

A SECURE ARCHITECTURE FOR DISTRIBUTED CONTROL OF TURBINE
ENGINE SYSTEMS

Thesis
Submitted to
The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree of
Master of Science in Computer Engineering

By
Justin R. Eise
Dayton, Ohio
May, 2019



A SECURE ARCHITECTURE FOR DISTRIBUTED CONTROL OF TURBINE
ENGINE SYSTEMS

Name: Eise, Justin R.

APPROVED BY:

Vamsy Chodavarapu, Ph.D.
Advisory Committee Chairman
Associate Professor, Electrical and
Computer Engineering

Michael Wicks, Ph.D.
Co-Advisor
Faculty Member, Electrical and
Computer Engineering

Guru Subramanyam, Ph.D.
Committee Member
Professor, Electrical and Computer
Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation
Professor
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering

© Copyright by

Justin R. Eise

All rights reserved

2018

ABSTRACT

A SECURE ARCHITECTURE FOR DISTRIBUTED CONTROL OF TURBINE ENGINE SYSTEMS

Name: Eise, Justin R.
University of Dayton

Advisor: Dr. Vamsy Chodavarapu

As aircraft turbine engine technologies have evolved, a need has emerged to move from legacy monolithic control systems to a distributed paradigm. This thesis is based upon research performed to develop a distributed control architecture that is secure and robust against extreme environmental conditions. Background, the proposed architecture, design of a network smart node, and experimental testing of a high-temperature microprocessor are presented.

In compliance with the International Traffic in Arms Regulations of the United States,
substantial work has been omitted from this document.

ACKNOWLEDGMENTS

This work is funded by Spectral Energies, LLC under Small Business Innovation Research contract with the U.S. Air Force. The author acknowledges the support and guidance of Dr. Vamsy Chodavarapu and Dr. Michael Wicks throughout the research process. I am grateful for their support and for the valuable advice of fellow researchers Devin Smarra, Devin Spatz, and Dr. Dan Wetzel. I also offer my thanks to the staff of the Department of Electrical and Computer Engineering, especially our laboratory manager, Julie Motz, and our senior administrative assistant, Nancy Striebich. Finally, I express my appreciation for Scott Paepflow and his leadership of the University of Dayton community of music ministry.

TABLE OF CONTENTS

ABSTRACT	iii
EXPORT CONTROL STATEMENT	iv
ACKNOWLEDGMENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	ix
CHAPTER I. INTRODUCTION	1
1.1 A Background on Control of Turbine Engines in Aircraft	1
1.2 Problem Statement	2
CHAPTER II. ARCHITECTURE	4
2.1 Overview	4
2.2 Networking	5
2.3 Computing	7
2.4 Sensing	8
2.5 Actuator Control	10
2.6 Security	10
CHAPTER III. SMART NODE	12
3.1 Overview	12
3.2 Microcontroller	13
3.3 Oscillator	14
3.4 Operational Amplifiers and Precision Voltage Reference	14
3.5 Analog to Digital Converter	14
3.6 Differential Transceiver	16
3.7 Digital Temperature Sensor	16
3.8 Connectors	16
3.9 Software	16
3.10 Sensor Sampling and Signal Processing	18
3.11 Data Transmission and Structure	18
CHAPTER IV. EXPERIMENTAL TESTING	21
4.1 Ambient Thermal Environment	21
4.2 Radiant Thermal Environment	23

CHAPTER V. CONCLUSION	26
BIBLIOGRAPHY	27
APPENDIX A. Polynomial Function and MATLAB Code for Post-Processing of Thermocouple Data	30

LIST OF FIGURES

2.1	Distributed System Architecture	5
3.1	Smart Node Functional Diagram	12
3.2	Prixarc ARC-1	15
3.3	Vectron PX-7021	15
3.4	Cissoid CHT-RUBY	15
3.5	Cissoid CHT-ADC10	17
3.6	Texas Instruments SN65HVD11-HT	17
3.7	Texas Instruments LM95172	17
3.8	Min-e-Con BR-Series Connectors	18
4.1	Experiment Diagram: Ambient Thermal Environment	22
4.2	Experimental Results: Ambient Thermal Environment	22
4.3	Experimental Results: 1 V	23
4.4	Experimental Results: 0 V	23
4.5	Experiment Diagram: Radiant Thermal Environment	24
4.6	Experimental Results (Raw): Radiant Thermal Environment	25
4.7	Experimental Results (Low-Pass Filtered): Radiant Thermal Environment .	25

LIST OF TABLES

3.1	Definition of Terms in Figure 3.1	13
3.2	Transmission Frame Structure	20

CHAPTER I

INTRODUCTION

1.1 A Background on Control of Turbine Engines in Aircraft

The first operational tactical aircraft to employ turbomachinery as the primary powerplant was the German-made Messerschmitt Me 262 Swallow. It was equipped with two Junkers Jumo 400 turbojets, an eight-stage axial compressor design that represented the state of the art in 1942. Electronic control systems were not available, so the turbojet relied upon mechanical systems and pilot input for feedback. For example, a mechanical governor with flyweights and a piston fuel spill system was used to control engine rotational rate. By the conclusion of World War II, the engine's developers had demonstrated 150 hours of service life in flight testing.[1]

Ten years later, in 1955, the United States had developed a number of jet aircraft, including the U-2 Dragon Lady, a high-altitude reconnaissance aircraft designed to avoid advanced air defenses by flying 70,000 feet above sea level. In the stratosphere, the U-2 could not be reached by contemporary Soviet interceptors and anti-aircraft missile systems, however the U-2's single Pratt & Whitney J57 turbojet was vulnerable to loss of combustion stability if the pilot did not maintain precise control of its operation. In the event of a flameout, the pilot would have to descend to restart his engine, making the aircraft vulnerable to attack.[2]

Both the Me 262 and the U-2 would have benefitted from digital electronic engine controls had such technology been available at the time. With no precision control over engine temperature, the Jumo 400 relied on advanced metallurgy to extend engine life, putting difficult demands on strategic wartime resources. The J57 required the pilot to ensure the required conditions for air-fuel combustion, however that workload could have been relieved by a control system with full authority over the engine.

