ENGINEERING DESIGN PROGRAM:

ENGINE COMMUNICATIONS CARD

By

EDWARD BRUNTON

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree
With Honors in

Electrical and Computer Engineering

THE UNIVERSITY OF ARIZONA

MAY 2019

Approved by:

4/11/19

Dr. Gary Redford
Department of Engineering

ABSTRACT: The “Honeywell Engine Communications Card” provides a common interface between a memory bus and several signals while providing several features. The inputs signals will be transmitted in parallel. The Engine Communications Card will transmit several protocols include CAN, ARINC 429, RS-485/422, MIL-STD-1553, and Ethernet. The Engine Communications Card will have a chassis resilient to vibrations and fluctuating temperature. The Engine Communications Card is modular so that one or more of the input signals may be missing without affecting the operation of the card. Additionally, the Engine Communications Card circuit design is such that if a protocol is removed at board printing time, the Integrated Circuit chip responsible for that protocol can be left off the board as a cost saving measure. During runtime, any number of protocols can be shut off without affecting the performance of the system.

Group Work Division:

Within the “Engine Communications Card” project, I was responsible for writing the final report and for creating the software that would be deployed to the communications card’s MPU. By taking on the responsibility of writing the final report, this allowed the remained of the teammates to reprioritize their responsibilities and work the physical circuit card. Normally, for a senior design project a systems or industrial engineer would take on the responsibility for writing this report, but given our lack of one, I took on this responsibility. In addition to this role, I wrote the software for the project, which would normally take a dedicated individual. Hence, I took on the roll of two people for this project.

ENGR 498A/B Cross Disciplinary Design

Project #18018

Engine Communications Card
Final Report

Revision 4/28/2019 7:21:41 PM

Team Members:
Yousuf Ahmed Alamoudi
Alejandro Alvarez
Samuel Bessette
Edward D Brunton
Hasti Khamsehzadeh
Brian Winkler

Sponsor Contact:
Josh Runkel

Class Mentor:
Mark Brazier

Sponsors:

Honeywell

THE UNIVERSITY OF ARIZONA

Contents

1 Scope and Introduction

1.1 Overview

The “Honeywell Engine Communications Card” provides a common interface between a memory bus and several signals while providing several features. The inputs signals will be transmitted in parallel. The Engine Communications Card will transmit several protocols include CAN, ARINC 429, RS-485/422, MIL-STD-1553, and Ethernet. The Engine Communications Card will have a chassis resilient to vibrations and fluctuating temperature. The Engine Communications Card is modular so that one or more of the input signals may be missing without affecting the operation of the card. Additionally, the Engine Communications Card circuit design is such that if a protocol is removed at board printing time, the Integrated Circuit chip responsible for that protocol can be left off the board as a cost saving measure. During runtime, any number of protocols can be shut off without affecting the performance of the system.

1.2 Project Scope

The Engine Communications Card project includes creating the design of a printed circuit board that handles incoming and outgoing CAN, ARINC 429, RS-485/422, MIL-STD-1553, and Ethernet signals. The Engine Communication Card’s processor ingests all received data and the repeats the received information in a first in first out queue sequence. This design has been printed as a four-layer printed circuit board. The project also includes the design of a chassis which will contain the circuit card, protecting it from environmental hazards such as extreme temperatures and vibrations. Within the scope of the project, this chassis is not physically being created and is not being physically tested. Rather, the circuit card may have thermal shock testing performed while powered off and will have testing performed to determine its ability to echo incoming signals of the designed for communication protocols.

Although considered in the project proposal, the Engine Communications Card does not interface with an external memory module. Instead, the Engine Communications Card has its own internal memory to store communication data before re-transmitting the data.

1.3 Purpose

Currently, no universal communications cards are available. Hence, custom communication cards are required for every engine configuration. When it comes time to obtaining telemetry and other previously recorded information, current solutions lack the ability to aggregate multiple sources of data together. With the Honeywell Engine Communications Card, a single solution will be suitable for most engine configurations allowing for easier serviceability and mass data collection. Additionally, because the design has a modular setup in both the software and the circuitry, unnecessary protocols can be removed during board fabrication without changing the fundamental design. If priorities of protocols change at a later date, the priorities of the protocols can be adjusted with the onboard dip switches to ensure that the most important data is always collected. Based on this prototype design, future designs can be created for additional protocol handling and additional interface options. Additionally, in future designs, a single external memory module can be configured to the board to provide the ability for external data processing.

Through the modular design, this communication card has the potential to save time and money in future development costs. When creating a new communication card, rigorous testing must be performed on the card for EMF and HDI compliance. The current system of creating a new communication card for every engine design means that each card must be tested and verified. With this

engine communications card, the verification tests could be run a single time for all engines. Hence, not only is money saved on only having to design a single communications card, but also in verifying the design for every engine configuration.

1.4 Components of TDP

Within the technical data package which is discussed in the Technical Data Package section, is the Engine Communications Card Assembly, Bill of Materials, Circuit Schematic, Printed Board Layout, Mechanical Drawing, Printed Chassis Assembly Drawing, System Models, Vibration Analysis, Temperature Analysis, Software Design Document, and Engine Communications Card Acceptance Test Procedure (ATP). The Engine Communications Card Assembly Drawing shows how the Printed Circuit Card Assembly connects to the Chassis Assembly drawing. Under the Printed Circuit Card Assembly is the Printed Board Layout and the Printed Wiring Board. A Circuit Schematic is included in the Technical Data Package but has additional information included in the Appendix. The Bill of Materials includes all major components. The Vibration Analysis is of both the chassis alone and the chassis with the Printed Circuit Board. The Thermal Analysis is of the chassis with the Printed Circuit Board. The Software Design Document includes the structure of the included software in a language agnostic form. The source code for the circuit board software is included in the Appendix. The Acceptance Test Procedure includes the tests for the circuit board's thermal resistance and the circuit board's software-hardware performance.

1.5 Changes since CDR

Document 101,000-0, the CDR Report, has undergone several revisions. Throughout these revisions, the mounting requirement and external memory module have been removed. Both the mounting requirement and external memory module require were removed due to the lack of further clarification on interfacing requirements and potential deadline delays they would cause. The Printed Wiring Board (PWB) and Software Package were added to the indentured document list because they were previously overlooked.

Document 100,540, the Latency Model, was updated to account for changes recommended in the team's March 13th meeting. Within this meeting, the MIL STD 1553 specification was discussed and defined to the team as a continuous protocol. Accordingly, the Parallel Model was updated so that the MIL STD 1553 bus was considered to always be in a state of saturation. While this model was update, the other protocols, ARINC 429, ARINC 664, ARINC 825, and RS 485/422, were updated in Table 1 to reflect more consistent updates with smaller packages during each transmission. The conclusion of the model stayed well within the range of passing with a margin of 68%.

Document 100,530, the Parallel Model, experienced an identical update to the update performed in the Parallel Model's Table 1. Similarly, the Latency Model, the passing margin was 77% leading to the conclusion that the new assumptions would not affect the functionality of the system.

Document 101,600, the Software Design Document (SDD), was overhauled to reflect step by step procedures taken within the software package. The SDD also underwent the removal of the external memory module which included the echo mode, or more commonly referred to as a loopback mode, being implemented as the full-time mode of the system. The Software Design Document had the addition of priority options being placed into the priority queue subassembly to allow the user to define which protocols should have the highest priority if the system will not be able to immediately address multiple protocols. Hence, within the scope system, 1.2 System Overview was updated to reflect the removal of the external memory module. 3 Software subassembly CSCI-wide Design Decisions was

modified to reflect the use of the processor's internal memory. Table 1 was modified to include the standard library for the memory interface to allow for a more robust handling of the system's memory. Section 4.3 Interface Design had the addition of specific information regarding which protocols the integrated circuit chips communicate to the processor. Section 5 was completely rewritten due to the removal of the external memory module and the revision of the step by step information of how the protocols are handled within the processor. Figure 2 was updated to show these latest changes. The traceability matrix in Section 6 had external memory requirements removed from it.

Document 101,900, the Acceptance Test Procedure, has also undergone several revisions. The images in the procedure have been updated to better match the circuit board design. Additionally, tests designed for the external memory module have been removed because those tests are no longer relevant to the revised scope of the project. The Acceptance Test Procedure was also updated to consider the safety protocol of the test equipment for the thermal acceptance test procedures.

2 Referenced Documents

Associated Environmental Systems SM-Series Brochure
C2000™ Microcontrollers Development Tools User Guide
Engine Communications Card Preliminary Design Review PDR 11/6/18 Revision
F4T Controller and D4T Data Logger Bases Manual
MAXIM Tutorial First Edition
MAXIM User's Manual Version 2.12
TMS320F28379D LaunchPad Quick Start Guide (Rev. A)

3 System Description and System Block Diagram

3.1 Overview

Within the scope of this Engine Communications Card project, the system requirements are broken down into Chassis Assembly Requirements (keeping in mind the chassis will only be designed, not fabricated), Circuit Board Assembly Requirements, and Software Requirements. As seen in the System Block Diagrams, the Circuit Board assembly interfaces with Engine Communication Signals of various signals (Ethernet, RS 485/422, MIL-STD-1553, ARINC 429, and CAN) and with the I/O Connectors on the Chassis. These I/O Connectors connect to Integrated Circuits to convert the respective signals into Serial Peripheral Interface (SPI) protocols that then interface with the TI processor running the software outlined in the SDD.

3.2 System Block Diagram

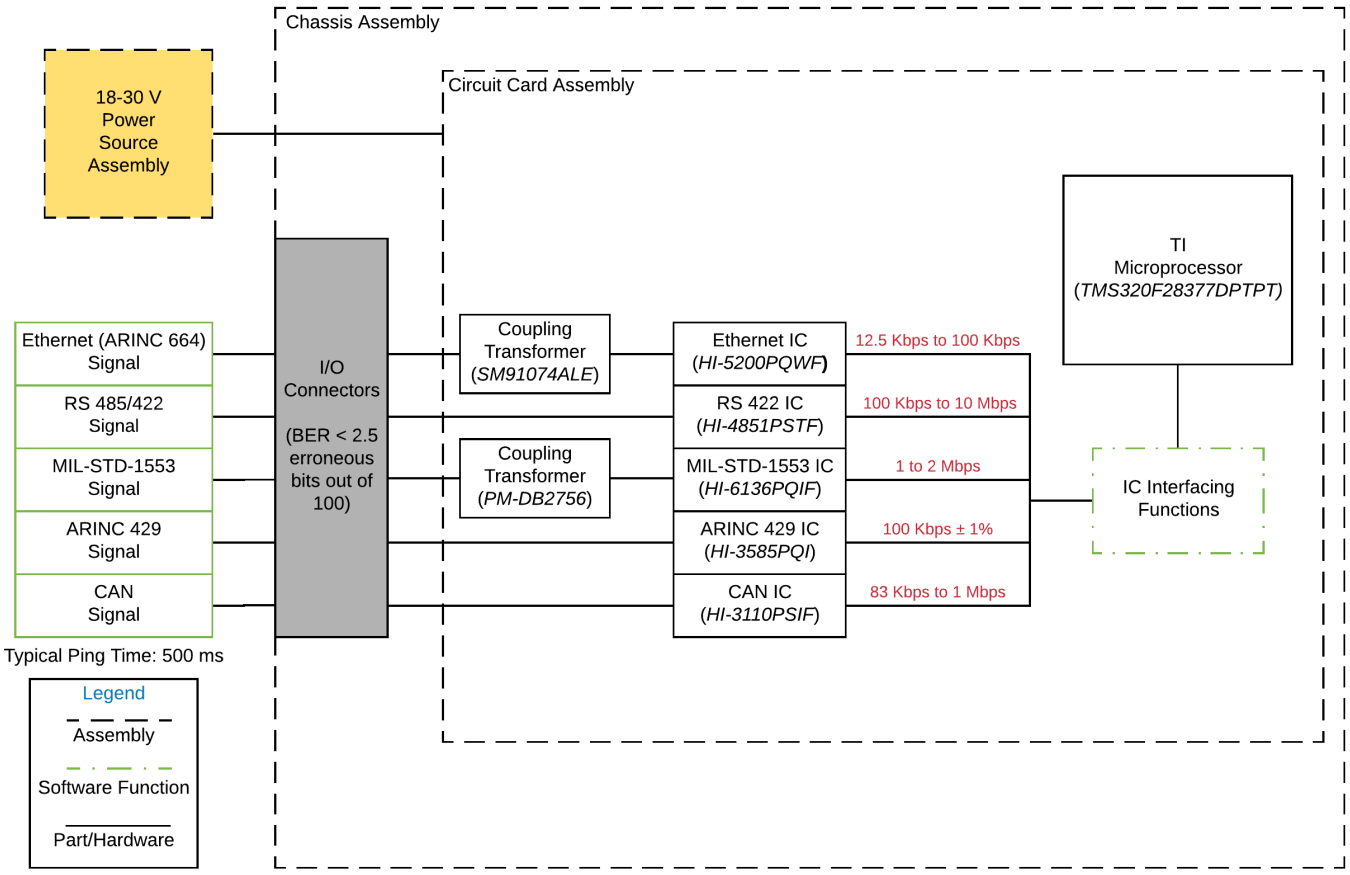

Figure 1 System Block Diagram with Numeric Values and part numbers

3.3 System Block Diagram Description

On the System Block Diagram, the Circuit Card Assembly is nested within the Chassis Assembly. The Chassis Assembly has the five necessary I/O connectors to interface with the clients' inputs and the client's power source. Within the Circuit card Assembly, transformers are in place for the Ethernet and MIL-STD-1553 protocols to ensure that the Integrated Circuits are able to handle the protocols with the least amount of trouble. The Integrated Circuits interface with the TI Microprocessor bidirectionally so that the microprocessor can return the sent information once it is completely in the TI chip's internal memory.

