Fault Protection in a Component-Based Spacecraft Architecture

By

Elwin Ong

B.S. Aerospace Engineering
University of California, Los Angeles (2001)

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology
May 13, 2003

Signature of Author______________
Department of Aeronautics and Astronautics
June 1, 2003

Certified by____
Professor Nancy G. Leveson
Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by______
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Fault Protection in a Component-Based Spacecraft Architecture

by

Elwin Ong

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

As spacecraft become more complex and autonomous, the need for reliable fault protection will become more prevalent. When coupled with the additional requirement of limiting cost, the task of implementing fault protection on spacecraft becomes extremely challenging. The current state-of-the-art Cassini fault protection software, for example, is a testament to the complexity and difficulty of implementing fault protection on spacecraft. This paper describes how domain knowledge about spacecraft fault protection can be captured and stored in a reusable, component-based spacecraft architecture. The spacecraft-level fault protection strategy for a new spacecraft can then be created by composing generic component specifications, each with component-level fault protection included. The resulting fault protection design can be validated by formal analysis and simulation before any costly implementation begins. As spacecraft technology improves, new generic fault protection logic may be added, allowing active improvements to be made to the foundation.

Thesis Supervisor: Dr. Nancy G. Leveson
Title: Professor of Aeronautics and Astronautics

Acknowledgements

I would like to thank Dr. Nancy Leveson, my thesis supervisor for her generous and immovable support. Your continued guidance will be greatly appreciated.

I would like to thank Kathryn Weiss for her contributions to this thesis project. We were put on this project because we were the only two Americans in the lab, but working with you has really been an intellectually, culturally, and socially enjoyable journey. I could not have asked for a more clever, understanding, and downright awesome thesis partner.

I would like to thank my family for their unwavering guidance and support. I would like to thank them for their understanding, their patience, their will, and their dreams. Without you, nothing I do will be worth the effort.

I would like to thank the wonderful people of the Software Engineering Research Lab. Thank you for being yourselves. It is always a pleasure to come into work every morning because of all of you. You have provided countless hours of laughs and entertainment, a commodity that is more valuable than any office can provide. I would especially like to thank JK, Karen, Mirna, Ed, and Natasha for your guidance in all matters of research, MIT, and life in general. Lastly, I would like to thank my fellow UCLA compatriots here at MIT, Damian, and Geoff. We've known each other since our first years at UCLA and it has been a tremendous pleasure to know and work with you guys. I hope we can continue with our weekly pilgrimage to Baja Fresh, continue to dominate on the sand courts, and continue to root for our beloved Bruins for years to come.

Table of Contents

* Appendix section (Pressure Regulation System) pages 1-32 have been ommitted from the Archives copy. This is the most complete version available.

List of Figures

1. Introduction

1.1. The Problem

Spacecraft science missions have traditionally been unique entities. The challenges of sending a probe to Saturn can be quite different from the challenges of rendezvousing with a comet. The uniqueness of each mission drives unique requirements for each spacecraft. These requirements in turn contribute to the diverse designs of spacecraft.

The unique design of each spacecraft is perhaps the single greatest challenge to developing cost-effective and reliable spacecraft fault protection software. Fault protection is system specific and traditionally cannot be designed or written until there is adequate information about the detailed spacecraft design. In addition, the process of designing a spacecraft is iterative, requiring many rounds of redesign: The design of the spacecraft in the initial design phases is likely to be very different from the final design. For this reason, fault protection software is often not implemented until the very end of the spacecraft design phase. This delay means that the fault protection software usually is developed as a separate and distinct entity from the rest of the spacecraft software design and by a separate group of engineers. These engineers may not have adequate first hand knowledge of the components for which they are designing the fault-protection, raising concerns about the correctness of the fault protection. Ideally, the engineers designing the Attitude Determination and Control Subsystem (ADCS), for example, should also design the fault protection software for the ADCS. This goal is difficult to achieve at the present time.

1.2. Related Work

The Mission Data System (MDS) is a project currently being developed at NASA's Jet Propulsion Laboratory (JPL). The aim of the project is to define and develop an advanced multi-mission end-to-end information architecture for deep space missions [2]. At the core of MDS is the idea of a state variable, which is a software object that captures information about the state of external observations of the system. State variables can be defined for almost any spacecraft property such as the state of a device (On/Off), physical properties such as the spacecraft attitude, temperature, mass, and tank pressure, or spacecraft parameters such as calibration coefficients. Using state variables, MDS forms goals, which are simply state variables with prioritized constraints in a given time interval. An example of a goal would be to be at a certain attitude during a particular time frame. Multiple goals combine in a hierarchy to form closed-loop control of spacecraft functions. The process of achieving goals involves elaborating high-level goals into lower-level achievable goals. As an example, the high-level goal of achieving a particular spacecraft attitude would be elaborated into low-level closed-loop command and control of individual attitude determination devices such as reaction wheels or reaction thrusters.

In MDS, fault protection is an integral part of the spacecraft design [7]. Fault protection is a result of the goal achievement process. Goals are achieved when an estimated state variable matches the observed state variable. If a goal is not met through the first elaboration of low-level goals, MDS searches for other ways to achieve the goal. In addition, the operational state of individual components such as a valve that has become stuck is explicitly modeled as state variables. Using this information and the result of the goal elaboration control, MDS determines and isolates the failed component, then reconfigures the system in order to achieve the high-level goal.

The objective of MDS is to provide a cost-effective way of developing spacecraft system software by specifying high-level goals, which can be elaborated and implemented with relative ease during spacecraft design or carried out autonomously onboard the spacecraft during its operation. By specifying high-level goals, the cost of developing expensive low-level code is minimized or eliminated. At present, MDS is still under development and no formal study has been conducted to evaluate the potential promises of the goal-based software architecture.

1.3. Thesis Contribution

This thesis outlines an alternative concept for developing cost-effective and reliable spacecraft fault protection. The idea is to develop spacecraft fault protection software by building and combining generic component specifications. The resulting component-based architecture aims to achieve the goal of cost-effective and reliable fault protection in two ways. First, fault protection design is integrated into the design of the rest of the spacecraft software. In so doing, design decisions about fault protection can be taken into account from the beginning of the spacecraft design phase. Moreover, by integrating fault protection into individual components, the same engineers working on a particular component or subsystem of the spacecraft will also be able to specify and design the fault protection related to that component or subsystem level, thus guaranteeing that fault protection is designed by engineers with first hand knowledge of that part of the system.

The second advantage of reusable architectural components is the ability to capture domain knowledge. Given the variety of spacecraft designs, the fundamental design principles of a spacecraft are and have remained relatively the same. All spacecraft share common requirements for power, attitude control, communication, etc. Furthermore, the technologies used to meet these requirements have also remained relatively static, a result of the high cost of space missions and the resulting risk-driven designs used to meet these requirements. As an example, the techniques and technologies used to perform attitude determination have been confined to a few varieties: inertial based (gyros), celestial vector referenced (sun sensors, star trackers, earth sensors), or magnetic field referenced (magnetometer). (The recent introduction of GPS based attitude determination can be regarded as a fundamentally new and unique technology.)

This is not to say that the techniques and technologies used on spacecraft today are the same as those used on the first space missions. The field of spacecraft design has certainly evolved, but the underlying principles have remained the same. Each time a spacecraft is designed, engineers discover improvements that make the existing techniques more accurate or more robust. Like most engineering disciplines, spacecraft design is an evolutionary rather than a revolutionary process.

The knowledge required to design a complex system such as a spacecraft is extensive and much of it must be gained through experience. There are not many people who have the expertise to design and build a system that can travel in the harsh and relatively unknown environment of space, across vast distances measurable by the time traveled by light. The importance of capturing this expertise becomes ever more essential as the complexity of spacecraft increases and new engineers replace those who retire.

The primary goal of this research is part of a wider effort to capture, store, and ultimately reuse domain knowledge intrinsic to spacecraft fault protection and spacecraft design in general. Past attempts to reuse design have been at the implementation or code level, resulting in some spectacular losses (the Ariane 501 [6] and Mars Climate Orbiter [3]). This thesis describes a new method of capturing and reusing spacecraft design, not at the implementation level but at the specification level, effectively creating a reusable spacecraft architecture. Specifications and models of components would be combined by the designers in a plug-and-play environment to create subsystems and ultimately a spacecraft design. Then changes would be made by the designers to tailor the component designs to the unique spacecraft requirements and the design validated using expert review, formal analysis, and simulation. Only after this system design validation is completed would implementation begin.

1.4. Thesis Outline

The structure of this thesis is divided into three parts: Background, Implementation, and Discussion. The background chapter is devoted to research concepts. Specifically, component-based spacecraft architecture, the concept of creating and reusing a generic library of spacecraft components to build unique spacecraft design specifications is discussed. Intent specifications and the benefits of using intent specifications to implement the component-based spacecraft architecture are also detailed in this chapter.

The implementation chapter covers specific details of applying the component-based spacecraft architecture to actual spacecraft fault protection software. Cassini's overpressurization fault protection software is used to demonstrate the application of the research concepts outlined in the background chapter.

The discussion chapter involves an informal comparison of the implementation of fault protection software using the component-based spacecraft architecture and current spacecraft fault protection architectures. Lastly, some possible issues and concluding remarks are included in this section.

2. Background

2.1. Component-Based System Engineering

The fundamental research concept behind this thesis is reusable component-based system engineering, which is the process of creating a system from components with pre-specified interfaces [10]. Unlike component-based software engineering, which seeks to reuse code or design at the implementation-level, the goal of component-based system engineering is to store domain knowledge by capturing generic design elements at the specification-level. By reusing design at the specification-level, the negative effects of code reuse are largely avoided, but the opportunity to significantly lower development cost still exists.

2.1.1. Generic Component Library

The first step in creating a library of reusable components is to determine how to divide the system into components. For a spacecraft, the easiest way to accomplish this task is to decompose the spacecraft design by functions. Most spacecraft can be decomposed into a number of different subsystems: ADCS, Power, Thermal, Guidance and Navigation (GNC), Command and Data Handling, Payload, etc. These subsystems each have specific and distinct functions that are required for any spacecraft. For example, every spacecraft must be able to properly determine its attitude and provide some means of controlling its orientation. Within each subsystem, there are a number of specific technologies available to accomplish the functions of that subsystem. As illustrated in Figure 1, for the ADCS subsystem, these include IMUs, sun sensors, star trackers, RCSs, and RWAs.

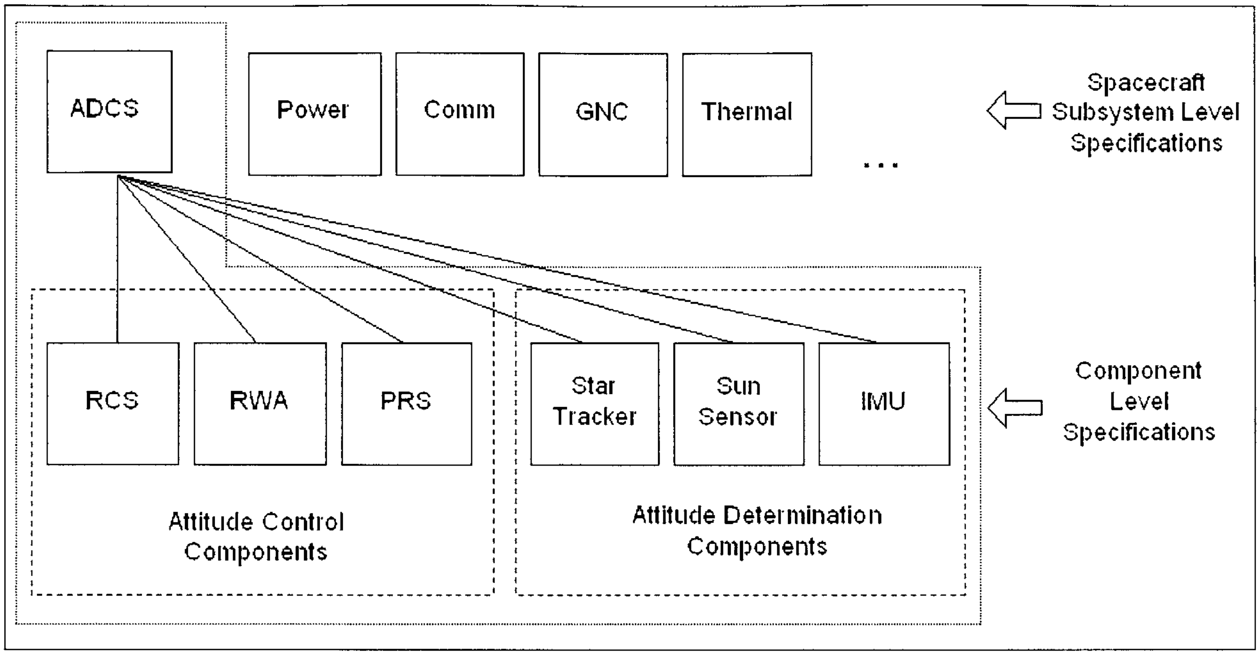


Figure 1 - Spacecraft Decomposition

2.1.2. Generic and Reusable

The success of any reuse process, whether at the implementation or specification level is dependent on the reusability of each component in the library. Each component specification must be easily tailored to fit unique spacecraft designs. To accomplish this, each component specification must be generic. Generic specifications can be created for groups of specific spacecraft technologies. As an example, there are primarily only two types of sun sensors currently available: analog or digital [9]. Functionally, all digital sun sensors work in the same manner and therefore, a generic specification of a digital sun sensor can be created that can be easily tailored to any digital sun sensor model. Details specific to a particular digital sun sensor such as the number of gray code bits used are highlighted in the generic specification document so that system engineers may easily identify and make appropriate changes during the actual design of the spacecraft.

2.1.3. Integrated Fault Protection

Each generic component specification describes the component's hardware and software. Each component specification also includes generic fault protection features such as redundancy management. By specifying basic fault protection features for each generic component and building a complete spacecraft specification from these components, fault protection is included in the design of the spacecraft from the very beginning. Instead of being a separate entity that cannot be completed until a final spacecraft design has been completed, fault protection becomes an integral part of the general spacecraft design process. The design requirements for fault protection can be used to drive design decisions from the beginning of the spacecraft design process. In addition, the engineers designing a particular component or subsystem will be responsible for designing the fault protection at that component or subsystem level, thus guaranteeing that fault protection is designed by engineers with first hand knowledge of that part of the system.

2.1.4. Domain-Specific Specifications

The reusable component library must be domain-specific to spacecraft engineering. There are primarily two reasons for restricting the component library to a specific domain. First, the main motivation of this research is to capture domain knowledge. The best way to capture domain knowledge is to have domain experts working on the design of actual spacecraft be in charge of creating and updating the reusable library. Like any engineering discipline, much of the knowledge required to design the system is gained through experience. The best way to ensure that this knowledge is captured properly is to provide a method by which those with direct first-hand knowledge of the system can capture and store the rationales, assumptions, and constraints involved with the design of the system. In Section 2.2, a specification methodology that supports domain knowledge capture will be introduced.

To test the concept of a component-based spacecraft architecture with reusable specifications, a small library of reusable, generic spacecraft component specifications has been created. These components include the ones shown in Figure 1. A generic ADCS subsystem specification has also been created. In the next chapter, a generic Pressure Regulation System (PRS) and an ADCS specification are modified and combined to form Cassini's overpressurization fault protection software to prove the applicability of the component-based spacecraft architecture.

2.2. Intent Specifications

The success of a reusable specification architecture (and indeed any extensive reuse) depends on capturing the design rationale and assumptions upon which the reusable component designs rest. This goal is accomplished by using intent specifications [5]. While the concept of reusable specification is not specific to a particular specification methodology, there are many benefits to creating the reusable library using intent specifications. Intent specifications were designed to aid engineers in managing requirements, design, and evolution of complex systems and a toolset exists to support the methodology.

The features of intent specifications and the toolset (called SpecTRM) that are the most important in the context of this thesis are:

- Intent specifications enforce good documentation practices, and they provide a means to specify and easily link requirements, design rationales, and design assumptions. The links support traceability and the ability to find the places that need to be changed when instantiating the generic specifications.

- Intent specifications were designed to facilitate communication between people from various disciplines. An intent specification is meant to be readable by engineers, managers, operators, maintainers, and anyone who is working on the project. Cognitive engineering and systems theory research were used to insure that the methodology and toolset can be picked up and used with ease. No formal training is required to read and create intent specifications.
- SpecTRM integrates the results of hazard analysis and other safety-related information into the engineering specifications, which aids in the design of fault protection mechanisms to mitigate the identified hazards.
- Intent specifications and SpecTRM include a formal modeling language called SpecTRM-RL, which allows users to formally model the blackbox behavior of the system. Having a formal model of the system allows for mathematical analyses and simulation of the system behavior before any costly code is implemented.

2.2.1. Human-Centered Specifications

Intent specifications is a specification methodology developed by Dr. Nancy Leveson and her students to assist engineers with the design and evolution of complex systems. Based on systems engineering principles, the methodology is designed from a human-centered perspective. The idea is that software and systems in general are human products, and specifications are tools that allow people to communicate ideas and solve problems. By building a specification methodology based on the way humans naturally perceive and tackle problems; typical engineering processes such as requirements capture, design, review and verification, maintenance, and evolution can be greatly enhanced [5].

2.2.2. Intent Specification Structure

	Environment	Operator	System and Components	V&V
Level 0	Project management plans, status information, safety plans, etc.			
Level 1 System Purpose	Assumptions Constraints	Responsibilities Requirements I/F Requirements	System Goals, High-level Requirements, Design Constraints, Limitations	Hazard Analysis
Level 2 System Principles	External Interfaces	Task Analyses Task Allocation Controls, displays	Logic Principles, Control Laws, Functional Decomposition and Allocation	Validation Plans and Results
Level 3 Blackbox Models	Environment Models	Operator Task and HCI Models	Blackbox Functional Models, Interface Specs	Analysis Plans and Results
Level 4 Design Rep.		HCI Design	Software and Hardware Design Specs	Test Plans and Results
Level 5 Physical Rep.		GUI and Physical Controls Designs	Software Code, Hardware Assembly Instructions	Test Plans and Results
Level 6 Operations	Audit Procedures	Operator Manuals Maintainance Training Materials	Error Reports, Change Requests, etc.	Performance Montitoring and Audits

Figure 2 - Intent Specification Abstraction

Intent specification is structured into seven levels as shown in Figure 2. Unlike most specification methodologies, each level of intent specification does not represent refinement of the system but rather views of the same system from different perspectives, shown in the vertical dimension. Levels 0, 1, 2, and 3 of intent specifications include the information generated during system design. Levels 4, 5, and 6 involve details carried out at implementation and during operation of the system.