In 1972, the first full-authority analog electronic engine control unit was developed by Garrett/Air Research for the TFE731 engine. The first known flight test of a full-authority *digital* engine control (FADEC) system occurred in 1981 with the Pratt & Whitney F100 afterburning turbofan, an engine designed for the F-15 air superiority fighter. The F100 FADEC was noteworthy for having eight digitally controlled variables. The first civilian Digital Electronic Engine Control (DEEC) system went into service in 1982 with the Pratt & Whitney JT9D high-bypass turbofan. The JT9D was initially developed in the 1960s for military applications and the Boeing 747 wide-body airliner.[3]

1.2 Problem Statement

In terms of control schema, the FADEC technology space has experienced enormous development since the first digital systems emerged. Self-tuning, on-board modeling, and advanced analytics have been deployed to high-technology aircraft in order to improve performance and reliability.[3] Development of electronics at the engine has not kept pace, although there are both commercial and military incentives for technological advancement.[4][5][6]

In contrast to legacy FADECs which are monolithic and temperature-sensitive, this work presents a distributed architecture for engine control that incorporates elements of extreme-environment electronics, networking of distributed systems, and cyber-security. The work

concludes with design discussion for a network smart node and experimental test results for a key enabling technology: a high-temperature microcontroller with 32-bit computing architecture.

CHAPTER II

ARCHITECTURE

2.1 Overview

The literature describes a distributed engine control system as composed of a hierarchy of nodes, their interconnections, and attached sensors and actuators.[7] A top-level primary computer (FADEC) is responsible for supervisory tasks and the application of control laws to the engine system. An intermediate node type (data concentrator) performs data consolidation and interfaces subordinate devices to the FADEC. The lowest-level node type (smart node) interfaces directly with sensors and actuators.

Three network architecture types are identified as candidates for the distributed system. In the first type, all smart nodes (SNs) are connected to a single data concentrator (DC), which in turn connects to the primary computer via the aircraft's main data bus. In the second type, an additional data concentrator and main data bus are added to create a redundant system. The third system type is fully distributed, omitting data concentrators and connecting smart nodes directly to the main data bus.[8]

The proposed architecture shown in Figure 2.1 adopts the second (multiple data concentrator, multiple data bus) approach. For portability between engine models, the architecture is considered scalable to an arbitrary number of data concentrators and smart nodes. The system is effectively limited by available network resources (Section 2.2) and the computing resources available at each smart node (Section 2.3).

Despite being at the lowest tier of the network, the smart node is the most complex due to requirements for processing analog sensor signals (Section 2.4) and control of actuators by power electronics (Section 2.5).

The architecture is security-focused and includes mitigation for supply chain interdiction and counterfeiting (Section 2.6).

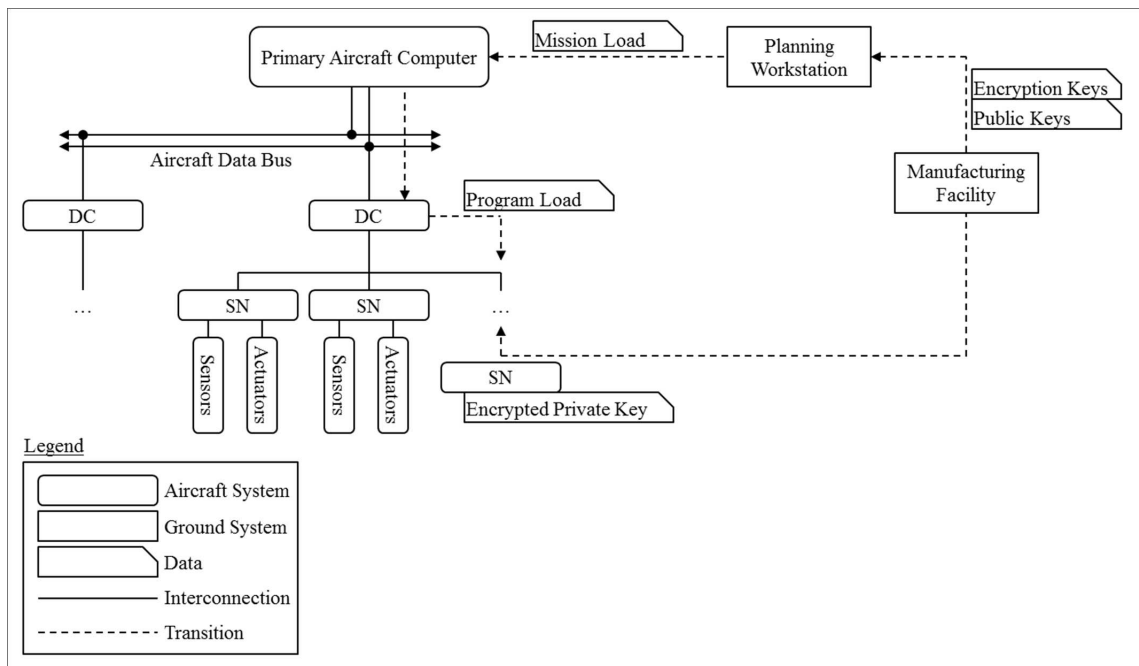


Figure 2.1: Distributed System Architecture

2.2 Networking

This section discusses options for providing networking resources to a distributed engine control architecture.

Although optical data bus standards such as ARINC 629 and MIL-STD-1773 have existed for some time, proliferation to publicly disclosed aircraft systems has been limited.

As compared to copper, optical fibers do offer benefits — typically lighter weight and resilience against electromagnetic threats — and it has been shown that such physical links may be suitable for a distributed engine control system.[9]

The design trade-off for an optical system is complexity. Every optical signal must be converted to a conductive interface, making non-optical interconnections the more simple and inexpensive solution. In consideration of these design trades, the discussion in this section will focus on options for copper-based networking.

RS-485. Otherwise known as TIA-485 or EIA-485, RS-485 is an electrical interface standard published by the Telecommunications Industry Association and most recently republished in 2012.[10] It is a serial communication link supporting multipoint communication between 32 devices and differential signaling over copper twisted-pair wire. The interface is controlled by a standard universal asynchronous receiver-transmitter (UART).

UART transceivers are ubiquitous in embedded systems, and vendors produce RS-485 integrated circuits capable of surviving extreme conditions.[11][12] RS-485 is electrically robust, however the UART protocol is not designed for advanced error correction or congestion control. Those functions must be implemented at a higher level in the embedded system.

CANbus. The Controller Area Network bus is a multi-master serial bus. Unlike a UART-based system, a CAN controller processes data from the bus as frames and performs functions such as cyclic redundancy checks and positive acknowledgement, effectively providing transport- and link-layer services to the microcontroller. The standard was originally developed for the automotive industry, and research for control of turbine engines was performed as early as 1999.[13]

Ethernet/BRAIN. In 2008, Honeywell Advanced Technology proposed the Braided Ring Availability Integrity Network (BRAIN) as a hybridized augmentation to aerospace and industrial Ethernets (e.g. ARINC 664, Ethernet Powerlink, EtherCAT, TTEthernet). BRAIN starts with a standard ring topology and adds skip links so that each node is connected to its neighbor and the next neighbor beyond. It also adds guardian capability and inherent tolerance against physical faults in the system.[14]

BRAIN is a flooding network, reducing latency to the order of bits. This is an advantage in a complex, high-gain realtime system. Multidrop serial networks (e.g. RS-485/UART) have the advantage of simplicity but cannot compete with Ethernet-based systems on a performance basis.