3.4 System Architecture

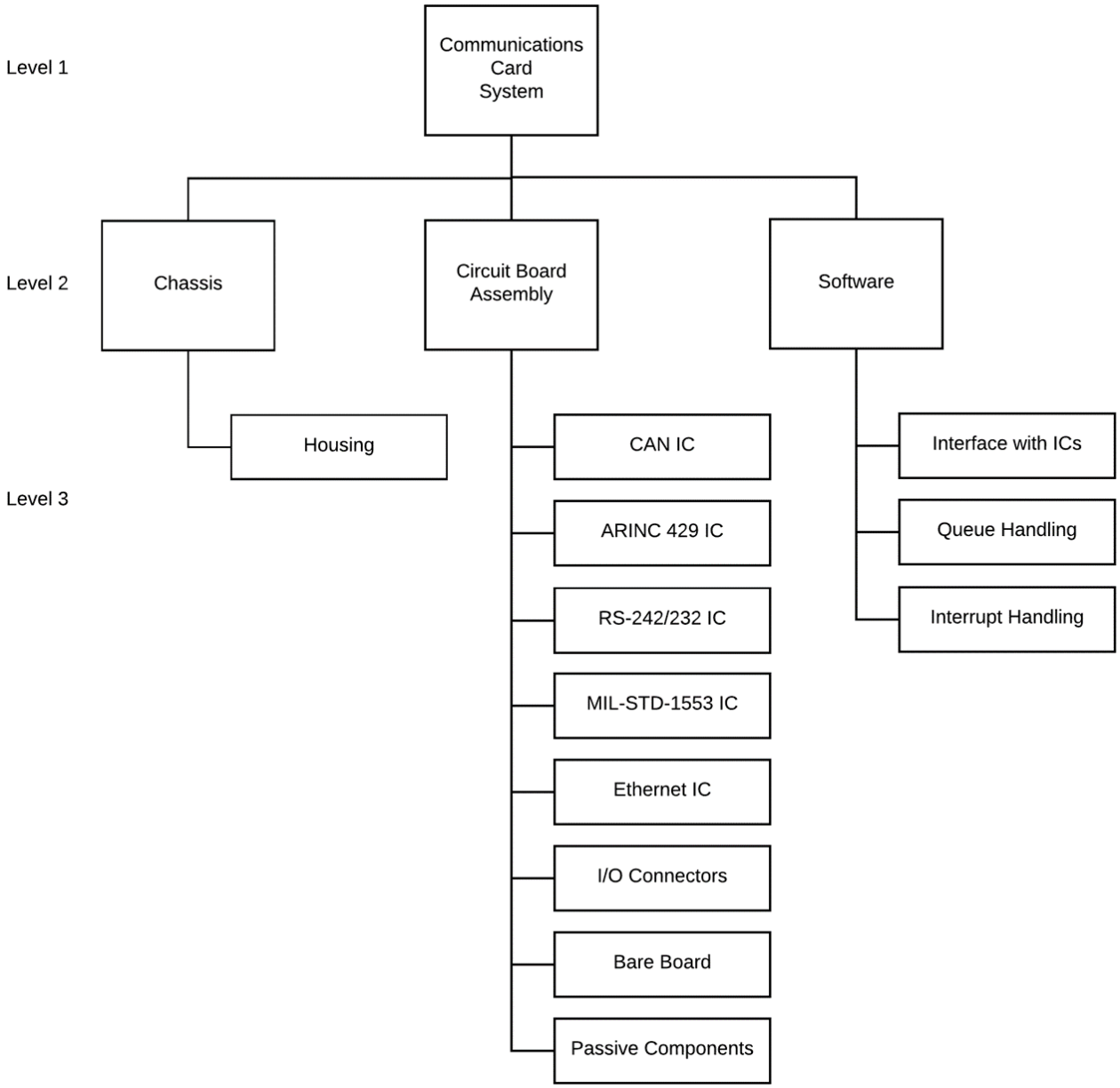

Figure 2 System Architecture

3.5 System Architecture Description

The System Architecture is broken into three primary categories on Level 2: The chassis, the Circuit Board Assembly, and the Software. The chassis includes the physical housing. This physical housing includes mounts to allow the Circuit Board Assembly to be placed within the chassis. However, for the sake of testing and based upon the original project scoped as outlined in the Project Proposal (see Appendix), the chassis is only being created as a design rather than being physically created. The Circuit Board Assembly includes the configurations of the Integrated Circuit chips, ensuring that they are appropriately powered and producing signals according to their respective communication protocols. The Software interfaces with the SPI protocols, SCI protocols and a parallel protocol from these chips and handles the priority queue, which determines which protocol will be processed first.

3.6 Requirements Traceability to Subsystem Architecture

System Requirement	VM	Sub-Assembly					
		Chassis Assembly	VM	Circuit Board Assembly	VM	Software	VM
4.1.2 I/O Connections	D I	2.0 Direct flow down	D I				
4.1.3 Housing	I	3.0 Direct flow down	I				
4.1.4 External Wiring	A D I	4.0 Direct flow down	A D I				
4.1.5 Internal Wiring	A D I	5.0 Direct flow down	A D I				
4.1.6 PCB Thermal Dissipation	T			1.0 Direct flow down	T		
4.1.7 Temperature Durability	A	6.0 Direct flow down	A	2.0 Derived, circuit board shall also withstand -15°C to 70°C temperature ranges.	A		
4.1.8 Vibration Durability	A	7.0 Direct flow down	A	3.0 Derived, circuit board shall also withstand 10-2000 Hz.	A		
4.2.1 Parallel Transmission	T A D			4.0 Derived, all IC cards should remain powered when at least three IC cards are in use.	T	1.0 Direct flow down	T A
4.2.2 ARINC 429	T			5.0 Direct flow down	T		
Rise and Fall Time	T			6.0 Direct flow down	T		
Bit Rate	T			7.0 Direct flow down	T		
Voltage - Transmit	T			8.0 Direct flow down	T		
Voltage - Receive	T			9.0 Direct flow down	T		
Voltage- Null	T			10.0 Direct flow down	T		

System Requirement	VM	Sub-Assembly					
		Chassis Assembly	VM	Circuit Board Assembly	VM	Software	VM
4.2.3 ARINC 664	T			11.0 Direct flow down	T		
Bit Rate	T			12.0 Direct flow down	T		
4.2.4 ARINC 825	T			13.0 Direct flow down	T		
Bit Rate	T			14.0 Direct flow down	T		
4.2.5 MIL-STD-1553	T			15.0 Direct flow down	T		
Rise and Fall Time	T			16.0 Direct flow down	T		
Bit Rate	T			17.0 Direct flow down	T		
Voltage Levels	T			18.0 Direct Flow Down	T		
4.2.6 RS-485/422	T			19.0 Direct flow down	T		
Bit Rate	T			20.0 Direct flow down	T		
Voltage Levels	T			21.0 Direct flow down	T		
4.3.1 HDI Compliance	I			22.0 Derived, the circuit board assembly will be composed of the HDI compliant ICs.	I		
4.3.2 Modularity	D			23.0 Derived, at least three data streams can be physically connected at once.	D	5.0 Derived, the MPU will handle three concurrent data streams.	D
4.4.1 Protocol Translation	T					7.0 Direct flow down	T

System Requirement	VM	Sub-Assembly					
		Chassis Assembly	VM	Circuit Board Assembly	VM	Software	VM
4.4.3 Latency	T A D			26.0 Derived, each IC shall run at less than 90% of the required 15 second latency requirement	D	9.0 Direct flow down	T A
4.4.4 Echo Mode without Memory Interaction	D					10.0 Direct Flow Down	D
4.5.1 Costs	A I			27.0 Derived – Keep costs less than $4000	A I		

Table 1 Subsystem Requirement Traceability Matrix

4 Technical Data Package

4.1 Overview

The Technical Data Package includes the items listed in the Indentured Document List, which were discussed in the section 1.4. Full copies of every document is included in the Appendix.

4.2 Indentured Document List

Part/Document Number			**Drawing/Document/Analysis**
101,000			Engine Communications Card Assembly
	101,000-0		Critical Design Review Report
	100,010		Circuit Schematic
	100,020		Printed Wiring Board
	100,030		Printed Board Assembly Drawing
	100,031		Chassis Drawing
	101,500		System Models
		100,501	Cost Model
		100,505	Conductor Model
		100,510	Vibration Model
		100,520	Temperature Model
		100,530	Parallel Model
		100,540	Latency Model
	101,600		Software Design Document

Part/Document Number			Drawing/Document/Analysis
	101,700		Software Package (Zip File)
	101,900		Engine Communications Card ATP

Table 2 Indentured Document List

4.3 Verification Table

System Requirement Verification				
Requirement	**Method**	**Limit/ Reference**	**Measured/Ref Value**	**Pass/ Fail Margin**
1. Chassis Requirements:	-	-	-	-
a. I/O Connectors: The chassis will have I/O connectors in accordance with military standards for each the MIL-STD-1553 connector, the ARINC 429 connector, the MIL-STD-1553 connector, and an ethernet connector.	D	>= 5 operational connections	1 Operational Connection	4 Connections Failed
b. PCB Thermal Dissipation: The system PCB shall withstand 300 cycles within the environmental temperature range of -15°C to 70°C.	T	>= 300 cycles	Removed in ECR	
c. Temperature Durability: Circuit Board-Chassis Assembly shall withstand the environmental temperature range of -15°C to 70°C.	A	< 125 Celsius	114 Celsius	11 Celsius
d. Vibration Durability: The circuit board-chassis assembly shall function within the critical modal frequencies.	A	< 0.47-inch circuit card displacement for ten frequencies	6/10 frequencies within limit	4 Frequencies Fail
2. Hardware Requirements	-	-	-	-
a. Parallel Transmission: The board shall transmit at least three protocols at once without putting a load more than 60% on the processor.	T	< 2 % error	Only one connection established	Indeterminate
	A	< 60 %	13.4%	46.6%

System Requirement Verification				
Requirement	**Method**	**Limit/ Reference**	**Measured/Ref Value**	**Pass/ Fail Margin**
	D	<2% error (34<1729 priority failures)	3/1728 failures = 0.17% error	1.82%
b. ARINC 429: The system shall communicate over the ARINC 429 protocol and pass the following standards: Rise and Fall Time equal to 1.5 µs ± 0.5µs., Bit Rate equal to 100 kbps ± 1%, Voltage – Transmit equal to -10.0V ± 1.0V to +10.0V ± 1.0V., Voltage – Receive equal to -6.5V to -13V to +6.5V to +13V., and Voltage – Null equal to V ± 0.5V.	T	100 kbps ± 1%, -10.0V ± 1.0V to +10.0V ± 1.0V, -6.5V to -13V to +6.5V to +13V, V ± 0.5V	100 kbps	Pass
c. ARINC 664: (Ethernet): The system shall communicate over the ARINC 664 (Ethernet) and pass the following standards: Bit rate equal to 12.5 kbs to 1 mbps.	T	12.5 kbs to 1 mbps	No connection established	Indeterminate
d. ARINC 825: The system shall communicate over the ARINC 825 (CAN) and pass the following standards: Bit Rate equal to 83 kbps to 1 mbps.	T	83 kbps to 1 mbps	No connection established	Indeterminate
e. MIL-STD-1553: The system shall communicate over the MIL-STD-1553 and pass the following standards: Rise and Fall Time equal to 100 to 300 ns, Bit Rate equal to 1.0 mbps ± 1%, and Voltage Levels equal to 18.0V to 27.0V.	T	100 to 300 ns, 1.0 mbps ± 1%, 18.0V to 27.0V	No connection established	Indeterminate
f. RS-485/422: The system shall communicate over the RS-485/422 and pass the following standards: Bit Rate equal to 100 kbps to 10 Mbps, and Voltage Levels equal to -7V to +12V.	T	100 kbps to 10 Mbps -7V to +12V	No connection established	Indeterminate
3. Circuit Board Requirements	-	-	-	-
a. HDI Compliance: The communications card shall be designed featuring HDI	I	Data Sheet	Data Sheet	Pass

System Requirement Verification				
Requirement	**Method**	**Limit/ Reference**	**Measured/Ref Value**	**Pass/ Fail Margin**
components to save weight and space.				
b. Modularity: The hardware design shall be modular, designed operate normally if protocol components are not populated on PCB.	D	1,2,3 connections	1 Connection Operated	Indeterminate results for 2, 3 Connectors
c. Memory Map: The board shall receive information and update the information within an internal memory map with less than a 2% error rate.	T	< 2% error	0 errors	Pass
	D	Memory Map receives communication from board	RS-422 Operated	Pass
4. Software Requirements	-	-	-	-
a. Translation of Protocols: The Engine Communication Card should be able to write and read to ARINC 429, ARINC 664, ARINC 825, MIL-STD-1553, and RS-485/422 data streams (using a transform function in the software) with less than a 2% error rate.	T	< 2% error	0 errors	Pass
b. Latency: The Engine Communications Card shall write to the memory module interface with a latency no longer than fifteen seconds.	T	< 15 seconds	48.024 nanoseconds	Pass – 15 seconds margin
	A	< 15 seconds	7.4 seconds	Pass
	D	Memory Module Interface Does Not Timeout	Software Stability after 120 minutes of continuous run	Pass
c. Echo Mode without Memory Interaction: For conducting tests that require an echo mode without memory interaction, the Engine Communications card should have an echo function that repeats, or echoes, the message received from	D	0 memory leaks	0 memory leaks	Pass

System Requirement Verification				
Requirement	**Method**	**Limit/ Reference**	**Measured/Ref Value**	**Pass/ Fail Margin**
a protocol back to the source directly.				
5. Cost Requirements	-	-	-	-
a. Costs: All parts shall cost less than $4000.	A	< $4000	$3465.78	Pass
	I	< $4000	$3153.34	Pass

Table 3 Verification Table

4.4 Document 101,000 – Engine Communications Card Assembly

The Engine Communications Card Assembly shows how the circuit card assembly and the chassis assembly come together in a single assembly. The complete drawing is available in the Appendix Document 101,000.

4.5 Document 101,000-0 Critical Design Review Report

The Critical Design Review Report (CDR) is included in the Appendix to this document. This report was onto revision (3) at the time of writing. The Critical Design Review Report summarizes the final scope and requirements of the project. Additionally, this report described a brief summary of the acceptance test procedures, the project schedule, and risks that could arise in the project.

4.6 Document 101,010 – Circuit Schematic

The Circuit Schematic, created with EagleCAD, shows how the various components of the board connect to each other electrically. It has purposely been designed in sections so that in the future, the client can elect not to print a section if the client knows that the protocol on a particular printed board will never be used. The complete Circuit Schematic is available in the Appendix Document 101,010. Zoomed in components are shown below.