The horizontal dimension is divided into four parts. The first column depicts information about the system's environment. The second column contains information related to human operators such as human factor design. The third column represents information regarding the system itself, and the fourth column integrates the verification and validation processes involved at each level. The structure of intent specification is designed with the purpose of capturing intent. Information at every level and column can be linked to form a specification, which answers "Why" in addition to "What" and "How". While Figure 2 provides a good graphical depiction of an intent specification, a specification document is linear. The translation of Figure 2 into a linear document requires nesting subsections. Figure 3 shows how an intent specification document is typically organized for Levels 0-3.

Each generic spacecraft component specification was captured in an intent specification from Level 0 to Level 3. Levels 4-6 were beyond the scope of this thesis since they involve specific implementation details. In the next chapter, the intent specification of a Pressure Regulation System and its associated overpressurization fault protection are analyzed in detail. A copy of the actual intent specification can also be found in the appendix.

1. Level 0 – Program Management Plans
    a. Program Management Plans
    b. System Safety Plans
2. Level 1 – System Level Goals, Requirements, and Constraints
    a. Introduction
    b. Historical Information
    c. Environment Description
    d. Environment Assumptions
    e. Environment Constraints
    f. System Functional Goals
    g. High-Level Requirements
    h. Design and Safety Constraints
    i. Operator Requirements
    j. System Interface Requirements
    k. System Limitations
    l. Hazard List and Hazard Log
    m. Hazard Analysis
    n. Verification and Validation
3. Level 2 – System Design Principles
    a. System Interface Design
    b. Controls and Displays
    c. Operator Task Design Principles
    d. System Design Principles
    e. Verification and Validation
4. Level 3 – Blackbox Behavior
    a. Behavioral Assumptions and Models of the Environment
    b. Communication
    c. User Model
    d. System Blackbox Behavior
    e. Verification and Validation

Figure 3 – Intent Specification Organization

3. Implementation

3.1. Cassini Pressure Regulation System Overview

To investigate the feasibility of component-based specification reuse of fault protection design, a generic Pressure Regulation System (PRS) specification was modified and combined with a generic ADCS subsystem specification to create a Cassini-like PRS fault protection specification [4].

The generic PRS specification is based on a design common to bipropellant propulsion systems [1]. The system includes three tanks, one for the pressurizing gas, which is Helium in this case, and two for the propellants (oxidizer and fuel). Besides the tanks, all other components of the PRS are dual redundant. Passive pressure regulators regulate the flow of Helium into each propellant tank. Latch valves and pyrotechnic valves are designed to allow the system to reconfigure to a redundant branch should the primary branch fail. The generic PRS specification has one dual-string pressure transducer for each propellant tank. Cassini's PRS includes an extra pressure transducer on the fuel tank. A block diagram of our simplified implementation of Cassini's PRS is shown in Figure 4.

The generic PRS specification is linked to a generic ADCS subsystem specification. The ADCS specification includes requirements and design principles associated with a typical ADCS subsystem. The ADCS specification is linked to other components specific to a particular spacecraft design such as Cassini. The interface between the PRS and ADCS are described in sections 3.3.10 and 3.4.1.

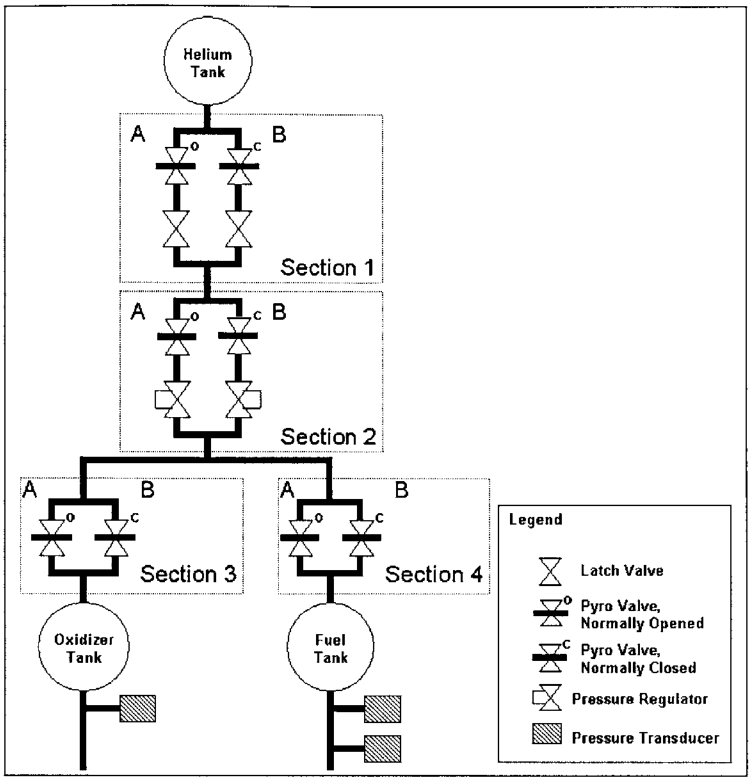


Figure 4 – Cassini PRS Block Diagram

Cassini's fault protection software is a traditional monitor and response algorithm. The PRS fault protection algorithms monitor telemetry from pressure transducers on each propellant tank and declare an overpressurization event if the telemetry exceeds predefined limits. Following the detection of an overpressurization event, the algorithm invokes a response, which sends predetermined commands to latch valves and pyrotechnic valves to cut off the flow of Helium into the propellant tanks.

There are two separate monitors and responses for the PRS. The design of each monitor is the same except for different overpressure limits. Each monitor checks the inputs from all three pressure transducers and declares an overpressurization event if two out of three pressure transducers are above the pressure limit.

Both responses are designed to cut off the flow of Helium through the prime regulator. The first response attempts to stem the flow through the prime regulator by closing the primary latch valve. The second response is designed to further isolate the prime regulator should the first response fail to stop the overpressurization event. The failure of the first response can result from the failure to command the latch valve or a failure of the latch valve itself. The second response initiates and engages pyrotechnic valves between the Helium tank and the prime regulator, permanently shutting the flow through the primary regulator.

Cassini's overpressurization fault protection algorithms are part of the larger spacecraft-wide fault protection software. In addition to other subsystem-specific fault protection algorithms, the software contains responses for system-level faults or faults that affect the operation of the entire spacecraft such as loss of attitude. The scope of this thesis is limited to the discussion of the component-level and subsystem-level fault protection.

3.2. Level 0 – PRS Intent Specification

Level 0 of intent specifications represents the system from the program management perspective. Information at this level includes organization-specific details such as program management plans and system safety policies. Level 0 is not specified for the PRS or any other generic spacecraft component specification. This level is dependent on specific organizational policies. The program management plans and safety policies of JPL, for example, might be very different from those at NASA Goddard or Boeing. The content and responsibility of completing Level 0 is best decided by individual organizations.

3.3. Level 1 – PRS Intent Specification

Level 1 is designed to provide users with a high-level conceptual view of the system. Level 1 can be described as the customer's view of the system. At this level, system goals are defined for each column depicted in Figure 2. Goals are very general statements of what the system should do. Refinement at this level involves taking functional goals and determining testable high-level requirements, system and environmental constraints, assumptions, and limitations. Also included in Level 1 are analyses on system properties such as the preliminary and system-level hazard analyses. Following the outline shown in Figure 3, the implementation of Level 1 for the PRS specification is now described in detail:

3.3.1. Introduction

The introduction is designed to give readers a broad picture of the system being described in the specification document. The introduction for the PRS is stated as follows: *

> The Pressure Regulation System (PRS) is a part of the propulsion subsystem. Its function is to regulate the pressure of the oxidizer and fuel tank onboard the spacecraft. The pressure regulation system consists of several components. They are the Helium tank, the propellant tanks (oxidizer, and fuel tank), latch valves, pyrotechnic valves, passive pressure regulators, pressure transducers, and associated piping. There are many possible designs associated with a given pressure regulation system. The following specification is based on a design common to bipropellant propulsion systems.
>
> The PRS has a software controller, which controls all active components of the PRS. The active components include each latch valve, pyrotechnic valve, and pressure transducer. The tanks, pressure regulators, and piping are purely mechanical devices and cannot be actively controlled during the spacecraft's operation.
>
> For purposes of identification, the latch valves and pyrotechnic valves of the PRS are divided into 4 sections, as depicted in Figure 1. Each section has two separate branches. The latch valve or pyrotechnic valve is named according to the section and branch that it resides.
>
> Section 1, for example, contains two latch valves and two pyrotechnic valves. The components on the left branch are labeled A and the components on the right branch are labeled B. For example, the latch valve on the left side of Section 1 is referred to as Latch Valve 1A.

* Excerpts from the actual specification documents are indented and formatted in Arial font.

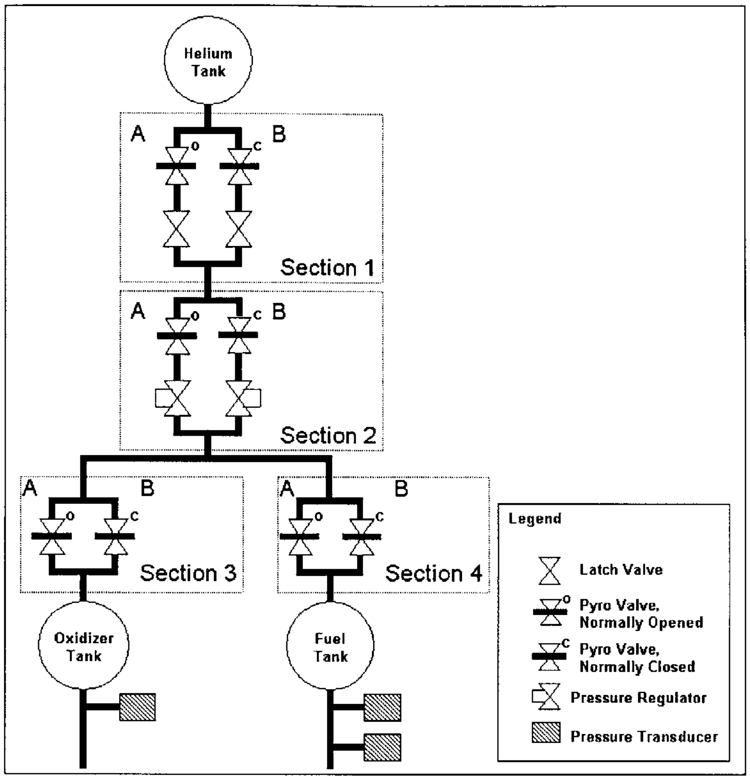


Figure 1

3.3.2. Historical Information

The historical information section of Level 1 provides information regarding previous or similar systems to assist readers in understanding the relation of the system being described to historical systems. The component-based spacecraft architecture described in this thesis and similar architectures such as MDS is included here. An excerpt from the PRS historical information section includes the following:

> This specification is based on the SpecTRM-GSC spacecraft component specification architecture. SpecTRM-GSC is a component based specification language designed specifically for the domain of spacecraft engineering.

3.3.3. Environment Description

The environment description section of Level 1 describes information about the environment in which the system is expected to operate. The PRS environmental description is given below:

> The PRS is only one component in a complex spacecraft environment. The PRS is part of the Propulsion Subsystem. Its function is to regulate the pressure within the oxidizer and fuel tanks. The oxidizer and fuel tanks contain propellants used for delta V maneuvers and attitude control of the spacecraft. As a result, the function of the PRS has a direct effect on the Attitude Determination and Control Subsystem (ADCS) of the spacecraft and interacts directly with the ADCS subsystem controller. The PRS will receive commands from the ADCS and alert the ADCS when reconfiguration or safing is needed.
>
> Like every component on the spacecraft, the PRS is expected to operate in the harsh environment of space. All components are expected to operate nominally in complete vacuum. Depending on the phase of the mission, the spacecraft will also experience a wide range of temperatures. All components are required to meet the mission survivable temperatures limits.
>
> *Note: Refer to the Thermal Subsystem Specification for information regarding spacecraft thermal design.*

Notes, rationales, and assumptions are documented whenever possible. These comments, shown in italics, are vital for capturing intent. They provide readers of the specification with a better understanding of the system by answering the "whys" in addition to "what" and "how."

3.3.4. Environment Assumptions

Assumptions about system in which the system will operate are documented in Level 1. The correct operation of the system and the hazard analyses will be dependent on these assumptions. The PRS has one environmental assumptions stated below:

[EA.1] Ground commands can only be received when the spacecraft is in direct sight of the Deep Space Network. There will also be a time lag between the time the signal is sent and when it is received. [→OR.1] [→OR.2]

Rationale: The spacecraft is not always in direct contact with Earth. At various times in the mission, the spacecraft will be traveling behind celestial bodies. Also, given the distance between Earth and the spacecraft, the command signals will take some time to reach and arrive from the spacecraft.

3.3.5. Environment Constraints

Environment constraints are constraints placed on the environment to ensure the proper function of the system.

[EC.1] Components from other parts of the spacecraft must not physically interfere with the function of the PRS.

3.3.6. System Functional Goals

System functional goals are usually developed in the early stages of the project and stated in very general terms. They are statements that provide a broad overview of what the system is to achieve. The functional goal of the PRS is stated as follows:

[FG.1] The PRS shall regulate the pressure of the oxidizer and fuel tanks to a specified pressure range. [→FR.1]

3.3.7. High-Level Requirements

One of the first steps during the development of any system is to come up with high-level requirements that will achieve the system functional goals. High-level requirements must be written in a form that is testable. For the PRS, the high-level requirements are stated as follows:

[FR.1] The PRS shall pressurize the oxidizer and fuel tanks by releasing Helium, which is stored at a higher pressure into the oxidizer and fuel tanks. [←FG.1]

Note: Since the Helium tank is connected to the oxidizer and fuel tanks, Helium will flow into the oxidizer and fuel tanks until the pressures in all three tanks are equalized unless there is pressure regulator in between the tanks.

Note: Pressure is a function of temperature. The pressure of the oxidizer and fuel tank will vary as their contents are expelled (endothermic chemical reaction). The temperature of the tank will also vary depending on the position of the spacecraft in relation to the sun. Heaters provide active control of the tank temperature. See Thermal Subsystem Specification. [↓DP.6.1]

FR.1 is a requirement, which achieves the goal FG.1. FR.1 is linked to the functional goal FG.1 as well as design principles in Level 2 of the intent specification document. Design principles in Level 2 state how the requirement will be satisfied. Adding hyperlinks such as the ones shown above form links between various sections in the specification document. SpecTRM provides a user-friendly interface to create and modify links as required. Up and down arrows are used to show links that are formed between two different levels while forward and backward arrows point to sections within the same level.

[FR.2] The PRS shall prevent an overpressurization of the oxidizer and fuel tanks. [→HL.1]

[FR.2.1] The PRS shall detect an overpressurization event in the oxidizer and fuel tank by checking the pressures in the oxidizer and fuel tanks. [↓DP.6]

Assumption: The pressure transducers are calibrated and provide accurate information on their associated tank pressures.

Note: Pressure readings are in pounds per square inch absolute (psia).

[FR.2.2] The PRS shall take appropriate action to stem an overpressurization event with automated responses.

[FR.2.2.1] Upon detecting an overpressurization event, the PRS will first attempt to stem the flow of Helium into the oxidizer and fuel tanks by closing the primary latch valve (Latch Valve 1A) within 16 seconds of detecting the overpressurization event. [↓DP.7]

[FR.2.2.2] If the pressure of the oxidizer and fuel tank continue to rise, the PRS shall command the appropriate pyrotechnic valve to engage and stop the flow of Helium within 120 seconds. [↓DP.8]

Note: The pyrotechnic valve cannot be reset. Once it is engaged, the valve will be closed for the remainder of the spacecraft's lifetime. Ground controllers must reconfigure the PRS to use the secondary branch in the event that the primary pyrotechnic valve is fired. [→SC.4]

[FR.3] The automated overpressurization fault responses shall be enabled and disabled by ground controllers or higher-level spacecraft commands. [↓DP.5]

Note: Higher-level spacecraft commands may be issued from subsystem controllers during various phases in the mission.

Refinements are performed within each section by nesting subsections as illustrated in FR.2.

3.3.8. Design and Safety Constraints

The design and safety constraints section is divided into non-safety related constraints and safety constraints. Constraints restrict how the system may achieve its goals. The PRS specification has four safety-related constraints.

[SC.1] The first overpressurization response must be executed within 16 seconds of reaching the Stage-1 overpressurization limit of 269 psia.

[SC.2] The second overpressurization response must be executed within 120 seconds of reaching the Stage-2 overpressurization limit of 380 psia.

[SC.3] The first overpressurization response must be executed before the pressure has reached the second response limit.

Rationale: The two overpressurization responses are not designed to execute in parallel with each other.

[SC.4] The pyrotechnic valves must not be engaged inadvertently. [→HL.2] [←FR.2.2.2]

Rationale: Pyrotechnic valves can only be engaged once. Once the valve has been engaged, it can never be returned to the previous position.

3.3.9. Operator Requirements

Operator requirements are stated to aid in the design of the human-computer interface, system logic, operator procedures, operator documentation, and training plans. The operator requirements captured for the PRS includes the following:

[OR.1] Ground controllers will monitor all telemetry from the spacecraft including telemetry specific to the PRS such as the position of each valve and the pressures of the oxidizer and fuel tanks. [←EA.1]

[OR.2] If an overpressurization response is carried out, ground controllers will reconfigure the PRS to redundant branches by commanding the appropriate latch valves and pyrotechnic valves. [←EA.1]

Note: The automated overpressurization responses are designed to shut off the flow of Helium into the oxidizer and fuel tanks, which prevents the regulation of pressure in the oxidizer and fuel tanks.

3.3.10.System Interface Requirements

System interface requirements documents the human-computer interface such as controls, displays, and aural alerts.

[SIR.1] The ground controller will enable and disable automated fault protection at any time by sending a command to the spacecraft. [↓CNTRL.1]

Assumption: There is a communication link between the spacecraft and the ground.

[SIR.2] The ground controller will manually command any active components of the PRS by sending individual commands to the spacecraft. [↓CNTRL.1]

Assumption: The automation must be disabled for manual commands.

[SIR.3] The ground controller will be able to view all PRS related telemetry. [↓DISP.1] [↓DISP.2] [↓DISP.3] [↓SI.1.5]

Assumption: There is a communication link between the spacecraft and the ground.

3.3.11.System Limitations

System limitations describe functional limitations. They may be related to basic functional requirements such as the following:

[SL.1] Pressurization of the oxidizer and fuel tanks will be limited to the amount of Helium stored. Once the pressure of the Helium tank has equalized with the oxidizer and fuel tanks, pressurization will cease.

