1. Project Context & Background

1.1. STRATHcube

STRATHcube is a student led satellite project at the University of Strathclyde that will be launched from the International Space Station with the aim of demonstrating the use of a Passive Bistatic Radar (PBR) for in-orbit detection of space debris. This project aims to create an engineering model of the downlink communication system for the satellite using commercially available development boards in compliance with relevant standards, a key step towards the preliminary design review with the European Space Agency (ESA) Fly Your Satellite program.

The satellite will include an Alén Space TOTEM Software Defined Radio (SDR) to handle both communication and PBR systems, which consists of an AMD Zynq 7020 System on Chip (SoC) connected to an Analog Devices AD9364 RF transceiver. Due to the high cost of space qualified hardware, it will be modelled by a Digilent Zedboard connected to an FMCOMMS1 AD9361 development board. This has an identical Zynq SoC with less external memory and a similar RF transceiver.

A summer research project investigated the downlink configuration, selecting a frequency range of 430-440MHz, creating a link budget and selecting DVB-S2 (Digital Video Broadcasting - Satellite - Second Generation) as the modulation scheme. This has extremely strong Forward Error Correction (FEC) capabilities and can adapt based on channel conditions to maximise spectral efficiency with Adaptive Coding and Modulation (ACM).

As this project ties into a larger mission, it was important to identify clear goals and limit scope. The primary goal is to create a detailed system definition and implementation of the downlink communication system that maximises data throughput over the duration of the mission. The detailed definition of packet structures, uplink communication system and final implementation on flight hardware are outside the scope of this project.

1.2. DVB-S2

1.2.1. Overview

The DVB-S2 standard is modular with several optional blocks and data formats depending on the application[1]. This project will use a single generic packetised stream with a fixed User Packet Length (UPL), as it includes 8 bit Cyclic Redundancy Check (CRC-8) error detection. The data stream is sliced into Data Field Length (DFL) sized chunks and a Baseband Frame (BBFRAME) header inserted. The FEC systems require the BBFRAME to be padded to a specific length that depends on the current ACM settings. After two coding stages, this becomes a FECFRAME which goes through constellation mapping, pilot insertion and scrambling to become a Physical Layer Frame. It is finally filtered and modulated up to the carrier frequency.

DVB-S2 uses four modulation schemes from QPSK to 32APSK, coding rates from 1/4 to 9/10 and two FECFRAME lengths. The ACM Router can change any of these parameters, as well as the DFL and UPL from one BBFRAME to the next to adapt to channel conditions. To minimise padding, a link layer protocol can be used called Generic Stream Encapsulation (GSE) that handles the fragmentation of the packet based on the current optimal DFL[2]

1.2.2. Implementation

One of the key challenges of this project is managing Programmable Logic (PL) resource usage. The flight hardware must handle uplink and downlink communications, PBR processing and existing code included by the manufacturer. A target of no more than 50% usage on any one resource was implemented to ensure feasibility. The second key challenge is reliability, verification, and adherence to standard. To ensure this, a MATLAB example DVB-S2 implementation using HDL Coder will be modified to minimise resource requirements and integrate with the rest of the system. Packet handling will be managed on the Processing System (PS), including ACM, scheduling, any GSE implementation and header configuration.

1.3. Adaptive Coding and Modulation Analysis

Due to the complexity of implementing ACM in the system, an investigation into the benefits over a fixed configuration that maximised availability was conducted. The satellite was simulated for the full mission duration at its initial altitude of 400km using the MATLAB Satellite Communications Toolbox. The elevation of the satellite during each pass over the Glasgow based ground station was binned into 5° increments. An existing link budget was used to calculate the Carrier to Noise Ratio (CNR) for each elevation bin and cross-referenced to the optimimum modulation and coding rate using the DVB-S2 standard with a 10dB margin and Bit Error Rate (BER) of 10^{-7}. The maximum possible throughput was calculated for each elevation bin and multiplied by the time spent at each elevation to find the total throughput for each strategy. The results are shown in Figure 1 and show a significant increase in throughput, proving that the complexity of implementing ACM is worth the benefits. The uplift would be further if the ground station was located at a lower latitude.

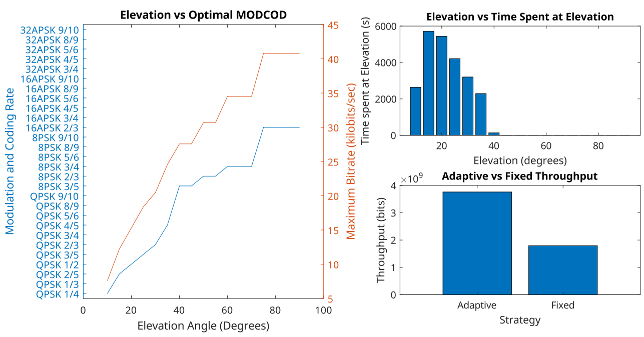


Figure 1: Left - Elevation vs Modulation and Coding Rate (Blue), Modulation and Coding rate to throughput (red).
Top Right - Time spent in each elevation bin.
Bottom Right - Throughput vs strategy.

