



OA-II Backplane Bus System Design

DR00001

Rev: A03
Jinzhi Cai
2019-08-17

Contents

1	Introduction	2
1.1	Scope	2
1.2	Purpose	2
1.3	Revision History	2
2	Bus System Requirements	3
2.1	Hardware Requirements	3
2.2	Software Requirements	3
2.3	Bandwidth Calculation	3
2.3.1	Low Speed Payload	3
2.3.2	High Speed Payload	4
2.3.3	Total bandwidth	5
3	Analysis of common bus protocols	6
3.1	I2C	6
3.2	SPI	6
3.3	UART	6
3.4	CAN	6
3.5	PCle	7
3.6	RapidIO	7
3.7	SpaceWire	7
3.8	Interlaken	7
3.9	Ethernet	8
4	Recommended System Design	9
4.1	Bus Structure	9
4.1.1	Initialization	9
4.1.2	Pre-launch	9
4.1.3	During Flight	9
4.1.4	Post-flight	10
4.1.5	Emergency Backup	10
4.1.5.1	Scenario 1 - Command Bus Failure	10
4.1.5.2	Scenario 2 - Data Bus Failure	10
4.1.6	Bus Data Summary	10
4.2	System Redundancy	11
4.3	CAN + SpaceWire	12
4.4	CAN + RapidIO	13
5	Bibliography	14

1 Introduction

1.1 Scope

This document analyzes the requirements for OA-II VEH system data transmission, and current bus technologies in the field. It outlines a system design to fulfill the needs of OA-II VEH system.

1.2 Purpose

The goal for the OA-II VEH backplane bus system is to construct a fast, versatile, compact, and highly robust backplane data transmission system.

1.3 Revision History

Rev#	Editor	Delta	Date
A01	Jinzhi Cai	Initialize	2019-7-19
A02	Jinzhi Cai	Add detail Ethernet	2019-7-22
A03	Gabriel Smolnycki	Edits for clarity	2019-08-17

Table 1: Summary of Revision History

2 Bus System Requirements

2.1 Hardware Requirements

Backplane Bus The bus needs to support interchangeable modules.

Vibration Resistant The bus needs provide resistance high shock and vibration.

Size The bus needs to fit inside the rocket payload section.

Topology The hardware implementation cannot be location dependent.

2.2 Software Requirements

Point to Point & Broadcast The bus needs to support broadcasting data.

Bandwidth The bus needs to support the max bandwidth given.

Topology The bus topology should be software defined and reconfigurable.

Priority The bus needs to support message priority levels, including real-time communication.

Multiple Speeds The bus needs to allow both high and low speed devices.

2.3 Bandwidth Calculation

2.3.1 Low Speed Payload

Low speed payloads generate data at a minimum of 10kHz/16bit.

- 4 high pressure sensors for propulsion system
- 2 low pressure sensors for pitot tube
- 4 high temperature sensors for propulsion system
- 4 low temperature sensors for electronics
- 4 low temperature sensors for batteries
- 2 low temperature sensor for ambient

$$4 + 2 + 4 + 4 + 4 + 2 = 20channels \quad (1)$$

$$16bits(b) = 2bytes(B) \quad (2)$$

$$10kHz \times 2B = 20kB/s \quad (3)$$

$$20kB/s \times 20 = 400kB/s \quad (4)$$

2.3.2 High Speed Payload

- 9 axis IMU
- GNSS system
- 4x 1080p@60Hz cameras

For a 9 axis IMU at 10kHz/16bit per axis:

$$9 \times 10kHz \times 2B = 180kB/s \quad (5)$$

For a typical GNSS module running at 10Hz¹:

- UTC launch time - 4 bytes
- Latitude - 4 bytes
- Longitude - 4 bytes
- Altitude - 4 bytes
- Direction + Ground speed - 4 bytes

$$4B \times 5 = 20B \quad (6)$$

$$10Hz \times 20B = 200B/s \quad (7)$$

For cameras, assume a compressed bitrate of 8Mb/s²:

$$8Mb/s = 1MB/s \quad (8)$$

$$1MB/s \times 4 = 4MB/s \quad (9)$$

¹Not including a safety factor

²High bitrate is necessary for high vibration environment

2.3.3 Total bandwidth

Device	Number	Bandwidth per Device	Total Bandwidth
High Pressure Sensors	4	20KB/s	80KB/s
Low Pressure sensors	2	20KB/s	40KB/s
High Temperature Sensors	4	20KB/s	80KB/s
Low Temperature Sensors	10	20KB/s	200KB/s
9 axis IMU	1	180KB/s	180KB/s
GNSS system	1	200B/s	200KB/s
Cameras	4	1MB/s	4MB/s

Table 2: Summary of Estimated Bandwidth

Using a factor of safety of 2 to compensate for encoding overhead:

$$(180kB/s + 4MB/s + 200B/s + 400kB/s) \times 2 \approx 10MB/s \quad (10)$$

3 Analysis of common bus protocols

3.1 I2C

I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. It was invented by Philips and now it is used by almost all major IC manufacturers [1].

I2C is a great low speed communication bus, however it does not support hardware priority level and software defined topology.

3.2 SPI

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards [2].

SPI allows the device to increase its bandwidth by increasing the data clock rate. However, it does not support hardware priority level or software defined topology.

3.3 UART

A universal asynchronous receiver-transmitter is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable [3].

UART does not require a clock line to transmit data. It also can use individual bitrates per device. However it is a point to point communication system, so it needs a switch for more than two devices. It also does not support high enough bandwidth to meet the requirements.

3.4 CAN

A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer [4].

CANbus has hardware priority level support and works up to 500kbps³ bandwidth. It also allows group broadcast as well as point to point communication.

³62.5kB/s

3.5 PCIe

PCI Express, officially abbreviated as PCIe or PCI-e, is a high-speed serial computer expansion bus standard, designed to replace the older PCI, PCI-X and AGP bus standards. It is the common motherboard interface for personal computers' graphics cards, hard drives, SSDs, Wi-Fi and Ethernet hardware connections [5].

PCI Express is a commonly used bus in personal computers. It supports up to 16GT/s⁴ per lane. However, it mostly uses a tree type topology. As a result, it is difficult to add a second master to the system.

3.6 RapidIO

The RapidIO architecture is a high-performance packet-switched interconnect technology. RapidIO supports messaging, read/write and cache coherency semantics [6].

RapidIO is a high speed connection that supports up to 5Gbps⁵ per lane. It also supports multi-master structure. Using a RapidIO switch, software defined technology is also possible. However, RapidIO is a new bus technology that is mainly used in DSPs, high speed FPGAs, and SoCs. It require significant hardware resources compared to the other buses.

3.7 SpaceWire

SpaceWire is defined in the European Cooperation for Space Standardization standard ECSS-E-ST-50-12C (replaces ECSS-E50-12A). The SpaceWire standard was authored by Steve Parkes, University of Dundee with contributions from many individuals within the SpaceWire Working Group from European Space Agency (ESA), European Space Industry, Academia and NASA [7]. There is also a successor called SpaceFibre, which allows up to 15 times higher data rates per lane, while maintaining significant compatibility with SpaceWire [8].

The SpaceWire uses the LVDS voltage standard which is a commonly use voltage standard in FPGAs. The PHY for SpaceWire is relatively simple and does not require significant resources. The newest SpaceWire bus supports speeds of up to 400Mbps per lane⁶.

3.8 Interlaken

Interlaken was invented by Cisco Systems and Cortina Systems in 2006, optimized for high-bandwidth and reliable packet transfers. It builds on the channelization and per channel flow control features of SPI-4.2, while reducing the number of integrated circuit (chip) I/O pins by using high speed SerDes technology [9].

⁴1.969 GB/s

⁵About 625MB/s

⁶50MB/s

Interlaken is designed to replace Ethernet. It supports port division and flow control, however the Interlaken PHY is not supported by many FPGAs. This means that it will require an extra PHY chip. It supports similar speeds to RapidIO⁷.