3.3.12.Hazard List

The hazard list contains all relevant hazards of the system. The hazard list is an essential part of the system design, including the design of the fault protection. Fault prevention techniques and fault protection mechanisms may be designed by identifying hazards and performing a hazard analysis.

[HL.1] The oxidizer tank and fuel tank pressure is over 269 psia. [←FR.2]

Rationale: Overpressurization of either tank will lead to catastrophic failure of the tanks, causing possible rupture and eventual explosion. Any explosion onboard the spacecraft will result in the failure of the entire spacecraft.

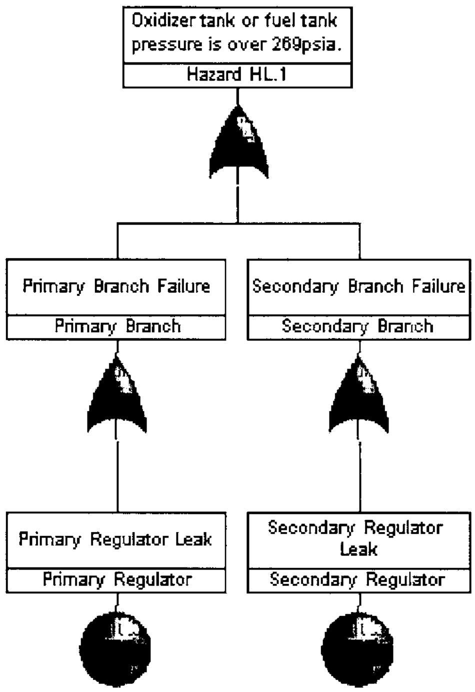
[HL.2] Pyrotechnic valves are fired inadvertently. [←SC.4]

Rationale: The pyrotechnic valve should only be engaged, either automatically by the fault protection software, or manually by ground controllers, when there are no other mechanisms available to stem an overpressurization event.

3.3.13.Hazard Analysis

Hazard analyses are vital for almost any safety-critical system. In intent specifications, the hazard analysis is integrated to the specification process, from the very start. Hazard analyses play an important role in the development of fault protection because fault protection is inherently a safety feature of the system. Various fault protection mechanisms, including hardware and software design as well as fault prevention mechanisms, can be driven by the results of the hazard analysis.

There are various methods of hazard analyses available. A basic fault tree analysis is included for the hazard HL.1 identified in the hazard list.



3.3.14. Verification and Validation

At level 1, the verification and validation requirements list out the various review procedures required for the system design. Participants and the results of the review procedures are also recorded.

Review Procedures

The PRS will involve several formal reviews. Each review will be performed at different stages in the design process. A brief description of each review will be described below:

Requirements Review - The requirements review will take place prior the initial design phase. The review is designed to ensure that all customer requirements are set and understood by both parties.

Formal Design Review - The formal design review will incorporate all aspects of the PRS design. It will take place after the initial design is complete. The formal design review will be focused on the design principles of the PRS. The review will encompass the design project in terms of structural, electromechanical, software, and safety aspects.

Operations Review - The operations review will take place after the PRS has been completed. The operations review is designed to focus discussion on the operations of the PRS with particular attention to ground controller training and procedural requirements.

Participants

Requirements Review - System Engineer, Customer Representative

Formal Design Review - System Engineer, Subsystem Engineers, Safety Engineer, Ground Controller, Outside Reviewer

Operations Review - System Engineer, Safety Engineer, Ground Controller

3.4. Level 2 – PRS Intent Specification

Level 2 of intent specifications contain the system design principles and engineering principles required to satisfy the requirements and constraints at Level 1. Similarly, Level 2 answers the questions "why" for the level below. Information at this level is informal, written normally in the form of English in addition to any applicable mathematical equations. For the PRS fault protection algorithms, Level 2 describes the design of the monitor and response algorithms of the fault protection software.

3.4.1. System Interface Design

To properly specify the interaction of the system with its environment, the interfaces between the system and each external component must be specified. This information is included in this section. For the PRS, the interface design describes the input and output information between the PRS controller and the ADCS controller:

[SI.1] PRS-ADCS Interface

[SI.1.1] The PRS receives an input command from the ADCS controller to enable or disable the automated fault protection software.

[SI.1.2] The PRS receives manual commands from ground controllers via the ADCS controller. Manual commands will include commands for all active components in the PRS.

[SI.1.2.1] There are two possible manual commands for each latch valve: Open and Close.

[SI.1.2.2] There are three possible manual commands for each pyrotechnic valve: Enable, Disable, and Fire.

[SI.1.2.3] Only one manual command is received and executed once every second.

[SI.1.3] The PRS sends the ADCS controller an output message indicating its current control mode. This output is updated once every second. [↓Level 3 Control Mode]

[SI.1.4] When the automated fault protection software has detected an overpressurization event, the PRS sends the ADCS an output message indicating that the system needs to be reconfigured. Automated responses are carried out when a message from the ADCS is received. [→DP.7.1] [→DP.7.2] [→DP.8.1] [→DP.8.2]

Rationale: Before carrying out reconfiguration commands, eg. Closing latch valves and pyrotechnic valves, the spacecraft must be in a safe state and there must be adequate power and other spacecraft resources. The PRS controller derives this information from the ADCS controller.

[SI.1.5] PRS telemetry is sent to ground controllers through the ADCS controller. [↑SIR.3]

[SI.1.5.1] An output message is sent to the ADCS controller once every second indicating the position of each latch valve: Open or Close

[SI.1.5.2] An output message is sent to the ADCS controller once every second indicating the position of each pyrotechnic valve: Open or Close

[SI.1.5.3] An output message is sent to the ADCS controller once every second indicating the state of each pyrotechnic valve: Normal, Enabled, or Fired.

[SI.1.5.4] An output message is sent to the ADCS controller once every second indicating the pressure reading from each pressure transducer.

Note: Each pressure transducer is dual string. There are a total of six pressure readings from the three pressure transducers.

3.4.2. Controls and Displays

The controls and displays section of Level 2 includes information specific to the design of human-computer interface. Information in this section will be in the form of design specification for the ground controller computer interface.

[CTRL.1] Each PRS command has a unique identifier. Ground controllers use a command line interface to send all commands to the PRS controller. Commands are typed in via a keyboard.

[DISP.1] Each PRS telemetry point is displayed on a monitor. Telemetry is updated once every second when there is direct contact with the spacecraft. [↑SIR.3]

[DISP.2] Pressure readings are highlighted in the following scheme: [↑SIR.3]

In green when the reading is between 200-269psia.

In yellow when the reading is between 270-300psia.

In red when the reading is between 301-400psia.

[DISP.3] The state of pyrotechnic valves are highlighted in the following scheme: [↑SIR.3]

In green when Normal.

In yellow when Enabled.

In red when Fired.

3.4.3. Operator Task Design Principles

The operator's expected tasks are documented in the operator task design principles section of Level 2. For the PRS, this includes tasks specific to the manual operation of the PRS. These tasks include the following:

[ODP.1] Ground controllers are required to enable PRS fault protection following spacecraft launch and checkout. PRS fault protection should remain enabled through the cruise phase of the mission.

[ODP.2] Ground controllers are required to ensure that PRS fault protection is disabled before the spacecraft enters orbit insertion mode.

Note: PRS fault protection is automatically disabled by the ADCS controller prior to orbit insertion, but ground controllers must ensure that it has been disabled.

3.4.4. System Design Principles

The system design principles section describes the basic principles or assumptions upon which the system design depends. System design principles are linked to functional goals and high-level requirements, constraints, limitations, hazard analysis as well as to lower level system design and implementation. Design rationales and assumptions must be captured whenever possible to provide readers with the design intent. The PRS fault protection design principles are represented in this section. Also captured here are the design principles related to the operation of the latch valves, pyrotechnic valves, and pressure transducers.

[DP.1] Latch Valve Design Principle

[DP.1.1] Each latch valve has a sensor that reports its current position once every second: Open or Close. [↓Level 3 Latch Valve Inputs]

[DP.1.2] There are two possible commands for each latch valve: Open and Close.

[DP.1.2.1] An Open Latch Valve command is issued to open the latch valve. A response is expected within 5 seconds after the command is sent. If a response is not recorded after 5 seconds, the latch valve is declared as Stuck Closed. [↓Level 3 Latch Valve Open Outputs] [↓Level 3 Latch Valve Status]

[DP.1.2.2] A Close Latch Valve command is issued to close the latch valve. A response is expected within 5 seconds after the command is sent. If a response is not recorded after 5 seconds, the latch valve is declared as Stuck Opened. [↓Level 3 Latch Valve Close Outputs] [↓Level 3 Latch Valve Status]

[DP.2] Pyrotechnic Valve Design Principle

[DP.2.1] Each pyrotechnic valve has a sensor that reports its current position once every second: Open or Close. [↓Level 3 Pyrotechnic Valve Inputs]

[DP.2.2] Each pyrotechnic valve only has one commandable state but requires two separate commands to engage the pyrotechnic mechanism. To fire a pyrotechnic valve, an Enable command must first be sent to enable the pyrotechnic mechanism. This command arms the pyrotechnic device. Following a 90 second delay (required by the pyrotechnic bank to charge), a second command to fire the pyrotechnic device is sent to open or close the valve permanently. A Disable command can be sent to disable the arming mechanism and return the pyrotechnic valve to a normal state. [↓Level 3 Pyrotechnic Valve Outputs]

[DP.2.3] The state of each pyrotechnic valve is inferred from commands sent. The state of the pyrotechnic valve becomes Enabled following an Enable command. If a Disable command is sent, the state returns to Normal. However, if a Fire command is sent after the Enable command, the state of the pyrotechnic valve enters Fired. It remains in this state for the rest of the spacecrafts operational period. [↓Level 3 Pyrotechnic Valve Status]

[DP.3] Pressure Transducer Design Principles

[DP.3.1] The PRS receives 2 inputs from each pressure transducer. The inputs contain information about the current pressure reading of each pressure transducer. The expected range of values is between 200-400 psia. Any input value outside of this range is deemed obsolete. The expected update rate of these inputs are once every second and are deemed obsolete after 5 seconds. [↑FR.1] [→DP.6.1] [↓Level 3 Pressure Transducer Inputs]

[DP.4] Fault Protection Design Principles Overview

The automated overpressurization fault response is an essential feature of the PRS. The purpose of the overpressurization fault protection software is to protect against an overpressurization of the propellant tanks due to a regulator leak [Hazard Analysis]. The overpressurization fault protection software includes two separate fault detection mechanisms and two different fault responses.

Stage 1 Overpressurization Detection and Response Overview: The Stage 1 overpressurization fault protection algorithm declares an overpressurization event when two of the three pressure transducers exceed the Stage 1 qualification pressure limit of 269 psia. The response is designed to close the primary latch valve (Latch Valve 1A) before the pressure has reached the Stage 2 qualification pressure limit of 380 psia.

Stage 2 Overpressurization Detection and Response Overview: The Stage 2 overpressurization fault protection algorithm declares an overpressurization event when two of the three pressures transducers exceed the Stage 2 qualification pressure limit of 380 psia. The response is designed to isolate the prime regulator from the helium tank. This is carried out by engaging the prime pyrotechnic valve (Pyrotechnic Valve 2A) and completely shut off the flow of Helium through the prime regulator.

The enable and disable strategy for the fault protection software is a vital component of the fault protection design. At certain times during the spacecraft mission, it may be necessary to disable the fault protection software to prevent inadvertent system responses from occurring. As an example, during Cassini's orbit insertion, commands not related to spacecraft orbit insertion are de-prioritized until orbit insertion is complete. Without disabling the PRS fault protection, a response, even if a real fault were present, could jeopardize the entire mission by interfering with vital orbit insertion procedures. The fault protection enable/disable strategy design principles is described below:

[DP.5] Fault Protection Enable/Disable Strategy:

[↑FR.3] [↓Stage 1 Overpressurization Control Mode] [↓Stage 2 Overpressurization Control Mode]

[DP.5.1] Both automated Stage 1 and Stage 2 overpressurization fault protection algorithms are enabled by default. They may be separately enabled or disabled via ground commanding or higher-level autonomous spacecraft commanding via the ADCS.

Note: An example of a higher-level autonomous spacecraft command to disable automated PRS fault protection occurs during orbit insertion when automated fault protection is turned off to prevent the automation from interfering with more critical commands.

[DP.5.2] Both fault protection algorithms are disabled following a response. For the Stage 1 Overpressurization algorithm, this occurs after the prime latch valve has been closed. For the Stage 2 Overpressurization algorithm, this occurs after the prime pyrotechnic valve has been enabled and fired.

Note: Ground controllers may re-enable Stage 1 and Stage 2 Overpressurization fault protection, but the effects of this procedure is not known.

Note: Both fault protection algorithms are disabled after the last response command is sent, eg, for Stage 1 response, LatchValve1AClose, and for Stage 2, PyroValve2AFire. This does not guarantee that the response has been successful. For example, the latch valve might have failed and sending a command to close the latch valve won't have any effect on the system.

As stated, the PRS fault protection software has two separate detection and response algorithms. The design of the two fault detection algorithms is the same except for different fault limits corresponding to Stage 1 and Stage 2 overpressurization limits. The detection algorithm design is described in the following design principles:

[DP.6] Stage 1 and Stage 2 Overpressurization Detection Design [↑FR.2.1]

[DP.6.1] Both Stage 1 and Stage 2 Overpressurization fault protection algorithms use pressure readings from the pressure transducer on the oxidizer tank and both pressure transducers on the fuel tank. [←DP.3] [↓Level 3 Overpressurization Fault Status]

Note: Two pressure transducers from the fuel tank are used because the fuel tank is less susceptible than the oxidizer tank to pressure changes from thermal conditions. [↑FR.1]

Note: All three pressure transducers may be powered by the same power source onboard the spacecraft. Appropriate fault prevention design of the power system must be carried out to prevent a single-point failure of all three pressure transducers. See Power Subsystem Specification.

[DP.6.2] Both Stage 1 and Stage 2 Overpressurization fault protection algorithms first determines if a pressure transducer input is valid by checking the following conditions:

1) The input is within the range of 200-400 psia

2) The input was received less than 5 seconds ago

[DP.6.3] The Stage 1 Overpressurization fault protection algorithm declares an overpressurization event for a pressure transducer if either input is above the Stage 1 qualification limit. The Stage 2 Overpressurization fault protection algorithm declares an overpressurization event for a pressure transducer only if both inputs are above the Stage 2 qualification limit. When only one input is valid, both algorithms use that input to determine overpressurization. When neither input is valid, neither algorithms indicate overpressurization for the pressure transducer.

Rationale: The differences in the algorithm logic bias the Stage 1 algorithm away from a false negative (failure to correctly respond to an overpressure) and the Stage 2 algorithm away from a false positive (indication of an overpressure when there is none). This design provides robustness in multiple-fault scenarios without increasing the risk of irreversible actions.

[DP.6.4] Both Stage 1 and Stage 2 Overpressurization fault protection algorithms declares an overpressurization event if two out of the three pressure transducers indicate an overpressure condition persistently for 5 seconds.

Rationale: The two out of three comparison prevents either an inadvertent response trip or a failure to detect an overpressure if there is a faulty pressure transducer. It also prevents an inadvertent response trip due to a thermally induced pressure condition in the oxidizer tank.

Rationale: The persistence condition prevents a transient condition from initiating the response.

The design principles of each fault response is described as follows:

[DP.7] Stage 1 Overpressurization Response Design [↑FR.2.2.1]

[DP.7.1] Upon declaring a Stage 1 overpressurization event, the Stage 1 Overpressurization algorithm sends a message to the ADCS requesting reconfiguration. [←SI.1.4] [↓Level 3 Overpressurization Fault Response]

Rationale: The ADCS is required to stop any attitude maneuvers before the PRS can be reconfigured. Additionally, the spacecraft may need to power off non-essential components to allow enough power for the fault protection response.

[DP.7.2] Upon receiving a "response ready" message from the ADCS, the Stage 1 Overpressurization algorithm closes the prime latch valve (Latch Valve 1A). [←SI.1.4]

[DP.8] Stage 2 Overpressurization Response Design [↑FR.2.2.2]

[DP.8.1] Upon declaring a Stage 2 overpressurization event, the Stage 2 Overpressurization algorithm sends an output to the ADCS requesting reconfiguration. [←SI.1.4] [↓Level 3 Overpressurization Fault Response]

Rationale: The ADCS is required to stop any attitude maneuvers before the PRS can be reconfigured. Additionally, the spacecraft may need to power off non-essential components to allow enough power for the fault protection response.

[DP.8.2] Upon receiving a "response ready" message from the ADCS, the Stage 2 Overpressurization algorithm sends an output to enable the prime regulator pyrotechnic valve (Pyrotechnic Valve 2A). [←SI.1.4]

[DP.8.3] 90 seconds after the enable command is sent, the response sends an output to fire the prime regulator pyrotechnic valve (Pyrotechnic Valve 2A).

Note: This command permanently closes the pyrotechnic valve.

3.5. Level 3 – PRS Intent Specification

The third level of an intent specification is a formal blackbox model of the specification. This model represents the externally observable behavior of the system being designed. The modeling language used is called SpecTRM-RL [5]. It is based on an underlying state machine but uses visualizations and notations that are designed to be easily readable and understood by engineers and anyone else who may have to contribute to the specification. The formal blackbox model of the PRS specification is described in this section with details of the modeling language as required to explain the model. For a more thorough description of the SpecTRM-RL modeling language, see the SpecTRM User Manual [8].

3.5.1. Formal Blackbox Model

The SpecTRM-RL modeling language is made up of several components:

- **Inputs** describe data coming into the system. All inputs have timing constraints to handle obsolescence.
- **Outputs** describe data leaving the system. Outputs also support timing constraints.
- **Modes** represent control modes or supervisory modes controlling the system.
- **State Variables** represent the states that can be inferred from the relationship between inputs and outputs of the system. State variables are enumerated values modeled to represent discrete transitions in the system.
- **Macros** are Boolean expressions abstracted out to simplify transition table logic.
- **Functions** calculate outputs and intermediate values.
- **Devices** are external to the system. They are sources of inputs and sinks of outputs. Devices represent boundary points of the system.
- **Messages** are used to convey information about the values of inputs and outputs.