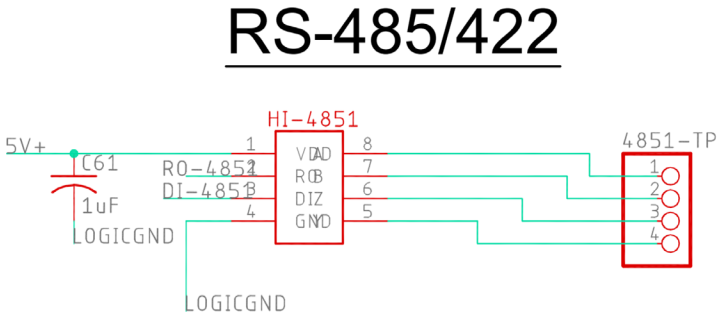

Figure 3 RS-482/422 Circuit

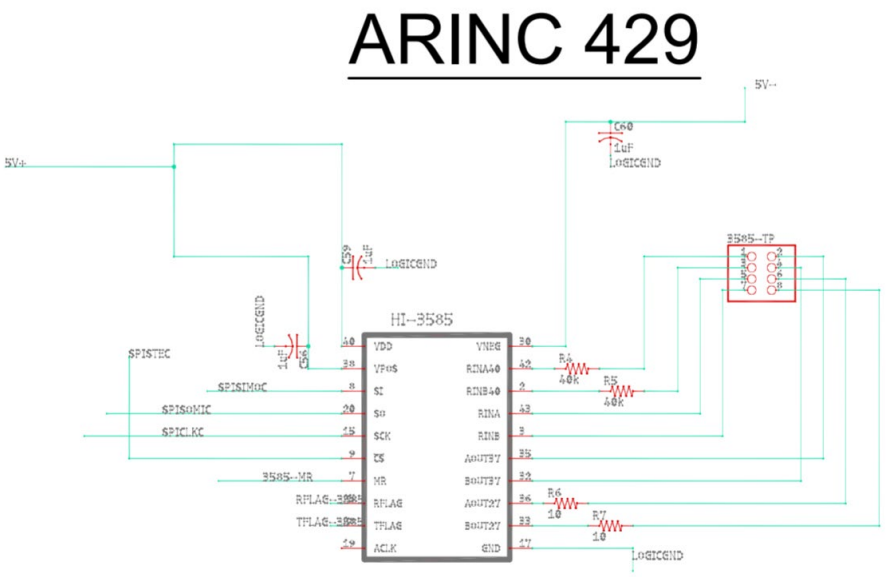

Figure 4 ARINC 429 Circuit

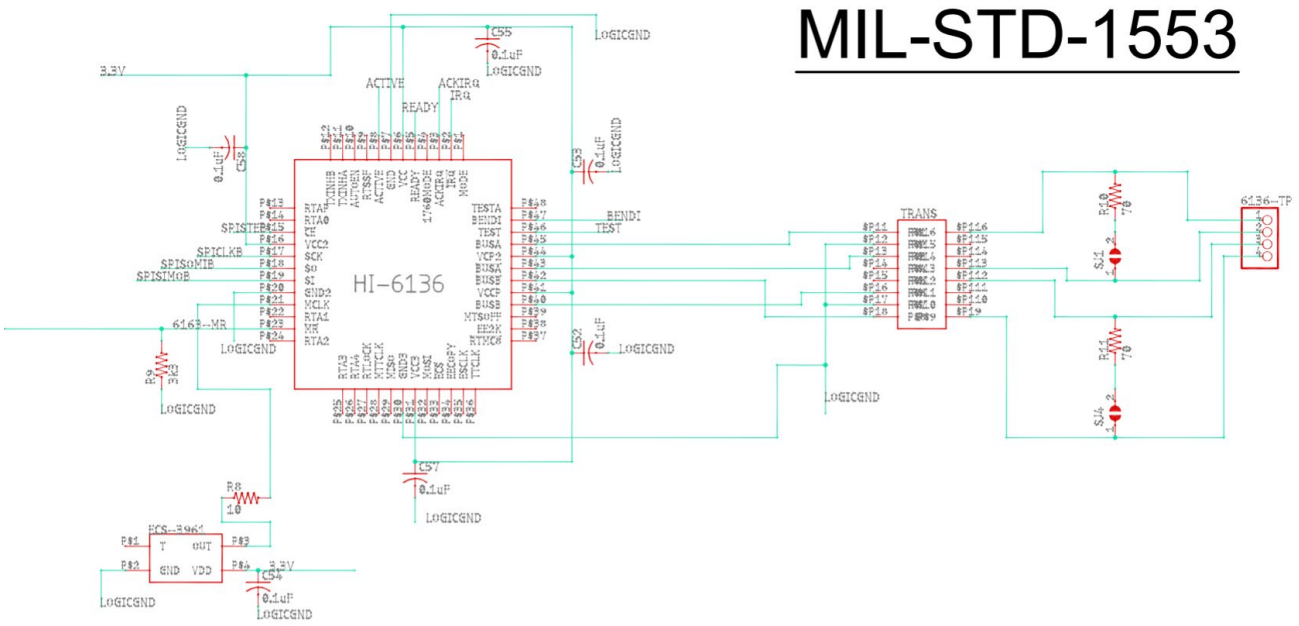

Figure 5 MIL-STD-1553 Circuit

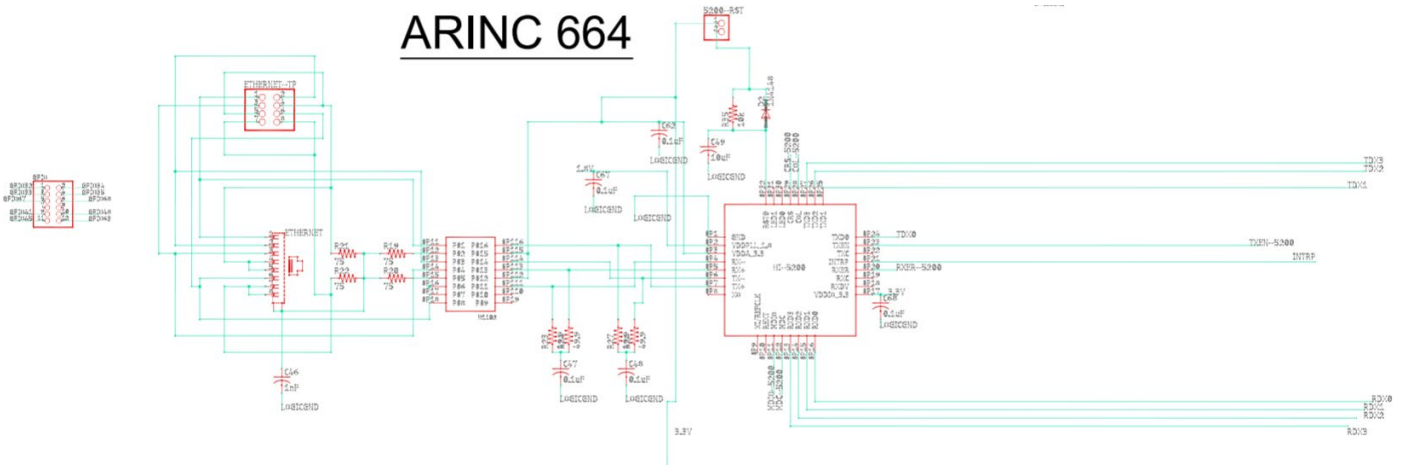

Figure 6 ARINC 664 Circuit

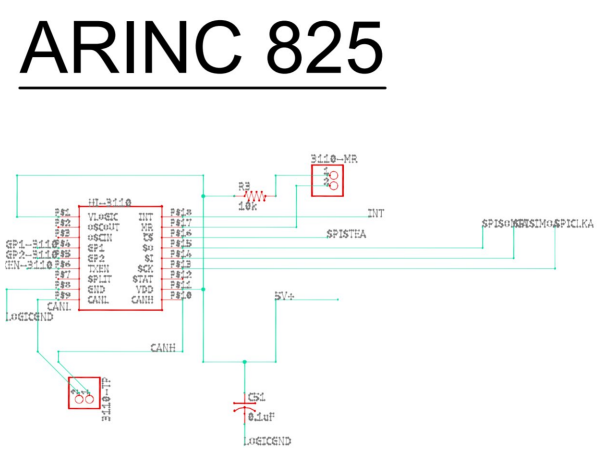

Figure 7 ARINC 825 Circuit

4.7 Document 101,020 – Printed Wiring Board Drawing

The printed board layout, also known as the printed wiring board, has the Holt Integrated Circuit chips and the TI Processor on them. In order to ensure that each of these chips has appropriate power delivered to them several resistors, inductors and capacitors have been added to the board. A complete view of the printed board layout is available in the Appendix Document 101,020.

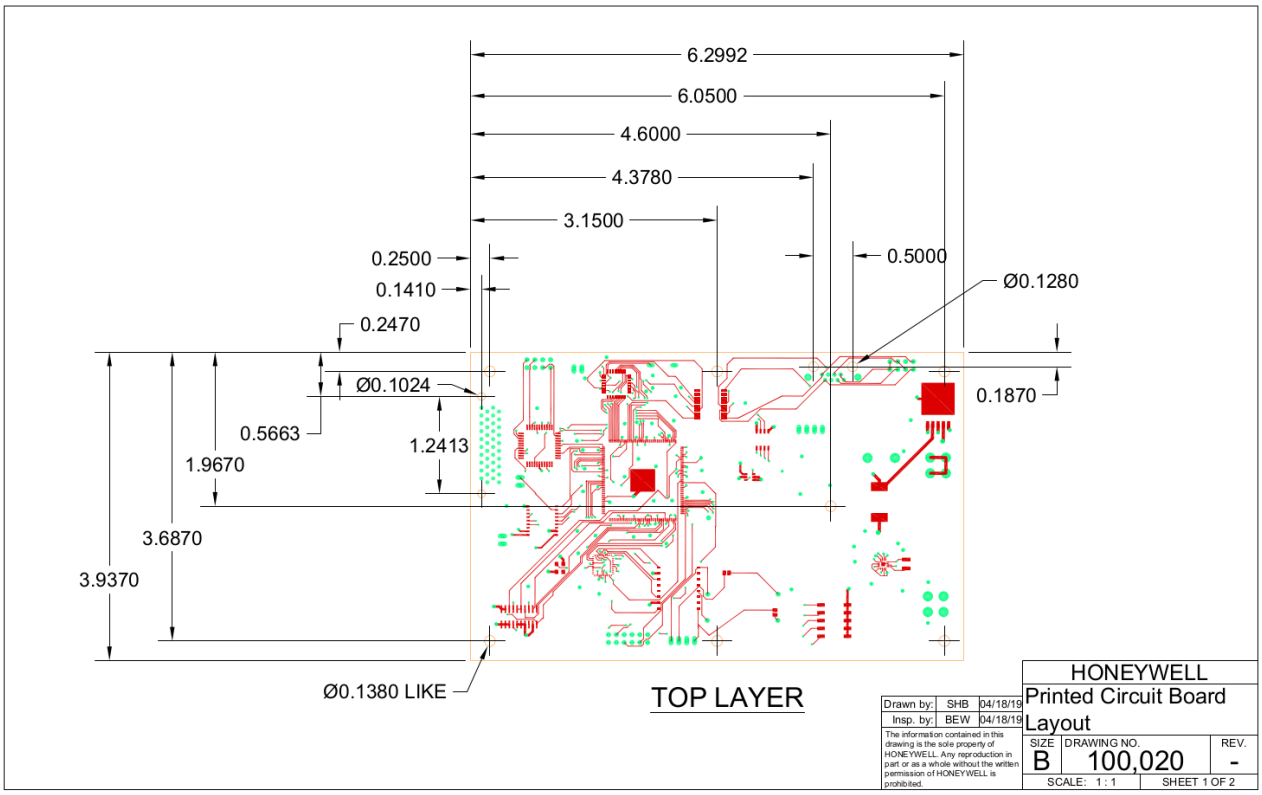

Figure 8 Printed Board Layout Overview

4.8 Document 101,030 – Printed Board Assembly Drawing

The Printed Board Assembly Drawing is available as Document 101,030 in the Appendix. The Printed Board Assembly document includes the Bill of Materials.

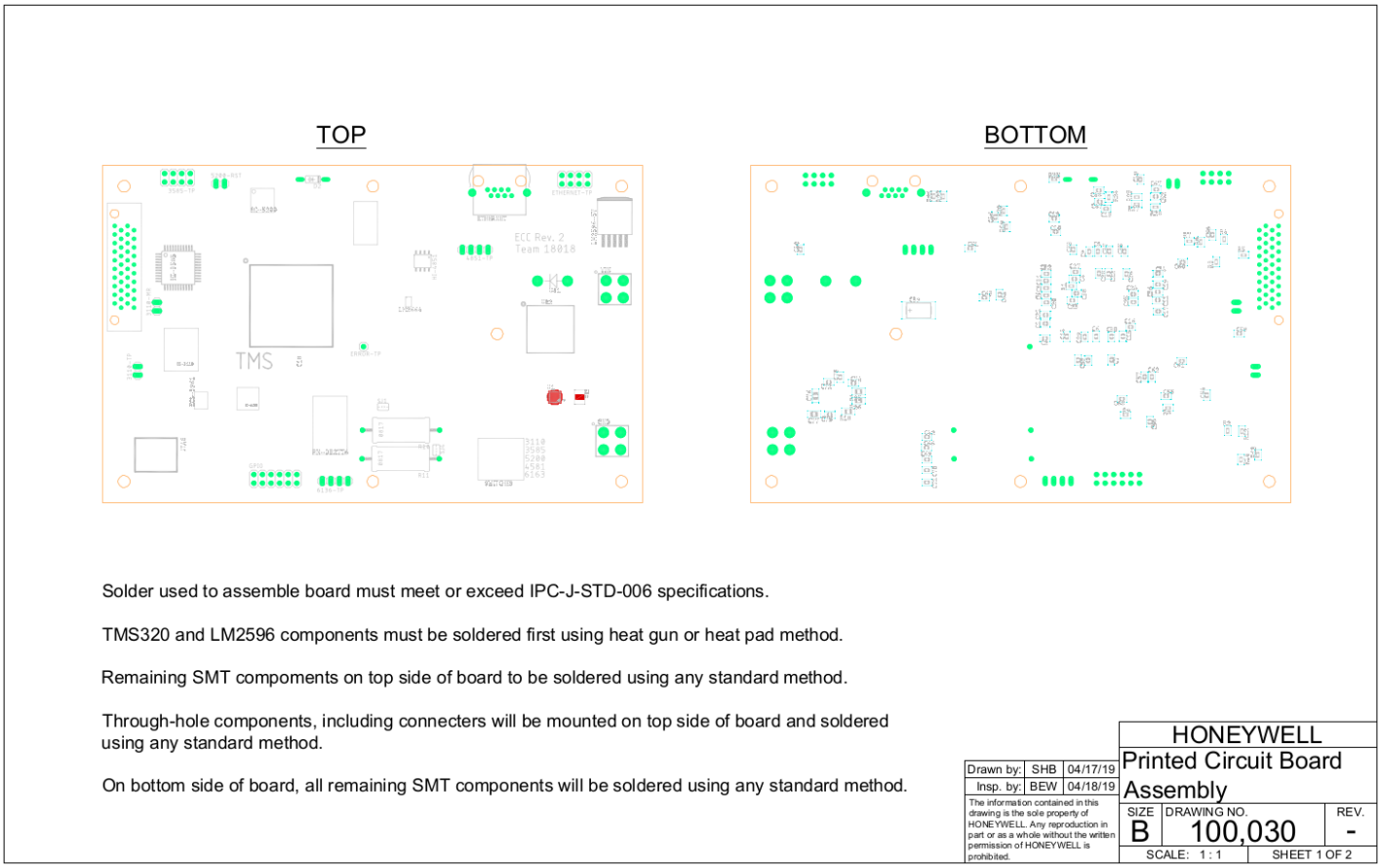

Figure 9 Printed Circuit Board Assembly Overview

4.9 Document 101,031 – Chassis Drawing

The chassis has a 1.700 inch by 5.000 inch by 7.000 inch. All of these dimensions have a tolerance of ± 0.005 inches. The material throughout the chassis is AL-6061 with a thickness of 0.25”. The complete Chassis Drawing is in the appendix as Document 101,031.