MIL-STD-1553B. The U.S. Department of Defense released MIL-STD-1553B in 1978 as a revision to the original standard published three years prior. The standard describes a serial data bus that was intended for avionics and has since proliferated to other systems. In common implementation, it features redundant differential interfaces capable of handling 30 devices. Dependability is achieved through redundancy; there are no accommodations at the network level. One malfunctioning node can compromise an entire bus. Additionally, MIL-STD-1553B is disadvantaged by the requirement for signaling at 18 V or greater.[15]

2.3 Computing

This section discusses compute resources for a node in a distributed engine control architecture. All of the microcontrollers listed are suitable for high-temperature (200° C) applications.

Honeywell HT83C51. The Honeywell HT83C51 is a legacy product based on the 8-bit Intel 8XC51FC design. The microcontroller runs at 16 MHz and is capable of addressing 64 kB of external program memory, 64 kB of external data memory, and 256 B of internal data memory.[16]

Vorago VA10800. The Vorago VA10800 and its derivative, the Prixarc ARC-1, are microcontrollers based on the 32-bit ARM Cortex-M0 core. The microcontroller runs at a maximum clock rate of 50 MHz and has 32 kB of data memory. Up to 128 kB of program memory are loaded from an external SPI ROM. The VA10800 is not equipped with a floating point unit, but floating point operations are supported by compiler libraries.[17]

Texas Instruments SM320F2812-HT. The Texas Instruments SM320F2812-HT is a robust, fully featured digital signal processing microcontroller. It is rated for approximately 1,000 hours of life at maximum operating temperature. If improved, it may be a good candidate for an engine control system, however it is not currently suitable for high reliability aerospace applications.[18]

2.4 Sensing

This section discusses sensors required to measure operating parameters of a turbine engine in aerospace applications (e.g. temperature, pressure, rotational rate). The analog front-end for a full sensor suite must provide a number of accommodations such as excitation voltages, constant current sources, and instrumentation amplifiers. In 2014, a fully integrated analog front-end was developed and tested by Honeywell International Inc.[19][20]

Strain/Pressure Gauge. Strain and pressure gauges are typically implemented in a Wheatstone bridge and require an excitation voltage for measurement. Engine operating

conditions are determined from the measured electrical properties of the sensor which vary according to environmental conditions.

Thermocouple. A thermocouple forms an electrical junction from two different conductors, and as a result of the thermoelectric effect, produces a voltage dependent upon the temperature of the junction. Because connecting the sensor to a voltage-measuring device introduces additional bimetal junction effects into the system, the “cold junction” temperature must also be measured in order to compensate the system and obtain a valid temperature reading.

Resistance Temperature Detector (RTD). An RTD provides temperature information according to the resistance-temperature characteristics of a monometallic material in the device. Similar to a strain gauge, the device is often incorporated into a resistor network.

Periodic Signal Processing. Periodic voltage producers, typically tachometers, employ devices such as the Hall effect sensor. In a high-noise environment, the sensor output must pass through an instrumentation amplifier prior to digital processing.

Linear Variable Differential Transducer (LVDT). An LVDT measures the linear position of a magnetic core encompassed by solenoidal coils of wire. The primary coil is excited by a sinusoidal alternating-current waveform, and phase and amplitude measurements are taken from secondary coils. An LVDT requires no mechanical connection to the device it is sensing, however because it will not operate with static excitation, it is electrically more complex in implementation than other sensors. The technique can also be applied for angular sensing.

2.5 Actuator Control

This section discusses control of electrically driven actuators in a turbine engine system. Actuators are required to operate a turbine engine and may include motors (e.g. DC, stepping, multi-phase) and solenoids. Some research has been performed in this area although not with emphasis on drive electronics.[21]

High-temperature power MOSFETs, gate drivers, and supporting devices have been developed by Belgian company Cissoid and tested in cooperation with the National Aeronautics and Space Administration (NASA).[22] Cissoid's gate drivers are fabricated with silicon-on-insulator (SOI) techniques, and power devices are manufactured with silicon carbide (SiC) and can provide $V_{DS} \leq 1,200$ V and $R_{ON} = 150$ m Ω at 225° C junction temperature.[23]

2.6 Security

This section proposes a security paradigm for the distributed architecture. The focus is on mitigating risk of smart node compromise by interdiction.

At manufacturing, a private key is created, encrypted, and written to a small block of nonvolatile memory in the smart node. No other data is written to the smart node. It is physically sealed and shipped to the aircraft for installation.

The corresponding encryption key, public keys, serial number, and program load are transmitted to the aircraft operating location via secure Internet. That information is loaded to the aircraft data cartridge during mission preparation, and it is read by the primary aircraft computer during system start.

A hybrid cryptographic communication system is employed across the distributed engine control network. Each data concentrator and its subordinate smart nodes constitute a security domain with a distinct set of public and private keys.

Private keys are stored in an encrypted state in the non-volatile memory of each network device. Upon system start, encryption keys and public keys are distributed to data concentrators and smart nodes from the primary aircraft computer.

When initialized, each data concentrator consolidates: (1) a smart node program, (2) a smart node "private key encryption key", and (3) public keys for the security domain. When consolidation is complete, the data concentrator writes the consolidated program load to each smart node's volatile memory. The data concentrator is able to control the reset function of a subordinate smart node.

Upon reception of the consolidated payload, the smart node starts its boot process. The smart node runs the consolidated program which: (1) attempts to decrypt its encrypted private key and upon successful decryption (2) starts the programmed functions of the smart node including communication on the network. If the private key or private key encryption key are inauthentic, the smart node will be rejected by the network.

Upon detection of an inauthentic smart node, the data concentrator may reinitialize the smart node or hold the smart node in the reset state.

Asymmetric cryptography such as the proposed public/private key system is computationally expensive but valuable for device authentication.[24][25][26] Therefore, once the system has established the authenticity of a device, each security domain switches to a symmetric encryption mode.

CHAPTER III

SMART NODE

3.1 Overview

This chapter discusses the design of a smart node embedded system in consideration of the research presented in Chapter II.

The proposed smart node described by Figure 3.1 provides capability for interface with a multi-node network, precision analog-to-digital-conversion, and control of a multiplexed analog front-end and other downstream devices. Components are selected for temperature rating and life consistent with aerospace applications.