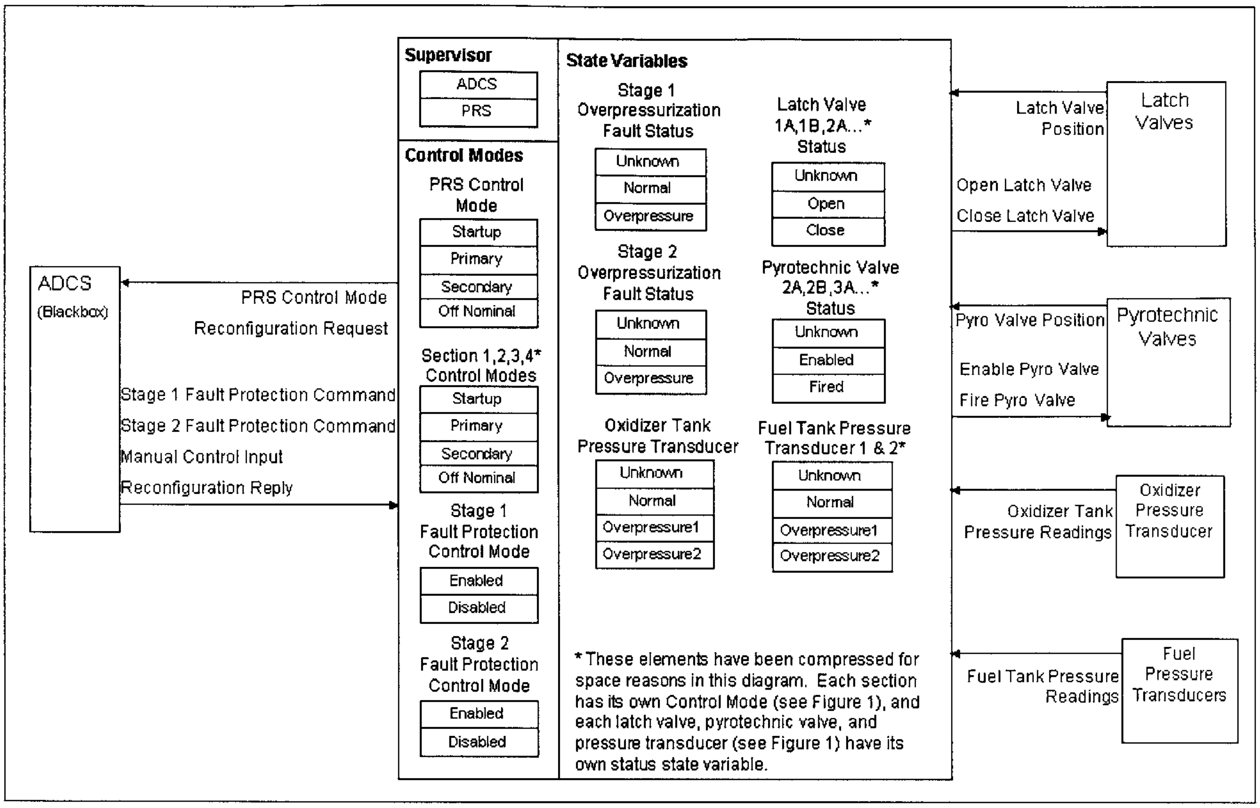


Figure 5 – PRS Blackbox Model

Using these SpecTRM-RL components, the blackbox behavior of the PRS software controller was modeled. The representation of the model is shown in Figure 5. The large gray box represents the PRS controller. The smaller boxes to the right are devices controlled by the PRS controller. These devices include the latch valves, pyrotechnic valves, and pressure transducers. The ADCS controller, shown on the left, controls the PRS. Control modes and state variables are shown within the system boundary. Supervisory modes are differentiated from other control modes to prevent mode confusion errors.

Logical expression in SpecTRM-RL are captured using a tabular notation known as an AND/OR table. An example of an AND/OR table used to describe the value of a control mode is shown in Figure 6.

Stage1OverpressureFaultProtection

Control Mode

Description: Determines Stage 1 Overpressurization Fault Protection Enabled or Disabled

Comments: NA

References: Stage1OverpressurizationResponse, Stage1FaultProtectionCommandInput, ↑DP 5

Appears In: LatchValve1AClose

DEFINITION

= Enabled

Power Up	T	F
Stage1FaultProtectionCommandInput is Enable	*	T
Stage1OverpressurizationResponse has Never Entered ResponseComplete	*	T

= Disabled

Power Up	F	F
Stage1FaultProtectionCommandInput is Enable	T	F
Stage1OverpressurizationResponse has Never Entered ResponseComplete	F	*
Stage1FaultProtectionCommandInput is Disable	F	T

Figure 6 – SpecTRM-RL Control Mode

The rows of the tables indicate AND relationships, while the columns represent ORs. An asterisk denotes a "don't care" condition. If any of the columns evaluates to TRUE, then the input takes the value shown (Enabled or Disabled). For example, the Stage 1 Overpressure Fault Protection Control Mode will transition to Enabled if the system is in Power Up (the system has just started) or the system is not in Power Up, but the component has received a command to enable fault protection AND the Stage 1 Overpressurization Response has never been executed (the fault response is designed to execute only once after which the response is disabled).

Also shown in Figure 6 above the AND/OR table are attributes relevant to the control mode specified. These attributes are included for each SpecTRM-RL element. They specify a range of information specific to each element such as timing behaviors, obsolescence, acceptable values, and feedback information. Attributes shared among all SpecTRM-RL elements include descriptions, comments, and references, which are simply links to other parts of the specification documents. As shown in Figure 6, the Control Mode element is linked to a design principle in Level 2. Links between Level 3 blackbox elements are automatically generated by the SpecTRM toolset.

State variables are used to model the controller's current representation of the plant or controlled system. Examples of state variables in the PRS blackbox model include the status of each valve and the inferred pressure states in each propellant tank, which are based on inputs from the pressure transducers.

The logic for when state variables take particular values are also described with an AND/OR table. The following is a description of the logic for determining a Level 1 overpressurization:

Stage1OverpressurizationFaultStatus

State Value

Obsolescence: 5 seconds

Exception-Handling: NA

Related Inputs: NA

Description: The Stage1OverpresurizationFaultStatus detects a Stage 1 overpressurization event.

Comments: NA

References: Stage1OverpressureCheck, tDP.fi

Appears In: ReconfiguringOutput, LatchValve1AClose, Stage1OverpressurizationResponse

DEFINITION

= Unknown

Power Up	T

= Normal

Power Up	F
Stage1OverpressureCheck	F

= Overpressure

Power Up	F
Stage1OverpressureCheck	T

Figure 7 - SpecTRM-RL State Variable

As specified in Level 2 (pg. 35), the design principle for declaring overpressurization requires 2 of the 3 pressure transducers to be over the overpressure limit. This logical construct is implemented by the state variable represented in Figure 7. The current value of the Stage 1 Overpressurization Fault Status depends on the value of the Stage 1 Overpressurization Check, which is defined as a macro and shown in Figure 8. Macros are simply separate AND/OR tables designed to manage the complexity of the specification.

Stage1OverpressureCheck

Macro

Description: This macro determines whether the Stage 1 overpressure limit has been reached. Stage 1 overpresure limit is declared when 2 out of the 3 pressure transducers on the oixidizer tank(1 pressure transducer) and fuel tank(2 pressure transducer) has reached overpressure Stage 1 limits.

Comments: NA

References: FuelPressureTransducer2Status, OxidizerPressureTransducerStatus, FuelPressureTransducer1Status

Appears In: Stage1OverpressurizationFaultStatus

DEFINITION

OxidizerPressureTransducerStatus in state OverPressure1	T	*	T
FuelPressureTransducer1Status in state OverPressure1	T	T	*
FuelPressureTransducer2Status in state OverPressure1	*	T	T

Figure 8 - SpecTRM-RL Macro

Input values are also described using an AND/OR table. An example of an input value is shown below in Figure 9.

OxidizerTankPressureA

Input Value

Source: <u>OxidizerPressureTransducer</u>

Message: <u>OxidizerPressureTransducerAMessage</u>

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: NA

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: None

Latency: NA

Min-time-after-output: NA

Exception-Handling: NA

Obsolescence: 5 seconds

Exception-Handling: NA

Description: String A Input from Oxidizer Pressure Transducer

Comments: NA

References: <u>OxidizerPressureTransducerAMessage</u>, <u>OxidizerPressureTransducer</u>, <u>1DP 3.1</u>

Appears In: <u>OxidizerStage2Check</u>, <u>OxidizerAObsoleteCheck</u>, <u>OxidizerStage1Check</u>, <u>OxidizerPressureTransducer</u>

DEFINITION

= Field Pressure from <u>OxidizerPressureTransducerAMessage</u>

Message for <u>OxidizerTankPressureA</u> was Received	T

= Previous Value

Message for <u>OxidizerTankPressureA</u> was Received	F
Time Since Message for <u>OxidizerTankPressureA</u> was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for <u>OxidizerTankPressureA</u> was Never Received	*	T	*
Time Since Message for <u>OxidizerTankPressureA</u> was Last Received > 5 seconds	*	*	T
Message for <u>OxidizerTankPressureA</u> was Received	*	*	F

Figure 9 - SpecTRM-RL Input

The input described in the figure above is a pressure reading from the pressure transducer on the oxidizer tank. The input changes whenever a new input is received from the external device through a message. The input takes on the previous value if a new value has not been received within the last 5 seconds of receiving the previous input. Finally, input obsolescence is explicitly specified if no new input is received after 5 seconds or if an input has never been received.

Outputs are triggered based on system conditions specified in an AND/OR table. An example of an output of the PRS blackbox model is shown in Figure 10. The output command to close Latch Valve 1A is called when the triggering condition specified in the AND/OR table in Figure 10 evaluates to true. In this case, the command is sent when the Stage 1 fault protection algorithm is enabled, a Stage 1 overpressurization event has been detected, and the latch valve is not already closed. The message contents indicate the value that the output will take.

The complete Level 3 including all other components PRS blackbox model are not presented in the body of this thesis but can be found in the appendix.

LatchValve1AClose

Output Command

Destination: LatchValve1A

Message: LatchValveOutputMessage

Acceptable Values: Close

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Open

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 5 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: LatchValve1APosition

Values: Close

Relationship: Value position input should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Declare Valve as Stuck Opened

Reversed By: LatchValve1AOpen

Description: Command to Close Latch Valve 1A

Comments: NA

References: Stage1OverpressureFaultProtection, Stage1OverpressurizationFaultStatus, LatchValveOutputMessage, LatchValve1A, LatchValve1APosition, DP.1.2.2

TRIGGERING CONDITION

Stage1OverpressureFaultProtection in mode Enabled	T
Stage1OverpressurizationFaultStatus in state Overpressure	T
LatchValve1APosition is Close	F

MESSAGE CONTENTS

Field	Value
COM	Close

Figure 10 - SpecTRM-RL Output

3.5.2. Analysis

The power of having a formal model of the specification is that it enables engineers to run mathematical analyses on the specification. SpecTRM currently has two analysis tools available. The first checks for non-determinism, i.e., whether there is more than one transition possible for a state or mode value given a set of inputs. If there are non-deterministic state or mode values in the model, the design of the system is inconsistent and probably needs to be modified.

The second analysis, called robustness, checks that for a given set of inputs, there is at least one transition available for a state or mode value, that is, that the specification of the behavior is logically complete.

Using these analyses, the PRS model was revised until all non-deterministic cases were eliminated. The robustness criteria proved to be more challenging to satisfy. Robustness is a function of how much is required of the system. A model that has non-robust cases does not necessarily mean that the model will not satisfy all the requirements and constraints set on the system. The analysis is extremely conservative and forces the engineer to think through all possible cases in which a non-robust scenario might exist.

3.5.3. Simulation

Perhaps the biggest benefit of having a formal model is that it allows engineers to simulate the specified system behavior. The behavior of the system may be viewed and tested at an early stage in the design process, before any code is written. Multiple formal specifications may be linked and simulated using SpecTRM. An entire formal spacecraft subsystem specification such as the ADCS, which is created by linking specifications of individual components such as a PRS, RCS, IMU, and Star Tracker, may also be simulated using the SpecTRM toolset.

Simulations of the specification can be extremely useful as designs are often changed in the early phases of the development process. By combining various generic components, the simulated interactions between components can be analyzed as the design evolves. In the context of fault protection, because each generic component specification includes component level fault protection, the subsystem fault protection software can be derived from the combination of the various components. SpecTRM also allows users to inject faults from various external components during simulation to determine the response of the specified fault protection design.

The component-based spacecraft architecture described in this thesis requires that the complete spacecraft specifications be an aggregation of individual component and subsystem-level specifications. In order to perform a spacecraft-level simulation therefore requires the simulation tool to be able to link multiple component and subsystem specifications together to form a complete specification simulation environment. Currently, SpecTRM supports a two-layer simulation environment in which one specification acts as a master to multiple slave models. Using a generic ADCS subsystem specification and the modified PRS specification, a subsystem-level simulation was performed. Below is a representation of the simulation environment provided by SpecTRM.

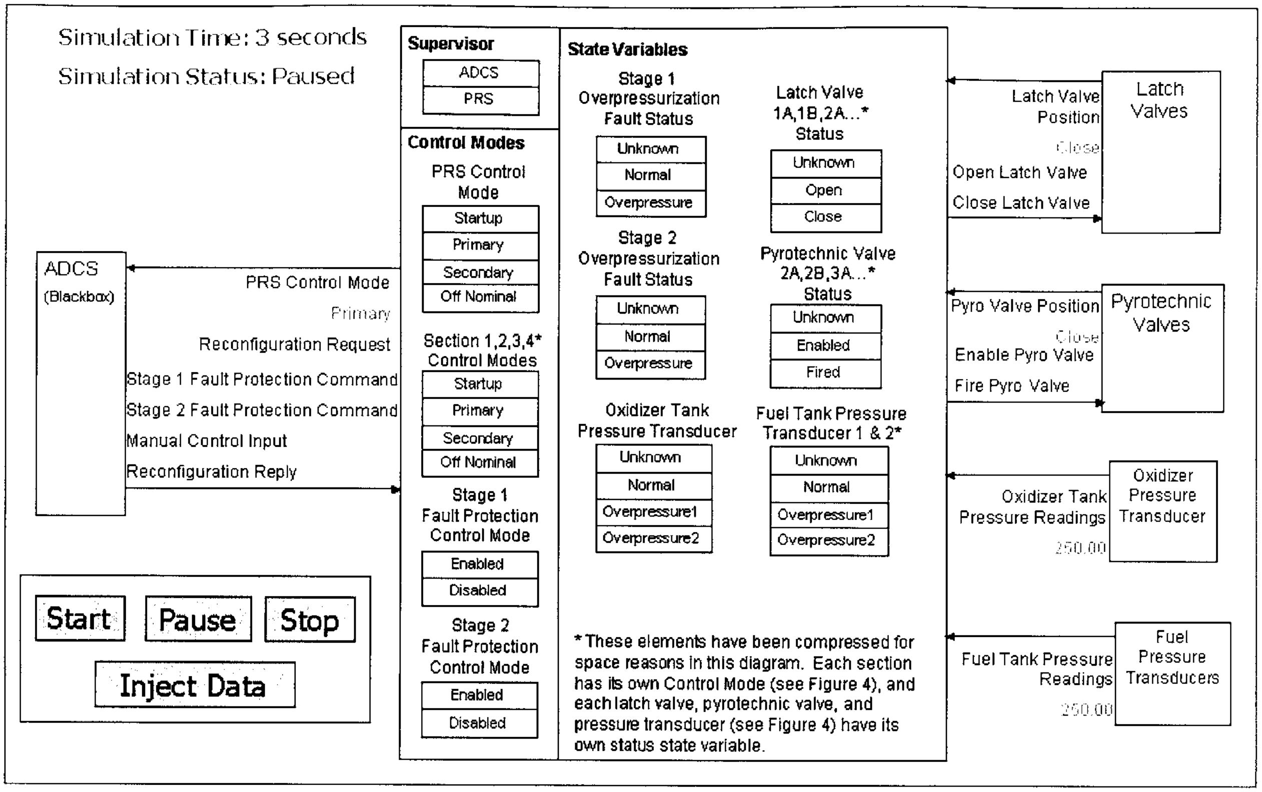


Figure 11 - SpecTRM Simulation Environment

On the top left corner of Figure 11 is the simulation status. Information here represents the current simulation time and the status of the simulation, which can be either started, stopped, or paused. On the lower left section of Figure 11 is the simulation control box. From here, the simulation may be started, paused, or stopped. In addition, during anytime in the simulation, data for any input or output value may be injected into the simulation by clicking on the Inject Data button. Within the system boundary, the current value of control modes and state variable are highlighted in yellow. The values of inputs to the system are shown below the name of each input and are highlighted in blue. If an output is triggered, the value of the output will also be shown below the name of the output name. The values of outputs are recorded in a text file format and can be analyzed following the completion of the simulation. By analyzing the inputs and outputs of the system simulation, engineers can determine whether the specification behavior has satisfied the requirements and constraints. For example, if inputs from two out of three pressure transducers are injected with data exceeding the Stage 1 overpressurization limit for a period of greater than 5 seconds, the expected response should be an output command sent to the latch valve 1A to close.

4. Discussion

4.1. Comparison of Architectures

There is a fundamental difference between the implementation of Cassini's fault protection software using the component-based specification reuse framework outlined in this paper and Cassini's actual fault protection software. In Cassini, the fault protection software is implemented as an independent entity, separate from the spacecraft's individual subsystem software. Our implementation, in contrast, encapsulates fault protection software within each subsystem specification. There are distinct advantages and disadvantages associated with each architecture. A formal comparison of each architecture is beyond the scope of this thesis. However, some general observations can be drawn.

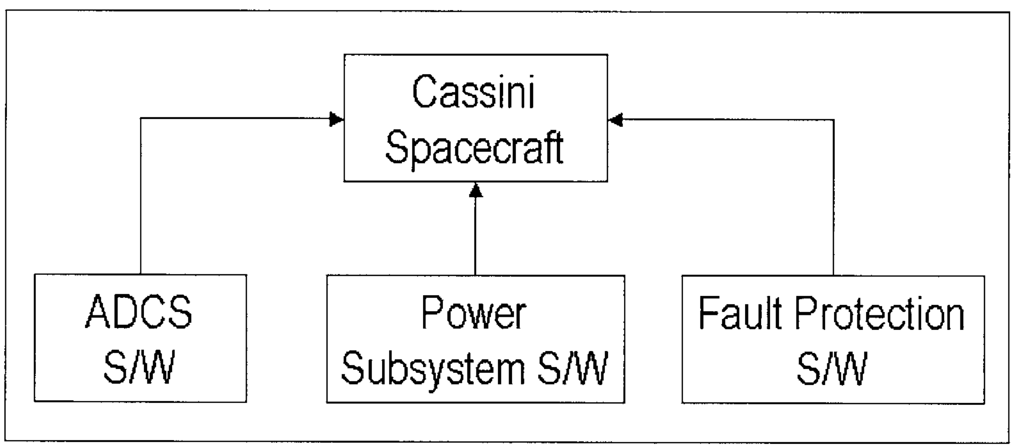


Figure 12 - Traditional Spacecraft Software Architecture

Figure 12 shows a simplified functional block diagram of a traditional spacecraft software architecture. Each individual subsystem is controlled by its own subsystem-level software. The software has direct access to the components on the spacecraft belonging to that subsystem. Fault protection functions are encapsulated in a separate and distinct software component. When a fault is detected, the fault protection software overrides the subsystem software, responds to the fault(s) detected, and then returns the spacecraft to normal operation or places the spacecraft is a safe state and waits for commands from the ground station.