3.9 Ethernet

Ethernet is a widely used bus that has many benefits to the data bus system, as its standard is implemented in many devices and drivers, and a router for it is easy to develop. It also can go up to 1Gbps⁸ with appropriate PCB design. However, it does not have a real time clock as with SpaceWire, and is not as fast as RapidIO or PCIe. Ethernet is also not designed for realtime applications [10].

⁷625MB/s

⁸125MB/s

4 Recommended System Design

4.1 Bus Structure

Based on the requirements of the bus, the OA-II Bus System will include two parts: the command bus, which is in charge of the initialization of devices bus and then prioritizes data transmission, and the data bus which is in charge of the transmission of large amounts of data (including telemetry) and of programs. The job for those two buses will be different during each stage⁹.

4.1.1 Initialization

Command Bus ON

Data Bus Configuration

In the initialization stage, OA-II VEH COM MCU will detect all the on board devices and run a self-test ensure all devices are functioning correctly. During this stage, all communication occurs via the command bus, and each device will initialize and configure its data bus interface. Finally, the MCU will send a request on the command bus and receive acknowledgment on the data bus.

4.1.2 Pre-launch

Command Bus ON

Data Bus ON

After the initialization stage, the vehicle will enter the pre-launch stage. At this stage, all programs will be loaded onto their respective modules via command bus and wait for an ignition signal for synchronization. All sensors will be running at full speed and deliver data to the MCU to monitor vehicle status via the data bus.

4.1.3 During Flight

Command Bus Standby

Data Bus ON

⁹More detail is included in *ES00007 - OA-II Payload Bus Specifications*

After ignition and during flight, all sensors will be running at full speed and deliver data to the MCU to monitor vehicle status and for recording via the data bus. The command bus will be in standby mode and ready to send an emergency messages.

4.1.4 Post-flight

Command Bus Standby

Data Bus OFF

After landing, the MCU will again run a self test on all modules before turning off power. The data bus will be off during the landing process.

4.1.5 Emergency Backup

4.1.5.1 Scenario 1 - Command Bus Failure

When COM FRU detects a command bus failure, it will disable COM MCU control and use the data bus to send out a Failure Alert to all modules. The data bus will only transmit critical data and commands. Other data units will be powered off or save data to their internal storage.

4.1.5.2 Scenario 2 - Data Bus Failure

When COM FRU detects a data bus failure, it will use command bus to communicate with MCU and check the MCU status. The mission will continue and all modules will switch communications to the secondary data bus, running at a lower speed.

4.1.6 Bus Data Summary

	From: COM	From: TAM	From: PAM
To: COM	X	Data Bus	Data Bus
To: TAM	Command Bus/Data Bus	X	Data Bus
To: PAM	Command Bus	Data Bus	X

Table 3: Communication routes

Command Bus	Data Bus
Mission critical	Not mission critical
Low bitrate data	High bitrate data
Realtime data	Research data
Internal communication	Communication with base station

Table 4: Bus Data Catalog

4.2 System Redundancy

The command bus will have two separate and redundant physical buses. The primary and secondary command buses will send the same command at the same time. Each device will receive information from the primary bus, and use the secondary bus for error checking.

Each unit will have a two lane connection to each data bus router. Both buses will send and receive the same information. Each device will receive information from the primary bus, and use the secondary bus for error checking.

During the Initialization stage, the primary data router will synchronize configuration with the second data routing core. When the configuration is finished, the link will be disabled.