The smart node is composed of a four-layer mixed-technology circuit board assembly and accompanying software.

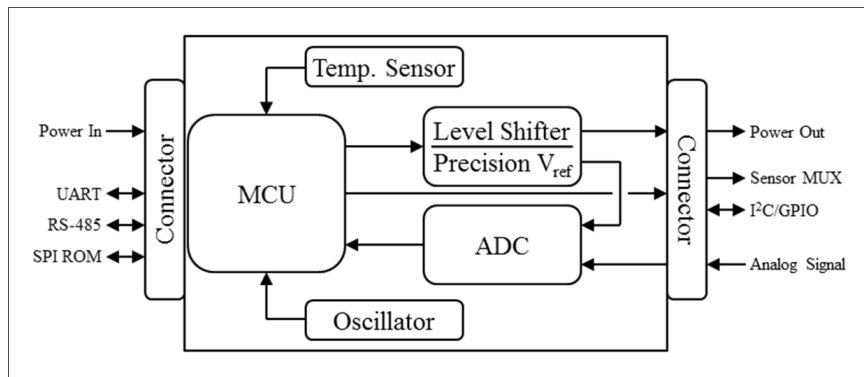


Figure 3.1: Smart Node Functional Diagram

Table 3.1: Definition of Terms in Figure 3.1

Term	Definition
Power In	All electrical power necessary to operate the system.
UART	Universal Asynchronous Receiver-Transmitter. A single-ended serial bus.
RS-485	RS-485 electrical interface. A differential serial bus.
SPI ROM	Device programming interface.
Connector	Connection to smart node circuit board.
MCU	Microcontroller Unit.
Temp. Sensor	Digital temperature sensor.
Oscillator	Electronic oscillator.
Level Shifter	Logic level shifter.
Precision V_{ref}	Precision voltage reference for ADC.
ADC	Analog-to-Digital Converter.
Power Out	Pass-through electrical power necessary to operate the analog front-end.
Sensor MUX	Lines for control of multiplexer at analog front-end.
I ² C/GPIO	Inter-Integrated Circuit bus and/or general purpose I/O.
Analog Signal	Analog signal input.

3.2 Microcontroller

The Prixarc ARC-1 shown in Figure 3.2 is an ARM Cortex-M0 microcontroller (MCU) designed for applications with temperatures from -55°C to 200°C . It is hardened against low-impedance pathing (latch-up) at the maximum rated operating temperature. The device is fabricated through a commercially available process that has been successfully demonstrated on other devices.[27][28] The fabrication process also has radiation hardening effects. The ARC-1 is packaged in 68-pin Ceramic Leadless Chip Carrier (CLCC).

The Cortex-M0 processor core employs the ARMv6-M architecture which includes 32-bit hardware multiplication capability. As implemented in the ARC-1, the core is supported

by a full complement of input/output hardware including UART, SPI, and I²C. 1.5 V and 3.3 V supply voltages are required.

3.3 Oscillator

The Vectron PX-7021 oscillator[29] shown in Figure 3.3 is designed for applications with temperatures from -55°C to 230°C . The part employed in the smart node is a semicustom version of the PX-7021 with 50 MHz output. It is composed of a crystal resonator with supporting oscillator circuit and output buffer.

The PX-7021 is designed to survive 3,000 G shock and 36 G sustained vibration in compliance with MIL-STD-883.

3.4 Operational Amplifiers and Precision Voltage Reference

The Cissoid CHT-RUBY[30] shown in Figure 3.4 is a general-purpose triple operational amplifier and bandgap voltage reference designed for applications with temperatures from -55°C to 225°C . The CHT-RUBY serves as a 3.3/5.0 V logic level shifter and precision voltage reference.

3.5 Analog to Digital Converter

The Cissoid CHT-ADC10[31] shown in Figure 3.5 is a successive-approximation ADC based on an R-2R network and designed for applications with temperatures from -55°C to 225°C .



Figure 3.2: Prixarc ARC-1



Figure 3.3: Vectron PX-7021



Figure 3.4: Cissoid CHT-RUBY

3.6 Differential Transceiver

The Texas Instruments SN65HVD11-HT[12] shown in Figure 3.6 is a 3-state differential line driver and differential input line receiver designed for high-temperature applications.

3.7 Digital Temperature Sensor

The Texas Instruments LM95172[32] shown in Figure 3.7 is an integrated digital output temperature sensor with pseudo-SPI interface. It is designed for applications with temperatures from -55°C to 200°C .

3.8 Connectors

The smart node PCB is configured for Min-e-Con BR-series connectors[33] shown in Figure 3.8. The connectors are rated for -55°C to 200°C .

3.9 Software

Software for the smart node runs on the FreeRTOS operating system. The operating system has been ported to the Cortex-M0 core and ARC-1 microprocessor's peripheral hardware.

The system operates in pre-emptive mode with no advanced memory management. By keeping the system relatively simple and optimized, the complete program load is limited to less than 32 kB.



Figure 3.5: Cissoïd CHT-ADC10

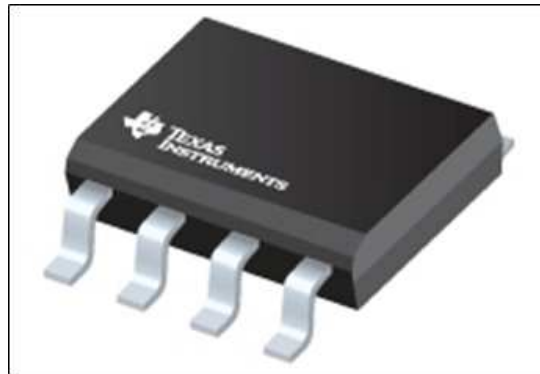


Figure 3.6: Texas Instruments SN65HVD11-HT

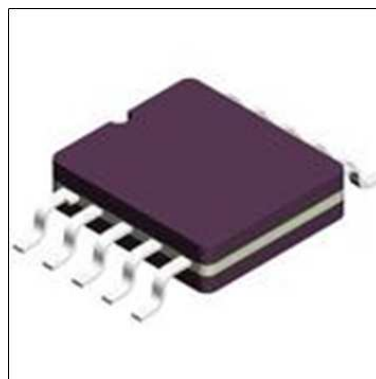


Figure 3.7: Texas Instruments LM95172

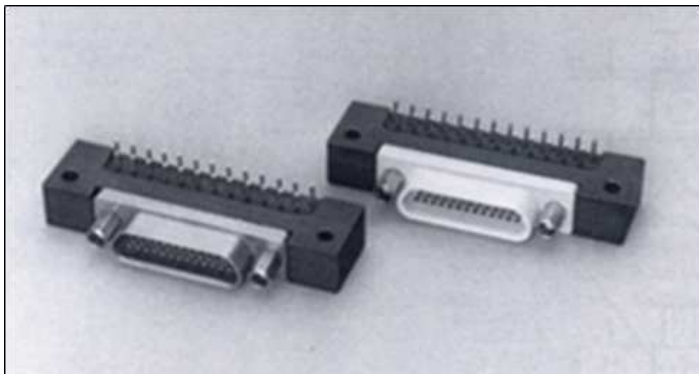


Figure 3.8: Min-e-Con BR-Series Connectors

3.10 Sensor Sampling and Signal Processing

The system is capable of rapidly sampling multiple sensors in arbitrary sequence. Sensor polling rate is limited by analog-to-digital conversion time, multiplexer switching in the analog front-end, and settling time after a sensor has been switched in to be sampled.

