1998

Data logging multiple aviation sensors using a Motorala [sic] HC11 microcontroller

Andrew James Wooding
Edith Cowan University

USE OF THESIS

Data Logging Multiple Aviation Sensors Using a Motorala HC11 Microcontroller

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Engineering (Electronic Systems)

Andrew James Wooding
0951380

November 1998

Principal Supervisor: Dr. Steven Hinckley
Supporting Supervisor: Mike Wetton

Faculty of Communications, Health and Science
School of Engineering and Mathematics
Edith Cowan University
Western Australia

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature

[redacted]

Date

8-4-1999

Data Logging Multiple Aviation Sensors Using a Motorola HC11 Microcontroller

ABSTRACT: This project is part of the research and development of a microprocessor based integrated instrumentation system for light aircraft. It involves interfacing a Motorola HC11 microcontroller with several sensor inputs. The project system is designed to data log battery voltage, cylinder head temperature, ambient temperature and engine RPM.

By Andrew Wooding

Supervisor: Steven Hinckley

CONTENTS

CHAPTER 1: Introduction

Instrumentation in present day light aircraft is achieved by many discrete systems operating independently. Unfortunately this approach to aircraft instrumentation has disadvantages that limit the functionality of the aircraft instrument system. A better approach, that allows greater functionality, would be to use microprocessor technology to create an integrated instrument system. This project is an essential step in the development of an advanced integrated light aircraft instrumentation system.

1.1 Present Day versus the Integrated Approach

The present day approach to aircraft instrumentation is depicted in figure 1.1. In this approach each quantity being measured is handled by independent systems. A quantity is measured by a sensor. An interface then converts the signal from the sensor to one that is readable by the display device for the quantity being measured. Finally useful information is supplied to the pilot. Unfortunately several problems in the present approach limit the functionality of light aircraft instrumentation systems

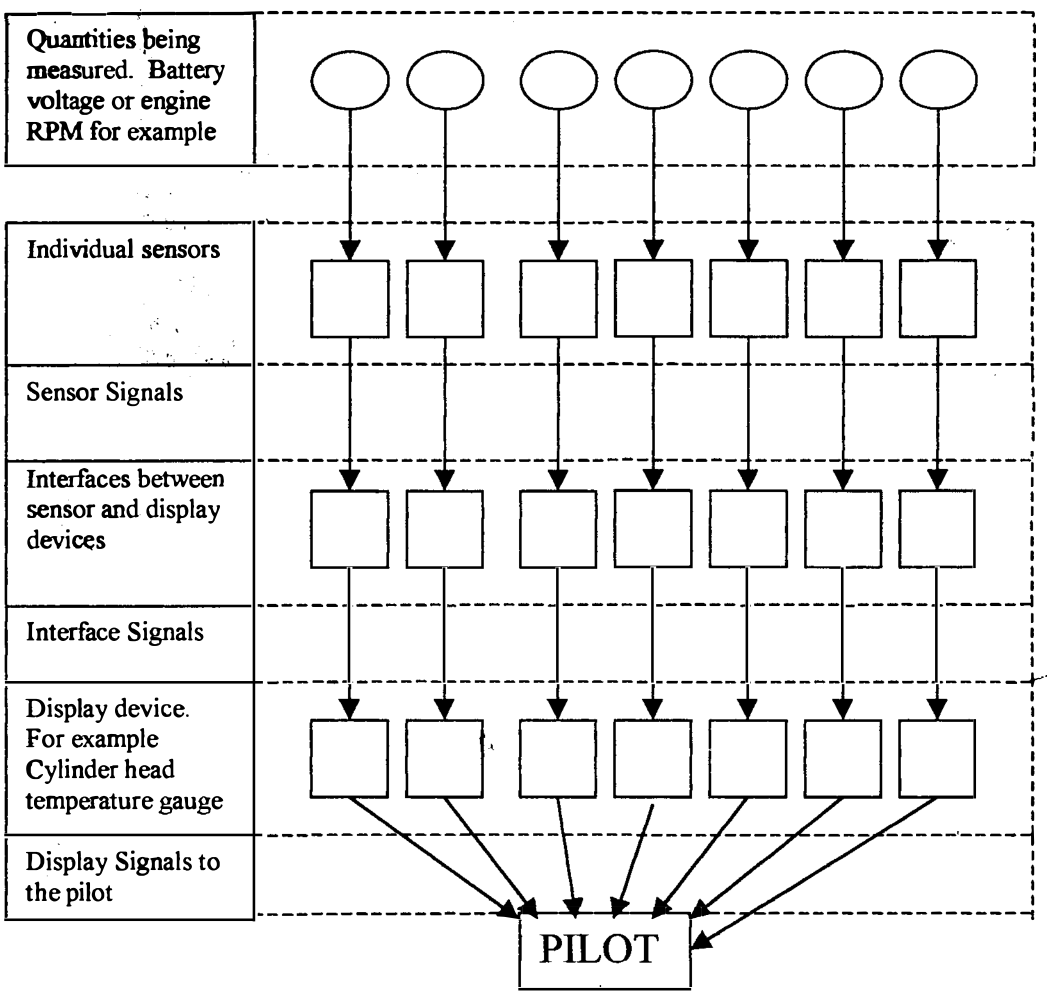


Figure 1.1 The present day approach to light aircraft instrumentation

The first of these problems is the increase in weight that each new instrument function brings to the aircraft. The increase in weight results because each new function needs its own sensing system, display and interfacing between the display and the sensing system. Light aircraft performance is directly related to the aircraft weight. Aircraft that weigh less consume less fuel and are more maneuverable. Any increase in the functionality of the aircraft instrumentation system comes at a cost of increased aircraft weight and therefore poorer aircraft performance.

The second problem limiting instrument system functionality in light aircraft is the amount of space needed to implement each new instrument function. Space on a light aircraft is finite and precious. As a result there is a limited amount of space that can be allocated to aircraft instrument systems. The limited space available and the space required to implement each instrument function using present methods limits the functionality of the light aircraft instrument system.

The third problem that limits the functionality of present day instrument systems is the high cost of implementing each instrument function. The high cost results because each instrument function requires not only sensing and interfacing equipment but also a display device. The cost of aircraft must be minimized in order for the aircraft manufacturer to remain competitive. The need to remain competitive and the high cost of implementing each instrument function limit the functionality of light aircraft instrument systems.

The integrated approach to aircraft instrumentation is depicted in Figure 1.2 In this approach all of the individual display units that were required in the old approach are not needed. Instead they are replaced by a small number of input processors, a system bus, a single output processor and a display unit. Since a single input processor will be able to handle approximately eight inputs the number of components required to implement each instrument function is reduced. This reduces the aircraft weight increase involved in implementing instrument function. The reduction in components required also reduces the cost and space used in implementing each instrument function. The integrated approach is an improvement on the present day approach because it allows increased instrument system functionality by reducing the problems that limit the functionality of light aircraft instrument systems.

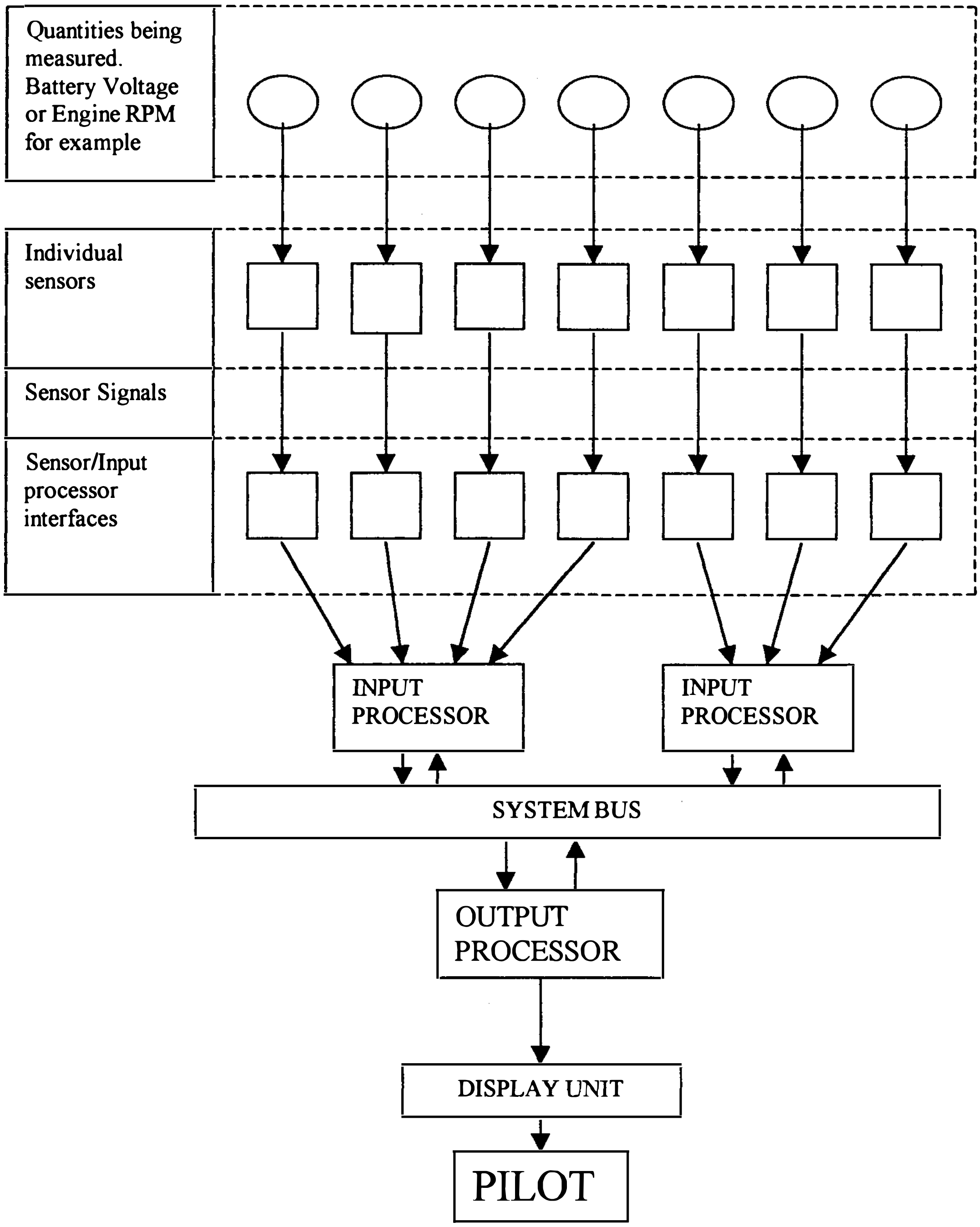


Figure 1.2 The integrated approach to light aircraft instrumentation

1.2 Research and Development requirements

Before an integrated light aircraft system can be implemented much research and development work must be performed. Areas where work is required include;

- The development of communication protocols for all devices attached to the system bus.
- The development of software routines for the output processor so that it may control the display unit.
- Interfacing input processors with several sensor inputs

This project concentrates on interfacing a single processor with several sensor inputs.

1.3 Project Aims

The aim of this project is to develop a data logging system that may be used in a light aircraft. The data logging system will be based on the Motorola HC11 microcontroller. The system will analyze signals from several sensors and convert the signals into useful information. The data logging system will perform a very similar role to a single input processor, its associated sensors and sensor interfaces in the integrated approach to aircraft instrumentation. The major difference will be that this data logging system will be a single stand alone system.

1.4 Project Scope

Due to the time allocated for this project it is concerned with only four sensor inputs. The inputs are;

- A battery voltage sensing circuit
- A K type thermocouple for determining cylinder head temperature
- An LM335 temperature sensor for measuring ambient temperature
- Engine RPM

The entire datalogging system from a dataflow perspective is depicted in figure 1.3.

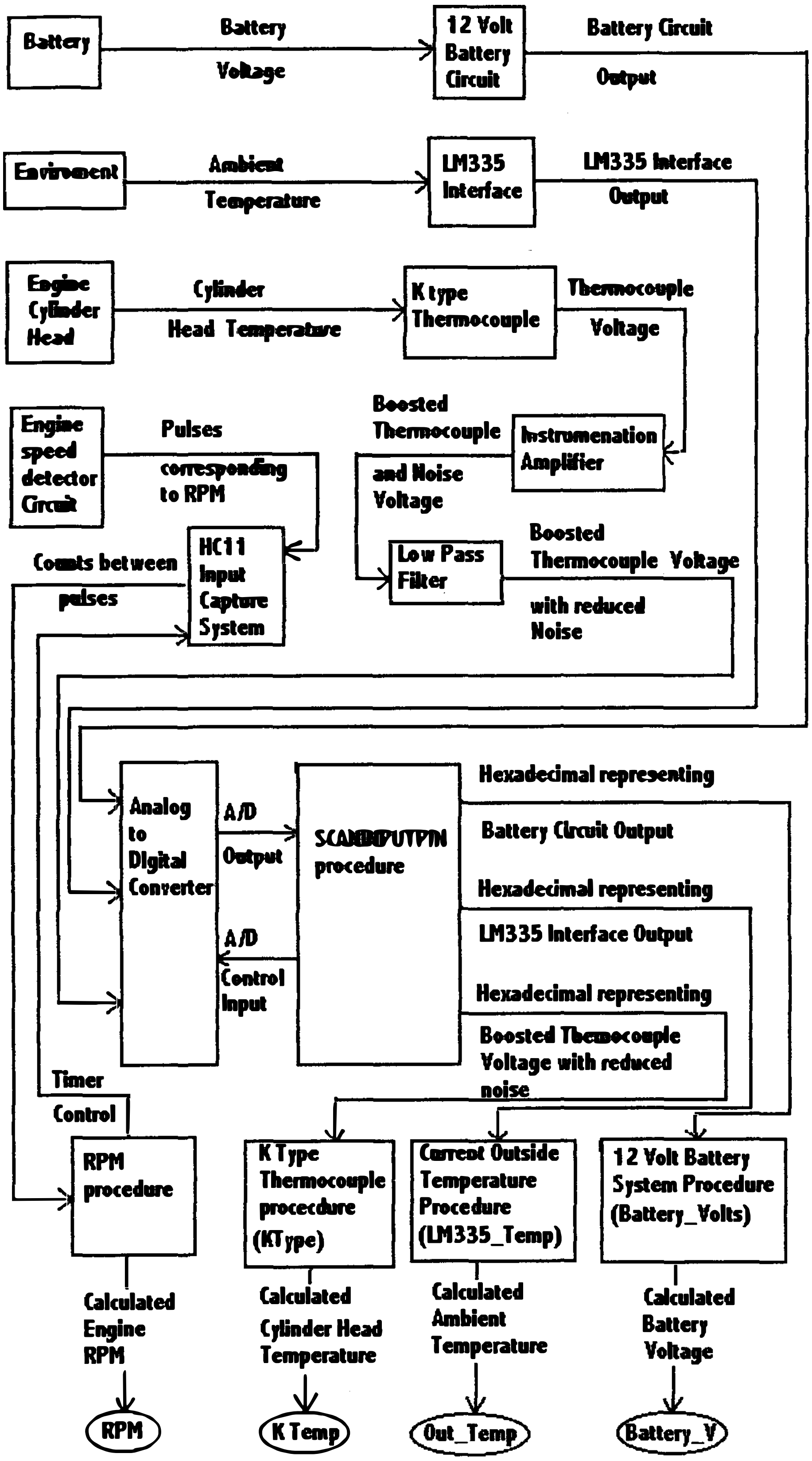


Figure 1.3 The project system

1.4.1 Why Battery Voltage ?

The battery voltage was chosen to be the first input for two reasons. The first reason is that the aircraft battery voltage gives vital information about the health of the aircraft's electrical system. Since the microprocessor based integrated aircraft system will rely on the aircraft's electrical system to supply it with power it is appropriate that a sensor that monitored the health of the aircraft's electrical system be given a high priority. The second reason for choosing the battery voltage as the first input to be developed was that the battery signal being measured is already electrical. Most other quantities being measured, for example temperature, are not electrical and therefore require more complicated interfacing.

1.4.2 Why cylinder head temperature measurement ?

Cylinder head temperature warns the pilot about potential problems with the aircraft engine. It also provides the pilot with information needed to operate the engine correctly. Because knowledge of engine cylinder head temperature is so important it was chosen as the third input to be developed in this project.

1.4.3 Why ambient temperature measurement ?

Ambient temperature measurement was required for two reasons. The first reason for needing ambient temperature measurement is due to the third input in the project involving a thermocouple. Thermocouples, alone, are only capable of determining relative temperature. In order to determine absolute temperature a reference temperature must be known. If ambient temperature is known then this can be used as a reference. The second reason was the fact that engine operation is influenced by the environment temperature in which it operates. Providing the pilot with information about the ambient temperature will aid in making some engine related decisions. For example whether to apply carburetor heating.

1.4.4 Why RPM measurement ?

The input processor analog to digital converter was used as part of the measurement of the previous three variables. RPM was chosen as the fourth quantity to be measured primarily because the best way to achieve RPM measurement does not involve A/D conversion.

1.5 To guide the reader

The remainder of this report is divided into two sections. The purpose of the first of these sections, titled "Relevant Theory," is written to provide the reader with the background knowledge necessary to understanding the information given in the second section of the report. The Relevant Theory section of this report is divided into three subsections. The first subsection describes methods of temperature measurement. The second subsection describes the concept of analog to digital conversion. The final subsection describes the

HC11 microprocessor. Knowledge gained in these three areas will be of great benefit when reading the second section of the report.

The second section of this report, titled "Project Detail" provides detail about the entire project. The section begins with an overview of the entire project system. The overview describes the various subsystems and the interrelationships between these subsystems. Next each subsystem, beginning with the operating system, is described in full detail. The purpose, specifications and performance of each system is included as part of this detail. Also included in this detail is the final system design and the methods involved in creating the design. Any relevant theory to the design is also included. The second section of the report concludes with some comments about the project system and indicates areas where further work is required.

CHAPTER 2: Relevant Theory

This section of the report will provide the background knowledge necessary to understand the "Project Detail" section of the report. Four important areas are discussed and these are;

- Temperature Measurement
- Analog-to-Digital Conversion
- The HC11 Microprocessor
- The HC11 Evaluation Board

2.1 Temperature Measurement

Temperature measurement is an important part of aircraft instrumentation. Presently many different temperature readings are taken, even in Very Light Aircraft, using a variety of non electrical and electrical methods. Some of these readings include Exhaust Gas Temperature, Manifold Temperature and Oil Temperature. Temperature measurements allow the pilot to operate the aircraft correctly and also can give warning of system failure.

2.1.1 Non-Electrical temperature measurement

There are many non-electrical methods for measuring temperature. Most of these methods rely on the expansion of solids, liquids or gases with increase in temperature. Two non electrical methods for measuring temperature are the bimetallic thermometer and the Bourdon tube. Non-electrical temperature measuring devices are difficult to interface with electronic equipment, and therefore not used in this project. The interested reader can find a more detailed description of non-electrical methods in Lombardo (1988, p26 – 27)

2.1.2 Electrical Temperature Measurement Methods

Electrical temperature measurement is based on the variation of some electrical quantity with respect to temperature. An important advantage of electrical temperature sensors is that they can be interfaced with electronic equipment. There are three categories of electrical temperature sensors. These are resistance temperature detectors (RTD), semiconductor temperature sensors and thermocouples. The most appropriate choice between these three types of devices depends on the application requirements. Common considerations when choosing between semiconductor sensors, RTDs and thermocouples are:

- Accuracy
- Temperature range
- Cost
- Speed of response
- Size
- Ruggedness
- Ease of use

2.1.2.1 Resistance Temperature Detectors (RTD)

Resistance temperature detectors rely on the fact that the resistivity of metals is related to temperature. The relationship is due to electron scattering caused by vibrations in the atomic lattice. Other causes of electron scattering are largely unrelated to temperature and can be ignored for the purposes of temperature measurement. RTDs are active devices, they require an external voltage source to drive them.

RTDs are constructed using materials that exhibit a useful resistance versus temperature relationship. A useful resistance versus temperature relationship is ;

- Smooth
- Stable
- Predictable

To achieve predictability the resistivity due to impurities must be kept constant. For this to occur the chemical and physical composition of the RTD must also be uniform. The best way to obtain uniform chemical composition is to use only pure substances. Consequently many RTD's are formed from the elements platinum, gold, silver, nickel and copper since these metals can be obtained in very pure forms and present stable and smooth resitivity versus temperature.

2.1.2.2 Semiconductor Temperature sensors.

Semiconductor temperature sensors work in a variety of ways. Thermistors are one type, and work on a similar principle to RTD's where the variation in the resistance is related to temperature. The resistance temperature relationship of thermistors is highly non-linear.

A second method relies on the relationship between diode reverse breakdown voltage and temperature. An example temperature sensor that relies on this relationship is the LM335. The important characteristics of the LM335 are;

- Linear temperature voltage relationship
- Accuracy to within one degree
- Operating temperature of –40°C to 100°C
- Low cost per unit
- Small package size

The spec sheet for the LM335 is reproduced in appendix A.

2.1.2.3 Thermocouples

Thermocouples are passive temperature sensors that develop an EMF relative to temperature. A thermocouple junction is produced when ever two dissimilar metals are brought into contact. A common thermocouple is the K type nickel-chromium vs nickel-aluminium. Power series for calculating thermocouple voltages based on temperature are

detailed in Appendix B. Thermocouples measure relative temperature and require referencing to gauge absolute temperatures.

2.1.2.4 Temperature sensor comparison

The characteristics of the three types of temperature sensors are compared in table 2.1.

Characteristic	RTD	Semiconductor temperature sensor	Thermocouples
Temperature range	-200°C to 650°C	-55°C to 150°C	-270°C to +2000°C
Cost	Generally 2 to 3 times thermocouple cost	Similar cost to RTD	Cheapest
Accuracy	High Less than 1°C error with calibration	High Less than 1°C error with calibration	Medium Typically 1°C to 4°C error
Tip Sensitivity	Not Tip sensitive	Not Tip sensitive	Tip sensitive
Minimum Size	Largest	Middle	Smallest
Reference required for absolute temperature measurement?	No	No	Yes
Exhibits Self Heating?	Yes	Yes	No
Needs power supply?	Yes	Yes	No
Linearity	Generally most linear	Usually highly non-linear but there are exceptions Eg LM335	Roughly linear over certain temperature ranges otherwise highly non-linear

Table 2.1 Temperature sensor characteristics by type

2.2 Analog-to-Digital Conversion

Analog-to-digital (A/D) conversion is the transformation of an analog signal into a digital signal. A block diagram depicting an A/D converter is shown in figure 2.1. The processes involved are sampling, quantizing and encoding. Each process is described below. A more thorough treatment of analog to digital can be found in Ziemer, Tranter and Fannin (1993 p. 335-371)

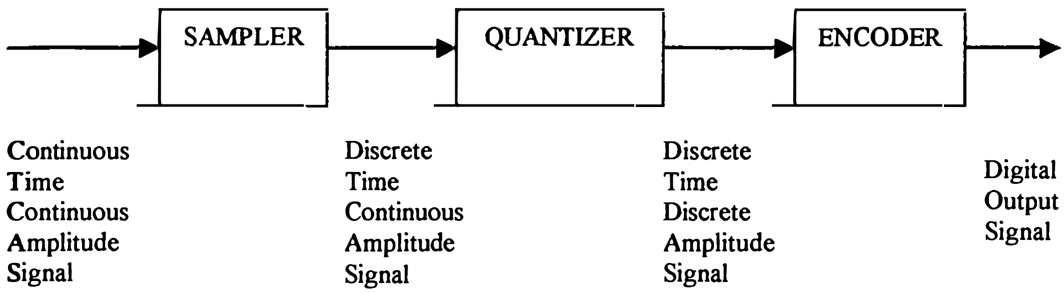


Figure 2.1 Analog to Digital converter block diagram

2.2.1 Sampling

Sampling involves measuring a continuous time signal at discrete time intervals. A diagram showing an analog signal and it samples is shown below in Figure 2.2.

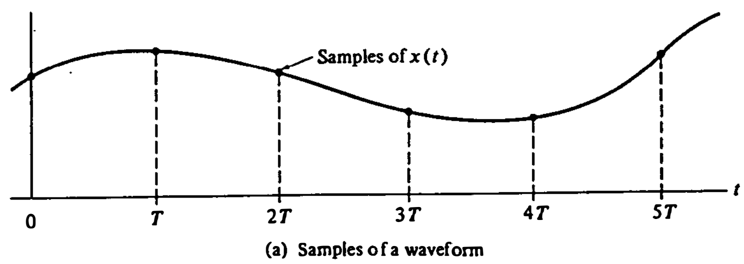


Figure 2.2 An analog signal and its samples
(Taken from Ziemer, Tranter and Fannin (1993 p. 354))

Figure 2.3 illustrates the logic of a sampling device. The sampling device multiplies the Continuous-time signal x(t) by the sampling function p(t). This results is the sampled signal $x_s(t)$. The value of $x_s(t)$ is shown below;

$$x_s(t) = p(t) * x(t)$$

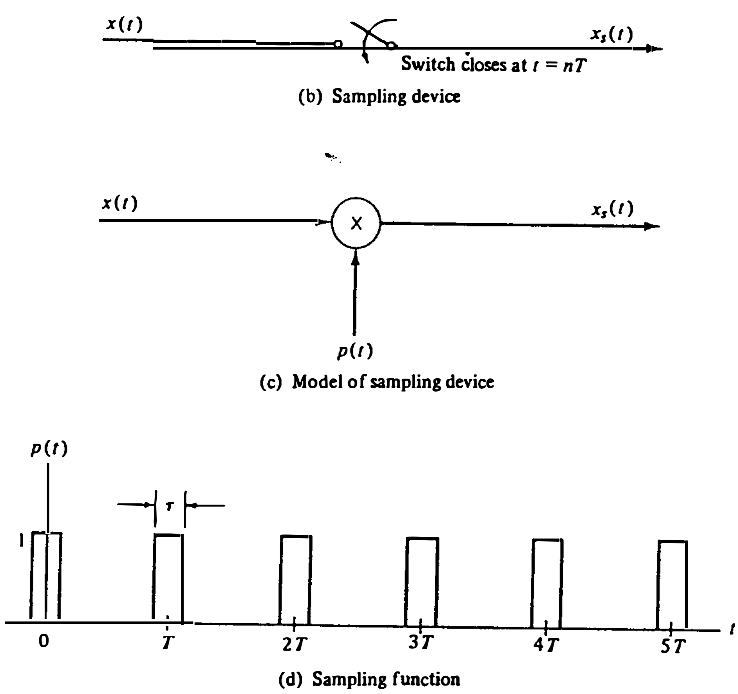


Figure 2.3 The logic of a sampling device
(Taken from Ziemer, Tranter and Fannin (1993 p. 354))

2.2.2 Quantizing and Encoding

Quantizing and encoding occurs after sampling and this process is shown in Figure 2.4. Quantizing involves rounding a sample value to the nearest of a finite set of allowable values. Every quantizing level is assigned a digital word of fixed word length. Encoding involves assigning each sample the digital word corresponding to the quantizing level to which the sample was rounded. For binary representation, the number of quantizing levels, q, and the digital wordlength, n, are related as follows;

$$q=2^n$$

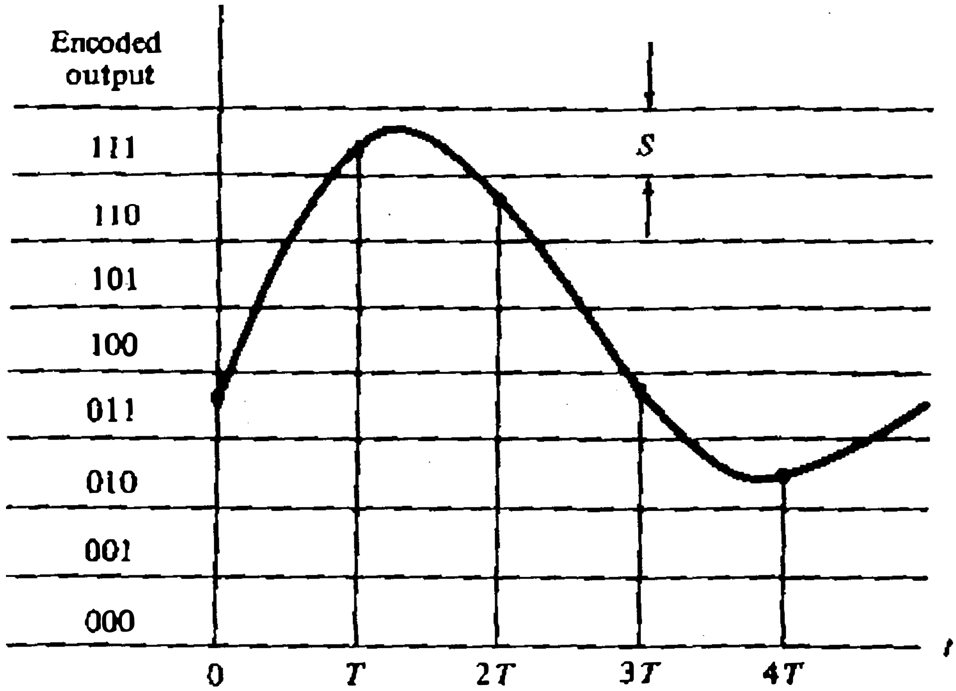


Figure 2.4 Quantizing and Encoding
(taken from Ziemer, Tranter and Fannin (1993 p. 368))

For the example shown in Figure 2.4, a three bit word is assigned to each quantizing level allowing eight quantizing levels. If S is the difference between quantizing levels, then the maximum error resulting from quantizing a sample is ± ½ S.

2.3 The HC11 Microprocessor

A microcontroller is required in this project to perform A/D conversion and to perform calculations. A microcontroller is a simple computer on a single chip designed for use in embedded applications. Different microcontrollers come with many different features. As a general rule microcontrollers contain a CPU, RAM, ROM and some extra features particular to the specific microcontroller. Microcontrollers are the "brain" inside many embedded systems.

The HC11 microprocessor is a microcontroller manufactured by Motorola. The HC11 is a relatively complex microcontroller with many extra features beyond the basic CPU, RAM and ROM. These additional features make the HC11 useful in many different embedded applications. Many of these additional features have been used throughout this project.

The MC68HC11E9 variant of the HC11 is used in this project. It is upon this processor variant that the EVBU, described later, is constructed around. The processor is based on high-density complementary metal-oxide semiconductor (HCMOS) technology. The major features of the M68HC11E9 are listed in HC11 MC68HC11E9 Technical Data (1991, p.1-1). The list is repeated IN Appendix C for reader convenience.

When planning to use the HC11 to perform an embedded application it is necessary to have some knowledge of the structure of the microprocessor. The structure of the HC11 is illustrated in Figure 2.5.

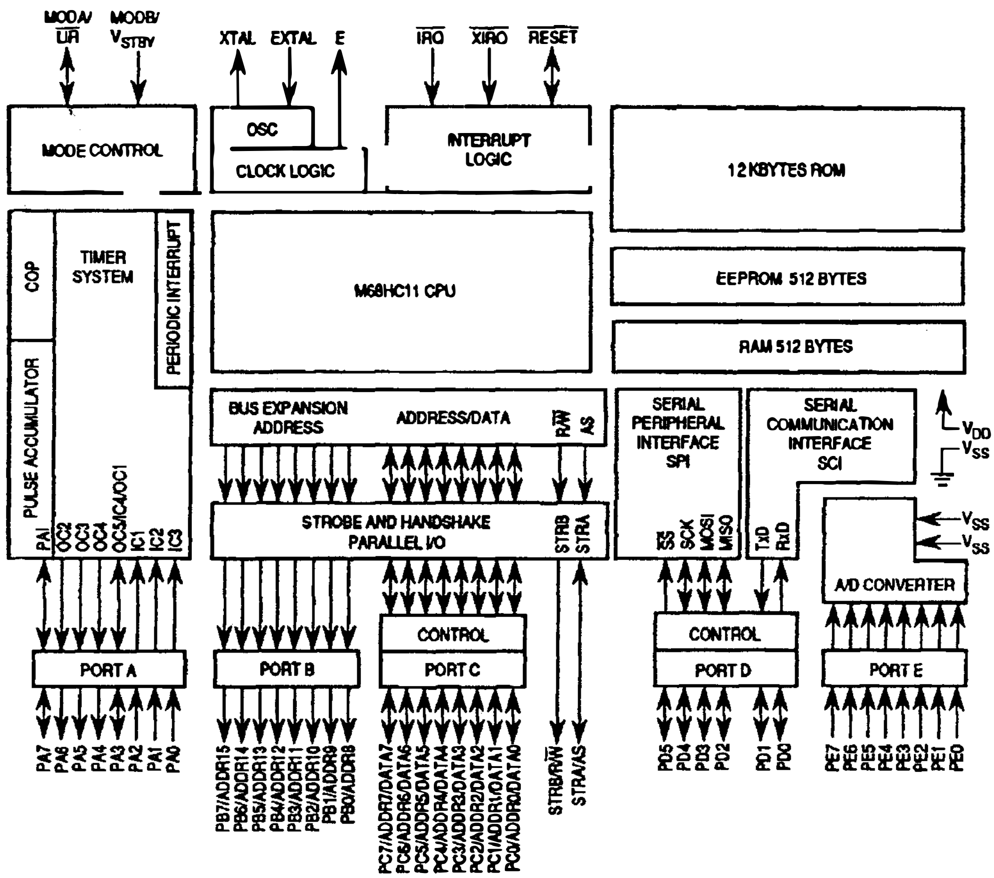


Figure 2.5 MC68HC11E9 Structure
Taken from (HC11 MC68HC11E9 Technical Data (1991, p. 1-2))

Not all of the HC11 features have been used during this project. Features that have been used or have been considered as part of the project system will be described in the following subsections. These features are;

- The Analog to Digital converter
- The timer system including;
  - ♦ Pulse Accumulator
  - ♦ Input Capture system
  - ♦ Computer Operating Properly watchdog system (COP)
- Clock Monitor System

2.3.1 Analog-to-Digital Converter

The HC11 contains an analog-to-digital (A/D) converter which may be connected to read from any of the pins of port E, as shown in figure 2.5. The concept of A/D conversion and the terminology used in this process have been explained previously. The A/D converter reads an analog input voltage level and generates a binary number corresponding to the approximate analog voltage. This particular converter uses a method called successive-approximation charge-redistribution. The HC11 A/D converter is an 8 bit converter.

2.3.1.1 Charge-Redistribution a 4 bit example

Charge-Redistribution is one of several techniques for A/D voltage conversion. Information about other A/D conversion techniques can be found in Sedra and Smith (1998, p.868). A Charge-Redistribution circuit consists of a voltage comparator, logic controlled switches, control logic and a binary-weighted capacitor array. This circuit performs the sampling, quantization and encoding of the analog signal.

Sampling of a voltage input is achieved by a parallel bank of capacitors and two logic controlled switches. A diagram of the analog to digital converter system in sample mode is shown below in Figure 2.6. The number next to each capacitor represents the relative capacitance of each capacitor. The leftmost capacitor in Figure 2.6, for example, has 8 times the capacitance of the rightmost capacitor.

