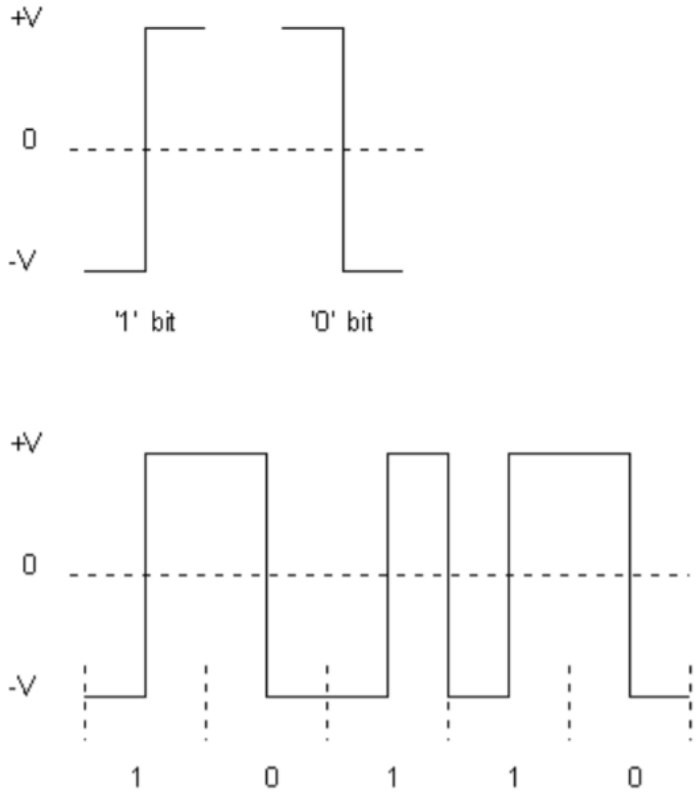


Figure 18 - Example Manchester Encoding Waveforms

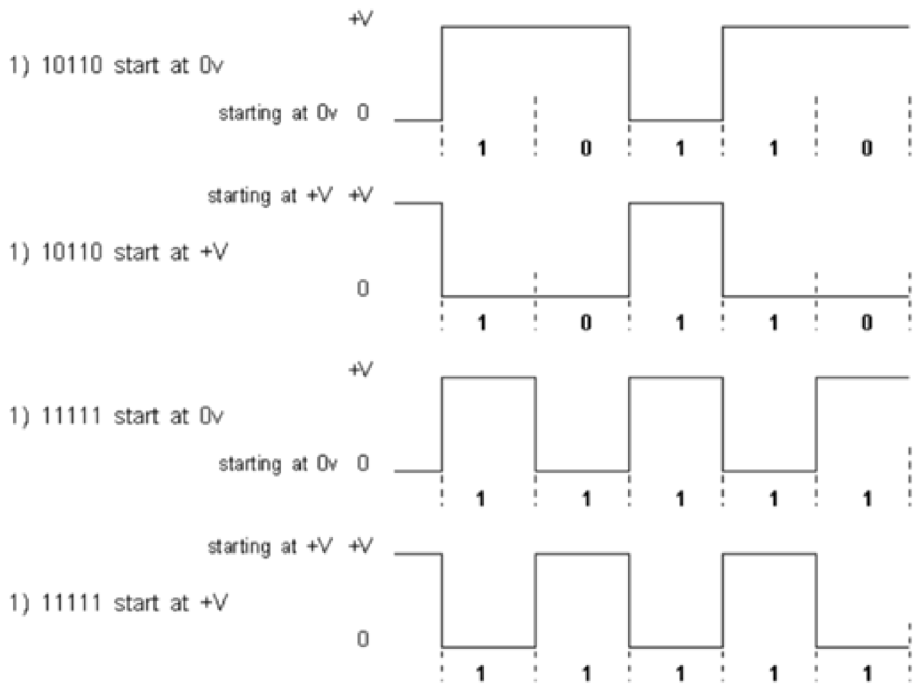


Figure 19 - Non-Return Zero Inversion Example

	PCS code-group [4:0] 4 3 2 1 0	Name	MII (TXD/RXD) <3:0> 3 2 1 0	Interpretation
D A T A	1 1 1 1 0	0	0 0 0 0	Data 0
	0 1 0 0 1	1	0 0 0 1	Data 1
	1 0 1 0 0	2	0 0 1 0	Data 2
	1 0 1 0 1	3	0 0 1 1	Data 3
	0 1 0 1 0	4	0 1 0 0	Data 4

Figure 20 - IEEE 802.3u 4B/5B Encoding Scheme

2.5.4 IEEE 802.3 Power over Ethernet Overview

The IEEE 802.3 Ethernet standard defines the specifications to include the distribution of power over an Ethernet based network, in addition to data communications within a network. The latest Power over Ethernet (PoE) capability is stipulated through IEEE 802.3bt (Man, Committee and Computer, 2019) utilising the 4-pair twisted pair cabling that has been qualified to be used within the standard. A PoE system comprises of three separated specified major elements, these are the; power supply defined as the Power Sourcing Equipment (PSE), the powered load defined as the Powered Device (PD) and twisted pair cabling connecting the two devices.

There have been many iterations of the IEEE 802.3 PoE Standard which has been enhanced since its introduction in 2003 with the 802.3af standard. Figure 21 (Microchip Technology Inc, 2019) provides a summary of the current PoE capabilities that are available, the key parameter being the total amount of output power that can be provided to a powered-device.

TABLE 2: PoE CAPABILITIES ON RATIFICATION

Type	Standard	PSE Minimum Output Power	PD Minimum Input Power	Cable Category	Cable Length	Power Over
Type 1	IEEE® 802.3af	15.4W	12.95W	Cat5e	100m	2 pairs
Type 2	IEEE® 802.3at	30W	25.5W	Cat5e	100m	2 pairs
Type 3	IEEE® 802.3bt	60W	51W-60W [1]	Cat5e	100m	2 or 4 pairs class 0-4 4 pairs class 5-6
Type 4	IEEE® 802.3bt	90W	71W-90W [1]	Cat5e	100m	4 pairs class 7-8

Note 1: Extended power capability allows PD input power to reach up to 60W for Type 3 and up to 90W for Type 4 if channel length is known.

Figure 21 - IEEE 802.3 PoE Standards Summary

An simple example of a PoE system is shown below in Figure 22 (Man, Committee and Computer, 2019). The diagram shows how the PSE utilises two unused twisted pairs to transmit power to the PD. Various topologies can be implemented which are categorised as Types within the IEEE 802.3bt standard which alters the physical architecture depending on the power source, number of components and the number of cables used. The data signal pairs can be utilised for power transmission also as differential signalling is utilised; therefore, the data signal is resultantly unaffected by a large DC offset.

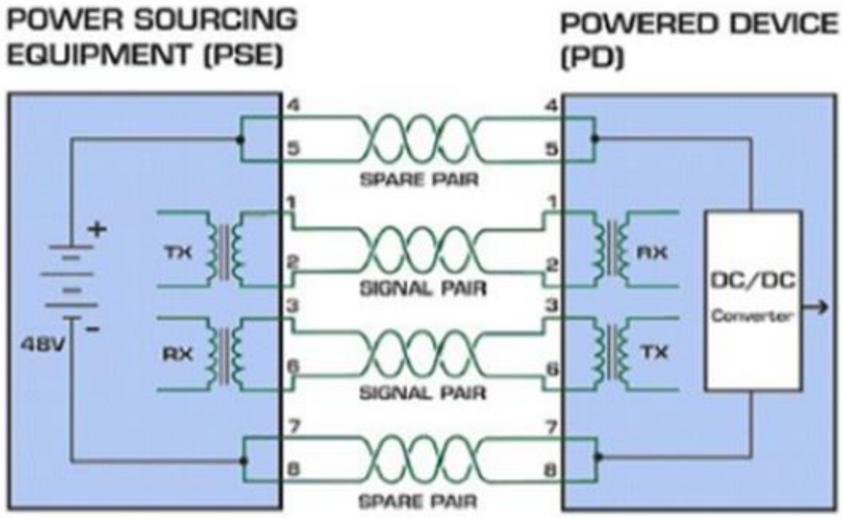


Figure 22 - Example PoE System

2.6 Research Evaluation

The findings from the research phase of the project identified the applicable requirements from both protocols to combine the OSI layers to achieve a full protocol stack. There has been considerate interest within the Space Industry and SpaceWire Community on the progression of SpaceWire with other technologies and in general the movement towards harmonious spacecraft system architectures for optimum efficiency. The methods, issues and interpretations have been explored for the implementation of SpaceWire into Ethernet and will inform the system development phase of the project.

It has been observed that the development of a SpaceWire PoE Protocol Stack offers a solution that assists the issues faced of sustainability and environmental factors that are continually challenged by the space industry. The deployment of a SpaceWire PoE system could reduce the costs of the manufacturing and operation of spacecraft throughout its lifecycle, whilst also reducing the amount of waste used in production and disposal.

3. System Design & Development

3.1 System Definition and Requirements

3.1.1 Use-Case

To develop a system that meets the project sponsor/user requirements to the fullest, it was requested that the qualitative objective of the project was expanded with the support of a use-case. Thales Alenia Space UK provided a use-case of how a SpaceWire PoE common architecture within a spacecraft would be utilised to conduct and manage mission systems during operations. The scope of the use-case for this project was refined as below;

"The use of a SpaceWire PoE common-architecture to switch on and off nodes of the mission system, negotiate the demand upon the request of power if the node can supply then transfer the maximum power. The SpaceWire PoE common architecture will also be used to distribute SpaceWire Packets within the mission system between nodes."

The top-level system use-case can be modelled diagrammatically using Unified Modelling Language (UML). From this diagram the primary actors on the system can be realised and the relevant activities that they are associated with.

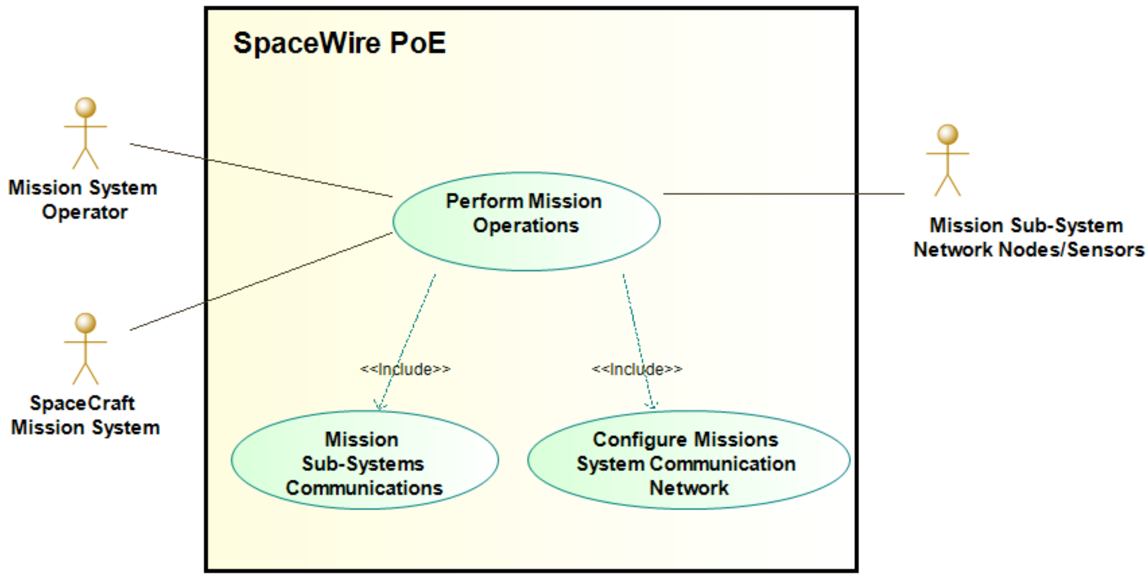


Figure 23 - SpaceWire PoE UML Use-Case Diagram

3.1.2 Requirements

From the documentation review of the ECSS and IEEE standards, Appendix C Appendix A demonstrates how the SpaceWire Protocol can be implemented using the existing layers of the Ethernet standard to fulfil a complete protocol stack. The Physical and Data-Link Layers of the OSI Model are implemented using hardware that are IEEE 802.3 PoE compliant. The Network Layer is implemented by software to receive and transmit SpaceWire Packets. The Application and Presentation Layers are not required for the functioning of the SpaceWire Protocol, although shall be used for functional testing, ad-hoc software development and debugging.

From the proposed SpaceWire and Ethernet full protocol stack identified in Appendix C a Systems Requirement Document (SRD) was generated to elicit the individual system requirements for each of the layers where applicable and the physical elements of the system, to meet the qualitative objective of the project and the given use-case. The SRD acted as the technical governance of the system development and would be used to assess the performance of the system during functional testing.

3.2 Hardware

The SRD defined the requirements that the SpaceWire PoE system hardware shall meet to achieve the qualitative objective of the project and given use-case in Section 3.1.1.

3.2.1 System Physical Architecture

The System Architecture Diagram shown in Figure 24 represents the typical topology of a spacecraft mission system and how a SpaceWire network would be utilised as the method of communications between mission sub-systems. To best replicate the use-case of such a system, it was decided to represent a communications system for a SpaceWire network as shown in Figure 6.

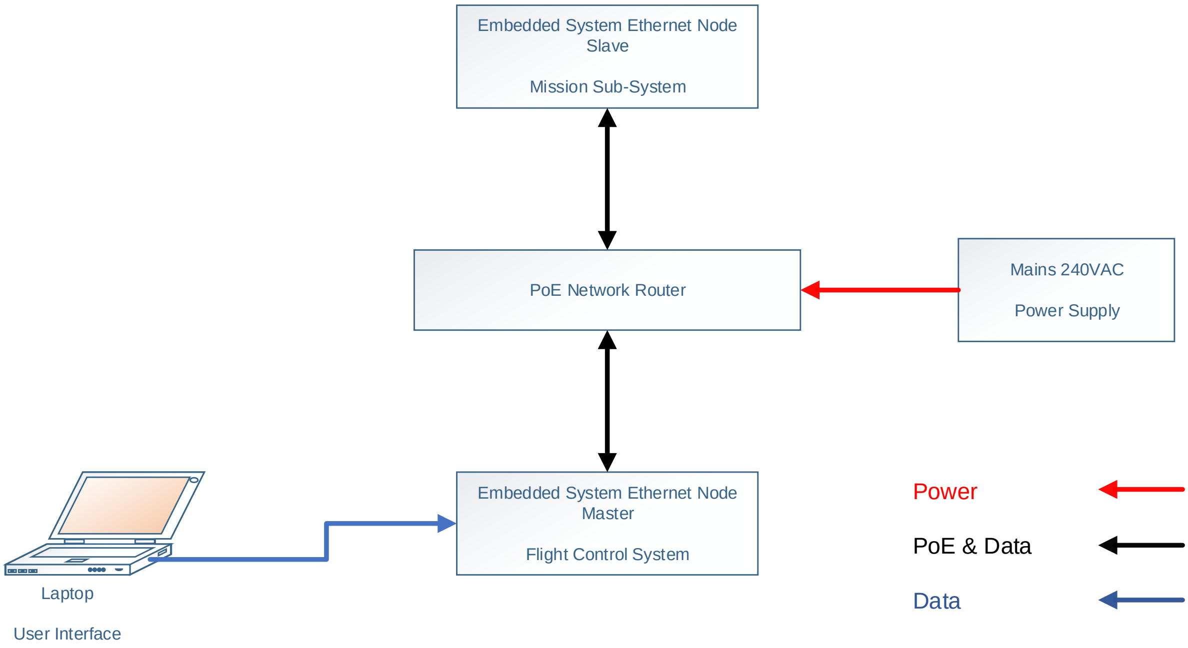


Figure 24 - SpaceWire PoE System Physical Architecture Diagram

The hardware for the SpaceWire PoE system was identified of which meets the SRD and represents the equivocal missions system physical architecture shown in Figure 24. The hardware selected for the system is listed below in

Table **1.**

Hardware	System Function	Description
ESP32-Ethernet-Kit-V1.1 (Figure 26)	Master Embedded Device	A System on Chip device, compliant to IEEE 802.3at
ESP32-Ethernet-Kit-V1.1 (Figure 26)	Slave Embedded Device	A System on Chip device, compliant to IEEE 802.3at
Tenda Desktop Switch with 4-Port PoE (Figure 25)	PoE Network Router	IEEE 802.3at Compliant PoE Network Switch
Cables	Power Delivery and Data	IEEE 802.3 Cat5e compliant cables

Table 1 - SpaceWire PoE System Hardware

Figure 25 - TEF1105P-4-63W Power over Ethernet Network Switch

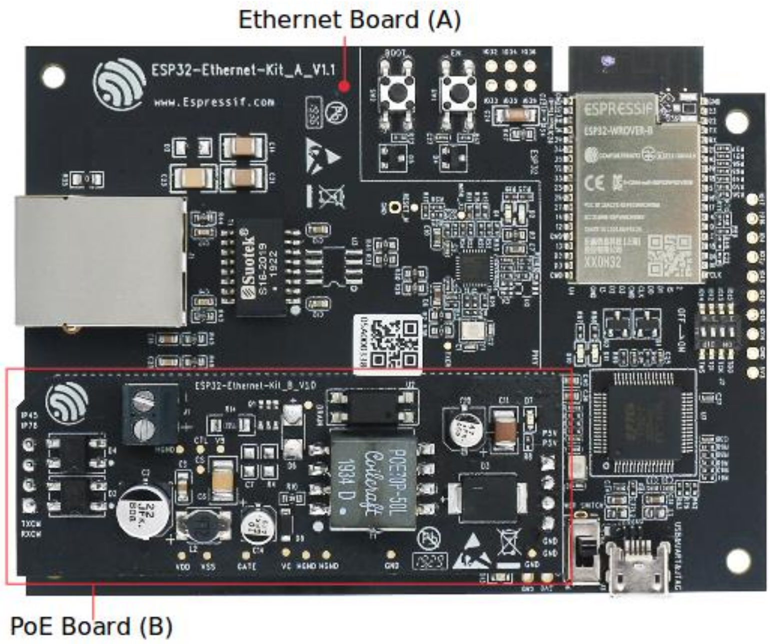


Figure 26 - ESP32 Ethernet Kit V1.1 Development Board

3.3 Software

The SRD defined the requirements that the SpaceWire PoE system software shall meet to achieve the qualitative objective of the project and given use-case.