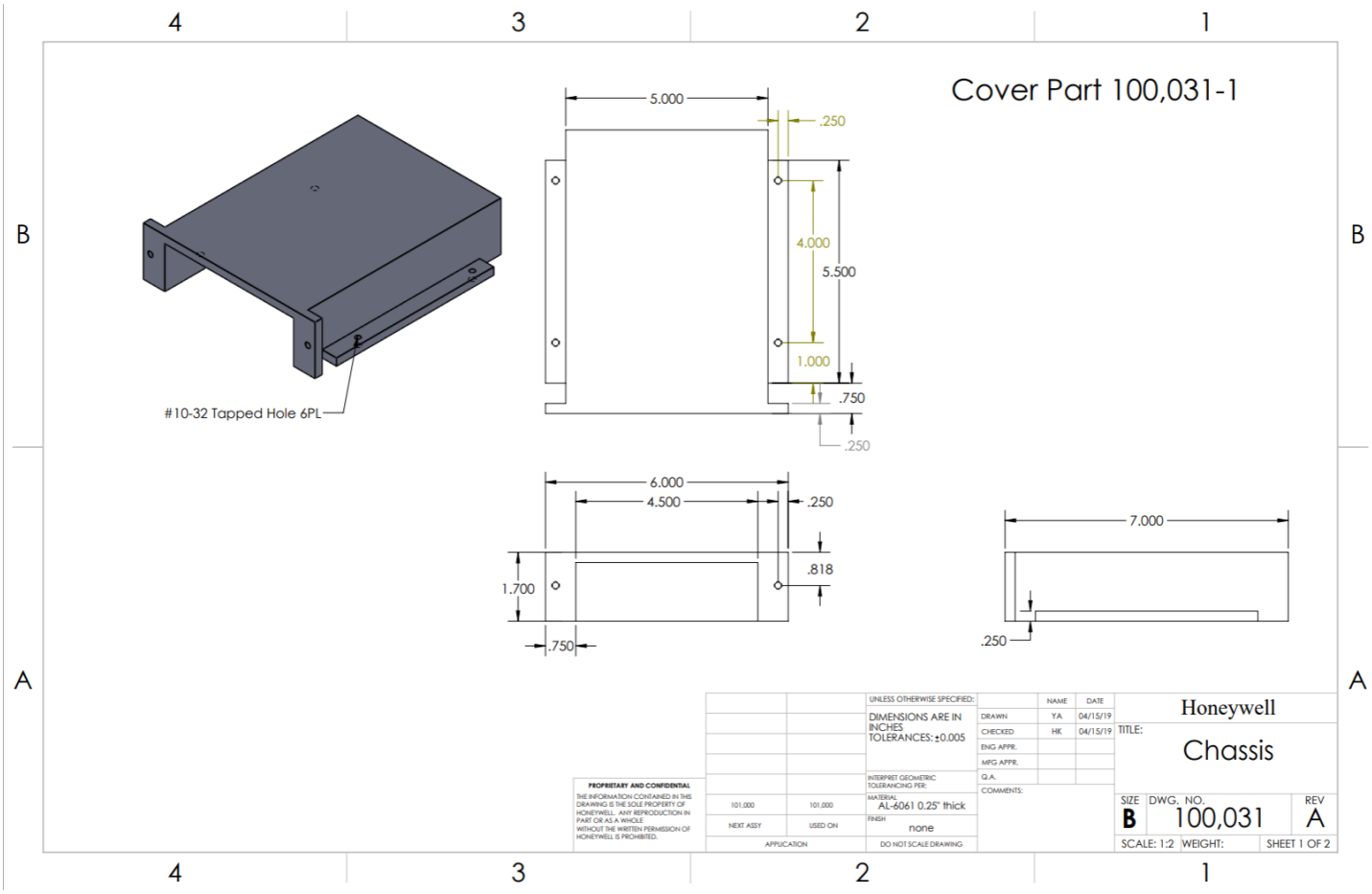

Figure 10 Chassis Drawing Overview

4.10 Document 101,500 – System Models

The System Models include the Cost Model, Conductor Model, Vibration Model, Temperature Model, Parallel Model, and Latency Model. Document 101,500 is the collection of Document 101,501, Document 101,505, Document 101,510, Document 101,520, Document 101,530, and Document 101,530. All of these documents are included in the Appendix.

4.11 Document 101,501 – Cost Model

The Cost Model (as shown in the Appendix Document 100,501) considers the costs from the Bill of Materials, which includes the cost of the Integrated Circuit (IC) Components, the Microcontroller, Design Day (such as shirts and poster boards) and Fabrication. From this model, the potential cost range of the project as a whole is obtained. From this model, the conclusion could be drawn that the $4,000 budget from requirement 4.5.1 – Costs was met.

4.12 Document 101,505 – Conductor Model

The Conductor Model (as shown in Appendix Document 100,505) considers the component (e.g. the Integrated Circuit cards) operating voltages and current requirements to find the necessary wire ratings, traces sizes and trace spacings. The Conductor Model provided information to the circuit card designers regarding the required power supplies in terms of peak voltage and peak amperage needed.

The power circuitry, under worst case scenarios, draws the greatest amperage in the 3.3V bus, which needs nearly 1A of current draw during operation. The next highest amperage requirement comes from the 1.2V bus, which requires nearly .5A. The remaining busses require less than 30mA, even under

peak load. Power sequencing is required by one of the parts, so a resistive element may be used to enable the 3.3V bus using the signal generated from the 5V bus.

4.13 Document 101,510 – Vibration Model

The Vibration Model (as shown in the Appendix Document 100,510) takes customer provided environmental vibration frequencies and outputs the vibration frequencies for the critical area within the circuit board - chassis assembly. This document includes an analysis of the viability of the Engine Communications Card assembly at ten critical frequencies of the assembly.

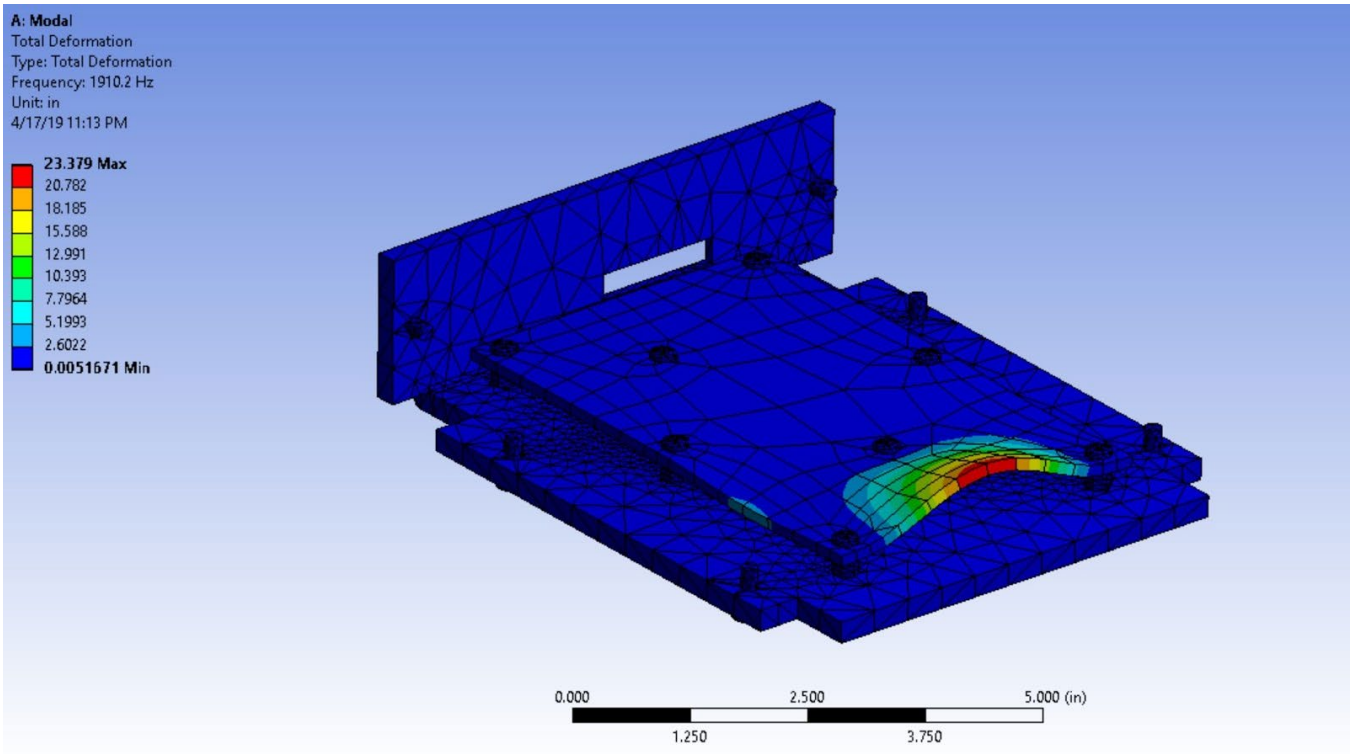

Figure 11 1910.2 Hz Vibration Simulation from Vibration Model

4.14 Document 101,520 – Temperature Model

The Temperature Model (as shown in the Appendix Document 101,520) considers the Environmental Temperature experienced by the circuit card and the power usage of the on-board Integrated Circuits. With the power usage of Integrated Circuits, a temperature range for those cards is derived. Using that temperature range and the environmental temperature range, the overall temperature range for the circuit board is generated.

Part	Ta= -15 C	Ta=70 C	Pwr (W)	θ (C/W)	Tj @ -15C	Tj @ 70C
CAN IC	-15	70	0.048	~50	-12.6	**72.4**
ARINC 429 IC	-15	70	0.05	~50	-12.5	**72.5**
RS-242/232 IC	-15	70	0.00825	~50	-14.5875	**70.4125**
ML-STD-1553 IC	-15	70	0.61	~45	12.45	**97.45**
ETHERNET IC	-15	70	0.18	~45	-6.9	**78.1**
TMs320F2835xD	-15	70	2.1	21	29.1	**114.1**
LT1761MPS5/1.8	-15	70	0.07	135	-5.55	**79.45**
TPS5410-EP-1	-15	70	0.293	75	6.975	**91.975**
TPS5410-EP-2	-15	70	0.066	75	-10.05	**74.95**
TPS5410-EP-3	-15	70	0.016	75	-13.8	**71.2**

Table 4 Temperature Model Summary Table

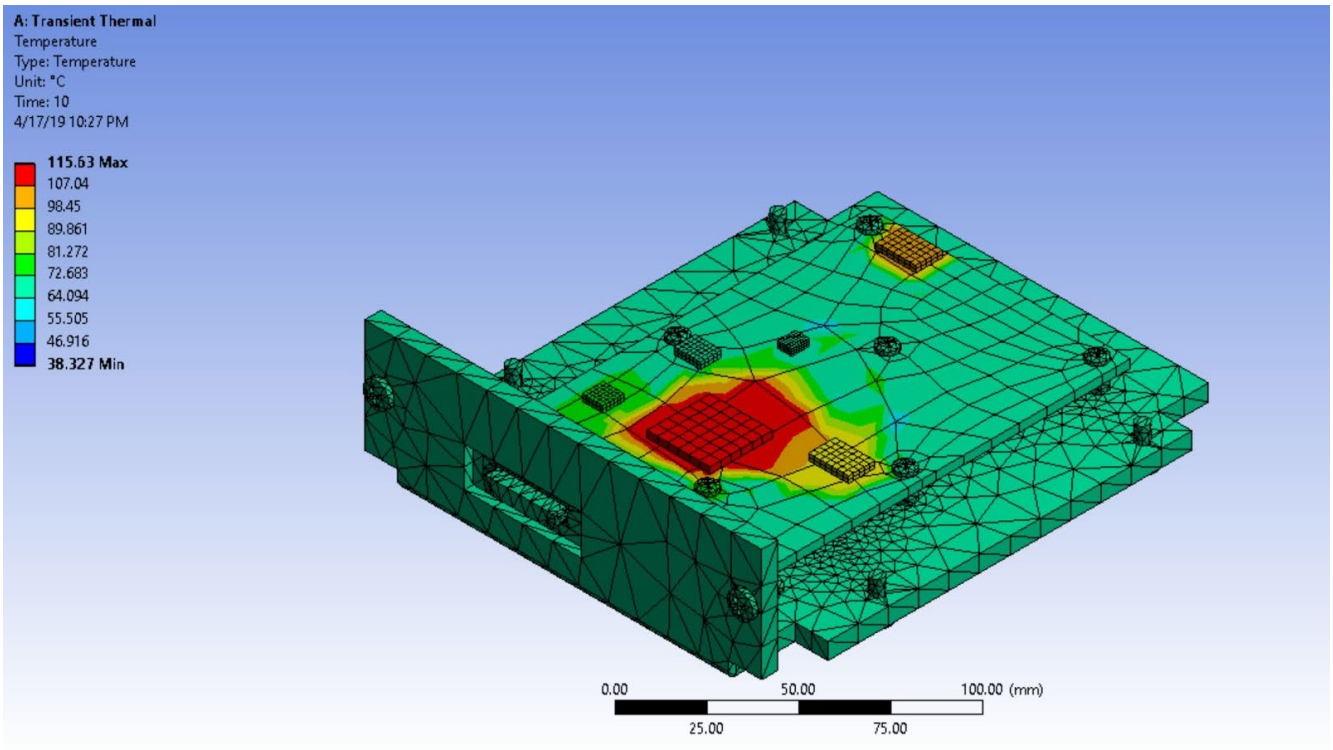

Figure 12 Ansys temperature model at 70 degrees Celsius

As shown in Table 3 Temperature Model, the calculated temperatures are within a safe range and are not reaching the maximum temperature of 120° C. When the card's temperature reaches $70°C$, the peak juncture temperate is the TMs320F2877D chip; the steady state temperature is $114.1°C$. In ANSYS, the peak temperature was calculated to be $114.32°C$. Hence, both the hand calculated temperature values and the ANSYS models pass the temperature model.