In contrast, the component-based spacecraft architecture described in this thesis encapsulates fault protection in each subsystem-level software component as shown below in Figure 13.

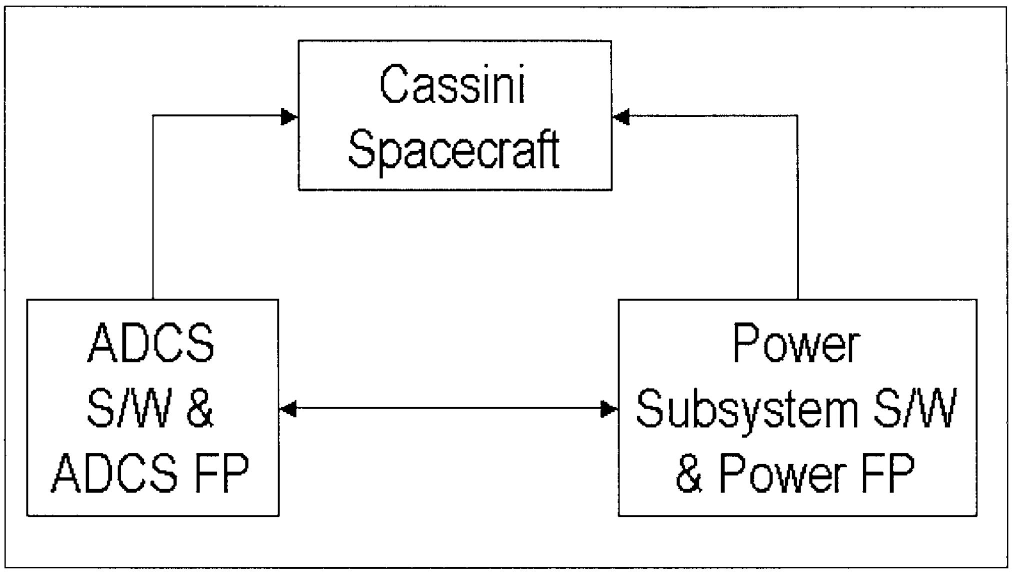


Figure 13 - Component-Based Spacecraft Architecture

During normal operation, this architecture performs as the traditional architecture described above. However, when a fault is detected, there is no longer a distinct controller that has priority over all other components. Each subsystem controller must therefore communicate with each and every other subsystem controller. An alternative architecture is described below in Figure 14.

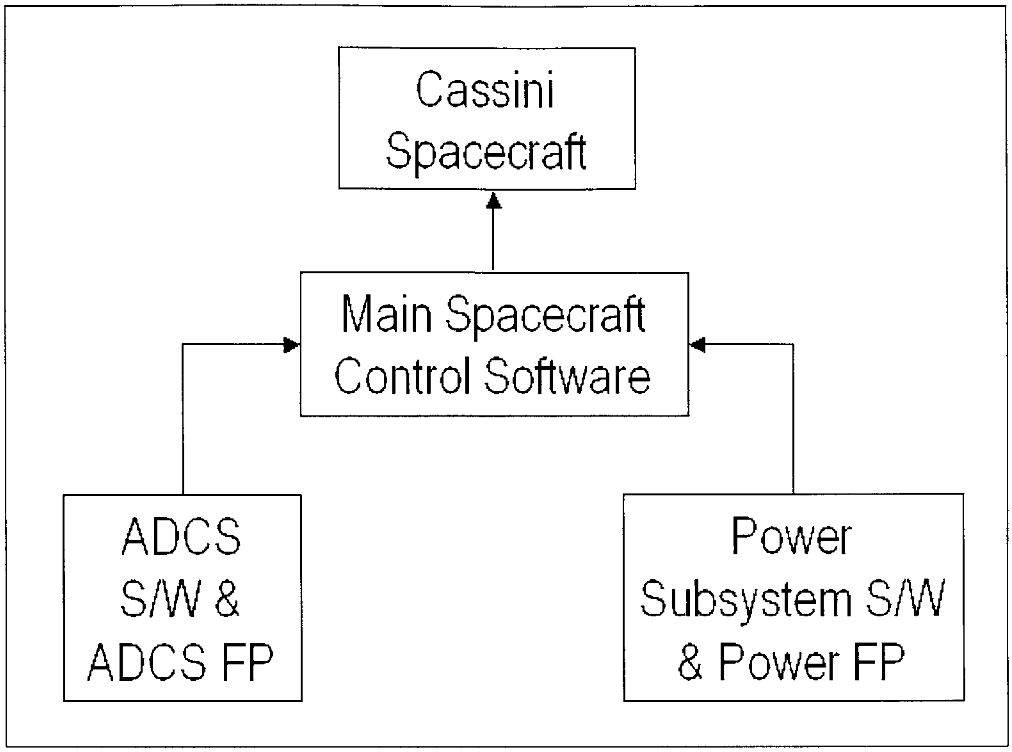


Figure 14 - Alternative Component-Based Spacecraft Architecture

In the architecture described in the figure above, a main spacecraft control component is introduced. This software component interfaces with each subsystem, assigns priorities for faults detected by each subsystem, and ensures that the fault responses are carried out based on the level of severity of each fault detected.

4.2. Possible Issues

4.2.1. Modification of Generic Specifications

The success of a generic domain-specific specification architecture relies on two qualities: the first is the ease with which each generic specification may be modified for a specific design. Starting from a generic PRS specification, it took one graduate student about a day to modify the generic specification to match Cassini's fault protection design. Although the initial generic PRS was based on a simplified version of Cassini's PRS, the time scale for creating the specification is still much less than the time it would have taken to build a specification from scratch. This issue could become more pronounced as spacecraft design evolves. The generic spacecraft component library created in this research project was intended to prove the applicability of component-based specification reuse. To use this framework effectively in actual spacecraft design will require the creation of many more generic component specifications. In addition, the library should be organizational-specific, created and modified by domain experts. Using the Cassini PRS, the research reported in this thesis has demonstrated that spacecraft design can be achieved by composing a specification from generic component specifications.

4.2.2. Scalability

The second challenge is scalability. So far, the component specifications and fault protection algorithms included are all from the same subsystem, but in a real spacecraft, much of the fault protection software is spacecraft-wide. A failure in the PRS affects the ADCS, for example, but in order to reconfigure the system, the spacecraft must ensure that there is enough power to carry out the reconfiguration procedures. For a typical spacecraft, this entails shutting down all nonessential components. An extension of the spacecraft architecture will be completed to evaluate the feasibility of applying this approach to the interactions between subsystems, such as the ADCS and Power subsystems.

4.3. Summary

In this thesis, a new spacecraft specification method has been described. By creating a library of generic spacecraft component specifications, a new spacecraft specification may be created by combining component specifications. The main advantage of using this new method is the ability to reuse domain knowledge captured in each generic component specification. In addition, by reusing spacecraft design at the specification-level, the issues related to reusing code is largely avoided. Each generic specification is written as an intent specification, which complements the component-based architecture. By using intent specifications, a model of each component may be formally analyzed. These analyses provide a way for engineers to view and analyze the system behavior before any costly code is implemented. SpecTRM provides a user-friendly and powerful environment from which multiple components may be easily combined and analyzed.

Using the component-based architecture, basic fault protection features are included in each component and subsystem specification. Fault protection becomes an integrated component of the spacecraft design, ensuring that requirements and constraints specific to fault protection is taken into account in the general spacecraft design process. In addition, from the hazard analysis integrated in intent specifications, fault protection features may be designed to mitigate defined hazards.

Using the component-based architecture concept, a small library of spacecraft component specifications was created. To prove the applicability of the concept, the Cassini PRS overpressurization fault protection software was implemented by combining a generic PRS specification and ADCS subsystem specification. The generic PRS specification, which was based on Cassini's PRS was easily modified to match Cassini's fault protection software. Using the analyses tools included in SpecTRM, non-determinism and robustness analyses were run on the modified Cassini PRS specification. A simulation was then performed to test that the specification behavior matched requirements and constraints set by Cassini's overpressurization fault protection specification.

Component-based spacecraft architecture differs from traditional spacecraft software architectures because fault protection is no longer a distinct software component. Future work would involve a formal analysis between the two architectures. More work will also be needed to investigate the reusability and scalability of the component-based architecture. While the PRS specification was easily modified, it was only one part of the Cassini fault protection software. Cassini's fault protection software includes 13 separate fault detection and response algorithms spanning across multiple subsystems. Many of the fault protection algorithms are also spacecraft-wide and not confined within an individual subsystem as the implementation carried out in this thesis. To formally analyze the component-based spacecraft architecture concept, a full range of components must be completed with components from multiple subsystems. By combining specifications from different subsystems, a more detailed analysis can be performed on the advantages and disadvantages of component-based spacecraft architecture.

Research Acknowledgement

This research in this paper was partially supported by NSF ITR Grant CCR-0085829, and by NASA Engineering for Complex Systems Program grant NAG2-1543.

References

[1] Brown, Charles D., *Spacecraft propulsion.* Washington, DC: American Institute of Aeronautics and Astronautics, c1996.

[2] Dvorak, Daniel, Rasmussen, Robert, Reeves, Glenn, Sacks, Al, "Software Architecture Themes in JPL'S Mission Data System." AIAA

[3] Euler, E.E., Jolly, S.D., and Curtis, H.H. "The Failures of the Mars Climate Orbiter and Mars Polar Lander: A Perspective from the People Involved". Guidance and Control 2001, American Astronautical Society, paper AAS 01-074, 2001.

[4] Jet Propulsion Laboratory, Cassini Orbiter Functional Requirements Book System Fault Protection Algorithms Rev A., Aug. 1997.

[5] Leveson, Nancy G., "Intent Specifications: An Approach to Building Human-Centered Specifications", IEEE Transactions on Software Engineering, Jan. 2000.

[6] Lions, J.L "Ariane 501 Failure: Report by the Inquiry Board". European Space Agency, 19 Jul. 1996.

[7] Rasmussen, Robert D. "Goal-Based Fault Tolerance for Space Systems Using the Mission Data System." IEEE, 2001.

[8] Safeware Corporation "SpecTRM User Manual". Safeware, 2002

[9] Sidi, Marcel J., *Spacecraft Dynamics and Control: A Practical Engineering Approach* Cambridge University Press, 1997.

[10] Weiss, K.A., ""Building a Reusable Spacecraft Architecture using Component-Based System Engineering", Masters Thesis, Aeronautics and Astronautics Dept., MIT, 2003.

Appendix

Pressure Regulation System Intent Specification

Pressure Regulation System

Pages 1-32 have been ommitted from the Archives copy. This is the most complete version available.

Outputs

LatchValve1AOpen

Output Command

Destination: LatchValve1A

Message: LatchValveOutputMessage

Acceptable Values: Open

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Close

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 5 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: LatchValve1APosition

Values: Open

Relationship: Value position input should transition to Open

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Declare Valve as Stuck Closed

Reversed By: LatchValve1AClose

Description: Command to Open LatchValve1A

Comments: NA

References: Section1ControlMode, SupervisorMode, ControlMode, LatchValveOutputMessage, LatchValve1A, LatchValve1APosition, □DP.1.2.2

TRIGGERING CONDITION

ManualControlInput is LatchValve1AOpen	T

MESSAGE CONTENTS

Field	Value
COM	Open

LatchValve1AClose

Output Command

Destination: LatchValve1A

Message: LatchValveOutputMessage

Acceptable Values: Close

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Open

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 5 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: LatchValve1APosition

Values: Close

Relationship: Value position input should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Declare Valve as Stuck Opened

Reversed By: LatchValve1AOpen

Description: Command to Close Latch Valve 1A

Comments: NA

References: Stage1OverpressureFaultProtection, Stage1OverpressurizationFaultStatus, LatchValveOutputMessage, LatchValve1A, LatchValve1APosition, DP.1.2.2

TRIGGERING CONDITION

ManualControlInput is LatchValve1AClose	T	*
SupervisorMode in mode PRS	F	T
Stage1OverpressurizationFaultStatus in state Overpressure	*	T
LatchValve1APosition is Close	*	F

MESSAGE CONTENTS

Field	Value
COM	Close

LatchValve1BOpen

Output Command

Destination: LatchValve1B

Message: LatchValveOutputMessage

Acceptable Values: Open

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Close

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 5 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: LatchValve1BPosition

Values: Open

Relationship: Value position input should transition to Open

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Declare Valve as Stuck Closed

Reversed By: LatchValve1BClose

Description: Command to Open LatchValve1B

Comments: NA

References: Section1ControlMode, SupervisorMode, ControlMode, LatchValveOutputMessage, LatchValve1A, LatchValve1APosition, ❑DP.1.2.2

TRIGGERING CONDITION

ManualControlInput is LatchValve1BOpen	T

MESSAGE CONTENTS

Field	Value
COM	Open

LatchValve1BClose

Output Command

Destination: LatchValve1B

Message: LatchValveOutputMessage

Acceptable Values: Close

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Open

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 5 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: LatchValve1BPosition

Values: Close

Relationship: Value position input should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Declare Valve as Stuck Opened

Reversed By: LatchValve1BOpen

Description: Command to Close Latch Valve 1B

Comments: NA

References: Stage1OverpressureFaultProtection, Stage1OverpressurizationFaultStatus, LatchValveOutputMessage, LatchValve1A, LatchValve1APosition, ❑DP.1.2.2

TRIGGERING CONDITION

ManualControlInput is LatchValve1BClose	T

MESSAGE CONTENTS

Field	Value
COM	Close

PyroValve1AEnable

Output Command

Destination: PyroValve1A

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve1ADisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

SupervisorMode in mode PRS	T	F
Stage2OverpressurizationFaultStatus in state Overpressure	T	*
ManualControlInput is PyroValve1AEnable	*	T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve1AFire

Output Command

Destination: PyroValve1A

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve1APosition

Values: Close

Relationship: PyroValve1APosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

SupervisorMode in mode PRS	T	F
Stage2OverpressurizationFaultStatus in state Overpressure	T	*
Time Since Message for PyroValve1AEnable was Last Sent > 90 seconds	T	*
ManualControlInput is PyroValve1AFire	*	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve1BEnable

Output Command

Destination: PyroValve1B

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve1BDisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve1BEnable	T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve1BFire

Output Command

Destination: PyroValve1B

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve1BPosition

Values: Close

Relationship: PyroValve1BPosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve1BFire	T
Time Since Message for PyroValve1BEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve2AEnable

Output Command

Destination: PyroValve2A

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve2ADisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve2AEnable		T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve2AFire

Output Command

Destination: PyroValve2A

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve2APosition

Values: Close

Relationship: PyroValve2APosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve2AFire	T
Time Since Message for PyroValve2AEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve2BEnable

Output Command

Destination: PyroValve2B

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve2BDisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve2BEnable	T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve2BFire

Output Command

Destination: PyroValve2B

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve2BPosition

Values: Close

Relationship: PyroValve2BPosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve2BFire	T
Time Since Message for PyroValve2BEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve3AEnable

Output Command

Destination: PyroValve3A

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve3ADisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition.

TRIGGERING CONDITION

ManualControlInput is PyroValve3AEnable

T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve3AFire

Output Command

Destination: PyroValve3A

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve3APosition

Values: Close

Relationship: PyroValve3APosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve3AFire	T
Time Since Message for PyroValve3AEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve3BEnable

Output Command

Destination: PyroValve3B

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve3BDisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve3BEnable		T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve3BFire

Output Command

Destination: PyroValve3B

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve3BPosition

Values: Close

Relationship: PyroValve3BPosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve3BFire	T
Time Since Message for PyroValve3BEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve4AEnable

Output Command

Destination: PyroValve4A

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve4ADisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve4AEnable		T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve4AFire

Output Command

Destination: PyroValve4A

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve4APosition

Values: Close

Relationship: PyroValve4APosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve4AFire	T
Time Since Message for PyroValve4AEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

PyroValve4BEnable

Output Command

Destination: PyroValve4B

Message: PyroValveEnableMessage

Acceptable Values: Enable

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Fire

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: 90 seconds

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: Not Available

Variables: NA

Values: NA

Relationship: NA

Min. time (latency): NA

Max time: NA

Exception-Handling: NA

Reversed By: PyroValve4BDisable

Description: Command to Enable Pyrotechnic Device.

Comments: NA

References: SupervisorMode, LatchValve1AStatus, PyroValveEnableMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve4BEnable | T

MESSAGE CONTENTS

Field	Value
COM	Enable

PyroValve4BFire

Output Command

Destination: PyroValve4B

Message: PyroValveFireMessage

Acceptable Values: Fire

Units: NA

Granularity: NA

Exception-Handling: Not Available

Hazardous Values: Not Available

Timing Behavior: See Below

Initiation Delay: 0 seconds

Completion Deadline: NA

Output Capacity Assumptions: Not Available

Load: Not Available

Min time between outputs: 0 seconds

Max time between outputs: NA

Hazardous Timing Behavior: Not Available

Exception-Handling: Not Available

Feedback Information: See Below

Variables: PyroValve4BPosition

Values: Close

Relationship: PyroValve4BPosition should transition to Close

Min. time (latency): 0 seconds

Max time: 5 seconds

Exception-Handling: Not Available

Reversed By: NA

Description: Command to Fire Enabled Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, SupervisorMode, PyroValveFireMessage, PyroValve1A, PyroValve1APosition

TRIGGERING CONDITION

ManualControlInput is PyroValve4BFire	T
Time Since Message for PyroValve4BEnable was Last Sent > 90 seconds	T

MESSAGE CONTENTS

Field	Value
COM	Fire

ControlModePrimaryOutput

Display Output

Destination: ADCS

Message: ControlModeOutputMessage

Acceptable Values:

Units:

Granularity:

Hazardous Values:

Update Requirements:

Update Delay:

Update Completion Deadline:

Output Capacity Assumptions:

Update Load:

Min update rate:

Max update rate:

Deletion Requirements (including data age):

Hazardous timing behavior:

Exception-Handling:

Failure Indication:

Reversed By:

Description:

Comments:

References: ControlMode, ControlModeOutputMessage, ADCS

TRIGGERING CONDITION

Time Since ControlMode Entered Primary <= 1 seconds	T

MESSAGE CONTENTS

Field	Value
Mode	PRIMARY

ControlModeSecondaryOutput

Display Output

Destination: ADCS

Message: ControlModeOutputMessage

Acceptable Values:

Units:

Granularity:

Hazardous Values:

Update Requirements:

Update Delay:

Update Completion Deadline:

Output Capacity Assumptions:

Update Load:

Min update rate:

Max update rate:

Deletion Requirements (including data age):

Hazardous timing behavior:

Exception-Handling:

Failure Indication:

Reversed By:

Description:

Comments:

References: ControlMode, ControlModeOutputMessage, ADCS

TRIGGERING CONDITION

Time Since ControlMode Entered Secondary <= 1 seconds	T

MESSAGE CONTENTS

Field	Value
Mode	SECONDARY

ControlModeOffNominalOutput

Display Output

Destination: ADCS

Message: ControlModeOutputMessage

Acceptable Values:

Units:

Granularity:

Hazardous Values:

Update Requirements:

Update Delay:

Update Completion Deadline:

Output Capacity Assumptions:

Update Load:

Min update rate:

Max update rate:

Deletion Requirements (including data age):

Hazardous timing behavior:

Exception-Handling:

Failure Indication:

Reversed By:

Description:

Comments:

References: ControlMode, ControlModeOutputMessage, ADCS

TRIGGERING CONDITION

Time Since ControlMode Entered OffNominal <= 1 seconds	T

MESSAGE CONTENTS

Field	Value
Mode	OFFNOMINAL

ReconfiguringOutput

Display Output

Destination: ADCS

Message: ReconfiguringOutputMessage

Acceptable Values: {ReconfiguringPRS}

Units: NA

Granularity: ??