3.3.1 Embedded Systems Development Overview

The source-code for the embedded system was developed using an Integrated Developed Environment (IDE), using the ESP Integrated Development Framework (ESP-IDF) plugin for Microsoft Visual Studio Code. The ESP32-Ethernet-Kit-V1.1 contains peripherals and functionalities such as WiFi and Ethernet Drivers that are accessible through the ESP-IDF and will be utilised for the system.

3.3.2 UML System Design Diagrams

The use of UML Diagrams assisted in the design and verification that the SpaceWire PoE system was being designed correctly. From the UML diagrams (Appendix E) it was able to be identified if the system design inhibited the required interactions and behaviour cited from the given use-case and SRD. From these diagrams the high-level software architecture could be understood and implemented for the SpaceWire PoE system. The scope of the UML Diagrams depicting the behaviour of the system only includes the function of sending and receiving SpaceWire packets; this is due to the Power over Ethernet function handled autonomously by COTS hardware, therefore this was not required to be considered.

The Sequence Diagram illustrates the functions that would occur for the sending (red box) and sending and receiving (yellow box) of SpaceWire Packets between nodes on the mission system. The State Diagram was generated to depict the different states that the system would be in for the given use-case and what interactions/event driven actions would occur.

3.3.3 Network Layer Implementation

The Network Layer of the SpaceWire PoE Protocol Stack is responsible for the formatting of the individual ethernet frames. It was identified further processing is required for the transmission of SpaceWire Packets within ethernet frames; therefore an artificial header one byte in size has been added within the payload of the ethernet frame to address the type of data between a N-Char and Control Code. During development there was time to incorporate an additional feature to re-assemble SpaceWire Packets that were larger than the size of an Ethernet Frame, although due to unknown issues this functionality could not be proven during testing and therefore has not been included. Table 2 shows the chosen values and descriptions of the artificial header bytes.

Header Byte	SpaceWire	SpaceWire PoE Value	Description
EOP_HEADER	N/A	0b101010 (char *)	The payload contains N-Char Packet
EOP	0bX101 (X = Parity bit)	0b100010 (char ‘)	SpaceWire End of Packet Marker
CCODE_HEADER	NULL Time-Code Interrupt-Code	0b100011 (char #)	The payload contains a control code

Table 2 - SpaceWire PoE Packet Header-Bytes

The hardware selected for the project (ESP-32-Ethernet-V1.1) is already supported with libraries and drivers for ethernet physical and datalink layers which are accessible through the ESP-IDF. The Ethernet Packets are “ready-built” for the user with only the MAC Addresses, Type/Length (optional), Payload/Padding (if required) required to be implemented by the user. The areas in grey are not user accessible and are implemented by the ESP-IDF; the padding which increases the payload size to the minimum of 46 bytes is automatically added for the user in frames for small sized transmissions.

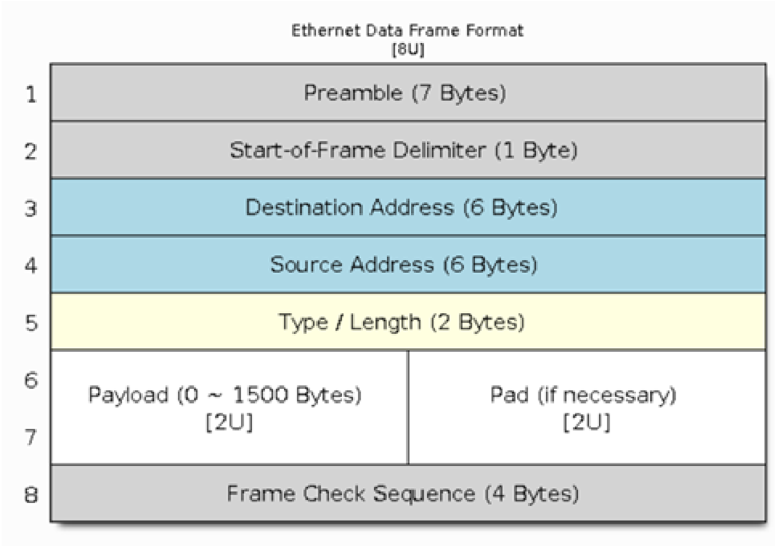


Figure 27 - ESP-IDF Ethernet Frame

SpaceWire networks utilise logical or path addressing to direct packets to the intended node in a network, although in this implementation the MAC addresses are used, which are required to be inserted to the packet through the host device in software. Each of the hardware devices inhibit their own unique MAC Address, which were identified as shown in

Table 3. Custom MAC addresses were not assigned to the hardware in this implementation, instead using the factory default addresses provided by the OEM which were suitable, due to the addresses not being uni-cast (sent to all devices on network).

Hardware Device	MAC Address
ESP32 Master Device	7c:9e:bd:cf:00:03
ESP32 Slave Device	7c:9e:bd:ce:ff:f3

Table 3 - SpaceWire PoE System MAC Addresses

Figure 28 diagrammatically shows the proposed SpaceWire PoE Packet format that was adopted for the system, implemented through the network layer of the protocol stack. The coloured elements show the origin of the header requirements for the packet. The proposed packet format is for the use-case of transmitting N-Char data, where the minimum number of data that can be sent is 0 bytes (46 bytes with padding) and maximum 1498 bytes per packet to meet the requirements of the ethernet frame.

SpaceWire Packets of sizes larger than 1498 bytes would be re-assembled using multiple ethernet frames and further processing.

Sending one SpaceWire Packet per ethernet frame is not the most efficient method of transmissions, therefore multiple SpaceWire Packets with the appropriate header-byte and EOP Marker can be inserted into the Ethernet payload section as shown in Figure 29. Further processing would be required to separate the received frame into the individual SpaceWire Packets.

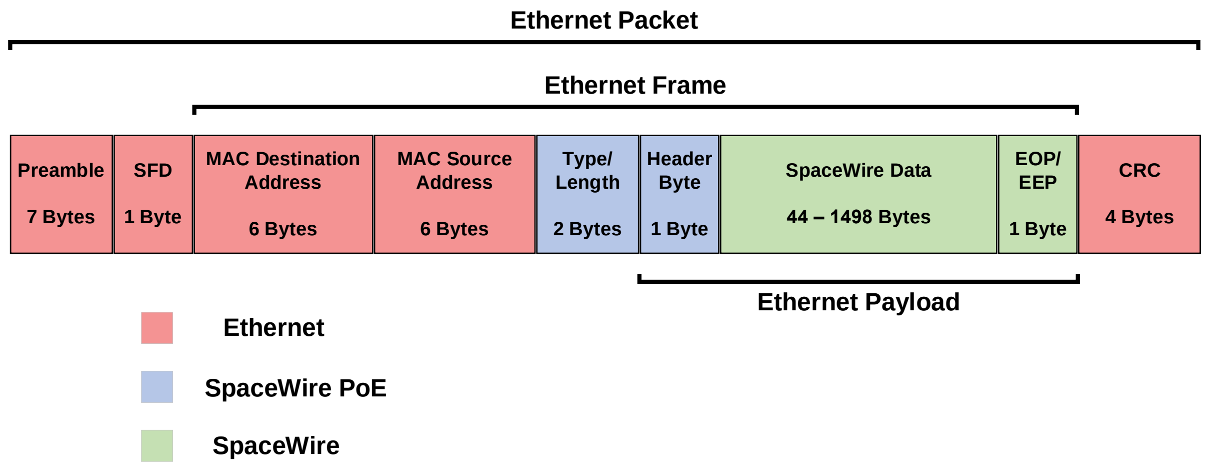


Figure 28 - SpaceWire PoE Packet Diagram

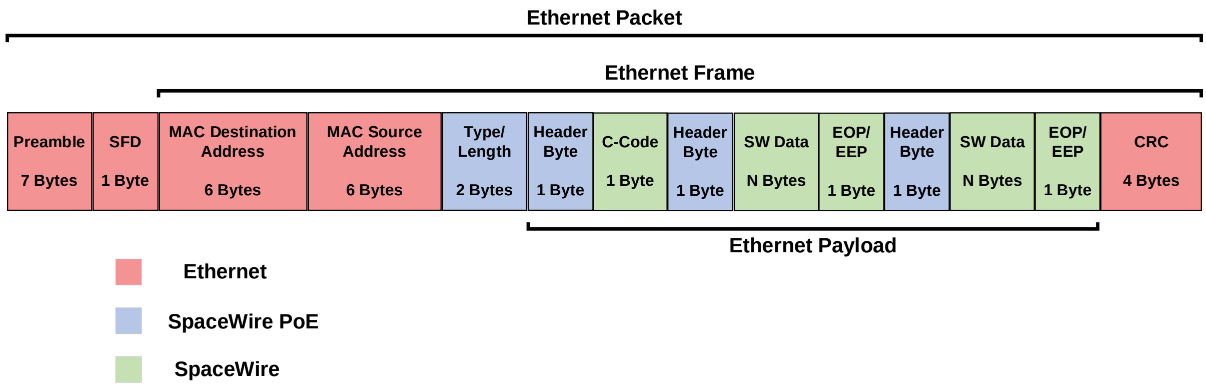


Figure 29 - SpaceWire PoE Multiple SW Packets Per Frame Diagram

3.3.4 Physical and Data Layer Implementation

The physical and Data Link Layers of the SpaceWire PoE System are already implemented via the ESP32 Ethernet Kit Development Board. The ESP32 Module contains an Ethernet MAC Interface complaint to IEEE 802.3 2008, and inhibits an external physical interface device (PHY) to connect to the physical LAN Bus (IP101 GRI ASIC). No changes were made to the development board for the SpaceWire PoE system.

3.4 Hardware and Software Integration

The hardware and software components of the project were easily integrated due to the built-in drivers and libraries within the ESP-IDF for the ESP32 development board, which could be easily flashed with the required source-code. The ethernet network switch is IEEE 802.3at compliant therefore no additional firmware/software was required for its integration into the system. The whole realisation of the SpaceWire PoE system is as shown in
Figure 30.

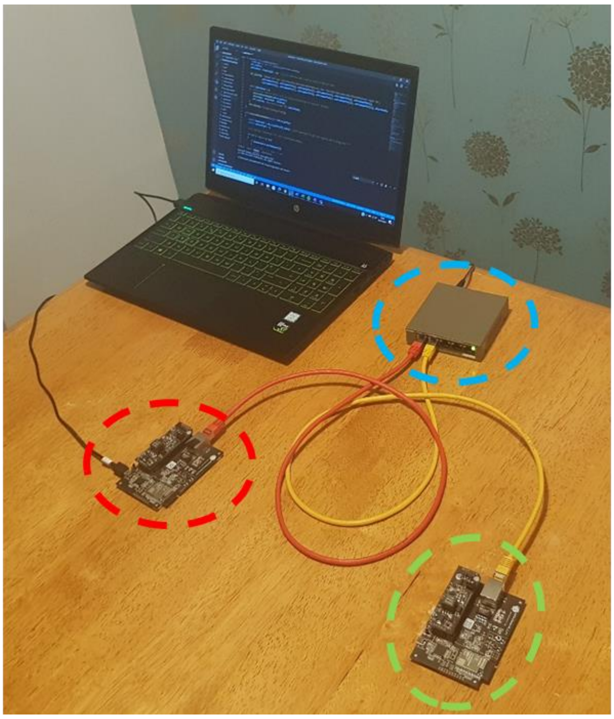


Figure 30 - SpaceWire PoE System

4. System Testing and Analysis

4.1 Testing Strategy

The testing strategy phase of the project sought to validate the SpaceWire PoE system requirements have been met as specified within the SRD and the system use-case where appropriate. Test-cases were generated in reference to the system requirements of the SpaceWire PoE system. The opportunity to conduct further testing was available due to completing activities earlier than anticipated, therefore theoretical analysis was conducted to quantitatively assess the system's performance to identify; issues, performance, shortfalls and areas of opportunity for further work.

If the pandemic restrictions were not enforced, it would have been desirable to validate the findings from the theoretical testing with physical test apparatus to output real qualitative data for comparison.

4.2 System Integration Testing

The System Integration Testing did not require in-depth testing due to the system requirements relating to system interfaces and compatibility being achieved through hardware compliancy, such as IEEE 802.3 RJ-45 and USB2.0. Therefore, the validation of SR's 5.0, 5.1, 5.2, and 5.3 were indirectly proven via the Functional Testing of the system.

4.3 System Functional Testing

4.3.1 Testing Configuration

The system was configured as shown in Figure 31 throughout the Functional Testing of the system. The arrows represent the connections between the components of the system, solid arrows being physical (Ethernet and USB), and dashed wireless (WiFi). The purpose of introducing the wireless element was to demonstrate that the SpaceWire PoE could function in isolation (no external connections), sending the required test data to a Server/Client Terminal Window (PuTTY).

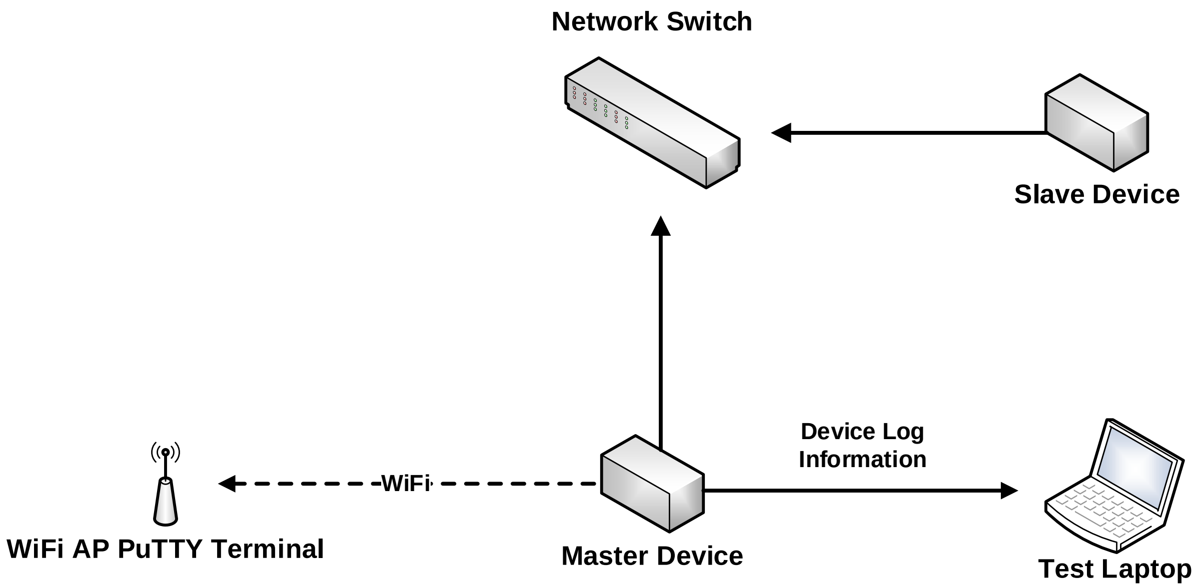


Figure 31 - Functional Testing System Configuration

4.3.2 Test-Cases and Results

To validate that the System Requirements for the SpaceWire PoE system had been met; test cases were derived to in reference to the Threshold and where applicable Objective requirements which are related to the Network Layer of the SpaceWire PoE Protocol Stack. The test-cases defined in Appendix F were deployed to the system with the results (compliance level) stated. The test data captured during the test-cases is shown in Appendix G . The SpaceWire PoE System achieved Full Compliance (FC) to 3 of the test-cases and Partial Compliance (PC) to 1 test-case. The Partial Compliancy was due to the lack of Time-Code Registers implemented in the Slave and Master Devices to hold the time value at a system level; further development could achieve FC with no anticipated issues. The achieved compliance against the system requirements is documented in the SRD (Appendix D). From this testing it can be determined that the SpaceWire PoE System achieves the project aim in accordance with the defined use-case and qualitative criteria.