The ARC-1 microprocessor has a 32 kB system memory space and is capable of performing 32-bit arithmetic operations in hardware, making digital signal processing efficient compared to more limited 8- and 16-bit architectures.

3.11 Data Transmission and Structure

Data is transmitted by the microprocessor's UART which can be configured for point-to-point or multi-drop modes. Each 16-byte frame contains a destination address, source address, command byte, five 16-bit samples, an 8-bit checksum, and two termination bytes as shown in Table 3.2. In multi-drop mode, the smart node can communicate with up to 255 other nodes including an upstream data concentrator.

The smart node can transmit at up to 2,000,000 baud providing a theoretical raw frame transmission rate greater than 15 kHz. In practice, the frame transmission rate will be lower due to protocol overhead as shown in Equation 3.1. The UART is configured for one start bit, an 8-bit word, one parity bit, and one stop bit.

$$\begin{aligned}
 \text{effective frame rate} &= \frac{8}{11} \frac{2000000 \text{ baud}}{(8 \text{ bits/byte})(16 \text{ bytes/frame})} \\
 &= 11,364 \text{ frames/second}
 \end{aligned} \tag{3.1}$$

Assuming no competition for the bus and maximum utilization by the microcontroller, the available data throughput supports five sensors at a reporting rate greater than 10 kHz. If implemented in multidrop mode, throughput will naturally decrease due to the effects of time division multiplexing.

Table 3.2: Transmission Frame Structure

Byte	Description
0	Destination Address
1	Source Address
2	Command Byte
3	Sensor 1, Byte 1
4	Sensor 1, Byte 2
5	Sensor 2, Byte 1
6	Sensor 2, Byte 2
7	Sensor 3, Byte 1
8	Sensor 3, Byte 2
9	Sensor 4, Byte 1
10	Sensor 4, Byte 2
11	Sensor 5, Byte 1
12	Sensor 5, Byte 2
13	Checksum
14	Termination Byte 1
15	Termination Byte 2

CHAPTER IV

EXPERIMENTAL TESTING

Development of a turbine engine control system requires hardware-in-the-loop testing.[34] This chapter discusses extreme-environment experimentation on a 32-bit microcontroller, the central component of a network smart node. Testing of the microcontroller was performed on a representative circuit board assembly. Two experiments were conducted.

4.1 Ambient Thermal Environment

The first experiment, shown in Figure 4.1, tests an ambient thermal environment and closed-loop control system in which the smart node is placed in an environmental chamber, and a desktop computer workstation emulates a data concentrator node. Temperature sensing is performed by a resistance temperature detector (RTD) that is sampled, processed, and transmitted from the smart node to the data concentrator. The data concentrator implements a hysteresis control system that transmits temperature-change commands to the environmental chamber controller. Limits for the control loop are -40°C and $+160^{\circ}\text{C}$.

Temperature data, shown in Figure 4.2, was sampled from the smart node at 0.2 Hz for 12 hours producing 8,640 data points. The transmission error rate from the smart node to the data concentrator was 0.0%.

To test smart node behavior over the temperature range, a square wave with limits 0 V and 1 V was provided from a signal generator for sampling. A plot of the 1 V samples is provided in Figure 4.3. A plot of the 0 V samples is provided in Figure 4.4.