4.15 Document 101,530 – Parallel Model

The Parallel Model (as shown in the Appendix Document 100,530) takes in Data Transmission Frequencies and Package Sizes (ARINC 429, ARINC 664, ARINC 825, MIL STD 1553, RS 485/422) to calculate the processor's ability to handle multiple signals at once. The numerical representation of this ability to Peak Processor Load, or in other words, the maximum portion the processor's cycles that will be utilized.

The ARINC 429, ARINC 825, and MIL STD 1553 protocols have available IC chips that will allow for buffering to assist with throughput handling. The peak processor load is desired to be less than 60% (see PDR requirement 4.2.1) as it only takes up a maximum of 13.4% of the processor's available load, providing a 77.6% model margin. As seen by the tables within the Parallel Model document, this analysis passed the requirement. Accordingly, the used TI processor (see Bill of Materials, Document 101-001) meets the parallel model passing requirements.

4.16 Document 101,540 – Latency Model

The Latency Model (as shown in document 100,540) takes in the Data Transmission Frequencies and Package Sizes (ARINC 429, ARINC 664, ARINC 825, MIL STD 1553, RS 485/422) to calculate the system's latency from the critical path of the circuit. In this model, the critical path is the maximum

amount of time from receiving a data transmission from any one of the data I/O's, to outputting a response to that same data I/O. The path will pass through the processor and the customer's memory module. For more information on critical path tracing in circuits, read "A critical path tracing algorithm for synchronous sequential circuits" by Menon, Levendel, and Abramovici in the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Hence, latency is determined by the critical path.

Protocol	Total Delay (seconds)
ARINC 429*	4.202089762
ARINC 664	1.401601212
ARINC 825*	1.488648923
MIL STD 1553*	1.015641212
RS 485/422	1.456801212
Critical Path Latency	**4.202089762**

Table 5 Total Input Delays by Protocol

The total system latency is 4.202 seconds. This is well below the required 15 seconds or less latency of the PDR's 4.4.3 system requirement, All the ICs fell below the 13.5 seconds or less derived latency requirement. Hence, there is a 68.87% margin, so the project is slated to meet the latency requirements with ease.

4.17 Document 101,600 – Software Design Document

This document, the Appendix Document 101,600, serves to provide an overview of the system software design for the Engine Communications Card. It defines the software modules needed to implement each communication protocol as well as to interface with the on-board memory.

The software system includes handling interactions with the memory interface and data inputs. The IC firmware will not be interacted with and shall be treated as a black box purely for accepting inputs from and providing outputs to in terms of software consideration.

This document provides the Computer Software Components (CSCs) and Computer Software Configuration Item (CSCIs) that the software is designed on.

Throughout the CSCI, code is written in the language C, minimizing the required number of libraries. Local memory resources are of low priority to our design. Meanwhile processor clock cycles is critical in the design. Hence, Queue and Interrupt handling is its own CSC so that a robust and reusable solution can be created that will prevent the latency requirement from being missed. In a general sense, communications from signals lacking an integrated circuit card with a memory buffer is highest priority in the queue followed by other signals from the integrated circuit card.

The memory interface has the potential to be modified to interact with an external memory module. However, for this particular design, the memory interface is an interface to the processor's internal 1MB flash memory.

The IC interface will handle not only signals from the integrated circuit chips but also from the data streams lacking an integrated circuit chip, such as the Ethernet protocol or any SPI protocol. For security purposes, the IC interface will not directly handle any wireless communications so that any would be intruder would have to directly interface with the card to have access to it. Additionally, as the card itself does not handle any deterministic processing of the data processed through it, all handled data should be as secure as the least secure device connected to the Engine Communications Card.

For safety, the card will automatically terminate communications that experience significant errors or false and reset the corresponding interrupts; thusly avoiding errors that would halt the system. Any communication that external systems deem important should be sent again upon recognizing that the Engine Communications Card did not respond with an affirmative response. Additionally, the card is configured to stop accepting communication that is simply a repeated sequence to avoid babbling issues common in the MIL-STD-1553 protocol.

Nowhere in our CSCs will there be a graphic user interface as all graphic user interfaces is implemented externally to the system dependent on specific customer requirements.

4.18 Document 101,700 - Software Package

This document is in the Appendix. Alternatively, Document 101,600 provides a summary of how the software package was designed and which interfaces connect to one another. The software package includes the modified library files required to run the selected SPI, SCI, and parallel protocols on the Engine Communications Card. The software package then includes a modified version of the SPI loopback interrupts example from TI. This modified file removed the default loopback behavior of the SPI, SCI, and parallel protocols. In its stead the modified file has specific handles to input the data indirectly into its internal priority queue. Finally, in the software package are two queue2.c files and a testing script. The queue2.c file within the TI file folder is the helper file to the SPI loopback interrupt example. This file can be interfaced with to allow for the temporary storage of data transmitted to the TI chip until the TI chip has available cycles to transmit the data stored back through the respective chip. The other queue2.c file is a self-contained file for testing the functionality of the priority queue. This file can be run on its own as a proof of concept that the priority queue works as expected allowing different protocols to broadcast information to and from the chip. With the python testing script, this file can be run over and over 1728 times with different protocol configurations to prove that the testing script will not fail under different use configurations. This testing script was used in the software demonstrations for the verification status table.

4.19 Document 101,900 - Engine Communications Card Acceptance Test Procedure

This document information has been relocated to Acceptance Test Procedure in Section 4. The full acceptance test procedure can be found in the Appendix under document 101,900. The Acceptance Test Procedure Contains two sets of acceptance tests: Communication Protocol Tests and Thermal Tests.

5 Acceptance Test Procedure

5.1 Summary

The communication protocol acceptance tests require the external memory module that Honeywell will provide, and the temperature test requires thermal shock chamber, which Honeywell will provide. Therefore, the communication protocol tests will occur at Honeywell's Tucson Facility. All tests were concluded on 4/25/19. The required tests are summarized in Table 3 Acceptance Tests Summary. Some components of the ATP have been omitted from this section for the sake of brevity. The full tests are detailed in the Engine Communications Card ATP (Document 101,900).

Requirement Title	System	Circuit Board	Software
4.1.6 PCB Thermal Shock Resistance	-15 to 70 degrees C		
4.2.1 Parallel Transmission	Transmit >=3 signals w/ <2% errors		
4.4.1 Protocol Translation	Write and read to ARINC 429, ARINC 664, ARINC 825, MIL-STD-1553, and RS-485/422 data streams with less than a 2% error rate	Meet voltage requirements from 4.1.4-5	<2% error rate
4.4.3 Latency	<15 seconds		

Table 6 Acceptance Tests Summary

5.2 Communication Protocol Tests Summary

5.2.1 To test the 4.4.3 Latency requirement, demonstrate latency is under fifteen seconds using an external microcontroller or debugger, create a data stream in ARINC 429, ARINC 664, ARINC 825, MIL-STD-1553, and RS-485/422 (one at a time) and set the Engine Communications Card to an echo mode. Once each has been confirmed to be under 15 seconds, test and record the amount of time each dataset takes to return. See the Engine Communications Card ATP (Document 101,900) for full details.

5.2.2 To test the 4.2.1 Parallel Transmission requirement, demonstrate that these three signals can be transmitted at once by transmitting them into the board, using an external microcontroller, simultaneously to see that the Engine Communications Card can still perform to specifications. Test the written memory by reading with an external development board and record the error rate. See the Engine Communications Card ATP (Document 101,900) for full details.

5.3 Thermal Tests

5.3.1 To test the 4.1.6 PCB Thermal Tests, perform this procedure after the Communication Protocol Test procedure. Using an oven chamber provided by Honeywell the circuit board is tested undergoing 300 cycles within the environmental temperature range. Inspect that the board still functions after going through the heat chamber. Use a multimeter before and after the over chamber test to check for shorts. Once the board as been inspected, the entirety of the Communication Protocol Test procedure must be run once again. See the Engine Communications Card ATP (Document 101,900) for full details.

5.4 Communication Protocol Test Procedure

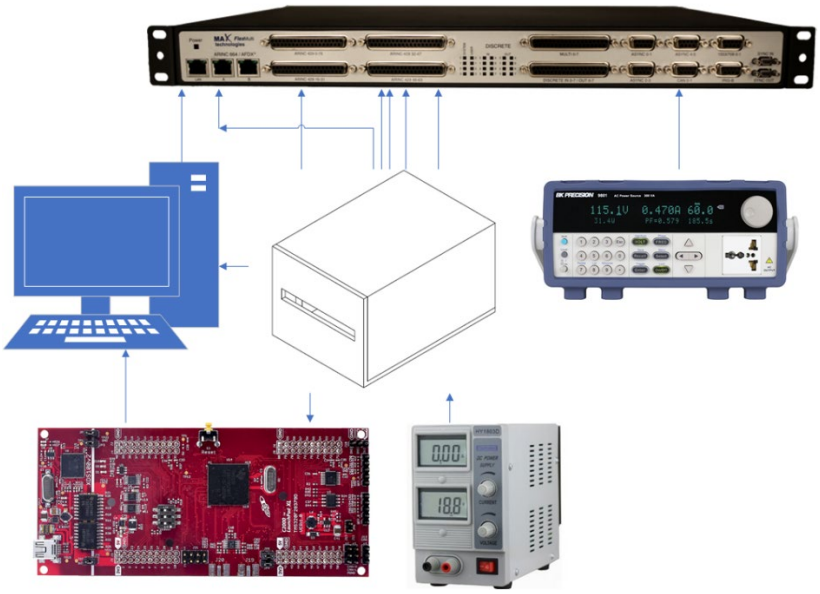

Figure 13 High Level Concept of Connections

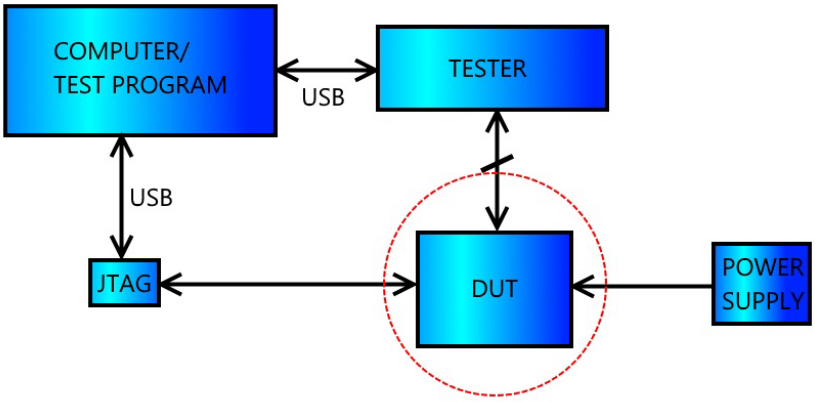

Figure 14 Interconnectedness

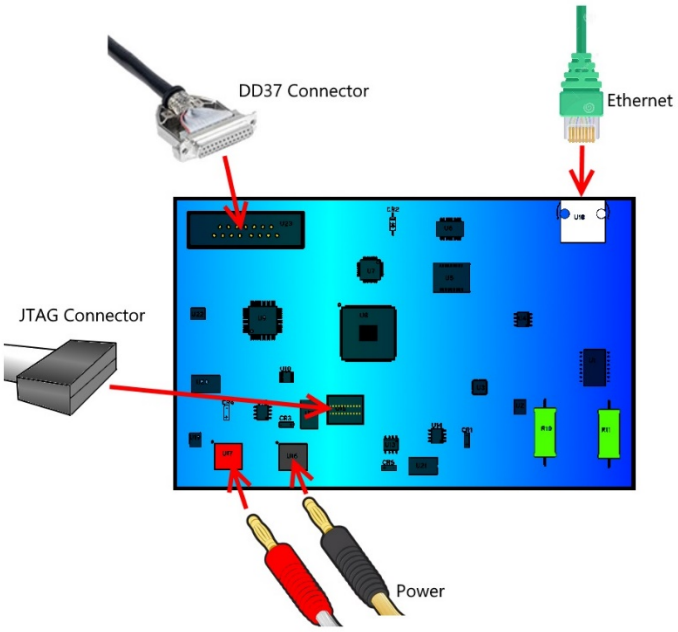

Figure 15 Board Connections

5.4.1 Connect the computer to the FlexMulti 664 using a RJ45 Ethernet Cable.

Figure 16 Connecting the computer to the router

5.4.2 Connect FlexMulti 664 to AC 120V power using the included power cable.

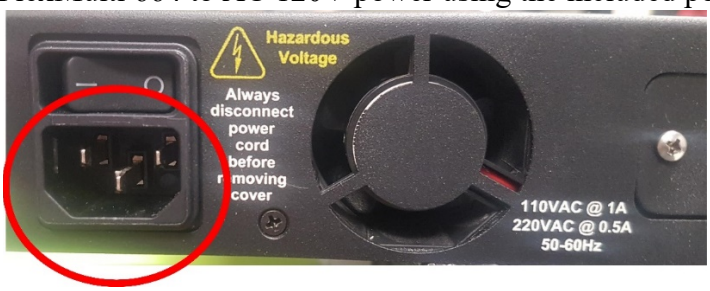

Figure 17 Flexmulti 664 Rear View

5.4.3 On the Engine Communication card to the FlexMulti 664 using two DE-9 cables in the DE-9 connectors, using one DE-15 cable in the DE-15 connector, one RJ45 cable in the RJ45 connector and one DD-37 cable in the DD-37 connector (Figure 3 Board Connections).