4.4 System Theoretical Analysis

Theoretical Analysis was conducted with the use of Python Scripts and MATLAB functionalities to plot graphs that graphically display the performance of the SpaceWire

PoE system, particularly the efficiency of the transmission of frames containing SpaceWire data.

4.4.1 Frame vs Payload Data Transmission Efficiency

Figure 32 shows the efficiency of the SpaceWire PoE system and related protocols in terms of the ratio of payload (SpaceWire) data per frame/packet, in the scenario of sending one SpaceWire Packet with a payload containing N-Char data in the range of 1-1498 bytes. The efficiency shown for each protocol is in the context of the packets/frames being implemented for an end to end link (no logical/path addressing for SpaceWire required).

It can be observed that the SpaceWire protocol is significantly more efficient for the transmission of low-sized data payloads. The Ethernet protocol requires a minimum of 64 bytes to be transmitted even for low-sized payloads and the inflexion point on the graph signifies this point, therefore being less efficient in the range of sending between 0-46 bytes. The SpaceWire PoE system is only minimally less efficient than the Ethernet Protocol in the range of 40-400 bytes due to total header size of the frame being larger by 2 bytes for the header-byte and EOP marker, which eventually converges with the Ethernet protocol over 400 bytes where the difference in efficiency can be deemed negligible. The graph reinforces the assessment made within the SpaceWire community of interest (Figure 3), stating the efficiency losses when utilising ethernet for transmission of SpaceWire data.

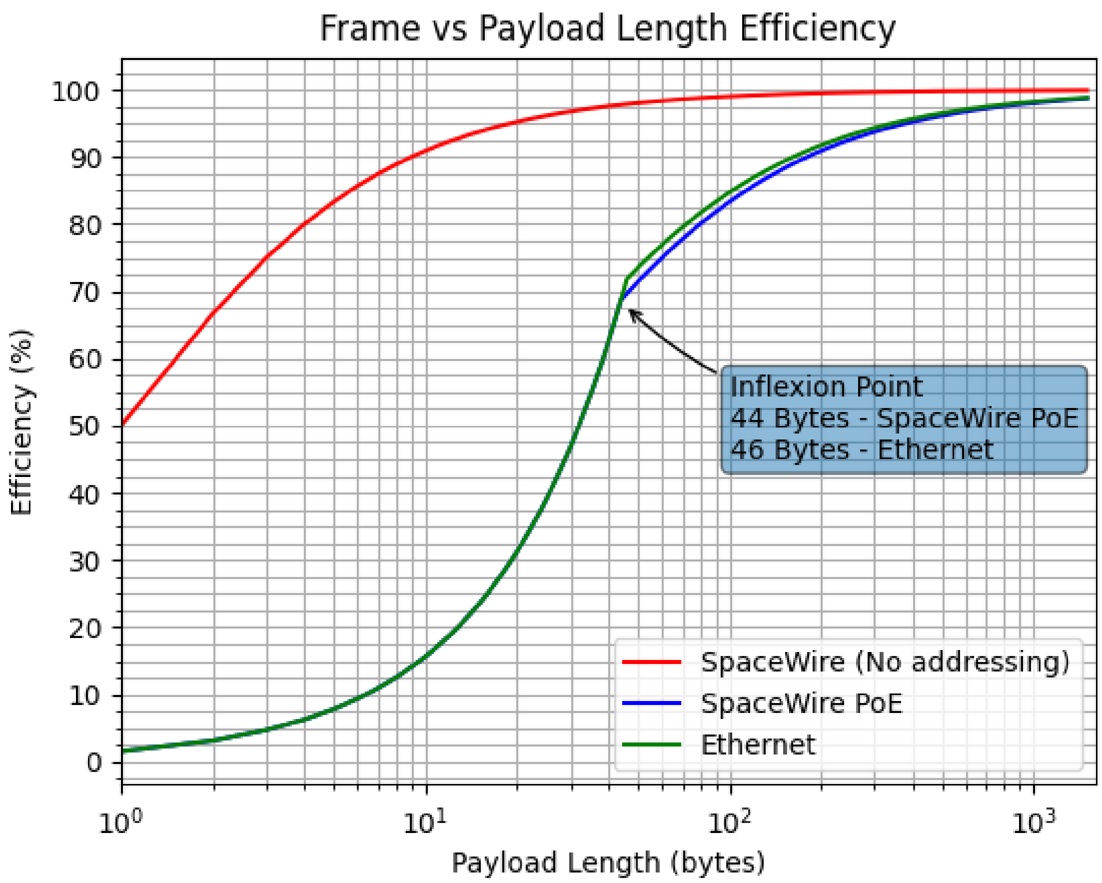


Figure 32 - SpaceWire PoE vs SpaceWire vs Ethernet Frame/Payload Efficiency

The maximum and minimum efficiencies of transmission can be manually calculated using Equation 1 and 2. Equation 1 is used for SpaceWire Packet lengths under or equal to 46 bytes, and Equation 2 for SpaceWire Packet Lengths over 46 bytes.

$$\eta = 100\left(\frac{L_{SpW}-L_{SH}}{L_{eth}}\right) \qquad \ldots \qquad (1)$$

$$\eta = 100\left(\frac{L_{SpW}-L_{SH}}{L_{EH}+L_{SpW}}\right) \qquad \ldots \qquad (2)$$

η = Efficiency (%)

L_{SpW} = Length of SpaceWire Packet (bytes)

L_{eth} = Minimum Length of Ethernet Frame (bytes)

L_{EH} = Length of Ethernet Frame Headers (bytes)

L_{SH} = Length of SpaceWire PoE Headers (bytes)

Efficiency Calculation	L_{SpW}	L_{eth}	L_{EH}	L_{SH}	η (%)
Maximum	1500	N/A	18	2	98.68
Minimum	3	64	N/A	2	1.56

Table 4 - SpaceWire PoE Max and Min Transmission Efficiencies

Table 4 shows the values used to derive the theoretical maximum and minimum efficiencies that the SpaceWire PoE system could achieve. Therefore, the efficiency (η) of sending a single SpaceWire Packet within a SpaceWire PoE system can be expressed as shown in Equation 3.

$$1.56 \leq \eta \leq 98.68 \qquad \ldots \qquad (3)$$

4.4.2 Multiple SpaceWire Packets Per Frame Efficiency

For more efficient use of the Ethernet frames inhibited within the SpaceWire PoE implementation, multiple SpaceWire Packets can be inserted (Figure 29) per frame. In the instance of multiple SpaceWire Packets inserted into the Ethernet frame of varying length and type (N-Char or C-Code), the efficiency cannot be assumed to increase in a linear behaviour similar to that demonstrated in
Figure **32**. To simulate the volatility of efficiency with respect to the number SpaceWire Packets that can be sent within an Ethernet frame, a script was generated that randomises the length of the Ethernet Frame payload and subsequently the length of each SpaceWire packet to be inserted within the frame.

The results shown in Figure 33 are split into two data-sets for comparison, that being the size of the SpaceWire Packets being inserted into the Ethernet frames are between the ranges of 1-1498 bytes for the top plot and 1 – 250 bytes for the bottom plot. This was to demonstrate the disparity in efficiency depending on the use-case for the

SpaceWire PoE System, in relation to sending larger packets for video/images or sending smaller packets for telemetry data.

For both plots it can be observed that for SpaceWire Data under or equal to 44 bytes that the variance in efficiency is significant with no correlation with the number of packets sent. The use-case of sending larger packets of SpaceWire Data (1-1498 bytes) results in an efficiency nominally between 90-100%; this is due to requirement of fewer header-bytes and EOP/EPP Markers within the frame, increasing the ratio of SpaceWire Data to the total frame length. Achieving an equivalent efficiency of nominally between 90-100% for smaller packet sizes is achieved when ten or more are inserted into the ethernet frame. From this analysis it can inform whether the use of SpaceWire PoE is suitable for the individual use-case, performing at an acceptable level of efficiency.

Due to the random nature of the data being generated for the plots, the specific points of interest of change of rate of efficiency, in particular the inflexion point shown in Figure 32, cannot be visualised. This explains why there is an observable lack of density between the plotted data points of the blue and red markers. Further analysis is required to evaluate the extremities of the inflexion point and minimum/maximum efficiencies possible.

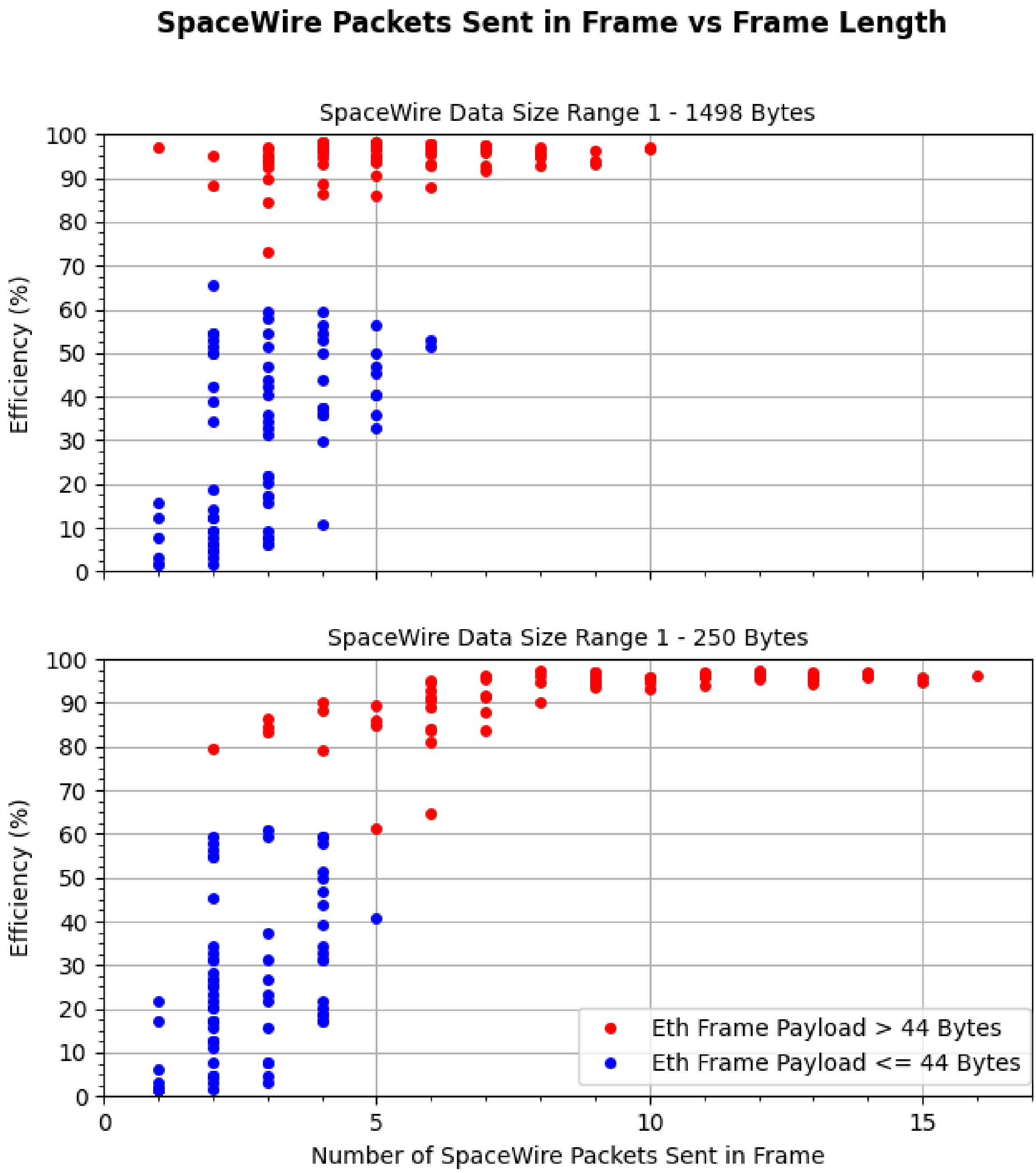


Figure 33 - Multiple SpaceWire Packets vs Frame Length Efficiency

4.4.3 Power Usage

The total amount of power required to transmit and receive SpaceWire data via the SpaceWire PoE protocol can be estimated from the use of Table 5 (Sivaraman *et al.*, 2011). It states the estimated power or energy consumption of the operation of an ethernet network and the transmission of ethernet packets over a network.

Energy component and description	Our estimate
Power consumed by unconnected NetFPGA card (P_C)	6.936 W
Power consumed per connected Ethernet port (P_E)	1.102 W
Per-Packet processing energy (E_p)	197.2 nJ
Per-Byte energy (E_b)	3.4 nJ
Per-Byte Ingress storage energy (E_{rs})	0.8nJ
Per-Byte Egress storage energy (E_{ts})	1.6 nJ
Per-Byte transmit/receive energy (E_{tx}, E_{rx})	0.5 nJ

Table 5 - Ethernet Estimated Power and Energy Consumption

The total amount of power to process an ethernet packet can be approximated using Equation 4 and P_n being 8127. The value of 1.6mW can be an underestimate due to this calculation considering the least amount of packets sent per-second, therefore for high-traffic consisting of small payloads this value could increase. The provided energy value (Table 5) includes the processing of IP headers which is not applicable to SpaceWire PoE, although the implemented protocol inhibits more bytes as headers for processing the data, especially in the instance of multiple SpaceWire Packets within an Ethernet Frame. Therefore the cited value is acceptable approximation for these theoretical calculations.

$$P_p \approx \frac{E_p P_n}{t} \approx \frac{197.2 \times 8127}{1} \approx 1.6mW \quad \ldots \qquad (4)$$

P_p = Total Power to process packets (mW)

E_p = Per Packet Processing Energy (nJ)

P_n = Total number of packets per second (Ns^{-1})

t = Total time (s)

The amount of energy required for the entire lifecycle of an ethernet byte is shown in Equation 5 (Sivaraman *et al.*, 2011); the definitions of each constant are given in Table 5.

$$E_b = (E_{rx} + E_{rs}) + (E_{tx} + E_{ts}) \qquad \ldots \qquad (5)$$

To derive the estimated maximum total power for the transmission of ethernet bytes, Equation 6 was used with P_n being 8127, to convert the given energy value to power consumption for the maximum amount of bytes per second for a Link Speed of 100Mbps.

$$P_b \approx \frac{E_b P_s P_n}{t} \approx \frac{3.4 \times 10^{-9} \times 1500 \times 8127}{1} \approx 41.45mW \quad \ldots \qquad (6)$$

P_b = Total Power of Bytes (mW)

E_b = Energy per Byte (nJ)

P_s = Packet Payload Size

P_n = Packets Per Second (Ns^{-1})

t = Total time (s)

From the derived power values for the processing and transmission of ethernet data, the total power for a SpaceWire PoE system can be expressed as shown in Equation 7.

$$P_T \approx P_C + NP_E + N(P_p + P_b) \qquad \ldots \qquad (7)$$

P_T = Total Power Consumption (W)

P_C = Power Consumption of Power Sourcing Equipment (W)

P_p = Total Power to process Packets (W)

P_E = Power Consumed by Ethernet Port (W)

P_b = Total Power of Bytes (W)

N = Number of Ethernet Nodes in System

Table 6 shows the approximated power consumption of the implemented SpaceWire PoE system for varying data-rates. The value of P_C used is an estimate from the use of the electrical specifications of the power supply for the Ethernet Network Switch. The values of P_p and P_b are adjusted by a factor of 10 to consider the change in energy consumption due to the increase or decrease in time to transmit and receive a byte or process a packet for the data-rates of 10Mbps and 1000Mbps. The number of nodes N reflects that the PSE shall be providing power to two nodes. The power consumption of the embedded system has not been considered in this calculation, although it is assumed to be of negligible impact.

Link Speed (Mbps)	Power Consumption of PSE, P_C (W)	Power Consumption of Ethernet Port, P_E (W)	Total Power to Process Packets, P_p (W)	Total Power for Bytes, P_b (W)	No. of Nodes, N	Total Power Consumption, P_T (W)
100	4	1.102	0.00016	0.004145	2	6.21
100	4	1.102	0.0016	0.04145	2	6.29
1000	4	1.102	0.016	0.4145	2	7.065

Table 6 - Total Estimated Power Consumption Calculation

The results from Table 6 are graphically presented in Figure 34. It can be observed that the power consumption to establish and maintain the ethernet link is responsible for the majority of the power consumption and the transmission of SpaceWire Data is only minimal in comparison. The variance of data-rates complements the findings in (Sivaraman *et al.*, 2011) that the higher the data-rate the higher the total power consumption.