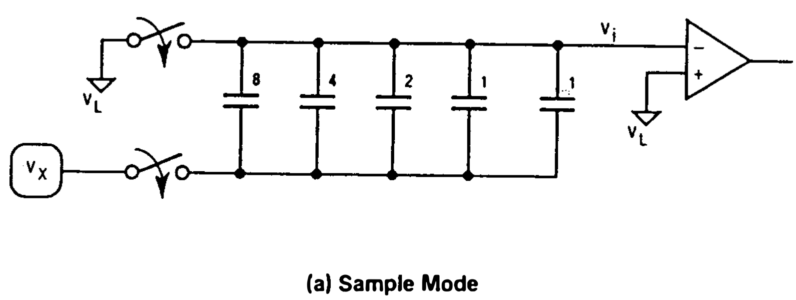


Figure 2.6 Sample mode
(Taken from HC11 M68HC11E9 Reference Manual, 1991, p.12-2)

After a sample has been taken the system enters the hold mode mode. The hold mode is shown in Figure 2.7. During the hold mode charge is trapped on the capacitors and the system is isolated from the voltage it is sampling. It is important to note that V_L is disconnected from the top plates and then connected to the bottom plates of the capacitors. Since charge is conserved the voltage that appears at the negative input of the comparitor (Vi) is of the same magnitude but opposite sign to the voltage being sampled(Vx). Mathematically this is expressed as follows;

$$V_i = -V_x$$

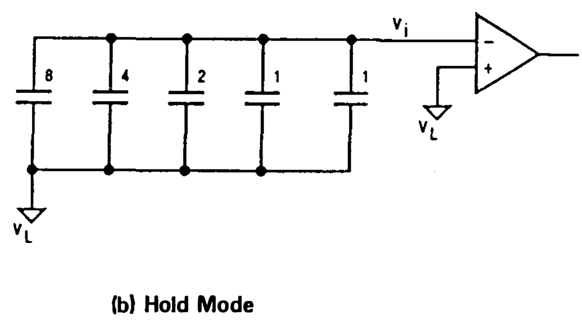


Figure 2.7 Hold Mode
(Taken from HC11 M68HC11E9 Reference Manual, 1991, p.12-2)

The final stage of the process is where encoding and quantization occur. This stage is called the Charge-Redistribution phase. The Charge-Redistribution phase is illustrated in Figure 2.8. In this part of the A/D process each capacitor, beginning with the largest one, is switched from V_L (0.0) V to V_H. If the output of the compariator, after each capacitor is switched, is equal to a logic one then the bottom plate of the last capacitor switched remains at V_H. If the output of the comparitor is equal to a logic zero then the bottom plate of the last capacitor switched is connected to V_L before switching the next capacitor. Each time a capacitor is switched the output from the comparitor is stored in one of the bits of the successive approximation register, also shown in figure 2.8. The first bit of the successive approximation register to be modified will be the MSB and the last bit modified will be the LSB.

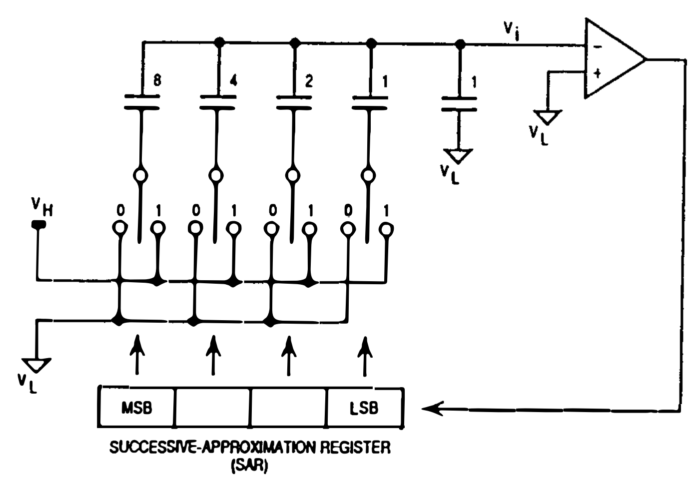


Figure 2.8 Charge redistribution phase
(Taken from HC11 M68HC11E9 Reference Manual, 1991, p.12-2)

2.3.2 Timer System

The HC11 timer system is driven by the main oscillator and clock generator. As well as providing a clock signal, the timer system also includes five other subsystems. These subsystems are shown in Figure 2.9.

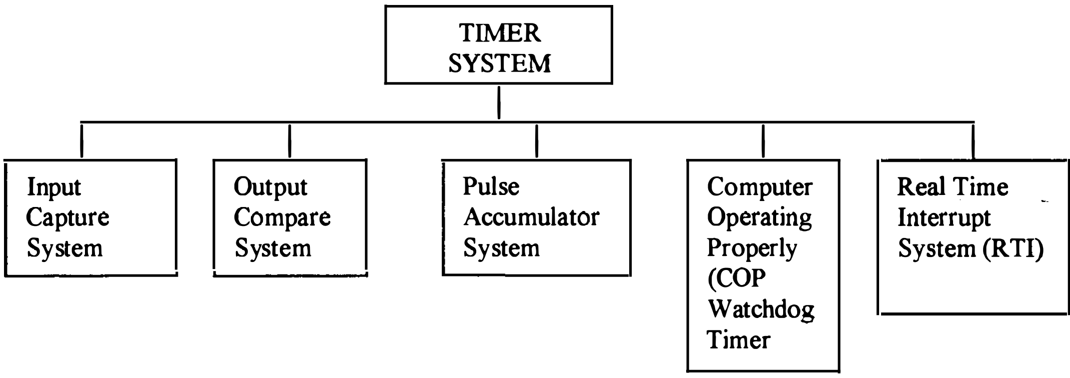


Figure 2.9 Timer Systems

All of these timer subsystems are described in HC11 MC68HC11E9 Technical Data (1991, p. 9-1 to 9-22). The timer subsystems are also described in detail in HC11 MC68HC11E9 Reference Manual (1991, p. 10-1 to 11-12). The timer subsystems that are of interest for this project are the Pulse Accumulator System, Input Capture System and the Computer Operating Properly Watchdog System.

2.3.2.1 Pulse accumulator system

The pulse accumulator is an 8-bit counter that may operate in two modes

- Event Counting Mode
- Gated Time Accumulation Mode

The pulse accumulator is connected to pin 7 of port A as shown in Figure 2.5. Software is used to enable, select the mode of, and determine the polarity of the signals recognized by the pulse accumulator system. There are two independent interrupts that can be generated by the pulse accumulator system. The first interrupt is generated whenever the 8 bit counter overflows. The second interrupt is generated each time a specified pulse edge is detected at the pulse accumulator system. Detailed information about the pulse accumulator system is given in HC11 MC68HC11E9 Reference Manual (1991, p. 11-1 to 11-12)

2.3.2.1.1 Event Counting Mode

In the event counting mode, the 8 bit counter is incremented every time a specific edge is detected at the pulse accumulator pin. A certain event, for example a button being pushed, could be made to cause the specified edge to occur at the pulse accumulator pin. The pulse accumulator system in the event counting mode can then be used to determine how many events have occurred. It is also possible for the pulse accumulator system to generate an interrupt after a set number of events.

2.3.2.1.2 Gated Time Accumulation Mode

In the gated time accumulation mode, the 8 bit counter is incremented every 64^{th} E-clock cycle, provided the signal at the pulse accumulator pin is at a specified binary level. The binary level that permits counting is chosen in software. The gated time accumulation mode is primarily used for measuring the period of pulses.

2.3.2.2 Input Capture System

The input capture system is used to determine the time a specific event occurs. The HC11 contains a 16 bit free-running counter that represents physical time. The input capture system uses this counter to record the time when an external event occurs. This is achieved by latching the counter contents, when a specified signal edge becomes present at an input capture pin. The timer value is stored in the input capture register corresponding to the appropriate input capture pin.

There is a large number of uses that can be found for the HC11 Input Capture system. Two common uses of input capture are determining the period between two successive logic pulses and determining the period of a single pulse.

The important characteristics of the Input Capture System are;

- There are three port A input pins that may be used for input capture. These are shown as PA0, PA1 and PA2 in Figure 2.5.
- There is one port A bi-directional pin, PA3 in Figure 2.5, that may be used for input capture if it is not being used for output compare. This results in a maximum of four pins that may be used for input capture.
- Each input capture pin has its own input capture register. This allows up to four input capture functions to be performed concurrently.
- Each input capture register is 16 bit.
- The free running counter may operate at four different speeds.
- Each input capture pin may be configured, in software, to capture rising edges, falling edges or any edge.
- An interrupt may be generated each time the 16 bit free running counter overflows.
- An interrupt, specific to the input capture pin in question, may be generated each time an input capture occurs.

Full details about the HC11 input capture system can be found in the HC11 MC68HC11E9 Reference Manual (1991, p. 10-2 to 10-27)

2.3.2.3 Computer Operating Properly Watchdog System

The computer operating properly (COP) watchdog system's purpose is to prevent software failures. This watchdog consists of a timer driven counter and interrupt generating circuitry. When the COP system is active, software is required to periodically reset the COP counter to avoid it overflowing. If the counter is allowed to overflow then it may be assumed that the software is not executing properly. If the COP counter overflows a system RESET is issued which will restart the software and therefore prevent system failure. The HC11 MC68HC11E9 Reference Manual (1991, p. 5-7) describes the COP watchdog timer in more detail.

2.3.2 Clock monitor system

The clock monitor is often used in conjunction with the Computer operating properly watchdog timer COP to ensure that software is being executed correctly. The COP is able to detect when software is not executing in the correct sequence. The COP system is not able to detect when software is not executing because the system clock has stopped. The clock monitor system can be used to initiate a system RESET at any time it detects that the system clock has stopped.

2.3.2.1 Considerations when using the Clock monitor

Two things must be considered when using the clock monitor system. The first consideration is the speed the system clock. The second consideration is whether any stop instructions will be used in the software.

2.3.2.1.1 System clock speed and the clock monitor

To understand why the speed of the HC11 clock affects the operation of the clock monitor, it is necessary to understand how the clock monitor works. The clock monitor circuit is based on an internal resistor-capacitor (RC) delay. If no clock edges are detected within this RC time delay, the clock monitor generates a system RESET

The RC time-out delay is not constant between individual HC11 microprocessors. The difference is the delay is a result of variations in the HC11 manufacturing process. To avoid problems the clock monitor should not be used with an E-clock frequency of less than 200kHz.

2.3.2.1.2 STOP instructions and the clock monitor

A STOP instruction stops all of the HC11 internal clocks until an interrupt or a RESET occurs. Naturally if a STOP instruction is executed while the clock monitor system is operating then the clock monitor will immediately initiate a RESET. This action will defeat the purpose of the STOP instruction. As a result, except for the purpose of clock monitor testing, the clock monitor must be disabled before a STOP instruction is executed.

2.4 HC11 Evaluation Board (EVBU)

All programs were written for and tested on a M68HC11EVBU (EVBU). The M68HC11EVBU is a universal HC11 evaluation board manufactured by Motorola. Programs developed for this board may need to be modified before being used on other HC11 processors. Appendix H contains a diagram depicting the EVBU. Four areas related to the EVBU need to be discussed. These are;

- RS232 Interface
- Monitor Program (BUFFALO)
- Operating Modes
- Available Memory

2.4.1 RS232 Interface

The EVBU has a built in RS232 interface. The RS232 interface makes it possible to connect the evaluation board to a personal computer (PC). The PC user may then communicate with the EVBU using a terminal program. The RS232 interface makes it possible to develop programs on the PC before downloading them to the EVBU.

2.4.2 Monitor Program (BUFFALO)

The EVBU has a monitor program stored in ROM called BUFFALO (Bit User Fast Friendly Aid to Logical Operations). BUFFALO allows two methods to assemble and run user Code. These methods are;

- Using the line assembler provided as part of BUFFALO
- Assembling code on a host computer and then downloading it to the EVBU. Code to be downloaded must be in the Motorola S-Record format.

Except in the very early stages of the project the second method was used to assemble user code.

2.4.3 Operating Modes

The EVBU allows for two modes of operation selected by adjusting the Program Execution Select Header, jumper 2 on the EVBU. The modes are;

- Normal Mode
- EEPROM Jump Mode

2.4.3.1 Normal Mode

In the normal mode of operation the BUFFALO monitor program is executed straight after a RESET. In this mode of operation the user program is then run by issuing a GO instruction to BUFFALO. The normal mode of operation does limit the ability of some user programs to perform the tasks for which they are written.

The HC11 has some protected registers that may only be altered in the first 64 E-clock cycles after a RESET. In the normal mode the BUFFALO monitor program is always executed before the user program and this prevents the user program executing in the

period when the HC11 protected registers may be modified. This prevents the user program modifying the protected registers and can cause some programs to fail.

2.4.3.2 EEPROM Jump Mode

The EEPROM jump mode of operation overcomes the limitations imposed by the normal mode of operation. In this mode the HC11 begins executing code at the EEPROM address $B600, meaning the user code can run straight out of a RESET. Consequently the user program can modify the protected registers and is not restricted by several other limitations imposed by the BUFFALO monitor. The EEPROM jump mode of operation was a great advantage during this project because;

- it made possible the testing of the Clock Monitor system
- it allowed the correct set-up of the timer Prescaler for the RPM system.

2.4.3.3 Program Execution Header

The program execution select header is used to determine the operating mode of the EVBU. The header, J2 on the EVBU, is connected to pin 0 of port E of the HC11. Depending on the position of the jumper the logic level at pin 0 is either set to a 0 or 1. The mode of operation is then determined by reading the logic level at pin 0. This becomes a problem if pin 0 is used for A/D converter operations. To avoid these problems port 0 has not been used for A/D converter operations. This leaves 7 A/D converter inputs available for use in the project.

2.4.4 Available Memory

The EVBU used in this project was operated in the single-chip mode. The result of this is that the memory of the entire system is limited to;

- 12 K of ROM
- 512 bytes of RAM
- 512 bytes of EEPROM

The 12 K of ROM contains BUFFALO. As a consequence none of the ROM is available for developing user programs. The 512 bytes of RAM is located in addresses $0000 to $01FF. Some of the this RAM is used by the BUFFALO monitor program leaving 325 bytes available for user programs. The RAM is shown in the memory map depicted in figure 2.10. The 512 bytes of EEPROM is left completely available to the user. In the EVBU system the EEPROM exists in memory addresses $B600 to $B7FF. When both EEPROM and RAM are used the EVBU provides a maximum of 837 bytes for user programs. This is a small amount and has imposed limitations on program development throughout the project.

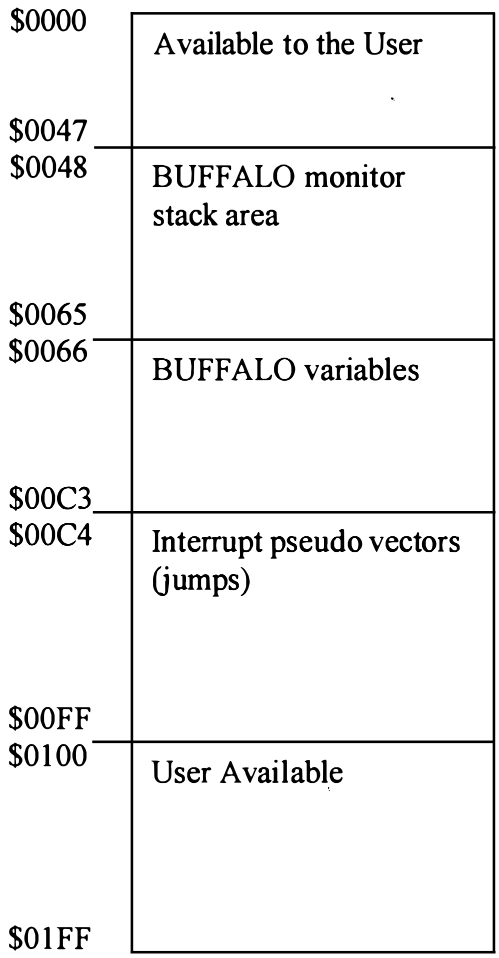


Figure 2.10 memory map of the EVBU RAM.

Project Detail

Project Detail Prologue

This section of the report fully describes the project. This section begins with an overview of the entire project system. During the overview, the project system is described from three different perspectives. These perspectives are;

1. Major Subsystem Perspective
2. Hardware Perspective
3. Software Perspective

The purpose of the overview section is to describe the overall system structure. The next, step after the overview, is to provide detail. Detail is provided by fully describing each subsystem, beginning with the operating system. After the final subsystem is described, the report concludes with comments about the project system and indicate areas for further research and development.

Project Overview

The project system was designed to achieve several objectives. The primary objective of the project system is to determine the value of four physical variables and store these values in the HC11 memory. This is accomplished by implementing the project system dataflow, depicted in Figure 1.3 of the Introduction section of this report. Other project system objectives are;

- To generate a warning if the value of a physical variable is outside of its normal range
- To avoid system failure by accidental halting of the system clock
- To avoid system failure due to a software failure
- The system to be modular.

The project system can be viewed from three different perspectives. These perspectives are;

1. Major Subsystem Perspective
2. Hardware Perspective
3. Software Perspective

The following sections describe the project from each of the perspectives.

Major Subsystem Perspective

The project system is comprised of seven major subsystems that interact in order to accomplish the system objectives. The subsystems and their interrelationships are shown in Figure 3.1. The subsystems are described briefly in Table 3.1

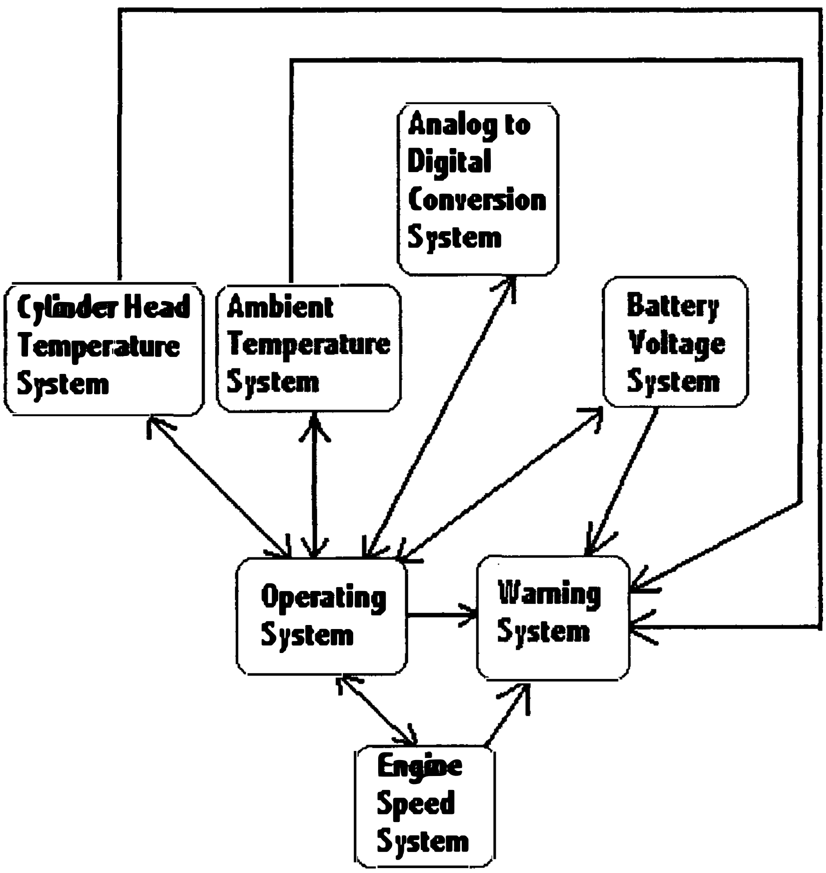


Figure 3.1 Project System from a subsystem perspective

Subsystem Name	Description
Operating System	• Centre of the project system • Co-ordinates and supports the other subsystems to achieve system objectives
Analog-to-Digital conversion system	• Converts analog signals into digital ones so they may be processed in software • Essential to the operation of; ➢ The Battery Voltage System ➢ The Cylinder Head Temperature System ➢ The Ambient Temperature System
Battery Voltage System	• Determines the voltage of a 12 volt battery. • Sets the warning corresponding to the battery voltage
Cylinder Head Temperature System	• Determines engine cylinder head temperature • Sets the warning corresponding to the cylinder head temperature
Ambient Temperature System	• Determines ambient temperature • Provides reference temperature for the cylinder head temperature system
Engine Speed System	• Determines engine RPM • Sets the warning corresponding to the engine speed
Warning System	• Provides a warning corresponding to each input in the project system • Provides a master warning if any input is determined to be operating outside its proper range. • Is partly implemented within the; ➢ Battery Voltage System ➢ Cylinder Head Temperature System ➢ Ambient Temperature System ➢ Engine Speed System

Table 3.1 Short Subsystem Descriptions

Hardware Perspective

The project system, from a hardware perspective, is depicted in Figure 3.2. The centre of the project system is the HC11. The HC11 is essential to the implementation of all of the project subsystems. The HC11's wide range of capabilities make the project system possible. The other hardware components support the HC11 and a brief description of each component is given in Table 3.2

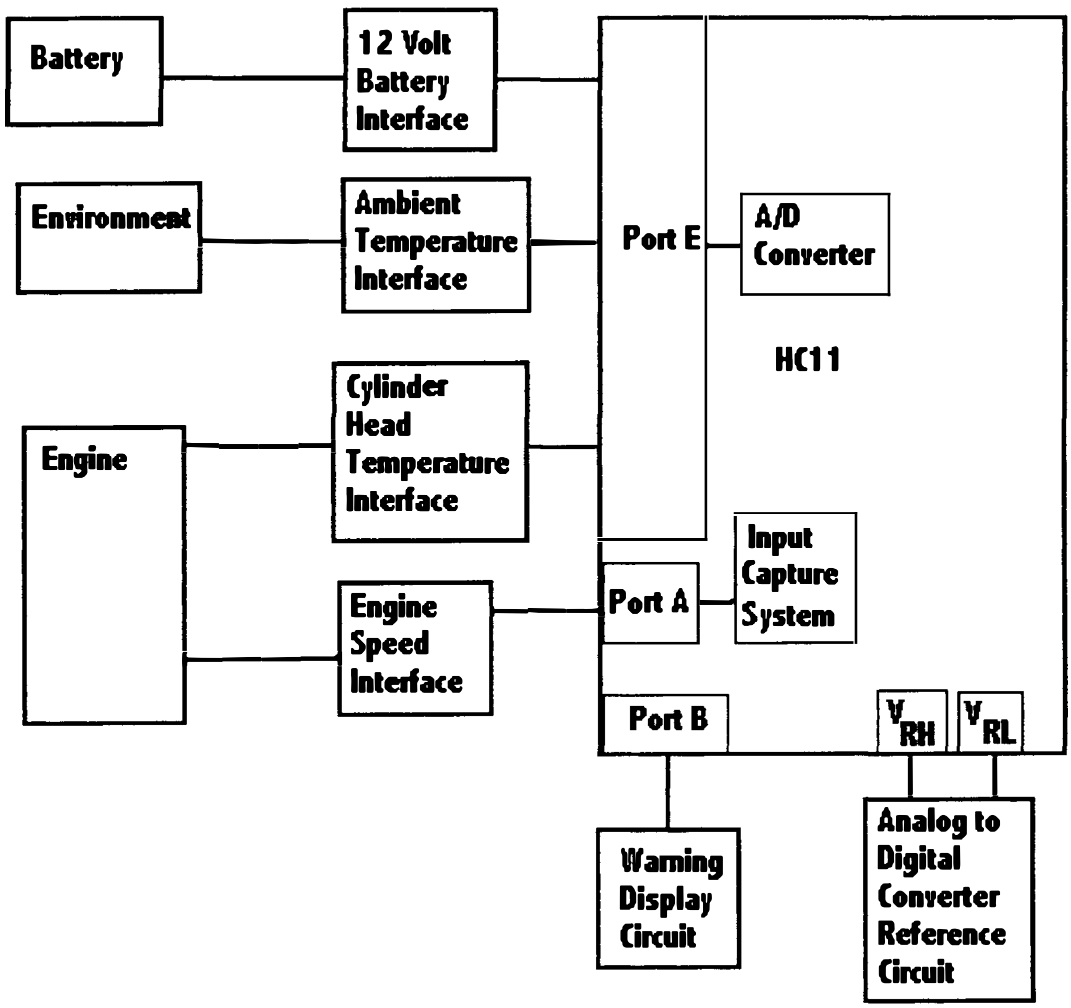


Figure 3.2 Project System from a Hardware Perspective

Hardware Name	Description
12 Volt Battery Interface	• Part of the Battery Voltage System • Converts battery signal to one that can be read by the HC11 A/D converter
Ambient Temperature Interface	• Part of the Ambient Temperature System • Converts ambient temperature into a signal that can be read by the HC11 A/D converter
Cylinder Head Temperature Interface	• Part of the Cylinder Head Temperature System • Converts cylinder head temperature into a signal that can be read by the HC11 A/D converter
Engine Speed Interface	• Part of the Engine Speed System • Converts the engine speed into a signal that can be read by the pulse accumulator
Warning Display Circuit	• Part of the Warning System • Displays the warning corresponding to one input • Displays the master warning
Analog to Digital Converter Reference Circuit	• Part of the Analog to Digital Converter System • Sets the HC11 A/D converter resolution • Provides a voltage reference for the HC11 A/D converter

Table 3.2 Hardware Descriptions

Software Perspective

The project system can be viewed from a software perspective. The software, which is written in assembly language, consists of eight modules. The hierachy of the modules is shown in Figure 3.3. Each module is described briefly in Table 3.3. All project subsystems are partially implemented in software.

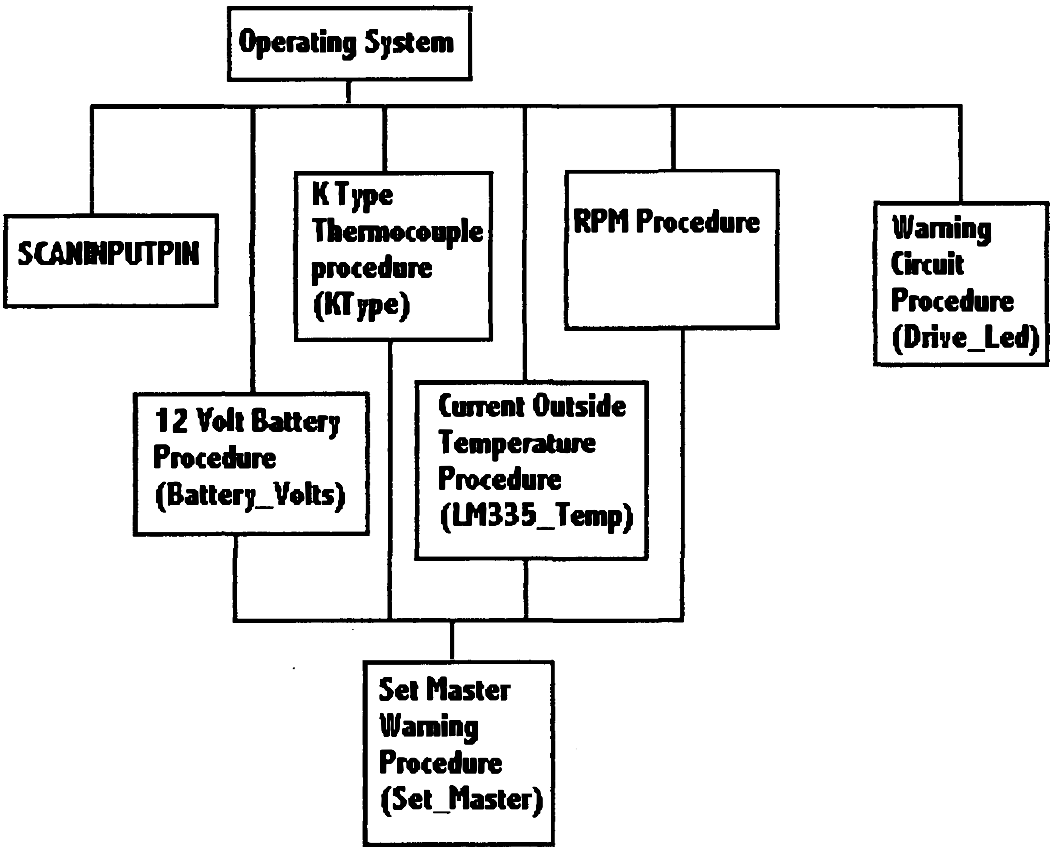


Figure 3.3 Project System from a Software Perspective

Module Name	Description
Operating System	• Provides control for the entire system • Manages memory
SCANINPUTPIN	• Part of the Analog-to-Digital Conversion System • Operates the HC11 A/D converter
12 Volt Battery Procedure (Battery_Volts)	• Part of the Battery Voltage System and Warning System • Determines Battery Voltage based on the voltage present at an A/D converter pin • Sets warning corresponding to Battery Voltage
K Type Thermocouple Procedure (Ktype)	• Part of the Cylinder Head Temperature System • Determines cylinder head temperature based on the voltage present at an A/D converter pin and a reference value • Sets warning corresponding to cylinder head temperature
Current Outside Temperature Procedure (LM335_Temp)	• Part of the Ambient Temperature System • Determines ambient temperature based on the voltage present at an A/D converter pin • Sets warning corresponding to Ambient Temperature
RPM Procedure	• Part of the Engine Speed System • Determines engine speed based on the time between logic pulses present at an input capture pin • Sets warning corresponding to engine speed
Warning Circuit Procedure (Drive_Led)	• Part of the Warning System • Controls the Warning Display Circuit
Set Master Warning Procedure (Set Master)	• Sets the Master Warning

Table 3.3 Software Module Descriptions

CHAPTER 3: Operating System

From a subsystem perspective, the operating system is the centre of the project system. The operating system performs all of the supporting and co-ordinating functions necessary to allow the project system to accomplish its objectives. The functions performed by the operating system are;

- Memory Management
- Detecting System Crashes
- Setting up the Analog-to-Digital converter System
- Initializing the stack
- Updating the procedure local data offset
- Calling procedures
- Resetting Warnings after each complete loop
- Setting up the Prescaler for use in input capture applications.

3.1 Memory Management

It is very important any computer system that memory be managed properly. Failure to do so can lead to systems doing unexpected things or crashing. This system is designed to be part of an integrated aircraft system. A failure in such a system could lead to disaster. Very careful consideration has been given to memory management.

3.1.1 Memory and the EVBU

Using the EVBU has affected how memory has been allocated to different parts of the system. The three objects that need to be stored in the EVBU memory are;

- Program Variables
- Stack
- Program Instructions

3.1.1.1 Program Variables and the Stack

A map of the EVBU RAM is shown in figure 3.4. The EVBU expects the application software variables and stack to be placed in Free Area 1. It is possible to use Free Area 2 for storing variables and the stack but this can cause problems. The project stack and variables only reside in Free Area 1.

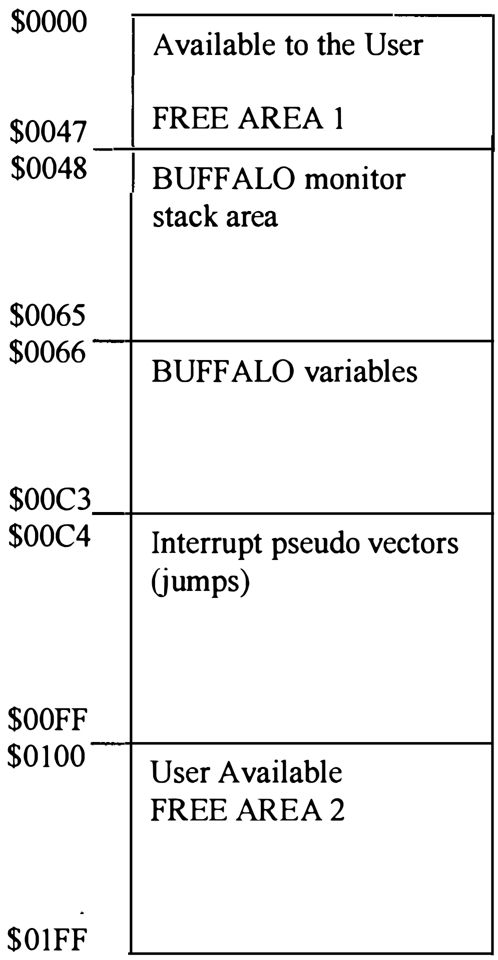


Figure 3.4 Memory map of the EVBU RAM.

3.1.1.1.1 Avoiding stack/program variables memory clash

Both the stack and the program variables share Area 1 of the RAM. To avoid memory clashes it is essential that the stack and the program variables never overlap. Overlap is avoided by dividing RAM Area 1 into two sections. A memory map demonstrating these two sections is shown in Figure 3.5. The first section contains program variables and includes memory addresses $0000 to $002F. The second section is allocated for the stack includes memory addresses $0030 to $0047.

$0000 – $002F	Area allocated for program variables
$0030 – $0047	Area allocated for the stack

Figure 3.5 Areas for stack and Local variables

3.1.1.1.2 Stack Space

The stack size is limited to the 24 bytes in addresses $0030 to $0047. The stack may be used for parameter passing, storing return addresses and for holding temporary procedure data. To avoid the stack exceeding its allocated space the amount of available stack space must be considered each time a procedure is called or when the stack is used to store temporary data. This consideration is particularly important when calling procedures, like SCANINPUTPIN, that make heavy use of the stack. The SCANINPUTPIN procedure is covered in the Analog-to-Digital Conversion System section.

The SCANINPUTPIN procedure uses the stack to store the results of each "scan average." The maximum number of "scan averages" that can be stored is dependant on the space available to the stack. The SCANINPUTPIN is called by the mainline when the stack is empty. Two bytes of the stack are used to store the return address when the SCANINPUTPIN procedure is called. This leaves 22 bytes available for use in the SCANINPUTPIN procedure.

Each "scan average" requires 1 byte of the stack space. Since there are 22 bytes of stack space available to the SCANINPUT procedure the maximum number of "scan averages" that may be stored is 22. This means that the maximum value for the input parameter "NumberOfScans" that will not cause a the stack to exceed its allocated space is 22.

3.1.1.1.3 Global and local program variables

The program variables can be divided into two categories;

- Global Variables
- Local Variables

The memory allocated to program variables is sub-divided into an area for each of these categories.

The global variables are accessible to all procedures within the program including the mainline. Some of the major global program variables are the "warning page", and the "current value page." The global variables are allocated the space between memory locations $0000 and $001F inclusive. This limits the global variable space to 32 bytes

The local variables are only accessible to their owner procedure and only exist while that procedure is running. Local procedure variables are allocated addresses $0020 through to $002F for a total of 16 bytes.

3.1.1.2 System memory layout

A detailed layout of the memory of the HC11 is shown in Figure 3.6

$0000..$0007	Current Value page (CurrentVal)
$0008	(Master_Warning)
$0009..$0012	Warning page (Warning)
$0013	Out_Temp
$0014..$0015	Battery Voltage (Battery_V)
$0016..$0017	K type Thermocouple temp (Ktemp)
$0018..$001F	Space left available for other global variables
$0020..$002F	Space for use by procedure local variables
$0030..$0047	Space for use by the program stack
$0048..$0065	BUFFALO monitor stack area
$0066..$00C3	BUFFALO variables
$00C4..$00FF	Interrupt pseudo Vectors (jump table)
$0100..$01FF	Program Code
$0200..$0FFF	Not used. There is no memory present at these addresses
$1000..$103F	HC11 Control registers
$1040..$B5FF	No memory available Some addresses used by the EVBU for I/O
$B600..$B7FF	EEPROM This is used for program code
$B800 onwards	No memory available Some addresses used for interrupt vectors and I/O

Figure 3.6 EVBU Memory Map for the Project System

3.2 Detecting System Crashes

It is essential for an aircraft sensor monitoring system to be exceptionally reliable. One method used to make the software as reliable is possible is to detect when the system software has stopped and reset the system. Detecting when the system software is not operating properly is achieved by using the HC11 Clock Monitor and Computer Operating Properly watchdog.

3.2.1 Clock monitor

The system software is developed to operate continuously. This can only occur if the system clock is not allowed to stop. The consequences of the system clock stopping for anything but a very short interval of time could be quite catastrophic. As a result the HC11 clock monitor will be used to reset the system if the system clock is ever stopped.