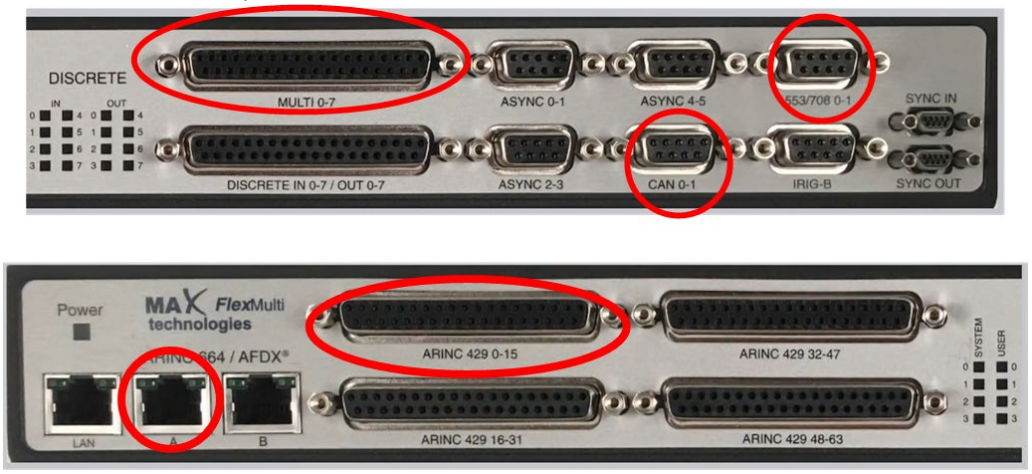

Figure 18 Cable Connections on the FlexMulti 664

5.4.4 Set the DC power supply to 28V and connect the banana plug wire to it.

Figure 19 Connect DC Power at 28V

5.4.5 Launch the Max Technologies MAXIM Air software

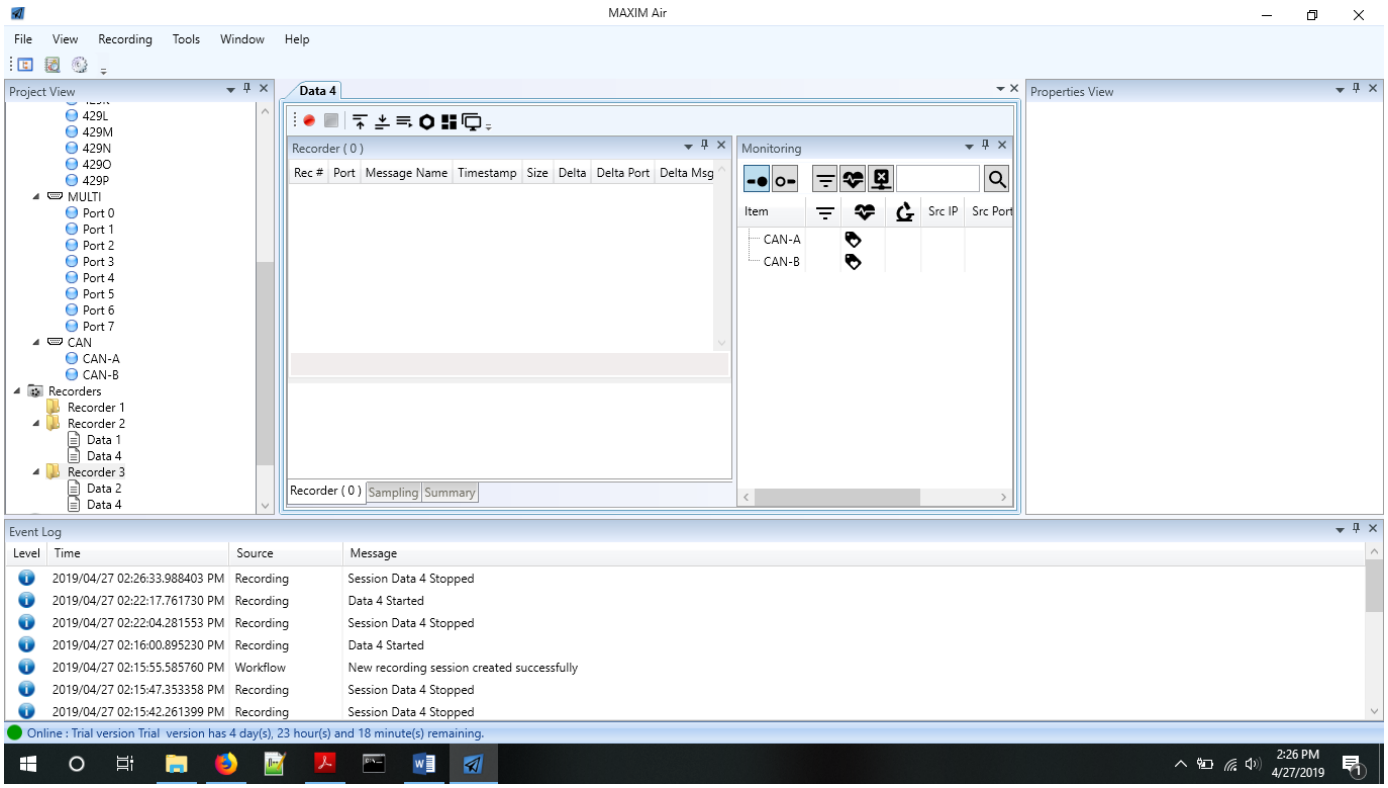

Figure 20 MAXIM Air set to Transmit on CAN Bus

5.4.6 Connect the other banana plug wires in the Engine communications card power input.

5.4.7 Connect the C2000 Delfino MCU F28379D LaunchPad™ development kit to the computer using the micro-USB cable.

5.4.8 Power on the FlexMulti 664 using the on/off switch on the rear of the device (Figure 7 Cable Connections on the FlexMulti 664).

5.4.9 Load the Test Project on the MAXIM Air application to transmit with the FlexMulti 664.

5.4.9.1 Ensure that the FlexMulti device is connected by temporarily turning off the computer's firewall.

5.4.9.2 Click on the network icon on the computer's taskbar and enter "Network and Internet Settings"

5.4.9.3 Click on "Change adapter options" and right-click on "Ethernet"

5.4.9.4 Select "Properties" then select "Internet Protocol Version 4 (TCP/IPv4)"

5.4.9.5 Click on "Properties" and select the radio button "Use the following IP address"

5.4.9.6 In the IP address field type in "192.168.0.X" where X is any number from 2 to 255; Leave "Subnet mask" to the default when you click into it (255.255.255.0) and leave "Default gateway" empty. Click "Ok".

5.4.9.7 Open the Command Prompt and run the command "ping 192.168.0.1"

5.4.9.8 A successful message will read 4 replies from 192.168.0.1, a message that the address is unreachable means the computer IP was not configured correctly, a general failure means that the computer's firewall is blocking the ping.

5.4.9.9 In MAXIM Air, begin a new project if none is available. In the window pop up, enter the device IP address in the field provided and click "Add". When prompted for a username & password, enter "admin" to both fields.

5.4.9.10 The MAXIM Air will detect the ports available, from the top bar, click on properties to open the Properties window to change protocol port properties such as transmission speed, state and library definitions used.

5.4.9.11 In the Project View window, right-click on Definitions and load in definitions from the file path “C://…/Documents/MAXIMAIR/Definitions”

5.4.9.12 Right-click on Recorders and add a recorder, right-click the recorder to open it.

5.4.9.13 From the protocol ports in the Project View window, add in any desired protocol ports to test.

5.4.9.14 Run the recorder by clicking on the red circle button.

5.4.10 On the computer open “Command Prompt.”

5.4.11 In command prompt change to the directory of protocolAutoRun.py

5.4.12 In command prompt run the following script “python -c 'from protocolAutoRun import *; phase1()'”

5.4.13 Automatic (by script): Adjust the static mode variable in the Engine Communication Software to ‘EchoWithoutMemory’ and flash the software to the Engine Communication Card following the instructions in the C2000 Microcontrollers Development Tools User Guide.

5.4.14 Start the Maxim FlexMulti Device software on the computer and set the output mode to ASYNC RS-422/485.

5.4.15 Begin the sample data transmission, collecting parity error (1.1), stop bit error data (1.2).

5.4.16 Start the Maxim FlexMulti Device software on the computer and set the output mode to MIL-STD-1553.

5.4.17 Begin the sample data transmission, collecting parity error (1.3), stop bit error data (1.4), and latency from the precise 64-bit timetag (1.5).

5.4.18 Start the Maxim FlexMulti Device software on the computer and set the output mode to ARINC 429.

5.4.19 Begin the sample data transmission, collecting parity error (1.6) and latency from the precise 64-bit timetag (1.7).

5.4.20 Start the Maxim FlexMulti Device software on the computer and set the output mode to ARINC 664.

5.4.21 Begin the sample data transmission, collecting parity error (1.8).

5.4.22 Start the Maxim FlexMulti Device software on the computer and set the output mode to ARINC 825.

5.4.23 Begin the sample data transmission, collecting CRC error (1.9), Segment error (1.10) and a precise 64-bit timetag as latency (1.11).

5.4.24 Start the Maxim FlexMulti Device software on the computer and set the output mode to ARINC 825, ARINC 664 and ASYNC RS-422/485.

5.4.25 Automatic (by script): Begin the sample data transmission, collecting the following

5.4.25.1 Automatic (by script): Parity error of the ARINC 664 sample (1.12)

5.4.25.2 Automatic (by script): CRC error (1.13), Segment error (1.14) and a precise 64-bit timetag for latency (1.15) of the ARINC 825 sample

5.4.25.3 Automatic (by script): Parity error

5.5 Thermal Test Procedure

Figure 21 Oven Chamber Control Panel

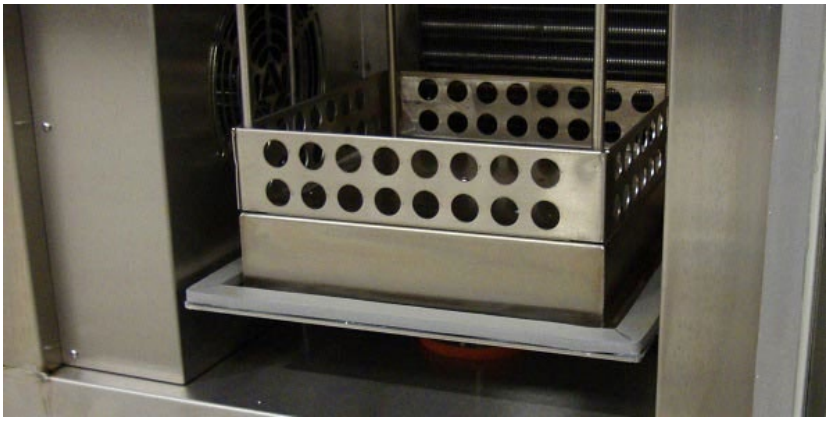

Figure 22 Thermal Shock Chamber Basket

a. Confirm that the circuit board has passed the tests communication protocol tests (7.1.1 through 7.1.5 from the PDR)
b. Confirm that the oven is safe to use:
    i. Inspect gaskets for cracks and complete seal.
    ii. Inspect all hoses (Blower, GN2, Air, etc.). and cables outside the chamber-hooked up, secure, cracks.
    iii. Check for cracks inside the chamber floor membrane.
    iv. Check for debris throughout chamber floor, including fixture mounting holes.
    v. Check compressed air line connections, fittings for leaks (listen for leaks)
    vi. Check Liquid Nitrogen lines for excessive condensation.
    vii. Check that the Solenoid Valve is secure and without leaks.
    viii. Check hoses for cracks and cracks I the chamber floor membrane. Check refrigerator hose for cracks.
    ix. Check for debris throughout the chamber floor including fixture mounting holes.
c. Follow the instructions in F4T Controller and D4T Data Logger Bases Manual to configure the Thermal Shock Test Chamber to the following settings:
    i. Cold chamber at -15 degrees Celsius
    ii. Hot chamber at 70 degree Celsius (use automated computer-controlled process on Thermal Shock Test Chamber if available on model) (2.1)

iii. Cycle count to 300 cycles

d. Configure the following cycle settings in the software following the manual:
   i. Phase 1 of cycle: keep the circuit board in the hot cycle temperature for 5 minutes
   ii. Phase 2 of cycle: adjust the temperature to that of cold cycle temperature at the maximum ramp speed of the temperature chamber
   iii. Phase 3 of cycle: keep the circuit board in the cold cycle temperature for 5 minutes
   iv. Phase 4 of cycle: adjust the temperature to that of hot cycle temperature at the maximum ramp speed of the temperature chamber
e. Place the circuit board carefully in the pneumatic basket of the Thermal Shock Test Chamber
f. Start the thermal shock testing following the F4T Controller and D4T Data Logger Bases Manual
g. Confirm that the circuit board still functions through the communication protocol tests (7.1.1 to 7.1.5) by re-running them (record information in Table 2.2). Record overall result (2.4)
h. If the circuit board fails, use a digital multimeter to find the locations where shorts occurred (compare measured voltage values with the voltage values on the circuit diagram (Document 100,010) to see if any measured values have been inadvertently grounded).

6 Models/ Analyses

6.1 Summary

The models and analyses used for the Engine Communications Card were developed to meet the various requirements of the system. The Cost model served as the primary budgetary system to ensure the decided-on design scheme would be affordable. The Conductor Model considered the amperage and voltage draw of every individual IC card on the board to determine if the chosen power supplies would be sufficient for the power requirements along each power rail. The Vibration Model and the Temperature Model were used to analyze the survivability of the system assembly in extreme temperatures and at critical frequencies. The Latency and the Parallel models analyzed the processor's ability to handle multiple protocols being sent to the circuit card at once. The Risk analysis considered the various risks that could be encountered during the project. Together all of these models provided a degree of confidence that the design would be operational when created beyond the design space.

