

# Plan for Software Aspects of Certification

for the

**<Program Name>**

Document No: <Doc Number>

Revision: -

\_\_\_\_\_  
<Name>, Program Manager

\_\_\_\_\_  
Date

\_\_\_\_\_  
<Name>, Technical Project Lead

\_\_\_\_\_  
Date

\_\_\_\_\_  
<Name>, Engineer

\_\_\_\_\_  
Date

\_\_\_\_\_  
<Name>, Quality Engineer

\_\_\_\_\_  
Date

## **Notice**

This document and the information contained herein are the property of <company name>. Any reproduction, disclosure or use thereof is prohibited except as authorized in writing by <company name>. Recipient accepts the responsibility for maintaining the confidentiality of the contents of this document.



## Table of Contents

Section	Page
<b>1.0 INTRODUCTION .....</b>	<b>11</b>
1.1 Purpose .....	11
1.2 Scope .....	11
1.3 Definitions .....	11
1.4 Part Number and Nomenclature .....	12
1.5 Special Considerations Cross Reference .....	13
1.6 Team Members and Signature Authority .....	14
1.6.1 <i>Independent Reporting Structure</i> .....	15
1.6.1.1 Software Quality Assurance Independence .....	15
1.6.1.2 Verification Independence of DO-178C Objectives .....	16
1.6.2 <i>Signature Authority</i> .....	18
1.7 Organizational Responsibilities .....	19
1.8 Acronyms and Abbreviations .....	21
1.9 Applicable Documents .....	22
1.9.1 <i>External Documents</i> .....	22
1.9.2 <i>Internal Documents</i> .....	22
<b>2.0 SYSTEM OVERVIEW .....</b>	<b>23</b>
2.1 Mechanical Systems Top level Diagram .....	23
2.1.1 <i>System Functions Allocated to Mechanical Hardware</i> .....	23
2.2 Electrical Systems Top Level Block Diagram .....	24
2.2.1 <i>System Functions Allocated to Electrical Hardware and Software</i> .....	25
2.2.1.1 Data Acquisition Printed Circuit Board #1 .....	25
2.2.1.2 Input / Output Printed Circuit Board #2 .....	25
2.2.1.3 Monitor Printed Circuit Board #3 .....	25
2.3 System Functional Description .....	26
2.3.1 <i>System Failure Conditions</i> .....	27
2.3.2 <i>High-Level Hardware Functions and Contribution to Potential Failures</i> .....	27
2.3.3 <i>Safety and Partitioning</i> .....	27
<b>3.0 SOFTWARE OVERVIEW .....</b>	<b>28</b>
3.1 Software Architectural Block Diagram .....	28
3.2 Processor #1 .....	28
3.2.1 <i>States and Modes</i> .....	28
3.2.2 <i>Tasks</i> .....	28
3.3 Processor #2 .....	28
3.3.1 <i>States and Modes</i> .....	28
3.3.2 <i>Tasks</i> .....	28
3.4 COTS Software Identification .....	28
3.4.1 <i>Real Time Operating System</i> .....	28
3.4.2 <i>Board Support Package</i> .....	28
3.4.3 <i>Compiler Provided Libraries</i> .....	28
3.5 Deactivated Code Partitioning .....	29
3.5.1 <i>USB Interface Code</i> .....	29
3.5.2 <i>RS-232 Interface Code</i> .....	29
3.5.3 <i>Ethernet Interface Code</i> .....	29
3.5.4 <i>Boot Load Code Partitioning</i> .....	29
3.6 Safety and Partitioning .....	30

3.6.1	<i>Safety Monitoring</i> .....	30
3.7	Resource Sharing <Example Text> .....	30
3.8	Redundancy .....	30
3.9	Fault Tolerance <Example Text> .....	30
3.10	Timing and Task Scheduling .....	30
3.10.1	<i>Timing</i> <Example Text> .....	30
3.10.2	<i>Task Scheduling</i> <Example Text> .....	31
<b>4.0</b>	<b>CERTIFICATION CONSIDERATIONS</b> .....	<b>32</b>
4.1	Certification Basis and Means of Compliance .....	32
4.2	Issue Papers and Certification Review Items (CRI) .....	32
4.3	Software Development Assurance Levels .....	33
4.3.1	<i>Display DAL and Worst Case Failure Condition</i> .....	33
4.3.2	<i>Command DAL and Worst Case Failure Condition</i> .....	33
4.3.3	<i>Monitor DAL and Worst Case Failure Condition</i> .....	34
4.3.4	<i>Video Processor DAL and Worst Case Failure Condition</i> .....	34
4.4	Software Level Determination .....	35
4.4.1	<i>DO-178C Objectives By Design Assurance Level</i> .....	36
4.5	Compliance Matrix .....	37
4.5.1	<i>Software Conformity Objectives</i> .....	64
4.6	Certification Authority Level of Involvement .....	71
4.6.1	<i>Certification Authority Criteria for Level of Involvement</i> .....	73
4.6.2	<i>Proposed Level of Involvement</i> .....	73
<b>5.0</b>	<b>SOFTWARE LIFECYCLE</b> .....	<b>74</b>
5.1	V-Model Development Approach .....	75
5.2	Development of Multiple DAL's Within A Single Lifecycle Process .....	76
5.2.1	<i>Section 5 Applicability for Levels A, B and C CSCI(s)</i> .....	76
5.2.2	<i>Section 5 Applicability for Level D CSCI(s)</i> .....	77
5.3	Team Member Responsibility Summary .....	77
5.4	Relationship Between Processes and Activities .....	81
5.5	Interaction Among Processes .....	82
5.5.1	<i>System Lifecycle Flow Diagram</i> .....	82
5.5.2	<i>Hardware and Software Lifecycle Flow Diagram</i> .....	83
5.5.3	<i>Software Incremental Development Lifecycle Flow Diagram</i> .....	84
5.6	Means of Providing Feedback .....	85
5.6.1	<i>Feedback to the System and Safety Process</i> .....	85
5.6.2	<i>Feedback to the Development and Integral Processes</i> .....	86
5.7	Traceability of Reviews and Analysis Results .....	87
5.7.1	<i>Transition Review Planning</i> .....	88
5.7.2	<i>Peer Review Planning</i> .....	88
5.8	Summary of Problem Reporting Methods .....	89
5.9	Software Planning Process .....	90
5.9.1	<i>Software Planning Process Objectives</i> .....	90
5.9.2	<i>Software Planning Process Inputs</i> .....	90
5.9.3	<i>Software Planning Process Outputs</i> .....	91
5.9.4	<i>Software Planning Process Activities</i> .....	91
5.9.5	<i>Technical Interfaces</i> .....	92
5.9.6	<i>Software Planning Process Tool Usage</i> .....	94
5.9.7	<i>Software Planning Process Transition Criteria</i> .....	94
5.9.7.1	<i>Transition Criteria for Entry into Planning Process</i> .....	94
5.9.7.2	<i>Transition Criteria for Exit from Planning Process</i> .....	95