3.2.1.1 Setting up the clock monitor.

To use the clock monitor system the clock monitor must be enabled. Enabling the clock monitor is achieved by setting the clock monitor enable (CME) bit in the OPTION register to 1. The OPTION register is outlined below in Figure 3.7. Bits marked with an "X" do not affect the operation of the clock monitor and therefore may be 0 or 1. The operating system updates the OPTION register within the first 64 E-clock cycles of a system RESET. This is done because the OPTION register is a protected register. The Clock Monitor system can only be used in the EVBU's EEPROM Jump Mode due to the need for changing the value of a protected register.

OPTION register address = $1039

Bit number	7	6	5	4	3	2	1	0
Bit name	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
Bits for clock monitor	X	X	X	X	1	0	X	X
Value out of Reset	0	0	0	1	0	0	0	0

Figure 3.7 Option register setup for the clock monitor.

3.2.1.2 Testing the Clock monitor system

To test the clock monitor system it is necessary to stop the system clock. Stopping the system clock is achieved by using the STOP instruction. When a stop instruction is executed the system clock ceases to count. If the clock monitor system is operating correctly a system RESET will be generated shortly after the system clock stops.

Although the concepts behind testing the system clock are simple to understand, the process is not so simple. There are additional processes that must occur in order to effectively test the system clock. Some of the additional processes that must be performed are specific to the Motorola EVBU used in throughout this project. The additional processes that must be performed are;

- Enabling the STOP instruction
- Placing a NOP before a STOP instruction
- Setting up the clock monitor interrupt vector

3.1.1.2.1 Enabling the STOP instruction

The STOP instruction may be enabled or disabled. When enabled a STOP instruction halts all of the HC11 system clocks until an interrupt or a RESET is detected. While in the stop state all register and memory values remain unchanged. When disabled, a STOP instruction behaves like a NOP instruction. A NOP instruction causes no change other than incrementing the program counter. To test the clock monitor system, using a STOP instruction, it is necessary for STOP instructions to be enabled.

The STOP instruction is enabled and disabled by adjusting the S bit in the Condition Code Register (CCR). The S bit in the CCR is called the "Stop Disable Bit." The CCR register is shown below in figure 3.8. When the S bit in the CCR is set to 0 the STOP instruction is enabled. The values of the various CCR straight after a RESET are shown in figure 3.8. Bits marked with a ? have an indeterminant value. Bits marked with an "X" have no effect on STOP instructions. It can be seen from figure 3.8 that the STOP instruction is always disabled by a RESET. The value of the S bit in the CCR register can be altered using the TAP instruction. To use the STOP instruction to test the clock monitor it is necessary to set the S bit to 0, using the TAP instruction, before issuing the STOP instruction.

Bit number	7	6	5	4	3	2	1	0
Bit name	S	X	H	I	N	Z	V	C
Bits for enabling a STOP	0	X	X	X	X	X	X	X
Value out of Reset	1	1	?	1	?	?	?	?

Figure 3.8 CCR register with the STOP instruction enabled

3.1.1.2.2 Placing a NOP before a STOP instruction

It is necessary to place a NOP instruction immediately before any STOP instruction. The reason is explained fully by the HC11 M68HC11 Reference Manual (1991, p. A-93). Basically an error in some M68HC11 mask sets can cause incorrect recovery from a

STOP instruction. Fortunately correct recovery will always occur if a NOP instruction is placed before a STOP instruction.

3.1.1.2.3 Setting up the Clock monitor interrupt vector

When the clock monitor system causes a RESET, the program counter is loaded with the contents of the address $FFFC, $FFFD. Code to deal with the Clock Monitor RESET may normally then be placed at the address pointed to by $FFFC, $FFFD. The address $FFFC, $FFFD is called the "clock monitor interrupt vector."

In the EVBU the address pointed to by $FFFC, $FFFD is always $00FD. The address $00FD is part of the EVBU interrupt jump table. The jump table consists of a series of three-byte fields. To use a vector specified in the interrupt vector jump table the user must insert a JUMP instruction followed by an extended opcode in the three-byte field. The three byte vector field for the clock monitor covers addresses $00FD, $00FE and $00FF.

During testing of the clock monitor, the three byte vector field will point to the start of the monitor program (BUFFALO). This is achieved by placing an extended jump opcode at address $00FD and the hexadecimal value $E000 in the addresses $00FE:$00FF.

3.1.1.2.4 Successful Clock monitor testing

The events that occur in a successful clock monitor test are;

1. A program that will cause the clock monitor to stop is loaded into memory
2. The BUFFALO monitor and debugger is used to set the three byte vector field for the clock monitor
3. A "GO" instruction is issued
4. The test program runs for a short period of time and then stops the CPU clocks.
5. The BUFFALO monitor prompt appears on the terminal screen.

If the clock monitor system fails in some way then the BUFFALO monitor prompt will not appear.

When the clock monitor was tested all the events required for successful testing occurred therefore the clock monitor testing was successful.

3.2.2 Computer Operating Properly (COP) Watchdog

The project system must not fail due to an unforseen problem in the system software. Failure can be avoided by issuing a system RESET if the software ceases to operate properly. The Computer Operating Properly (COP) feature of the HC11 is used for this purpose.

3.2.2.1 Setting up the COP

The COP must be setup properly in order to be used. COP system setup involves;

- Choosing the length of a COP timer period
- Enabling the COP

3.2.2.1.1 Choosing the COP timer period

If the COP system is to be used a time must be chosen within which the main program must reset the clock timer to avoid a system RESET. There are four time periods to choose from and they are selected by modifying the CR1 and CR2 bits of the OPTION register. The time periods and their corresponding CR1 and CR2 bit values are shown in Table 3.4

CR1	CR2	Timeout period (milliseconds)
0	0	16.384
0	1	65.536
1	0	262.14
1	1	1049

Table 3.4 Option register bit values and corresponding COP time periods

The time period chosen for the project system was the 262.14 millisecond period. This period was chosen because;

- It still represents a relatively short delay before a RESET that is unlikely to be noticed by the system user
- It allows for exceptionally long programs to still run without a RESET occuring.

The OPTION register is outlined below in Figure 3.9. Bits marked with an "X" do not affect the operation of the COP and therefore may be 0 or 1. The value of the OPTION register bits out of a system RESET is also shown.

OPTION register address = $1039

Bit number	7	6	5	4	3	2	1	0
Bit name	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
Bits for correct COP Time period	X	X	X	X	1	0	X	X
Value out of Reset	0	0	0	1	0	0	0	0

Figure 3.9 OPTION register setup for the COP

3.2.2.1.2 Enabling the COP

In order to enable the COP system the CONFIG register must be modified. The NOCOP bit of the CONFIG register must be cleared to zero in order to enable the COP The CONFIG register value with the COP enabled is shown in figure 3.10. Bits marked with an X do not affect the COP operation and may be either 0 or 1.

CONFIG register address = $103F

Bit number	7	6	5	4	3	2	1	0
Bit name	0	0	0	0	NOSEC	NOCOP	ROMON	EEON
Bits for correct COP Time period	0	0	0	0	X	0	X	X
Value out of Reset	0	0	0	1	0	0	0	0

Figure 3.10 CONFIG register settings to enable the COP

Use of the EVBU has prevented the modification of the CONFIG register. This has prevented testing of the COP system.

3.3 Setting up the Analog-to-Digital converter System

The Analog to Digital (A/D) converter system needs to be setup before it can be used. To ensure successful operation the A/D converter system the operating system must;

- Choose whether the Analog to digital converter system will be clocked by a special on-chip RC oscillator or the system clock.
- Enable the Analog to digital converter Charge pump
- Stabilize the charge pump before allowing any analog to digital conversion.

3.3.1 Choosing the clocking method for the Analog to digital converter

The analog to digital converter may be clocked either by a special on-chip oscillator or by the system clock. The RC oscillator should be used when the system clock is running to slowly to ensure accurate conversions. The HC11 M68HC11 Reference Manual (1991, p. 12-12) states that the system E clock should not be used for A/D conversion when operating below frequencies 750 kHz.

In the case of this project the system, the E clock will be operating at a speed of 2 MHz. This system clock speed is high enough to operate the A/D converter system. When the A/D converter system is clocked by the system clock the conversion sequence is synchronized to the main HC11 timers. There are benefits resulting from using the system clock, instead of the on-chip RC oscillator, to drive the A/D converter. The benefits are;

- The comparator output is sampled at relatively quiet processor times reducing error due to internal noise
- Result-register updates automatically occur during a period when reads do not occur preventing interference between updates and reads

The system clock will be used to drive the analog to digital converter system.

The CSEL control bit in the OPTION register is used to determine the A/D converter clock source. The CSEL bit must be set to 0 to cause the system E clock to drive the A/D converter system. The OPTION register is shown below in figure 3.11. Bits marked "X" do not have an effect on the choice of clocking for the A/D converter system.

OPTION register address = $1039

Bit number	7	6	5	4	3	2	1	0
Bit name	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
Bits for correct clock source	X	0	X	X	X	0	X	X
Value out of Reset	0	0	0	1	0	0	0	0

Figure 3.11 OPTION register settings for selecting the A/D converter clock source

3.3.2 Enabling the Charge pump

The HC11 has a charge pump that is an essential part of the A/D converter system. In order for the A/D converter to be used, the charge pump must be switched on. The charge pump is always switched off by a system RESET. The charge pump is controlled by the ADPU bit in the OPTION register. To switch on the charge pump the ADPU bit must be set to 1. The OPTION register setup for allowing A/D conversion is shown below if figure 3.12. Bits marked X do not affect the operation of the A/D converter charge pump.

OPTION register address = $1039

Bit number	7	6	5	4	3	2	1	0
Bit name	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
Bits for A/D	1	X	X	X	X	0	X	X
Value out of Reset	0	0	0	0	0	0	0	0

Figure 3.12 OPTION Register settings to enable A/D conversion

The A/D converter charge pump requires time to stabilize after being switched on. If A/D conversion takes plus before the charge pump has stabilized then errors in the conversion will occur. It is important to ensure there is a delay between switching on the A/D converter charge pump and making any A/D conversions. According to the HC11 M68HC11 Techical Data (1991, p. 10-5), stabilisation of the A/D charge pump requires a delay of at least 100μs.

Since the project system will be operating at an E clock speed of 2 MHz, a 200 clock cycle delay is required between switching on the charge pump and making any A/D conversions. The delay is implemented by placing a delay loop immediately after the code that switches on the A/D converter. The delay loop code is shown below.

```
LDAA #$40
CHARGED:   DECA
BNE CHARGEAD
```

3.4 Initializing the stack

The HC11 has a stack that is used for;

- Storing temporary data
- Passing procedure parameters
- Storing return addresses.

Normally the operating system software must initialize the stack by assigning a value to the stack pointer. The EVBU has a predefined stack area and as a consequence no stack initialization is done by any of the code used in this project. If a different HC11 variant were to be used the stack would need to be initialized.

3.5 Updating the procedure local data offset

Throughout all of the code in this project procedure local data is referenced by the indexed addressing mode. The index register Y has been used for this purpose. The address of any local data is referenced by adding the local variable offset to the current value of the index register Y. All local variable offsets are assigned in the individual procedures concerned and must be positive values. When one procedure calls another, the index register must be incremented by a sufficient amount to avoid memory corruption by the next procedure's local data. It is the responsibility of the calling procedure to increment the index register Y by the correct amount.

The operating system is implemented in the mainline code. This mainline does most of the procedure calling. The local data to the operating system is the global data to the whole system. During the initialization part of the mainline the index register must be given a value such that any local variables of called procedures do not corrupt the system global variables. At present the highest address used for a global variable in any of the project program versions is $0017. The index register Y is currently initialized to a value of $0020 to allow for future expansion. The code to initialize the local variable index register Y is shown below.

```
LDY #$0020
```

3.6 Calling procedures

The HC11 has two instructions designed for calling procedures. These two instructions are called JSR an RTS. All procedures are called using the JSR instruction. All returns from procedures are implemented using the RTS instruction

3.6.1The JSR instruction

The long name for the JSR instruction is "Jump to Subroutine." A JSR instruction causes the current program counter value to be stored in the top two locations of the stack and a new value to be placed in the program counter. This effectively causes a software jump. The JSR instruction can be used in the extended, direct or indexed addressing modes. Register values before the jump occurs are not stored. A full explaination of the JSR instruction can be found in the HC11 M68HC11 Reference Manual (1991, p. A-62).

3.6.2 The RTS instruction

The long name for the RTS instruction is "Return from Subroutine." A RTS instruction causes the program counter to take on the value stored on the top two locations of the stack. A full explaination of the RTS instruction can be found in the HC11 M68HC11 Reference Manual (1991, p. A-85).

3.7 Resetting Warnings after each complete loop

Warnings are fully explained in the "Warning System" section. At the beginning of each mainline loop, before any input handling procedures are called, the operating system is required to set all of the "warnings" to the value "System not handled." This is achieved by giving each "warning" the hexadecimal value $05. The "master warning" is the only exception and is assigned hexadecimal value $00 instead.

The code to set the "master warning" is shown below.

```
LDAA #$00
STAA Master_Warning
```

The code to set the other "warnings" to "system not handled" is shown below.

```
LDX #Warning
LDAB #$05
LDAA #$00
Clear_Warnings:
STAB 0, X
INX
INCA
CMP #09
BLS Clear_Warnings
```

3.8 Setting up the Prescaler

It has been decided that the timer prescaler must cause the timer system to operate at a speed four times slower than the system clock. Reasons for this choice are explained in the "RPM system" section.

The prescaler factor is chosen by altering the prescaler bits, PR1 and PR2, in the TMSK2 register. These bits are used to select the prescaler divide-by ratio. The bits can only be written once and the write must occur within 64 cycles after RESET.
A table showing the prescaler divide-by ratio versus prescaler bits is Table 3.5.

Prescaler bits PR[1:0]	Prescaler divide-by ratio
0 0	1
0 1	4
1 0	8
1 1	16

Table 3.5 Prescaler divide-by ratio versus prescaler bits relationship

To achieve the desired prescaler divide-by ratio the PR1 bit must be cleared to 0 and the PR0 bit must be set to 1. This is shown in figure 3.13 which illustrates the TMSK2 register. Bits marked "X" do not affect the timer prescaler.

TMSK2 register   Address = $1024

Bit number	7	6	5	4	3	2	1	0
Bit name	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
Value required to set the Prescaler correctly	X	X	X	X	0	0	0	1
Value out of reset	0	0	0	0	0	0	0	0

Figure 3.13 TMSK2 register values to setup the prescaler

The code to set the prescaler is shown below.

```
LDAA #$01
STAA $1024
```

It is important to remember that this code must be executed within the first 64 clock cycles after a reset.

CHAPTER 4: Warning System

Aircraft instruments often have color coded regions on them indicating different operating ranges. An example of this color coding, for the case of an aircraft tachometer, is described by Yeo, Bowers and Bennett (1996);

> Tachometers are marked in 100 rpm increments and are colour coded. A green arc indicates the normal operating range and a red line indicates maximum permissible rpm. Some aircraft tachometers have a yellow band within the green arc which denotes a range in which engine vibration may occur that could damage aircraft components if allowed to continue. Continuous operation is therefore prohibited in this range. (p. 97)

The objective of the Warning System is to emulate the coloured arcs on aircraft instruments. This section explains;

- How the coloured arcs are emulated. This is done in the section headed Warnings
- How the values of the Warnings are displayed. This is done in the section headed The Warning Display Circuit.

4.1 Warnings

The Warning System achieves its objective by associating a "warning" with each system input. There are five warning values. The "warning" values are explained below in Table 3.6.

Warning Name	Description	Corresponding hexadecimal value in memory
Green	This is the system warning when the system is operating within its normal range.	00
Low Red	The HC11 has detected that the system is operating below its normal operating range. This could be the result of a monitoring system failure. In the case of an RPM monitoring system it could mean the aircraft engine is switched off.	01
Yellow	This is equivalent to the yellow arc on some aircraft instruments. A yellow warning is invoked when the system is operating in a range that is higher than normal.	02
Red	The HC11 has detected that the system is operating above its allowable operating range. This could also be the result of a monitoring system failure.	03
System not handled	No procedure is operating on the warning at this memory location. The System not handled warning level is explained further under the heading "System not Handled."	05

Table 3.6 Warning levels

Currently there are 10 consecutive, 1 byte, memory locations set aside to hold warnings. These consecutive memory locations are referred to as the "Warning Page". Each A/D converter input pin has been allocated 1 location. A single location has been allocated for the RPM system. A further location has been set aside for an extra, yet unknown, input. The first location of the Warnings page is pointed to by a constant value pointer named "Warning"

4.1.1 System Not Handled

At the start of each mainline cycle all of the warnings are set to "system not handled." When a system input procedure is called it modifies the warning corresponding to the system input that the procedure is handling. At the end of one mainline cycle any warnings still set to "system not handled" have not been modified by an input system procedure. This is useful in software debugging. The "system not handled" warning is also useful to quickly determine which input pins are currently in use.

4.1.2 Warning priority

For any one system input there can only be one warning value. It is possible for the conditions required for more than one warning value to be satisfied at any one time. A common case is when an input value is high enough for a warning to be red. In this case the input is also high enough for the warning to be yellow. A priority system has been developed to determine the warning in the case of more than one warning value-condition being satisfied. The priorities are shown in table 3.7. Higher priority warning values override lower priority ones. For example, if the conditions for both red and the yellow warning values are satisfied, the warning will be set to red. All procedures that handle inputs determine their corresponding warning based on this priority scheme.

Warning	Priority
Red	4
Low red	3
Yellow	2
Green	1
System not handled	0

Table 3.7 Warning priorities

4.1.3 Master Warning

The "master warning" is different from the other warnings. The master warning may only have two values. The two allowable values for the master warning are on and off. The master warning is set to off if all other system warnings are green, yellow or system not handled. If any system warning is set to low red or red then the master warning is set to on. At the start of each mainline cycle the master warning is set off. Any procedure setting a warning to a low red or red value is also responsible for setting the master warning to on. Setting the master warning to on is accomplised by calling the procedure "Set_Master"

4.1.3.1 Set_Master Procedure

The master warning is set on by calling the Set_Master procedure. Upon termination of the Set_Master procedure all register values remain the same as at the time of calling the Set_Master procedure.

4.1.3.2 MasterWarning representation in memory

The master warning is represented as a single byte memory location. This byte is called Master_Warning in the system code. The allowable values and their corresponding hexadecimal representations are shown below in Table 3.8

Master warning value	Hexadecimal representation
On	01
Off	00

Table 3.8 Master warning values and their representations

4.2 The Warning display circuit

The warning display circuit is a simple circuit used to give visual information about the warning level for one of the A/D input pins. In a real aircraft system this circuit would not be included. In a real aircraft system the HC11 would send warning levels to the output processor. For the project purposes however it is useful to include this circuit to give visual information about how the system is operating. The circuit consists of four different coloured LEDs, each representings one possible warning value. There is also an extra red LED that is used to show the status of the system's Master Warning.

4.2.1 LED colour versus warning relationship

Only one of the four coloured LEDs may be lit at one time. The decision about which LED is to be lit is made in software procedure Drive_Led. The Drive_Led procedure decides which LED is to be lit based on the current warning level of a single input. Table 3.9 outlines the relationship between the LED lit and the current warning value.

Current Warning Value	Colour of the LED lit
Low Red	Orange
Green	Green
Yellow	Yellow
Red	Red
System not handled	No LEDs lit

Table 3.9 LED colour vs warning level

4.2.2 The master warning LED

The master warning value is displayed through the use of a second red LED. When the Master Warning is set on, the red is switched on. When the Master Warning level is set off the red LED remains switched off.

4.2.3 How the LEDs are driven

The LEDs are to driven by the port B pins of the HC11. The actual port B pin numbers and the EVBU header pin numbers for each LED are outlined in Table 3.10.

LED colour	Port B pin number	EVBU header pin number
Master warning red LED	4	38
Red	3	39
Yellow	2	40
Orange	1	41
Green	0	42

Table 3.10 LED connections

4.2.4 Warning Circuit Design

The Warning circuit consists of 5 LED circuits. Each LED circuit consists of a LED and a current limiting resistor R. The cathodes of all the LEDs are connected to the system ground. One end of the current limiting resistor R is the connected to the appropriate port B pin. A single LED circuit is shown in Figure 3.14.

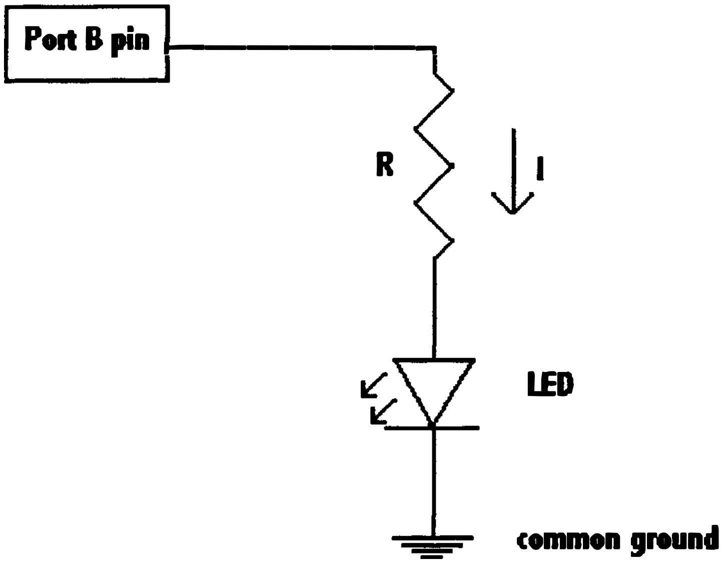


Figure 3.14 Single LED circuit

4.2.4.1 Determination of the value of the Resistor R

The normal operating current for a LED is about 20 mA. The approximate operating voltage for a LED is dependant on the colour of the LED. The operating voltage for the LED colours used in the warning display circuitry varies from 1.8 to 2.2 Volts. The exact operating voltage is not critical and for the circuit design an operating voltage of 1.8 Volts was chosen for all the LEDs. The voltage generated between the HC11 port B pins and ground ,by a logic 1, is 5 volts. Since the operating current in known and the node voltages are known it is possible determine the value of the resistor R. The value of the resistor R was determined as follows;

$$R = \frac{V_{in} - V_L}{I}$$

$$R = 160\Omega$$

where

V_{in} = the HC11 port B pin voltage = 0V

V_L = The voltage developed across the LED = 1.8V

I = the circuit current = 20mA

The HC11 is only capable of supplying a small current to all of its output pins. The HC11 may need to drive both the master warning LED and one other LED at a single time. If each LED is to be driven by a 20mA current then the HC11 needs to supply 40mA. It is unreasonable to expect the HC11 to supply 40 mA.

To reduce the LED current a larger value for the resistor R is required. Experiments were conducted to determine a higher value for the resistor R that would still cause the LEDs to light brightly enough when the circuit was driven by a 5 volt supply.

4.2.4.1.1 The final resistor value R and its consequences

The final value for the resistor R was chosen to be 1.2KΩ. There is no cause for this resistor value to be highly accurate so a 5% tolerance resistor was used. The larger value for the resistor R results in the LEDs glowing less brightly than usual. The LEDs still glow brightly enough to be observed easily in normal lighting conditions. With the resistor value R set to 1.2KΩ the HC11 is able to reliably drive up to two LEDs. Attempting to drive more than 3 LEDs causes the HC11 to RESET. It is possible to drive 3 LEDs but the HC11 is inclined to RESET after a random interval of time.

CHAPTER 5: Analog-to-Digital Conversion System

Several of the project subsystems require an analog voltage signal to be converted into a digital signal. This conversion is necessary because software can only operate on digital information. The Analog-to-Digital (A/D) conversion system's primary purpose is to convert voltages present at the A/D input pins to a digital value so as to allow the proper function of the;

- Battery Voltage System
- Cylinder Head Temperature System
- Ambient Temperature System

The A/D conversion system provides the link between the hardware and software sections of each of these three subsystems. This relationship is shown is Figure 3.15.

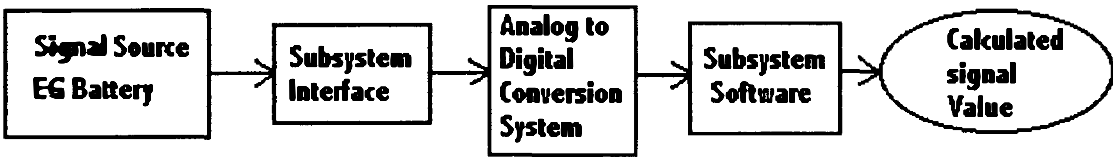


Figure 3.15 A/D Conversion system role in subsystem dataflow.

The A/D Conversion System is depicted in Figure 3.16. The input to the system is an analog voltage in the range 0 to 5V. The output from the system is a one byte hexadecimal value corresponding to the input voltage.

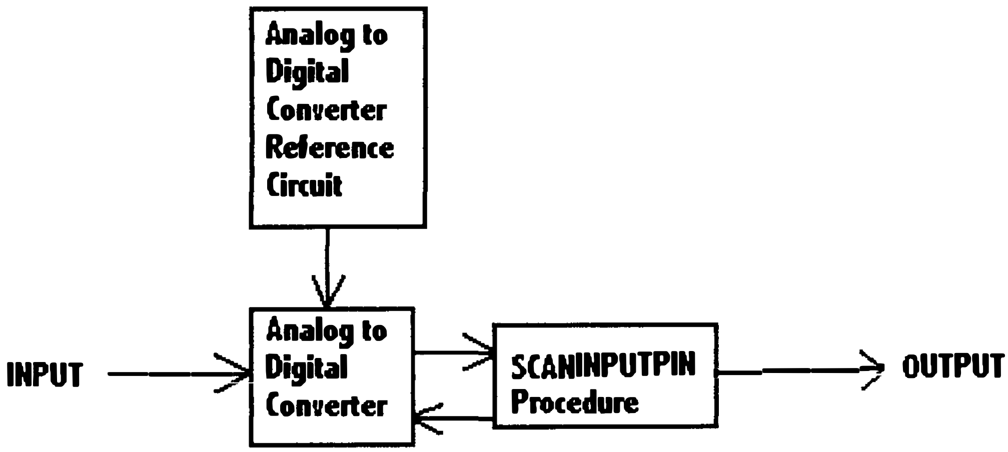


Figure 3.16 A/D Conversion System

The three major components of the A/D Conversion System, as illustrated in Figure 3.16, are described in the sections;

- Analog-to-Digital Converter
- A/D converter Reference Circuit
- SCANINPUTPIN Procedure

5.1 Analog-to-Digital Converter

The A/D converter is an essential part of the A/D conversion system. The A/D converter reads an Analog input Voltage and uses the Charge-Redistribution method to convert this voltage into a digital value. The concepts of A/D conversion and Charge-Redistribution are explained in the Relevant Theory Section.

In the project system several input sources are connected to the A/D converter via interface circuitry. In general the resulting circuit takes the form shown in Figure 3.17. All voltages present at a A/D converter input pin are referenced to ground. The input signal may or may not be referenced to ground.

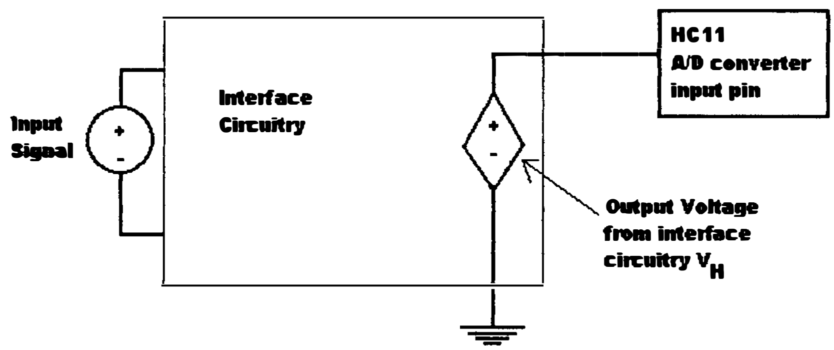


Figure 3.17 Circuitry connecting an input source to the HC11

The A/D Converter needs both hardware and software support to operate. Hardware is required because the A/D converter needs;

- A Reference Voltage with which to compare the input Voltage signal
- Input to determine the A/D conversion resolution.

Software is required because the A/D converter needs control inputs such as;

- When to perform sampling
- How many samples to take

Software is also required to read and store data generated by the A/D converter.

In the project system, hardware support is provide by the Analog-to-Digital Converter Reference Circuit and software support is supplied by the SCANINPUTPIN Procedure.

5.2 Analog-to-Digital Converter Reference Circuit

The A/D Converter Reference Circuit, shown in Figure 3.18, performs two tasks necessary to the operation of the A/D converter. These tasks are;

- Setting the A/D converter voltage reference point
- Setting the A/D converter voltage resolution and input range

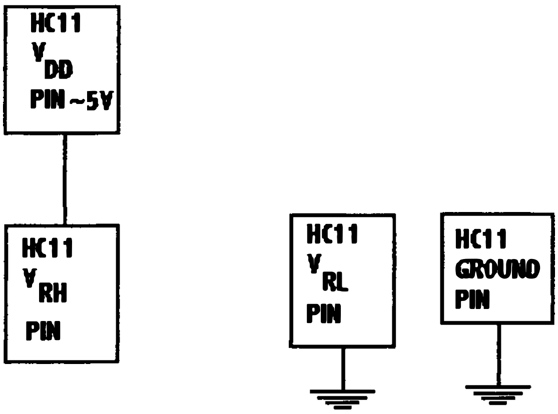


Figure 3.18 A/D Converter Reference Circuit

5.2.1 Setting the Reference Point

The A/D converter uses the Voltage present at the V_{RL} pin of the HC11 as the reference point for all A/D conversions. The project system requires all A/D inputs to be referenced to ground and this is achieved by connecting the VRL pin to ground as shown in Figure 3.18.

5.2.2 Setting the Voltage Range and Resolution

The A/D converter is operated over a 5 volt range resulting in a quantizing level step size of 0.01953125V. The A/D converter operating range is determined by the voltage difference between the V_{RH} and V_{RL} pins of the HC11. The voltage range of the A/D converter is set by connecting the V_{RH} pin to the V_{DD} pin as shown in Figure 3.18.

The two reasons for choosing a 5 volt operating range are;

- Motorola guarantees the A/D converter accuracy at the 5 volt range
- The V_{DD} pin of the HC11 provides an easy to obtain 5 volt reference thus simplifying the A/D Converter Reference Circuit.

5.3 SCANINPUTPIN procedure

The SCANINPUTPIN's primary task is to use the A/D digital converter to determine a hexadecimal value that best represents the analog voltage present at one of the A/D input pins. This section explains this procedure by;

- Describing the tasks the procedure performs
- Outlining the procedure input parameters
- Outlining the procedure local variables
- Outlining the procedure Output parameters
- Providing a description of all the parameters and local variables
- Describing how the procedure improves the signal to noise ratio

5.3.1 Procedure Tasks

- Accept as a parameter the number of the A/D pin to scan
- Accept as a parameter a number equal to ¼ of the A/D conversions to be performed. (Explained later in the sections "NumberOfScans" and "Averaging A/D results to reduce the effects of noise"
- Perform the appropriate number of A/D conversions
- Average the A/D conversions to improve the signal to noise ratio
- Return the average to the main program

5.3.2 Input Parameters

The input parameters for this procedure are outlined in Table 3.11.

Parameter name	Where the parameter is stored during passing
PinNumber	Accumulator A
NumberOfScans	Accumulator B

Table 3.11 input parameters for SCANINPUTPIN

.

5.3.3 Local variables

The local variables that are used in this procedure and their offsets are outlined in Table 3.12.

Local variable name	Size of the variable in bytes	Memory offset for the local variable	Address offsets covered by the variable
PinNumber	1	$00	$00
NumberOfScans	1	$01	$01
Counter	1	$02	$02
Total	2	$03	$03,$04

Table 3.12 local variables for SCANINPUTPIN

5.3.4 Output parameters

The SCANINPUTPIN procedure has one output parameter. The parameter name and where it is stored during a return from the SCANINPUTPIN procedure is shown below in table 3.13.

Output parameter name	Where parameter is stored during passing
Average	AccA

Table 3.13 Output parameters for SCANINPUTPIN

5.3.5 Explanations of parameters and local variables

5.3.5.1 PinNumber

The HC11 has 8 analog input pins numbered from 0 through to 7. Each time the procedure SCANINPUTPIN is called it scans one of these analog input pins. The variable PinNumber holds a value between 0 and 7 that determines which analog input pin is scanned.

5.3.5.2 NumberOfScans

In the HC11 A/D conversion operations are performed in groups of four. The four results from these conversions are stored in the result registers ADR1 to ADR4. The SCANINPUTPIN procedure averages the values in the four results registers. For the purposes of this explanation the process just outlined is referred to as an "A/D scan." and the average that is calculated at the end of an "A/D scan" is called the "scan-average."

The SCANINPUTPIN procedure can be set to perform multiple "A/D scans." The SCANIPUTPIN procedure then averages the "scan-averages" from each of the "A/D scans" to finally determine the number corresponding to A/D pin Voltage. The variable NumberOfScans is an integer value determining the number of "A/D scans" to perform. Since the result of each "A/D scan" is stored on the stack the maximum value for NumberOfScans is limited by the available stack space. NumberOfScans is a one byte variable

IMPORTANT NOTE:
If the stack is allowed to overflow the program will fail. It is therefore vitally important to ensure that the NumberOfScans specified will not result in a stack overflow. The maximum value for NumberOfScans will be dependant on the stack size and therefore is dependant on the system where this procedure is used.

5.3.5.3 Counter

Counter is a variable that keeps a record of how many "A/D scans" have been performed. Counter is initially set to the same value as NumberOfScans. The counter is decremented one for each "A/D scan" performed. Scanning ceases when counter equals zero. Counter is a one byte variable

5.3.5.4 Total

Total is a variable that is used in the process of averaging the A/D scans. Total is a 2 byte variable

5.3.5.5 Average

Average is the hexadecimal value representing the voltage present at the A/D pin being scanned as determined by the procedure SCANINPUTPIN. Average is a single byte in size.

5.3.6 Improving the Signal to Noise Ratio

The voltage measured by the A/D converter is determined by the signal being measured and by environmental noise. The signal to noise ratio can be improved by using signal averaging. In the averaging process several A/D conversions are performed and then their results are averaged. This process performs a similar function to a Low Pass filter.

The minimum number of A/D conversions that are averaged to determine a final result in the SCANINPUTPIN procedure is four. Four conversions are averaged when the input parameter NumberOfScans is set to one. It is possible to average the results of a larger number of A/D conversions, increasing the level of software filtering. This is done by specifying a larger number for the input parameter NumberOfScans.

5.3.7 Algorithm

The algorithm for the procedure SCANINPUTPIN is shown in appendix E.

5.3.8 Code

The code for the procedure SCANINPUTPIN is shown in the file ver4sub* in appendix F.