6.2 Cost Models

The Cost Model (as shown in the Appendix Document 100,501) considers the costs from the Bill of Materials, which includes the cost of the Integrated Circuit (IC) components, the Microcontroller, Design Day (such as shirts and poster boards) and Fabrication. From this model, the potential cost range of the project as a whole is obtained. From this model, the conclusion could be drawn that the $4,000 budget from requirement 4.5.1 – Costs was met. The analysis and hand calculated bill of materials both found the budget margin to be $864.66.

6.3 Conductor Model

The Conductor Model (as shown in Appendix Document 100,505) considers the component (e.g. the Integrated Circuit chips) operating voltages and current requirements to find the necessary wire ratings, traces sizes and trace spacings. The Conductor Model provided information to the circuit card designers regarding the required power supplies in terms of peak voltage and peak amperage needed.

The power circuitry, under worst case scenarios, draws the greatest amperage in the 3.3V bus, which needs 1A of current draw during operation. The next highest amperage requirement comes from the 1.2V bus, which requires 0.5A. The remaining busses require less than 30mA, even under peak load.

Power sequencing is required by one of the parts, so a resistive element may be used to enable the 3.3V bus using the signal generated from the 5V bus. The conductor model therefore meets the 4.1.4-4.1.5 as each trace from the printed circuit card meets the IEEE recommendation for the amperage and voltage of the wire.

6.4 Vibration Model

The Vibration Model (as shown in the Appendix Document 100,510) takes customer provided environmental vibration frequencies and outputs the vibration frequencies for the critical area within the circuit board - chassis assembly. The vibration model must meet requirement 4.1.8 and survive through 10 user defined frequencies and amplitudes. The Vibration Model resulted in six passing stress tests at critical frequencies. However, four critical frequencies experienced complete failures where the circuit card was simulated to come in contact with the ceiling of the case. However, this model is not necessarily a failure as these are simply frequencies at which the circuit card cannot be operated at. Accordingly, given that none of the critical frequencies were at the customer defined critical frequencies to avoid, the vibration model and analysis can be considered a success.

6.5 Temperature Model

The Temperature Model (as shown in the Appendix Document 101,520) considers the Environmental Temperature experienced by the circuit card and the power usage of the on-board Integrated Circuits. With the power usage of Integrated Circuits, a temperature range for those cards is derived. Using that temperature range and the environmental temperature range, the overall temperature range for the circuit board is generated.

Part	Ta= -15 C	Ta=70 C	Pwr (W)	θ (C/W)	Tj @ -15C	Tj @ 70C
CAN IC	-15	70	0.048	~50	-12.6	**72.4**
ARINC 429 IC	-15	70	0.05	~50	-12.5	**72.5**
RS-242/232 IC	-15	70	0.00825	~50	-14.5875	**70.4125**
ML-STD-1553 IC	-15	70	0.61	~45	12.45	**97.45**
ETHERNET IC	-15	70	0.18	~45	-6.9	**78.1**
TMs320F2835xD	-15	70	2.1	21	29.1	**114.1**
LT1761MPS5/1.8	-15	70	0.07	135	-5.55	**79.45**
TPS5410-EP-1	-15	70	0.293	75	6.975	**91.975**
TPS5410-EP-2	-15	70	0.066	75	-10.05	**74.95**
TPS5410-EP-3	-15	70	0.016	75	-13.8	**71.2**

Table 7 Temperature Model

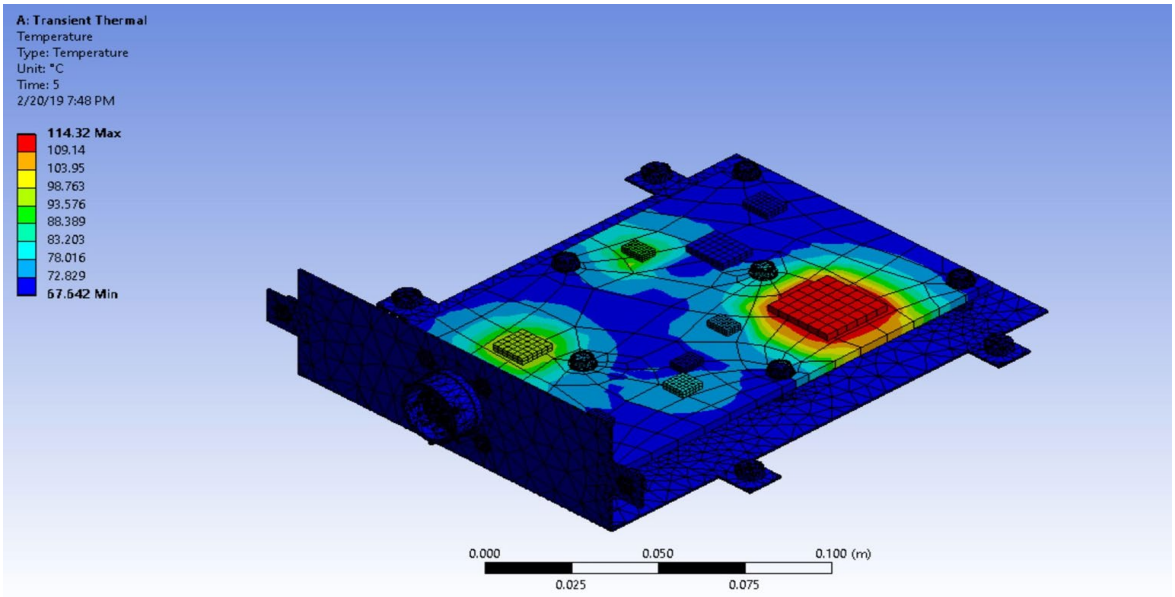

Figure 23 Ansys temperature model at 70 degrees Celsius

As shown in Table 3 Temperature Model, the calculated temperatures are within a safe range and are not reaching the maximum temperature of 120° C. When the card's temperature reaches $70°C$, the peak juncture temperate is the TMs320F2877D chip; the steady state temperature is $114.1°C$. In ANSYS, the peak temperature was calculated to be $114.32°C$. Hence, both the hand calculated temperature values and the ANSYS models pass the temperature model based on system requirement 4.1.6 and system requirement 4.1.7 which require the PCB and case will survive in a powered down state from -15°C to 70°C by staying below a junction temperature of 120°C.

6.6 Parallel Model

The Parallel Model (as shown in the Appendix Document 100,530) takes in Data Transmission Frequencies and Package Sizes (ARINC 429, ARINC 664, ARINC 825, MIL STD 1553, RS 485/422) to calculate the processor's ability to handle multiple signals at once to fulfill system requirement 4.2.2. The numerical representation of this ability to Peak Processor Load, or in other words, the maximum portion the processor's cycles that is utilized.

The ARINC 429, ARINC 825, and MIL STD 1553 protocols have available IC chips that will allow for buffering to assist with throughput handling. The peak processor load is desired to be less than 60% (see PDR requirement 4.2.1) as it only takes up a maximum of 13.4% of the processor's available load, providing a 77.6% model margin. As seen by the tables within the Parallel Model document, this analysis passed the requirement. Accordingly, the used TI processor (see Bill of Materials, Document 101-001) meets the parallel model passing requirement 4.2.1.

6.7 Latency Model

The Latency Model (as shown in document 100,540) takes in the Data Transmission Frequencies and Package Sizes (ARINC 429, ARINC 664, ARINC 825, MIL STD 1553, RS 485/422) to calculate the system's latency from the critical path of the circuit in order to fulfill system requirement 4.4.3. In this model, the critical path is the maximum amount of time from receiving a data transmission from any one of the data I/O's, to outputting a response to that same data I/O. The path will pass through the processor and the customer's memory module. For more information on critical path tracing in circuits, read "A critical path tracing algorithm for synchronous sequential circuits" by Menon, Levendel, and

Abramovici in the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Hence, latency is determined by the critical path.

Protocol	Total Delay (seconds)
ARINC 429*	4.202089762
ARINC 664	1.401601212
ARINC 825*	1.488648923
MIL STD 1553*	1.015641212
RS 485/422	1.456801212
Critical Path Latency	**4.202089762**

Table 8 Total Input Delays by Protocol

The total system latency is 4.202 seconds. This is well below the required 15 seconds or less latency of the PDR's 4.4.3 system requirement, all the ICs fell below the 13.5 seconds or less derived latency requirement. Hence, there is a 68.87% margin, so the project is slated to meet the latency requirement 4.4.3 with ease.

6.8 Risk Model

Throughout the project, Risk (1) was encountered and resolved using the mitigation plan. None of the other risks that were considered ahead of time were encountered. However, an unaccounted-for risk did occur where one of the user specifications remained undefined for too long. When this risk was encountered, the decision was made to use Contingency (2), where the requirement was altogether eliminated. The mitigation plan of working modularly proved successful as to not cause any complete redesigns of the circuit design or the software design.

Rank	Description	Mitigation	Contingency
1	IC component total failure	Buy at least 2 ICs upfront (3 for items with high lead times)	Use ICs from a different manufacturer or develop our own
2	Software not developed by deadline	Begin working on software immediately and develop modularly for each block	Reduce feature size and remove more robust error handling
3	Supplier delay	Order as early as possible	Order components from third party suppliers
4	Processor shorting and IC ESD damage	Perform communication protocol tests first, handle parts as little as possible, practice ESD best-practices	Use one of the extra ICs
5	Project goes over budget	Carefully anticipate each individual component cost and buy standard ICs rather than military rated ICs	Remove non-critical signals to cut down on IC costs

Table 9 Risk Table

Likelihood	Consequences				
	Insignificant (Risk is easily mitigated by normal day to day process)	Minor (Delays up to 10% of Schedule or additional cost)	Moderate (Delays up to 30% of schedule or additional cost)	Major (Delays up to 50% of schedule or additional cost)	Catastrophic (Project abandoned)
Almost Certain (≥ 90%)	High	High	Extreme	Extreme	Extreme
Likely (50 - 89%)	Moderate	High	High	Extreme	Extreme
Moderate (10 - 49%)	Low	Moderate	High **(2)** (3)	Extreme	Extreme
Unlikely (3 - 9%)	Low	Low **(1)(4)**	Moderate	High	Extreme
Rare (< 3%)	Low	Low **(5)**	Moderate	High	High

Table 10 Risk Analysis

7 Acceptance Test Results

7.1 Overview

Some components of the ATP have been omitted from this section for the sake of brevity and to avoid repeating Section 5, Acceptance Test Procedure. The complete acceptance test information including referenced documents and test equipment is listed inside of Document 101,900 in the Appendix. Through an ECR, the thermal tests were made optional.

7.2 Overview – Communication Protocol Tests

This procedure outlines the acceptance tests to be performed regarding the communication protocol requirements. This test verifies that the Engine Communications Card meets latency, memory, and parallel transmission requirements. All tests are automated using a Python script called protocolAutoRun.py.

To test the 4.4.3 Latency requirement, demonstrate latency is under fifteen seconds using an external microcontroller or debugger, create a data stream in ARINC 429, ARINC 664, ARINC 825, MIL-STD-1553, and RS-485/422 (one at a time) and set the Engine Communications Card to an echo mode. Once each has been confirmed to be under 15 seconds, test and record the amount of time each dataset takes to return.

To test the 4.2.1 Parallel Transmission requirement, demonstrate that these three signals can be transmitted at once by transmitting them into the board, using an external microcontroller, simultaneously to see that the Engine Communications Card can still perform to specifications. Record the error rate.

Test #	Test	Requirements
7.1.1	Latency requirement	< 15 seconds
7.1.2	Parallel Transmission	< 2% error rate

Table 11 Table of Communication Protocol Tests

7.3 Communication Protocol Test Data Sheet – Pre-Thermal Tests

Communication Protocol Acceptance Test Data Sheet			
Referenced ATP Paragraph Number: 1.1			
Analysis Referenced (for verification by T/A): Parallel Transmission Analysis and Latency Analysis			
Name of Time: Communication Protocol Tests			
Unit Under Test (UUT): Name: Engine Communications Card Part Number: 101,000 Serial Number: N/A			
Result Overall (Pass/Fail): Test 7.1.1 (Pass/Fail based on 1.5, 1.11, 1.15) Test 7.1.2 (Pass/Fail based on 1.12-1.17)		Date of Test: 4/25/19	
Recording of Test Measurement:	Requirement (SRD, with Tolerance):	Test Equipment Error:	Adjusted Test Limit:
2.1. RS-422 Error Bits: 0 error bits	1.1. <2%	1.1. <1%	1.1. <3%
2.2. RS-422 Latency: 48.024 ns	1.2. <15 Seconds	1.2. <1 Second	1.2. <16 Seconds
2.3. MIL-STD-1553 Parity Error:	1.3. <2%	1.3. <1%	1.3. <3%
2.4. MIL-STD-1553 Stop Bit Error:	1.4. <2%	1.4. <1%	1.4. <3%
2.5. MIL-STD-1553 Latency:	1.5. < 15 Seconds	1.5. < 1 Second	1.5. < 16 Seconds
2.6. ARINC 429 Parity Error:	1.6. <2%	1.6. <1%	1.6. <3%
2.7. ARINC 429 Stop Bit Error:	1.7. <2%	1.7. <1%	1.7. <3%
2.8. ARINC 664 Parity Error:	1.8. <2%	1.8. <1%	1.8. <3%
2.9. ARINC 825 CRC Error:	1.9. <2%	1.9. <1%	1.9. <3%
2.10. ARINC 825 Segment Error:	1.10. <2%	1.10. <1%	1.10. <3%
2.11. ARINC 825 Latency Error:	1.11. <15	1.11. <1	1.11. <16
2.12. In Parallel ARINC 664 Parity Error:	Seconds	Second	Seconds
2.13. In Parallel ARINC 825 CRC Error:	1.12. <2%	1.12. <1%	1.12. <3%
2.14. In Parallel ARINC 825 Segment Error:	1.13. <2%	1.13. <1%	1.13. <3%
2.15. In Parallel ARINC 825 Latency:	1.14. <2%	1.14. <1%	1.14. <3%
2.16. In Parallel ARINC 422 Parity Error:	1.15. <15	1.15. <1	1.15. <16
2.17. In Parallel ARINC 422 Stop Bit Error:	Seconds	Second	Seconds
	1.16. <2%	1.16. <1%	1.16. <3%
	1.17. <2%	1.17. <1%	1.17. <3%
Computations, (Include Analysis Results, if any): None required			
Signatures: Tester: _____			

Customer: _____

7.4 Overview – Thermal Tests

The thermal test is performed using an oven chamber provided by Honeywell, the circuit board is tested undergoing 300 cycles within the environmental temperature ranging from -15 to 70 degree Celsius. Inspect that the board still functions after going through the heat chamber. Use a multimeter before and after the oven chamber test to check for shorts. Confirm that the circuit board has passed the communication protocol test once before and again after completing the thermal test.