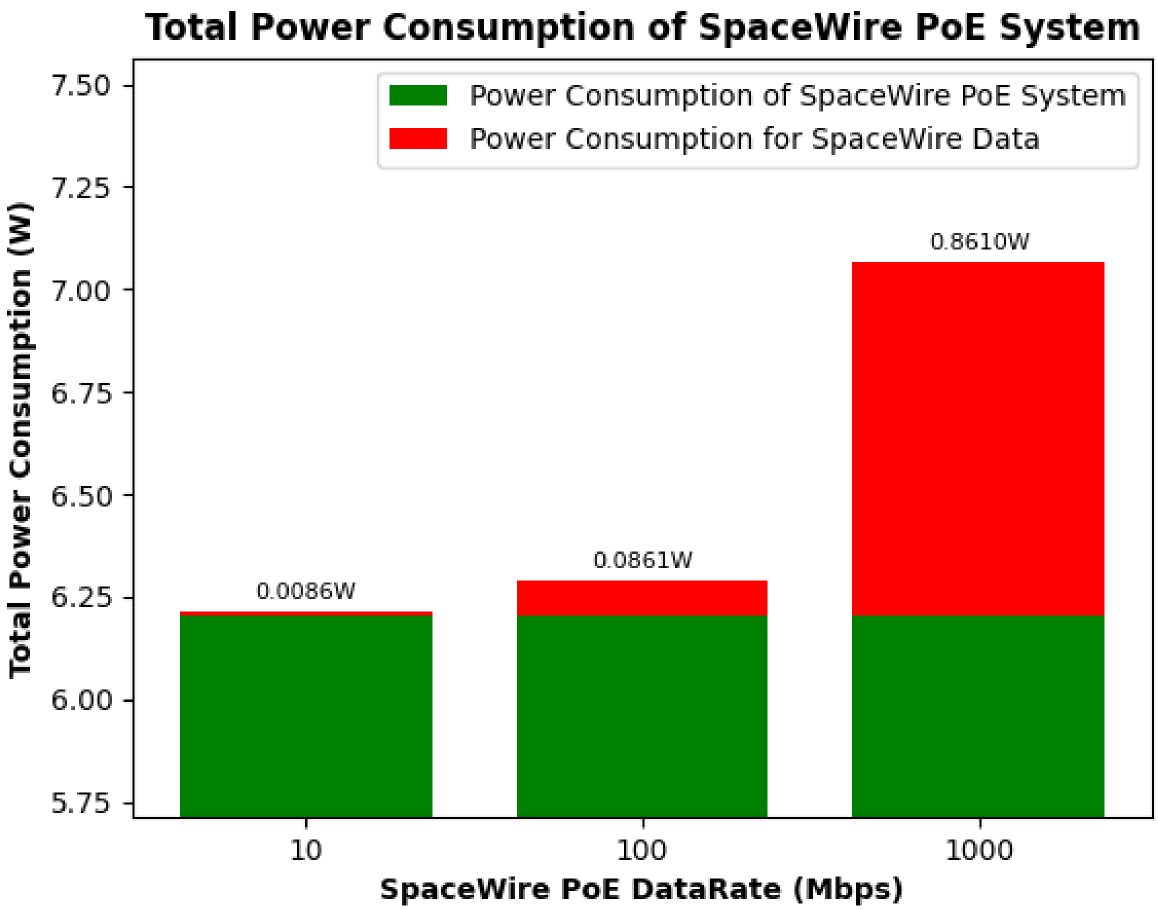


Figure 34 - SpaceWire PoE Power Consumption Analysis

4.4.4 Bandwidth

The maximum bandwidth that can be achieved in the link of each protocol can be derived with the use of Equation 8 and 9. To directly compare the protocols the data-link speed was benchmarked at 100Mbps for all protocols; it is to be noted that SpaceWire is capable of data-link speeds of between 2 – 200 Mbps and up to 400+ in specific cases.

$$P_n = \frac{L_S}{P_S \times B} \qquad \ldots \qquad (8)$$

$$BW = P_n \times P_S \times B \qquad \ldots \qquad (9)$$

P_n = No. of Packets per second (Ns^{-1})

L_s = Data-Link Speed (Mbps)

P_s = Size of Packet (Bytes)

B = Size of Byte (Bits)

BW = Bandwidth (Mbps)

Protocol	Data-Link Speed, L_s ($Mbps$)	Packet Size, P_s	Size of Bytes, B (bits)	No. of Packets per Second, P_N
Ethernet	100	1538	8	8127
SpaceWire	100	1538	8	124
SpaceWire PoE	100	100001	8	8127

Table 7 - Packets/Frames per Second Calculations

Protocol	Number of Packets per Second, P_N	Payload Size, P_s (bytes)	Size of Bytes, B (bits)	Bandwidth, BW (Mbps)
Ethernet	8127	1500	8	97.52
SpaceWire	124	100000	8	99.2
SpaceWire PoE	8127	1498	8	97.39

Table 8 - Bandwidth Calculations

The maximum data-rate achieved by each protocol is as shown in Table 7 and Table 8. It can be observed that in the context of the maximum utilisation of the data-links, SpaceWire is faster and subsequently more efficient by a margin of 1.81% in comparison to an Ethernet based protocol. The margin would increase linearly as the data-link speed of the SpaceWire link is increased in speed.

4.4.5 Latency

The additional latency of delivery for SpaceWire data for the ethernet based protocols can be theoretically approximated with the use of Equation 11.

$$t_l = \frac{BH}{L_s} \quad \ldots \quad (11)$$

H = Size of Headers (bytes)

B = Size of Byte (bits)

L_s = Data-Link Speed (Mbps)

Protocol	Packet Header Size, H (bytes)	Data-Link Speed, L_s (Mbps)	Size of Byte (bits)	Latency, t_l (µs)
Ethernet	26	100	8	2.08
SpaceWire PoE	28	100	8	2.24

Table 9 - Latency Calculations

The additional Latency (t_l) can be approximated to be 2µs for both. This could only present an issue if the SpaceWire PoE Network is utilising time-codes for system synchronisation purposes. The approximated latency does not also factor in the 32-bit CRC for ethernet frames, although this is anticipated to be in the nanosecond scale so does not significantly impact the calculated approximations.

4.5 Issues

From the Literature Review, System Development and System Testing that has been conducted, key observations and issues have been identified with the implemented SpaceWire PoE system.

4.5.1 Efficiency Volatility

From the theoretical analysis conducted on the SpaceWire PoE system, it was identified that the efficiency of sending SpaceWire Data is not linear if sending multiple SpaceWire

Packets within a single ethernet frame. The volatility in efficiency as demonstrated in Figure 33 is due to the insertion of artificial header-bytes to distinguish the type of data being sent (N-Char or C-Code). The SpaceWire PoE system is not optimal for mission system architectures that only require the transmission of SpaceWire Data, that consists of nominally smaller sized packets or control-codes that do not utilise the full length of the ethernet frame. The SpaceWire Protocol provides a more efficient solution for the transmission of small sized packets due to the requirement of less headers.

The SpaceWire PoE system is best utilised for sending larger sized packets, that make full use of the Ethernet Frame Size, or use multiple Ethernet Frames to send SpaceWire Packets that are larger than 1498 Bytes for maximum efficiency. This was not within scope of the developed system; it is not hypothesised that this could not be achieved with further development.

The SpaceWire PoE system for small-sized SpaceWire Packets can still be viable if the overall efficiency of the mission system is improved. For example, if the adoption of a SpaceWire PoE mission system architecture loses efficiency in data transmission, although weight savings from reduced cabling eliminate the efficiency offsets.

4.5.2 IEEE 802.3 PoE Functionality Access

A limitation that was identified during the system development phase was the inability to access the PoE functionalities and how these could be optimised to the User's requirements. The nature of how PoE operates is that it is introduced as a passive element where the power identification and control management are achieved through ASIC's in COTS solutions. Therefore for the implementation of a SpaceWire PoE system in reference to the specified User use-case, further development would be required to understand how the PoE function can be optimised to perform as desired.

Figure 35 depicts a conceptual solution to allow the PoE function to be enabled and disabled at the user's discretion, that can be incorporated with COTS solutions. The MOSFETs are driven into inactive or saturation states dependent on the enable or disable signal by the Power Sourcing Equipment to initiate the Detection, Classification and Power Delivery states of PoE. The introduction of additional components on the two twisted-pairs will need to be investigated, whether this has impacts on the compliancy to the Ethernet Standard and also if it effects the system's performance (power losses, impedance mismatching).

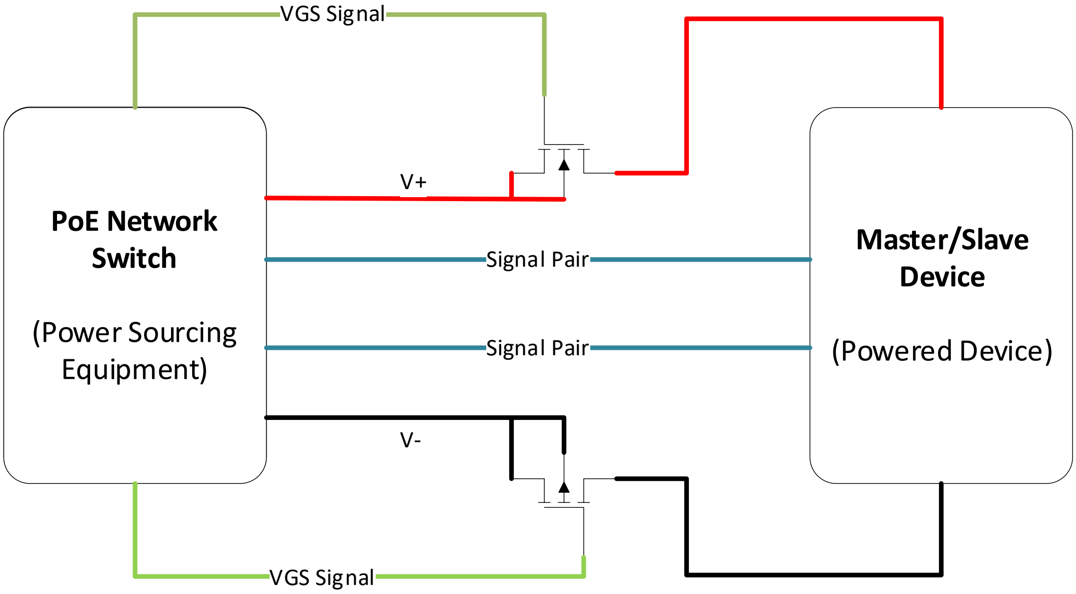


Figure 35 – SpaceWire PoE COTS Solution

4.5.3 Limitations of Testing and Implementation

Due to the limitations of COVID-19 and no access to bespoke hardware and software, in depth technical analysis could not be conducted on the SpaceWire PoE system. In depth analysis of the system's performance characteristics can further inform the engineering judgement of whether the SpaceWire PoE system is viable to be used in mission system architectures.

Functional Testing was limited to visual inspection of terminal windows and embedded devices log report, with the use of byte values of type char (human readable characters). In real-world implementation terms, these would not be adopted, where the SpaceWire PoE header-bytes (Table 2) could be replicated as single bits and the whole byte contains other information such as length and packet number to increase efficiency and usefulness.

4.5.4 Detection of Errors

The handling of error detection is processed in the Data-Link Layer of the SpaceWire PoE system; of which the Ethernet layer is implemented. If an error is to occur (CRC check is wrong) there is no current means to ascertain an error has occurred in the link or data before transmission, therefore not being able to place an EEP Marker (header-byte) in the SpaceWire Packet. For SpaceWire Data that is larger than an ethernet frame in size, this will need to be investigated as to how packets are recovered in the event of errors, although it could be determined that the CRC check is sufficient.

4.6 Summary of Findings

The functional testing of the SpaceWire PoE system validated that the threshold requirements identified in the SRD have been met (fully compliant) through the generated test-cases. The theoretical analysis of the system collectively signifies that there is a degradation in the performance of the system, specifically the efficiency of transmission due to the headers within the ethernet frame. Issues encountered of the SpaceWire PoE system have been identified, which detail the implications and challenges on developing the system to a suitable technology readiness level for future use in the space industry.

5. Project Evaluation

5.1 Project Analysis and Summary

The project was completed within the given time constraints and in accordance with the baselined project schedule generated at the initiation of the project. The project activities carried out aligned to the V-Model Systems Engineering approach (Figure 2); defining the requirements, implementing a developed system, and to test the system to verify and valid if it met the requirements. The project was managed successfully with an appropriate methodology adopted to define and test a conceptual system against the given use-case.

The developed SpaceWire PoE System met all the required Threshold requirements stated in the SRD, although did not achieve any Objective requirements. All the identified aims and objectives were achieved and answered the identified research questions, to provide sufficient information and engineering judgement to develop the required system.

From the analysis conducted on the system; findings, issues and areas of future work were able to be identified and recommended to the project supervisor and Thales Alenia Space UK, of which an additional presentation was held with representatives of Thales Alenia Space UK. Therefore the outcome of this project was that it achieved in supporting an international-consortium partner with Project MOSAR, contributing to the wider studies of future spacecraft design.

5.2 Self-Assessment and Reflection

The process utilised for the management of this project was very effective, providing a clear chronological approach to identifying, implementing, and evaluating a solution. From this project I have further reinforced my skills as a Systems Engineer, specifically the methodology of identification, implementation, and evaluation of user and system requirements. I have further developed skills directly applicable to my Electronic Engineering discipline, these being; Embedded Hardware Development, Scripting Analysis, Technical Report Writing, Software Development (C++, Python), UML/sysML

Engineering Modelling and eliciting technical requirements from professional engineering standards. The project has also further developed my interpersonal and project management skills which include the following; Stakeholder Analysis, Communication Skills, Meeting Minutes, Project and Engineering Planning and Remote Presentation Skills.

The activities associated with project were directly applicable the UK-SPEC Engineering Competencies of a Professional Engineer, specifically the A, B and C competencies, as my project primarily focused on the management and utilisation of new technologies to solve existing problems. The relevancy of the activities conducted during the project were recorded in the Project Logbook which are shown in (Appendix B).

The project has provided me sufficient evidence that in combination with my previous work experience and achievement of a BEng degree, it will support my application of Chartered Engineer Status with the IET. The project also directly will contribute to the overarching study of Project MOSAR, to improve the sustainability and adaptability within the space industry for the next-generation of spacecraft systems, informing the future strategy of spacecraft design. This will also contribute to the environmental challenges faced by the space-industry by having identified a solution that requires less resources throughout the lifecycle of a spacecraft system.

5.3 Future Work and Research

5.3.1 IEEE 802.3 PoE Bespoke Implementation

Section 4.5.2 details the issue of the use of COTS hardware that implements the IEEE 802.3 Power over Ethernet functionalities, that are controlled and managed by ASIC solutions. This prohibits access to optimise the system to behave as would be as required as stated within the given Use-Case by Thales Alenia UK. Therefore future research is required that would seek to develop the SpaceWire PoE system to inhibit a bespoke implementation of the PoE functionality, that allows for the system to be operated as required by the given Use-case, primarily to enable and disable ethernet nodes when required for the transmission of SpaceWire Data.

5.3.2 Network Discovery and Configuration

To support the aim of project MOSAR, further research and testing could be conducted to determine how a SpaceWire PoE system performs under a changing Network Architecture. This could be the detection, link and transmission of data to a changing network consisting of ethernet nodes and how this is managed by a mission system. Research in this area would directly support the current literature that is being generated by the community of interest, which contributes towards the synchronisation and harmonisation of modular spacecraft mission sub-systems.

5.3.3 System Performance Analysis

The scope of the project only set qualitative aims and system requirements criteria. To characterise and understand the performance of the SpaceWire PoE system, qualitative data and testing could be carried out, to better ascertain its viability and limitations for its use-case in a spacecraft mission system. The key quantitative parameters that could be attained include latency, error probabilities, link start-speed, power budget and bandwidth.

5.3.4 Legacy SpaceWire Equipment

It is to be considered that the adoption of new communications protocol within a mission systems architecture will not be interoperable with legacy systems that have adopted solely the SpaceWire protocol. It is required that a study is conducted that ascertains if and how a system that has adopted the SpaceWire protocol can operate with the proposed SpaceWire PoE protocol. This will require a study of the physical, data-link and network layers similar to that conducted in this report, except working in reverse.

5.3.5 SpaceWire PoE Network Layer Improvements

The implementation of the Network Layer can be improved as detailed in Section 4.5.3. The current implementation uses artificial headers that are a byte in size to distinguish different types of packets and the required SpaceWire markers such as the EOP and EPP. Further work can seek to refine the current proposed protocol to reduce the overhead and/or increase the usefulness of the bytes sent within the ethernet frame. Artificial header bytes for the SpaceWire PoE system could include type, packet number, length for a more error resistant and efficient transmissions.