CHAPTER 6: Battery Voltage System

The Battery Voltage System's primary purpose is to determine the voltage of a 12 volt battery and store this value in memory. The system also performs the task of setting the warning corresponding to battery voltage. The battery system is supported in its function by the operating system and the Analog-to-Digital Conversion System. The Battery Voltage System is essential to the operation of the Warning System. The relationship between the Battery Voltage System and the other subsystems is depicted in Figure 3.1

To achieve its objectives the Battery Voltage System implements the dataflow shown in Figure 3.19. The Operating System and the Analog-to-Digital Conversion Systems are described in their respective sections. This section will describe;

- The 12 Volt Battery System specifications
- The 12 Volt Battery Interface
- The 12 Volt Battery System Procedure
- The testing of the 12 Volt Battery System

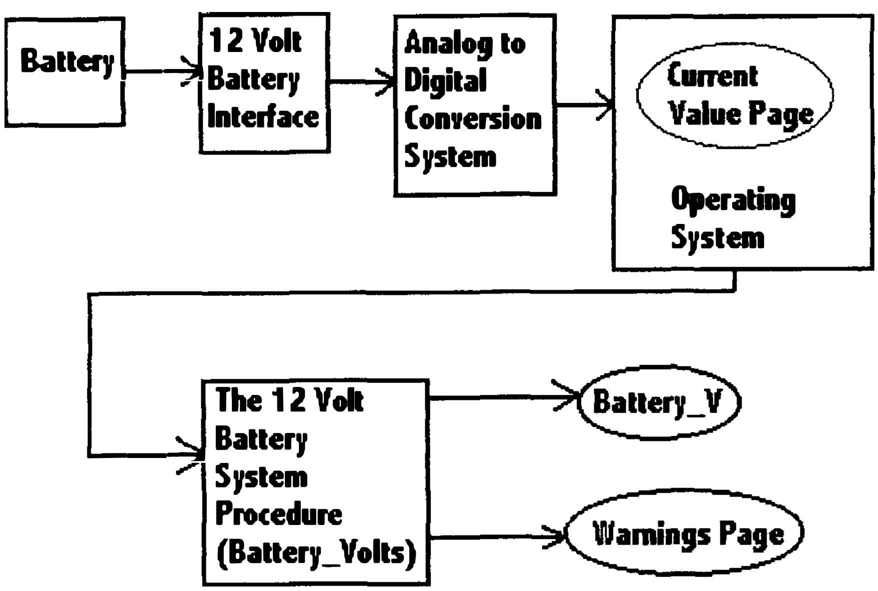


Figure 3.19 The Battery Voltage System

6.1 The Battery Voltage System Specifications

The system specifications are as shown in Table 3.14;

Interface Input Voltage	
Operating Range	0 to 16 Volts
Maximum negative voltage due to accidental terminal reversal that will not cause damage	-16 Volts
Interface Output Voltage	
Interface Output Voltage Range under normal conditions	0 to 5 Volts
Accuracy	
Maximum specified voltage variation from real voltage	±0.25 Volts
Power Consumption of 12 Volt Battery Interface	
Power consumption at typical operating voltage = 12.6 Volts	0.05 Watts
Worst case power consumption Voltage = 16 Volts	0.08 Watts

Table 3.14 Battery Voltage System Specifications

6.2 The 12 Volt Battery Interface

The 12 Volt Battery Interface is required to allow the HC11 to determine battery voltage. The major role of the circuit is to scale the battery voltage to a level that can be read by the HC11 A/D converter. The Interface must protect the HC11 A/D converter from damage by large voltages. It is desirable for the battery interface to draw as little current from the battery as is practically possible. The battery interface has been designed with the above considerations. The 12 Volt Battery Interface consists of a single circuit. The design of this circuit will now be explained.

6.2.1 Input Voltage Range

The 12 Volt Battery Circuit will be required to measure voltage over the range attainable by a 12 volt lead acid battery. According to (Lead Acid Battery, n.d., p.6) a maintenance free battery can reach a voltage of 16 volts while charging. This will be the upper limit of the input voltage for the 12 volt battery circuit. It is desirable for the system to be able to detect when no battery has been connected to the 12 volt battery circuit. Consequently the minimum voltage within the battery circuit operating range will be 0 Volts. This voltage will generate 0 Volts at the appropriate HC11 A/D input pin.

It is possible that the battery could be connected to the circuit backwards by accident. The circuit has been designed to protect the HC11 A/D converter in the case of accidental reversing of polarity. A negative voltage of up to 16 volts will not damage the system.

6.2.2 Determining the required scaling factor

The HC11 A/D converter can measure voltages in the range of 0 to 5 volts. The Input range for the 12 volt battery circuit is 0 to 16 volts. The circuit must provide a linear relationship between input voltage and output voltage. The 12 volt battery circuit determines the output voltage by scaling the input voltage by a certain factor. The required scaling factor of the 12 volt battery circuit was determined as follows:

$$\text{Scaling factor required} = \frac{\text{Maximum HC11 input voltage}}{\text{Maximum battery circuit input voltage}}$$

$$\text{Scaling factor required} = \frac{5}{16}$$

$$\text{Scaling factor required} = 0.3125$$

6.2.3 Voltage Divider

A voltage divider is a simple resistive circuit that can be used to deliver an output voltage at some fraction of the input voltage. A simple voltage divider is shown in Figure 3.20.

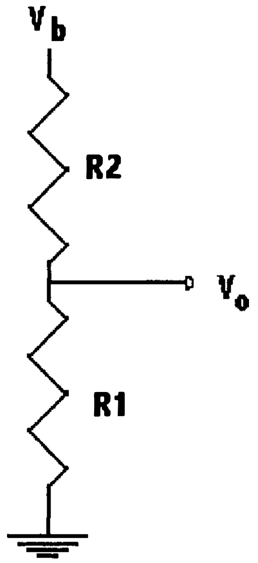


Figure 3.20 Voltage Divider

The output voltage Vo is related to the input voltage Vb by the following formula.

$$V_o = V_b\left(\frac{R1}{R1+R2}\right)$$

A voltage divider will be used to scale the battery voltage to a level that can be read by the HC11 analog to digital converter.

6.2.4 Determining the resistor ratio to achieve the required voltage scaling

To use a voltage divider to scale the input voltage it is necessary to determine the required ratio between the two voltage divider resistors. This ratio was determined as follows:

Let

V_o = HC11 analog input pin voltage

Vb = Voltage at the input of the 12 volt battery circuit

R1= value for the resistor R1 in ohms

R2 = Value for the resistor R2 in ohms

We require:

$$V_o = 0.3125 * V_b$$

Substituting this value into the voltage divider formula yields:

$$0.3125 * Vb = Vb\left(\frac{R1}{R1+R2}\right)$$

This can be solved to determine the relationship between the two resistor values.

R2 = 2.2 * Rl

6.2.5 Determining the value of the resistor R1

The choice of the value for the resistor R1 will involve a compromise between:

- The level of protection provided to the HC11 A/D converter
- The power consumption of the 12 volt battery circuit
- The accuracy of the voltage measurements made by the HC11.

The minimum allowable value for the resistor Rl will be determined by the level of protection required by the HC11 and the need to minimize the circuit power consumption. High values for the resistor R1 will reduce the accuracy of the measurement made by the A/D converter system so the maximum allowable value for the resistor R1 will be determined by system accuracy requirements.

6.2.5.1 Determining the minimum value of the resistor R1

A diagram showing the 12 volt battery circuit connected to an HC11 analog input pin is shown below in Figure 3.21. The internal input protection device for the HC11 A/D converter is also shown in Figure 3.21.

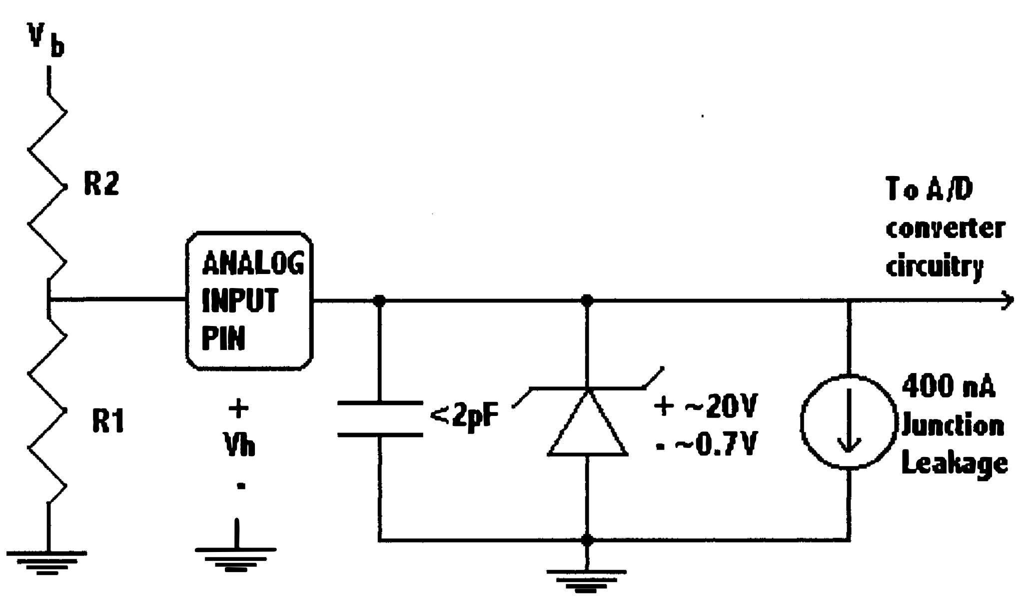


Figure 3.21 The 12 volt battery circuit and the HC11 A/D converter protection circuit

The resistor R1 must be of sufficient resistance so as to protect the HC11 A/D converter in the case of accidental reversal of battery circuit input voltage. The worst case situation would be a battery circuit input voltage of –16 volts.

Protection of the A/D converter is achieved by limiting the current through the A/D converter input pin. The current must be restricted to avoid CMOS latchup which will destroy the A/D converter. To prevent latchup, the current at a pin should still be limited to 25mA or less.

The worst case situation, where a –16 Volt input voltage is connected to the 12 volt battery circuit, is depicted in the equivalent circuit shown in figure 3.22.

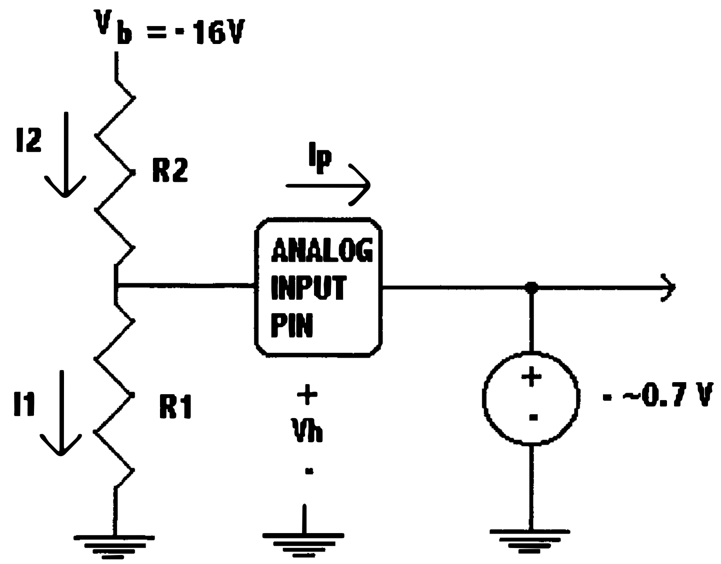


Figure 3.22 Worst case: Accidental reversal of battery voltage.

The voltage divider would normally cause the voltage at the HC11 analog input pin to be –5V. In this situation, the input protection zener diode clamps the voltage at the analog input pin to about –0.7 volts. The current through the analog input pin can be calculated as shown below.

$$I_p = I_2 - I_1$$

$$I_p = \frac{-15.3}{R2} - \left(\frac{-0.7}{R1}\right)$$

Since we know that R2= 2.2 * R1 we can eliminate R2 and simplify to get;

$$I_p = \frac{-6.255}{R1}$$

Using a maximum value for the HC11 pin current is then possible to determine a minimum safe value for the resistor R1.

$$0.025 \geq \frac{-6.255}{R1}$$
$$R1 \geq 250\Omega$$

The minimum value for the resistor R1 that will protect the system from the accidental connection of -16 Volts to the battery circuit is 250Ω.

6.2.5.2 Determining the maximum value for the resistor R1.

According to the HC11 M68HC11 Referance Manual (1991a, p.12-16), a series resistor of more than 10kΩ will degrade A/D converter accuracy. The loss of accuracy is due to the junction leakage shown in Figure 3.21. The resistor R2 is in series between the analog input pin and the battery voltage. This means the maximum allowable value for the resistor R2 is 10kΩ. Since it has already been decided that the resistor R2 must have a value 2.2 times greater than the resistor R1 the maximum allowable value for the resistor R1 is 4.55kΩ.

6.2.6 The final resistor values for R1 and R2

The final value chosen for the resistor R1 was 1 kΩ. This value is between the allowable range of 250Ω and 4.55kΩ. The resulting value for R2 then becomes 2.2 kΩ. Both 1 kΩ and 2.2 kΩ resistors are common resistor values and therefore easily and cheaply obtainable.

6.2.6.1 The 12 volt battery system power consumption

The resulting worst case current drain on the aircraft battery will occur when the battery voltage, V_b, is 16 volts. Under worst case conditions the current drawn from the battery by the 12 volt battery circuit is 5mA. This worst case current, I_d, was determined using the formula shown below. R_{total} is the combined resistance of the resistors R1 and R2 in series.

$$I_d = \frac{V_b}{R_{total}}$$

The power consumption of the circuit can be calculated using the formula;

$$P=V*I$$
$$P=16*0.005$$
$$P= 0.08 \text{ Watts}$$

6.2.7 Tolerances of the resistors R1 and R2.

The tolerance of the resistors affect the accuracy of the battery voltage measurements. If 5% tolerance resistors are used then the largest possible error, due to resistor tolerance, in the calculated battery voltage would be 1.12Volts. If 1% tolerance resistors are used the largest possible error, due to resistor tolerance, will be 0.22 volts. The maximum error due to resistor tolerance will occur when the battery voltage is 16 volts. The method for calculating the error in measured voltage due to resistor tolerance is shown in appendix D.

6.2.7.1 Tolerance Values chosen for a real aircraft system.

The battery voltage part of the integrated aircraft system must determine the voltage of the aircraft battery to a level of accuracy that makes the system useful. According to (Lead Acid Battery, n.d., p.5) a 0.25 ±0.05 V change in the stabilized open circuit voltage of a lead acid battery corrisponds to a percentage charge change of 25%. The change in the stabilised open circuit voltage from a percentage charge of 100% to 0% is 0.9Volts. For open circuit battery voltage information to be useful it must be accurate to 0.25 V. This cannot be guaranteed when resistor tolerances of 5% are used. In worst case 1% tolerance resistors can cause an error in calculated battery voltage of 0.22. Therefore, in the stabilised open circuit case, 1% tolerance resistors are needed to ensure that the measured battery voltage is useful.

6.2.7.2 Tolerance of the resistors in the prototype circuit

The prototype 12 volt battery circuit was constructed on a breadboard using 5% tolerance resistors. Higher tolerance resistors were used because these were readily available at the time of circuit construction. All test data related to the 12 volt battery circuit comes from this prototype circuit.

6.3 The 12 Volt Battery System Procedure (Battery_Volts)

The 12 volt battery system procedure is referred to as "Battery_Volts." The primary objective of this procedure is to determine the voltage of a 12 volt battery, based on the voltage present at the A/D converter pin connected to the 12 Volt Battery Interface. The secondary objective of the this procedure is to set the warning corresponding to the battery voltage.

6.3.1 Tasks

- Receives as a parameter the number of the A/D pin connected to the 12 volt battery circuit
- Reads the hexadecimal representation of the voltage at the appropriate A/D input
- Sets "warnings" based on the voltage present at the appropriate A/D input
- Calculates battery Voltage based on the voltage at the A/D input
- Stores the calculated battery voltage in Battery_V, a global variable.

6.3.2 Required Data

In order to operate properly the procedure code needs to have access to the values:

- CurrentVal
- Warning
- Battery_V
- BALow_Red
- BAYellow
- BARed

The definitions of the required data for the procedure Battery_Volts are explained below.

6.3.2.1 CurrentVal

The hexadecimal values for the voltages at each of the analog to digital converter pins is stored in 8 consecutive addresses. These 8 addresses are referred to as the Current Value page. The lowest address contains the value representing the voltage at Port E pin 0. The next address contains the value representing the voltage at Port E pin 1 and so forth. The constant CurrentVal is a pointer to the first address of the Current Value page.

6.3.2.2 Warning

The constant Warning, is a pointer to the first address of the Warnings page. The Battery_Volts procedure requires this pointer so that it can update the warning corresponding to the battery voltage.

6.3.2.3 Battery_V

Battery_V is a pointer to the location where the battery voltage is stored. The procedure needs this pointer so that it can store the calculated battery voltage in the correct address.

6.3.2.4 BALow_Red

This constant is the value that determines whether a Low Red warning is set. If the A/D pin voltage is equal to or below this value then a Low Red warning is set.

6.3.2.5 BAYellow

This constant is the value that determines whether a Yellow warning is set. If the A/D pin voltage corresponding to the battery circuit is above or equal to this value a Yellow warning is set.

6.3.2.6 BARed

This constant is the value that determines whether a Red warning is set. If the A/D pin voltage corresponding to the battery circuit is above or equal to this value a Red warning is set.

6.3.3 Required procedures

In order to operate correctly the Battery_Volts procedure requires the existence of the Set_Master procedure. The starting address of the Set_Master must be available to the Battery_Volts procedure so that the Battery_Volts procedure may call the Set_Master procedure.

6.3.4 Input Parameters

The input parameter for the Battery_Volts procedure is outlined below in Table 3.15.

Parameter name	Where the parameter is stored during passing
BAPin	Accumulator A

Table 3.15 Input parameters for the Battery_Volts procedure

6.3.5 Local variables

The local variables that are used in this procedure and their offsets are outlined in Table 3.16.

Local variable name	Size of the variable in bytes	Memory offset for the local variable	Address offsets covered by the variable
BAPin	1	$00	$00

Table 3.16 Local variable for the procedure Battery_Volts

6.3.6 Values that may be changed at the conclusion of Battery_Volts

There are three values that may be changed at the conclusion of the Battery_Volts procedure. These values are explained under the headings Battery_V, Battery Warning and Master_Warning.

6.3.7 Definition of Values Changed, Parameters and Variables

6.3.7.1 BAPin

The 12 volt battery circuit is connected to one of the A/D input pins. The Battery_Volts procedure needs to know which A/D input pin is connected to the battery circuit. The parameter and local variable BAPin specifies which A/D pin is connected the 12 volt battery circuit. Legal values for BAPin range from 0 to 7. BAPin is one byte in size.

6.3.7.2 Battery_V

The pointer Battery_V points to the location where the calculated battery voltage is stored. At the conclusion of the Battery_Voltage procedure the calculated battery voltage is updated. The calculated battery voltage is stored in 2 consecutive bytes. The high byte of the battery voltage represents the integer part of the battery voltage. The low byte of the battery voltage represents the fractional part of the battery voltage. The low byte of the battery voltage is stored as a binary-weighted fraction.

6.3.7.3 Battery Warning

The 12 volt battery circuit is connected to one of the HC11 A/D pins. At the conclusion of the Battery_Volts procedure the warning corresponding to this A/D pin will be updated.

6.3.7.4 Master_Warning

If the warning corresponding to the 12 volt battery circuit A/D pin is set to the Low Red or Red conditions then the Master_Warning will also be set on.

6.3.8 How the Battery Voltage is calculated

The voltage at the A/D pin connected to the 12 volt battery circuit is related to the battery voltage by the following formula:

$$V_H = 0.3125 * V_B$$

where

V_H = Voltage at the HC11 A/D pin

V_B = Battery Voltage

The hexadecimal number corresponding to the voltage a the A/D input pin is related to the voltage at the A/D input pin by the following relationship;

$$H = \frac{256}{5} V_H$$

where

H = The hexadecimal value corresponding to the Voltage at the A/D pin

V_H = The voltage at the HC11 A/D pin

Putting these two equations together and rearranging them leads to a method of approximating the battery voltage. The approximation is based on the hexadecimal number corresponding to A/D input pin voltage. This is shown in the equation below;

$$V_B^* = \frac{H}{16}$$

As Hexadecimal numbers this becomes

$$\$V_B^* = \frac{\$H}{\$10.00}$$

where

V_B^* = the approximate battery voltage

The Battery_Volts procedure uses this approximation to determine the battery voltage.

6.3.9 Algorithm

The algorithm for the procedure Battery_Volts is shown in appendix E.

6.3.10 Code

The code for the procedure Battery_Volts is shown in appendix F.

6.4 Testing & Performance

The Battery Voltage System was tested for input voltages between 0 and 16.25 Volts. Overall the system performed as expected over the entire testing range. The testing methods and results are explained in the following sections.

Testing was conducted by using a variable voltage source to simulate battery voltages. The testing circuit is shown in Figure 3.23.

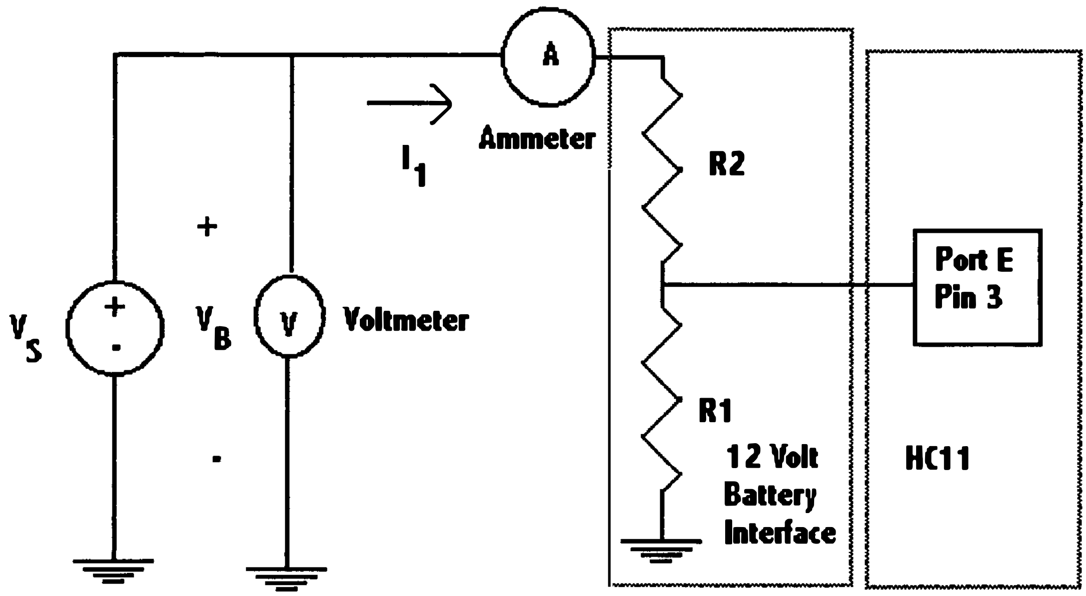


Figure 3.23 Testing circuit for the Battery Voltage System

The test circuit component Vs was a laboratory power supply. This supply was capable of providing a variable DC voltage between 0 and 20 Volts.

The testing of the 12 Volt Battery System was conducted according to the following method.

1. The software was setup to operate the Battery_Volts procedure for input at A/D pin number 3. This was required since the 12 Volt Battery Interface was connected to A/D pin 3.
2. The software was setup to run the Drive_Led procedure for input A/D pin number 2. This was required so that the part of the Battery_Volts procedure that sets warnings could be tested.
3. The test circuit was constructed as shown in Figure *
4. The Warning display circuit was connected to the HC11 in the normal configuration. This was done so that the warnings generated by the Battery_Volts procedure could be displayed.

5. The voltage on the laboratory supply was set to zero and the supply was then switched on.
6. A digital multimeter was used to measure the voltage V_B as shown in figure *. 3.23
7. A digital multimeter was used to measure the current I_1.
8. The HC11 system software was started.
9. The status of the warning display circuit LEDs was noted. This information was used to determine if the part of the Battery_Volts procedure that sets warnings was working correctly.
10. A HC11 system RESET was issued and the BUFFALO monitor program was used to determine the hexadecimal value corresponding to the voltage at the A/D input pin. This value is useful to determine how accurately the A/D converter and the SCANINPUTPIN procedure were determining the voltage at the A/D input pin.
11. The BUFFALO monitor program was used to determine the hexadecimal value of the Variable Battery_V. The variable Battery_V contains the calculated voltage of the battery. This value was used to determine how accurately the Battery_Volts procedure was calculating the battery voltage based on the voltage present at the A/D input pin.
12. The Voltage of the laboratory power supply was increased by a small amount. The procedure steps 6-12 were then repeated several times. The process stopped when the voltage V_B was 16.25V. A voltage of 16.25 volts corresponds to a battery voltage higher than the system is specified to handle. There was little value in testing the system further outside of its operating range.

The numerical results of the above testing procedure are shown in ~~table * of~~ appendix *. G These results are depicted graphically in Figure 3.24 through to Figure 3.26.

Figure 3.24 A/D converter Accuracy for the Battery Voltage System

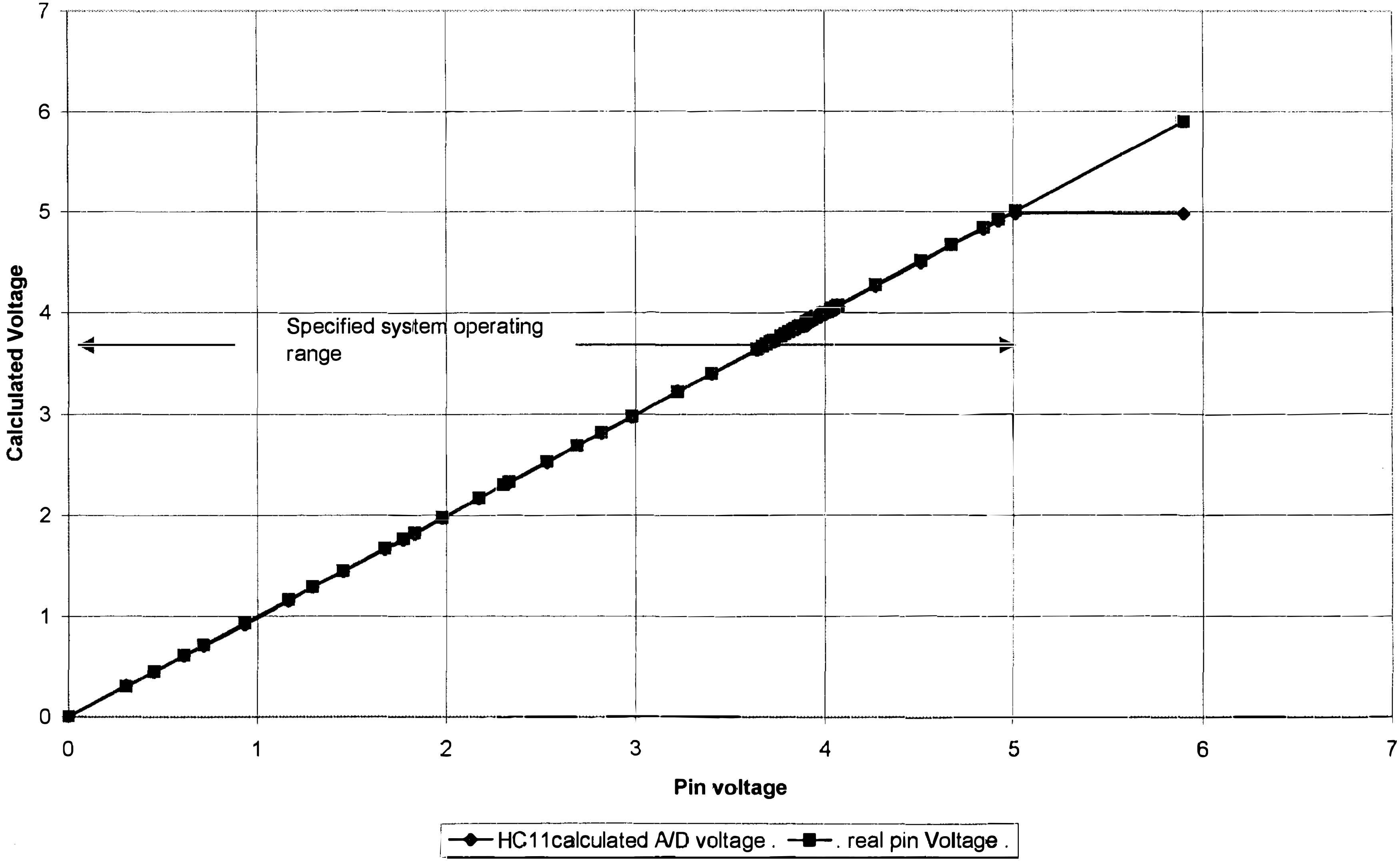


Figure 2.25 AD converter maximum error during testing

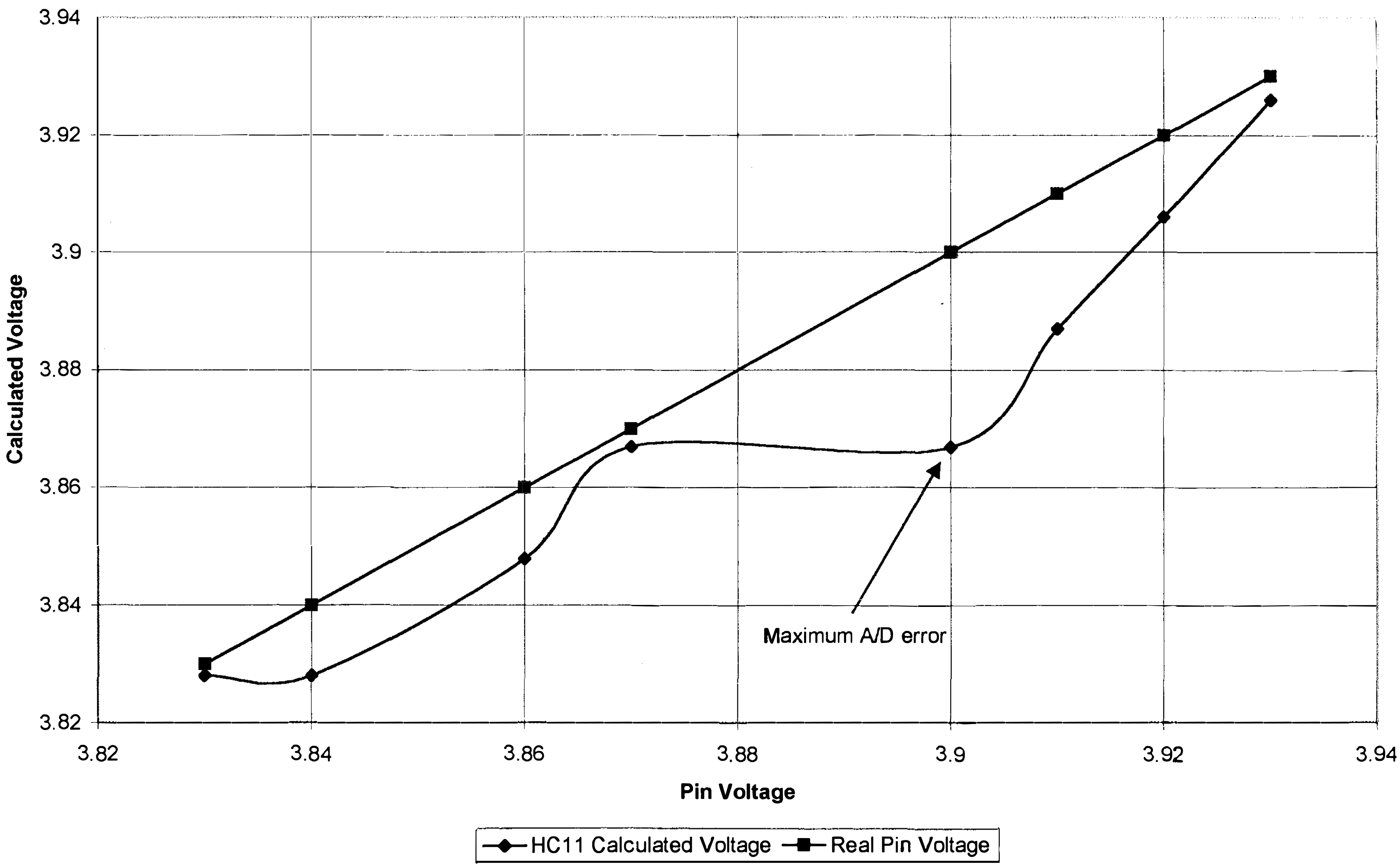
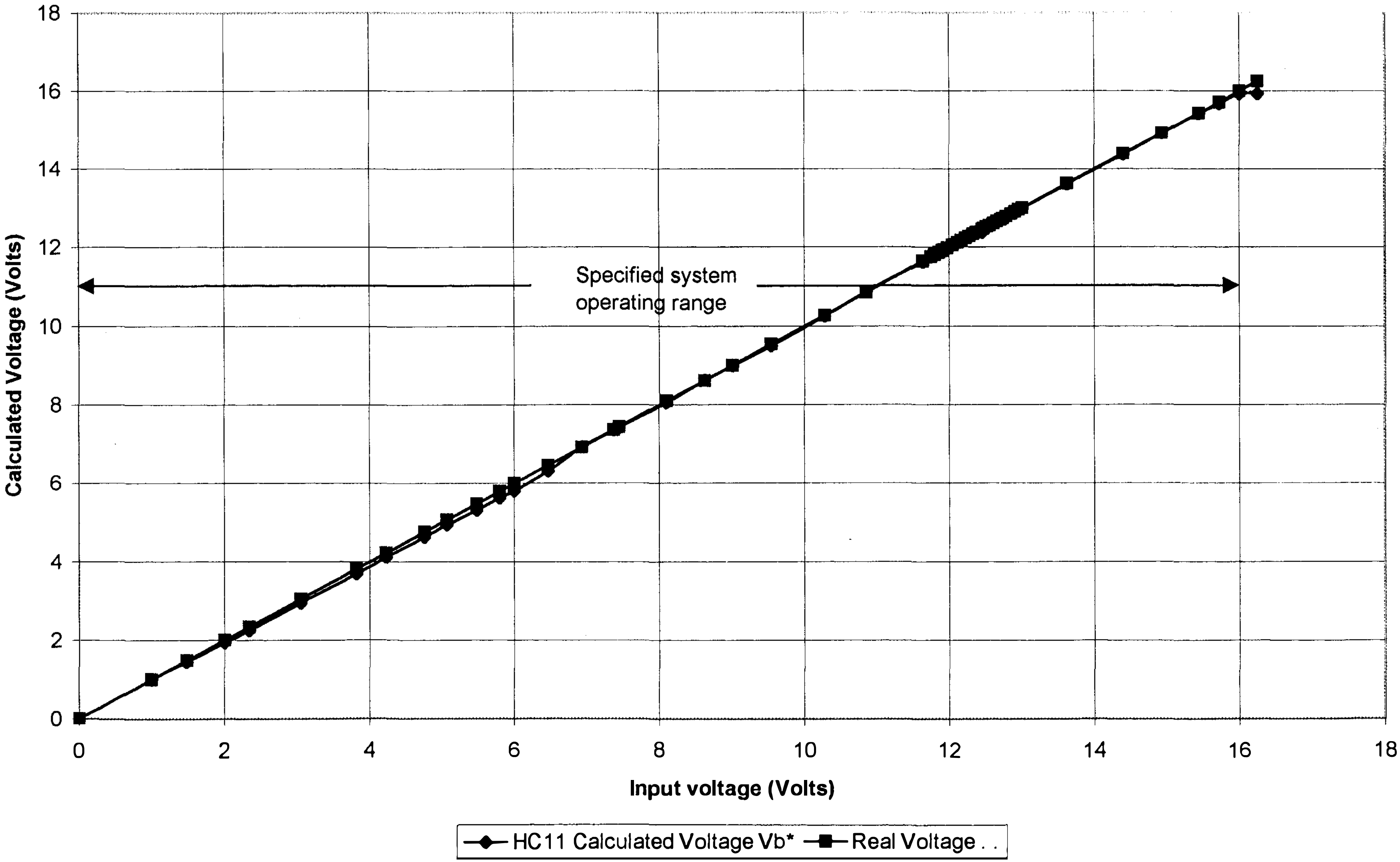


Figure 3.26 Battery Voltage System Accuracy



Overall the A/D converter performed as expected. Figure 3.24 compares the voltage determined by the Analog to digital converter with the voltage at the A/D converter pin as calculated from the current I_1. The difference between the A/D converter measured voltage and the actual input pin voltage is so small that it is difficult to discern the difference between the two lines in Figure 3.24. Figure 3.25 is an enlarged version of Figure 3.24 covering the section of data where the maximum error between A/D converter measured voltage and real pin voltage occurs. This maximum error was 0.033V. The A/D converter is setup with a quantization resolution of 0.01953125V. The maximum error corresponds to a 2 bit error in the A/D converter measured voltage.