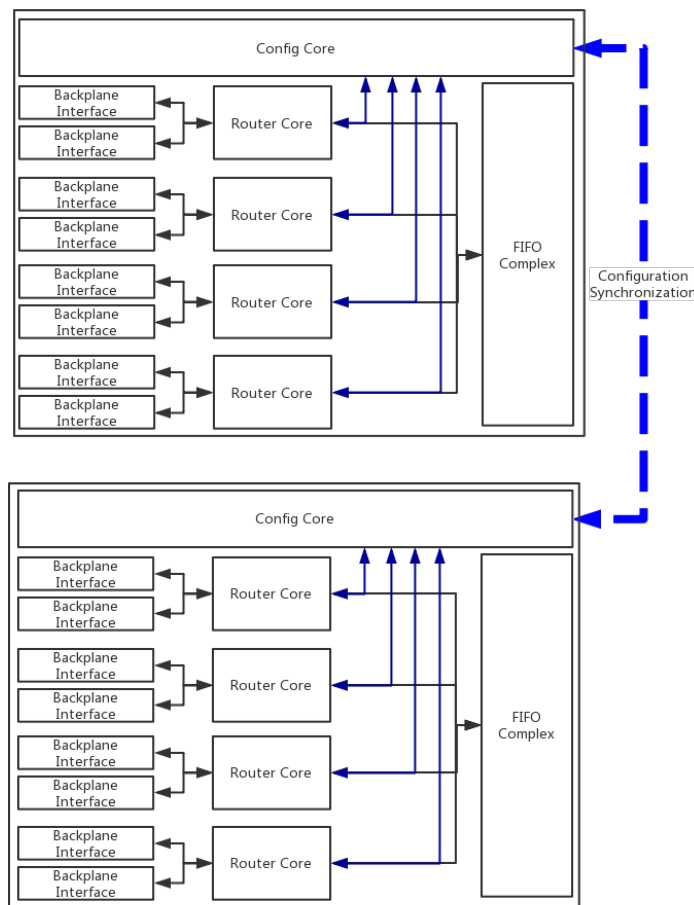


Figure 1: Data Router Structure

4.3 CAN + SpaceWire

Command Bus CAN

Data Bus SpaceWire

In this setup, the command bus uses CAN. It provides prioritized data transmission for small amounts of data transmission, and speeds up to 500kbps, which is enough bandwidth for command data transmission. The data bus uses SpaceWire which provides enough bandwidth for high speed data transmission. It also includes a timestamp feature that will help in sensor fusion. The system is built around two routers, where one is the main router that will be used for primary data transmission and the other is a backup router which can take over in the event the main router fails.

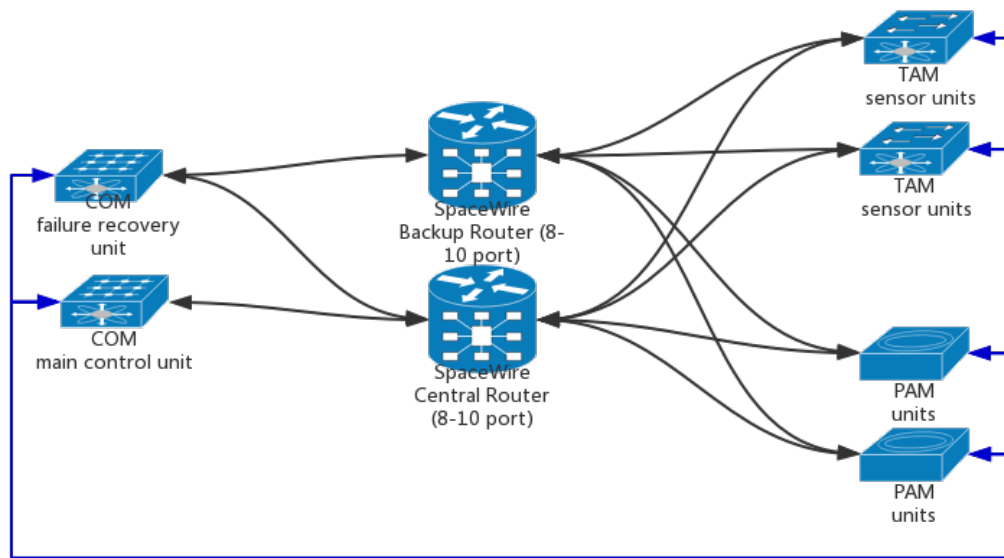


Figure 2: Recommended Design A

4.4 CAN + RapidIO

Command Bus CAN

Data Bus RapidIO

In this setup, CAN bus still is the main command bus due to its prioritized data transmission. However, the data bus uses RapidIO, where each lane can carry 10 times data rate compared to SpaceWire. This will handle most future advanced missions without significant hardware upgrades. This will require a separate ASIC for as a RapidIO router. This setup is more difficult to implement, but is a viable upgrade path from a CAN + SpaceWire design.