1.4. Packet Handling

1.4.1. XTCE

The XML Telemetric and Command Exchange (XTCE) standard[3] allows consistent definition of data packets, algorithms and commands to be used within space systems. This has been created by the Consultative Committee for Space Data Systems (CCSDS) in order to promote inter-operability and has has been selected for the project. This allows definition of generic packets for development without restricting the final system.

As XTCE is quite a complex standard so a NASA tool called Core Flight System (CFS) Command and Data Dictionary (CCDD) will be used to manage packet definitions. This can also be used to generate C headers for parsing, or an existing Python library may be used. The Python approach could be simpler to implement, but the C headers will be more efficient and allow for easier integration with the rest of the system.

1.4.2. Interfacing

The TOTEM SDR has multiple interfaces to other systems. The main ones for flight are Controller Area Network (CAN) and Inter-Integrated Circuit (I2C), with Ethernet included for ground testing. The system may also have to send packets from other programs running on the PS. This will be achieved using a First In First Out (FIFO) buffer with Linux Inter-Process Communication (IPC) mechanisms.

1.4.3. Scheduling & Compression

The onboard data handling has been analysed in a previous dissertation project. This identified the need to prioritise housekeeping data, such as GNSS data and power telemetry, over primary payload data. This project also defined multiple packet structures, along with their priority. These will be used to create sample packets for testing of the system.

This project also identified the need for compression of the data to reduce the amount to be transmitted, with the lossless POCKET algorithm being selected for this purpose. A C++ implementation of the newer POCKET+ algorithm is available that has been created based on the CCSDS standard and will be used on the PS.

1.4.4. ACM Router + PL Interface

The ACM router is responsible for managing the configuration of the DVB-S2 system. This includes selection of modulation, coding rates, FECFRAME length and DFL. In the final system these decisions will be made based on information received from the ground station, however for this project they will be made based on a pre-defined schedule.

This block will handle the alignment of packets to data fields and pass this to the PL, along with control data. The interface driver will be created using the libiio library created by Analog Devices.