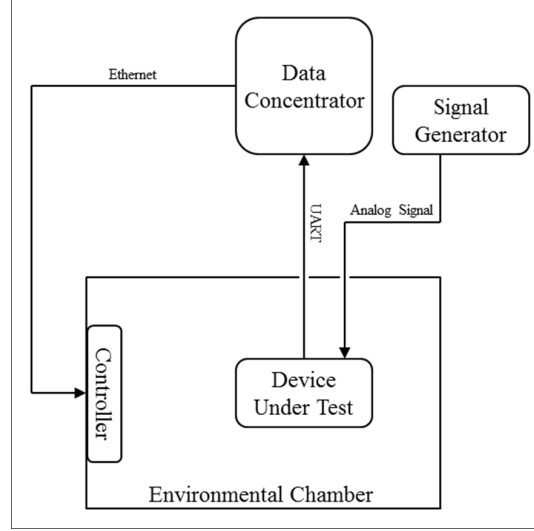


Figure 4.1: Experiment Diagram: Ambient Thermal Environment

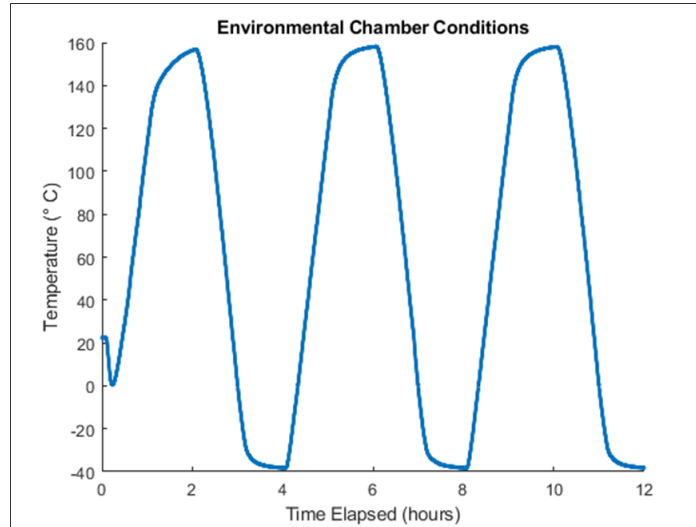


Figure 4.2: Experimental Results: Ambient Thermal Environment

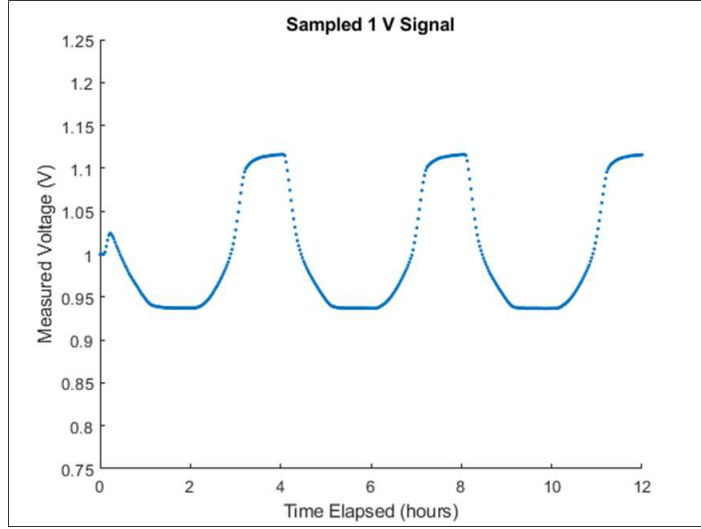


Figure 4.3: Experimental Results: 1 V

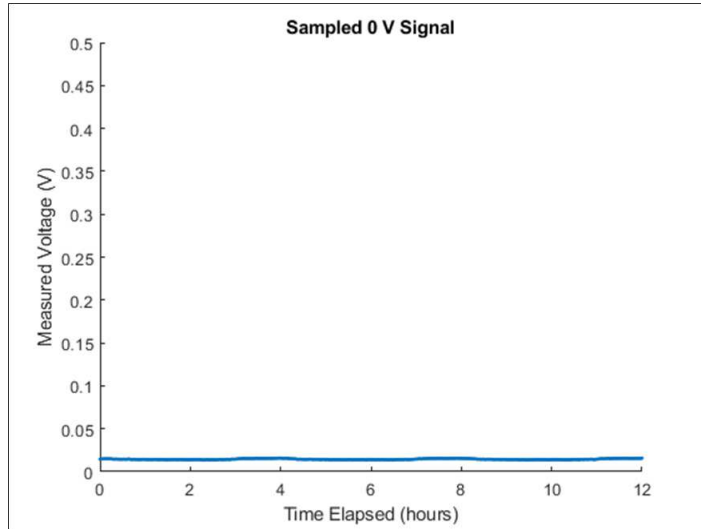


Figure 4.4: Experimental Results: 0 V

4.2 Radiant Thermal Environment

The second experiment, shown in Figure 4.5, tests a radiant thermal environment in which the smart node is suspended over a planar thermal source. A thermocouple is placed

on the surface of the microcontroller and sampled by a National Instruments VirtualBench instrument. Cold-junction compensation is performed in post-processing. As in the first experiment, temperature information is sampled from an RTD, processed, and transmitted from the smart node to the data concentrator. Additionally, a 5 V DC voltage source is sampled by the smart node.

Data, shown in Figures 4.6 and 4.7, was sampled at 0.5 Hz for 1,000 hours producing 1.8 million data points per source. Data plotted in Figure 4.7 was processed through a moving average filter with a window size of 5 minutes ($n = 150$).

The transmission error rate from the smart node to the data concentrator was 0.000002%. The maximum microprocessor surface temperature was 197.7° C. There were four thermal shock events with average $\Delta T = 30^\circ \text{ C}$ and rate 5° C per minute.

Thermocouple data was post-processed with a rational polynomial function that improves error and computational efficiency when compared to the published National Institute of Standards and Technology 9th-order polynomial formula.[35] The function and associated Matlab code are provided in Appendix A.