Hazardous Values: ??

Update Requirements:

Update Delay:

Update Completion Deadline:

Output Capacity Assumptions:

Update Load:

Min update rate:

Max update rate:

Deletion Requirements (including data age):

Hazardous timing behavior:

Exception-Handling:

Failure Indication:

Reversed By:

Description:

Comments:

References: SupervisorMode, Stage2OverpressurizationFaultStatus, Stage2OverpressurizationResponse, Stage1OverpressurizationResponse, Stage1OverpressurizationFaultStatus, ReconfiguringOutputMessage, ADCS

TRIGGERING CONDITION

SupervisorMode in mode PRS	T	T
Stage1OverpressurizationFaultStatus in state Overpressure	T	*
Stage1OverpressurizationResponse in state ResponseComplete	F	*
Stage2OverpressurizationFaultStatus in state Overpressure	*	T
Stage2OverpressurizationResponse in state ResponseComplete	*	F

MESSAGE CONTENTS

Field	Value
Reconfiguring	ReconfiguringPRS

Modes

ControlMode

Control Mode

Description: This is the main control mode of the PRS

Comments: NA

References: Section1ControlMode, Section2ControlMode, Section4ControlMode, Section3ControlMode, ❑SI. 1.3

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose

DEFINITION

= Primary

Section1ControlMode in mode Primary	T
Section2ControlMode in mode Primary	T
Section3ControlMode in mode Primary	T
Section4ControlMode in mode Primary	T

= Secondary

Section1ControlMode in mode Secondary	T	*	*	*
Section2ControlMode in mode Secondary	*	T	*	*
Section3ControlMode in mode Secondary	*	*	T	*
Section4ControlMode in mode Secondary	*	*	*	T
Section1ControlMode in mode OffNominal	F	F	F	F
Section2ControlMode in mode OffNominal	F	F	F	F
Section3ControlMode in mode OffNominal	F	F	F	F
Section4ControlMode in mode OffNominal	F	F	F	F

= OffNominal

Section1ControlMode in mode OffNominal	T	*	*	*
Section2ControlMode in mode OffNominal	*	T	*	*
Section3ControlMode in mode OffNominal	*	*	T	*
Section4ControlMode in mode OffNominal	*	*	*	T

Stage1OverpressureFaultProtection

Control Mode

Description: Determines Stage 1 Overpressurization Fault Protection Enabled or Disabled

Comments: NA

References: Stage1OverpressurizationResponse, Stage1FaultProtectionCommandInput

Appears In: LatchValve1AClose

DEFINITION

= Enabled

Power Up	T	F
Stage1FaultProtectionCommandInput is Enable	*	T
Stage1OverpressurizationResponse has Never Entered ResponseComplete	*	T

= Disabled

Power Up	F	F
Stage1FaultProtectionCommandInput is Enable	T	F
Stage1OverpressurizationResponse has Never Entered ResponseComplete	F	*
Stage1FaultProtectionCommandInput is Disable	F	T

Stage2OverpressureFaultProtection

Control Mode

Description: Determines Stage 2 Overpressurization Fault Protection Enabled or Disabled

Comments: NA

References: Stage2OverpressurizationResponse, Stage2FaultProtectionCommandInput

Appears In: LatchValve1AClose

DEFINITION

= Enabled

Power Up	T	F
Stage2FaultProtectionCommandInput is Enable	*	T
Stage2OverpressurizationResponse has Never Entered ResponseComplete	*	T

= Disabled

Power Up	F	F
Stage2FaultProtectionCommandInput is Enable	T	F
Stage2OverpressurizationResponse has Never Entered ResponseComplete	F	*
Stage2FaultProtectionCommandInput is Disable	F	T

Section1ControlMode

Control Mode

Description: Section 1 Control Mode

Comments: NA

References: PyroValve1AStatus, PyroValve1BStatus, PyroValve1APosition, PyroValve1BPosition

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose, ControlMode

DEFINITION

= Startup

Power Up	T

= Primary

Power Up	F
PyroValve1APosition is Open	T
PyroValve1APosition is Close	F
PyroValve1AStatus has Never Entered Fired	T
PyroValve1BPosition is Close	T
PyroValve1BPosition is Open	F
PyroValve1BStatus has Never Entered Fired	T

= Secondary

Power Up	F
PyroValve1APosition is Open	F
PyroValve1APosition is Close	T
PyroValve1AStatus has Never Entered Fired	F
PyroValve1BPosition is Close	F
PyroValve1BPosition is Open	T
PyroValve1BStatus has Never Entered Fired	F

= OffNominal

Power Up	F	F	F	F
PyroValve1APosition is Open	T	*	F	*
PyroValve1APosition is Close	F	*	F	*
PyroValve1AStatus has Never Entered Fired	F	*	*	*
PyroValve1BPosition is Close	*	T	*	F
PyroValve1BPosition is Open	*	F	*	F
PyroValve1BStatus has Never Entered Fired	*	F	*	*

Section2ControlMode

Control Mode

Description: Section 2 Control Mode

Comments: NA

References: PyroValve2AStatus, PyroValve2BStatus, PyroValve2BPosition, PyroValve2APosition

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose, ControlMode

DEFINITION

= Startup

Power Up	T

= Primary

Power Up	F
PyroValve2APosition is Open	T
PyroValve2APosition is Close	F
PyroValve2AStatus has Never Entered Fired	T
PyroValve2BPosition is Close	T
PyroValve2BPosition is Open	F
PyroValve2BStatus has Never Entered Fired	T

= Secondary

Power Up	F
PyroValve2APosition is Open	F
PyroValve2APosition is Close	T
PyroValve2AStatus has Never Entered Fired	F
PyroValve2BPosition is Close	F
PyroValve2BPosition is Open	T
PyroValve2BStatus has Never Entered Fired	F

= OffNominal

Power Up	F	F	F	F
PyroValve2APosition is Open	T	*	F	*
PyroValve2APosition is Close	F	*	F	*
PyroValve2AStatus has Never Entered Fired	F	*	*	*
PyroValve2BPosition is Close	*	T	*	F
PyroValve2BPosition is Open	*	F	*	F
PyroValve2BStatus has Never Entered Fired	*	F	*	*

Section3ControlMode

Control Mode

Description: Section 3 Control Mode

Comments: NA

References: PyroValve3AStatus, PyroValve3BStatus, PyroValve3APosition, PyroValve3BPosition

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose, ControlMode

DEFINITION

= Startup

Power Up	T

= Primary

Power Up	F
PyroValve3APosition is Open	T
PyroValve3APosition is Close	F
PyroValve3AStatus has Never Entered Fired	T
PyroValve3BPosition is Close	T
PyroValve3BPosition is Open	F
PyroValve3BStatus has Never Entered Fired	T

= Secondary

Power Up	F
PyroValve3APosition is Open	F
PyroValve3APosition is Close	T
PyroValve3AStatus has Never Entered Fired	F
PyroValve3BPosition is Close	F
PyroValve3BPosition is Open	T
PyroValve3BStatus has Never Entered Fired	F

= OffNominal

Power Up	F	F	F	F
PyroValve3APosition is Open	T	*	F	*
PyroValve3APosition is Close	F	*	F	*
PyroValve3AStatus has Never Entered Fired	F	*	*	*
PyroValve3BPosition is Close	*	T	*	F
PyroValve3BPosition is Open	*	F	*	F
PyroValve3BStatus has Never Entered Fired	*	F	*	*

Section4ControlMode

Control Mode

Description: Section 4 Control Mode

Comments: NA

References: PyroValve4BStatus, PyroValve4AStatus, PyroValve4BPosition, PyroValve4APosition

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose, ControlMode

DEFINITION

= Startup

Power Up	T

= Primary

Power Up	F
PyroValve4APosition is Open	T
PyroValve4APosition is Close	F
PyroValve4AStatus has Never Entered Fired	T
PyroValve4BPosition is Close	T
PyroValve4BPosition is Open	F
PyroValve4BStatus has Never Entered Fired	T

= Secondary

Power Up	F
PyroValve4APosition is Open	F
PyroValve4APosition is Close	T
PyroValve4AStatus has Never Entered Fired	F
PyroValve4BPosition is Close	F
PyroValve4BPosition is Open	T
PyroValve4BStatus has Never Entered Fired	F

= OffNominal

Power Up	F	F	F	F
PyroValve4APosition is Open	T	*	F	*
PyroValve4APosition is Close	F	*	F	*
PyroValve4AStatus has Never Entered Fired	F	*	*	*
PyroValve4BPosition is Close	*	T	*	F
PyroValve4BPosition is Open	*	F	*	F
PyroValve4BStatus has Never Entered Fired	*	F	*	*

SupervisorMode

Supervisory Mode

Description: Determines the current Supervisor of the PRS

Comment: NA

References: Stage2FaultProtectionCommandInput, Stage1FaultProtectionCommandInput

Appears In: PyroValve4BEnable, PyroValve1BFire, ReconfiguringOutput, PyroValve3BEnable, PyroValve1BEnable, PyroValve2BFire, PyroValve1AEnable, LatchValve1AOpen, PyroValve1AFire, PyroValve3BFire, PyroValve3AFire, PyroValve4BFire, PyroValve2BEnable, PyroValve3AEnable, PyroValve2AEnable, LatchValve1BOpen, PyroValve2AFire, PyroValve4AEnable, LatchValve1BClose, PyroValve4AFire

DEFINITION

= PRS

Power Up	T	F	F
Stage1FaultProtectionCommandInput is Disable	*	F	*
Stage2FaultProtectionCommandInput is Disable	*	*	F

= ADCS

Power Up	F
Stage1FaultProtectionCommandInput is Disable	T
Stage2FaultProtectionCommandInput is Disable	T

State Values

LatchValve1ACommand

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the last command sent to the Latch Valve

Comments: NA

References: LatchValve1AOpen, LatchValve1AClose

Appears In: LatchValve1AStatus

DEFINITION

= Unknown

Power Up	T	*	F
Message for LatchValve1AOpen was Sent	*	F	T
Message for LatchValve1AClose was Sent	*	F	T

= Open

Power Up	F
Message for LatchValve1AOpen was Sent	T
Message for LatchValve1AClose was Sent	F

= Close

Power Up	F
Message for LatchValve1AOpen was Sent	F
Message for LatchValve1AClose was Sent	T

LatchValve1AStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: LatchValve1APosition

Description: Determines the Status of the Latch Valve

Comments: NA

References: LatchValve1ACommand, LatchValve1APosition

Appears In: PyroValve1AEnable

DEFINITION

= Unknown

Power Up	T	*
LatchValve1APosition is Obsolete	*	T

= Open

Power Up	F
LatchValve1APosition is Open	T
LatchValve1ACommand in state Unknown	T

= Close

Power Up	F
LatchValve1APosition is Close	T
LatchValve1ACommand in state Unknown	T

= StuckOpen

Power Up	F
LatchValve1APosition is Open	T
Time Since LatchValve1ACommand Entered Close > 5 seconds	T
LatchValve1ACommand in state Close	T

= StuckClose

Power Up	F
LatchValve1APosition is Close	T
Time Since LatchValve1ACommand Entered Open > 5 seconds	T
LatchValve1ACommand in state Open	T

LatchValve1BCommand

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the last command sent to the Latch Valve

Comments: NA

References: LatchValve1BOpen, LatchValve1BClose

Appears In: LatchValve1BStatus

DEFINITION

= Unknown

Power Up	T	*	F
Message for LatchValve1BOpen was Sent	*	F	T
Message for LatchValve1BClose was Sent	*	F	T

= Open

Power Up	F
Message for LatchValve1BOpen was Sent	T
Message for LatchValve1BClose was Sent	F

= Close

Power Up	F
Message for LatchValve1BOpen was Sent	F
Message for LatchValve1BClose was Sent	T

LatchValve1BStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: LatchValve1BPosition

Description: Determines the status of the Latch Valve

Comments: NA

References: LatchValve1BCommand, LatchValve1BPosition

Appears In: PyroValve1BEnable

DEFINITION

= Unknown

Power Up	T	*
LatchValve1BPosition is Obsolete	*	T

= Open

Power Up	F
LatchValve1BPosition is Open	T
LatchValve1BCommand in state Unknown	T

= Close

Power Up	F
LatchValve1BPosition is Close	T
LatchValve1BCommand in state Unknown	T

= StuckOpen

Power Up	F
LatchValve1BPosition is Open	T
Time Since LatchValve1BCommand Entered Close > 5 seconds	T
LatchValve1BCommand in state Close	T

= StuckClose

Power Up	F
LatchValve1BPosition is Close	T
Time Since LatchValve1BCommand Entered Open > 5 seconds	T
LatchValve1BCommand in state Open	T

PyroValve1AStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the Status of teh Pyrotechnic Valve

Comments: NA

References: PyroValve1AEnable, PyroValve1AFire

Appears In: Section1ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve1AEnable was Sent	*	F
Message for PyroValve1AFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve1AEnable was Sent	T
Message for PyroValve1AFire was Sent	F

= Fired

Power Up	F
Message for PyroValve1AFire was Sent	T

PyroValve1BStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the statusu of the Pyrotechnic Valve

Comments: NA

References: PyroValve1BFire, PyroValve1BEnable

Appears In: Section1ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve1BEnable was Sent	*	F
Message for PyroValve1BFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve1BEnable was Sent	T
Message for PyroValve1BFire was Sent	F

= Fired

Power Up	F
Message for PyroValve1BFire was Sent	T

PyroValve2AStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve2AEnable, PyroValve2AFire

Appears In: Section2ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve2AEnable was Sent	*	F
Message for PyroValve2AFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve2AEnable was Sent	T
Message for PyroValve2AFire was Sent	F

= Fired

Power Up	F
Message for PyroValve2AFire was Sent	T

PyroValve2BStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve2BEnable, PyroValve2BFire

Appears In: Section2ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve2BEnable was Sent	*	F
Message for PyroValve2BFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve2BEnable was Sent	T
Message for PyroValve2BFire was Sent	F

= Fired

Power Up	F
Message for PyroValve2BFire was Sent	T

PyroValve3AStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve3AEnable, PyroValve3AFire

Appears In: Section3ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve3AEnable was Sent	*	F
Message for PyroValve3AFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve3AEnable was Sent	T
Message for PyroValve3AFire was Sent	F

= Fired

Power Up	F
Message for PyroValve3AFire was Sent	T

PyroValve3BStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve3BEnable, PyroValve3BFire

Appears In: Section3ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve3BEnable was Sent	*	F
Message for PyroValve3BFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve3BEnable was Sent	T
Message for PyroValve3BFire was Sent	F

= Fired

Power Up	F
Message for PyroValve3BFire was Sent	T

PyroValve4AStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve4AEnable, PyroValve4AFire

Appears In: Section4ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve4AEnable was Sent	*	F
Message for PyroValve4AFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve4AEnable was Sent	T
Message for PyroValve4AFire was Sent	F

= Fired

Power Up	F
Message for PyroValve4AFire was Sent	T

PyroValve4BStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Pyrotechnic Valve

Comments: NA

References: PyroValve4BEnable, PyroValve4BFire

Appears In: Section4ControlMode

DEFINITION

= Unknown

Power Up	T	*
Message for PyroValve4BEnable was Sent	*	F
Message for PyroValve4BFire was Sent	*	F

= Enabled

Power Up	F
Message for PyroValve4BEnable was Sent	T
Message for PyroValve4BFire was Sent	F

= Fired

Power Up	F
Message for PyroValve4BFire was Sent	T

OxidizerPressureTransducerStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Oxidizer Pressure Transducer

Comments: NA

References: OxidizerStage2Check, OxidizerAObsoleteCheck, OxidizerBObsoleteCheck, OxidizerStage1Check

Appears In: Stage1OverpressureCheck, Stage2OverpressureCheck

DEFINITION

= Unknown

Power Up	T	F
OxidizerAObsoleteCheck	*	T
OxidizerBObsoleteCheck	*	T

= Normal

Power Up	F	F	F
OxidizerAObsoleteCheck	F	T	F
OxidizerBObsoleteCheck	F	F	T
OxidizerStage1Check	F	F	F
OxidizerStage2Check	F	F	F

= OverPressure1

Power Up	F	F	F
OxidizerAObsoleteCheck	F	T	F
OxidizerBObsoleteCheck	F	F	T
OxidizerStage1Check	T	T	T
OxidizerStage2Check	F	F	F

= OverPressure2

Power Up	F	F	F
OxidizerAObsoleteCheck	F	T	F
OxidizerBObsoleteCheck	F	F	T
OxidizerStage1Check	F	F	F
OxidizerStage2Check	T	T	T

FuelPressureTransducer1Status

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Fuel Pressure Transducer

Comments: NA

References: Fuel1Stage1Check, Fuel1Stage2Check, Fuel1BObsoleteCheck, Fuel1AObsoleteCheck

Appears In: Stage1OverpressureCheck, Stage2OverpressureCheck

DEFINITION

= Unknown

Power Up	T	F
Fuel1AObsoleteCheck	*	T
Fuel1BObsoleteCheck	*	T

= Normal

Power Up	F	F	F
Fuel1AObsoleteCheck	F	T	F
Fuel1BObsoleteCheck	F	F	T
Fuel1Stage1Check	F	F	F
Fuel1Stage2Check	F	F	F

= OverPressure1

Power Up	F	F	F
<u>Fuel1AObsoleteCheck</u>	F	T	F
<u>Fuel1BObsoleteCheck</u>	F	F	T
<u>Fuel1Stage1Check</u>	T	T	T
<u>Fuel1Stage2Check</u>	F	F	F

= OverPressure2

Power Up	F	F	F
<u>Fuel1AObsoleteCheck</u>	F	T	F
<u>Fuel1BObsoleteCheck</u>	F	F	T
<u>Fuel1Stage1Check</u>	F	F	F
<u>Fuel1Stage2Check</u>	T	T	T

FuelPressureTransducer2Status

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: Determines the status of the Fuel Pressure Transducer

Comments: NA

References: Fuel2BObsoleteCheck, Fuel2Stage2Check, Fuel2Stage1Check, Fuel2AObsoleteCheck

Appears In: Stage1OverpressureCheck, Stage2OverpressureCheck

DEFINITION

= Unknown

Power Up	T	F
Fuel2AObsoleteCheck	*	T
Fuel2BObsoleteCheck	*	T

= Normal

Power Up	F	F	F
Fuel2AObsoleteCheck	F	T	F
Fuel2BObsoleteCheck	F	F	T
Fuel2Stage1Check	F	F	F
Fuel2Stage2Check	F	F	F

= OverPressure1

Power Up	F	F	F
Fuel2AObsoleteCheck	F	T	F
Fuel2BObsoleteCheck	F	F	T
Fuel2Stage1Check	T	T	T
Fuel2Stage2Check	F	F	F

= OverPressure2

Power Up	F	F	F
Fuel2AObsoleteCheck	F	T	F
Fuel2BObsoleteCheck	F	F	T
Fuel2Stage1Check	F	F	F
Fuel2Stage2Check	T	T	T

Stage1OverpressurizationFaultStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: The Stage1OverpresurizationFaultStatus detects a Stage 1 overpressurization event.