From Figure 3.26 it can be seen that the Battery Voltage System determines very close to the real voltage at its input when the voltage is within the system operating range. The maximum error between the calculated voltage and the real voltage, that occurred within the system operating range, was 0.1875V.

6.4.1 Testing the system warnings

The trigger values for the Battery_Volts procedure were set to the following values during testing.

Warning constant name	Warning constant value
BALow Red	$00
BAYellow	$77
BARed	$FF

Table 3.17 Warning triggers

Warnings are set by the Battery_Volts procedure based on the A/D converter voltage. The orange LED of the warning display circuit was lit only when the voltage at the A/D pin was 0V. The green LED only became lit for the case were the hexadecimal value corresponding to A/D voltage was between $01 and $76. The Yellow LED was lit for the cases where the hexadecimal value corresponding to A/D voltage was between $77 and $FE. The Red LED only became lit for the cases where the hexadecimal value corresponding to A/D voltage was equal to $FF. These results coincided with expectations.

6.4.2 Significance of the Testing Results

The following conclusions have been determined from the testing of the Battery Voltage System;

- The warnings are being set properly by the Battery_Volts procedure
- The Battery Voltage System can be expected to calculate battery voltage accurate to ±0.1875V. This is within the specified operating accuracy of ±0.25V
- The calculated battery voltage is inaccurate for voltages outside the system operating range. As a result the system should not be operated outside of its specified range.

CHAPTER 7: Cylinder Head Temperature System

The primary objective of the Cylinder Head Temperature System is to determine engine cylinder head temperature and store this temperature in the system memory. The system also performs the secondary task of setting the warning corresponding to cylinder head temperature. The Cylinder Head Temperature System relies upon the Operating System and the Analog-to-Digital Conversion system. The Cylinder Head Temperature System is vital to the function of the Warning System. The relationship between the Cylinder Head Temperature System and the other subsystems is illustrated in Figure 3.1.

To achieve the system objectives the dataflow illustrated in Figure 3.27 was implemented.

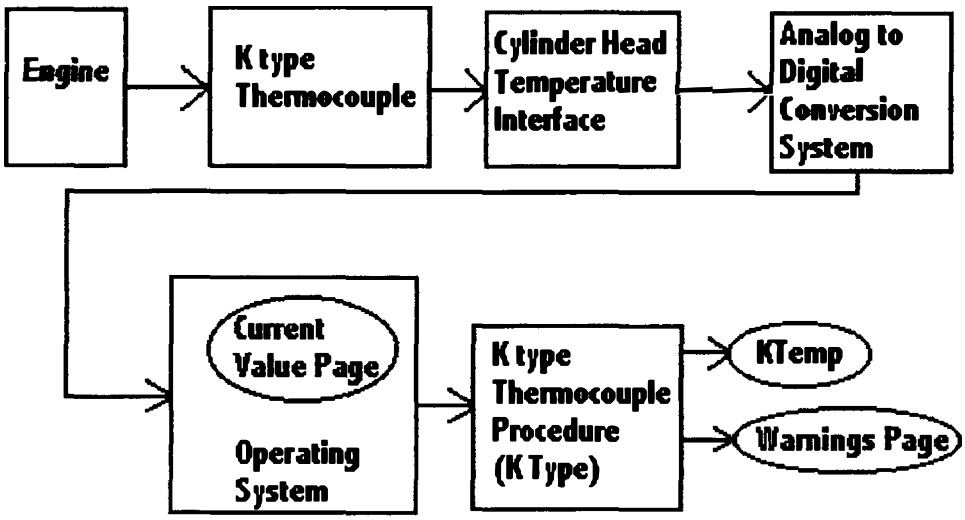


Figure 3.27 The Cylinder Head Temperature System

This section describes;

- The Cylinder Head Temperature System Specifications
- The K Type thermocouple
- The Cylinder Head Temperature Interface
- The Thermocouple Procedure (KType)
- The Testing of the Cylinder Head Temperature System

7.1 Cylinder Head Temperature System Specifications

Table 3.18 shows the specifications for the Cylinder Head Temperature System.

Input Temperature	
Operating Temperature Range	50 to 306°C
Interface input Voltage Range	
Input voltage range for the interface under normal conditions	0 to 10.3872mV
Maximum voltage at interface input that has been tested	29.8 mV
Interface Output Voltage	
Interface output Voltage range under normal operating conditions	0 to 5V
Maximum tested interface output voltage	5.97V
Accuracy	
Maximum specified Temperature variation from real temperature	4°C

Table 3.18 System specifications

7.2 K Type Thermocouple

The K type thermocouple is the most commonly used of the various different thermocouple types. The main reasons the thermocouple is so widely used are;

- its large temperature range
- its cheapness.

As with all thermocouples it is a passive temperature sensor and requires referencing to determine absolute temperature.

The K type thermocouple was chosen for measuring cylinder head temperature because;

- Unlike semiconductor temperature sensors the thermocouple can easily operate at 300°C, the maximum temperature an engine cylinder head is ever likely to reach.
- The thermocouple is cheaper than other thermocouples and RTDs
- The thermocouple has a large voltage/temperature coefficient when compared to other thermocouples.

7.2.1 Voltage Versus Temperature Approximation

The thermocouple voltage versus temperature relationship is not linear. A power series expansion for temperature versus voltage relationship is provided by TC LTD (1997, p. 9) and is reproduced in Appendix B. Nine coefficients for the power series expansion are provided. This power series expansion is difficult to implement in assembly language, a linear approximation would be preferable.

The approximate linear coefficient of the relationship between temperature at 100°C intervals from -200 °C to 1300°C is provided by Sheingold (1980, p. 132). The operating range for the thermocouple in the aircraft system will be from 0°C to 300°C. A single approximate linear coefficient has been calculated by averaging the coefficients for 0, 100, 200 and 300°C given in Sheingold (1980, p. 132). The coefficients and their average are shown below in Table 3.19.

Temperature or average	Coefficient µV/°C
0	39.5
100	41.4
200	39.9
300	41.5
Average	40.575

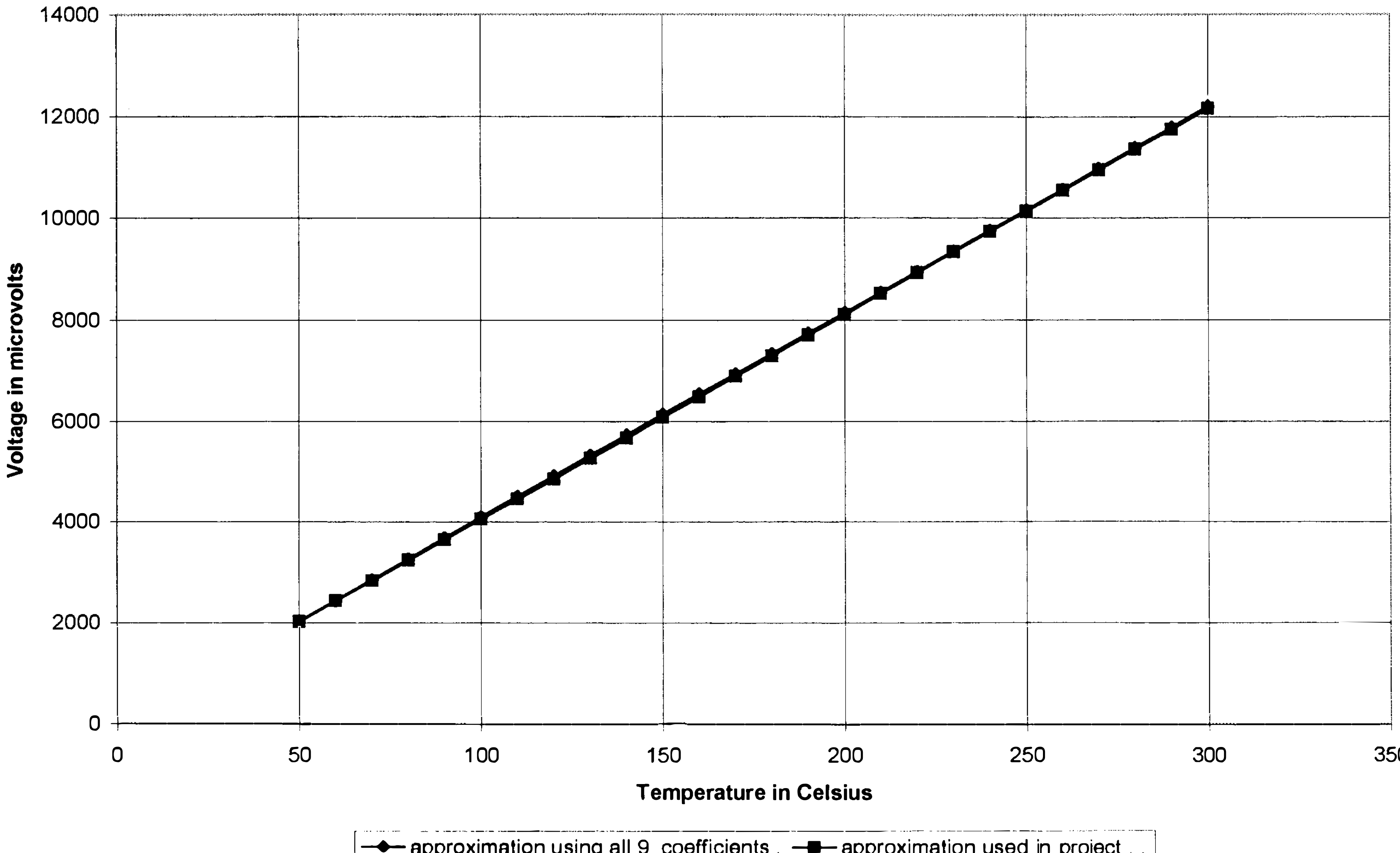
Table 3.19 Coefficients and there average
Data taken from Sheingold (1980, p. 132)

The calculated average coefficient can be used to determine a linear approximation that relates thermocouple voltage to thermocouple temperature. The equation is shown below.

Let V = thermocouple voltage
Let T = temperature in Celsius
$$V = 40.575 * 10^{-6} * T$$

Microsoft Excel was used to evaluate the expected thermocouple voltage versus various temperatures using the nine terms of the power series approximation. Excel was also used to evaluate the expected thermocouple voltage for the same temperatures using the above linear approximation. The numerical results of this exercise are shown in appendix G. From the Excel results the maximum error between the two approximation methods was 57.50µV. This corresponds to a temperature error of less than 2°C. A graph showing the results from the two methods is shown in figure 3.28 on the following page.

Figure 3.28 Comparison between 9th order approximation and project linear approximation



Over the range of temperatures that the K type thermocouple will specified to measure, the maximum error between the linear approximation outlined and the power series approximation was less than 60μV, corresponding to an error of less than 2°C. The linear approximation method is far easier to use. The HC11 software calculates cylinder head temperature from the thermocouple voltage based on the linear approximation outlined above.

7.3 Cylinder Head Temperature Interface

The Cylinder Head Temperature Interface performs the tasks of;

- Boosting the thermocouple Voltage to a level that can be read by the HC11
- Improving the signal to noise ratio present at the A/D input pin of the HC11

The Cylinder Head Temperature Interface consists of two circuits;

- An Instrumentation Amplifier
- A Low Pass Filter

. The Cylinder Head Temperature Interface is shown in Figure 3.29.

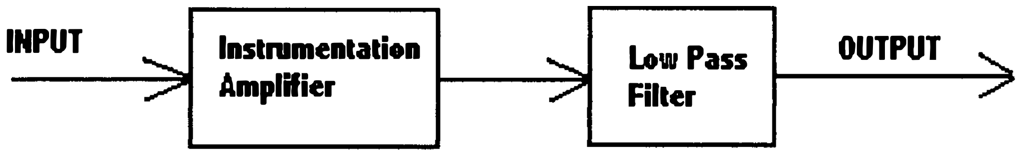


Figure 3.29 Cylinder Head Temperature Interface

7.3.1 Instrumentation amplifier

The HC11 analog to digital converter is unable to effectively measure the small signals developed by the K type thermocouple. An instrumentation amplifier is required to boost the K type thermocouple signal to a level readable by the HC11.

An instrumentation amplifier is a differential amplifier often used in instrumentation. The amplifier boosts the differential signal present at its input terminals by a pre-set value. An ideal instrumentation amplifier has an infinite common mode rejection ratio. Unfortunately no instrumentation amplifier is ideal.

Desirable characteristics of an instrumentation amplifier include;

- High input impedance
- Low input offset
- Low input offset drift
- Low nonlinearity
- Stable gain
- And low output impedance

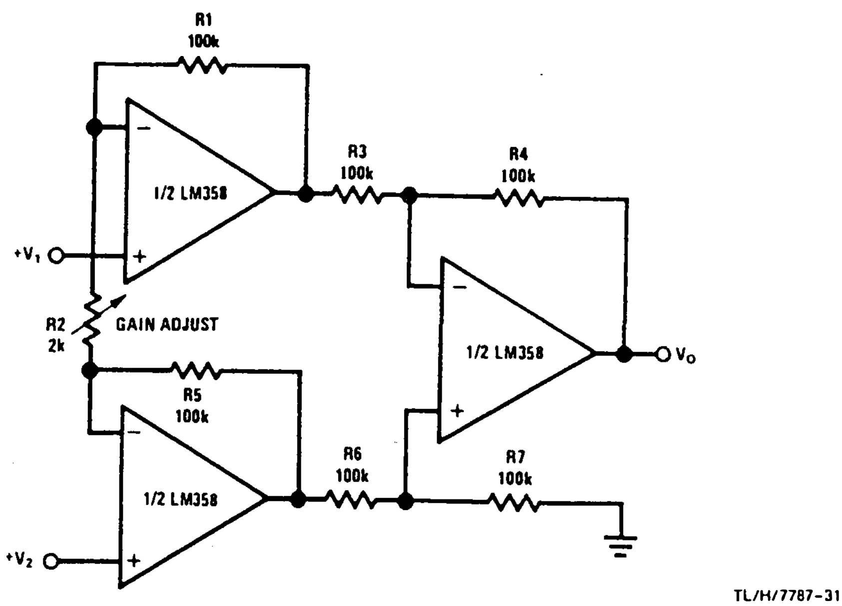
Instrumentation amplifiers are useful in other applications as well as interfacing thermocouples. Some other common uses of instrumentation amplifiers include the interfacing of strain gage bridges, current shunts, and biological probes. Instrumentation amplifiers also find uses in the preamplification of small differential signals superimposed on large common-mode voltages.

Instrumentation amplifiers come in a number of different forms. It is possible, for example to purchase instrumentation amplifiers as pre-built circuit modules. It is also possible to purchase instrumentation amplifiers as integrated circuits. Instrumentation amplifiers can also be user-assembled circuits based on operational amplifiers. The instrumentation amplifier used in this project will be a user designed and assembled circuit. In a real aircraft system a military-aerospace specified integrated circuit instrumentation amplifier should be used because integrated circuit units are smaller, lighter and far more reliable.

7.3.1.1 The Classical Instrumentation Amplifier Circuit

The classical instrumentation amplifier circuit is based around 3 operational amplifiers. The circuit contains two stages. Figure 3.30 depicts the classical instrumentation amplifier circuit.

High Input Z Adjustable-Gain DC Instrumentation Amplifier



If R1 = R5 & R3 = R4 = R6 = R7 (CMRR depends on match)

$$V_O = 1 + \frac{2R1}{R2} (V_2 - V_1)$$

As shown $V_O = 101\ (V_2 - V_1)$

Figure 3.30 Classical Instrumentation amplifier.

The first stage consists of two operational amplifiers. The first stage generally performs the tasks of;

- Providing the required instrumentation amplifier gain
- Providing a high input impedance
- Providing a low output impedance source to the second stage

The second stage performs the task of taking the difference between the outputs of the first stage and thus rejecting the common-mode signal. The second stage is a difference amplifier circuit based around a single operational amplifier. Implementing higher gains than one in second stage leads to a lowering of the input resistance of the stage. Low input resistance is an undesirable property in amplifiers. The second stage of the instrumentation amplifier is generally designed for a gain of one due to this input resistance problem.

7.3.1.2 Input Voltage Range

The instrumentation amplifier input voltage range requirements are based on the physical properties of the K type thermocouple and the temperature range over which the thermocouple will be measuring. The thermocouple will be expected to measure temperatures in the range of 50 to 306°C. The average thermocouple voltage change per C° change over the range of temperatures 50 to 306°C was determined in the Voltage versus Temperature Approximation section. This average voltage change was 40.575 µV/°C. Over the operating temperature range there is a 10.3872mV change in voltage. The input voltage range for the instrumentation amplifier is 10.3872mV.

7.3.1.3 Determining the amplifier gain

The HC11 A/D converter can measure voltages in the range of 0 to 5 volts. The input voltage range for the instrumentation amplifier is 10.3872mV. The instrumentation amplifier will provide a linear relationship between input and output voltage within its operating range. The instrumentation amplifier is expected to supply all the gain between the thermocouple and the HC11 A/D pin, all other elements of the Cylinder Head Temperature Intergace interface will have their gains set to one. The required amplifier gain was determined as follows;

$$\text{Ampifier gain} = \frac{\text{HC11 input Voltage range}}{\text{K type thermocouple voltage range}}$$

$$\text{Ampifier gain} = \frac{5}{0.0103872}$$

$$\text{Ampifier gain} \approx 481$$

7.3.1.4 The project Instrumentation Amplifier

The project instrumentation amplifier circuit is depicted in figure 3.31. The operational amplifiers A1 and A2 form the first stage of the amplifier. Amplifier A3 forms the second stage of the instrumentation amplifier.

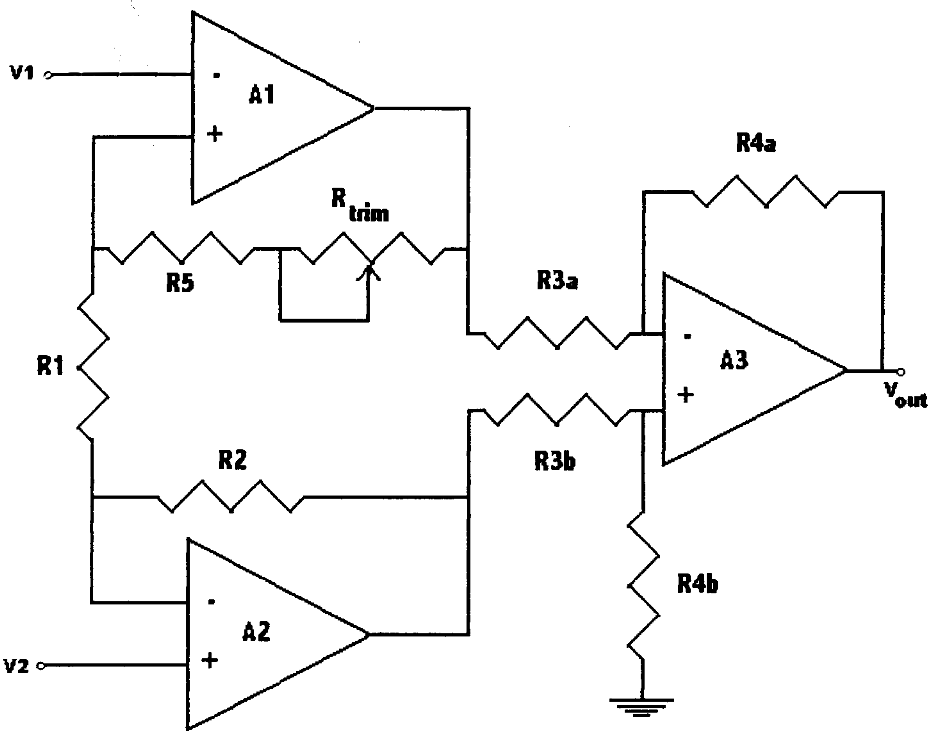


Figure 3.31 The project instrumentation amplifier

7.3.1.5 Determining the Values of the Resistors R3 and R4.

The resistors R3a, R3b, R4a and R4b determine the gain of the second stage of the instrumentation amplifier. Resistors R4a and R4b must be of the same value to ensure balance. Resistors R3a and R3b must also be the same value. The second stage of the instrumentation amplifier is to have a gain of one. The differential gain of the second stage is related to the resistor values R3 and R4 by the following equation;

$$G_2 = \frac{R4}{R3}$$

where G_2 = the second stage gain

It is obvious from the second stage gain equation that to achieve the required second stage gain of one the resistors R3a, R3b, R4a and R4b must be all the same value. The remaining task is to choose a "practically convenient value" for the resistors (Sedra & Smith, 1998, p. 91). Sedra and Smith suggest that 10KΩ is such a "Practically convenient value. The value chosen for the resistors R3a, R3b, R4a, R4b is 10KΩ.

Determining the values for the resistors R1, R2, R5 and R_{trim}

The resistors R1, R2, R5, and R_{trim} determine the balance and the gain of the first stage of the instrumentation amplifier. The required first stage gain is 481V/V. The gain of the instrumentation amplifier is related to these resistor values by the following equation;

$$G_1 = 1 + \frac{R2 + R5 + R_{trim}}{R1}$$

where G_1 = the gain of the first stage

The instrumentation amplifier is balanced when;

$$R2 = R5 + R_{trim}$$

The value of the resistors R2 and R1 were determined assuming that the instrumentation amplifier is balanced. The ratio between the resistors R2 and R1 was determined as follows;

$$G_1 - 1 = \frac{R2 + R5 + R_{trim}}{R1}$$

since the amplifier is assumed to be balanced

$$\frac{G_1 - 1}{2} = \frac{R2}{R1} = 240$$

The final chosen values for the resistor R1 and R2 was 1 KΩ and 240KΩ respectively. These values were chosen because they represented the best compromise between reducing the required output current of the two operational amplifiers and maintaining a high enough circuit current to provide immunity to noise.

The values chosen for the resistor R5 and the trimpot R_{trim} were 235KΩ and 10KΩ respectively. These values were chosen so that the effects of varying the circuit gain and balance could be investigated if desired.

7.3.1.6 Choosing the Resistor Tolerance Values

The tolerance of the resistors used in the instrumentation amplifier circuit is a determining factor in the accuracy of the circuit gain. The accuracy of the instrumentation amplifier circuit gain is a determining factor in the accuracy of the thermocouple temperature measurements. The tolerance of the resistors used in the instrumentation amplifier circuit must be chosen dependant on the required accuracy of the temperature measurements.

While determining resistor tolerances it is assumed that the resistors R5 and R_{trim} are replaced with a resistor of the same type as R2 called R2b. This was done to simplify the design process.

The two worst cases that need to be considered are the maximum gain that can occur due to resistor tolerances and the minimum gain that can occur due to resistor tolerances. The resistor values for these worst cases are given in Table 3.20. Table 3.20 shows the worst cases for resistor tolerances of both 1% and 5%. The resulting gain of each stage, as well as the total gain for the instrumentation amplifier, are also shown in Table 3.20.

Case	R1 KΩ	R2 KΩ	R2b KΩ	R3a KΩ	R3b KΩ	R4a KΩ	R4b KΩ	G_1	G_2	Total Gain
Maximum gain for 5% tolerance	0.95	252	252	9.5	9.5	10.5	10.5	531.52	1.11	589.9872
Minimum gain for 5% tolerance	1.05	228	228	10.5	10.5	9.5	9.5	435.29	0.90	391.7571
Maximum gain for 1% tolerance	0.99	242.4	242.4	9.9	9.9	10.1	10.1	490.70	1.02	500.5140
Minimum gain for 1% tolerance	1.01	237.6	237.6	10.1	10.1	9.9	9.9	471.50	0.98	462.0700

Table 3.20 The effects of resistor tolerance on the instrumentation amplifier gain

The error in the thermocouple temperature readings is dependent on the error in the instrumentation amplifier gain. The worst case error occurs at the highest temperature of 306°C. The worst case errors in the thermocouple temperature measurement due to variations in instrumentation amplifier gain are shown in Table 3.21.

Case	Gain	Measured temperature when at 306 °C	Worst temperature error at 306°C
Maximum gain for 5% tolerance	589.9872	375	+69 °C
Minimum gain for 5% tolerance	391.7571	249	-57 °C
Maximum gain for 1% tolerance	500.5140	318	+12 °C
Minimum gain for 1% tolerance	462.0700	294	-12 °C

Table 3.21 Worst case thermocouple temperature measurement error

As can be seen in the data shown in Table 3.21, an exceptionally large error in measured temperature could result if 5% tolerance resistors are used. The error can be greatly reduced by using 1% tolerance resistors. The amplifier design used in the project uses 1% tolerance resistors.

7.3.1.7 Choosing the Operational Amplifiers for the Instrumentation Amplifier

Before going further it should be noted that the operational amplifiers here are not aerospace specified. Therefore none of the operational amplifiers mentioned here would be used to construct an instrumentation amplifier for a real aircraft. In a real aircraft an aerospace specified integrated circuit instrumentation amplifier would be the most likely solution to this problem. It is still useful however to see the design process involved in the breadboard instrumentation amplifier.

Five operational amplifiers were considered for the breadboard implementation of the instrumentation amplifier. These operational amplifiers are;

- LM324
- LM301
- LM358
- LM627
- LM741

The spec sheets for each of these amplifiers is included in Appendix A

Each stage of the instrumentation amplifier has different requirements. The operational amplifiers were evaluated for each stage of the instrumentation amplifier. The operational amplifiers were evaluated based on;

- Availability
- Maximum Input offset voltage and possibility of offset compensation.
- Component size
- Power supply requirements
- Cost

7.3.1.7.1 Availability

The LM301 was found not to be easily available. As a result the LM301 was eliminated from further consideration.

7.3.1.7.2 Input Offset Voltage and Offset Compensation

The input offset voltage of the op-amps used in the instrumentation amplifier has a direct effect on the accuracy of the temperature measurements taken from the thermocouple. The input offset voltage of the op-amps has a more pronounced effect in the first stage of the instrumentation amplifier.

7.3.1.7.3 Effect of Stage Gain on Sensitivity to Voltage offsets

The gain of each instrumentation amplifier stage governs the amount of temperature measurement error that an offset voltage in the operational amplifiers in that stage will cause. Larger stage gains result in higher sensitivity to input offset voltages. A change in the output of the instrumentation amplifier of 0.01953125 corresponds to a 1°C change in calculated temperature. Stage two of the instrumentation amplifier has a gain of one. An offset voltage of 19.53125mV in the stage two op-amp will correspond to a 1°C change in temperature. The first stage of the instrumentation amplifier has a gain of 481. An offset voltage of 40.23129μV in either of the first stage op-amps will result in a 1°C change in temperature.

7.3.1.7.4 Input Offset Voltage and Compensation

The maximum offset voltage of an operational amplifier is generally specified at a temperature of 25°C. This was the case for the op-amps investigated. The instrumentation amplifier will not always be operating in an environment where the temperature is fixed. Operational amplifier offset voltage is subject to drift with changes in temperature. It is important to also consider the maximum offset voltage that can occur at the maximum and minimum operating temperatures.

It is assumed that the instrumentation amplifier will not be operating in environment temperatures below 0°C. The instrumentation amplifier is assumed not to be operating in environmental temperatures above 50°C. There is a 25°C difference between the minimum operating temperature and the temperature where the maximum offset voltage is specified. There is also a difference of 25°C between the maximum operating temperature and the temperature where the maximum offset is specified. It is necessary to allow for a change of 25C° to determine the maximum voltage offset that can occur during operation. The maximum voltage offset that can occur during operation was calculated according to the method explained in the section headed Low Pass Filter under the heading Input offset voltage and offset compensation. The maximum offset voltages for each of the operation amplifiers is summarised in Table 3.22.

Op-amp	Maximum offset voltage V_{off}
LM324	7.750mV
LM358	7.5mV
LM627	0.125mV
LM741	6.375mV

Table 3.22 Operational amplifier maximum offset voltages

The LM627 and the LM741 allow offset compensation. This allows a reduction of the maximum offsets of these two op-amps to 0.015mV and 0.375mV. The cost of offset compensation is an increase in the number of components used in and the complexity of the instrumentation amplifier circuit.

7.3.1.7.5 Temperature Measurement Error versus Offset Voltage for Stage 1

The worst case temperature measurement error introduced at stage one of the instrumentation amplifier due to op-amp offset voltage is shown in Table 3.23. This error was calculated as follows;

$$T_{error} = \text{maximum error in calculated temperature}$$

$$V_{off} = \text{maximum op-amp offset voltage}$$

$$\frac{\partial T}{\partial V} = \text{change in calculated temperature per volt} = 24627.2°\text{C/V}$$

$$A = \text{number of operational amplifiers in the stage}$$

$$T_{error} = A * V_{off} * \frac{\partial T}{\partial V}$$

Op-amp and mode of operation	Maximum offset voltage in mV	Worst error induced in °C
LM324	7.750	381.72
LM358	7.5	369.41
LM627 (uncompensated)	0.125	6.16
LM627 (compensated)	0.015	0.74
LM741(uncompensated)	6.375	313.00
LM741 (compensated)	0.375	18.48

Table 3.23 worst case temperature measurement error due to stage 1 op-amp offset voltage

7.3.1.7.6 Temperature Measurement Error Versus Offset Voltage for Stage 2

The worst case temperature measurement error introduced at stage two of the instrumentation amplifier due to op-amp offset voltage is shown in Table 3.24. The error was calculated using the same method as for stage 1 except that the change in calculated temperature per volt of stage 2 is 51.2 and the number of amplifiers in the stage is 1.

Op-amp and mode of operation	Maximum offset voltage in mV	Worst error induced in °C
LM324	7.750	0.3968
LM358	7.5	0.3840
LM627 (uncompensated)	0.125	0.0064
LM627 (compensated)	0.015	0.0008
LM741(uncompensated)	6.375	0.3264
LM741 (compensated)	0.375	0.0192

Table 3.24 Worst case temperature measurement error due to stage 2 op-amp offset voltage

7.3.1.7.7 Component Size

It is desirable to implement both the instrumentation amplifier and the low pass filter on the same breadboard. The breadboard has limited space. As a consequence of the limited breadboard size smaller components are desirable. The LM627, LM358 and LM741 are available in 8 pin DIPs. The LM358 is available in a 14 pin DIP.

7.3.1.7.8 Power Supply Requirements

The LM324 and LM358 have an advantage of being able to operate with either a single or a double power supply. The LM741 and the LM627 both require two power supplies.

7.3.1.7.9 Cost

The cost analysis for the op-amps was done based on the Altronics catalogue (1997, p .61). The LM627 is unavailable from Altronics and its price is based on the receipt received from Dick Smith electronics for the purchase of 2 such op-amps. The retail price per op-amp for purchases of less than 10 op-amps and more than ten op-amps is shown in table 3.25

Op-Amp name	Price per unit for less than 10 units	Price per unit for more than 10 units
LM324	$1.40	$1.25
LM358	$1.25	$1.15
LM627	$4.95	Unknown
LM741	$2.00	$1.80

Table 3.25 Respective op-amp prices

7.3.1.7.10 The Chosen Operational Amplifier for Stage 1

There are two operational amplifiers in stage 1. These op-amps are shown as A1 and A2 in figure 3.31. The LM627 was chosen to implement stage 1 of the instrumentation amplifier. The LM627 is to be operated in the compensated mode. The decision to use LM627 in the compensated mode was made because;

- The specified design target for the accuracy of the temperature measurement performed by the K type thermocouple was 5°C. The LM627 in the compensated mode is the only implementation that leads to a maximum error due to voltage offset that is less than 5°C. Using LM627 in the compensated mode is the only way this design target could be reached.

7.3.1.7.11 The Chosen Operational Amplifier for Stage 2

The LM741 was chosen to implement stage 2 of the instrumentation amplifier. The LM741 was implemented in the uncompensated mode. The reasons for choosing the LM741 in the uncompensated mode were;

- The LM741 has a slightly smaller maximum offset that either the LM358 or the LM324. This will lead to marginally better accuracy in temperature measurement.
- Using the LM627 will achieve little relative improvement in the accuracy of the temperature measurements. The LM627 costs more than twice the price of the LM741. Such a cost for so little improvement can not be justified.
- The LM741 could be latter compensated if accuracy needed to be improved
- The LM741 is smaller than the LM358 and this aided fitting the entire instrumentation amplifier and low pass filter circuits onto the breadboard.

7.3.2 Low Pass Filter

There is likely to be a lower signal to noise ratio out of the instrumentation amplifier than out of the other interface circuits. If this noise is not eliminated then the HC11 will be likely to yield inaccurate readings for the K type thermocouple temperature sensor. A low pass filter is required to reduce the noise at the output of the Instrumentation amplifier.

7.3.2.1 Reasons for Low Signal to Noise Ratio

The instrumentation amplifier that will be used to boost the K type thermocouple voltage is a high gain amplifier. The gain of the amplifier will be 481 volts per volt. The reason the amplifier needs to be high gain is that the signal that is produced by a K type thermocouple is very small compared to the signals that the HC11 analog to digital converter was designed to read.

A K type thermocouple produces a much smaller signal than the other input devices used in this system. The K type thermocouple input source will be inside the same aircraft and therefore in close proximity to all the other input sources. Due to the nearness to other input sources the thermocouple will consequently be exposed to similar levels of noise. A smaller signal coupled with similar levels of noise results in a smaller signal to noise ratio at the output of the K type thermocouple when compared with the other input sources.

The instrumentation amplifier can not discriminate between the real signal and noise. The instrumentation amplifier will amplify both the signal and the noise by the same amount. As a result the signal to noise ratio at the output of the instrumentation amplifier can be expected to be equal to the signal to noise ratio at the input of the instrumentation amplifier, when noise generated within the amplifier are neglected. Since there will be a relatively low signal to noise ratio at the input of the instrumentation amplifier there will also be a relatively low signal to noise ratio at the output of the instrumentation amplifier.

7.3.2.2 Frequency Spectrum of the Cylinder Head Temperature Signal

The cylinder head temperature of an aircraft engine does not change rapidly. The most rapid changes in cylinder head temperature are likely to occur during the first few seconds after a cold start. The pilot is not interested in the cylinder head temperature during the first few seconds of operation. The pilot is generally interested in the cylinder head temperature because it indicates;

- When the engine has warmed enough to safely apply power
- When the engine is running too hot

During the times when the pilot is interested in cylinder head temperature the change in cylinder head temperature will not be rapid, a change of one degree per second would be considered very fast.

The pilot is only concerned with cylinder head temperature during times when the change in cylinder head temperature will be relatively slow. During the times the pilot is interested in the cylinder head temperature the cylinder head temperature signal will be apparent in the low end of the frequency spectrum.

7.3.2.3 Boosting the Signal-to-Noise Ratio

Generally the majority of a particular signal will exist in a section of the overall frequency spectrum. Noise is present throughout the whole frequency spectrum. A filter can be used to block frequencies where very little of the signal power exists and pass frequencies where the majority of signal power occurs. Since noise is generally present throughout the whole of the frequency spectrum cutting out large sections of the frequency spectrum significantly reduce the amount of noise. Very little of the signal power is lost because the frequencies blocked to not contain much of the signal power. The result is that the signal power is reduced slightly while the noise power is reduced greatly. These leads to a higher signal to noise ratio.