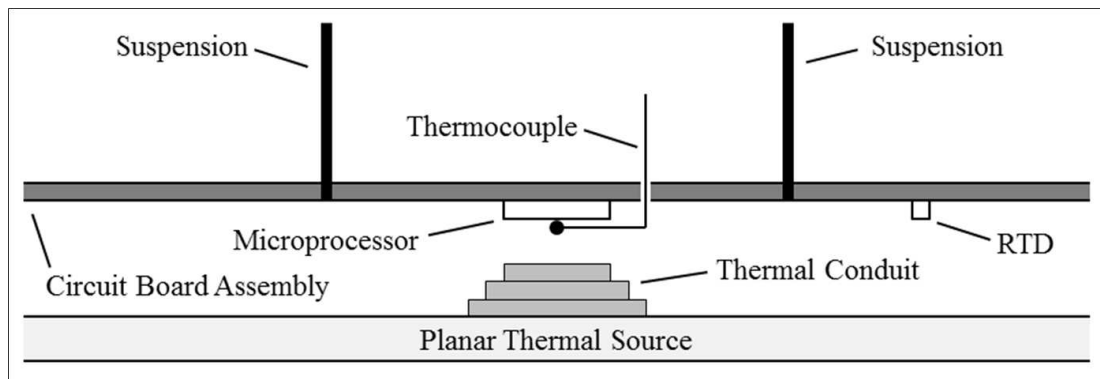


Figure 4.5: Experiment Diagram: Radiant Thermal Environment

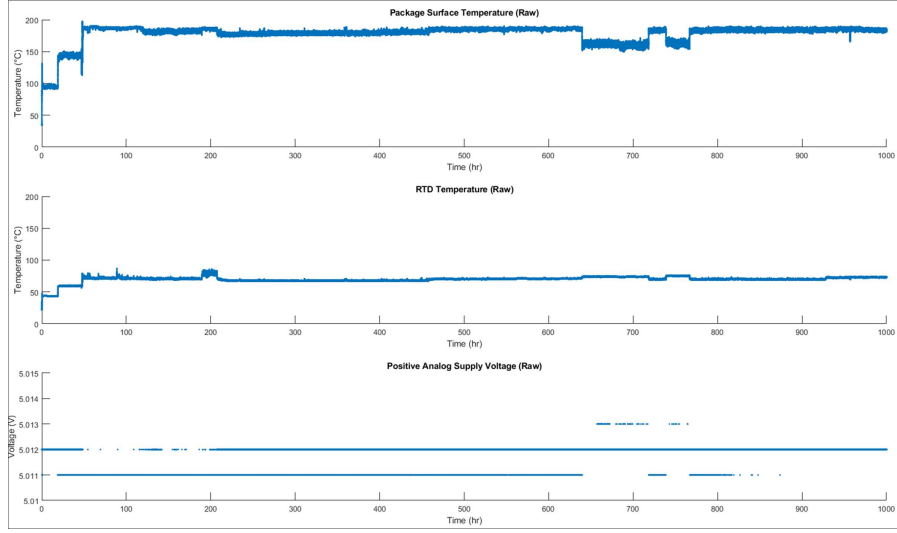


Figure 4.6: Experimental Results (Raw): Radiant Thermal Environment

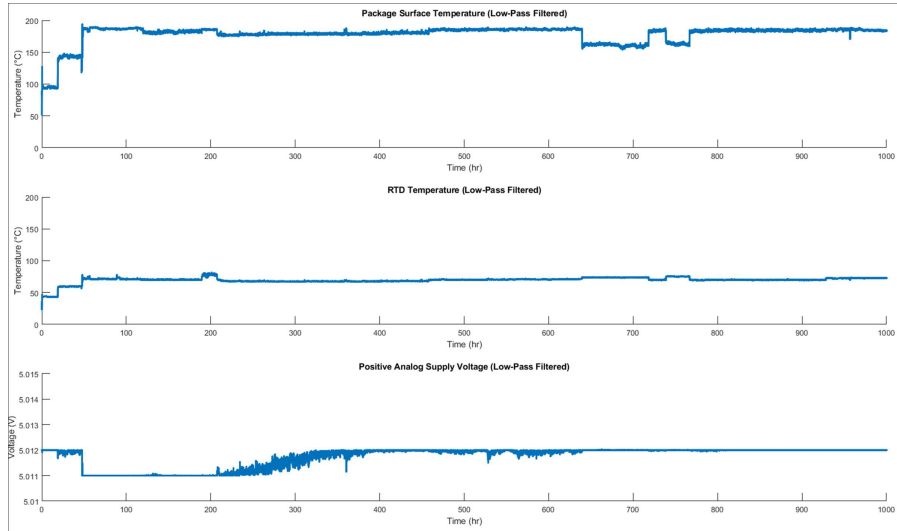


Figure 4.7: Experimental Results (Low-Pass Filtered): Radiant Thermal Environment

CHAPTER V

CONCLUSION

Distributed control of turbine engine systems has been an area of active research for greater than 20 years. Early work focused on proof-of-concept systems that were not robust or practical for real aerospace applications. Since then, there have been advances in high temperature semiconductors, microcontroller design, network topologies, and system integration. This thesis has presented an overall system architecture, design of a network smart node, and results from experimentation with a 32-bit microcontroller at extreme temperatures.

This research provides a baseline for a modernized turbine engine control system. Future work will require development and integration of the complete networked system, an analog front-end, sensors, and actuators.

BIBLIOGRAPHY

- [1] J. Foster, "Design Analysis of Messerschmitt Me-262 Jet Fighter: Part II - The Power Plant," *Aviation*, vol. 44, no. 11, 1945.
- [2] B. R. Rich and L. Janos, *Skunk Works*. New York: Back Bay Books, 1994.
- [3] L. C. Jaw and S. Garg, "Propulsion control technology development in the United States: A historical perspective," *NASA Technical Memorandum*, no. 213978, 2005.
- [4] A. Behbahani and B. Tulpule, "Perspective for distributed intelligent engine controls of the future," *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2010.
- [5] D. Culley, "Transition in gas turbine control system architecture: Modular, distributed, and embedded," *NASA Technical Memorandum*, no. 216806, 2010.
- [6] A. Behbahani, "Adaptive distributed intelligent control architecture for future propulsion systems," *AFRL Technical Memorandum*, no. 228, 2007.
- [7] J. D. Castro, C. Palmer, and A. Behbahani, "Meeting the requirements of distributed engine control via decentralized, modular smart sensing," *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Joint Propulsion Conferences*, 2010.
- [8] J. DeCastro, L. Tang, C. Byington, and D. Culley, "Analysis of decentralization and fault-tolerance concepts for distributed engine control," *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2009.
- [9] R. K. Belapurkar, R. K. Yedavalli, and B. Moslehi, "Stability of fiber optic networked decentralized distributed engine control under time delays," *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2009.
- [10] *TIA-485: Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*. Telecommunications Industry Association (TIA).
- [11] "Holt HI-4853 Datasheet," 2018. [Online]. Available: <http://holtic.com/products/3063-hi-4853.aspx>
- [12] "Texas Instruments SN65HVD11-HT Datasheet," 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/sn65hvd11-ht.pdf>

- [13] H. Thompson, B. Pérez, D. Lee, D. Ramos-Hernandez, P. Fleming, and C. Legge, "A CANbus-based safety-critical distributed aeroengine control systems architecture demonstrator," *Microprocessors and Microsystems*, vol. 23, no. 6, pp. 345–355, 1999.
- [14] B. Hall, M. Paulitsch, D. Benson, and A. Behbahani, "Jet engine control using ethernet with a BRAIN (postprint)," *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2008.
- [15] *MIL-STD-1553B: Aircraft Internal Time Division Command/Response Multiplex Data Bus*. U.S. Department of Defense, 1978.
- [16] "Honeywell HT83C51 Datasheet," 2018. [Online]. Available: <https://aerospace.honeywell.com/en/products/navigation-and-sensors/high-temperature-microelectronics>
- [17] "Vorago VA10800 Datasheet," 2018. [Online]. Available: <https://www.voragotech.com/vorago-products>
- [18] "Texas Instruments SM320F2812-HT Datasheet," 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/sm320f2812-ht.pdf>
- [19] B. W. Ohme, M. R. Larson, B. Tulpule, and A. Behbahani, "Characterization of circuit blocks for configurable analog-front-end," *IMAPS International Conference on High Temperature Electronics(HiTEC)*, 2014.
- [20] B. Tulpule, A. Behbahani, B. Ohme, M. Larson, J. Gerety, and A. Steines, "A system on chip (soc) asic chipset for aerospace and energy exploration applications," *IMAPS International Conference on High Temperature Electronics(HiTEC)*, 2014.
- [21] B. Tulpule and A. R. Behbahani, "System on chip (SOC) ASIC chipset for smart actuators in distributed propulsion systems," *IMAPS International Conference on High Temperature Electronics(HiTEC)*, 2016.
- [22] R. Patterson and A. Hammoud, "Performance of an SOI boot-strapped full-bridge MOSFET driver, type CHT-FBDR, under extreme temperatures," *Technical Memorandum, NASA Electronic Parts and Packaging Program*, 2009.
- [23] "Cisoid Product Catalog," 2018. [Online]. Available: <http://www.cisoid.com/>
- [24] A. Ramachandran, Z. Zhou, and D. Huang, "Computing cryptographic algorithms in portable and embedded devices," in *2007 IEEE International Conference on Portable Information Devices*, May 2007, pp. 1–7.
- [25] C. T. R. Hager, S. F. Midkiff, J. . Park, and T. L. Martin, "Performance and energy efficiency of block ciphers in personal digital assistants," in *Third IEEE International Conference on Pervasive Computing and Communications*, March 2005, pp. 127–136.