Comments: NA

References: Stage1OverpressureCheck, ❑DP.6

Appears In: ReconfiguringOutput, LatchValve1AClose, Stage1OverpressurizationResponse

DEFINITION

= Unknown

Power Up	T

= Normal

Power Up	F
Stage1OverpressureCheck	F

= Overpressure

Power Up	F
Stage1OverpressureCheck	T

Stage1OverpressurizationResponse

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: LatchValve1APosition

Description: Determines the status of the Response

Comments: NA

References: Stage1OverpressurizationFaultStatus, LatchValve1APosition, ❑DP 7

Appears In: ReconfiguringOutput, Stage1OverpressureFaultProtection

DEFINITION

= Unknown

Power Up	T	*
Stage1OverpressurizationFaultStatus in state Unknown	*	T

= Normal

Power Up	F
Stage1OverpressurizationFaultStatus in state Normal	T
LatchValve1APosition is Close	F

= Responding

Power Up	F
Stage1OverpressurizationFaultStatus in state Overpressure	T
Time Since Stage1OverpressurizationFaultStatus Entered Overpressure >= 5 seconds	T

= ResponseComplete

Power Up	F
LatchValve1APosition is Close	T
Stage1OverpressurizationFaultStatus has Never Entered Overpressure	F

Stage2OverpressurizationFaultStatus

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: NA

Description: The Stage2OverpresurizationFaultStatus detects a Stage 2 overpressurization event.

Comments: NA

References: Stage2OverpressureCheck, ❑DP.6

Appears In: ReconfiguringOutput, Stage2OverpressurizationResponse

DEFINITION

= Unknown

Power Up	T

= Normal

Power Up	F
Stage2OverpressureCheck	F

= Overpressure

Power Up	F
Stage2OverpressureCheck	T

Stage2OverpressurizationResponse

State Value

Obsolescence: Not Available

Exception-Handling: Not Available

Related Inputs: PyroValve1APosition

Description: Determines the status of the response

Comments: NA

References: Stage2OverpressurizationFaultStatus, PyroValve1APosition, ❑DP.8

Appears In: ReconfiguringOutput, Stage2OverpressureFaultProtection

DEFINITION

= Unknown

Power Up	T	*
Stage2OverpressurizationFaultStatus in state Unknown	*	T

= Normal

Power Up	F
Stage2OverpressurizationFaultStatus in state Normal	T

= Responding

Power Up	F
Stage2OverpressurizationFaultStatus in state Overpressure	T
Time Since Stage2OverpressurizationFaultStatus Entered Overpressure >= 5 seconds	T

= ResponseComplete

Power Up	F
PyroValve1APosition is Close	T
Stage2OverpressurizationFaultStatus has Never Entered Overpressure	F

Macros and Functions

OxidizerAObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: OxidizerTankPressureA

Appears In: OxidizerPressureTransducerStatus

DEFINITION

OxidizerTankPressureA is Obsolete	T	*	*
OxidizerTankPressureA < 200	*	T	*
OxidizerTankPressureA > 400	*	*	T

OxidizerBObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: OxidizerTankPressureB

Appears In: OxidizerPressureTransducerStatus

DEFINITION

OxidizerTankPressureB is Obsolete	T	*	*
OxidizerTankPressureB < 200	*	T	*
OxidizerTankPressureB > 400	*	*	T

OxidizerStage1Check

Macro

Description: Checks input for Stage 1 Overpressurization Limit

Comments: NA

References: OxidizerTankPressureA, OxidizerTankPressureB

Appears In: OxidizerPressureTransducerStatus

DEFINITION

OxidizerTankPressureA >= 269	T	*
OxidizerTankPressureA < 300	T	*
OxidizerTankPressureB >= 269	*	T
OxidizerTankPressureB < 300	*	T

OxidizerStage2Check

Macro

Description: Checks input for Stage 2 Overpressurization Limit

Comments: NA

References: OxidizerTankPressureA, OxidizerTankPressureB

Appears In: OxidizerPressureTransducerStatus

DEFINITION

OxidizerTankPressureA >=300	T
OxidizerTankPressureA <= 400	T
OxidizerTankPressureB >=300	T
OxidizerTankPressureB <= 400	T

Fuel1AObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: FuelTankPressure1A

Appears In: FuelPressureTransducer1Status

DEFINITION

FuelTankPressure1A is Obsolete	T	*	*
FuelTankPressure1A < 200	*	T	*
FuelTankPressure1A > 400	*	*	T

Fuel1BObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: FuelTankPressure1B

Appears In: FuelPressureTransducer1Status

DEFINITION

FuelTankPressure1B is Obsolete	T	*	*
FuelTankPressure1B < 200	*	T	*
FuelTankPressure1B > 400	*	*	T

Fuel1Stage1Check

Macro

Description: Checks input for Stage 1 Overpressurization Limit

Comments: NA

References: FuelTankPressure1A, FuelTankPressure1B

Appears In: FuelPressureTransducer1Status

DEFINITION

FuelTankPressure1A >= 269	T	*
FuelTankPressure1A < 300	T	*
FuelTankPressure1B >= 269	*	T
FuelTankPressure1B < 300	*	T

Fuel1Stage2Check

Macro

Description: Checks input for Stage 1 Overpressurization Limit

Comments: NA

References: FuelTankPressure1A, FuelTankPressure1B

Appears In: FuelPressureTransducer1Status

DEFINITION

FuelTankPressure1A >=300	T
FuelTankPressure1A <= 400	T
FuelTankPressure1B >=300	T
FuelTankPressure1B <= 400	T

Fuel2AObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: FuelTankPressure2A

Appears In: FuelPressureTransducer2Status

DEFINITION

FuelTankPressure2A is Obsolete	T	*	*
FuelTankPressure2A < 200	*	T	*
FuelTankPressure2A > 400	*	*	T

Fuel2BObsoleteCheck

Macro

Description: Checks validity of Input

Comments: The input is considered obsolete if it is below 200 psia or above 400 psia.

References: FuelTankPressure2B

Appears In: FuelPressureTransducer2Status

DEFINITION

FuelTankPressure2B is Obsolete	T	*	*
FuelTankPressure2B < 200	*	T	*
FuelTankPressure2B > 400	*	*	T

Fuel2Stage1Check

Macro

Description: Checks input for Stage 1 Overpressurization Limit

Comments: NA

References: FuelTankPressure2A, FuelTankPressure2B

Appears In: FuelPressureTransducer2Status

DEFINITION

FuelTankPressure2A >= 269	T	*
FuelTankPressure2A < 300	T	*
FuelTankPressure2B >= 269	*	T
FuelTankPressure2B < 300	*	T

Fuel2Stage2Check

Macro

Description: Checks input for Stage 1 Overpressurization Limit

Comments: NA

References: FuelTankPressure2A, FuelTankPressure2B

Appears In: FuelPressureTransducer2Status

DEFINITION

FuelTankPressure2A >=300	T
FuelTankPressure2A <= 400	T
FuelTankPressure2B >=300	T
FuelTankPressure2B <= 400	T

Stage1OverpressureCheck

Macro

Description: This macro determines whether the Stage 1 overpressure limit has been reached. Stage 1 overpresure limit is declared when 2 out of the 3 pressure transducers on the oixidizer tank(1 pressure transducer) and fuel tank(2 pressure transducer) has reached overpressure Stage 1 limits.

Comments: NA

References: FuelPressureTransducer2Status, OxidizerPressureTransducerStatus, FuelPressureTransducer1Status

Appears In: Stage1OverpressurizationFaultStatus

DEFINITION

OxidizerPressureTransducerStatus in state OverPressure1	T	*	T
FuelPressureTransducer1Status in state OverPressure1	T	T	*
FuelPressureTransducer2Status in state OverPressure1	*	T	T

Stage2OverpressureCheck

Macro

Description: This macro determines whether the Stage 2 overpressure limit has been reached. Stage 2 overpresure limit is declared when 2 out of the 3 pressure transducers on the oixidizer tank(1 pressure transducer) and fuel tank(2 pressure transducer) has reached overpressure Stage 2 limits.

Comments:

References: FuelPressureTransducer2Status, OxidizerPressureTransducerStatus, FuelPressureTransducer1Status

Appears In: PyroValve4BEnable, PyroValve2BEnable, PyroValve3BEnable, PyroValve3AEnable, PyroValve2AEnable, PyroValve4AEnable, Stage2OverpressurizationFaultStatus

DEFINITION

OxidizerPressureTransducerStatus in state OverPressure2	T	*	T
FuelPressureTransducer1Status in state OverPressure2	T	T	*
FuelPressureTransducer2Status in state OverPressure2	*	T	T

Inputs

LatchValve1APosition

Input Value

Source: LatchValve1A

Message: LatchValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: LatchValve1AOpen, LatchValve1AClose

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: LatchValveInputMessage, LatchValve1A

Appears In: LatchValve1AOpen, LatchValve1AClose, LatchValve1AStatus, Stage1OverpressurizationResponse, LatchValve1A

DEFINITION

= Field Position from LatchValveInputMessage

Message for LatchValve1APosition was Received	T

= Previous Value

Message for LatchValve1APosition was Received	F
Time Since Message for LatchValve1APosition was Last Received <= 1 seconds	T

= Obsolete

Power Up	T	*	*
Message for LatchValve1APosition was Never Received	*	T	*
Time Since Message for LatchValve1APosition was Last Received <= 1 seconds	*	*	T
Message for LatchValve1APosition was Received	*	*	F

LatchValve1BPosition

Input Value

Source: LatchValve1B

Message: LatchValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: LatchValve1BOpen, LatchValve1BClose

Latency: Not Available

Min-time-after-output: 1 second

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: LatchValveInputMessage, LatchValve1B

Appears In: LatchValve1BOpen, LatchValve1BClose, LatchValve1BStatus, LatchValve1B

DEFINITION

= Field Position from LatchValveInputMessage

Message for LatchValve1BPosition was Received	T

= Previous Value

Message for LatchValve1BPosition was Received	F
Time Since Message for LatchValve1BPosition was Last Received <= 1 seconds	T

= Obsolete

Power Up	T	*	*
Message for LatchValve1BPosition was Never Received	*	T	*
Time Since Message for LatchValve1BPosition was Last Received > 1 seconds	*	*	T
Message for LatchValve1BPosition was Received	*	*	F

PyroValve1APosition

Input Value

Source: PyroValve1A

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve1AEnable, PyroValve1AFire

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve1A

Appears In: PyroValve1BFire, PyroValve4BFire, PyroValve2BFire, PyroValve2AFire, PyroValve1AEnable, PyroValve4AFire, PyroValve3BFire, PyroValve1AFire, PyroValve3AFire, Section1ControlMode, Stage2OverpressurizationResponse, PyroValve1A

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve1APosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve1APosition was Received	F
Message for PyroValve1APosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve1APosition was Never Received	*	T

PyroValve1BPosition

Input Value

Source: PyroValve1B

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve1BEnable, PyroValve1BFire

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve1B, PyroValve2APosition

Appears In: PyroValve1BEnable, Section1ControlMode, PyroValve1B

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve2APosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve2APosition was Received	F
Message for PyroValve2APosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve2APosition was Never Received	*	T

PyroValve2APosition

Input Value

Source: PyroValve2A

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Available

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve2AEnable, PyroValve2AFire

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve2A

Appears In: PyroValve4BEnable, PyroValve3BEnable, PyroValve2BEnable, PyroValve3AEnable, PyroValve2AEnable, PyroValve4AEnable, Section2ControlMode, PyroValve2A, PyroValve1BPosition

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve2APosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve2APosition was Received	F
Message for PyroValve2APosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve2APosition was Never Received	*	T

PyroValve2BPosition

Input Value

Source: PyroValve2B

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve2BEnable, PyroValve2BFire

Latency: Not Available

Min-time-after-output: 1 second

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve2B

Appears In: Section2ControlMode, PyroValve2B

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve2BPosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve2BPosition was Received	F
Message for PyroValve2BPosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve2BPosition was Never Received	*	T

PyroValve3APosition

Input Value

Source: PyroValve3A

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve3AEnable, PyroValve3AFire

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve3A

Appears In: Section3ControlMode, PyroValve3A

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve3APosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve3APosition was Received	F
Message for PyroValve3APosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve3APosition was Never Received	*	T

PyroValve3BPosition

Input Value

Source: PyroValve3B

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve3BEnable, PyroValve3BFire

Latency: Not Available

Min-time-after-output: 1 seconds

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve3B

Appears In: Section3ControlMode, PyroValve3B

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve3BPosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve3BPosition was Received	F
Message for PyroValve3BPosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve3BPosition was Never Received	*	T

PyroValve4APosition

Input Value

Source: PyroValve4A

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve4AEnable, PyroValve4AFire

Latency: Not Available

Min-time-after-output: 1 second

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve4A

Appears In: Section4ControlMode, PyroValve4A

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve4APosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve4APosition was Received	F
Message for PyroValve4APosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve4APosition was Never Received	*	T

PyroValve4BPosition

Input Value

Source: PyroValve4B

Message: PyroValveInputMessage

Possible Values (Expected Range): {Open, Close}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 1 second

Max-Time-Between-Inputs: 1 second

Max-Time-Before-First-Input: 5 seconds after system startup

Related Outputs: PyroValve4BEnable, PyroValve4BFire

Latency: Not Available

Min-time-after-output: 1 second

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Last Recorded Value of Position Sensor

Comments: NA

References: PyroValveInputMessage, PyroValve4B

Appears In: Section4ControlMode, PyroValve4B

DEFINITION

= Field Position from PyroValveInputMessage

Power Up	F
Message for PyroValve4BPosition was Received	T

= Previous Value

Power Up	F
Message for PyroValve4BPosition was Received	F
Message for PyroValve4BPosition was Never Received	F

= Obsolete

Power Up	T	*
Message for PyroValve4BPosition was Never Received	*	T

OxidizerTankPressureA

Input Value

Source: OxidizerPressureTransducer

Message: OxidizerPressureTransducerAMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: NA

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: None

Latency: NA

Min-time-after-output: NA

Exception-Handling: NA

Obsolescence: 5 seconds

Exception-Handling: Not Available

Description: String A Input from Oxidizer Pressure Transducer

Comments: NA

References: OxidizerPressureTransducerAMessage, OxidizerPressureTransducer, ❑DP.3.1

Appears In: OxidizerStage2Check, OxidizerAObsoleteCheck, OxidizerStage1Check, OxidizerPressureTransducer

DEFINITION

= Field Pressure from OxidizerPressureTransducerAMessage

Message for OxidizerTankPressureA was Received	T

= Previous Value

Message for OxidizerTankPressureA was Received	F
Time Since Message for OxidizerTankPressureA was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for OxidizerTankPressureA was Never Received	*	T	*
Time Since Message for OxidizerTankPressureA was Last Received > 5 seconds	*	*	T
Message for OxidizerTankPressureA was Received	*	*	F

OxidizerTankPressureB

Input Value

Source: OxidizerPressureTransducer

Message: OxidizerPressureTransducerBMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: NA

Latency: NA

Min-time-after-output: NA

Exception-Handling: NA

Obsolescence: 5 seconds

Exception-Handling: Not Available

Description: String B Input from Oxidizer Pressure Transducer

Comments: NA

References: OxidizerPressureTransducerBMessage, OxidizerPressureTransducer

Appears In: OxidizerStage2Check, OxidizerBObsoleteCheck, OxidizerStage1Check, OxidizerPressureTransducer

DEFINITION

= Field Pressure from OxidizerPressureTransducerBMessage

Message for OxidizerTankPressureB was Received	T

= Previous Value

Message for OxidizerTankPressureB was Received	F
Time Since Message for OxidizerTankPressureB was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for OxidizerTankPressureB was Never Received	*	T	*
Time Since Message for OxidizerTankPressureB was Last Received > 5 seconds	*	*	T
Message for OxidizerTankPressureB was Received	*	*	F

FuelTankPressure1A

Input Value

Source: FuelPressureTransducer1

Message: FuelPressureTransducer1AMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: NA

Latency: NA

Min-time-after-output: NA

Exception-Handling: NA

Obsolescence: 5 seconds

Exception-Handling: Not Available

Description: String A Input from Fuel Pressure Transducer 1

Comments: NA

References: FuelPressureTransducer1AMessage, FuelPressureTransducer1

Appears In: Fuel1Stage1Check, Fuel1Stage2Check, Fuel1AObsoleteCheck, FuelPressureTransducer1