7.3.2.4 Active and passive filters

Filters can be active or passive. Passive filters consist of resistors and capacitors. Passive filters have two advantages. One advantage of passive filters is they do not need a power supply. The second advantage of passive filters is they require fewer components than active filters. Active filters consist of op-amps and/or transistors as well as resistors and capacitors. Active filters have some advantages over passive filters. Active filters;

- Can have adjustable gain. This means the signal need not be attenuated in the filtering process
- Are easier to adjust than passive filters
- Do not have loading problems since active filters have high input impedance and low output impedance.

7.3.2.4.1 Reasons for choosing an active filter

An active filter has been chosen to improve the signal to noise ratio of the K type thermocouple part of the system. The reasons for choosing an active filter to filter the output of the instrumentation amplifier are;

- The accuracy of the gain in the signal between the thermocouple output and the HC11 input is directly related to the accuracy of the temperature measurements taken by the system. The adjustable gain of an active filter makes achieving the correct gain between the K type thermocouple output and the HC11 input easier.
- The high input impedance and low output impedance of active filters will make an active filter easier to integrate into the overall system.

7.3.2.5 Types of filters

Filters are classified by the frequencies they pass and block. The Three most common types of filters are;

- Low pass filters
- High pass filters
- Band pass filters

Other Filter types include;

- Band reject filters
- All pass filters

7.3.2.5.1 Low pass filters

Low pass filters pass frequencies between DC and a certain cutoff frequency. The frequency and phase responses for an ideal low pass filter are shown below in Figure 3.32 and Figure 3.33 respectively

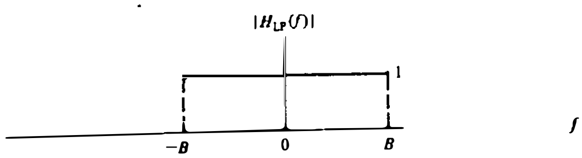


Figure 3.32 frequency response of an ideal low pass filter

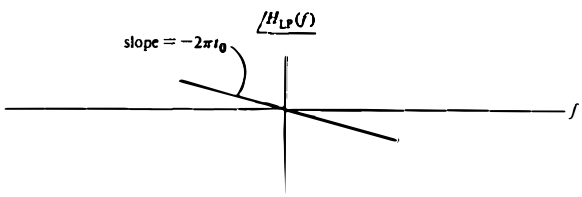


Figure 3.33 Phase response of an ideal low pass filter

7.3.2.5.2 High Pass Filters

High pass filters pass frequencies above a certain threshold frequency. The frequency and phase responses for an ideal high pass filter are shown in Figure 3.34 and 3.35

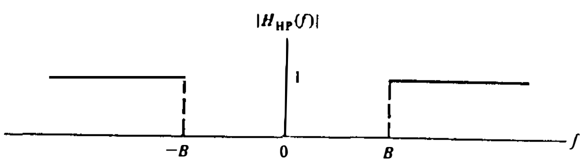


Figure 3.34 frequency response for an ideal high pass filter

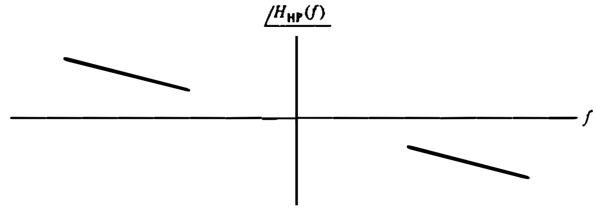


Figure 3.35 phase response for an ideal high pass filter

7.3.2.5.3 Band pass filters

Band pass filters pass frequencies between a low threshold frequeny and a high threshold frequency. A band pass filter can be generated by cascading a low pass and high pass filter. The frequency and phase responses for an ideal band pass filter are shown below in figure 3.36 and 3.37.

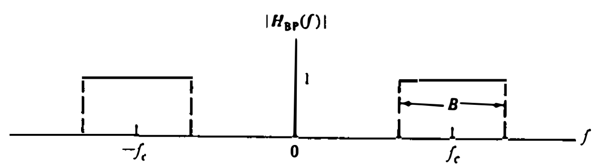


Figure 3.36 Frequency response of an ideal band pass filter

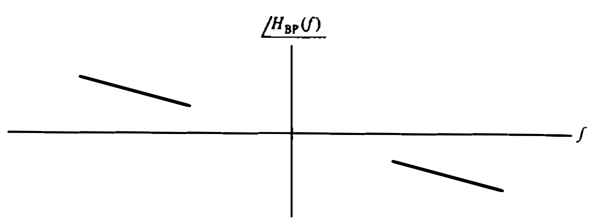


Figure 3.37 Phase response of an ideal band pass filter.

7.3.2.5.4 Choosing the Filter Type.

The K type thermocouple signal will be primarily based in the low end of the frequency spectrum. The most appropriate filter type for passing the thermocouple signal and blocking noise will be a low pass filter.

7.3.2.6 Types of Low Pass Filters

Three common types of low pass filters are the ;

- Bessel filter
- Butterworth low pass filter
- Chebyshev low pass filter

Butterworth filters are characterized by having a flat passband and stop band. Chebyshev filters have a ripple passband and a flat stopband. The Bessel filter attempts to maintain linear phase in the passband.. The Chebyshev filter attenuates the fastest with respect to frequency for any filter of nth order.

7.3.2.6.1 Choosing the Appropriate Low Pass Filter Type.

The type of low pass filter to use is dependant on the system requirements. Couch explains some common reasons for choosing a particular filter type:

The Chebyshev filter is used when a sharp attenuation characteristic is required when using a minimum number of circuit elements. The Bessel filter is often used in data transmision when the pulse shape is to be preserved since it attempts to maintain a liear phase response in the passband. The Butterworth filter is often used as a compromise between the Chebyshev and Bessel characteristics.

(1997, p. 246)

A Butterworth low pass filter is the most appropriate type for this particular application because it has a sharper attenuation characteristic than the Bessel filter while retaining a flat pass band not present in the Chebyshev filter.

7.3.2.6.2 Choosing the Order of the Low Pass Filter.

Higher order filters achieve greater attenuation rates with change in frequency. Higher order filters also require more components and are more complicated than lower order filters. As a consequence higher order filters are more expensive. The low pass filter used in this project will be a first order low pass filter. If testing reveals that a higher order filter is necessary then a higher order filter will then be used.

7.3.2.7 First Order Low Pass Butterworth Filter Design

A general first order low pass filter circuit is shown below in Figure 3.38

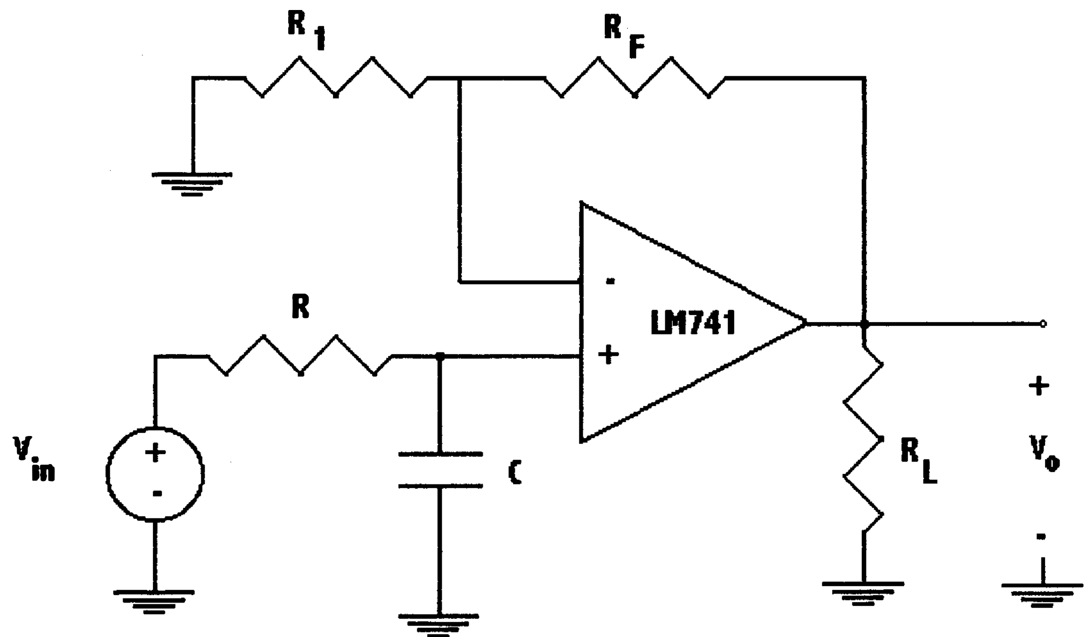


Figure 3.38 The basic first order low pass Butterworth filter

The transfer function for this low pass filter circuit is expressed below.

$$\frac{V_o}{V_{in}} = \frac{A_F}{1 + j\left(\frac{f}{f_H}\right)}$$

where $\frac{V_o}{V_{in}}$ = gain of the filter as a function of frequency

$A_F = 1 + \frac{R_F}{R1}$ = passband gain of the filter

f = frequency of the input signal

$f_H = \frac{1}{2\pi \ RC}$ = high cutoff frequency of the filter

The steps in designing a low pass filter are:

1. Choose a value of high cutoff frequency
2. Select a value of C less than or equal to 1μF. Mylar or tantalum capacitors give better performance.
3. Calculate the value of R using $R = \frac{1}{2\pi \ f_H C}$
4. Finally, select values of R1 and R_F dependant on the desired passband gain using $A_F = 1 + \frac{R_F}{R1}$

7.3.2.7.1 Choosing a Value of the High Cutoff Frequency

The frequency $f = f_H$ is called the cutoff frequency because the gain of the filter at this frequency is down by 3dB from 0 Hz. Since the HC11 will not be expected to update the stored values of its inputs more than 10 times a second a frequency of 10 Hz will be chosen for the cutoff frequency. The exact value of the cutoff frequency is not critical and a variation of ± 3Hz is acceptable

7.3.2.7.2 Selecting a Value for C

One of the common values for tantalum capacitors is 0.47µF. Capacitors of this type are available relatively cheaply, 70 cents each at Altronics (Altronics, 1997, p. 48). The low pass filter in this project will be implemented using a 0.47µF tantalum capacitor. This capacitor has a 10% tolerance.

7.3.2.7.3 Determining the Value of the resistor R

The value of R required to achieve, exactly, the desired cutoff frequency was calculated as shown below.

$$R = \frac{1}{2\pi f_H C}$$

$$R = \frac{1}{2\pi * 10 * 0.47 * 10^{-6}}$$

$$R = 33862.75\Omega$$

The value of the resistor R was chosen to be 30KΩ. Assuming all parts in the low pass filter circuit are ideal this leads to a cutoff frequency F_H of;

$$f_H = \frac{1}{2\pi RC}$$

$$f_H = \frac{1}{2\pi * 30000 * 0.47 * 10^{-6}}$$

$$f_H = 11.287Hz$$

This value of cutoff frequency is acceptable.

The reasons for choosing a 30KΩ were;

- 30KΩ is a common resistor value in both 5% and 1% tolerance resistors and therefore is available cheaply (Altronics, 1997, p. 53)
- 33KΩ resistors are only commonly available in 1% tolerance are consequently more expensive. A packet of 10 1% tolerance 0.25W resistors costs 70 cents but a packet of 10 5% tolerance 0.25W resistors costs 40 cents at retail prices. (Altronics, 1997, p. 53)

As has already been alluded to the resistor R is to be 5% tolerance resistor.

7.3.2.7.4 Deviation in the Cutoff Frequency due to Resistor and Capacitor Tolerances.

The maximum deviation in cutoff frequency due to resistor and capacitor tolerances was calculated as follows. The resistor R is a 5% tolerance resistor. This yields a;
Lowest value for R = 28500Ω
Highest value for R = 31500Ω

The capacitor C has a 10% tolerance. This yields a;
Lowest value for C = 0.4465μF
Highest value for C = 0.4935μF

The filter cutoff frequency is related to the resistor and capacitor values by the following equation

$$f_H = \frac{1}{2\pi RC}$$

The highest possible cutoff frequency occurs when the resistor and capacitor are at their lowest possible values. In this case the cutoff frequency becomes;

$$f_H = \frac{1}{2\pi * 28500 * 0.4465 * 10^{-6}}$$

$$f_H = 12.507\text{Hz}$$

The lowest possible cutoff frequency occurs when the resistor and capacitor are at their highest possible values. In this case the cutoff frequency becomes;

$$f_H = \frac{1}{2\pi * 31500 * 0.4935 * 10^{-6}}$$

$$f_H = 10.238\text{Hz}$$

These maximum and minimum values of cutoff frequency are still within the allowable range of 10 ± 3 Hz.

7.3.2.7.5 Choosing the values of R1 and R_F to achieve the desired gain

The passband gain of the low pass filter is determined by the equation

$$A_F = 1 + \frac{R_F}{R1}$$

The desired passband gain is 1. In order to achieve desired low pass filter gain of 1 the term $\frac{R_F}{R1}$ must be equal to 0. This can be achieved by either setting R_F to 0 or R1 to infinity. The final low pass filter design has both $R_F = 0$ and R1 = infinity.

7.3.2.8 Choosing the Low Pass Filter Operational Amplifier

Before going any further it is important to note that none of the operational amplifiers considered during the low pass filter development are aerospace specified. As a consequence it is important that these devices are not used in a real aircraft. There are similar devices that meet the required aircraft specifications. Despite the fact that the devices used in the breadboard simulation may not be used in a real aircraft system the method involved in choosing the appropriate op amp for the breadboard simulation still applies for real aircraft applications.

Five operational amplifiers were considered for the low pass filter. These operational amplifiers are;

- LM324
- LM301
- LM358
- LM627
- LM741

The decision to choose a particular operational amplifier for the low pass filter was based on;

- Availability
- Maximum Input offset voltage and possibility of offset compensation.
- Component size
- Power supply requirements
- Cost

7.3.2.8.1 Availability

The LM301 was found not to be easily available. As a result the LM301 was eliminated from further consideration

7.3.2.8.2 Input Offset Voltage and Offset Compensation

The input offset voltage of the op-amp used in the low pass filter has a direct effect on the accuracy of temperature measurements taken from the thermocouple. A 0.01953125 volt change in the voltage experienced at the HC11 analog to digital converter corresponds to a 1°C change in temperature. Since the low pass filter circuit in this application has a gain of one, the change in voltage at the HC11 A/D pin due to offset voltage of the op-amp is equal to the input offset voltage of the op-amp.

The maximum offset voltage for each of the four remaining op-amps is shown in Table 3.26.

Operational amplifier	Maximum offset voltage at 25°C
LM324	7mV
LM358	7mV
LM627	0.110mV
LM741	6mV

Table 3.26 maximum offset voltages at 25 °C.

The offset voltage of an operational amplifier generally changes with temperature. It is important to also consider the maximum offset voltage that can occur at the maximum and minimum environment temperatures. It is assumed that the low pass filter will not be operating in environmental temperatures below 0°C. The low pass filter is assumed to not be operating in area where the ambient temperature is greater than 50°C. There is a 25 C° difference between the maximum operating temperature and the temperature where the maximum offset voltage is specified in the datasheets. There is also a 25 C° difference between the minimum operating temperature and the temperature where the maximum offset voltage is specified in the datasheets. We need allow for a change of 25 C° to determine the maximum voltage offset that can occur during operation. The maximum voltage offset that can occur during operation was worked out as follows.

Maximum offset change per degree temperature change = V_1

Maximum increase in offset due to temperature change of 25° C = V_T

Maximum offset at 25° C = V_{25}

Maximum offset voltage = V_{off}

$V_{25} = 25 * V_1$

$V_{off} = V_r + V_r$

The results for each of the four op-amps is shown below in table 3.27

Op-amp	V_{25}	V_I	V_T	V_{off}
LM324	7mV	30μV	0.750mV	7.750mV
LM358	7mV	20μV	0.5mV	7.5mV
LM627	0.110mV	0.6μV	0.015mV	0.125mV
LM741	6mV	15μV	0.375mV	6.375mV

Table 3.27

Offset compensation is possible on the LM627 and the LM741. This allows us to effectively reduce the maximum offsets of these two op-amps to 0.015mV and 0.375mV respectively at the expense of adding additional components to the low pass filter circuit.

7.3.2.8.3 Component size

It is desirable to implement both the instrumentation amplifier and the low pass filter on the same breadboard. The breadboard has limited space and consequently smaller components are desirable. The LM627, LM358 and LM741 are available in 8 pin DIPs. The LM358 is are available in 14 pin DIP.

7.3.2.8.4 Power Supply Requirements

The LM324 and LM358 have an advantage of being able to operate with either a single or a double power supply. The LM741 and the LM627 both require two power supplies.

7.3.2.8.5 Cost

The cost analysis was done based on the Altronics catologue (1997, p.61). The LM627 is unavailable from Altronics and its price is based on the receipt received from Dick Smith electronics for the purchase of 2 such op-amps. The retail price per op-amp for purchases of less than 10 op-amps and more than ten op-amps is shown in Table 3.28.

Op-amp	Price per unit for less than 10 units	Price per unit for more than ten units
LM324	$1.40	$1.25
LM358	$1.25	$1.15
LM627	$4.95	Unknown
LM741	$2.00	$1.80

Table 3.28 Respective op-amp prices

7.3.2.8.6 The Chosen Operational Amplifier

The LM741 operational amplifier was chosen for the low pass filter circuit. The reasons for choosing the LM741 are;

- Much of the breadboard's space had already been devoted to the instrumentation amplifier so a smaller size package was of great benefit
- The LM741 has a slightly smaller offset than the LM358 and LM324. This will lead to marginally better accuracy in temperature measurement
- The LM627 although far higher performance than the LM741 costs more than twice the price of the LM741. The effect of using an LM627 instead of the LM741 would result in only a marginal improvement on system accuracy.
- The LM741 could be compensated if accuracy needed to be improved.

7.3.2.9 The low pass filter design

The low pass filter design is shown below in Figure 3.39. Component values for the circuit are listed in Table 3.29

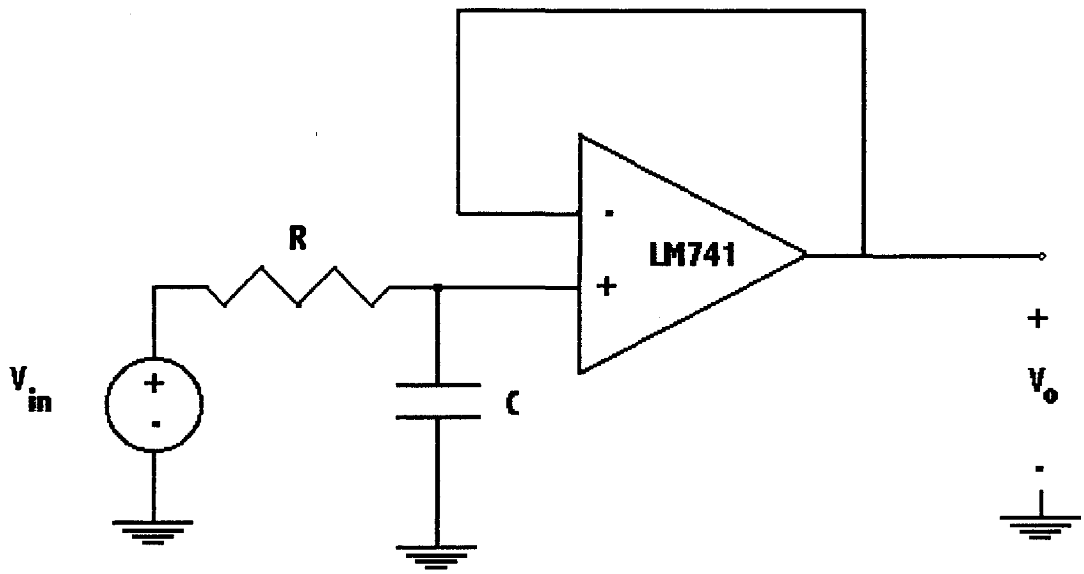


Figure 3.39 Final Filter design

Component Name	Value
R	30KΩ ± 5%
C	0.47μF ± 10%

Table 3.29 Low Pass Filter Component Values

7.4 K Type Thermocouple Procedure (KType)

The main task of the K type thermocouple procedure (KType)is to determine the temperature of the K type thermocouple. The K type thermocouple is connected the input of the instrumentation amplifier. The output of the instrumentation amplifier is connected to the input of the low pass filter circuit. The output of the low pass filter circuit is connected through a protection resistor to one of the HC11's A/D input pins. The K type thermocouple procedure calculates the K type thermocouple temperature based on the voltage at the A/D input pin and a compensating value.

7.4.1 Tasks

The KType procedure performs several tasks. These tasks are;

- Receives as a parameter the number of the A/D pin connected to the low pass filter output.
- Reads the hexadecimal representation of the voltage at the appropriate A/D input.
- Sets "warnings" based on the voltage present at the appropriate A/D input.
- Receives as a parameter a number that is used to perform junction compensation of the thermocouple.
- Determines thermocouple temperature based on the A/D pin voltage and the junction compensating value
- Stores the calculated thermocouple voltage in KTemp.

7.4.2 Required Data

In order to operate correctly the KType procedure requires access to the following values;

- CurrentVal
- Warning
- KTemp
- KTLow_Red
- KTYellow
- KTRed

The definitions of the required data for the procedure KType are explained below.

7.4.2.1 CurrentVal

The hexadecimal values for the voltages at each of the analog to digital converter pins is stored in 8 consecutive addresses. These 8 addresses are referred to as the Current Value page. The lowest address contains the value representing the voltage at Port E pin 0. The next address contains the value representing the voltage at Port E pin 1 and so forth. The constant CurrentVal is a pointer to the first address of the Current Value page. The KType procedure requires CurrentVal so it can read the value corresponding to the voltage at the A/D input pin connected to the low pass filter.

7.4.2.2 Warning

The constant Warning is a pointer to the first address of the Warnings page. The KType procedure requires this pointer so that it can update the warning corresponding to the K type thermocouple.

7.4.2.3 KTemp

The calculated K type thermocouple temperature is stored in a location pointed to by the pointer KTemp. The procedure needs this pointer so that it can store the calculated K type thermocouple temperature in the correct memory location.

7.4.2.4 KTLow_Red

This constant is the value that determines whether the KType procedure sets a Low Red warning. If the A/D pin voltage is equal to or below this value then a Low Red warning will be set

7.4.2.5 KTYellow

This constant is the value that determines whether the KType procedure sets a Yellow warning. If the A/D pin voltage is equal to or below this value then a Yellow warning is set.

7.4.2.6 KTRed

This constant is the value that determines whether the KType procedure sets a Red warning. If the A/D pin voltage is equal to or below this value then a Red warning is set.

7.4.3 Required Procedures

The KType procedure requires the Set_Master procedure in order to operate correctly. The starting address of the Set_Master must be available known to the Battery_Volts procedure.

7.4.4 Input Parameters

The input parameters for the KType procedure are outlined in Table 3.30

Parameter name	Where the parameter is stored during passing
KTPin	Accumulator A
KTJunctionT	Accumulator B

Table 3.30 Input parameters for the KType procedure

7.4.5 Local Variables

The local variables that are used in this procedure and their offsets are outlined in Table 3.31.

Local variable name	Size of the variable in bytes	Memory offset for the local variable	Address offsets covered by the variable
KTPin	1	01	01
KTJunctionT	1	00	00
KTKTemp	2	02	02,03

Table 3.31 Local variable information for the KType procedure

7.4.6 Values Changed

There are several values that may be changed at the conclusion of the KType procedure. These changes are explained under the headings KTemp, K Type Warning and Master_Warning.

7.4.7 Definition of Parameters, Variables and Values Changed

7.4.7.1 KTPin

Only one A/D pin is connected to the low pass filter. There are 7 A/D input pins. The KType procedure needs to know which A/D pin is connected to the low pass filter. The parameter and local variable KTPin specifies which A/D input pin is connected to the low pass filter circuit. Legal values for KTPin range from 0 to 7. KTPin is one byte in size.

7.4.7.2 KTJunctionT

In order to determine absolute temperatures thermocouples need a reference temperature. The KTJunction variable and input parameter specifies a reference temperature for the thermocouple. This reference temperature also is used to compensate for any junction thermocouple voltages.

7.4.7.3 KTKTemp

The actual thermocouple temperature is temporarily stored while the KType procedure is running. KTKTemp is where the thermocouple temperature is temporarily stored. Two bytes are required to temporarily store temperature. Since the thermocouple is assumed to be always measuring temperatures above 0°C the variable KTKTemp is unsigned.

7.4.7.4 KTemp

The pointer KTemp points to the location where the calculated thermocouple temperature is stored. At the conclusion of the KType procedure the calculated K type thermocouple temperature is updated. The calculated K type thermocouple temperature is stored in 2 consecutive bytes. The high byte of the K type thermocouple temperature is pointed to by the pointer KTemp.

7.4.7.5 K Type Warning

The K type thermocouple is associated with one of the A/D pins. At the conclusion of the KType procedure the warning corresponding to this A/D pin is updated.

7.4.7.6 Master_Warning

I the warning corresponding to the K type thermocouple low pass filter A/D pin is set to the Low Red or the Red condition then the Master_Warning will also be set on.

7.4.8 How the Thermocouple Temperature is Calculated

The interface circuitry between the K type thermocouple and the HC11 A/D input pin is designed to cause a 0.0193125 Volt change per C° change. A 0.0193125 volts is also the size of the quantization levels of the HC11 A/D converter when it is set up in the project configuration. This matching greatly simplifies the calculation of the K type thermocouple voltage. The matching simplifies calculation because it leads to a 1:1 relationship between degrees Celsius and the hexadecimal value corresponding to the HC11 A/D input pin voltage.

The K type thermocouple temperature is calculated according to the following equation;

$T_K = V_H + T_J$

Where;

V_H = Hexadecimal value corresponding to HC11 pin voltage

T_j Junction compensation value

T_K Calculated thermocouple temperature

7.4.9 Algorithm

The algorithm for the procedure KType is shown in appendix E

7.4.10 Code

The code for the procedure KType is shown in the in appendix F.

7.5 System Testing

Testing of the cylinder head temperature system was conducted by simulating a thermocouple voltage at the input of the Cylinder Head Temperature Interface. The testing circuit is shown in Figure 3.40.

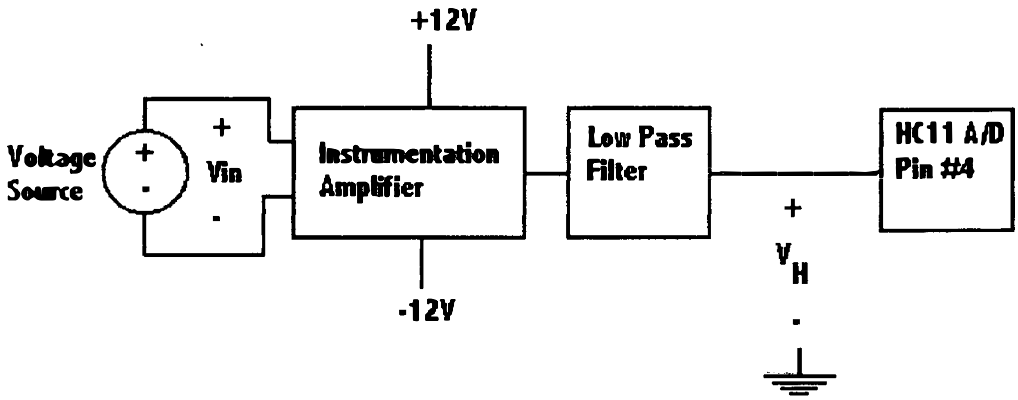


Figure 3.40 Cylinder Head Temperature system testing circuit

The Voltage source shown in Figure 3.40 was implemented using a variable voltage laboratory power supply and a voltage divider circuit. This implementation allowed the Voltage Vin to be varied from 0 to 10.5mV.

The testing of the Cylinder Head Temperature System was conducted according to the following method.

1. The software was setup to run operate the KTemp procedure for input at A/D pin number 4. This was required since the Cylinder Head Interface circuitry was attached to A/D pin 4.
2. The reference temperature parameter for the KTemp procedure was set to a constant value of 20°C.
3. The software was setup to run the Drive_Led procedure for input at A/D pin number 4. This was required so that the warning generating part of the KTemp procedure could be tested
4. The test circuit was conducted as shown in Figure 3.40.
5. The Warning Display Circuit was connected to the HC11 in the normal configuration. This was done so that the warning corresponding to the simulated cylinder head temperature could be tested.
6. The voltage source was set to 0V and then switched on
7. A digital multimeter was used to measure the voltages V_H and Vin as shown in Figure 3.40
8. The software was started
9. The status of the warning display circuit LEDs was noted. This information was used to determine if the KTemp procedure sets warnings correctly

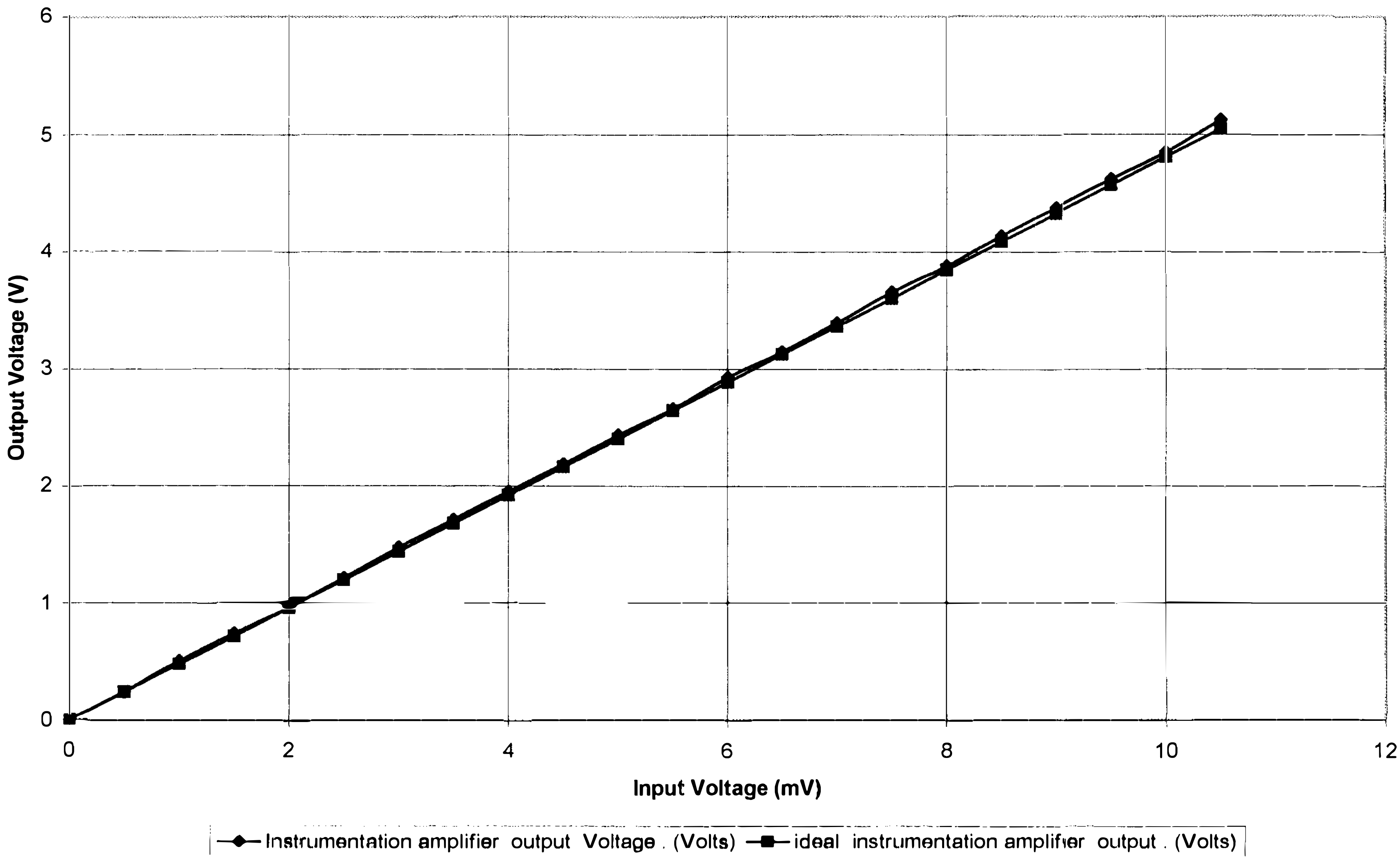
10. A HC11 system RESET was issued and the BUFFALO monitor program was used to determine the hexadecimal value corresponding to the voltage at the A/D input pin. This value was useful to determine how accurately the A/D Conversion System was determining the voltage at the A/D input pin
11. The BUFFALO monitor program was used to determine the hexadecimal value of the variable KTemp. KTemp contains the calculated temperature of the K Type Thermocouple. This value was used to calculate how accurately the cylinder head temperature was calculated based on the voltage present at the A/D pin.
12. The supply voltage was increased by 0.5mV. The procedure steps 7-12 were then repeated several times. The process stopped when the Voltage Vin was 10.5mV. This voltage is outside the system operating range so there was no point in testing further.

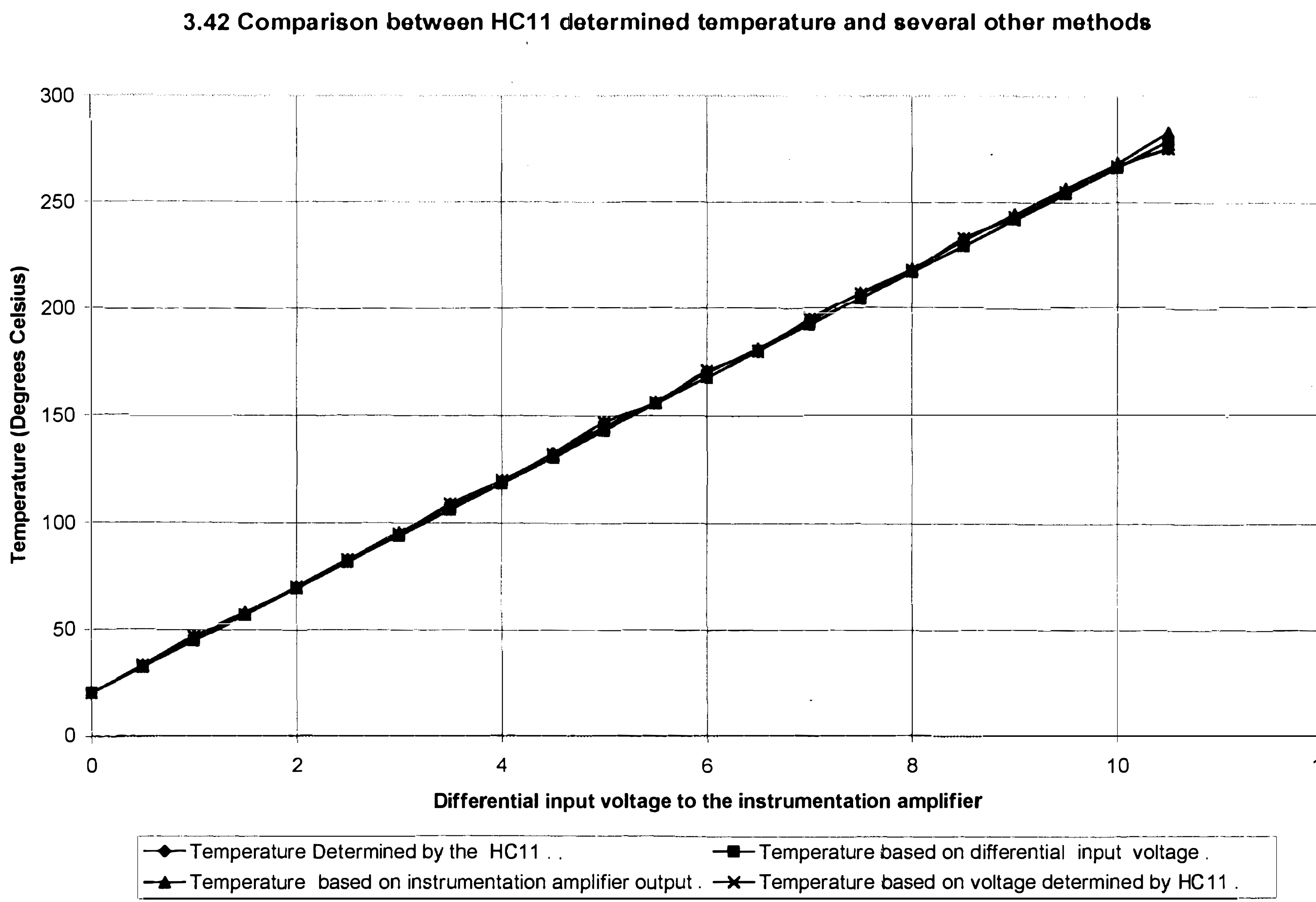
The numerical results of the above testing procedure are shown in Appendix G.