- [26] R. Cramer and V. Shoup, “Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack,” *SIAM Journal of Computing*, vol. 33, no. 1, pp. 167–226, 2003.
- [27] R. Fuller, W. Morris, J. Gwin, R. Lowther, D. Gifford, and P. Young, “Performance of sst 16mb sram,” *Hardened Electronics and Radiation Technology (HEART) Conference*, 2009.
- [28] R. Fuller, W. Morris, D. Gifford, R. Lowther, J. Gwin, J. Salzman, D. Alexander, and K. Hunt, “Hardening of Texas Instruments’ VC33 DSP,” in *2010 IEEE Radiation Effects Data Workshop*, 2010.
- [29] “Vectron PX-702 Datasheet,” 2018. [Online]. Available: <https://www.vectron.com/products/hitemp/px-702.pdf>
- [30] “Cisoid CHT-RUBY Datasheet,” 2018. [Online]. Available: <http://www.cisoid.com/files/files/products/gemstone/CHT-RUBY.pdf>
- [31] “Cisoid CHD-ADC10 Datasheet,” 2018. [Online]. Available: <http://www.cisoid.com/files/files/products/river/CHT-ADC10.pdf>
- [32] “Texas Instruments LM95172 Datasheet,” 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm95172-q1.pdf>
- [33] “Min-e-Con BR-Series Datasheet,” 2018.
- [34] D. Culley, R. Thomas, and J. Saus, “Integrated tools for future distributed engine control technologies,” *NASA Technical Memorandum*, no. 217883, 2013.
- [35] P. Clifford, “Efficient thermocouple calibration and measurement,” 2018. [Online]. Available: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/temperature-measurement/thermocouple/calibration-table>

APPENDIX A

Polynomial Function and MATLAB Code for Post-Processing of Thermocouple Data

$$T - T_0 = \frac{p_1(v - v_0) + p_2(v - v_0)^2 + p_3(v - v_0)^3 + p_4(v - v_0)^4}{1 + q_1(v - v_0) + q_2(v - v_0)^2 + q_3(v - v_0)^3}$$

$$T = T_0 + \frac{(v - v_0)(p_1 + (v - v_0)(p_2 + (v - v_0)(p_3 + p_4(v - v_0))))}{1 + (v - v_0)(q_1 + (v - v_0)(q_2 + q_3(v - v_0)))}$$

```
function [T] = typeKtemp(Vm, Tcj)

%% This function calculates temperature from the measured voltage of a
% type K thermocouple including cold junction compensation.
%
% Junction temperature data is from the NIST ITS-90 table.
%
% All units are mV and deg .
%
% The junction functions and coefficients are from the Mosaic Industries
% rational polynomial function approximation:
%
```



```

%% Initialize

% clear all; close all; clc;

%% Cold Junction Calculation

% Coefficients for -20 to 70 deg C (-0.778 to 2.851 mV) cold junction.

To = 2.5000000E+01;
Vo = 1.0003453E+00;
p1 = 4.0514854E-02;
p2 = -3.8789638E-05;
p3 = -2.8608478E-06;
p4 = -9.5367041E-10;
q1 = -1.3948675E-03;
q2 = -6.7976627E-05;

% Calculate compensation voltage.

P = (Tcj - To)*(p1 + (Tcj - To)*(p2 + (Tcj - To)*(p3 + p4*(Tcj - To))));
Q = 1 + (Tcj - To)*(q1 + q2*(Tcj - To));
Vcj = Vo + (P/Q);

% fprintf('Cold junction voltage: %f mV\n',Vcj);

%% Hot Junction Calculation with Cold Junction Compensation

% Hot junction voltage (mV)

```

```

Vtc = Vm + Vcj;

if (Vtc >= -6.404) && (Vtc < -3.554)

    bin = 1;

elseif (Vtc >= -3.554) && (Vtc < 4.096)

    bin = 2;

elseif (Vtc >= 4.096) && (Vtc < 16.397)

    bin = 3;

elseif (Vtc >= 16.397) && (Vtc < 33.275)

    bin = 4;

elseif (Vtc >= 33.275) && (Vtc <= 69.553)

    bin = 5;

else

    T = NaN;

    return

end

% Coefficients for each voltage range.

%

% bin:  1                2                3

% mV:   -6.404 to -3.554 | -3.554 to 04.096 | 04.096 to 16.397 |

% deg:  -250   to -100   | -100   to 100    | 100    to 400    |

%

% bin:  4                5

% mV:   16.397 to 33.275 | 33.275 to 69.553

```

```

% deg: 400      to 800      | 800      to 1200
%
coeff = [-1.2147164E+02      -8.7935962E+00      3.1018976E+02
         6.0572562E+02      1.0184705E+03;                                % To
        -4.1790858E+00      -3.4489914E-01      1.2631386E+01
         2.5148718E+01      4.1993851E+01;                                % Vo
         3.6069513E+01      2.5678719E+01      2.4061949E+01
         2.3539401E+01      2.5783239E+01;                                % p1
         3.0722076E+01      -4.9887904E-01      4.0158622E+00
         4.6547228E-02      -1.8363403E+00;                                % p2
         7.7913860E+00      -4.4705222E-01      2.6853917E-01
         1.3444400E-02      5.6176662E-02;                                % p3
         5.2593991E-01      -4.4869203E-02      -9.7188544E-03
         5.9236853E-04      1.8532400E-04;                                % p4
         9.3939547E-01      2.3893439E-04      1.6995872E-01
         8.3445513E-04      -7.4803355E-02;                                % q1
         2.7791285E-01      -2.0397750E-02      1.1413069E-02
         4.6121445E-04      2.3841860E-03;                                % q2
         2.5163349E-02      -1.8424107E-03      -3.9275155E-04
         2.5488122E-05      0.0];                                % q3

```

```
To = coeff(1, bin);
```

```
Vo = coeff(2, bin);
```

```
p1 = coeff(3, bin);
```

```
p2 = coeff(4, bin);
```

```
p3 = coeff(5, bin);
```

```
p4 = coeff(6, bin);
```

```
q1 = coeff(7, bin);
```

```
q2 = coeff(8, bin);
```

```
q3 = coeff(9, bin);
```

```
P = (Vtc - Vo)*(p1 + (Vtc - Vo)*(p2 + (Vtc - Vo)*(p3 + p4*(Vtc - Vo))));
```

```
Q = 1 + (Vtc - Vo)*(q1 + (Vtc - Vo)*(q2 + q3*(Vtc - Vo)));
```

```
T = To + (P/Q);
```