5.9.8	<i>Integral Processes</i> .....	96
5.9.8.1	Software Verification Process Objectives and Activities .....	96
5.9.8.1.1	Software Verification Plan Preparation .....	96
5.9.8.1.2	Software Reviews and Analysis.....	96
5.9.8.1.2.1	Software Planning Review .....	97
5.9.8.2	Software Configuration Management Objectives and Activities.....	98
5.9.8.2.1	Configuration Management Plan Preparation .....	98
5.9.8.2.2	Configuration Identification, Baselines and Traceability.....	98
5.9.8.2.3	Configuration Status Accounting .....	98
5.9.8.2.4	Problem Reporting, Tracking and Corrective Action .....	99
5.9.8.2.5	Change Control and Change Review .....	99
5.9.8.3	Software Quality Assurance Objectives and Activities .....	100
5.9.8.3.1	Software Quality Assurance Plan Preparation.....	100
5.9.8.3.2	SQA Independence during the Planning Process.....	100
5.9.8.3.3	SQA Lifecycle Process Audits .....	100
5.9.8.3.4	SQA Conformity Review Planning .....	100
5.9.8.3.5	Software Transition Criteria Satisfaction Review .....	101
5.9.8.3.6	SQA Reporting and Corrective / Preventive Action.....	101
5.9.8.4	Certification Liaison Objectives and Activities .....	102
5.9.8.4.1	Means of Compliance and Planning .....	102
5.9.8.4.2	Compliance Substantiation.....	102
5.10	Software Requirements Process.....	103
5.10.1	<i>Software Requirements Process Objectives</i> .....	103
5.10.2	<i>Software Requirements Process Inputs</i> .....	103
5.10.3	<i>Software Requirements Process Outputs</i> .....	103
5.10.4	<i>Software Requirements Process Activities</i> .....	104
5.10.5	<i>Technical Interfaces</i> .....	106
5.10.6	<i>Software Requirements Process Tool Usage</i> .....	107
5.10.7	<i>Software Requirements Process Transition Criteria</i> .....	108
5.10.7.1	Transition Criteria for Entry into Requirements Process.....	108
5.10.7.2	Transition Criteria for Exit from Requirements Process.....	109
5.10.8	<i>Integral Processes</i> .....	110
5.10.8.1	Software Verification Process Objectives and Activities .....	110
5.10.8.1.1	Software Reviews and Analysis.....	110
5.10.8.1.1.1	Software Requirements Review .....	111
5.10.8.2	Software Configuration Management Objectives and Activities.....	111
5.10.8.2.1	Configuration Identification, Baselines and Traceability.....	111
5.10.8.2.2	Configuration Status Accounting .....	111
5.10.8.2.3	Problem Reporting, Tracking and Corrective Action .....	113
5.10.8.2.4	Change Control and Change Review .....	113
5.10.8.3	Software Quality Assurance Objectives and Activities .....	114
5.10.8.3.1	SQA Lifecycle Process Audits .....	114
5.10.8.3.2	Software Transition Criteria Satisfaction Review .....	114
5.10.8.3.3	SQA Reporting and Corrective / Preventive Action.....	114
5.10.8.4	Certification Liaison Objectives and Activities .....	115
5.10.8.4.1	Means of Compliance and Requirements .....	115
5.10.8.4.2	Compliance Substantiation.....	115
5.11	Software Design Process.....	116
5.11.1	<i>Software Design Process Objectives</i> .....	116
5.11.2	<i>Software Design Process Inputs</i> .....	116
5.11.3	<i>Software Design Process Outputs</i> .....	116
5.11.4	<i>Software Design Process Activities</i> .....	116

5.11.5	<i>Technical Interfaces</i> .....	117
5.11.6	<i>Software Design Process Tool Usage</i> .....	118
5.11.7	<i>Software Design Process Transition Criteria</i> .....	120
5.11.7.1	Transition Criteria for Entry into Design Process .....	120
5.11.7.2	Transition Criteria for Exit from Design Process .....	121
5.11.8	<i>Integral Processes</i> .....	122
5.11.8.1	Software Verification Process Objectives and Activities .....	122
5.11.8.1.1	Software Reviews and Analysis.....	122
5.11.8.1.1.1	Software Preliminary Design Review .....	123
5.11.8.1.1.2	Software Critical Design Review .....	124
5.11.8.2	Software Configuration Management Objectives and Activities.....	125
5.11.8.2.1	Configuration Identification, Baselines and Traceability.....	125
5.11.8.2.2	Configuration Status Accounting .....	125
5.11.8.2.3	Problem Reporting, Tracking and Corrective Action .....	126
5.11.8.2.4	Change Control and Change Review .....	126
5.11.8.3	Software Quality Assurance Objectives and Activities .....	127
5.11.8.3.1	SQA Lifecycle Process Audits .....	127
5.11.8.3.2	Software Transition Criteria Satisfaction Review .....	127
5.11.8.3.3	SQA Reporting and Corrective / Preventive Action.....	127
5.11.8.4	Certification Liaison Objectives and Activities .....	128
5.11.8.4.1	Means of Compliance and Requirements .....	128
5.11.8.4.2	Compliance Substantiation.....	128
5.12	Software Coding Process .....	129
5.12.1	<i>Software Coding Process Objectives</i> .....	129
5.12.2	<i>Software Coding Process Inputs</i> .....	129
5.12.3	<i>Software Coding Process Outputs</i> .....	129
5.12.4	<i>Software Coding Process Activities</i> .....	130
5.12.5	<i>Technical Interfaces</i> .....	130
5.12.6	<i>Software Coding Process Tool Usage</i> .....	131
5.12.7	<i>Software Coding Process Transition Criteria</i> .....	132
5.12.7.1	Transition Criteria for Entry into Code Process .....	132
5.12.7.2	Transition Criteria for Exit from Code Process.....	133
5.12.8	<i>Integral Processes</i> .....	134
5.12.8.1	Software Verification Process Objectives and Activities .....	134
5.12.8.1.1	Software Reviews and Analysis.....	134
5.12.8.1.1.1	Software Code Review .....	134
5.12.8.2	Software Configuration Management Objectives and Activities.....	135
5.12.8.2.1	Configuration Identification, Baselines and Traceability.....	135
5.12.8.2.2	Configuration Status Accounting .....	135
5.12.8.2.3	Problem Reporting, Tracking and Corrective Action .....	136
5.12.8.2.4	Change Control and Change Review .....	136
5.12.8.3	Software Quality Assurance Objectives and Activities .....	137
5.12.8.3.1	SQA Lifecycle Process Audits .....	137
5.12.8.3.2	Software Transition Criteria Satisfaction Review .....	137
5.12.8.3.3	SQA Reporting and Corrective / Preventive Action.....	137
5.12.8.4	Certification Liaison Objectives and Activities .....	138
5.12.8.4.1	Means of Compliance and Requirements .....	138
5.12.8.4.2	Compliance Substantiation.....	138
5.13	Integration Process .....	139
5.13.1	<i>Integration Process Objectives</i> .....	139
5.13.2	<i>Integration Process Inputs</i> .....	139