5.3.6 Error Recovery

In the scenario of transmission or link errors of the SpaceWire PoE system, there is no current way to determine when a link has occurred during transmission and the insertion of an EEP as described in Section 4.5.4. Further investigation is required on the requirement of error checking and how the system can recover from segmented packets.

5.4 Recommendations

The implementation of the SpaceWire PoE protocol is dependable on the wider sustainability and savings that are made on the intended spacecraft or mission system. Suitable engineering judgement should be made on the overall gain of implementing a singular system architecture for the transmission of data and distribution of power, where the potential savings of; mass in cables, re-usable resources and reduction in maintenance outweigh that of the loss in efficiency of not solely using SpaceWire Network architectures. It is recommended a study is conducted on future spacecraft and their inhibited mission system use-cases to determine whether the SpaceWire PoE protocol is a viable system to develop further and implement, or whether the current SpaceWire protocol is suitably sufficient.

6. Conclusion

In conclusion, the aims and objectives of the project were achieved. A SpaceWire PoE system was developed and evaluated, of which it was ascertained a combined SpaceWire and Ethernet Protocol Stack is valid and can be implemented with existing COTS Hardware. Issues and Future Work were identified that will better inform the validity of the SpaceWire PoE System deployed on spacecraft, for the purpose of sub-systems inter-communications and power distribution achieved through a singular architecture. It has been recommended that a wider study is conducted to ascertain whether the adoption of a singular architecture as demonstrated with the SpaceWire PoE system, is beneficial and could act as a solution to the engineering and environmental challenges faced by the space industry.

References

Conference (2016) *International SpaceWire Conference*. In: 2016 Yokohoma, Japan: IEEE.

Devices, A. (2005) *A Beginner's Guide to Ethernet 802.3*.

EJR-Quartz (2021) *MOSAR*. Available from: https://www.h2020-mosar.eu/ [Accessed 25 April 2021].

European Cooperation for Space Standardisation - ESA Requirements and Standards Division (2019) *ECSS-E-ST-50-12C Rev.1, SpaceWire - Links, Nodes, Routers and Networks*.

Forcepoint (2020) *What is the OSI Model?* Available from: https://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/%0Ahttps://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/%0Ahttps://www.forcepoint.com/cyber-edu/osi-model#:~:text=The [Accessed 7 April 2021].

IEEE (1995) *IEEE 802.3u-1995 - Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T (Clauses 21–30)*.

IEEE Computer Society (2018) *IEEE Standard for Ethernet 802.3-2018*.

INCOSE (2007) *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*.

Krzysztof, R., Jameux, Piotr, T., Witold, H., Rafał, R., Vangelis, K., Nikos, P. and David, J. (2016) *SpaceWire network managment using network discovery and configuration protocol: Space networks and protocols, short paper*.

Liudmila, K., Elena, S. and Yuriy, S. (2016) *Synchronization of distributed interrupts delivery in aerospace onboard networks: Network and protocols, short paper*.

Man, L.A.N., Committee, S. and Computer, I. (2019) *IEEE Standard for Ethernet Amendment 2: Physical Layer and Management Parameters for Power over Ethernet over 4 pairs*.

Microchip Technology Inc (2019) *Next-Generation PoE: IEEE 802.3bt White Paper*.

OSI (1994) *ISO/IEC 7498-1:1994 - Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*.

Rozanov, V. and Yablokov, E. (2014) *Problems of Developing SpaceWire - Ethernet Bridge and Transferring SpaceWire Packages over Ethernet*.

Sivaraman, V., Vishwanath, A., Zhao, Z. and Russell, C. (2011) Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router. In: *2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011*. 2011 doi:10.1109/INFCOMW.2011.5928833.

Susan, C., Mazen, S. and Angkasa, K. (2016) *Using SpaceWire time codes for spacecraft time synchronisation: SpaceWire missions and applications, short paper*.

Vinogradov, A., Yablokov, E. and Yachnaya, V. (2019) *Upgrade of Ethernet-SpaceWire Protocol*.

Appendices

Appendix A – Project Gantt Chart

Appendix B – Project Management Documents

Appendix C – SpaceWire and Ethernet Full Protocol Stack

Appendix D – SpaceWire PoE System Requirements Document

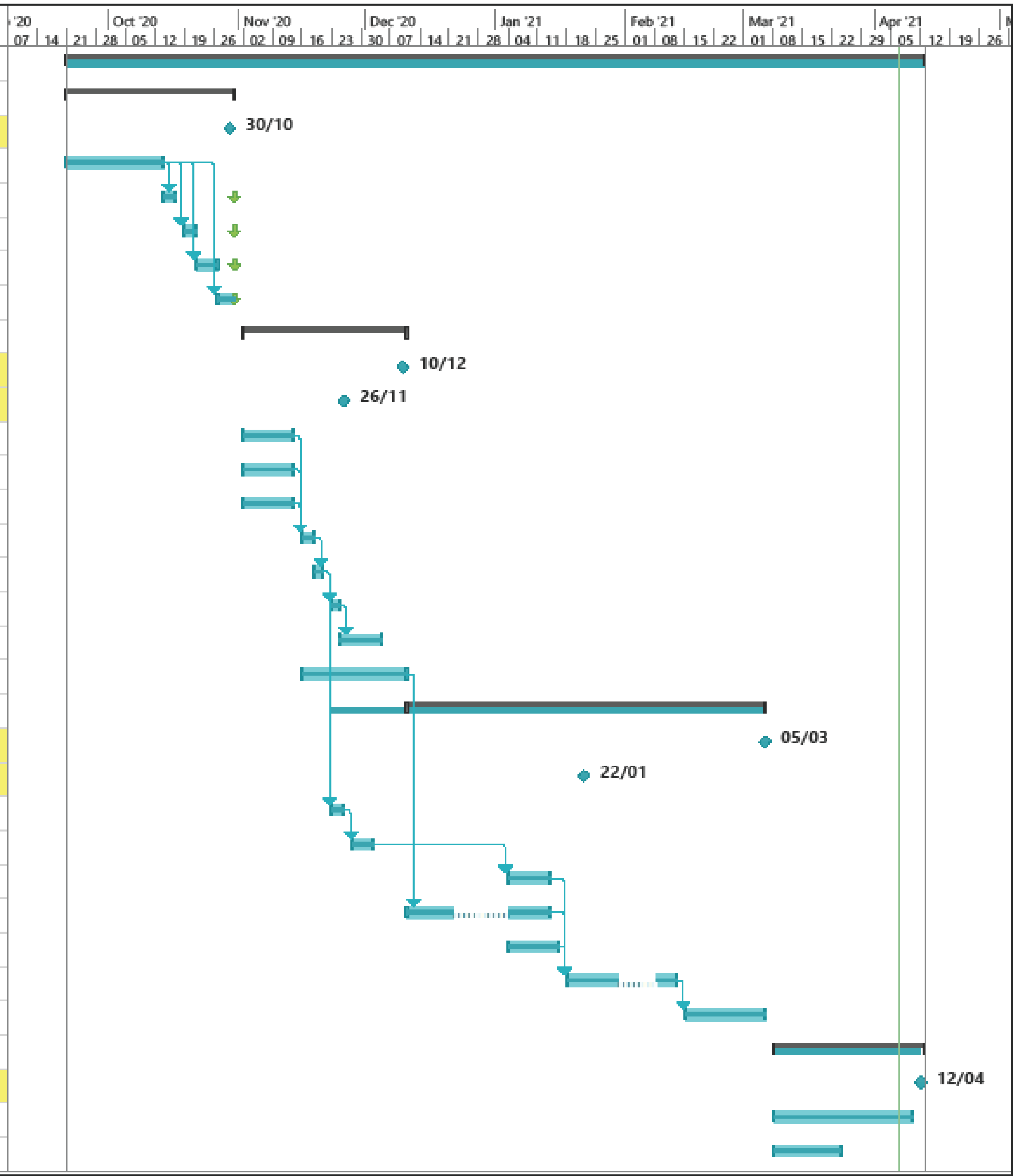
Appendix E – SpaceWire PoE UML Diagrams

Appendix F – Functional Test-Cases

Appendix G – System Functional Testing Results

Appendix A - Project Gantt Chart

ID	Task Name	Duration	Start
1	1 SpaceWire and IEE PoE Individual Project	146 days?	Mon 21/09/20
2	1.1 Project Concept Initiation	30 days	Mon 21/09/20
3	1.1.1 Project Initiation Concept Documentation Submitted and Endorsed	0 days	Fri 30/10/20
4	1.1.2 Project Initiation Concept Proposal uploaded onto Engineering Databas	17 days	Mon 21/09/20
5	1.1.3 Project Ethics Checklist	3 days	Wed 14/10/20
6	1.1.4 Project Resources Form	3 days	Mon 19/10/20
7	1.1.5 Project Security Form	3 days	Thu 22/10/20
8	1.1.6 Project Risk Assessment	4 days	Tue 27/10/20
9	1.2 Project Literature Review and Testing Preparation	29 days	Mon 02/11/20
10	1.2.1 Interim Project Research Proposal Submitted	0 days	Thu 10/12/20
11	1.2.2 Project Hardware and Software Resources Baselined	0 days	Thu 26/11/20
12	1.2.3 ECSS-E-ST-50-12C SpaceWire Review	10 days	Mon 02/11/20
13	1.2.4 IEEE 802.3 Ethernet Review	10 days	Mon 02/11/20
14	1.2.5 IEEE 802.3bt PoE Review	10 days	Mon 02/11/20
15	1.2.6 PoE Development Hardware Research	3 days	Mon 16/11/20
16	1.2.7 Functional Testing System Architecture Diagram	2 days	Thu 19/11/20
17	1.2.8 Project Resources Request Form	2 days	Mon 23/11/20
18	1.2.9 EDM Supplier Order Delivery	8 days	Wed 25/11/20
19	1.2.10 Interim Project Research Proposal Document	19 days	Mon 16/11/20
20	1.3 Project Hardware and Software Development, Testing and Analysis	61 days?	Fri 11/12/20
21	1.3.1 Functional Testing and Analysis Completed	0 days	Fri 05/03/21
22	1.3.2 SpaceWire Hardware & Software Implementation Complete	0 days	Fri 22/01/21
23	1.3.3 Systems Requirement Document	3 days	Mon 23/11/20
24	1.3.4 System Logical Architecture Diagram	4 days	Sat 28/11/20
25	1.3.5 Software UML Diagram	8 days	Mon 04/01/21
26	1.3.6 SpaceWire Communications Protocol Software Development	15 days	Fri 11/12/20
27	1.3.7 Hardware and Software Integration	10 days	Mon 04/01/21
28	1.3.8 SpaceWire Protocol Practical Functional Testing	15 days	Mon 18/01/21
29	1.3.9 Further Development and Testing	15 days	Mon 15/02/21
30	1.4 Project Final Report and Poster	26 days	Mon 08/03/21
31	1.4.1 Submission of Project Dissertation and Presentation	0 days	Mon 12/04/21
32	1.4.2 Project Final Dissertation	25 days	Mon 08/03/21
33	1.4.3 Project Presentation	12 days	Mon 08/03/21



Project: Individual_Project_PoE
Date: Wed 07/04/21

Task, Split, Milestone, Summary, Project Summary, Inactive Task, Inactive Milestone, Inactive Summary, Manual Task, Duration-only, Manual Summary Rollup, Manual Summary, Start-only, Finish-only, External Tasks, External Milestone, Deadline, Progress, Manual Progress

Page 1

Appendix B - Project Management Documents

Project Security Form

Project Resources/Expenditure Form

Risk Assessment Form

Logbook UK SPEC Matrix

Contact Register and Meeting/Actions Log

UK Engineering Council Guidance on Security

Security is an important consideration for any project and in any workplace. UK SPEC provides guidance on security and outlines how it expects professional engineers to conduct themselves with respect to security. This guidance is split into six key principles. Your task is to visit the UK SPEC security page and read through the tips provided for each principle.

Using the tips on the webpage provide at least one example for each of the six principles that you can directly link to your project or that should be considered in the role of Professional Engineer. Then summarise your findings in the comments box below.

The link is https://www.engc.org.uk/security

1. How would you adopt a security-minded approach to your project?

The project originates from Industry that is currently leading the research of compartmentalisation of spacecraft modules, otherwise known as Project MOSAR. The information that is detailed within this project may be commercially sensitive and require the use of an NDA. Therefore I will approach the project with the upmost professionalism and consider the security aspects of the project at the forefront of decision making. With an NDA it provides assurance that potential sensitive information is not distributed to parties or organisations that have no access to the information.

2. What should you consider in order to apply responsible judgement and take a leadership role?

The project may entail communications and the exchange of information between Industry Partners of UWE and myself. Therefore to maintain a good relationship that emphasises on the importance of withholding a good standard of security awareness, the appropriate means of communication will be used depending on the sensitivity of the information. Such as if restricted infrastructure/networks shall be utilised or the exchange of information in a Secure Communications Information Facility (SCIF).

3. How will you ensure that your project complies with legislation and codes, while understanding intent and prepared to seek further improvements?

Legislation and Codes of Conduct will be identified at the initiation of the project and regularly reviewed throughout the duration of the project. An example of such legislation would be the Official Secrets Act 1989 which would restrict the distribution of information, ensuring professionalism and integrity from the outset.

4. How do you ensure good security-minded communications?

The governance for security for a project can be stipulated through the use of a Security Aspects Letter. The Security Aspects Letter details the different elements of a project and states the classification of the information which directs how that information should be treated i.e stored, transmitted, disseminated etc. The letter encourages a security minded approach to the information contained within the project and makes stakeholders aware of the sensitivity of the project.

5. How would you go about understanding, complying with and seeking to improve lasting systems for security governance?

The security aspects of a Product, System or Service should be considered at the earliest convenience to appropriately manage and mitigate any security risks that are identified. Governance can be provided by implementing a Security Management Plan which identifies the Roles and Responsibilities for a project and programme of work to achieve a desired outcome, and to also regularly review the security status. With tis governance in place, it provide assurance that a lasting system can be introduced and continue to be considering security awareness throughout its lifecycle.

6. What do you need to consider when contributing to public and professional awareness of security?

The UK SPEC identifies the ability of recognising the social, political and economic implications are important to identify the security risks. An example of where you might need to consider the awareness of security for the public may be when using Personal Data that can aggregated into identifying an individual or organisation.

<u>Summarise how you will address security in your project</u> *(max 200 words)*

Security awareness will be a key overarching principle throughout the project, to identify and mitigate accordingly any security related risks that may arise. If deemed appropriate a Security Risk Register will be generated and monitored throughout the project and reported using the appropriate reporting tools via the Project Supervisor.

Information will primarily be open-source although the aggregation of data and commercial strategy may potentially be commercially-sensitive. Information will be collated and confidential where necessary to avoid the risk of disseminating confidential information.

STUDENT FINAL YEAR PROJECT – EXPENDITURE/ORDER REQUEST

STUDENT'S NAME: Jack Martin	COURSE TITLE Electronic Engineering BEng(Hons)
NUMBER: 16031517	PROJECT SUPERVISOR PRINT NAME: Yaseen Zaidi
PROJECT TITLE: The ECSS SpaceWire Software Protocol implemented into IEEE 802.3 Power over Ethernet (PoE)	
EMAIL: jack3.martin@live.uwe.ac.uk	SIGNATURE:
MOBILE: 07930270505	DATE: 20 October 2020

PLEASE EMAIL REQUEST AS EARLY AS POSSIBLE AND LEAVE A CONTACT E-MAIL ADDRESS OR TELEPHONE NUMBER SO THAT WE CAN LET YOU KNOW ABOUT ANY POTENTIAL SUPPLY PROBLEMS.
PLEASE NOTE THERE IS A £75.00 LIMIT

To avoid delay please show the suggested Supplier but please include full details e.g. reference number, dimensions, colour etc. each sheet must be signed and dated by your Project Supervisor then Email to 2Q30 fet.pr@uwe.ac.uk
Attach links to web order in the Email, separately to the scanned form.