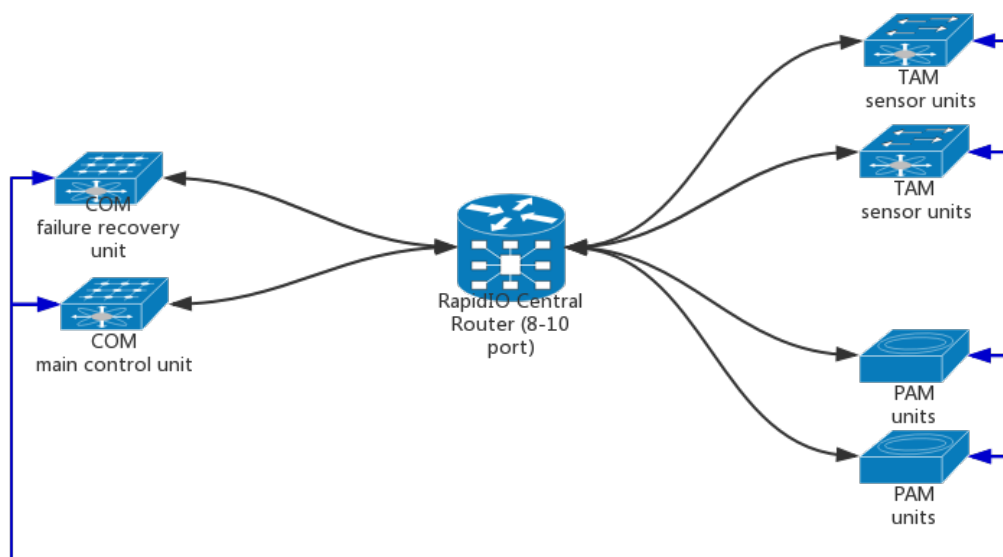


Figure 3: Recommended Design B

5 Bibliography

- [1] aaronRong, *I2C*, 2019. [Online]. Available: <https://www.cnblogs.com/aaronLinux/p/6218660.html>.
- [2] Deng Tao, *SPI*, 2019. [Online]. Available: <https://www.cnblogs.com/deng-tao/p/6004280.html>.
- [3] zjy900507, *UART*, 2019. [Online]. Available: <https://blog.csdn.net/zjy900507/article/details/79789671>.
- [4] CANbus, *CAN*, 2019. [Online]. Available: <https://www.canbus.us/>.
- [5] Margaret Rouse, *Peripheral Component Interconnect Express*, 2019. [Online]. Available: <https://searchdatacenter.techtarget.com/definition/PCI-Expressd>.
- [6] RapidIO.org, *RapidIO*, 2019. [Online]. Available: <http://www.rapidio.org/#>.
- [7] ESA, *SpaceWire - Key Features*, 2017. [Online]. Available: <http://spacewire.esa.int/content/Home/KeyFeatures.php>.
- [8] —, *SpaceFibre / Onboard Data Processing / Space Engineering & Technology / Our Activities / ESA*, 2019. [Online]. Available: <http://spacewire.esa.int/content/Home/KeyFeatures.php>.
- [9] honeypot, *Interlaken*, 2019. [Online]. Available: http://blog.sina.com.cn/s/blog_43c625f101017otd.html.
- [10] qq_5072517, *Ethernet*, 2019. [Online]. Available: https://blog.csdn.net/qq_25072517/article/details/78478968.