5.13.3	Integration Process Outputs.....	139
5.13.4	Integration Process Activities .....	139
5.13.5	Technical Interfaces .....	140
5.13.6	Software Integration Process Tool Usage .....	142
5.13.7	Integration Process Transition Criteria.....	142
5.13.7.1	Transition Criteria for Entry into Integration Process .....	142
5.13.7.2	Transition Criteria for Exit from Integration Process .....	143
5.13.8	Integral Processes .....	144
5.13.8.1	Software Verification Process Objectives and Activities .....	144
5.13.8.1.1	Software Reviews and Analysis.....	144
5.13.8.1.1.1	System Integration Review .....	144
5.13.8.2	Software Configuration Management Objectives and Activities.....	145
5.13.8.2.1	Configuration Identification, Baselines and Traceability.....	145
5.13.8.2.2	Configuration Status Accounting .....	145
5.13.8.2.3	Problem Reporting, Tracking and Corrective Action .....	146
5.13.8.2.4	Change Control and Change Review .....	146
5.13.8.3	Software Quality Assurance Objectives and Activities .....	147
5.13.8.3.1	SQA Lifecycle Process Audits .....	147
5.13.8.3.2	Software Transition Criteria Satisfaction Review .....	147
5.13.8.3.3	SQA Reporting and Corrective / Preventive Action.....	147
5.13.8.4	Certification Liaison Objectives and Activities .....	148
5.13.8.4.1	Means of Compliance and Requirements .....	148
5.13.8.4.2	Compliance Substantiation.....	148
5.14	Software Testing Process .....	149
5.14.1	Software Testing Process Objectives .....	149
5.14.1.1	Integration Test Objectives .....	149
5.14.1.2	Verification Test Objectives .....	150
5.14.2	Software Testing Process Inputs .....	150
5.14.3	Software Testing Process Outputs.....	150
5.14.4	Software Testing Process Activities .....	151
5.14.4.1	Test Case and Test Procedure Development .....	151
5.14.4.2	Test Execution and Test Results Compilation .....	151
5.14.5	Technical Interfaces .....	152
5.14.6	Software Testing Process Tool Usage.....	154
5.14.7	Software Testing Process Transition Criteria .....	154
5.14.7.1	Transition Criteria for Entry into Software Testing Process .....	154
5.14.7.2	Transition Criteria for Exit from Software Testing Process .....	155
5.14.8	Integral Processes .....	156
5.14.8.1	Software Verification Process Objectives and Activities .....	156
5.14.8.1.1	Software Reviews and Analysis.....	156
5.14.8.1.1.1	Test Case and Test Procedure Reviews and Analysis .....	156
5.14.8.1.1.2	Requirements-Based Test Coverage Analysis .....	156
5.14.8.1.1.3	Data Coupling and Control Coupling Analysis .....	156
5.14.8.1.1.4	Structural Coverage Analysis .....	157
5.14.8.1.1.5	Source Code To Object Code Traceability Analysis .....	158
5.14.8.1.1.6	System Verification Review .....	158
5.14.8.2	Software Configuration Management Objectives and Activities.....	158
5.14.8.2.1	Configuration Identification, Baselines and Traceability.....	158
5.14.8.2.2	Configuration Status Accounting .....	160
5.14.8.2.3	Problem Reporting, Tracking and Corrective Action .....	160
5.14.8.2.4	Change Control and Change Review .....	161



5.14.8.3	Software Quality Assurance Objectives and Activities .....	161
5.14.8.3.1	SQA Lifecycle Process Audits .....	161
5.14.8.3.2	Software Test Transition Criteria Satisfaction Review.....	161
5.14.8.3.3	SQA Reporting and Corrective / Preventive Action.....	162
5.14.8.4	Certification Liaison Objectives and Activities .....	162
5.14.8.4.1	Means of Compliance and Requirements .....	162
5.14.8.4.2	Compliance Substantiation.....	162
<b>6.0</b>	<b>SOFTWARE LIFECYCLE DATA .....</b>	<b>163</b>
6.1	Overview .....	163
6.2	Relationship of Lifecycle Data to Other Data Defining the System .....	164
6.3	Trace Data.....	165
6.3.1	<i>Trace Data Objective Evidence of Compliance.....</i>	<i>165</i>
6.4	Software Lifecycle Data to Be Produced and Controlled.....	166
6.5	Relationship of Lifecycle Data to Other Data Defining the System .....	168
6.6	Software Lifecycle Data to be Submitted to Certification Authority .....	169
6.7	Software Control Categories.....	170
6.8	Software Lifecycle Data DER Delegation Plan.....	171
<b>7.0</b>	<b>SCHEDULE.....</b>	<b>172</b>
7.1	Master Project Schedule .....	172
7.2	Stages of Involvement Audit Schedule.....	173
7.3	Certification Authority Web Interface.....	174
7.3.1	<i>Qualtech Compliance Management System .....</i>	<i>175</i>
7.3.1.1	SecureWeb Security Management System .....	176
7.3.1.2	Problem Reporting Management System.....	177
7.3.1.3	Change Impact Analysis Management System.....	178
7.3.1.4	Document Review Management System .....	179
7.3.1.5	Reviews and Analysis Management System.....	180
7.3.1.6	Requirements Traceability Management System .....	181
7.3.1.7	Coverage Analysis Management System .....	182
<b>8.0</b>	<b>ADDITIONAL CONSIDERATIONS .....</b>	<b>183</b>
8.1	Use of Previously Developed Software .....	183
8.2	Tool Qualification .....	186
8.2.1	<i>Criteria 1 Tools.....</i>	<i>186</i>
8.2.1.1	Qualification of Criteria 1 Tools.....	186
8.2.2	<i>Criteria 2 Tools.....</i>	<i>186</i>
8.2.3	<i>Criteria 3 Tools.....</i>	<i>187</i>
8.2.3.1	Qualification of Criteria 3 Tools.....	187
8.3	Alternative Methods .....	187
8.3.1	<i>Product Service History .....</i>	<i>188</i>
8.4	Field Loadable Software.....	189
8.5	Option Selectable Software .....	190
8.6	User Modifiable Software .....	190
8.7	Multiple-Version Dissimilar Software .....	190
8.8	COTS Software.....	190
8.9	Use of Suppliers, Sub-Tier Suppliers and Off-Shore Facilities.....	191
8.9.1	<i>Supplier Identification and Roles.....</i>	<i>192</i>
8.9.1.1	Acme Consultants, Inc.....	192
8.9.1.2	Supplier Competence Questionnaire Example .....	194
8.9.1.3	Supplier Management Plan.....	196