When orders are received we will contact you when ready for collection from the Project Room (2Q30)

QTY	DESCRIPTION	SUPPLIER	CATALOGUE Part No	ORDER NUMBER	PARTS Received	PARTS Collected	COST Incl. VAT
2	ESP32 Ethernet Kit V1.2	Mouser UK	356-ESP32ETHERNETKI				£50.32
1	Ethernet Network Switch	CPC Farnell	CS32096				£25.01
1	Cat 6 Ethernet Cable Black 1m	RS Components	411-227				£2.92
1	Cat 6 Ethernet Cable Yellow 1m	RS Components	411-520				£2.92
1	Cat 6 Ethernet Cable Red 1m	RS Components	411-374				£2.92
1	Startech Male USB A to Male USB Micro B USB Cable, USB 2.0, 2m	RS Components	186-2794				£4.56
TOTAL							£148.97

University of the West of England
BRISTOL

GENERAL RISK ASSESSMENT FORM

Describe the activity being assessed: Functional Testing of an Embedded System Communications Protocol in a home environment	**Assessed by: Jack Martin**	**Endorsed by:**
Who might be harmed: Jack Martin (Student) **How many exposed to risk:** 1	**Date of Assessment: 12/10/2020**	**Review date(s): 10/11/2020**

Hazards Identified ***(state the potential harm)***	**Existing Control Measures**	**S**	**L**	**Risk Level**	**Additional Control Measures**	**S**	**L**	**Risk Level**	**By whom and by when**	**Date completed**
Non-Ionising Radiation (RADHAZ including Medical Devices and Personnel)	CE Marked (SAR Testing) CEMFAW Compliance through SAR Testing	2	1	2	None Required. **ALARP**	2	1	2	N/A	12/10/20
Hazardous Materials (Batteries and Manufacturing Materials)	CE Marked Development Boards/Equipment RoHS Compliance achieved compliance demonstrated through CE Marking	3	1	3	None Required. **ALARP**	3	1	3	N/A	12/10/20
Electrical Sources (Mains 240V AC)	CE Marked Equipment (EU/UK Legislation Compliance) Electrical Sources enclosed by Design and hazard only present if damaged. IEEE 802.3 Compliant Equipment	N/A	N/A	N/A	Hazard/Accident scenario deemed in-credible. **Delete and Retain**	N/A	N/A		N/A	12/10/20

Electrical Sources (<50VDC)	No voltages are exposed by design. CE Marked Products. Circuit Protection by design. OEM Manuals, Warnings and Cautions	2	1	1	None Required. **ALARP**	2	1	2	N/A	12/10/20
Cables causing Trips and Falls	None. Not anticipated that the use of the cables will cause an accident.	N/A	N/A	N/A	Hazard/Accident scenario deemed in-credible. **Delete and Retain**	N/A	N/A		N/A	12/10/20
Incorrect Operation of Equipment (Lack of EMC Compliance) – Smoke, Fumes, Electrical Failure etc.	CE Marked Development Boards for Radiating Element (ESP32 Module) Development Board FCC Certified Interfacing equipment suitable for use-case	2	1	2	The Hazard Accident is very unlikely due to the current control measures in place (Design and Certification). Although lack of assurance via EU EMC compliance pertains the hazard to remain Open. **ALARP.**	2	1	2	N/A	10/11/20

RISK MATRIX: (To generate the risk level).

Very likely 5					
Likely 4					
Possible 3					
Unlikely 2					
Extremely unlikely 1		**(3)**	**(1)**		
Likelihood (L) ↑ Severity (S) →	**Minor injury – No first aid treatment required 1**	**Minor injury – Requires First Aid Treatment 2**	**Injury - requires GP treatment or Hospital attendance 3**	**Major Injury 4**	**Fatality 5**

ACTION LEVEL: (To identify what action needs to be taken).

POINTS:	RISK LEVEL:	ACTION:
1 – 2	NEGLIGIBLE	No further action is necessary.
3 – 5	TOLERABLE	Where possible, reduce the risk further
6 - 12	MODERATE	Additional control measures are required
15 – 16	HIGH	Immediate action is necessary
20 - 25	INTOLERABLE	Stop the activity/ do not start the activity

Chartered Engineer			Evidence
A	Chartered Engineers shall use a combination of general and specialist engineering knowledge and understanding to optimise the application of advanced and complex systems The applicant shall demonstrate that they:		
A1	have maintained and extended a sound theoretical approach to enable them to develop their particular role	• Formal training, post-graduation related to their role • Technical training to develop knowledge for a new role • Understanding of the current and emerging technology and technical best practice in their area of expertise • Development of a broader and deeper knowledge base through research and experimentation. • Learning and developing new engineering theories and techniques in the workplace. • Learning and developing new engineering knowledge in a different industry or role	Utilisation of the Systems Engineering V-Model to inform project methodology and associated activities throughout project lifecycle. Communications theory of the OSI Model used as reference to inform approach of requirements elicitation. Developed understanding of modelled based systems engineering (sysML and UML) and implemented methodology for the development of the project. Attained skills and experience to further the role of a systems engineer

A2	are developing technology solutions to unusual or challenging problems using their knowledge and understanding and/or dealing with complex technical issues or situations with significant levels of risk	• Carrying out technical research and development • Developing new designs, processes or systems based on new or evolving technology • Carrying out complex and/or non-standard technical analyses • Developing solutions involving complex or multi-disciplinary technology. • Developing and evaluating continuous improvement systems. • Developing solutions in safety-critical industries/applications	Technical review of the IEEE 802.3 and ECSS SpaceWire Standards Generation of UML Models for the SpaceWire PoE system Deploying new technology of Power over Ethernet into existing technology SpaceWire PoE Theoretical Analysis using Python Scripts Development of SpaceWire PoE system using SRD, UML Diagrams and other means

	Chartered Engineer		Evidence
B	Chartered Engineers shall apply appropriate theoretical and practical methods to the analysis and solution of engineering problems. The applicant shall demonstrate that they:		
B1	take an active role in the identification and definition of project requirements, problems and opportunities	• Identifying projects or technical improvements to products, processes or systems • Preparation of specifications taking account of functional and other requirements • Establish user requirements • Reviewing specifications and tenders to identify technical issues and potential improvements • Carrying out technical risk analysis and identified mitigation measures • Consider and implement new and emerging technologies	Attained User Use-Case to identify system requirements Define SpaceWire PoE Systems Requirements Documents (SRD) Identified dependencies, limitations and constraints that would effect the project The use of continuously emerging technology such as Power over Ethernet for current engineering challenges Identified the testing limitations and therefore set the appropriate qualitative project objectives
B2	can identify the appropriate investigations and research needed to define the work required to complete an engineering task and conduct these activities effectively	• Identifying and agreeing appropriate research methodologies • Investigating a technical issue, identifying potential solutions and determining the factors needed to compare these • Identifying and carrying out physical tests or trials and analysing and evaluating the results • Carrying out technical simulations or analysis • Preparing, presenting and agreeing design recommendations, with appropriate analysis of risk, and taking account of cost, quality, safety, reliability, appearance, fitness for purpose, security (including cyber-security), intellectual property (IP) constraints and opportunities, and environmental impact	Identified the research questions to inform the System Development and Testing/Assessment phases Comparative analysis of the existing solutions and proposed solution (Ethernet vs SpaceWire vs SpaceWire PoE) Theoretical Analysis conducted with scripting tools (Python/MATLAB) Project Risk Assessment and appropriate Tolerable and ALARP sentencing

Chartered Engineer			Evidence
B3	can implement engineering tasks and evaluate the effectiveness of engineering solutions	• Ensuring that the application of the design results in the appropriate practical outcome • Implementing design solutions, taking account of critical constraints, including due concern for safety, sustainability and disposal/decommissioning • Identifying and implementing lessons learned • Evaluating existing designs or processes and identifying faults or potential improvements including risk, safety and life cycle considerations • Actively learning from feedback on results to improve future design solutions and build best practice	Project Tasks identified and sorted into Project Schedule (Gantt Chart) Verification and Validation Testing to analyse the effectiveness of proposed SpaceWire PoE system Project Risk Assessment ALARP Level identified Communications theory adopted and implemented for solution Future Areas of Research identified to progress solution

Chartered Engineer			Evidence
C	Chartered Engineers shall provide technical and commercial leadership. The applicant shall demonstrate that they:		
C1	plan the work and resources needed to enable effective implementation of a significant engineering task or project	• Systematically review the factors affecting the project implementation including safety, sustainability and disposal/decommissioning considerations • Carrying out a task or project risk assessment and identifying mitigation measures • Lead on preparing and agreeing implementation plans and method statements • Negotiating and agreeing arrangements with customers, colleagues, contractors and other stakeholders, including regulatory bodies • Ensuring that information flow is appropriate and effective	Project Schedule Baselined and updated Project Resources identified and obtained
C2	manage (organise, direct and control) programme or schedule, budget and resource elements of a significant engineering task or project	• Operate or define appropriate management systems including risk register and contingency systems • Managing the balance between quality, cost and time • Monitoring progress and taking appropriate actions when required • Establishing and maintaining appropriate quality standards within legal and statutory requirements • Interfacing effectively with customer, contractors and other stakeholders	Safety Risk Assessment conducted Managing Project Budget, justification of using more than allocated budget for the purposes of the project objectives Regular meetings held with Supervisor to update progress and direct future work Meetings with Thales Alenia Space UK to detail progress and advice on further work

Chartered Engineer			Evidence
C3	lead teams or technical specialisms and assist staff to meet changing technical and managerial needs	• Agreeing objectives and work plans with teams and individuals • Reinforcing team commitment to professional standards • Leading and supporting team and individual development • Assessing team and individual performance, and providing feedback • Seeking input from other teams or specialists where needed and managing the relationship	Project Proposal Document Agreed Scope with Supervisor Approach Thales Alenia Space UK for Use-Case
C4	bring about continuous improvement and promote best practice	• Promoting quality throughout the organisation and its customer and supplier networks • Developing and maintaining operations to meet quality standards e.g. ISO 9000, EQFM, balanced scorecard • Supporting or directing project evaluation and proposing recommendations for improvement. • Implementing the results of lessons learned	

Chartered Engineer			Evidence
D	Chartered Engineers shall demonstrate effective communication and interpersonal skills. The applicant shall demonstrate that they:		
D1	communicate effectively in English with others at all levels.	• Preparing reports, drawings, specifications and other documentation on complex matters • Leading, chairing, contributing to and recording meetings and discussions • Exchanging information and providing advice to technical and non-technical colleagues. • Engaging or interacting with professional networks	Final Dissertation Report Interim Proposal Document Viva presentation Project Supervisor interactions
D2	present and discuss proposals, justifications and conclusions clearly	• Contributing to scientific papers or articles as an author • Preparing and delivering presentations on strategic matters • Preparing bids, proposals or studies • Identifying, agreeing and leading work towards collective goals	Publication of Final Dissertation Paper to EASA endorsed Journal Project presentation to Thales Alenia Space UK on findings and issues

Chartered Engineer			Evidence
D3	demonstrate personal and social skills and awareness of diversity and inclusion issues	• Knowing and managing own emotions, strengths and weaknesses • Being confident and flexible in dealing with new and changing interpersonal situations • Identifying, agreeing and working towards collective goals • Creating, maintaining and enhancing productive working relationships, and resolving conflicts. • Being aware of the needs and concerns of others, especially where related to diversity and equality	

Chartered Engineer			Evidence
E	Chartered Engineers shall demonstrate a personal commitment to professional standards, recognising obligations to society, the profession and the environment. The applicant shall demonstrate that they:		
E1	understand and comply with relevant codes of conduct.	• Demonstrating compliance with the code of conduct of your Professional Engineering Institution • Leading work within all relevant legislative and regulatory frameworks, including social and employment legislation.	
E2	understand the safety implications of their role and can apply safe systems of work	• Identifying and taking responsibility for own obligations for health, safety and welfare issues • Ensuring that systems satisfy health, safety and welfare requirements • Developing and implementing appropriate hazard identification and risk management systems and culture • Managing, evaluating and improving these systems • Applying a sound knowledge of health and safety legislation, for example; HASAW 1974, CDM regulations, OHSAS 18001:2007 and company safety policies. OHSAS 18001:2007 and company safety policies.	Project Risk Assessment

Chartered Engineer			Evidence
E3	understand the principles of sustainable development and apply them in their work	• Operating and acting responsibly, taking account of the need to progress environmental, social and economic outcomes simultaneously • Recognising how sustainable development principles can be applied in your day to day work • Providing products and services which maintain and enhance the quality of the environment and community, and meet financial objectives • Understanding and securing stakeholder involvement in sustainable development • Using resources efficiently and effectively in all activities	Project contributes towards the improvement of environmental sustainability of spacecraft systems

E4	carry out and record Continuing Professional Development (CPD) necessary to maintain and enhance competence in own area of practice	• Undertaking reviews of own development needs • Planning how to meet personal and organisational objectives • Carrying out planned and unplanned CPD activities • Maintaining evidence of competence development • Evaluating CPD outcomes against any plans made • Assisting others with their own CPD.	

Chartered Engineer			Evidence
E5	understand the ethical issues that may arise in their role and carry out their responsibilities in an ethical manner	• Understanding the ethical issues that you may encounter in your role • How you have dealt with a specific ethical issue • Giving an example of where you have applied ethical principles as described in the Statement of Ethical Principles published by the Engineering Council and the Royal Academy of Engineering which lists four fundamental principles to guide engineers and technicians in achieving the high ideals of professional life: ➢ Honesty and integrity ➢ Respect for life, law, the environment and public good ➢ Accuracy and rigour ➢ Leadership and communication www.engc.org.uk/professional-ethics • Giving an example of where you have applied/upheld ethical principles as defined by your organisation or company	Ethics Checklist completed at project initiation

Contact Register			
Meeting	**Date**	**Attendees**	**Actions**
Initial Project Meeting	01/10/2020	Jack Martin Yaseen Zaidi	20201001/01 20201001/02 20201001/03
Project Proposal And Ethics Checklist	13/10/2020	Jack Martin Yaseen Zaidi	20201013/01
Project Resources and Progress	03/11/2020	Jack Martin Yaseen Zaidi	20201103/01
Project Progress and Resources	10/11/2020	Jack Martin Yaseen Zaidi	
Interim Proposal Report and Progress Meeting	01/12/2020	Jack Martin Yaseen Zaidi	20201201/01
PoE Use-Case	02/12/2020	Yaseen Zaidi Matthew Rowlings Michael Walshe	
Project Baseline Progress and Future Work Meeting	25/02/2021	Yaseen Zaidi Jack Martin	
Project Viva	23/03/2021	Yaseen Zaidi Jack Martin	
Project Findings Presentation	25/03/2021	Jack Martin Yaseen Zaidi Matthew Rowlings Michael Walshe	

Meeting	Date	Attendees	Records of Discussion	Actions
Initial Project Discussion	01/10/2020	Jack Martin Yaseen Zaidi	Introduction from JM regarding career history, BEng and general interests with regards to Project Concepts and ideal work. YZ introduction on background and project inspiration. Discussion was held on the project and how it can be tailored to the electronics discipline and specifically embedded systems. YZ explained premise of demonstrating the functional proof of SpaceWire protocol implemented into PoE IEEE 802.3bt to allow subsystems to communicate on a CAN bus network architecture. It could start simply just by sharing telemetry data between nodes, and then further onto more complex messages and mission scenarios. YZ stated there is a potential for the use of Thales facilities to aid in the project. JM asked if there was the possibility to include Software Reliability and Validation to the project, YZ responded to confirm it would be beneficial to include this such as DO-178C. Closing remarks of the meeting consisted of JM thanking YZ for his time YZ stating to provide all project documentation to him when ready to submit the project to the engineering database.	**20201001/01** JM Open **20201001/02** YZ Open **20201001/03** JM Open
Project Proposal and Ethics Checklist	13/10/2020	Jack Martin Yaseen Zaidi	JM and YA discussed the projects activities and content of work in relation to UWE's Ethics Checklist. It was concluded that the project inhibits minimal risk and can be uploaded to the Engineering Database. JM presented to YA the potential system architecture for functional testing of the SpaceWire Protocol. JM stated that for PoE 802.3bt	**20201001/01** JM Closed **20201013/01** JM Open

			development hardware and supporting components will most greatly exceed the £75 budget. YZ advised to reduce the scope of the functional testing to acquire cheaper components.	
Project Resources and Progress	03/11/2020	Jack Martin Yaseen Zaidi	JM and YZ discussed the implications of the requested components for functional testing being over-budget and the subsequent viable options if full funding could not be attained. YZ stated to JM the objective to submit an email to the project office to receive full funding for all components. The scope of the project was re-evaluated and refined to avoid duplication/convolution with other related project currently under the supervision of YZ. The project was confirmed to be specific on the use of the SpaceWire Protocol Network Layer and its implementation on hardware. JM informed YZ of the submission of Project Artefacts, these were to be emailed to YZ at the earliest convenience.	**20201103/01** JM Open
Project Progress and Resources	10/11/2020	Jack Martin Yaseen Zaidi	JM and YZ discussed the outstanding form for the hardware requested to be signed off by the appropriate signatory to allow items to be sent to the UK. JM asked for guidance on the use of sysML and UML for the development phases of the project. YZ advised to focus on the implementation of the protocol software and UML diagram, although the sysML model for physical architecture is desirable. JM and YZ discussed the projects overarching aims and how to	

			verify and validate the requirements. It was decided that the implementation of the SpaceWire Protocol and if it meets the need of a simplified use-case was the best approach.	
Interim Proposal Report and Progress Meeting	01/12/2020	Jack Martin Yaseen Zaidi	JM and YZ discussed the Interim Proposal marking criteria and what extra material is required. YZ proposed the research of surrounding material from the scientific community "SpaceWire Conference 2016" added to the report. YZ to schedule a meeting or e-mail with Thales Alenia UK for further briefing on the topic. JM demonstrated the current UML diagrams that have been generated for the project and asked for feedback. YZ discussed and provided a specific use-case that can be used for the project to inform the testing plans.	 **20201201/01** YZ Open
Project Progress Meeting	02/02/2012	Jack Martin Yaseen Zaidi	JM raised a concern that the Use-Case provided by Thales Alenia is not achievable with the current hardware. JM to state in the report that the PoE Control is for future work. YZ confirmed that is certain requirements cannot be met, focus the report on the problems and how they can be achieved in future work YZ stated that further analysis could be done on potential other solutions, and how these could be implemented in contrast to SpaceWire. Such as CAN 2.0, Mil-Std	
Project Baseline Progress and	25/02/2021	Jack Martin Yaseen Zaidi	JM stated that he has achieved his baseline milestones for the project with the ambition to	