1.5. System Architecture

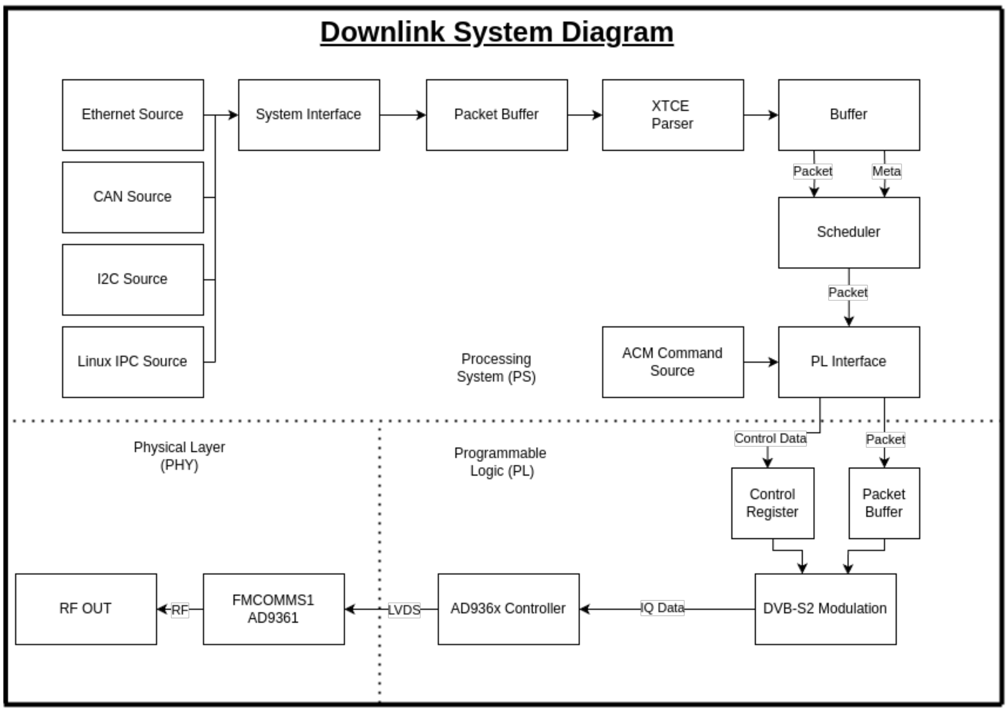
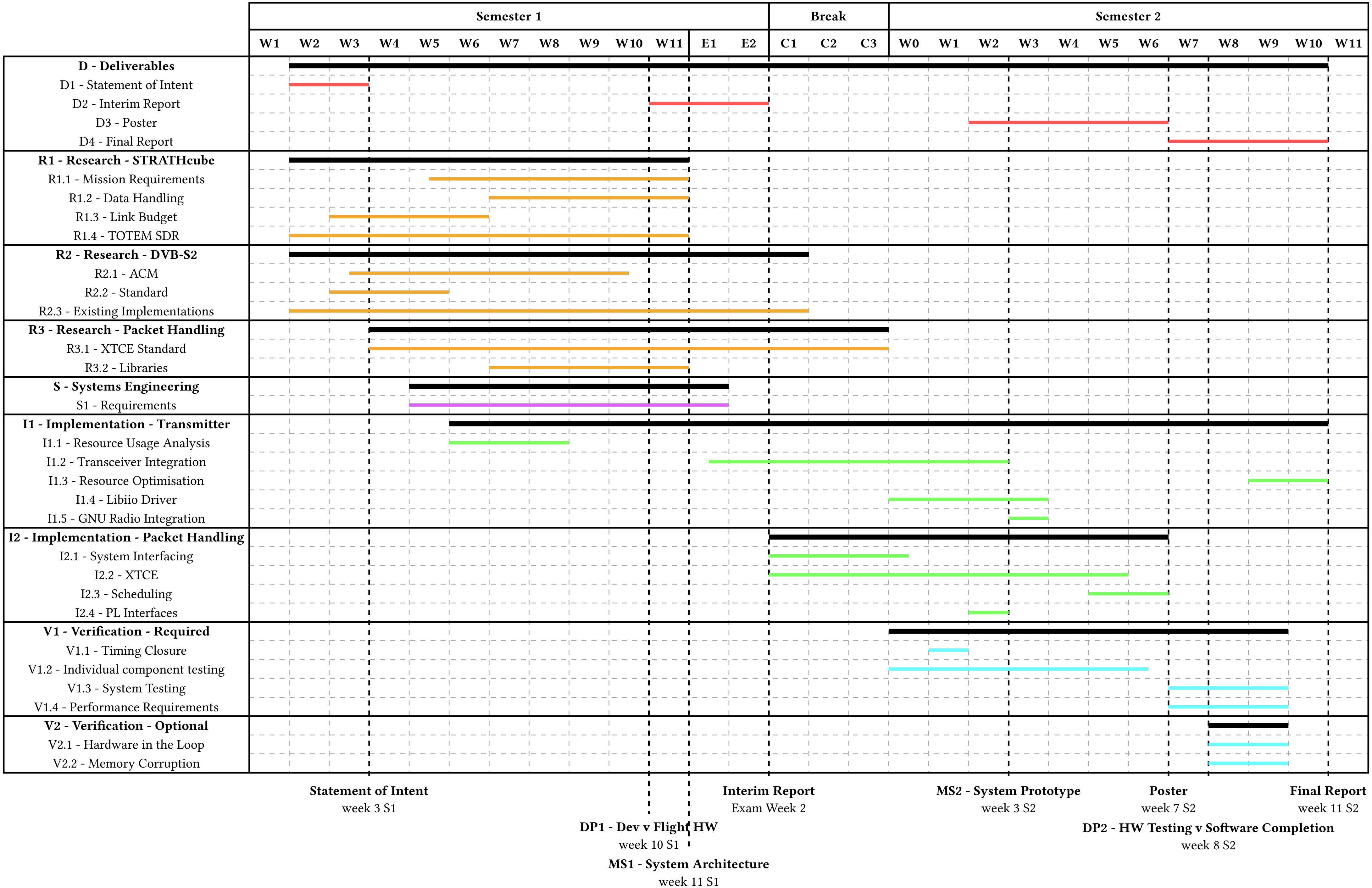


Figure 2: System Diagram

2. Project Timeline



3. Project Review

3.1. Project Plan

The first semester has been devoted to the research and planning of the project. The outcome of which has been a detailed analysis of each sub-block required for the final system and the identification of open source resources and relevant standards to ensure success.

There are three key milestones outlined in the project timeline. The first being the system architecture, which has been defined in Section 1. The second is the creation of a “System Prototype” which will demonstrate the minimum features necessary to send a packet through the system. This is targeted for Week 3 of Semester 2.

There are two key decision points, marked “DP1” and “DP2”. DP1 has been reached, and marks the decision to target development hardware, as the TOTEM SDR could not be procured in time for the project. DP2 will be reached in Week 8 of Semester 2 and will mark the decision to focus on end to end testing of the system versus continuing feature development.