8.10 Deviations and Modifications to Plans ..... 198

### List of Figures

Figure 1-1 Organization Chart.....	15
Figure 2-1 System Level Block Diagram .....	24
Figure 2-2 System Functional Diagram.....	26
Figure 5-1 Relationship Between Processes and Activities.....	81
Figure 5-2 System Lifecycle Flow Diagram.....	82
Figure 5-3 Hardware and Software Lifecycle Diagram .....	83
Figure 5-4 Software Incremental Development Lifecycle Flow .....	84
Figure 5-5 System and Safety Process Feedback Flow Diagram .....	85
Figure 5-6 Lifecycle Process Feedback Flow Diagram .....	86
Figure 7-1 Certification Master Schedule .....	172
Figure 7-2 Certification Authority Web Intereface .....	174
Figure 7-3 Qualtech Compliance Management System .....	175
Figure 7-4 SecureWeb Login Screen .....	176
Figure 7-5 Problem Reporting Management System.....	177
Figure 7-6 Change Impact Analysis Management System.....	178
Figure 7-7 Document Review Management System.....	179
Figure 7-8 Reviews and Analysis Management System.....	180
Figure 7-9 Requirements Traceability Management System .....	181
Figure 7-10 Coverage Analysis Management System .....	182

### List of Tables

Table 1-1 Definitions .....	12
Table 1-2 Part Number and Nomenclature .....	12
Table 1-3 Team Members and Signature Authority .....	14
Table 2-1 System Failure Conditions.....	27
Table 4-1 List of Compliance Documents.....	32
Table 4-2 List of Issue Papers and CRI's.....	32
Table 4-3 Design Assurance Levels.....	36
Table 4-14 Compliance Matrix - Software Conformity Process .....	70
Table 4-15 Certification Authority Involvement Based on DAL .....	71
Table 4-16 Software Certification Experience .....	71
Table 4-17 Software Development Capability .....	71
Table 4-18 Software Service History .....	72
Table 4-19 System and Software Application Complexity.....	72
Table 4-20 FAA DER Capabilities – Todd R. White.....	72
Table 4-21 Level of Involvement Criteria Scoring .....	73
Table 5-1 V-Model Relationship Table .....	75
Table 1-4 Summary of Problem Reporting Methods .....	89
Table 6-1 Lifecycle Data To be Produced .....	166
Table 6-2 Lifecycle Data To Certification Authority.....	169
Table 6-3 Software Control Categories.....	170
Table 6-4 Software DER Delegation Plan .....	171
Table 8-1 Software Development Tools .....	186
Table 8-2 Software Verification Tools.....	187

## 1.0 INTRODUCTION

### 1.1 Purpose

This Plan for Software Aspects of Certification (PSAC) defines the processes, procedures, methods and standards to be used and the lifecycle data to be produced in order to satisfy the objectives of DO-178C and any additional objectives required to satisfy the certification basis of the aircraft. Once approved, this PSAC represents an agreement between the applicant and the customer and/or certification authority.

### 1.2 Scope

This plan will be used by the customer and/or certification authority to determine if the Software Lifecycle Process is commensurate with the rigor required for the level of software being developed. Once approved, it is implemented during the software lifecycle development. This Plan for Software Aspects of Certification complies with the documentation requirements of RTCA/DO-178C, Section 11.1.

### 1.3 Definitions

Definition	Meaning
COTS IC	Any COTS digital or hybrid electronic device which does not execute software in a specific core. COTS ICs may be bus controllers, flip-flop, multiplexers, converters, memories... The hardware functions implemented within these components may be simple or complex. (See also definition of SEH below).
COTS Graphical Processor	Any COTS microcontroller specifically designed for graphical applications. COTS graphical processors for airborne systems are required to have built in mitigation against Hazardous and Misleading information(HMI).
COTS Microcontroller	Any IC which executes software in a specific core area (Central Processing Unit) and implements peripheral hardware elements such as, for example, input/output (I/O), bus controllers... Such a peripheral element may be considered simple (e.g. a UART, A/D, D/A) or complex (e.g. a bus controller). (See also definition of SEH below).
Highly complex COTS microcontroller	Any microcontroller where at least one of the statements below is true: <ul style="list-style-type: none"> <li>- more than one Central Processing Unit (CPU) is embedded and they use the same bus (which is not strictly separated or which uses the same single port memory)</li> <li>- several complex interfaces are dependent on each other and exchange data</li> <li>- several internal busses are integrated and are used in a dynamic way (for example, a dynamic bus switch matrix)</li> </ul>

Definition	Meaning
Integrated Circuit	<p>A circuit (also IC, microcircuit, microchip, silicon chip, or chip) consisting of elements inseparably associated and formed in-situ on or within a single substrate to perform an electronic circuit function. Among the most advanced integrated circuits are the microprocessors or "cores", digital memory chips, ASICS and bus controllers. Integrated circuits can be classified into analog, digital and hybrid signal:</p> <ul style="list-style-type: none"> <li>- Digital integrated circuits are typically microprocessors, DSPs, an micro controllers and work using binary mathematics to process "one" and "zero" signals.</li> <li>- Analog ICs, such as sensors, power management circuits, and operational amplifiers, work by processing continuous signals. They perform functions like amplification, active filtering, demodulation, mixing, etc.</li> <li>- Hybrid ICs can combine analog and digital circuits on a single chip to create functions such as A/D converters and D/A converters.</li> </ul> <p>Digital and Hybrids ICs include ASIC, COTS ICs, highly complex, COTS Microcontroller, Microprocessor, COTS Microcontroller, COTS Graphical Processor, PLD, CEH, SEH.</p>
Microprocessor	<p>A single Central Processing Unit which executes software and does not contain any additional integrated peripheral hardware element such as a UART, A/D, D/A, bus controller, Time Processing Unit, Memory Management Unit, watchdog, etc.</p>

**Table 1-1 Definitions**

1.4 Part Number and Nomenclature

Part Number	Nomenclature

**Table 1-2 Part Number and Nomenclature**

1.5 Special Considerations Cross Reference

<b>Software Consideration</b>	<b>Applicable</b>	<b>Addressed In Doc / Section</b>
General Processor, DSP, etc.	<b>Y/N</b>	
COTS Graphical Processor (CGP)	<b>Y/N</b>	
Product Service History	<b>Y/N</b>	
Legacy System Software	<b>Y/N</b>	
Previously Developed Software	<b>Y/N</b>	
Field Loadable Software	<b>Y/N</b>	
Option-Selectable Software	<b>Y/N</b>	
User-Modifiable Software	<b>Y/N</b>	
Safety Monitoring Software	<b>Y/N</b>	
COTS Software	<b>Y/N</b>	
Public Domain Software	<b>Y/N</b>	
Parameter Data / Databases	<b>Y/N</b>	
Operating System	<b>Y/N</b>	
Compiler Provided Libraries	<b>Y/N</b>	
Board Support Package	<b>Y/N</b>	
Dead, Unreachable or Deactivated Code	<b>Y/N</b>	
DAL Justification	<b>Y/N</b>	
Derived / Safety Requirements	<b>Y/N</b>	
Robustness Requirements / Testing	<b>Y/N</b>	
Structural Coverage Analysis	<b>Y/N</b>	
Data and Control Coupling Analysis	<b>Y/N</b>	
Partitioning (Multi-DAL)	<b>Y/N</b>	
Exhaustive Input Testing	<b>Y/N</b>	
Development Tool Qualification	<b>Y/N</b>	
Verification Tool Qualification	<b>Y/N</b>	
Single Event Upset (SRAM)	<b>Y/N</b>	
Formal Methods	<b>Y/N</b>	
Model Based Development	<b>Y/N</b>	
Object Oriented Technology	<b>Y/N</b>	
Domestic / Off-Shore Supplier	<b>Y/N</b>	
International Cert (EASA, TCCA, etc.)	<b>Y/N</b>	
Open Problem Report Management	<b>Y/N</b>	
Issue Papers / Certification Review Items	<b>Y/N</b>	