Test #	Test	Requirements
7.2.1	Thermal durability	300 Cycles

7.5 Thermal Tests Data Sheet

Thermal Acceptance Test Data Sheet (optional)			
Referenced ATP Paragraph Number: 2.1			
Name of Time: The Thermal Test			
Unit Under Test (UUT): Name: Engine Communications Card Part Number: 101,000 Serial Number: N/A			
Result (Pass/Fail):		Date of Test:	
Recording of Test Measurement: 3.1. Temperature: 3.2. Cycles: 3.3. Time per cycle: 3.4. Board survival:	Requirement (SRD, with Tolerance): 2.1. <-15- and >70-degrees C 2.2. >=300 2.3. >4:30 and < 5:30 2.4. True	Test Equipment Error: N/A N/A +/- 1 seconds N/A	Adjusted Test Limit: N/A N/A >4:29 and < 5:31 N/A
Computations, (Include Analysis Results, if any): None required			
Signatures: Tester: _____ Customer: _____			

7.6 Communication Protocol Test Data Sheet – Post-Thermal Tests

Post Thermal Acceptance Test Communication Protocol Acceptance Test Data Sheet (optional)
Referenced ATP Paragraph Number: 1.1
Analysis Referenced (for verification by T/A): Parallel Transmission Analysis and Latency Analysis
Name of Time: Communication Protocol Tests
Unit Under Test (UUT): Name: Engine Communications Card

Post Thermal Acceptance Test Communication Protocol Acceptance Test Data Sheet (optional)			
Part Number: 101,000 Serial Number: N/A			
Result Overall (Pass/Fail): Test 7.1.1 (Pass/Fail based on 1.5, 1.11, 1.15, 1.22) Test 7.1.2 (Pass/Fail based on 1.12-1.17)		Date of Test:	
Recording of Test Measurement: 1.1 RS-422 Error Bits: 1.2. RS-422 Latency: 1.3. MIL-STD-1553 Parity Error: 1.4. MIL-STD-1553 Stop Bit Error: 1.5. MIL-STD-1553 Latency: 1.6. ARINC 429 Parity Error: 1.7. ARINC 429 Stop Bit Error: 1.8. ARINC 664 Parity Error: 1.9. ARINC 825 CRC Error: 1.10. ARINC 825 Segment Error: 1.11. ARINC 825 Latency Error: 1.12. In Parallel ARINC 664 Parity Error: 1.13. In Parallel ARINC 825 CRC Error: 1.14. In Parallel ARINC 825 Segment Error: 1.15. In Parallel ARINC 825 Latency: 1.16. In Parallel ARINC 422 Parity Error: 1.17. In Parallel ARINC 422 Stop Bit Error: :	Requirement (SRD, with Tolerance): 1.1. <2% 1.2. <15 Seconds 1.3. <2% 1.4. <2% 1.5. < 15 Seconds 1.6. <2% 1.7. <2% 1.8. <2% 1.9. <2% 1.10. <2% 1.11. <15 Seconds 1.12. <2% 1.13. <2% 1.14. <2% 1.15. <15 Seconds 1.16. <2% 1.17. <2%	Test Equipment Error: 1.1. <1% 1.2. <1 Second 1.3. <1% 1.4. <1% 1.5. < 1 Second 1.6. <1% 1.7. <1% 1.8. <1% 1.9. <1% 1.10. <1% 1.11. <1 Second 1.12. <1% 1.13. <1% 1.14. <1% 1.15. <1 Second 1.16. <1% 1.17. <1%	Adjusted Test Limit: 1.1. <3% 1.2. <16 Seconds 1.3. <3% 1.4. <3% 1.5. < 16 Seconds 1.6. <3% 1.7. <3% 1.8. <3% 1.9. <3% 1.10. <3% 1.11. <16 Seconds 1.12. <3% 1.13. <3% 1.14. <3% 1.15. <16 Seconds 1.16. <3% 1.17. <3%
Computations, (Include Analysis Results, if any): None required			
Signatures: Tester: _____ Customer: _____			

8 Final Budget

The final budget had a remaining $846.66. See the Printed Circuit Board Assembly (Document 100,030) for the complete itemized item list.

Doc #	Requestor	Vendor(s)
1	Brian Winkler	Blackhawk

Doc #	Requestor	Vendor(s)
2	Brian Winkler	Digi-Key
3	Brian Winkler	Holt
4	Brian Winkler	Arrow
5	Brian Winkler	Arrow
6	Brian Winkler	Holt
7	Brian Winkler	Digi-Key
8	Brian Winkler	Digi-Key, OshPark
9	Brian Winkler	OshPark
10	Edward D Brunton	Amazon, FedEx
11	Samuel Bessette	Arrow, Autodesk, Digi-Key, Mouser, OfficeMax
12	Alejandro Alvarez	OfficeMax
	Total	**$3,153.34**

Table 12 Final Budget

9 Lessons Learned

9.1 Overview

The current design failed to meet the original requirements of the project sponsor as laid out in the 9/11/2018 revision of the Project Proposal where the board was intended to "[output] the data to a common memory." Although this requirement was removed through an Engineering Change Request, the goal and purpose of the project as a whole was missed. Without the ability to output data to a common memory bus, a circuit card that meets all of the project specifications would only serve to act as a debugging tool of multiple protocols at once, providing any device that is supposed to transmit a protocol an opportunity to receive its own information back. However, devices, such as the FlexMulti 664, already perform this function with greater customizability than the Engine Communications Card.

During the design process, small delays that pushed back the schedule and eliminate redesign time should have been handled with greater concern and quicker response. While maintaining the design with all five protocols may have seemed right at the time, scaling the project back early on could have freed up resources to ensure that more than one protocol could work all the way through the system.

The failures in the Acceptance Test Procedure were caused not only by a faulty design but also by a test procedure that was not subdivided enough. The design of the test procedure went on the premise that the integrated circuit cards and the TI microprocessor would have no communication issues whatsoever. Hence, the powering of the integrated circuit cards was never able to be tested despite it being a solidified requirement since the Preliminary Design Review. Additionally, although the software had unit testing performed on it outside of the Acceptance Test Procedure, no official test was configured to provide proof of operation, resulting in a project that was a greater failure than it potentially was.

Overall, while the design of the system was not fundamentally flawed, the team often lost sight on what the grander goal of the project was and got lost in the details of implementation. Future interactions of this project would want to focus on ensuring the project worked from start to finish rather than attempting to get the entire system up and running all at once. While the modular style design used was well intentioned, the order in which the components were designed should have been switched up, prioritizing at least partial functionality of the system.

9.2 Project Successes

The project was successful in the overarching design. From trade studies early on and progress though the project, the team found that anytime a unique protocol had to be handled directly by the central microprocessor, unintended consequences would occur elsewhere in the code. By splitting the project up so that integrated circuit chips were able to take on the responsibility of managing communication with complex communication protocols, the team was able to devote less time to learning about specific communication protocols and how to handle unique errors to each of those protocols to designing a system that had a consolidating interface that would avoid such inconveniences. That said, additional research could have been put in through the process to determine the expected loads from each communication protocols rather than expecting another source to provide that information as this design was intended to be modular and versatile for a number of different environments.

The chassis design ended up meeting requirements for the project, providing a safe enclosure to secure the Engine Communications Card, based on the analysis. By the end of the design process, the chassis survived all vibration analysis tests without any points of failure.

Throughout the design process, the team did an effective job at relaying changes to the mentor and the sponsor. During biweekly weekly meetings with the team mentor, the team would ensure that at least one individual was present for the entirety of the time to communicate team concerns with the mentor and take feedback from the mentor. When creating major changes to the design, the team consistently reached out to the team sponsor to ensure that the sponsor was keyed into changes that were occurring and to ensure that sponsor knew of questions that the team had. If communication channels ever had difficulties arise, the team brought these issues to the mentor and took advice from the mentor on how to move forward through uncertainty.

9.3 Project Failures

The software design lacked any official acceptance test procedures. Hence, although some components of the software were shown to be working in unit tests, in official tests, no definitive evidence existed to show if the software needed a design revision or if the hardware needed a design revision. Likewise, all hardware protocol tests relied on the software working, so all the tests failed in the communication protocol category could have been do to the software malfunctioning or due to the hardware malfunctioning. The team should have designed completely separate acceptance test procedures for both the software and the hardware so that the assemblies could be isolated, allowing for at least a solo success rather than a joint failure.

The removal of the external memory module, although a necessary action at the point in the project that it was removed, defeated the purpose of the project. Additionally, many of the current designs still are made around the idea of an external memory module, causing some of the same issues that having an external memory module could have caused. Accordingly, the removal of the external memory module should have been performed far sooner in the design process, preferably by the Preliminary Design Review or at least by the Critical Design Review. If the external memory module had been removed at that point, or incorporated into the design scope of the project (as in the memory module being a part of the board itself), then another solution could have been created to allow for the communication protocols to communicate with an external source as opposed to being limited to only back to themselves.

During the design process, the team was given open specifications that had ambiguities open to interpretation. While the team could have embraced these ambiguities and used them to create a dynamic and innovative design, the team was slow to act on defining their own specific requirements. This inaction was especially present when it came to be designing the temperature and vibration requirements as it relates to an enclosure. Through the process, these specifications were vague and caused signification delays for the Critical Design Review and Technical Data Package.

9.4 Looking Back

Throughout the design process, scheduling, both meetings and tasks, remained a consistent failure of the team. To do the process over again, the team should have embraced Microsoft Projects more and worked on using a unified calendar system for accountability. Additionally, meeting notes with attendance details should have been taken with greater consistency to ensure that individuals could be held accountable for deadlines and provided with assistance from other team members when timelines began to slip.

During the design process a more Agile style of design should have been used to avoid late stage failures. Early on in the process, the model of the microprocessor kept on changing based on how the design was going. However, had the design been produced in a more iterative fashion, multiple options could have been created and then the practicality of those design decided based on the iterative successes or failures in the design. Producing such prototypes could have produced better estimations for the numbers used in the trade studies in the Preliminary Design Review and allowed for a greater level of confidence and understanding of the design in later stages of the project.

A major issue that arose during the design process was obtaining the standardized requirements for each communication protocol as some of those standards were behind paywalls. During the Preliminary Design stage of the process, these standards were overlooked at that point and too many uneducated assumptions were made about the protocol operation. Accordingly, instead of waving off the uncertainties about the design, the team should have taken a deeper look at each protocol and opened up the team budget to buying documentation about these protocols if another individual within Honeywell was unable to obtain the information for the team. If this would have created budget limitations, then at that point, the team should have seriously considered removing a communication protocol from the requirements so that fewer and higher quality implementations of the known communication protocols could be performed. The team's way of addressing the communication protocols overstretched some of the team's resources preventing the delivery of a quality and functional final product. For example, when the team picked an integrated circuit chip for the MIL STD 1553 specification, the team failed to look through the integrated circuit chip's documentation thoroughly to see that specific registers much be requested from the SPI master in order to facilitate the transmission of data to and from this integrated circuit chip. This lack of foresight early on could have been mitigated by scheduling additional meetings with individuals familiar with these protocols. Hiring a consultant on these protocols early on to point out these oversights could have saved redesign time and improved the test results of the Engine Communications Card.

Often throughout the project, the team focused on optimizing the design rather than making sure the design was functional. To do it over again, the team should have flipped this paradigm on its head and focused heavily on ensuring functionality before optimizing components for size or power consumption, neither of which were requirements listed in the Critical Design Review.

9.5 Moving Forward

Under the assumption that this project will be repeated so that the design can be made into a functional one, future designs and tests must focus in on the communication between the microprocessor and the integrated circuit chips. This was the one of the greatest oversights in the project itself since the two systems were designed entirely separately leading to communication errors between the two subdivided design teams.

Based on the current issues with the SPI bus on the TI microprocessor, in a redesign, using another microprocessor with better documentation for the SPI protocol that does not solely rely on interrupts that are difficult to debug could prove better for a development environment. The current TI solution, although somewhat function, caused difficulties in pinpointing where software or hardware faults occurred. Although the team still believes the TI microprocessor is the best solution, a better solution may exist for a prototyping environment for a project that has not been done before such as this one.

In a redesign of the circuitry, a serious consideration should be made to find alternative power solutions as the power modules proved to be problematic in the powering of the board. With a high amperage requirement, some components in the team designed power supplies tended to fail and posed a risk of damaging the integrated circuit cards, delaying the development process. Because size was not a requirement for the project, larger, already made and proven solutions should be considered in the next iteration of the project so that the board can have full functionality before optimization.

The team would also suggest physically creating a chassis in the future. Upon reflection, the thermal acceptance test results are highly questionable because the Engine Communications Card will never be completely exposed when in any real environment. For the results to be meaningful, the circuit card should be in a realistic environment which includes not only the heat fluctuations of the environment but also the chassis, which may retain heat from cycle to cycle and may help to decrease the speed of temperature change of the circuit card itself.

10 Summary

The Engine Communications Card did not have the functionality of a communication card. Nevertheless, the Engine Communications Card's fundamental design is sound and has the potential to be transformed into a functional communication card if additional time and resources are invested into it with additional external support to reevaluate the existing requirements and determine if they are suitable based upon the lessons learned. The team stands with the idea that this project is doable but remain skeptical on some aspects of the practicality of implementing certain protocols given their requirements, which often require the host of the Engine Communications Card knowing details of a particular implementation of a protocol, as protocols have a variety of different implementations and configurations. Now that a prototype of this project has been designed, future design teams must create trade studies to determine if a 'universal' Engine Communications Card will require too much customization time for every application to even be considered universal as opposed to simply customizing a solution based on every particular situation's requirements.

11 Appendix

The following documents are ordered by document number. See the indentured document table for details.