Future Work Meeting		Matthew Rowlings	complete more work where possible. A summary of the findings were presented to YZ and MR. MR offered guidance on the future work of the project and possible work that could be completed prior to the submission on the 29th April. These included researching into areas such as; Power Requirements of Avionics. Bandwidth, Requirements refinement and Physical Interfaces	

Action S/N	Description	Action Owner	Target Date	Comments/Updates
20201001/01	~~JM to research and summarise the initial feasibility of demonstrating the functional proofing of implementing SpaceWire protocol into PoE~~	JM	13/10/20	JM completed initial research and produced draft project schedule. **Complete**
20201001/02	~~YZ to determine if Thales facilities and/or SME guidance is available for use to assist project~~	YZ	30/10/20	03/11/20. Ongoing due to restrictions and lockdown. **Closed/Rejected.**
20201001/03	~~JM to send all required Project Initiation Documentation to YZ prior to submitting to the Engineering Database~~	JM	13/10/20	**Complete**
20201013/01	~~JM to define initial system architecture for functional testing and complete Student Resources Request Forms to acquire required hardware and software~~	JM	23/10/20	Project Request Forms endorsed by YZ and sent On 20/10/20 **Complete**
20201103/01	~~JM to send Project Artefacts to YZ via e-mail.~~	JM	09/11/20	E-mail sent 10th November. **Complete.**
20201201/01	~~YZ to schedule a meeting with Thales Alenia.~~	YZ	07/12/20	**Complete.**

Appendix C - SpaceWire and Ethernet Full Protocol Stack

OSI layer	Layer Implementation	Project Implementation	Description/Function
Application	User GUI/Embedded System	Data Request/Control Signals	Shall act as the User Interface for manual requests of data and control codes to be distributed to SpaceWire Nodes
Presentation	Embedded System	N-Chars, ASCII	Shall take the data from the network layer (packets) and encode them into the appropriate format depending on the purpose. ASCII Characters to be used for human-readable.
Session	Not used.	N/A	
Transport	Not used.	N/A	
Network	SpaceWire	Software Protocol	Shall format the data and control codes received from the User Interface or other SpaceWire Nodes through the data-link layer into the SpaceWire Packet format.
Data-Link	IEEE 802.3 Ethernet	IEEE 802.3 Ethernet PoE compliant hardware	Shall establish and maintain a link with other Ethernet Nodes on the Network. Shall encode the SpaceWire packets received from the Network Layer into Frames to using Manchester Encoding (10Mbps) or 4B/5B NRZ-I Encoding (100Mbps). Shall decode the bitstream received from the Physical Layer into Frames to be passed up to the Network Layer to receive the SpaceWire Packet. Shall direct the Frames to the required Destination Address using the MAC protocol.
Physical	IEEE 802.3 Ethernet	IEEE 802.3 Ethernet PoE compliant hardware	Shall drive and receive the bitstream data signals through the twisted pair cabling using differential signalling.

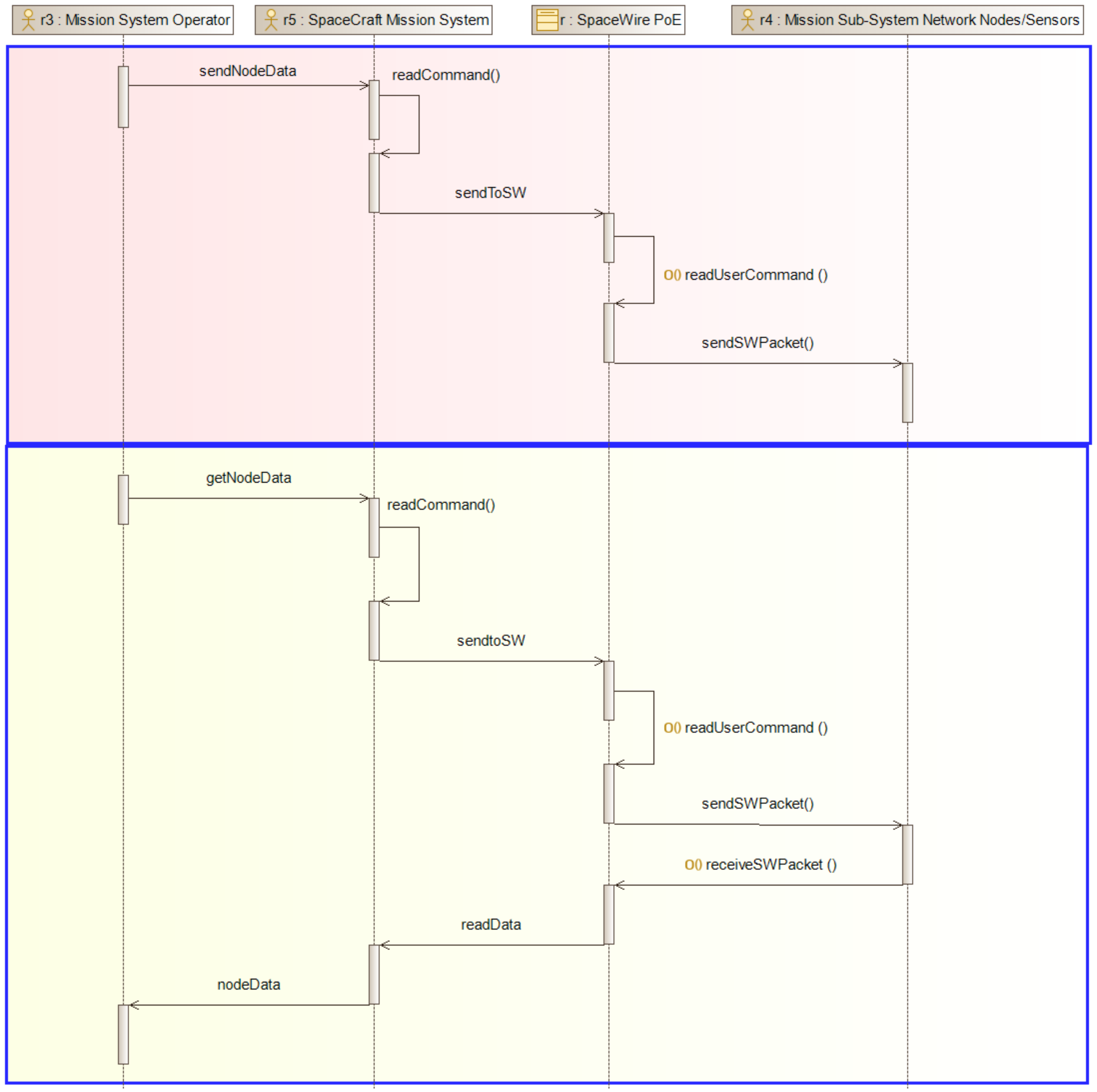
Appendix D - SpaceWire PoE System Requirements Document

SpaceWire & PoE System Requirements Document (SRD) V1.0						
System Requirement ID	**Requirement**	**Priority**	**Verification**	**Validation**	**Comments**	**SpaceWire PoE System Compliance Achieved**
Network Layer						
SR1.0	Shall send and receive packets over a SpaceWire PoE network	Threshold	UML Modelling Physical Architecture Diagram	User Validation Test Functional Testing		**FC**
SR1.0.1	A SpaceWire packet shall comprise of one or more data characters followed by an End of Packet (EoP) or Error end of Packet Marker (EEP)	Threshold	Software Design Diagram	Functional Testing	EEP not able to be proven for this requirement, EOP proven EEP in theory is not required for SpaceWire PoE due to Ethernet 32-bit CRC	**FC**
SR1.0.2	The start of a packet shall either be: - The first data character sent after a link is initialised or re-initialised following a disconnect. - The data character that immediately follows an EOP or EEP	Threshold	Software Design Diagram	Functional Testing	EOP and EEP's are not iaw SpaceWire, bespoke header bytes identifiers are required to be incorporated for SpaceWire PoE system	**FC**
SR1.0.3	The end of a packet shall be indicated by an EOP or EEP.	Threshold	Software Design Diagram	Functional Testing	EOP and EEP's are not iaw SpaceWire, bespoke header bytes identifiers are required to be incorporated for SpaceWire PoE system	**FC**
SR1.0.4	The packet shall contain zero or more data characters.	Threshold	Software Design Diagram	Functional Testing	Zero data characters not tested, although not anticipated to fail if zero data characters are sent	**FC**
SR1.0.5	The remaining data characters following the destination address and up to the EOP/EPP shall act as the cargo/payload	Threshold	Software Design Diagram	Functional Testing	Destination Address is implemented through IEEE 802.3 MAC addressing	**FC**
SR1.1	Shall distribute a time-code service which sends and receives time-codes over a SpaceWire network	Objective	Software Design Diagram	Functional Testing		**PC**
SR1.1.1	The SpaceWire network node shall have one or more time-code registers	Objective	Software Design Diagram	Functional Testing		**N/A**
SR1.1.2	Each time-code register in a node shall hold the last time code transmitted or received over the endpoints its associated with	Objective	Software Design Diagram	Functional Testing		**N/A**
SR1.1.3	A time-code shall be a broadcast code with the broadcast type set to 0b00	Objective	Software Design Diagram	Functional Testing	Unable to use 0b00 due to testing limiations, only 8-bit chars can be used	**N/A**
SR1.2	Shall distribute an interrupt-code service which sends and receives distributed interrupts over a SpaceWire network	Objective	Software Design Diagram	Functional Testing		**N/A**
SR1.2.1	A distributed interrupt-code shall be a broadcast code with the broadcast type set to 0b10	Objective	Software Design Diagram	Functional Testing	Unable to use 0b00 due to testing limiations, only 8-bit chars can be used	**N/A**

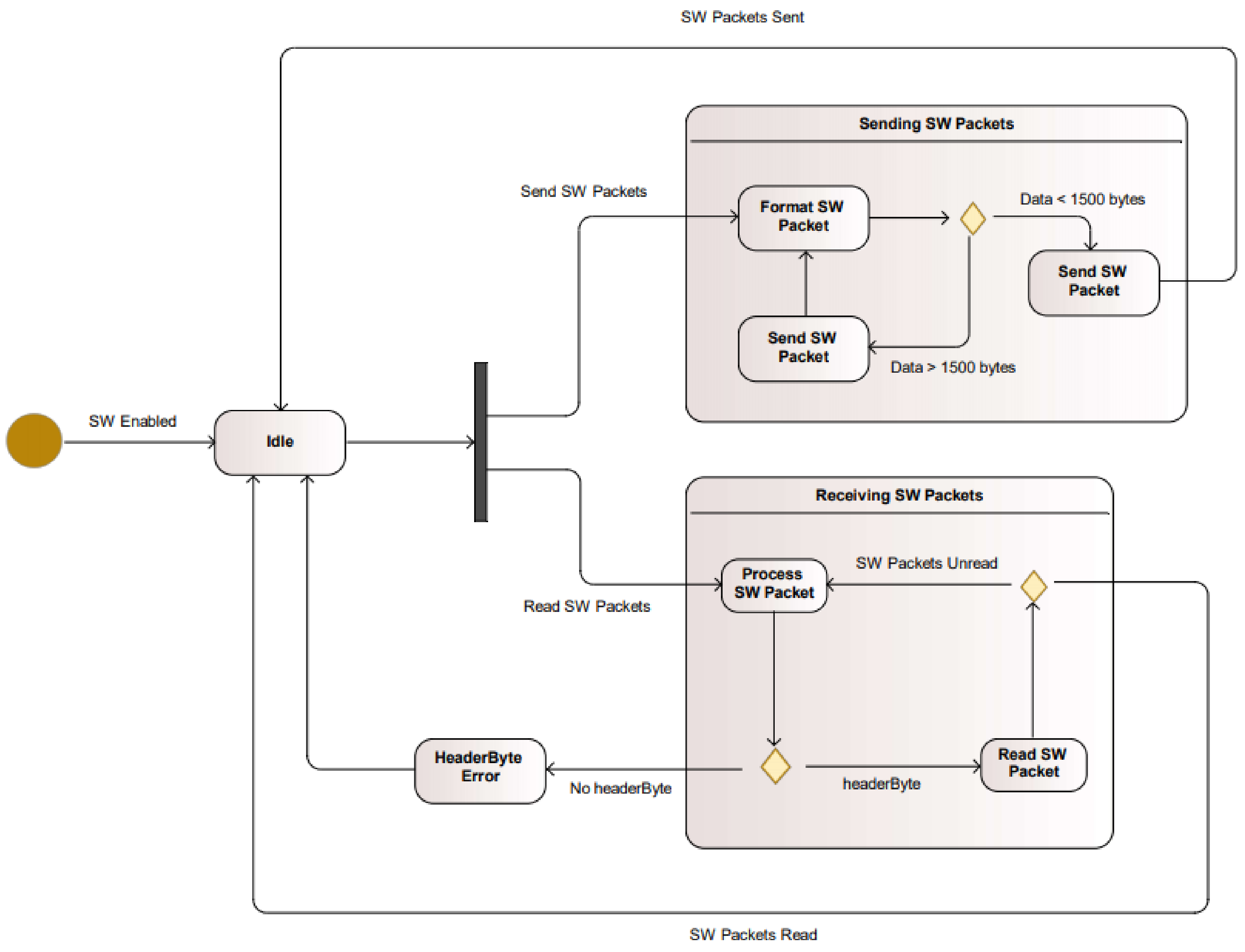
SR1.3	The order priority of sending characters and control codes shall be as follows; Broadcast-Codes (highest priority) FCTs N-Chars Nulls	Objective	Software Design	Functional Testing	Nulls and FTC's not required to be sent may not be required to be sent	**N/A**
SR1.4	The order priority of which the network layer passes broadcast codes to the data-link layer are as follows: - Time-code (highest priority) - Interrupt Acknowledgement Code - Interrupt code (lowest priority)	Objective	Software Design Diagram	Functional Testing		**N/A**
Data-Link Layer						
SR2.0	Shall provide an N-Char service which sends and receives N-Chars over a SpaceWire link	Threshold	Physical Architecture Diagram IEEE 802.3 Compliant Hardware	Functional Testing	The N-Char Service is handled at the Data-Link Layer which is implemented throuhg the IEEE 802.3 Ethernet Protocol	**FC**
SR2.1	Shall provide a broadcast code service which sends and receives broadcast codes (time-codes and distributed interrupt codes over a SpaceWire link	Objective	Software Design Diagram	Functional Testing	Implemented into Network Layer in relation to SR1.1 and SR1.2	**PC**
SR2.2	Shall provide a character encoding service which encodes characters and control codes into symbols, serialise those symbols and data-strobe encode them for transmission over SpaceWire Physical layer	N/A	N/A	N/A	The encoding of data characters will be implemented using IEEE 802.3 Compliant Hardware/Encoding Scheme. Control Codes have to be implemented into the network layer as part of the cargo. Requirement cannot be fully met.	**N/A**
SR2.3	Shall provide a decoding service which recovers the data bit-stream from the data strobe signals from the physical layer, de-serialise the bit-stream and decode the resulting symbols into characters and control codes.	N/A	N/A	N/A	The decoding of data characters will be implemented using IEE 802.3 Hardware/Encoding Scheme. Control Codes have to be implemented into the network layer as part of the cargo. Requirement cannot be fully met.	**N/A**
SR2.4	Shall direct the SpaceWire packets contained within Ethernet Frames to the required destination address	Threshold	Physical Architecture Diagram IEEE 802.3 Compliant Hardware	Functional Testing	Implemented through IEEE 802.3 Ethernet use of MAC Addresses. No compliance to SpaceWire	**FC**
Physical Layer						
SR3.0	Shall provide a transmit service which transmits the data and strobe signals from the encoding layer over the physial medium.	Threshold	Physical Architecture Diagram IEEE 802.3 Compliant Hardware	User Validation Test Functional Testing	Strobe signals not able to be incorporated into implementation due to IEEE 802.3 PoE protocol utilised. Partial compliance to SpaceWire	**PC**