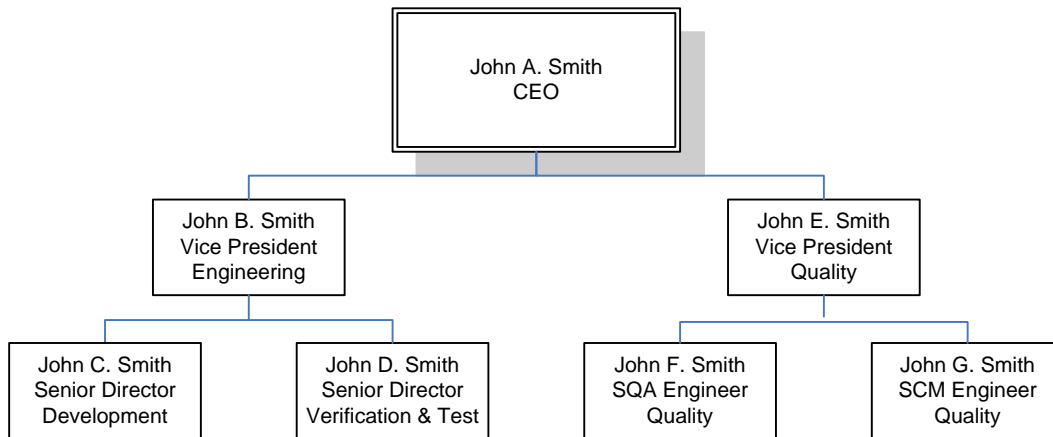
1.6 Team Members and Signature Authority

Name	Title
<b>Signature Authority / PRB &amp; CCB Members:</b>	
John Smith	Systems Engineer
John Smith	Software Lead
John Smith	Independent Verification Engineer
John Smith	Software Quality Engineer
John Smith	Certification Liaison DER (PRB & CCB Only)
<b>Team Members:</b>	
John Smith	Systems Engineer
John Smith	Safety Engineer
John Smith	Software Design Engineer
John Smith	Configuration Management Specialist
John Smith	FAA Software Designated Engineering Representative
<b>Peer Review Members:</b>	
John Smith	Systems Engineer
John Smith	Software Engineer
John Smith	Safety Engineer (mandatory when derived requirements reviewed)
<b>Transition Review Team:</b>	
John Smith	Project Engineer
John Smith	Certification Manager
John Smith	Software Quality Engineer (SQA Records)
John Smith	Systems Engineer
John Smith	Safety Engineer
John Smith	Independent Verification Engineer (V&V Records)
John Smith	Configuration Management Specialist (CM Records)

**Table 1-3 Team Members and Signature Authority**

### 1.6.1 Independent Reporting Structure

The following chart shows that the Quality Assurance Organization is independent from Engineering. It also demonstrates that the verification and test activities are performed independently by someone other than the development engineer.



**Figure 1-1 Organization Chart**

#### 1.6.1.1 Software Quality Assurance Independence

SQA assesses the software life cycle processes and their outputs to obtain assurance that objectives are satisfied, deficiencies are detected, evaluated, tracked, and resolved, and software product and software life cycle data conform to certification requirements. The SQA role is an oversight role for the entire project. As such, SQA does not perform any of the development or verification activities. In addition, the SQA reporting structure is independent of Engineering, Test, Manufacturing, and Project Management.



1.6.1.2 Verification Independence of DO-178C Objectives

The following matrix shows the DO-178C objectives that will be satisfied with independence. In addition, the item being verified and the role of the individual performing the verification is also shown.

<b>Table</b>	<b>Objective</b>	<b>Verification Activity</b>	<b>Item Being Verified</b>	<b>Interpretation</b>
A-3(1)	Software high-level requirements comply with system requirements.	Reviews and Analyses of the High-Level Requirements	High-level requirements	The reviews and analyses of the high-level requirements will be performed by a person(s) other than the developer of the high-level requirements.
A-3(2)	High-level requirements are accurate and consistent.			
A-3(7)	Algorithms are accurate.			
A-4(1)	Low-level requirements comply with high-level requirements.	Reviews and Analyses of the Low-Level Requirements	Low-level requirements	The reviews and analyses of the low-level requirements will be performed by a person(s) other than the developer of the low-level requirements.
A-4(2)	Low-level requirements are accurate and consistent.			
A-4(7)	Algorithms are accurate.			
A-4(8)	Software architecture is compatible with high-level requirements.	Reviews and Analyses of the Software Architecture	Software architecture	The reviews and analyses of the software architecture will be performed by a person(s) other than the developer of the software architecture.
A-4(9)	Software architecture is consistent.			
A-4(13)	Software partitioning integrity is confirmed.			
A-5(1)	Source Code complies with low-level requirements.	Reviews and Analyses of the Source Code	Source Code	The reviews and analyses of the Source Code will be performed by a person(s) other than the developer of the Source Code.

<b>Table</b>	<b>Objective</b>	<b>Verification Activity</b>	<b>Item Being Verified</b>	<b>Interpretation</b>
A-5(2)	Source Code complies with software architecture.			
A-5(6)	Source Code is accurate and consistent.			
A-6(3)	Executable Object Code complies with low-level requirements.	Requirements-Based Testing	Executable Object Code	<p>The person(s) who created a set of low-level requirements-based test cases should not be the same person(s) who developed the associated Source Code from those low-level requirements. It follows that:</p> <ol style="list-style-type: none"> <li>1. The same person(s) could develop the low-level requirements and the Source Code, provided another person(s) develops the test cases from those low-level requirements, or</li> <li>2. The same person(s) could develop the low-level requirements and their associated test cases, provided another person(s) develops the Source Code.</li> </ol>
A-6(4)	Executable Object Code is robust with low-level requirements.			
A-7(1)	Test procedures and expected results are correct.	Reviews and Analyses of the Test Procedures	Test procedures	The reviews and analyses of the test procedures will be performed by a person(s) other than the developer of the test procedures.