To ensure time is left for the final report, a design freeze will be implemented during week 9 of semester 2. This means that no major features or changes will be implemented from that point onwards to ensure that the system is finalised and tested in advance of the final report deadline.

3.2. Technical Risks

Risks	Mitigation	Likelihood	Severity	L*S
Catastrophic Dataloss	1. Use of cloud storage 2. Use of Git source control 3. Regular snapshots of work	1	5	5
Severe Illness	1. Creation of draft submissions on Myplace to avoid missing deadlines 2. Report personal circumstances on Pegasus 3. Discussion of reduction of scope with supervisor	1	4	4
Breakage of key equipment	1. Only work in designated areas 2. Use of equipment as designed, with care given for fragile connectors 3. Identification of alternatives available in the department	2	3	6
“Feature Creep”	1. Creation of requirements document 2. Regular review of work with supervisor 3. Scope freeze decided at Interim report, with options for further reduction identified in project plan	3	2	6
Difficulty in system integration	1. Regular project meetings with supervisor 2. Adherence to best practices in RTL design with continous verification and testing 3. Use of version control to track changes 4. Use of a modular design to allow verification of components 5. Use of resources from manufacturers to ensure compatibility	3	2	6
Extended learning curves of new languages and tools	1. Use of online resources 2. Use of existing knowledge of similar tools to reduce learning curve 3. Use of manufacturer resources	3	2	6
System not fit for purpose	1. Design with requirements in mind 2. Use of open-source examples to ensure compatibility and reliability 3. Use of a modular design to allow for changes and unit testing	2	3	6

3.3. Sustainability

STRATHcube’s primary mission relates to a pressing issue in the space industry. The PBR will demonstrate a technology that could fill a critical resolution gap in debris detection that ground based radars cannot. The importance of space situational awareness cannot be understated, as the amount of operational satellite increases there will be a higher risk of impact with debris too small to be detected with current methods.

The more data that can be downlinked during the mission, the more opportunities there are to observe passing debris and verify the operation of the PBR. As such, the successful completion of this project will directly contribute to the success of the mission and an important aspect of space sustainability.

3.4. Ethical Considerations and EDI

The main ethical considerations for this project lie in compliance to relevant telecommunications regulations from the International Telecommunication Union (ITU) and Office of Communications (OFCOM). During development, all transmissions will be conducted using certified equipment and through cables to ensure that no harmful interference is caused.

The resulting source code will be made available to the public under an open-source license to ensure that the project is transparent and accessible to all. This will also allow for the project to be continued for further use in STRATHcube or as a resource for other projects.

Aditionally, space is a resource that underpins modern society and it is important to ensure that it is used responsibly. The primary and secondary missions of STRATHcube will positively contribute to space sustainability and as such will have a positive impact that will support society as a whole.

4. References

[1] “DVB-S2 Standard.”

[2] “TS 102 606-1 - V1.2.1 - Digital Video Broadcasting (DVB); Generic Stream Encapsulation (GSE); Part 1: Protocol.”

[3] “XML Telemetric and Command Exchange—Version 1.2,” 2020.

[4] “MATLAB DVB-S2 HDL Transmitter.” Accessed: Dec. 03, 2024. [Online]. Available: https://uk.mathworks.com/help/satcom/ug/dvbs2-hdl-transmitter.html

[5] “gr-dvbs2rx GitHub Repository.” Accessed: Dec. 03, 2024. [Online]. Available: https://igorauad.github.io/gr-dvbs2rx/

[6] Gavin Medley, Michael Chambliss, and Greg Lucas, “space_packet_parser.” Accessed: Dec. 03, 2024. [Online]. Available: https://github.com/medley56/space_packet_parser

[7] K. Deergha Rao, “Linear Block Codes,” *Channel Coding Techniques for Wireless Communications*. Springer India, New Delhi, pp. 73–126, 2015. doi: 10.1007/978-81-322-2292-7_4.

[8] A. Morello and V. Mignone, “DVB-S2: The Second Generation Standard for Satellite Broad-Band Services,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006, doi: 10.1109/JPROC.2005.861013.

[9] M. Eroz, F.-W. Sun, and L.-N. Lee, “DVB-S2 low density parity check codes with near Shannon limit performance,” *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 269–279, 2004, doi: 10.1002/sat.787.

[10] “nasa/CCDD.” Accessed: Dec. 10, 2024. [Online]. Available: https://github.com/nasa/CCDD

[11] D. J. Evans and A. Donati, “The ESA POCKET+ housekeeping telemetry compression algorithm: Why make spacecraft operations harder than it already is?,” in *2018 SpaceOps Conference*, Marseille, France: American Institute of Aeronautics, Astronautics, May 2018. doi: 10.2514/6.2018-2613.