The performance of the Cylinder Head Temperature Interface is compared to its ideal performance in Figure 3.41. In nearly all cases the real interface voltage was slightly higher than the ideal. This suggests that the gain of the prototype Cylinder Head Temperature Interface is slightly higher than the desired gain of 481. The maximum error between real and ideal instrumentation amplifier output voltage was 0.0525V. This corresponds to a 3 degree error in the measured temperature.

The performance of the Cylinder Head Temperature System is shown in Figure 3.42. The maximum error between the temperature determined by the system and the temperature based on the input voltage Vin was 3.8 °C.

Figure 3.41 Cylinder Head Temperature Interface




3.42 Comparison between HC11 determined temperature and several other methods
Temperature (Degrees Celsius)
300
250
200
150
100
50
0
0
2
4
6
8
10
12
Differential input voltage to the instrumentation amplifier
Temperature Determined by the HC11 . .
Temperature based on differential input voltage .
Temperature based on instrumentation amplifier output .
Temperature based on voltage determined by HC11 .

7.5.1 System Warnings

The trigger values for the KTemp procedure were set to the values shown in Table 3.32

Warning Constant Name	Warning Constant trigger value
KTLow_Red	$1E
KTYellow	$E1
KTRed	$10E

Table 3.32 Warning triggers

The warning value set by the KTemp procedure based on the calculated Cylinder Head Temperature. The orange LED, corresponding to a Low_Red warning, was only lit when the calculated temperature was below below $001E. The green LED only became lit for calculated temperatures between $001F and $00E0. The yellow LED was lit for calculated temperatures between $00E1 and $10D. The red LED was lit for calculated temperatures above or equal to $010E. These results coincided with expectations.

7.5.2 Significance of the Testing Results

The following conclusions are drawn from the testing results;

- The warning value for the cylinder head temperature is being set correctly by the KTemp procedure
- When operating within the specified operating range the Cylinder Head Temperature system can be expected to give results accurate to the nearest 4°C.
- The gain of the Cylinder Head Temperature Interface appears to be higher than the required value of 481. The accuracy of the Cylinder Head Temperature System could be improved by modifying the circuit to bring the gain of the Cylinder Heat Temperature Interface closer to 481.

CHAPTER 8: Ambient Temperature System

The Ambient Temperature System's primary purpose is to determine ambient temperature and store this temperature in memory. The system also performs the secondary task of setting the warning corresponding to ambient temperature. The Ambient Temperature System relies upon the Operating System and the Analog-to-Digital Conversion system. The Ambient Temperature System is vital to the function of the Warning System. The relationship between Ambient Temperature System and the other subsystems is illustrated in Figure 3.1.

To achieve its objectives the Ambient Temperature System implements the dataflow shown in Figure 3.43.

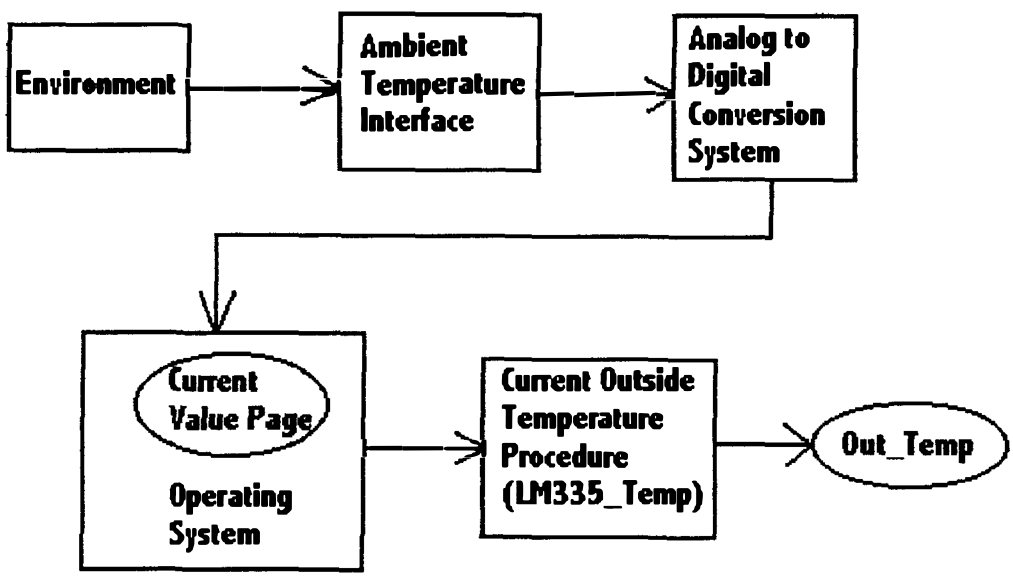


Figure 3.43 Ambient Temperature System

This section describes;

- Ambient Temperature System Specifications
- The LM335 temperature sensor
- The Ambient Temperature Interface
- The Ambient Temperature Procedure (LM335_Temp)
- The Testing of the Ambient Temperature System

8.1 System Specifications

Table 3.33 summarises the specifications for the Ambient Temperature System.

Temperature	
System Operating Range	-40 to 100°C
Operating Range of LM335 temperature sensor as stated in spec sheet	- 40 to 100°C
Operating Range of the LMM_335 Procedure due to procedure implementation	-40 to 127°C
Accuracy	± 3 °C

Table 3.33 Ambient Temperature Specifications

8.2 The LM335 Temperature Sensor

The LM335 temperature sensor is a semiconductor temperature sensor that is designed to operate as a 2-terminal zener diode. The reverse breakdown voltage is directly proportional to temperature at + 10mV/°K. This linear relationship between sensor output voltage and temperature is a very useful and unusual feature when compared with other temperature sensors. The linear relationship makes using the LM335 to determine temperature a relatively easy process. It is for this reason that the LM335 temperature sensor was used in the implementation of the Ambient Temperature System.

8.2.1 Features

The features of the LM335 are described in the LM335 spec sheet, included in Appendix A. Features of the LM335 of special interest are;

- Operating Temperature range
- Operating Current Range
- Accuracy
- Self Heating Effects
- Cost

8.2.2 Operating Temperature Range

The LM335 operates in the temperature range from -40°C to 100°C. This range is suitable for measuring ambient temperature. The operating temperature ranges for the sensors in the LM135 series are shown below in table 3.34. The LM335 sensor was chosen over the others in the series because the additional range of the other sensors was not needed.

Part number name	Minimum temperature °C	Maximum temperature °C
LM135	-55	150
LM235	-40	125
LM335	-40	100

Table 3.34 Operating temperature ranges for the LM135 series

8.2.3 Current Ranges

The LM335 is specified to operate correctly with reverse currents in the range of 400μA to 5mA. The operating current must be kept in this range for all possible system conditions when reliable temperature measurement is required. The LM335 can be damaged if exposed to too much forward or reverse current. To avoid damage the maximum reverse current that the LM335 may experience is 15mA. To avoid damage the forward current through the device must not exceed 10mA.

8.2.4 Accuracy

The accuracy of the LM335 temperature sensor determines the accuracy of the temperature measurements taken. The LM335 accuracy can be improved by calibrating the sensor. Error in °C for the temperature sensor at 25°C and at the extremes of its operating ranges is shown below in Table 3.35. Table 3.35 indicates accuracy for both calibrated and uncalibrated modes of operation.

Operating temperature range	Uncalibrated maximum error (°C)	Calibrated maximum error (°C)
$T_C = 25°C$	6	0
$T_{min} \leq T_C \leq T_{max}$	9	2

Table 3.35 Accuracy of the LM335 temperature sensor

8.2.5 Cost

The LM335 temperature sensor is available relatively cheaply. According to Altronics (1997, p. 61) the LM335 temperature sensor can be purchased at a retail price of $3.20 each. For purchases of 10 units or more the price drops to $3.00 per unit. The low cost per unit coupled with the low amount of support circuitry required to operate it make the LM335 the best sensor for measuring ambient temperature.

8.3 Ambient Temperature Interface

The primary objectives of the Ambient Temperature Interface is to convert temperature into an electrical signal that can be read by the A/D conversion system. The secondary objective of the Ambient Temperature Interface is to prevent damage to the A/D converter. The interface is designed to be low cost while still fulfilling its objectives. The LM335 interface circuit is shown below in Figure 3.44

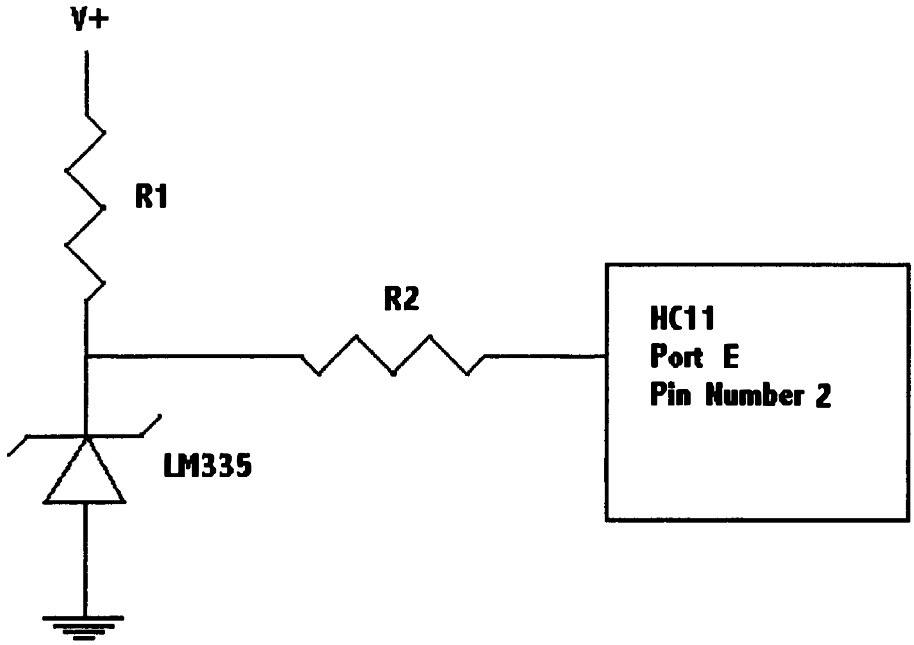


Figure 3.44 The LM335 interface circuit

8.3.1 Operating Temperature Range

The LM335 interface is designed to operate properly over the same range of temperatures as the LM335 temperature sensor. The interface operating temperature range is –40°C to 100°C.

8.3.2 Voltage supply

The voltage supplied by a 12 volt battery is not constant. The reason for the variation is that the voltage of a Lead Acid battery is related to its level of charge. Voltage increases as the battery charges and decreases as it discharges. The LM335 interface must operate properly for all likely battery voltages. Maintenance free lead acid battery voltage can reach 16 volts while charging (Lead Acid Battery, n.d., p.6). The LM335 system will be designed to operate correctly at a maximum supply voltage of 16 volts. According to (Lead Acid Battery, n.d., p. 4) a battery is considered discharged when its open circuit voltage is 11.7 volts, after a short application of a medium load. Under a load a lead acid battery voltage can drop below 11.7 volts. The LM335 system will be designed to operate correctly at minimum supply voltage of 7 Volts.

8.3.3 Compensating pin

The compensating pin of the device has been left floating, sacrificing accuracy to reduce cost. Not having a compensating circuit reduces the number of components required and the complexity of the circuit thereby reducing circuit cost.

8.3.4 The series resistor R1.

A series resistor is required between the LM335 and the voltage supply to limit the operating current. It is assumed that the LM335 system will be part of a 12 volt system. The LM335 temperature sensor has an operating current range of 400 μA to 5 mA. The value of the series resistor will be chosen to restrict the LM335 current to the current range specified for normal voltage supply conditions.

8.3.4.1 Determining the minimum value of R1

Resistor R1 must restrict LM335 current to a maximum of 5 mA during maximum current case conditions. Maximum current case conditions are assumed to be as follows;

- Supply voltage V+ = 16 volts
- LM335 voltage drop = 0 volts.

An a equivalent circuit for this situation is shown in Figure 3.45.

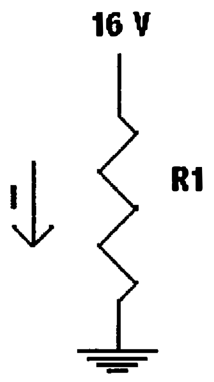


Figure 3.45 Maximum current case for the LM335 temperature sensor.

Using Ohms law we can determine the minimum value of R1.

$$R1_{min} = \frac{V}{I}$$

Which yields a minimum value of R1 being 3200 Ω.

8.3.4.2 Determining the maximum value of R1

The resistor R1 must be sufficiently small enough to prevent the LM335 current falling below its minimum value of 400μA during minimum current conditions. Minimum current conditions are defined as follows;

- Supply Voltage V+ = 7 Volts
- Operating temperature = 100 °C
- Resulting LM335 voltage due to temperature V_{LM} = 3.73V

This situation is depicted in the equivilent circuit shown in Figure 3.46.

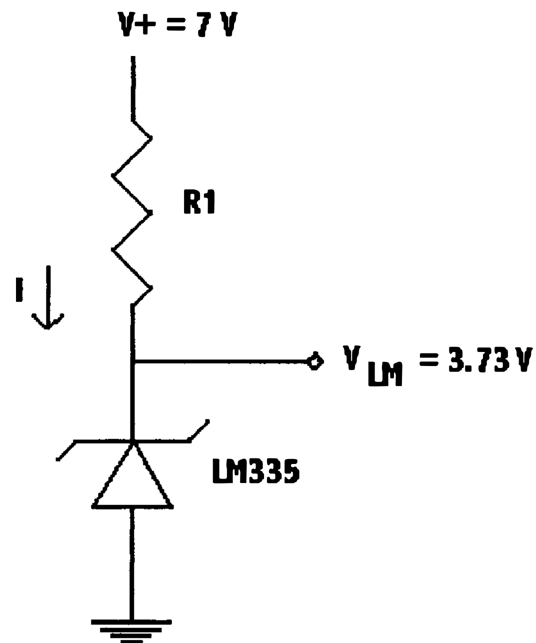


Figure 3.46 Minimum current case for the LM335 temperature sensor

Using Ohms law we can determine the maximum permissible value for the limiting resitor R1.

$$R1_{max} = \frac{V_{+} - V_{LM}}{I}$$

The resulting maximum value of R1 is 8175 Ω.

8.3.4.3 The chosen value of R1

The resistor R1 has a specified resistance of 6.8 KΩ. This value was chosen for the following reasons;

- The value is safely between the maximum and minimum values even when high tolerance resistors, 10% tolerance, are used

- 6.8KΩ is a common resistor value that is available cheaply in high tolerance resistors. Carbon film resistors with 5% tolerance and a power rating of 0.25W can be purchased'for 40c each. (Altronics, 1997, p. 61)
- By choosing a resistor value closer to the maximum allowed value we reduce system current. This reduces system power consumption and self heating effects in the LM335.

8.4 The Ambient Temperature Procedure (LM335_Temp)

The main task of the Ambient Temperature Procedure (LM335_Temp) is to determine the temperature of the LM335 temperature sensor. The procedure determines the temperature of the LM335 based on the voltage present at this A/D input pin.

8.4.1 Tasks

The LM335_Temp procedure performs several tasks. These tasks are;

- Receive as a parameter the number of the A/D pin connected to the LM335 interface circuitry
- Reads the hexadecimal representation of the voltage at the appropriate A/D input.
- Set ambient temperature warning based on the voltage present at the appropriate A/D input.
- Calculates the LM335 temperature based on the voltage at the A/D input
- Store the calculated LM335 temperature in Out_Temp

8.4.2 Required data

In order to operate the LM335_Temp procedure requires access to the values;

- CurrentVal
- Warning
- Out_Temp
- OTLow_Red
- OTYellow
- OTRed

The definitions of the required data for the procedure LM335_Temp are explained below.

8.4.2.1 CurrentVal

The hexadecimal values for the voltages at each of the analog to digital converter pins is stored in 8 consecutive addresses. These 8 addresses are referred to as the Current Value page. The lowest address contains the value representing the voltage at Port E pin 0. The next address contains the value representing the voltage at Port E pin 1 and so forth. The constant CurrentVal is a pointer to the first address of the Current Value page. The

LM335_Temp procedure requires CurrentVal so it can read the value corresponding to the voltage at the A/D input pin connected to the LM335 interface.

8.4.2.2 Warning
The constant Warning is a pointer to the first address of the warnings page. The LM335_Temp procedure requires this pointer so that it can update the warning corresponding to the LM335 temperature.

8.4.2.3 Out_Temp
Out_Temp is a pointer the location where the calculated LM335 temperature is stored. The procedure needs this pointer so that it can store the calculated LM335 temperature in the correct address.

8.4.2.4 OTLow_Red
The constant OTLow_Red is the value that determines whether a Low Red warning is set. If the A/D pin voltage is equal to or below this value the a Low Red warning is set.

8.4.2.5 OTYellow
The constant OTYellow is the value that determines whether a Yellow warning is set. If the A/D pin voltage is equal to or above this value then a Yellow warning is set.

8.4.2.6 OTRed
The constant OTRed is the value that determines whether a Red warning is set. If the A/D pin voltage is equal to or above this value then a Red Warning is set.

8.4.3 Required Procedures
The LM335_Temp procedure requires Set_Master procedure. The starting address of the Set_Master procedure must be known to LM335_Temp so that the LM335_Temp may call the Set_Master procedure.

8.4.4 Input Parameters
There is one input parameter for the LM335_Temp procedure. This input parameter is shown in Table 3.36.

Parameter Name	Where the parameter is stored during passing
OTPin	Accumulator A

Table 3.36 Input parameter for the LM335_Temp procedure.

8.4.5 Local variables

The LM335_Temp procedure has one local variable. This variable is shown in Table 3.37.

Local variable name	Size of the variable in bytes	Memory offset for the local variable	Address offsets covered by the variable
OTPin	1	$00	$00

Table 3.37 Local variables for LM335_Temp

8.4.6 Values that may be changed at the conclusion of the LM335_Temp procedure

There are three values that may be changed by the time the LM335_Temp procedure terminates. These values are explained under the headings Out_Temp, LM335 Warning and Master_Warning

8.4.7 Definitions of Parameters, Variables, and Values Changed

8.4.7.1 OTPin

The LM335 interface output is connected to one of the HC11's A/D pins. The LM335_Temp procedure needs to know which A/D pin is connected to the LM335 interface. The parameter and local variable OTPin specifies which A/D pin is connected to the LM335 interface circuit. Legal values for OTPin are in the range 0 to 7. OTPin is a one byte variable.

8.4.7.2 Out_Temp

The pointer Out_Temp points the location where the calculated LM335 temperature is stored. At the conclusion of the LM335_Temp procedure the calculated LM335 temperature is updated in the HC11 memory. The calculated LM335 temperature is stored in a single byte. The LM335 temperature is a signed number and therefore can be negative.

8.4.7.3 LM335 Warning

The LM335 interface is connected to one of the HC11 A/D pins. At the conclusion of the LM335_Temp procedure the warning corresponding to this A/D pin will be updated.

8.4.7.4 Master_Warning

If the warning corresponding to the LM335 interface A/D pin is set then the Master_Warning will be set on.

8.4.8 How the LM335 temperature is calculated.

The basic method for calculating LM335 temperature based on the input voltage to the HC11 A/D pin is expressed in the equations below

The Voltage at the A/D pin is related to the temperature of the LM335 in Kelvin. This relationship is expressed as follows;

$$V = \frac{T_K}{100}$$

where

V = The voltage at the A/D input pin

T_K = The temperature of the LM335 temperature sensor in degrees Kelvin

The temperature in Kelvin is related to the temperature in Celsius by;

$$T_K = T_C + 273$$

where

T_C = temperature in Celsius

Combining these equations and solving for voltage we get;

$$V = \frac{T_C + 273}{100}$$

The hexadecimal value representing the voltage at the A/D input pin is related to the voltage at the A/D pin by;

$$H = \frac{256}{5} V$$

where

H = The HC11 hexadecimal number corresponding to the voltage at the A/D pin

We can combine these last two equations and solve them for an approximation of the temperature of the LM335 in degrees Celsius.

$$T_C^{\bullet} = \frac{500}{256}(H - 140)$$

where

$T_C^{\bullet}$ = approximated temperature in degrees Celsius

This equation could, in theory, be implemented into HC11 code to solve for temperature. In practice the HC11 division instruction makes this solution difficult to implement.

8.4.8.1 The HC11 integer division instruction problem

The HC11 integer division instruction "Performs an unsigned integer divide of the 16-bit numerator in D accumulator by the 16-bit denominator in index register X and sets the condition codes accordingly (HC11 M68HC11 Reference Manual, 1991, p. A-56). The LM335 temperature sensor part of the system is specified to work with negative temperatures down to -40°C. The possibility of negative temperatures poses a problem when using an unsigned divide instruction. The method for calculating the LM335 temperature must overcome the unsigned divide of a negative number problem.

8.4.8.2 The solution to the unsigned division problem

The solution to the unsigned division problem is outlined in this paragraph. The lowest specified operating temperature for the LM335 temperature sensor part of the system is -40°C. Instead of subtracting 140 from the HC11 hexadecimal representation of the voltage at the A/D pin a smaller value will be subtracted. This smaller value will cause a temperature of -40 °C to read as 0 after the subtraction. In order make the calculated temperature correct compensation must be done to cancel the effect of the smaller number being subtracted. The compensation involves subtracting 40 from the result after the division. Since the compensation occurs after the division a negative result is no longer a problem.

The smaller number to subtract instead of 140 was worked out as follows.
The requirement is

$$H = 0, \text{when } T_C = -40$$

so

$$H = 0 = 256\frac{(T_C + 273)}{500} + x$$

$$x = 119.296$$

or in hex

$$x = \$77$$

where x = the number to subtract

The method implemented in the HC11 code for calculating the temperature of the LM335 temperature sensor is expressed as an equation below.

$$T_C^{*} = \frac{500}{256}(H - 119) - 40$$

The method implemented in the HC11 code is mathematically equivalent to the basic method determined earlier except for the approximation of the number 119.296 with 119. This approximation leads to an error that increases the calculated temperature by 0.578125 above the real temperature.

8.4.8.3 Allowed temperature range due to the code implementation

One byte is used to store the calculated LM335 temperature. Since the temperature may be negative the byte must represent a signed integer. Since the byte represents a signed integer the maximum temperature allowed becomes 127°C. The minimum allowed temperature is a result of the unsigned nature of the IDIV instruction. The minimum temperature the LM335_Temp procedure is designed to handle is -40°C.

8.4.8.4 Approximation of the coefficient 500/256

Mathematically multiplying a number by 500/256 is equivalent to dividing by 0.512. The code implements the multiplication by 500/256 by dividing by a number approximately equal to 0.512. The actual number that is divided by is 0.51171875. As a hexadecimal number the number 0.51171875 is $00.83. A more precise approximation of the number 0.512 could not be used due to the 16 bit implementation of the HC11 IDIV instruction.

Approximating the coefficient 500/256 leads to a small error in the calculated temperature. The error is worst at higher temperatures. At 127°C, the maximum temperature allowed by the procedure, the error induced is 0.0696. The maximum temperature the LM335 part of the system is designed to operate at a maximum temperature of 100°C. At 100°C the error induced by the coefficient approximation is 0.0584°C.

8.4.9 Worst case error induced by the procedure

The LM335 temperature sensor part of the system is specified to operate at a maximum temperature of 100°C. The maximum error induced by the LM335_Temp procedure occurs at this temperature. The compromises in code implementation of the temperature calculation lead to a maximum error of 0.6477.

8.4.10 Algorithm

The Algorithm for the LM335_Temp procedure is shown in Appendix E.

8.4.11 Code

The code for the LM335_Temp procedure is shown in Appendix F.

8.5 Testing and Performance

The LM335_Temp procedure and the analog to digital converter part of the temperature sensor system were tested across the voltage range 0 to 4.43V. The testing methods and results are explained in the following sections.

Testing was performed by applying a voltage to the A/D converter input pin 2 to simulate the LM335 temperature sensor interface output voltage. The testing circuit is shown below in figure 3.47.

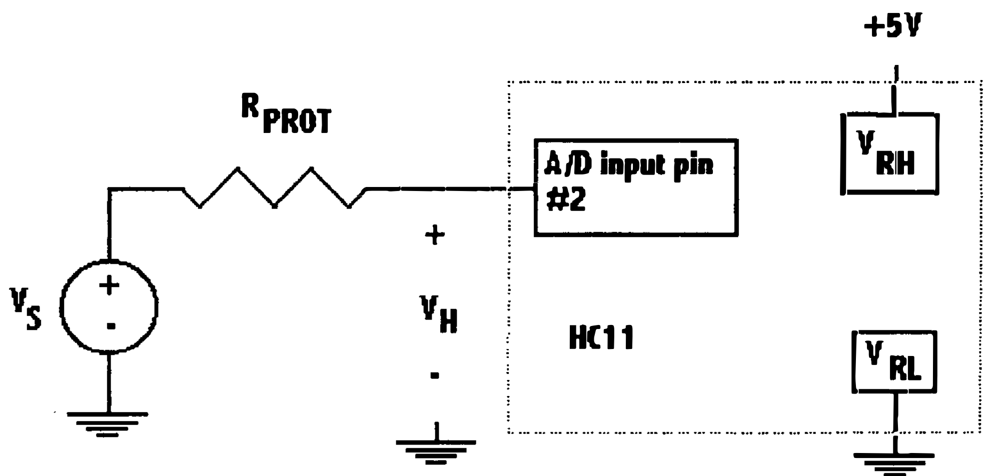


Figure 3.47 Testing circuit for LM335_Temp procedure and A/D converter

The test circuit component V_S was a laboratory power supply. This supply was capable of providing a variable DC voltage between 0 and 5 volts.

The resistor RPROT was a 1KΩ resistor used to protect the HC11 A/D converter from damage by excessive currents.

The testing of the A/D converter and the LM335_Temp procedure was conducted according to the following method.

1. The software was setup to operate the LM335_Temp procedure for input at A/D pin number 2. This was required since the circuit simulating LM335 voltage was connected to A/D pin 2.
2. The software was setup to run the Drive_Led procedure for the input at A/D pin number 2. This was required so that the part of the LM335_Temp procedure that sets warnings could be tested.
3. The test circuit was constructed as shown in Figure 3.47
4. The Warning display circuit was connected to the HC11 in the normal configuration. This was done so that the warnings generated by the LM335_Temp procedure could be displayed

5. The voltage of the laboratory supply was set to zero and the supply was then switched on.
6. A digital multimeter was used to measure the voltage V_H as shown in Figure 3.47.
7. The HC11 system software was started
8. The status of the warning display circuit LEDs was noted. This information was used to determine if the part of the LM335_Temp procedure that sets warnings was working correctly.
9. A HC11 system reset was issued and the BUFFALO monitor program was used to determine the hexadecimal value corresponding to the voltage at the A/D input in. This value is useful to determine how accurately the A/D converter and SCANINPUTPIN procedure were determining the voltage at the A/D input pin.
10. The BUFFALO monitor program was used to determine the hexadecimal value of the variable Out_Temp. The variable Out_Temp contains the calculated temperature of the LM335 temperature sensor. This value was used to determine how accurately the LM335_Temp procedure was calculating LM335 temperature based on the voltage present at the A/D input pin.
11. The voltage of the laboratory power supply was increased by a small amount. The procedure steps 6 – 11 were then repeated several times. The process stopped when the voltage VH was 4.43 V. A voltage of 4.43 volts corresponds to a temperature higher than the LM335_Temp procedure is specified to operate. There was little value in testing the system further outside of its operating range.

The numerical results of the above testing procedure are shown in Appendix G. These results are depicted graphically in figures 3.48 through to figure 3.50

Oerall the analog to digital converter performed as expected. Figure 3.48 and figure 3.49 compare the voltage determined by the analog converter during testing with the actual voltage at the A/D converter input pin as measured by the multimeter. The A/D converter measured voltage so closely matched the actual voltage at the A/D pin that is difficult to discern the two different lines in Figure 3.48. Figure 3.49 is an enlarged version of Figure 3.48 covering a section of the data including where the maximum error between A/D converter measured voltage and real voltage occurs. This maximum error was 0.022V. The A/D converter setup has a quantization resolution of 0.01953125V. The maximum error during this test therefore corresponds to a 1 bit error in the A/D converter measured voltage. The maximum percentage error occurring in the A/D converter measured voltage was 0.66%.

The LM335_Temp procedure operated correctly within the specified system operating range of -40°C to 100°C. Figure 3.50 depicts the temperature determined by the LM335_Temp procedure based on the voltage determined by the A/D converter. The temperature corresponding to the A/D converter voltage based on the ideal LM335 voltage temperature relationship is also shown.

It can be seen in figure 3.50 that the LM335_Temp procedure determines voltage very close to the ideal case within the specified system operating range. Within the specified