<b>Table</b>	<b>Objective</b>	<b>Verification Activity</b>	<b>Item Being Verified</b>	<b>Interpretation</b>
A-7(2)	Test results are correct and discrepancies explained.	Reviews and Analyses of the Test Results	Test results	The reviews and analyses of the test results will be performed by a person(s) other than the person(s) who performed the tests.
A-7(3)	Test coverage of high-level requirements is achieved.	Requirements-Based Test Coverage Analysis	Test cases	The requirements-based test coverage analysis will be performed by a person(s) other than the developer of the test cases.
A-7(4)	Test coverage of low-level requirements is achieved.			
A-7(5)	Test coverage of software structure (modified condition/decision) is achieved.	Structural Coverage Analysis	Test cases, test procedures, and/or test results	The exact independence required depends on how the structural coverage analysis is carried out. For example, if the structural coverage analysis is performed on the test cases, then the structural coverage analysis will be performed by a person(s) other than the developer of the test cases. Similarly, if the structural coverage analysis is performed on the test procedures and test results, then the structural coverage analysis will be performed by a person(s) other than the developer of the test procedures and test results.
A-7(6)	Test coverage of software structure (decision coverage) is achieved.			
A-7(7)	Test coverage of software structure (statement coverage) is achieved.			
A-7(8)	Test coverage of software structure (data coupling and control coupling) is achieved.			

### 1.6.2 Signature Authority

“Signature Authority”, as used herein, is a project-level approval authority that is composed of the following members at minimum:

- Project Manager

- Software Lead
- Independent Verification Engineer
- Software Quality Engineer

Other team members may be required to sign specific documents in addition to this core group, depending on the content of the document. For example, a Configuration Management Specialist is required to sign the Software Configuration Management Plan to acknowledge acceptance of the CM processes defined therein.

### 1.7 Organizational Responsibilities

<b>Project Role</b>	<b>Department</b>	<b>Responsibility</b>
Project Engineer	Program Management	Manage all project activities, including planning, coordination of system life cycle processes, and scheduling of major milestones. Provides liaison with company management.
Software Lead	Software Engineering	Manage all aspects of software development including planning, technical direction, collection and analysis life cycle data, coordination of life cycle processes and scheduling of major milestones. Participate in the software release, change, and baseline process as a required member of the SCCB.
Systems Engineer	Systems Engineering	Develop high level requirements and the high level architecture in accordance with the DO-178B guidelines and engineering procedures.
Safety Engineer	Systems Engineering	Performs the System Safety Assessment (SSA) which determines and categorizes the failure conditions of the system and from that defines safety related requirements. Reviews all requirements changes. Approves derived requirements.
Software Engineer(s)	Software Engineering	Develop software standards, architecture, requirements, design, source code and executable code in accordance with software procedures and DO-178B guidelines. Conduct reviews and analysis of life cycle data. Implement changes during verification testing as required.

<b>Project Role</b>	<b>Department</b>	<b>Responsibility</b>
Independent Verification Engineer(s)	System/Software/Test Engineering	Conduct independent verification including test development, reviews, analysis, and performing tests in accordance with the Software Verification Plan (SVP).
Software Quality Engineer	Software Quality Assurance	Assure that the software life cycle processes produce software that conforms to its requirements by ensuring that project activities are performed in compliance with the SQAP. Conduct Product Reviews, Process Reviews, and Audits throughout the software development life cycle in accordance with checklists provided in Appendix A of the SQAP.
Configuration Management Specialist	Configuration Management	Provide configuration management of all project life cycle data in accordance to the Software Configuration Management Plan. Assure that changes in hardware, software, documentation, and delivered products conform to the Configuration Item (CI) baseline(s). Ensures the integrity of configuration identification of released products. The CM manager or designee is the chair for the SCCB.
Certification Manager	Program Management/ Project Technical Lead	Ensure that the software development and life cycle data meet customer requirements and DO-178B and guidelines

## 1.8 Acronyms and Abbreviations

RAMS	Reviews and Analysis Management System
CAMS	Coverage Analysis Management System
CC1	DO-178C Control Category 1
CC2	DO-178C Control Category 2
CI	Configuration Item
CM	Configuration Management
COTS	Commercial off the Shelf
CPU	Central Processing Unit
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DER	Designated Engineering Representative
DRMS	Document Review Management System
FAA	Federal Aviation Administration
FHA	Functional Hazard Assessment
IVT	Independent Verification Testing
MC/DC	Modified Condition/Decision Coverage
MISRA	Motor Industries Software Reliability Association
MLCP	Master Load Control Procedure
PEMS	Project Event Management System
PRMS	Problem Reporting Management System
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Assessment
PVCS	Serena PVCS Version Control Software
QA	Quality Assurance
RTCA	Radio Technical Commission for Aeronautics
RTMS	Requirements Traceability Management System
SAS	Status Accounting System
SCI	Software Configuration Index
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCS	Software Coding Standard
SDD	Software Design Description
SDS	Software Design Standard
SDP	Software Development Plan
SECI	Software Environment Configuration Index
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SQE	Software Quality Engineer
SRS	Software Requirements Standard
SSA	System Safety Assessment
SVC&P	Software Verification Cases and Procedures
SVCP	Software Verification Cases and Procedures
SVP	Software Verification Plan
SWRD	Software Requirements Document
VR	Verification Results
VSS	Visual Source Safe

## 1.9 Applicable Documents

The following documents are listed for reference only. Each document is applicable to this plan only to the extent specified herein.

### 1.9.1 External Documents

RTCA/DO-178C	Software Considerations in Airborne Systems and Equipment Certification
FAA Order 8110.4C	Type Certification
FAA Order 8110.49	FAA, Software Approval Guidelines
AC 20-115C	Advisory Circular, RTCA Inc., Document DO-178C, Software Considerations in Airborne Systems and Equipment Certification

### 1.9.2 Internal Documents

<Ref Doc>	Plan for Software Aspects of Certification (Ref. DO-178C, 11.1)
<Ref Doc>	Software Development Plan (Ref. DO-178C, 11.2)
<Ref Doc>	Software Verification Plan (Ref. DO-178C, 11.3)
<Ref Doc>	Software Configuration Management Plan (Ref. DO-178C, 11.4)
<Ref Doc>	Software Quality Assurance Plan (Ref. DO-178C, 11.5)
<Ref Doc>	Software Requirements Standards (Ref. DO-178C, 11.6)
<Ref Doc>	Software Design Standards (Ref. DO-178C, 11.7)
<Ref Doc>	Software Code Standards (Ref. DO-178C, 11.8)
<Ref Doc>	Software Requirements Document (Ref. DO-178C, 11.9)
<Ref Doc>	Software Design Description (Ref. DO-178C, 11.10)
<Ref Doc>	Build Procedure for Source Code (Ref. DO-178C, 11.11)
<Ref Doc>	Load Control for Executable Object Code (Ref. DO-178C, 11.12)
<Ref Doc>	Software Verification Cases and Procedures (Ref. DO-178C, 11.13)
<Ref Doc>	Software Verification Results (Ref. DO-178C, 11.14)
<Ref Doc>	Software Environment Configuration Index (Ref. DO-178C, 11.15)
<Ref Doc>	Software Configuration Index (Ref. DO-178C, 11.16)
<Ref Doc>	Software Accomplishment Summary (Ref. DO-178C, 11.20)



## **2.0 SYSTEM OVERVIEW**

This section provides an overview of the system, including a description of the mechanical enclosure and interfaces, electrical functions and their allocation to hardware and software, the architecture, processor(s) used hardware / software interfaces and safety features.