Management Information Base						
SR4.0	Shall provide a set parameter service which writes control or configuration information to the other SpaceWire Layers	Objective			Partial compliance only able to be demonstrated due to constraints of COTS hardware implementing 802.3 Ethernet (Unknown access)	**N/A**
SR4.1	Shall provide a get status service which reads the status, current configuration and control values of the other SpaceWire layers	Objective			Partial compliance only able to be demonstrated due to constraints of COTS hardware implementing 802.3 Ethernet (Unknown access to ASICs etc.)	**N/A**
Interface, Compatibility and Interoperable						
SR5.0	The following hardware shall be compliant to IEEE 802.3 PoE: - Master Embedded Device Ethernet Node - Slave Embedded Device Ethernet Node - Ethernet Network Switch	Threshold	IEEE 802.3at Compliant Hardware	Functional Testing	COTS Hardware	**FC**
SR5.1	The following system components shall have USB2.0 interfaces: - Master embedded Device Ethernet Nodes - Slave Embedded Device Ethernet Node	Threshold	USB2.0 Compliant Hardware	Functional Testing	COTS Hardware	**FC**
SR5.2	The following system components shall have RJ45 Female Connectors: - Master embedded Device Ethernet Nodes - Slave Embedded Device Ethernet Node - Ethernet Network Switch	Threshold	IEEE 802.3 Compliant Hardware Physical Architecture Diagram	Functional Testing	COTS Hardware	**FC**
SR5.3	The system cabling shall be compliant to IEEE 802.3 Ethernet Cat6	Threshold	IEEE 802.3 Cat6 Compliant Cabling	Functional Testing	COTS Hardware	**FC**

Appendix E – SpaceWire PoE UML Diagrams



SpaceWire PoE UML Diagram 1 - Sequence Diagram



SpaceWire PoE UML Diagram 2 - State Machine Diagram

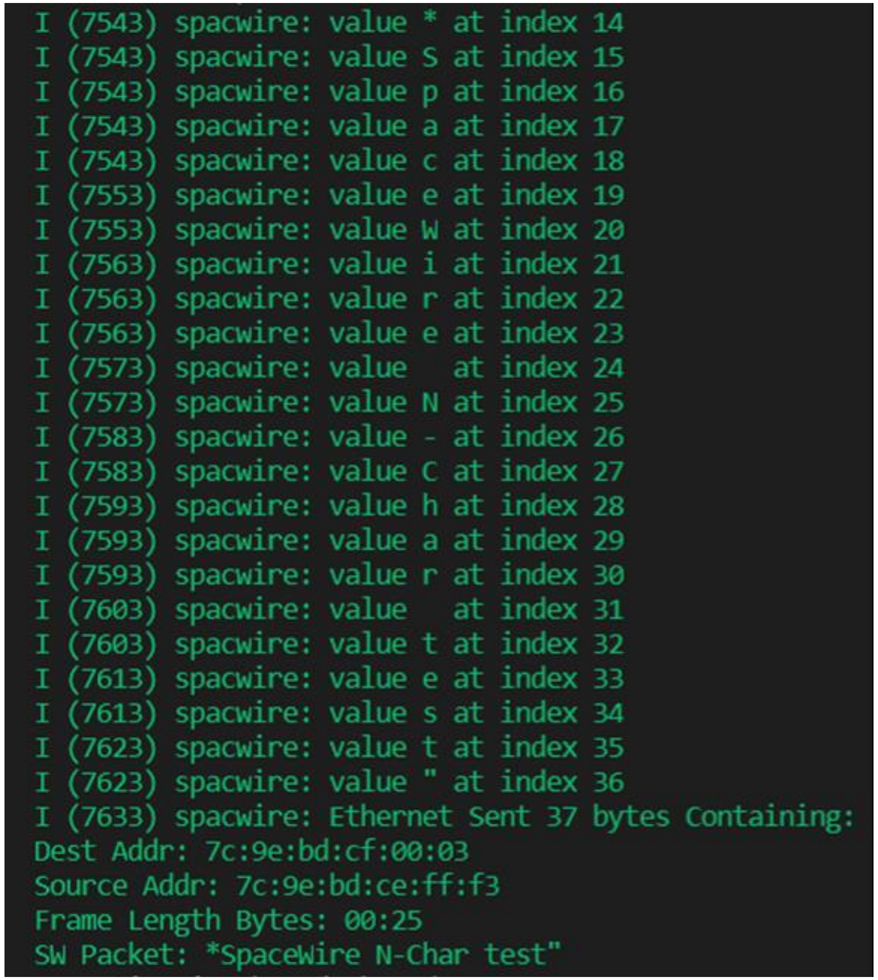
Appendix F - Functional Test-Cases

ID	Test-Case Name	System Requirements	Test-Case Description	Pass/Fail	Comments
01	SW N-Char Packets	1.0 1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 2.0 2.4	The Slave Device sends a string of data characters that are individually the size of a byte (N-Char), to the Master Device via SpaceWire Protocol using the SpaceWire PoE System. The SpaceWire Packet was defined as below; *SpaceWire N-Char test'	Pass	Due to testing limitations, the implementation uses bytes of char value to demonstrate the SpaceWire PoE Headers (Table 2). It is hypothesised that this does not invalidate the test results.
02	Multiple Packets per Frame	1.0 1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 2.0 2.4	The slave device sends multiple SpaceWire Packets within the Ethernet Frame to the Master Device. Demonstrates the same functionality as the Test ID01 test, although multiple packets are processed by the master device per frame. Multiple Packets formatted as below; *SW Packet 1'*SW Packet 2'*SW Packet 3'	Pass	Same as above.

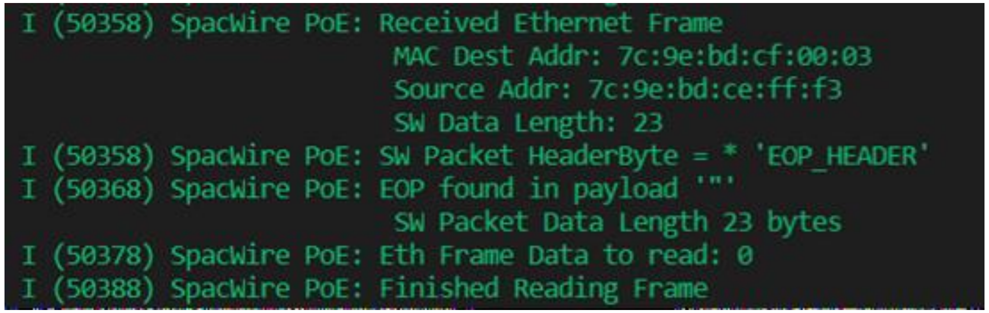
03	SW C-Code	1.1 1.1.1 1.1.2 1.1.3	The Slave Device sends a SpaceWire C-Code to the Master Device over the SpaceWire PoE system. SpaceWire Broadcast-Code shown as below; #1	Pass	Same as above. No implementation of Time-Code Register in SpaceWire PoE System to hold and update values.
04	Mixed Frame	1.0 1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 2.0 2.4	The Slave Device sends multiple SpaceWire Packets and SpaceWire C-Codes within a single ethernet frame to the Master Device. *SW Packet 1'#1*SW Packet 2'*SW Packet 3'#2#3	Pass	Same as above.

Appendix G - System Functional Testing Results

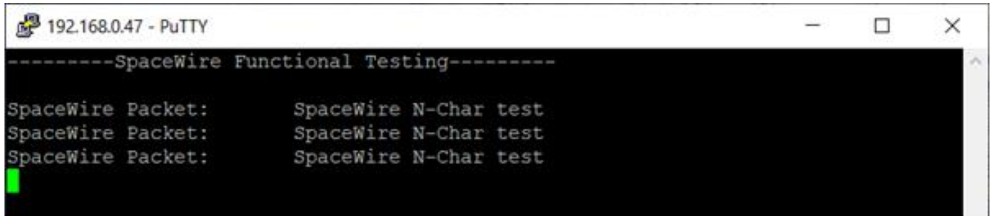
SpaceWire PoE N-Char Packets Testing


```
I (7543) spacwire: value * at index 14
I (7543) spacwire: value S at index 15
I (7543) spacwire: value p at index 16
I (7543) spacwire: value a at index 17
I (7543) spacwire: value c at index 18
I (7553) spacwire: value e at index 19
I (7553) spacwire: value W at index 20
I (7563) spacwire: value i at index 21
I (7563) spacwire: value r at index 22
I (7563) spacwire: value e at index 23
I (7573) spacwire: value   at index 24
I (7573) spacwire: value N at index 25
I (7583) spacwire: value - at index 26
I (7583) spacwire: value C at index 27
I (7593) spacwire: value h at index 28
I (7593) spacwire: value a at index 29
I (7593) spacwire: value r at index 30
I (7603) spacwire: value   at index 31
I (7603) spacwire: value t at index 32
I (7613) spacwire: value e at index 33
I (7613) spacwire: value s at index 34
I (7623) spacwire: value t at index 35
I (7623) spacwire: value " at index 36
I (7633) spacwire: Ethernet Sent 37 bytes Containing:
Dest Addr: 7c:9e:bd:cf:00:03
Source Addr: 7c:9e:bd:ce:ff:f3
Frame Length Bytes: 00:25
SW Packet: *SpaceWire N-Char test"
```

N-Char Packet Testing 3 - Slave Device Log Output


```
I (50358) SpacWire PoE: Received Ethernet Frame
                        MAC Dest Addr: 7c:9e:bd:cf:00:03
                        Source Addr: 7c:9e:bd:ce:ff:f3
                        SW Data Length: 23
I (50358) SpacWire PoE: SW Packet HeaderByte = * 'EOP_HEADER'
I (50368) SpacWire PoE: EOP found in payload '"'
                        SW Packet Data Length 23 bytes
I (50378) SpacWire PoE: Eth Frame Data to read: 0
I (50388) SpacWire PoE: Finished Reading Frame
```

N-Char Packet Testing 1 - Master Device Log Output


```
192.168.0.47 - PuTTY
---------SpaceWire Functional Testing---------

SpaceWire Packet:       SpaceWire N-Char test
SpaceWire Packet:       SpaceWire N-Char test
SpaceWire Packet:       SpaceWire N-Char test
```

N-Char Packet Testing 2 - PuTTY Output

<u>SpaceWire PoE Multiple Packets Per Frame Testing</u>

```
I (2433) spacwire: Ethernet Sent 53 bytes Containing:
Dest Addr: 7c:9e:bd:cf:00:03
Source Addr: 7c:9e:bd:ce:ff:f3
Frame Length Bytes: 00:35
SW Packet: *SW Packet 1'*SW Packet 2'*SW Packet 3'
```

Multiple Packet Testing 1 - Slave Device Log Output

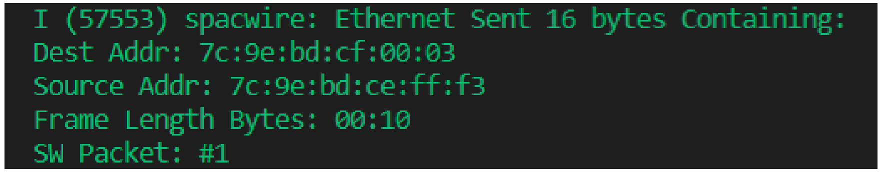
```
I (19868) SpacWire PoE: Received Ethernet Frame
                         MAC Dest Addr: 7c:9e:bd:cf:00:03
                         Source Addr: 7c:9e:bd:ce:ff:f3
                         SW Data Length: 39
I (19868) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (19878) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (19878) SpacWire PoE: Eth Frame Data to read: 26
I (19888) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (19898) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (19898) SpacWire PoE: Eth Frame Data to read: 13
I (19908) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (19918) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (19918) SpacWire PoE: Eth Frame Data to read: 0
I (19928) SpacWire PoE: Finished Reading Frame
```

Multiple Packet Testing 3 - Master Device Log Output

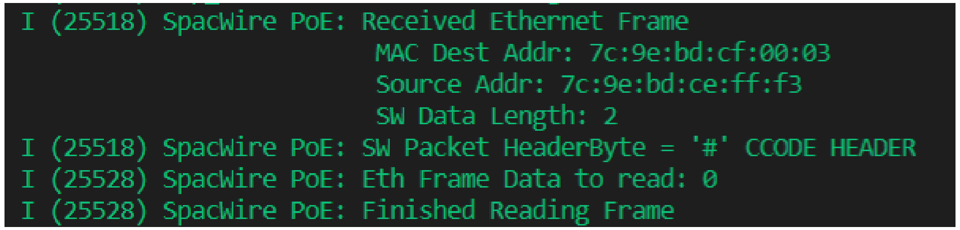
```
192.168.0.47 - PuTTY
---------SpaceWire Functional Testing---------

SpaceWire N-Char Packet:        SW Packet 1
SpaceWire N-Char Packet:        SW Packet 2
SpaceWire N-Char Packet:        SW Packet 3
SpaceWire N-Char Packet:        SW Packet 1
SpaceWire N-Char Packet:        SW Packet 2
SpaceWire N-Char Packet:        SW Packet 3
```

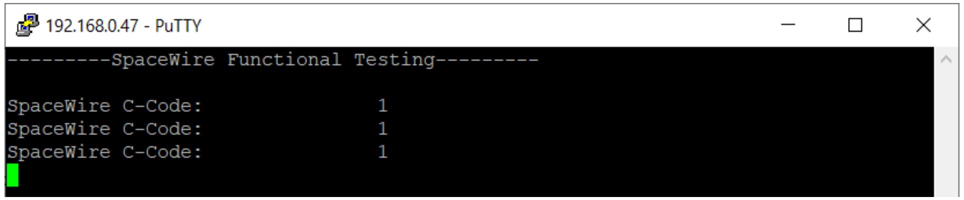
Multiple Packet Testing 2 - PuTTY Output



C-Code Testing 1 - Slave Device Log Output



C-Code Testing 3 - Master Device Log Output



C-Code Testing 2 - PuTTY Output

SpaceWire PoE Mixed Frame Testing

```
I (14523) spacwire: Ethernet Sent 59 bytes Containing:
Dest Addr: 7c:9e:bd:cf:00:03
Source Addr: 7c:9e:bd:ce:ff:f3
Frame Length Bytes: 00:3b
SW Packet: *SW Packet 1'#1*SW Packet 2'*SW Packet 3'#2#3
I (19523) spacwire: Ethernet Sent 59 bytes Containing:
Dest Addr: 7c:9e:bd:cf:00:03
Source Addr: 7c:9e:bd:ce:ff:f3
Frame Length Bytes: 00:3b
SW Packet: *SW Packet 1'#1*SW Packet 2'*SW Packet 3'#2#3
```

Mixed Packet Testing 1 - Slave Device Log Output

```
I (24858) SpacWire PoE: Received Ethernet Frame
                         MAC Dest Addr: 7c:9e:bd:cf:00:03
                         Source Addr: 7c:9e:bd:ce:ff:f3
                         SW Data Length: 45
I (24858) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (24868) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (24868) SpacWire PoE: Eth Frame Data to read: 32
I (24878) SpacWire PoE: SW Packet HeaderByte = '#' CCODE HEADER
I (24888) SpacWire PoE: Eth Frame Data to read: 30
I (24888) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (24898) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (24908) SpacWire PoE: Eth Frame Data to read: 17
I (24908) SpacWire PoE: SW Packet HeaderByte = '*' EOP_HEADER
I (24918) SpacWire PoE: EOP found in payload '''
                         SW Packet Data Length 13 bytes
I (24928) SpacWire PoE: Eth Frame Data to read: 4
I (24928) SpacWire PoE: SW Packet HeaderByte = '#' CCODE HEADER
I (24938) SpacWire PoE: Eth Frame Data to read: 2
I (24938) SpacWire PoE: SW Packet HeaderByte = '#' CCODE HEADER
I (24948) SpacWire PoE: Eth Frame Data to read: 0
```

Mixed Packet Testing 3 - Master Device Log Output

```
192.168.0.47 - PuTTY
---------SpaceWire Functional Testing---------

SpaceWire N-Char Packet:        SW Packet 1
SpaceWire C-Code:               1
SpaceWire N-Char Packet:        SW Packet 2
SpaceWire N-Char Packet:        SW Packet 3
SpaceWire C-Code:               2
SpaceWire C-Code:               3
SpaceWire N-Char Packet:        SW Packet 1
SpaceWire C-Code:               1
SpaceWire N-Char Packet:        SW Packet 2
SpaceWire N-Char Packet:        SW Packet 3
SpaceWire C-Code:               2
SpaceWire C-Code:               3
SpaceWire N-Char Packet:        SW Packet 1
SpaceWire C-Code:               1
SpaceWire N-Char Packet:        SW Packet 2
SpaceWire N-Char Packet:        SW Packet 3
SpaceWire C-Code:               2
SpaceWire C-Code:               3
```

Mixed Packet Testing 2 - PuTTY Output