DEFINITION

= Field Pressure from FuelPressureTransducer1AMessage

Message for FuelTankPressure1A was Received	T

= Previous Value

Message for FuelTankPressure1A was Received	F
Time Since Message for FuelTankPressure1A was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for FuelTankPressure1A was Never Received	*	T	*
Time Since Message for FuelTankPressure1A was Last Received > 5 seconds	*	*	T
Message for FuelTankPressure1A was Received	*	*	F

FuelTankPressure1B

Input Value

Source: FuelPressureTransducer1

Message: FuelPressureTransducer1BMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: NA

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: NA

Latency: NA

Min-time-after-output: NA

Exception-Handling: Not Available

Obsolescence: 5 seconds

Exception-Handling: Not Available

Description: String B Input from Fuel Pressure Transducer 1

Comments: NA

References: FuelPressureTransducer1BMessage, FuelPressureTransducer1

Appears In: Fuel1Stage1Check, Fuel1Stage2Check, Fuel1BObsoleteCheck, FuelPressureTransducer1

DEFINITION

= Field Pressure from FuelPressureTransducer1BMessage

Message for FuelTankPressure1B was Received	T

= Previous Value

Message for FuelTankPressure1B was Received	F
Time Since Message for FuelTankPressure1B was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for FuelTankPressure1B was Never Received	*	T	*
Time Since Message for FuelTankPressure1B was Last Received > 5 seconds	*	*	T
Message for FuelTankPressure1B was Received	*	*	F

FuelTankPressure2A

Input Value

Source: FuelPressureTransducer2

Message: FuelPressureTransducer2AMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: NA

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: NA

Latency: NA

Min-time-after-output: NA

Exception-Handling: Not Available

Obsolescence: 5 seconds

Exception-Handling:

Description: String A Input from Fuel Pressure Transducer 2

Comments: NA

References: FuelPressureTransducer2AMessage, FuelPressureTransducer2

Appears In: Fuel2Stage2Check, Fuel2Stage1Check, Fuel2AObsoleteCheck, FuelPressureTransducer2

DEFINITION

= Field Pressure from FuelPressureTransducer2AMessage

Message for FuelTankPressure2A was Received	T

= Previous Value

Message for FuelTankPressure2A was Received	F
Time Since Message for FuelTankPressure2A was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for FuelTankPressure2A was Never Received	*	T	*
Time Since Message for FuelTankPressure2A was Last Received > 5 seconds	*	*	T
Message for FuelTankPressure2A was Received	*	*	F

FuelTankPressure2B

Input Value

Source: FuelPressureTransducer2

Message: FuelPressureTransducer2BMessage

Possible Values (Expected Range): 200-400 psia

Units: psia

Granularity: NA

Exception-Handling: NA

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: 0 seconds

Max-Time-Between-Inputs: 5 seconds

Max-Time-Before-First-Input: 5 seconds

Related Outputs: NA

Latency: NA

Min-time-after-output: NA

Exception-Handling: Not Available

Obsolescence: 5 seconds

Exception-Handling: Not Available

Description: String B Input from Fuel Pressure Transducer 2

Comments: NA

References: FuelPressureTransducer2BMessage, FuelPressureTransducer2

Appears In: Fuel2BObsoleteCheck, Fuel2Stage2Check, Fuel2Stage1Check, FuelPressureTransducer2

DEFINITION

= Field Pressure from FuelPressureTransducer2BMessage

Message for FuelTankPressure2B was Received	T

= Previous Value

Message for FuelTankPressure2B was Received	F
Time Since Message for FuelTankPressure2B was Last Received <= 5 seconds	T

= Obsolete

Power Up	T	*	*
Message for FuelTankPressure2B was Never Received	*	T	*
Time Since Message for FuelTankPressure2B was Last Received > 5 seconds	*	*	T
Message for FuelTankPressure2B was Received	*	*	F

Stage1FaultProtectionCommandInput

Control Input

Source: ADCS

Message: FaultProtectionCommandInputMessage

Possible Values (Expected Range): {Enable, Disable}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: See Below

Load: Not Available

Min-Time-Between-Inputs: NA

Max-Time-Between-Inputs: NA

Exception-Handling: Not Available

Obsolescence: NA

Exception-Handling: Not Available

Description: This input tells the PRS controller whether to enable or disable Stage1 Overpressurizationfault protection.

Comments: NA

References: FaultProtectionCommandInputMessage, ADCS

Appears In: Stage1OverpressureFaultProtection, SupervisorMode, ADCS

DEFINITION

= Field COM from FaultProtectionCommandInputMessage

Power Up	F
Message for Stage1FaultProtectionCommandInput was Received	T

= Previous Value

Power Up	F
Message for Stage1FaultProtectionCommandInput was Received	F
Message for Stage1FaultProtectionCommandInput was Never Received	F

= Obsolete

Power Up	T	*
Message for Stage1FaultProtectionCommandInput was Never Received	*	T

Stage2FaultProtectionCommandInput

Control Input

Source: ADCS

Message: FaultProtectionCommandInputMessage

Possible Values (Expected Range): {Enable, Disable}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: NA

Load: NA

Min-Time-Between-Inputs: NA

Max-Time-Between-Inputs: NA

Exception-Handling: Not Available

Obsolescence: NA

Exception-Handling: Not Available

Description: This input tells the PRS controller whether to enable or disable Stage2 Overpressurizationfault protection.

Comments: NA

References: FaultProtectionCommandInputMessage, ADCS

Appears In: Stage2OverpressureFaultProtection, SupervisorMode, ADCS

DEFINITION

= Field COM from FaultProtectionCommandInputMessage

Power Up	F
Message for Stage2FaultProtectionCommandInput was Received	T

= Previous Value

Power Up	F
Message for Stage2FaultProtectionCommandInput was Received	F
Message for Stage2FaultProtectionCommandInput was Never Received	F

= Obsolete

Power Up	T	*
Message for Stage2FaultProtectionCommandInput was Never Received	*	T

ManualControlInput

Control Input

Source: ADCS

Message: ManualControlInputMessage

Possible Values (Expected Range): See Message

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: NA

Load: NA

Min-Time-Between-Inputs: NA

Max-Time-Between-Inputs: NA

Exception-Handling: Not Available

Obsolescence: 1 second

Exception-Handling: Not Available

Description: Manual Commands are sent through this input

Comments: NA

References: ManualControlInputMessage, ADCS

Appears In: ADCS

DEFINITION

= Field COM from ManualControlInputMessage

Message for ManualControlInput was Received	T

= Previous Value

Message for ManualControlInput was Received	F
Time Since Message for ManualControlInput was Last Received <= 1 seconds	T

= Obsolete

Power Up	T	*	*
Message for ManualControlInput was Never Received	*	T	*
Time Since Message for ManualControlInput was Last Received > 1 seconds	*	*	T
Message for ManualControlInput was Received	*	*	F

ResponseReadyInput

Input Value

Source: ADCS

Message: ResponseReadyInputMessage

Possible Values (Expected Range): {Ready}

Units: NA

Granularity: NA

Exception-Handling: Not Available

Timing Behavior: NA

Load: NA

Min-Time-Between-Inputs: NA

Max-Time-Between-Inputs: NA

Max-Time-Before-First-Input: NA

Related Outputs: ReconfiguringOutput

Latency: Not Available

Min-time-after-output: Not Available

Exception-Handling: Not Available

Obsolescence: 120 seconds, this is the maximum alotted fault protection response time

Exception-Handling: Not Available

Description: This input from the ADCS is required before the PRS may perform any reconfiguration.

Comments: NA

References: ResponseReadyInputMessage, ADCS

Appears In: ADCS

DEFINITION

= Field Ready from ResponseReadyInputMessage

Message for ResponseReadyInput was Received	T

= Previous Value

Message for ResponseReadyInput was Received	F
Time Since Message for ResponseReadyInput was Last Received <= 120 seconds	T

= Obsolete

Power Up	T	*	*
Message for ResponseReadyInput was Never Received	*	T	*
Time Since Message for ResponseReadyInput was Last Received > 120 seconds	*	*	T
Message for ResponseReadyInput was Received	*	*	F

Messages

ControlModeOutputMessage

Message

Description: Control Mode Output Message

Comments: NA

References:

Appears In: ControlModePrimaryOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, ADCS

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Mode		{PRIMARY, SECONDARY, OFFNOMINAL}		

FIELD DESCRIPTIONS

Field	Description

ReconfiguringOutputMessage

Message

Description: Reconfiguring Output Message

Comments: NA

References:

Appears In: ReconfiguringOutput, ADCS

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Reconfiguring		{ReconfiguringPRS}		

FIELD DESCRIPTIONS

Field	Description

ResponseReadyInputMessage

Message

Description: Response Ready Message

Comments: NA

References:

Appears In: ADCS, ResponseReadyInput

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Ready		{Ready}		

FIELD DESCRIPTIONS

Field	Description

FaultProtectionCommandInputMessage

Message

Description: Fault Protection Command Input Message

Comments: NA

References:

Appears In: ADCS, Stage2FaultProtectionCommandInput, Stage1FaultProtectionCommandInput

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
COM		{Enable, Disable}		

FIELD DESCRIPTIONS

Field	Description

ManualControlInputMessage

Message

Description: Manual Control Message

Comments: NA

References:

Appears In: ADCS, ManualControlInput

DATA REPRESENTATION

Enable,PyroValve1BFire,PyroValve2AEnable,PyroValve2AFire,PyroValve2BEnable,PyroValve2BFire,PyroValve3A

FIELD DESCRIPTIONS

Field	Description

LatchValveOutputMessage

Message

Description: Latch Valve Output Message

Comments: NA

References:

Appears In: LatchValve1BOpen, LatchValve1AOpen, LatchValve1BClose, LatchValve1AClose, LatchValve1B, LatchValve1A

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
COM	Latch Valve Command	{Open, Close}	0	0

FIELD DESCRIPTIONS

Field	Description
COM	Latch Valve Command

LatchValveInputMessage

Message

Description: Latch Valve Input Message

Comments: NA

References:

Appears In: LatchValve1B, LatchValve1A, LatchValve1BPosition, LatchValve1APosition

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Position	Latch Valve Position	{Open, Close}	0	0

FIELD DESCRIPTIONS

Field	Description
COM	Latch Valve Position

PyroValveInputMessage

Message

Description: Pyrotechnic Valve Input Message

Comments: NA

References:

Appears In: PyroValve3A, PyroValve1B, PyroValve2B, PyroValve4A, PyroValve2A, PyroValve3B, PyroValve4B, PyroValve1A, PyroValve2BPosition, PyroValve3APosition, PyroValve3BPosition, PyroValve1APosition, PyroValve4BPosition, PyroValve1BPosition, PyroValve4APosition, PyroValve2APosition

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Position	Pyro Valve Position	{Open, Close}	0	0

FIELD DESCRIPTIONS

Field	Description
COM	Latch Valve Position

PyroValveEnableMessage

Message

Description: Pyrotechnic Valve Enable Message

Comments: NA

References:

Appears In: PyroValve4BEnable, PyroValve3BEnable, PyroValve2BEnable, PyroValve3AEnable, PyroValve1BEnable, PyroValve2AEnable, PyroValve4AEnable, PyroValve1AEnable, PyroValve3A, PyroValve1B, PyroValve2B, PyroValve4A, PyroValve2A, PyroValve3B, PyroValve4B, PyroValve1A

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
COM	Pyro Valve Command	{Enable}	0	0

FIELD DESCRIPTIONS

Field	Description
COM	Pyro Valve Command

PyroValveFireMessage

Message

Description: Pyrotechnic Valve FireMessage

Comments: NA

References:

Appears In: PyroValve1BFire, PyroValve4BFire, PyroValve2BFire, PyroValve2AFire, PyroValve4AFire, PyroValve3BFire, PyroValve1AFire, PyroValve3AFire, PyroValve3A, PyroValve1B, PyroValve2B, PyroValve4A, PyroValve2A, PyroValve3B, PyroValve4B, PyroValve1A

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
COM	Pyro Valve Fire	{Fire}	0	0

FIELD DESCRIPTIONS

Field	Description
COM	Pyro Valve Command

OxidizerPressureTransducerAMessage

Message

Description: Oxidizer Pressure Transducer Message

Comments: NA

References:

Appears In: OxidizerPressureTransducer, OxidizerTankPressureA

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Oxidizer Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

OxidizerPressureTransducerBMessage

Message

Description: Oxidizer Pressure Transducer Message

Comments: NA

References:

Appears In: OxidizerPressureTransducer, OxidizerTankPressureB

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Oxidizer Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

FuelPressureTransducer1AMessage

Message

Description: Fuel Pressure Transducer Message

Comments: NA

References:

Appears In: FuelPressureTransducer1, FuelTankPressure1A

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Fuel Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

FuelPressureTransducer1BMessage

Message

Description: Fuel Pressure Transducer Message

Comments: NA

References:

Appears In: FuelPressureTransducer1, FuelTankPressure1B

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Fuel Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

FuelPressureTransducer2AMessage

Message

Description: Fuel Pressure Transducer Message

Comments: NA

References:

Appears In: FuelPressureTransducer2, FuelTankPressure2A

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Fuel Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

FuelPressureTransducer2BMessage

Message

Description: Fuel Pressure Transducer Message

Comments: NA

References:

Appears In: FuelPressureTransducer2, FuelTankPressure2B

DATA REPRESENTATION

Field Name	Description	Type	Start Bit	End Bit
Pressure	Pressure Reading of Fuel Tank	Real	0	255

FIELD DESCRIPTIONS

Field	Description
Pressure	Pressure Reading of Oxidizer Tank

Devices

ADCS

Device

Description: ADCS Controller

Comments: NA

References: FaultProtectionCommandInputMessage, ResponseReadyInputMessage, ReconfiguringOutputMessage, ManualControlInputMessage, ControlModeOutputMessage

Appears In: ControlModePrimaryOutput, ReconfiguringOutput, ControlModeSecondaryOutput, ControlModeOffNominalOutput, ManualControlInput, Stage2FaultProtectionCommandInput, Stage1FaultProtectionCommandInput, ResponseReadyInput

DATA PORTS

Message	URI
ControlModeOutputMessage	relativefile:PRS/ControlModeOutput.sim
ReconfiguringOutputMessage	relativefile:PRS/Reconfiguring.sim
ResponseReadyInputMessage	relativefile:PRS/ResponseReady.sim
FaultProtectionCommandInputMessage	relativefile:PRS/Stage1COM.sim
FaultProtectionCommandInputMessage	relativefile:PRS/Stage2COM.sim
ManualControlInputMessage	relativefile:PRS/ManualControl.sim

LatchValve1A

Device

Description: Latch Valve 1A

Comments: NA

References: LatchValveInputMessage, LatchValveOutputMessage

Appears In: LatchValve1AOpen, LatchValve1AClose, LatchValve1APosition

DATA PORTS

Message	URI
LatchValveOutputMessage	relativefile:PRS/LatchValve1AOutput.sim
LatchValveInputMessage	relativefile:PRS/LatchValve1AInput.sim

LatchValve1B

Device

Description: Latch Valve 1B

Comments: NA

References: LatchValveInputMessage, LatchValveOutputMessage

Appears In: LatchValve1BOpen, LatchValve1BClose, LatchValve1BPosition

DATA PORTS

Message	URI
LatchValveOutputMessage	relativefile:PRS/LatchValve1BOutput.sim
LatchValveInputMessage	relativefile:PRS/LatchValve1BInput.sim

PyroValve1A

Device

Description: Pyrotechnic Valve 1A

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve1AEnable, PyroValve1AFire, PyroValve1APosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve1AInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve1AEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve1AClose.sim

PyroValve1B

Device

Description: Pyrotechnic Valve 1B

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve1BFire, PyroValve1BEnable, PyroValve1BPosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve1BInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve1BEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve1BClose.sim

PyroValve2A

Device

Description: Pyrotechnic Valve 2A

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve2AEnable, PyroValve2AFire, PyroValve2APosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve2AInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve2AEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve2AClose.sim

PyroValve2B

Device

Description: Pyrotechnic Valve 2B

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve2BEnable, PyroValve2BFire, PyroValve2BPosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve2BInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve2BEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve2BClose.sim

PyroValve3A

Device

Description: Pyrotechnic Valve 3A

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve3AEnable, PyroValve3AFire, PyroValve3APosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve3AInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve3AEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve3AClose.sim

PyroValve3B

Device

Description: Pyrotechnic Valve 3B

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve3BEnable, PyroValve3BFire, PyroValve3BPosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve3BInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve3BEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve3BClose.sim

PyroValve4A

Device

Description: Pyrotechnic Valve 4A

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve4AEnable, PyroValve4AFire, PyroValve4APosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve4AInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve4AEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve4AClose.sim

PyroValve4B

Device

Description: Pyrotechnic Valve 4B

Comments: NA

References: PyroValveInputMessage, PyroValveEnableMessage, PyroValveFireMessage

Appears In: PyroValve4BEnable, PyroValve4BFire, PyroValve4BPosition

DATA PORTS

Message	URI
PyroValveInputMessage	relativefile:PRS/PyroValve4BInput.sim
PyroValveEnableMessage	relativefile:PRS/PyroValve4BEnable.sim
PyroValveFireMessage	relativefile:PRS/PyroValve4BClose.sim

OxidizerPressureTransducer

Device

Description: Oxidizer Pressure Transducer

Comments: NA

References: OxidizerPressureTransducerAMessage, OxidizerPressureTransducerBMessage

Appears In: OxidizerTankPressureA, OxidizerTankPressureB

DATA PORTS

Message	URI
OxidizerPressureTransducerAMessage	relativefile:PRS/OxidizerTransducerA.sim
OxidizerPressureTransducerBMessage	relativefile:PRS/OxidizerTransducerB.sim

FuelPressureTransducer1

Device

Description: Fuel Prressure Transducer 1

Comments: NA

References: FuelPressureTransducer1BMessage, FuelPressureTransducer1AMessage

Appears In: FuelTankPressure1A, FuelTankPressure1B

DATA PORTS

Message	URI
FuelPressureTransducer1AMessage	relativefile:PRS/FuelTransducer1A.sim
FuelPressureTransducer1BMessage	relativefile:PRS/FuelTransducer1B.sim

FuelPressureTransducer2

Device

Description: Fuel Prressure Transducer 2

Comments: NA

References: FuelPressureTransducer2AMessage, FuelPressureTransducer2BMessage

Appears In: FuelTankPressure2A, FuelTankPressure2B

DATA PORTS

Message	URI
FuelPressureTransducer2AMessage	relativefile:PRS/FuelTransducer2A.sim
FuelPressureTransducer2BMessage	relativefile:PRS/FuelTransducer2B.sim

5000
39