<Example Text>

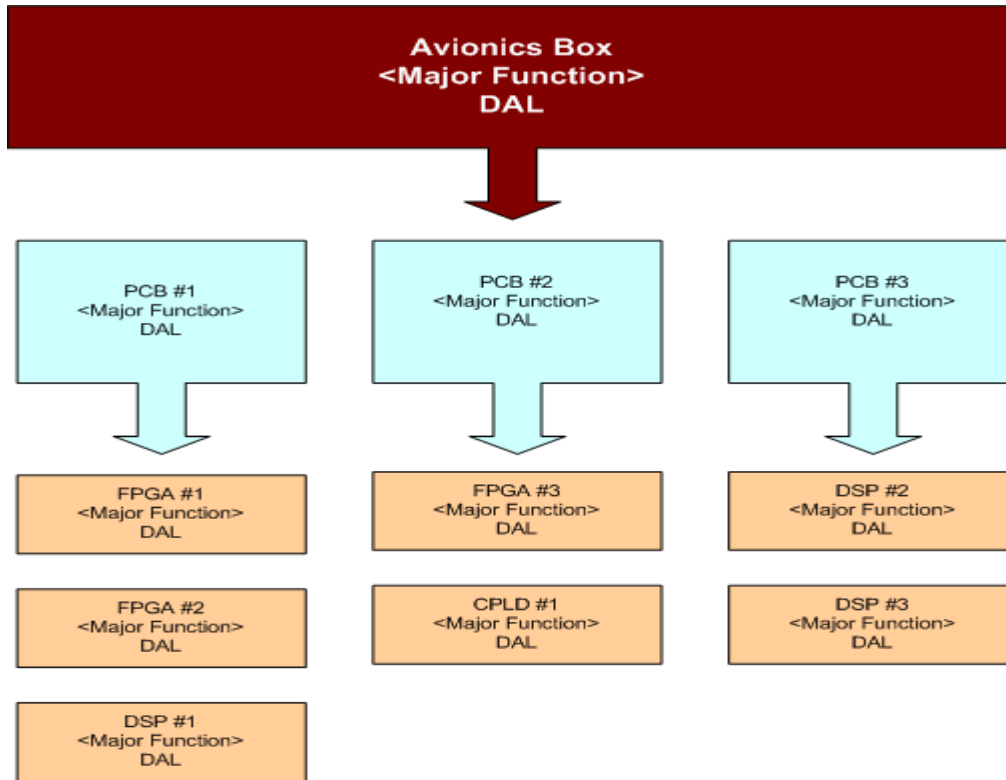
The Avionics Passenger Counter is a module that will keep track of how many passengers are currently in the aircraft/cabin. The current number of passengers in the cabin will be displayed on a display panel in real-time. The system will have a keypad entry so that the flight attendant can enter a passenger headcount correction/adjustment. The passenger headcount and any fault status information collected will be transmitted via ARINC 429 via the PIC processor and ARINC 429 I/O FPGA.

### 2.1 Mechanical Systems Top level Diagram

#### 2.1.1 System Functions Allocated to Mechanical Hardware

## 2.2 Electrical Systems Top Level Block Diagram

<This is a diagram of equipment showing a high-level interconnect of black boxes with emphasis on identifying the Printed Circuit Boards (PCB) at the LRU Level.>



**Figure 2-1 System Level Block Diagram**

2.2.1 System Functions Allocated to Electrical Hardware and Software

2.2.1.1 Data Acquisition Printed Circuit Board #1

The Data Acquisition PCB has four primary functions. They include...

Device	Primary Functions	COTS
FPGA #2	<Primary function implemented in this device>	N
FPGA #2	<Primary function implemented in this device>	N
DSP #1	<Primary function implemented in this device>	N
XYZ #1	<Primary function implemented in this device>	Y

2.2.1.2 Input / Output Printed Circuit Board #2

The Input / Output PCB have five primary functions. They include...

Device	Primary Functions	COTS
CPLD #1	<Primary function implemented in this device>	N
FPGA #1	<Primary function implemented in this device>	N
XYZ #1	<Primary function implemented in this device>	Y
XYZ #2	<Primary function implemented in this device>	Y
XYZ #3	<Primary function implemented in this device>	Y

2.2.1.3 Monitor Printed Circuit Board #3

The Monitor PCB has four primary functions. They include...

Device	Primary Functions	COTS
DSP #1	<Primary function implemented in this device>	N
DSP #2	<Primary function implemented in this device>	N
XYZ #1	<Primary function implemented in this device>	Y
XYZ #2	<Primary function implemented in this device>	Y

### 2.3 System Functional Description

<Example Text>

The passenger headcount function is performed by counting the number of entries and exits. The module uses an Entry/Exit-Sensor to detect passengers' movements in and out of the cabin. The number of passengers in the cabin is tracked by an up/down counter in an FPGA. The passenger load is displayed in real-time on the display panel. The keypad entry can asynchronously load the counter thereby adjusting/correcting the passenger load on the display/system.

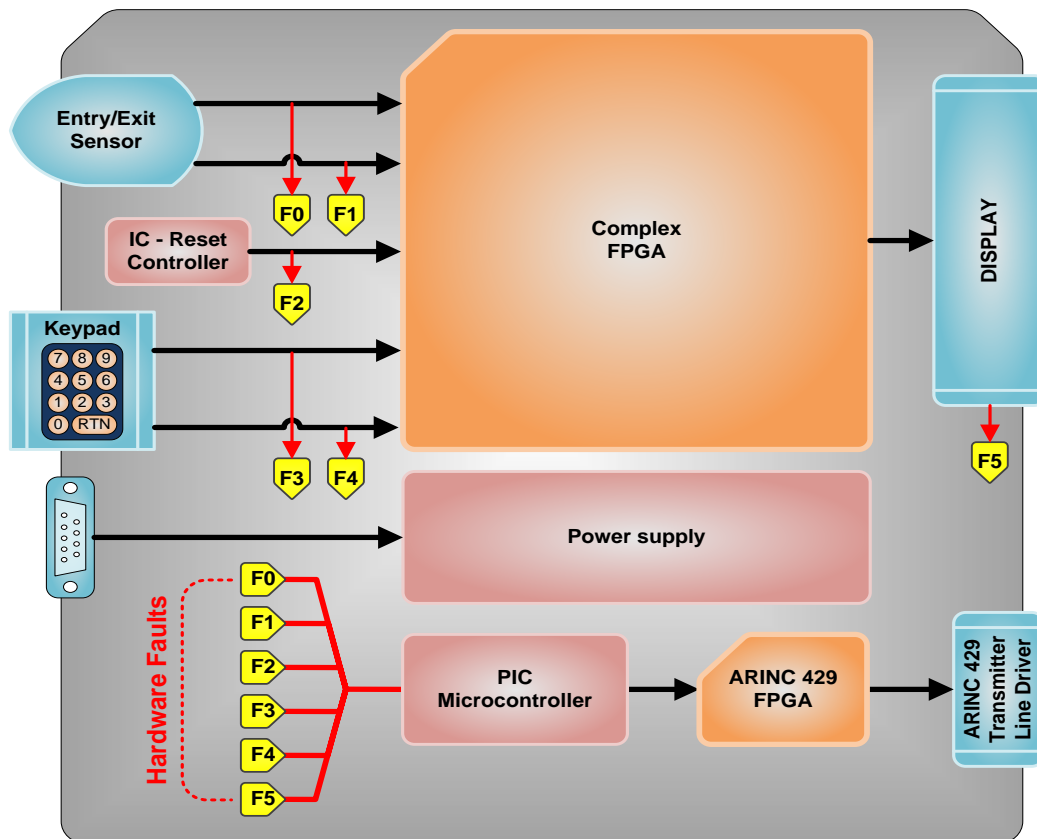


Figure 2-2 System Functional Diagram

2.3.1 System Failure Conditions

2.3.2 High-Level Hardware Functions and Contribution to Potential Failures

No	Function	Example of Potential Failure	DAL
01	Provision of secure and timely data flow to and from applications and Input / output devices, e.g. sensors and actuators	Transmission of deploy command to landing gear in flight	A
02	Controlled access to processing facilities	Omission in schedule of fuel system pumping control partition	B
03	Provision of secure data storage and memory management	Corruption of fuel system execution code by lower integrity partition	B
04	Provision of consistent execution state	Incorrect or inconsistent flight data is loaded into system	B
05	Provision of health monitoring and failure management	Fuel system partition shut down when no error has occurred	A
06	General provision of computing capability	Total loss of IMA platform	A

**Table 2-1 System Failure Conditions**

2.3.3 Safety and Partitioning

The system is composed of diagnostics and other fail-safe mechanisms used to ensure that failures of the system are detected and that the system goes to a safe state if it's unable to perform a safety function.

### **3.0 SOFTWARE OVERVIEW**

This section describes the software functions with emphasis on the proposed safety and partitioning concepts.

#### 3.1 Software Architectural Block Diagram

<Identifies each processor and the tasks to be performed within each>

#### 3.2 Processor #1

##### 3.2.1 States and Modes

##### 3.2.2 Tasks

#### 3.3 Processor #2

##### 3.3.1 States and Modes

##### 3.3.2 Tasks

#### 3.4 COTS Software Identification

##### 3.4.1 Real Time Operating System

<Describe the use of an RTOS, including qualification information>

##### 3.4.2 Board Support Package

<Describe the use of a BSP, including qualification information>

##### 3.4.3 Compiler Provided Libraries

<Describe the use of Libraries, including qualification information>

### 3.5 Deactivated Code Partitioning

How is the operational software protected from the activation of this code?

#### 3.5.1 USB Interface Code

#### 3.5.2 RS-232 Interface Code

#### 3.5.3 Ethernet Interface Code

#### 3.5.4 Boot Load Code Partitioning

The Boot Load Code is partitioned from the operational software and, if activated during flight, the crew will see a fault indication. Anomalous behavior of the Boot Load software, as shown by the system safety assessment, would cause or contribute to a failure of system function with no effect on aircraft operational capability or pilot workload. Software whose anomalous behavior would cause or contribute to a failure of the system function with no effect on aircraft operational capability or pilot workload is identified as **Level E**.



### 3.6 Safety and Partitioning

The system is composed of diagnostics and other fail-safe mechanisms used to ensure that failures of the system are detected and that the system goes to a safe state if it's unable to perform a safety function.

#### 3.6.1 Safety Monitoring

The following safety monitoring rules and techniques will be used:

- Safety monitoring software will be assigned the software level associated with the most severe failure condition category for the monitored function.
- Assessment of the system fault coverage of a monitor will be used to ensure that the monitor's design and implementation are such that the faults which it is intended to detect will be detected under all necessary conditions.
- The monitor and protective mechanism will not be rendered inoperative by the same failure that causes the failure condition.

### 3.7 Resource Sharing <Example Text>

The system is based on Superloop architecture. As such, resource sharing is managed to prevent deadlocks. For example, if the analog board suffers a critical failure with the Dual-Port RAM and locked, in this event, the board would automatically be shipped. This results in no data from the analog board; however, the balance of the system is fully operational.

### 3.8 Redundancy

<Information Here If Applicable>

### 3.9 Fault Tolerance <Example Text>

The software will be designed so that it will continue to operate, possibly at a reduced level, rather than failing completely, when some part of the system fails. During this condition, a visible fault indication will be present.

### 3.10 Timing and Task Scheduling

#### 3.10.1 Timing <Example Text>

The development team understands that numerous types of problems can arise when the software misses its timing constraints. Two types of constraints are recognized: hard timing requirements and soft timing requirements. A hard timing requirement denotes that if a constraint is missed, the program is in error; a late result is just as bad as a wrong result. A soft timing requirement indicates a constraint that the program should obtain on average, or one in which the value of the result degrades with time; occasionally missing a deadline is not considered an error.

The following timing-related problems that will be addressed in the system architecture:

- Priority inversion

- Schedulability and processor utilization
- Interrupt and event response times

It is recognized that such problems may go undetected by verification testing alone, since they only occur under specific timing conditions. As such, code review analysis is combined with testing to aid in uncovering these problems.

### 3.10.2 Task Scheduling <Example Text>

Scheduling is handled as a function of the Superloop architecture. Critical events are handled by hardware interrupts. Timers handle non-critical events.

## 4.0 CERTIFICATION CONSIDERATIONS

### 4.1 Certification Basis and Means of Compliance

The certification basis for the software is Title 14 CFR 2x.1301 and 14 CFR 2x.1309. RTCA/DO-178C has been selected as the means of compliance. The objectives will be satisfied during the lifecycle development and integral processes. The Quality Assurance Engineer will compile applicable artifacts and submit them to the Certification Authority. Verification objectives are satisfied through a combination of reviews and analysis, the development of test cases and procedures, and the subsequent execution of those test procedures. Reviews and analysis provide an assessment of the accuracy, completeness, and verifiability of the requirements, architecture, and design implementation.

The product will be developed in compliance with the following:

Reference / Issue	Description
FAA TSO-	TSO, Description
FAA Order 8110.4C	Type Certification

**Table 4-1 List of Compliance Documents**

### 4.2 Issue Papers and Certification Review Items (CRI)

Document Number	Title
Issue Paper SE-3.3	Guidance for Conducting Change Impact Analysis
Issue Paper SE-3.5	Use of Commercial Off The Shelf Software in Aircraft Avionics Systems
CRI F-10	Management of Open Problem Reports (Software and CEH)

**Table 4-2 List of Issue Papers and CRI's**