Figure 3.48 Analog to digital converter accuracy

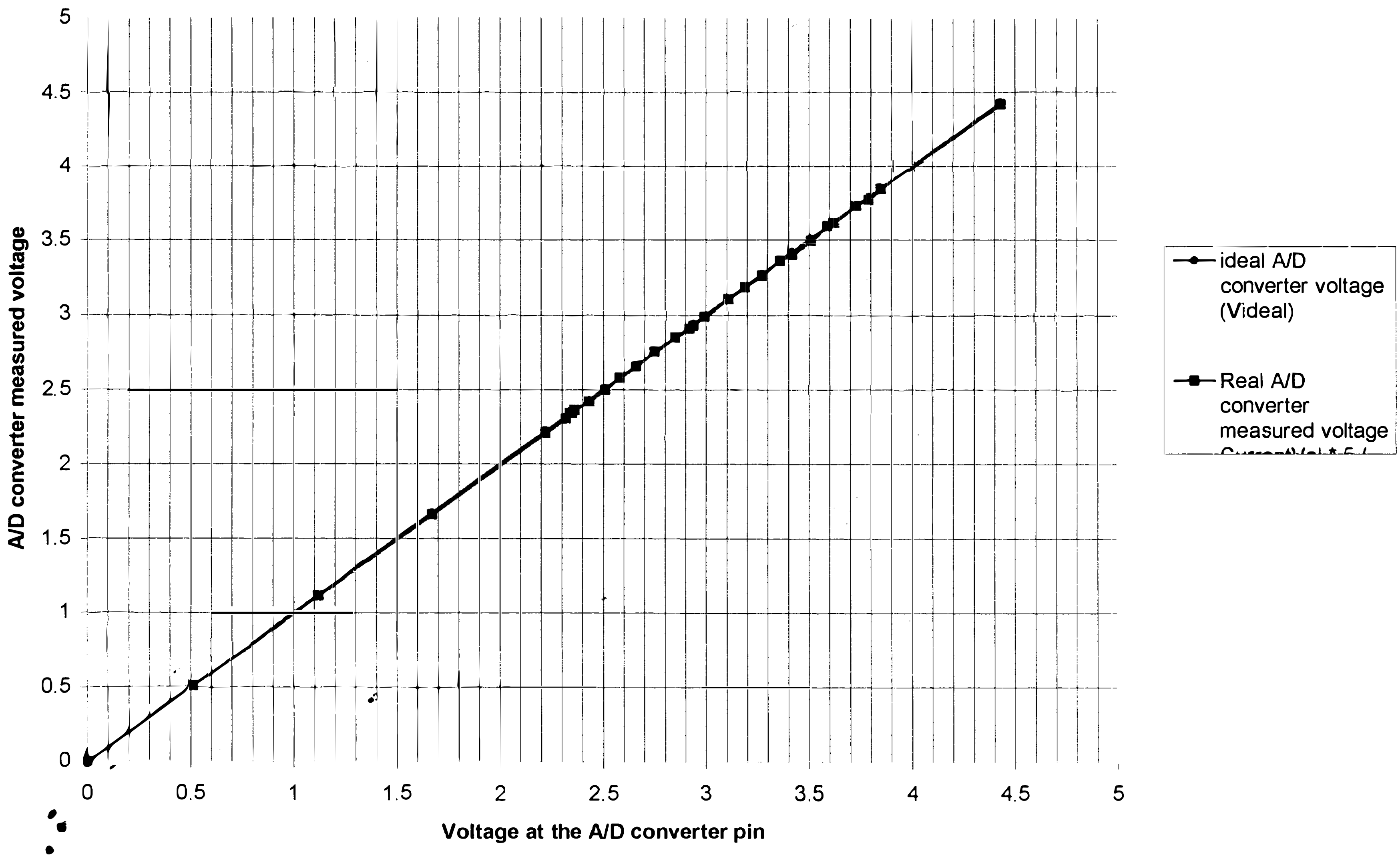


figure 3.49 A/D converter maximum error during testing

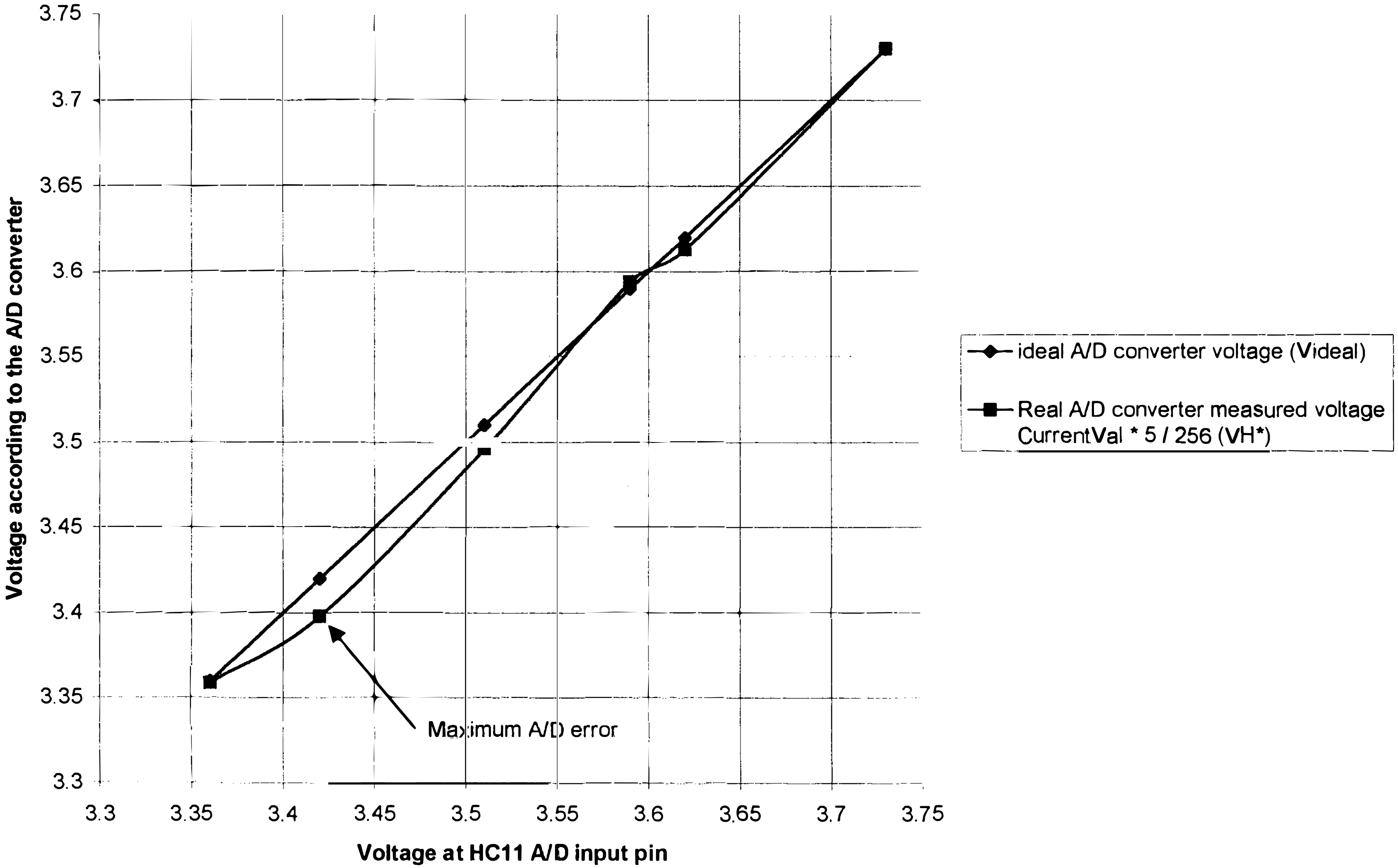


Figure 3.50 LM335_Temp procedure accuracy in the specified procedure range

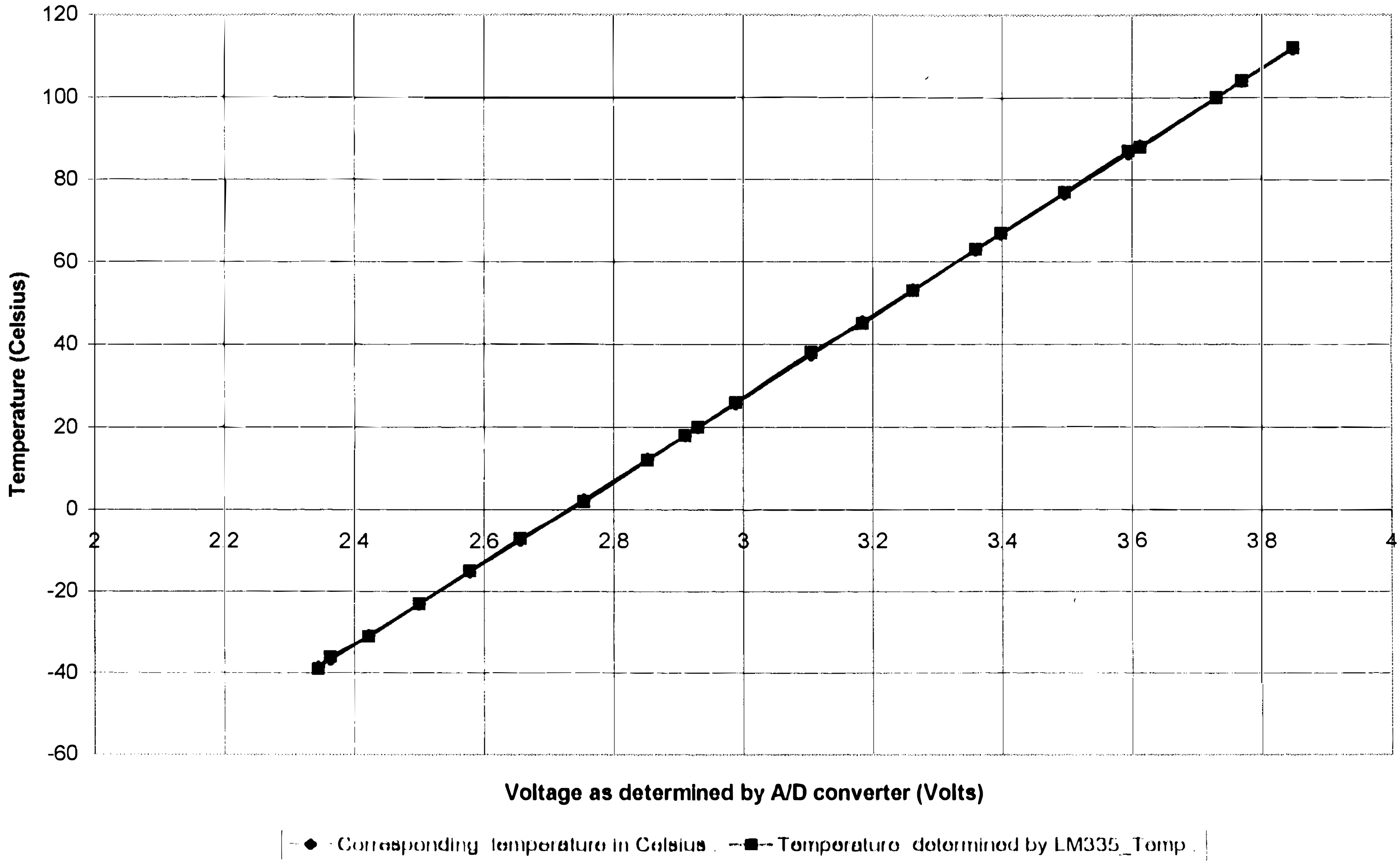
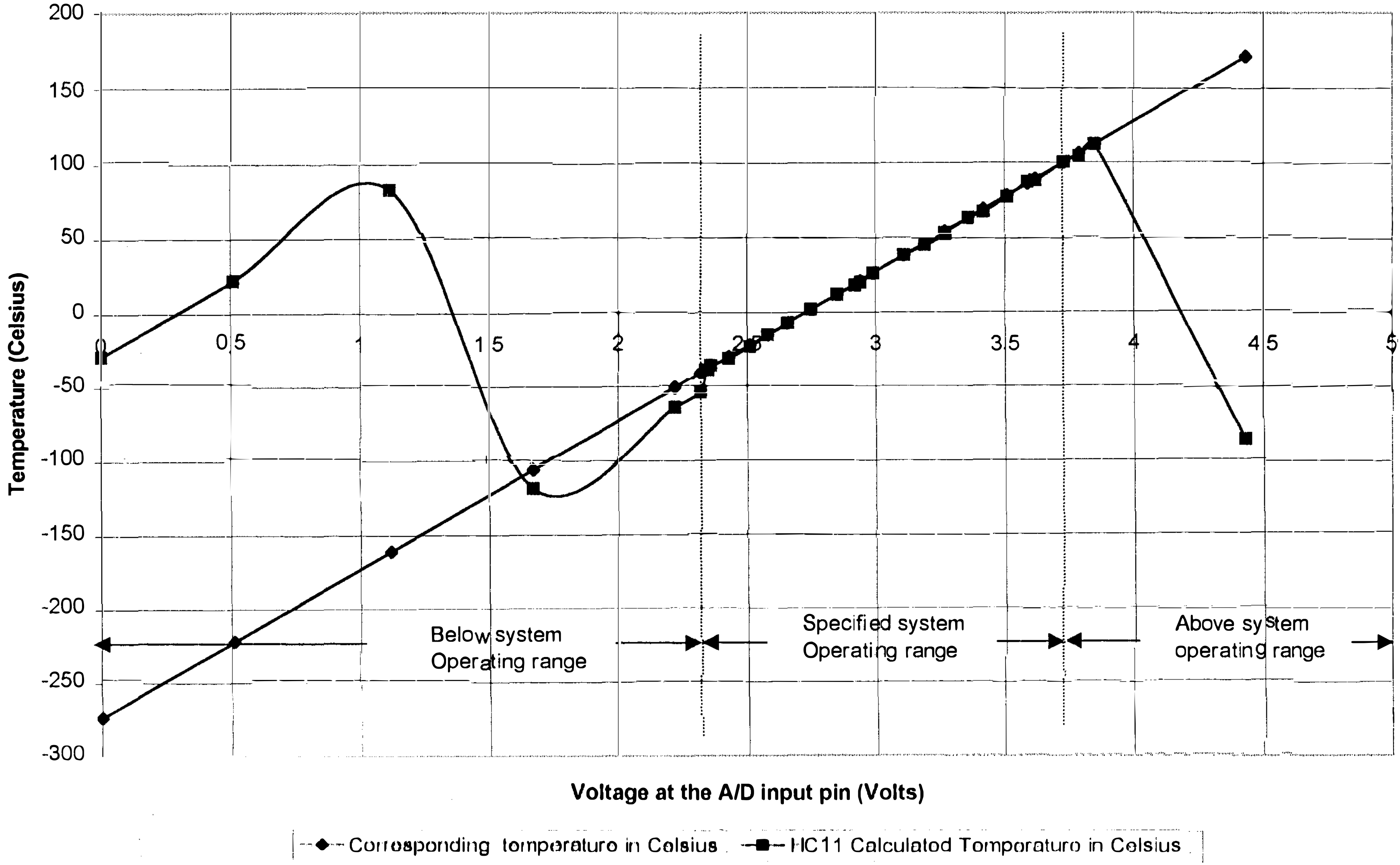


Figure 3.51 A/D converter and LM335_Temp procedure accuracy



system operating range the maximum error, that occurred during testing, between the temperature calculated by the LM335_Temp procedure and the temperature corresponding to the A/D voltage, according to the LM335 temperature sensor formula was 0.7°C.

Figure 3.51 depicts the calculated sensor temperature for different voltages at the A/D converter pin. Figure 3.51 also displays the ideal response based on the ideal LM335 temperature voltage relationship. Figure 3.51 demonstrates that the calculated temperature by the system closely follows the ideal value within the specified system operating range. The maximum error that occurred during the testing between the ideal and calculated value of temperature was 2°C.

8.5.1 Testing system warnings

The trigger values for the LM335_Temp warnings were set to the following values during testing.

Warning constant name	Warning constant value
OTLow_Red	$94
OTYellow	$96
OTRed	$98

Warnings are set by the LM335_Temp procedure based on the A/D converter voltage. The orange LED of the warning display circuit was lit for all cases where the hexadecimal value corresponding to A/D voltage was below $94. The green LED only became lit for the case where the hexadecimal value corresponding to A/D voltage was $95. The yellow LED was lit for the cases where the hexadecimal value corresponding to A/D voltage was above or equal to $96 and below $98. The red LED became lit for the cases where the hexadecimal value corresponding to A/D voltage was above or equal to $98. These results coincided with expectations.

8.5.1 Preliminary testing of the entire LM335 system

A simple test was conducted using the entire LM335 system to determine if the system was functioning. The test does not give any useful data about the system accuracy. A diagram showing the testing circuit is shown in Figure 3.52.

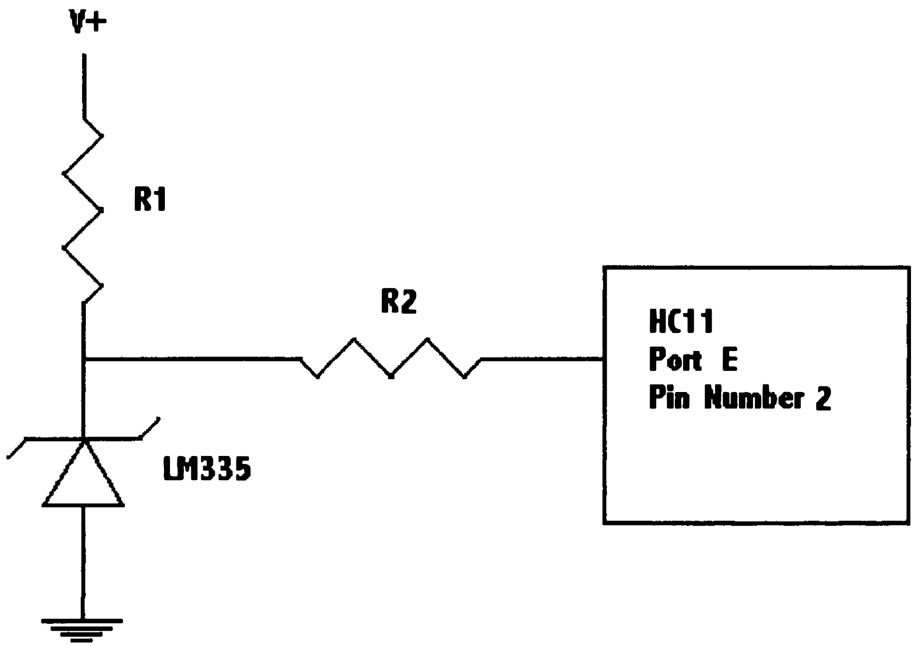


Figure 3.52 Whole system test circuit

8.5.1.1 Testing method

1. The software was setup to operate the LM335_Temp procedure for input at A/D pin number 2. This was required since the circuit simulating LM335 voltage was connected to A/D pin 2.
2. The software was setup to run the Drive_Led procedure for the input at A/D pin number 2. This was required so that the part of the LM335_Temp procedure that sets warnings could be tested.
3. The whole system was placed in an air conditioned environment. This environment was chosen because it was cool. A cool environment would cause the LM335 temperature to be in the range corresponding to a "Low_Red" warning.
4. The warning display circuit was connected to the HC11 in the normal configuration. This was done so that the warnings generated by the LM335_Temp procedure could be displayed.
5. The test circuit was set up as shown in Figure 3.52.
6. The HC11 system software was started.
7. The status of the warning display circuit LEDs was noted

8. Fingers were placed and left on the LM335 temperature sensor package to warm the sensor.
9. The status of the warning display circuit LEDs was monitored until the the Red LED corresponding to a Red warning level remained lit for 10 seconds.
10. Fingers were removed from the LM335 temperature sensor to allow it to cool.
11. The status of the LEDs was monitored until the orange LED corresponding to a Low_red warning level remained lit for 10 seconds.

8.5.1.2 Results

The Warning circuit's orange LED and master warning LED were lit at the beginning of the test. After fingers were applied the LEDs went through the following sequence;

1. Orange LED + Master_Warning LED Lit
2. LEDs flickered between (Orange + Master_Warning) and (Green Lit)
3. Only Green lit
4. LEDs flickered between Green and Yellow
5. Only Yellow lit
6. LEDs flickered between (Yellow) and (Red + Master_Warning)
7. Red remained lit

After fingers were removed the sequence repeated but in reverse order.

8.5.2 The significance of the testing results

The following conclusions have been determined from the testing of the LM335 temperature system.

- The warnings are being set properly by the LM335_Temp procedure
- The LM335 temperature sensor system can be expected to calculate temperature accurate to ±2°C if the output of the LM335 interface is perfect. This within the specified operating accuracy of ±3°C.
- The calculated temperature by the system outside the specified operating range is inaccurate. As a result the system should not be operated outside its specified range.
- The entire LM335 sensor system performs its desired function but with an unknown level of accuracy. Further testing using known environment temperatures is needed in order to determine system accuracy.

CHAPTER 9: Engine Speed System

The primary purpose of the Engine Speed system is to determine the engine RPM and store this value in memory. The secondary purpose of the system is to set the warning corresponding to engine speed. The Engine Speed System is supported by the Operating System. The Engine Speed System is essential to the operation of the Warning System. The relationship between the Engine Speed System and the other subsystems is illustrated in Figure 3.1

To achieve the system objectives the dataflow illustrated in Figure 3.53 must be implemented.

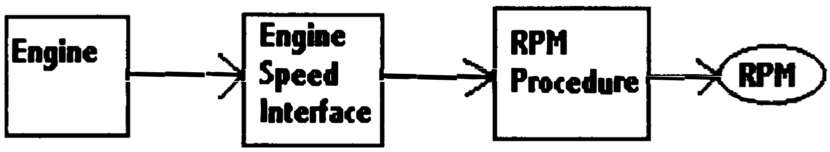


Figure 3.53 Engine Speed System

This section describes;

- The Engine Speed Interface
- RPM Procedure

9.1 Engine Speed Interface

The purpose of the engine speed interface is to generate a series of logic pulses with a period corresponding to the speed of the Engine. This interface could be implemented in several ways. Two possible implementations are;

- Using a light source and sensor and a black and white disc to generate logic pulses as the disc is spun by the engine
- Using circuitry to detect the when the ignition system generates a spark and generate a single logic pulse for each spark.

9.2 RPM Procedure

The primary task of the RPM procedure is to calculate the aircraft engine RPM. The procedure achieves performs this task by measuring the time between two successive electrical pulses. To accommodate different RPM interfaces the procedure can work with different number of pulses per engine revolution.

9.2.1 Tasks

The RPM procedure performs several tasks. These tasks include;

- Receive as a parameter the number of the input capture pin being used

- Receive as a parameter the number to multiply by to get the RPM from the pulse frequency
- Determine the period between successive pulses
- Given the pulse period determine the pulse frequency and then the RPM
- Set warnings based on the RPM
- Store the RPM

9.2.2 Special Requirements

In order to use the input capture system the timer prescaler must be set to a factor of 4. This results in a timer of 500 000Hz. This makes the timer pulse period 2μs. The RPM procedure requires a timer pulse period of 2μs.

9.2.3 Required Data

In order to operate the RPM procedure requires access to the following values;

- Warning
- RPM
- RPLow_Red
- RPYellow
- RPRed

The definitions of the required data for the RPM procedure are explained below.

9.2.3.1 Warning

The constant Warning is a pointer to the first address of the warning page. The RPM procedure requries this pointer so that it can update the warning corresponding to the RPM.

9.2.3.2 RPM

RPM is a pointer to the location where the calculated RPM is stored. The procedure needs this pointer so that it can store the calculated RPM in the correct address.

9.2.3.3 RPLow_Red

The constant RPLow_Red is the value that determines whether a Low Red warning is set. If the calculated RPM is equal to or below this value a Low Red warning is set.

9.2.3.4 RPYellow

The constant RPYellow is a value that determines whether a Yellow warning is set. If the calculated RPM is equal to or above this value a Yellow warning is set.

9.2.3.5 RPRed

The constant RPRed is the value that determines whether a Red warning is set. If the calculated RPM is equal to or above this value a Red Warning is set.

9.2.4 Required Procedures

The RPM procedure requires the Set_Master procedure. The starting address of the Set_Master procedure must be known to the RPM procedure so that it may call the Set_Master procedure when required.

9.2.5 Input Parameters

There are two input parameters for the RPM procedure. These input parameters are shown in Table 3.39

Paramter Name	Where the parameter is stored during passing
RPICPin	Accumulator A
RPRtoFRatio	Accumulator B

Table 3.39 Input parameters for the RPM procedure

9.2.6 Local Variables

The RPM procedure has many local variables. These variables are shown in table 3.40. To reduce memory usage some variables use the same memory offset. Variables using the same memory location are active at different parts of the procedure so no conflict occurs.

Local Variable Name	Size of the variable in bytes	Memory offset of the local variable	Address offsets covered by the variable
RPICPin	1	$00	$00
RPRtoFRatio	1	$01	$01
RPCountsHB	1	$02	$02
RPCountsLB	1	$03	$03
RPResult1	2	$04	$04, $05
RPResult2	2	$06	$06, $07
RPResult3	2	$08	$08, $09
RPPeriod	4	$0A	$0A, $0B, $0C, $0D
RPPeriodVLB	1	$0D	$0D
RPPeriodLB	1	$0C	$0C
RPPeriodMB	1	$0B	$0B
RPCount1HB	1	$04	$04
RPCount1LB	1	$05	$05
RPCount2HB	1	$06	$06
RPCount2LB	1	$07	$07
RPOvCount	1	$02	$02
RPShiftBy	1	$02	$02
RPICCount	1	$03	$03
RPFrequencyHB	1	$03	$03
RPFrequencyMB	1	$04	$04
RPFrequencyLB	1	$05	$05
RPFrequencyFH	1	$06	$06
RPFrequencyFL	1	$07	$07
RPICAddress	2	$08	$08, $09

Table 3.40 Local variables for the RPM procedure

9.2.7 Values Changed

There are three values that may be changed at the conclusion of the RPM procedure. These values are explained under the headings;

- RPM Warning
- RPM
- Master_Warning

Definitions and explanations of input parameters, local variables and values changed at the conclusion of the RPM procedure

9.2.7.1 RPICPin

The HC11 has four pins that may be used for input capture. The RPM procedure can be set to receive input from any of the first 3 input capture pins. The RPM procedure needs to know which input capture pin to use. The RPICPin input parameter and local variable specifies which input capture pin is to be used. Legal values for this variable are 1, 2 or 3. RPICPin is one byte in size.

9.2.7.2 RPRtoFRatio

The RPM procedure first determines the input pulse frequency before calculating RPM. The RPM is determined from the frequency by multiplying a constant value. This RPRtoFRatio is this constant value. Legal values of RPRtoFRatio are shown in table 3.41. The number of pulses that the interface circuit generates in one engine revolution determines the value chosen for RPRtoFRatio. Table 3.41 indicates the value for RPRtoFRatio for different interface circuits.

Number of pulses generated per engine revolution by the interface circuit	Correct Value for RPRtoFRatio As a decimal	Correct Value for RPRtoFRatio As a hexadecimal
1	60	$3C
2	30	$1E
3	20	$14
4	15	$0F
5	12	$0C
6	10	$0A
10	6	$06
12	5	$05
15	4	$04
20	3	$03
30	2	$02
60	1	$01

Table 3.41 Correct Values of RPRtoFRatio for various interface circuits

9.2.7.3 RPCountsHB and RPCountsLB

The RPM procedure determines the frequency of the input pulses based on the period of time between successive pulses. The time between successive pulses in timer counts is stored as a 16 bit value in RPCountsHB and RPCountsLB. RPCountsHB stores the most significant byte of the period. RPCountsLB stores the least significant byte of the period.

9.2.7.4 RPResult1, RPResult2 and RPResult3

The variables RPResult1, RPResult2 and RPResult3 are used in the conversion of the pulse period from timer counts to a period in seconds.

9.2.7.5 RPPeriod, RPPeriodVLB, RPPeriodLB and RPPeriodMB

The variables RPPeriod, RPPeriodVLB, RPPeriodLB and RPPeriodMB are used to store the pulse period. The period, in seconds, is represented as a 32-bit binary weighted fraction. The most signifcant byte of this fraction is stored in RPPeriod. The next most significant byte of this fraction is stored in RPPeriodMB. RPPeriodLB stores third most significant byte. The least significant byte of the fraction is stored in RPPeriodVLB. This is best illustrated by an example. For example if the period between pulses was 131.07ms = 0.13107s.
This value as a hexadecimal is $0.218CDE73.
The corresponding values for the variables RP Period, RPPeriodVLB, RPPeriodLB and RPPeriodMB would be as shown in table 3.42

Variable	Value
RPPeriod	$21
RPPeriodMB	$8C
RPPeriodLB	$DE
RPPeriodVLB	$73

Table 3.42 example variable values

9.2.7.6 RPCount1HB and RPCount1LB

The variables RPCount1HB and RPCount1LB are used to store the HC11 timer value when the first input capture occurs. The timer value is a 16-bit number. The variable RPCount1HB is used to store the high byte of this timer value. The low byte of the timer value is stored in RPCount1LB.

9.2.7.7 RPCount2HB and RPCount2LB

The timer value when the second input capture occurs is stored in the variables RPCount2HB and RPCount2LB. The timer value is a 16-bit number. The high byte of the timer value is stored in RPCount2HB and the low byte of the timer value is stored in RPCount2LB.

9.2.7.8 RPOvCount

The variable RPOvCount is used to keep track of the number of timer overflows that occur during the input capture process. This value needs to be known to ensure accurate estimation of the pulse period.

9.2.7.9 RPShiftBy

The frequency of the input pulses is determined from the period of the pulses using a division operation. Unfortunately the HC11 only supports a 16- bit division. As a consequence a compromise solution was developed to make the best use of the HC11 division instruction. The variable RPShiftBy is used as part of this solution. The division is further explained in the section headed "Determining the pulse frequency."**

9.2.7.10 RPICCount

This variable is used to keep track of the number of input capture operations that have occurred.

9.2.7.11 RPFrequencyHB, RPFrequecyMB, RPFrequecyLB, RPFrequencyFH and RPFrequencyFL

The variables RPFrequencyHB, RPFrequecyMB, RPFrequecyLB, RPFrequencyFH and RPFrequencyFL are used to store the calculated pulse frequency. The variables RPFrequencyHB, RPFrequecyMB and RPFrequecyLB store the integer part of the frequency. The most significant byte of the frequency is stored in the variable RPFrequencyHB. The next most significant byte of the frequency is stored in RPFrequencyMB. The least significant integer byte of the frequency is store in RPFrequencyLB. The variables RPFrequencyFH and RPFrequencyFL store the fractional part of the pulse frequency. The fractional part is stored as a binary-weighted fraction. The most significant byte of the fractional part is stored in RPFrequencyFH. The least significant byte of the fractional part is stored in RPFreqencyFL. This is best demonstrated by an example.

If the pulse frequency was calculated to be 7.631Hz then this would correspond to S7.A19 as a hexadecimal. This frequency would be stored as shown in Table 3.43

Variable	Value
RPFrequencyHB	$00
RPFrequencyMB	$00
RPFrequencyLB	$07
RPFrequencyFH	$A1
RPFrequencyFL	$90

Table 3.43 example variable values

9.2.7.12 RPICAddress

The variable RPICAddress is used as a pointer. This pointer points to were the input capture timer value is stored for the input capture pin being used. The value of RPICAddress can be determined from the following Table 3.44 based on the input capture pin.

Input capture pin number	Value of RPICAddress
1	$1010
2	$1012
3	$1014

Table 3.44 Correct RPICAddress value for different input capture pins

9.2.8 Input Capture versus the Pulse Accumulator

The HC11 offers contains two systems that could be used to determine engine RPM. These systems are called Input Capture and the Pulse Accumulator.

9.2.9 Determining RPM using Input Capture

The input capture system detects a particular logic pulse edge and records the timer value when the pulse edge occurs. A method for determining RPM using the HC11 input capture mechanism is as given as follows. It is possible to use the input capture system to determine the time between two successive pulses. The pulse period can be inferred from this recorded time. Given pulse period it is possible to determine pulse frequency. Finally RPM could be determined based on pulse frequency.

9.2.10 Determining RPM using the Pulse Accumulator

It is possible to use the HC11 pulse accumulator system to determine RPM. The HC11 pulse accumulator system can be configured to detect and count a particular pulse edge. The HC11 pulse accumulator could be configured to count the number of pulses that occur in a set time period. It then is a relatively simple process to multiply the number of pulses by an appropriate scaling factor in order to determine RPM.

9.2.11 The Method Chosen

The input capture method for determining engine RPM was chosen to be implemented. The reasons for choosing the input capture method instead of the pulse accumulator method were;

- The input capture system records a 16 bit timer value. The pulse accumulator records the number of pulses as a 8 bit value. This leads to the input capture system be able to determine RPM to higher resolution than a pulse accumulator system if pulse accumulator overflows are not handled
- The appropriate time period over which to count pulses in the pulse accumulator method is dependant on the number of pulses generated per engine revolution. The input capture method does not require this time period adjustment for different numbers of pulses per engine revolution. This is useful when the final RPM interface in not known.

9.2.12 Multiplying 16-bit Numbers

The HC11 multiply instruction performs a multiplication of two 8-bit numbers to yield a 16 bit result. For increased accuracy in the calculation of period a method was required that allowed multiplication of two 16-bit numbers to determine a 32-bit result. One such method can be seen implemented in RPM procedure algorithm shown in Appendix F.

9.2.13 Increasing the RPM Resolution by Most Significant Bytes Division

The HC11 division instructions are a 16-bit instruction bit instructions. Division is required to determine the pulse frequency from the pulse period. The pulse period which will be the division denominator is stored as a 32-bit value. A 16-bit division instruction can not be used to perform division using a 32-bit denominator. The simplest solution to this problem would be to perform the division using the 16 most significant bits of the period. This solution leads to large errors when the denominator is a small value.

To increase the accuracy the denominator for the division is taken to be the most significant non zero bits of the 32 bit period. The result of the division is then shifted by an appropriate number of bits so as to compensate for any change in the position of the radix point caused by the chosen division method. This method greatly improves the accuracy of the calculated RPM.

Areas for additional Work

Although this project has proved the capability of the HC11 microcontroller for use as an input processor in an integrated light aircraft system there is still much research and development required. Some areas for additional work include;

- Constructing a complete RPM System by implementing the RPM interface.
- Implement the whole RPM procedure on an HC11 with more memory and test it.
- Modifying the system so that the Cylinder Head, Battery Voltage and Ambient Temperature systems can handle more than one input
- Implementing of other subsystems to replace other aircraft instruments such as the Fuel level gauge.
- Use other microcontrollers, such an 8051 based microcontroller, to implement an input processor and compare the results with the HC11 implementation.

Conclusion

This project has proved the feasibility and practicality of developing an input processor for an integrated light aircraft system. The project has also demonstrated that the HC11 can be used to develop this input processor. This project has brought an integrated light aircraft instrumentation system one step closer to realisation.

REFERENCES

LM335 Precision Temperature Sensor. (no date). National Semiconductor Corporation

LM324 Low Power Quad Operational Amplifiers. (no date). National Semiconductor Corporation

LM358 Low Power Dual Operational Amplifiers. (no date). National Semiconductor Corporation

LM627 Precision Operational Amplifiers. (no date). National Semiconductor Corporation

LM741 Operational Amplifier. (no date). National Semiconductor Corporation

Lead Acid Battery. (no date). [Handout]. Available from Edith Cowan University, Joondalup Drive, Joondalup, 6027, Western Australia).

.Altronics. (1997) 1997/98 Electronic Products & Components Engineering Catalogue. [Catalogue]. Perth, Western Australia

HC11 M68HC11 Reference Manual. (1991). USA: Motorola Inc
Lead Acid battery circuit also uses lead acid battery reference

Yeo, M., Bowers, G., & Bennet,G. (1996). Handbook of Flight. Prospect Place West Perth Western Australia 6005: TAFE Publications of Western Australia

M68HC11EVBU Universal Evaluation Board Users's Manual. (1992) USA: Motorola Inc

HC11 M68hc11 Technical Data. (1991) USA: Motorola Inc

Sedra, A.S., & Smith, K.C. (1998). Microelectronic Circuits (4th Edition) NewYork: Oxford University Press

Leon, W. C. (1997). Digital and Analog Communicaiton Systems (5th Edition) Sydney: Prentice Hall of Australia Pty. Limited.

Ramakant and Gayakwad . (1988). Op-amps and linear integrating circuits (2nd Edition) Englewood Cliffs New Jersey: Prentace Hall.

TC LTD. (1997). Guide to thermocouple and resistance thermometry (issue 6.0) Uxbridge England: TC LTD

Sheingold, D. H. (1980). Transducer Interfacing Handbook Norward Massachusetts: Analog Devices, Inc.

Appendix A

This appendix contains the following Specification sheets;

- LM335 Temperature Sensor
- LM324 Low Power Quad Operational Amplifier
- LM358 Low Power Dual Operational Amplifier
- LM627 Low Noise Precision Operational Amplifier
- LM741 General Purpose Operational Amplifier

National Semiconductor Corporation

LM135/LM235/LM335, LM135A/LM235A/LM335A Precision Temperature Sensors

General Description

The LM135 series are precision, easily-calibrated, integrated circuit temperature sensors. Operating as a 2-terminal zener, the LM135 has a breakdown voltage directly proportional to absolute temperature at +10 mV/°K. With less than 1Ω dynamic impedance the device operates over a current range of 400 μA to 5 mA with virtually no change in performance. When calibrated at 25°C the LM135 has typically less than 1°C error over a 100°C temperature range. Unlike other sensors the LM135 has a linear output.

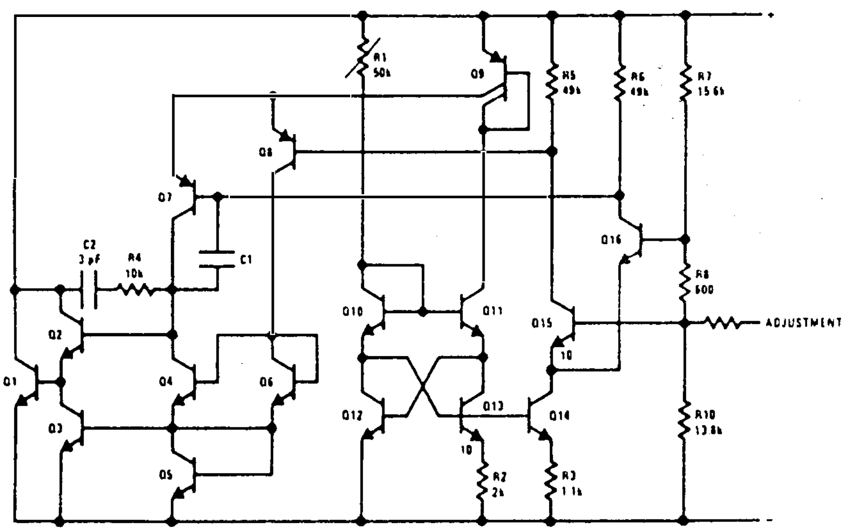
Applications for the LM135 include almost any type of temperature sensing over a −55°C to +150°C temperature range. The low impedance and linear output make interfacing to readout or control circuitry especially easy.

The LM135 operates over a −55°C to +150°C temperature range while the LM235 operates over a −40°C to +125°C temperature range. The LM335 operates from −40°C to +100°C. The LM135/LM235/LM335 are available packaged in hermetic TO-46 transistor packages while the LM335 is also available in plastic TO-92 packages.

Features

- Directly calibrated in °Kelvin
- 1°C initial accuracy available
- Operates from 400 μA to 5 mA
- Less than 1Ω dynamic impedance
- Easily calibrated
- Wide operating temperature range
- 200°C overrange
- Low cost

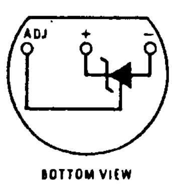
Schematic Diagram



TL/H/5698-1

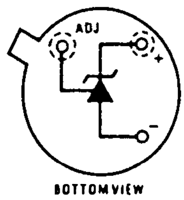
Connection Diagrams

TO-92
Plastic Package



Order Number LM335Z or LM335AZ
See NS Package Number Z03A

TO-46
Metal Can Package*



TL/H/5698-8

*Case is connected to negative pin

Order Number LM135H, LM235H, LM335H, LM135AH, LM235AH or LM335AH
See NS Package Number H03H

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/ Distributors for availability and specifications.
(Note 4)

Reverse Current	15 mA
Forward Current	10 mA
Storage Temperature	
TO-46 Package	−60°C to +180°C
TO-92 Package	−60°C to +150°C

Specified Operating Temp. Range

	Continuous	Intermittent (Note 2)
LM135, LM135A	−55°C to +150°C	150°C to 200°C
LM235, LM235A	−40°C to +125°C	125°C to 150°C
LM335, LM335A	−40°C to +100°C	100°C to 125°C

Lead Temp. (Soldering, 10 seconds)

TO-92 Package:	260°C
TO-46 Package:	300°C

Temperature Accuracy LM135/LM235, LM135A/LM235A (Note 1)

Parameter	Conditions	LM135A/LM235A			LM135/LM235			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$T_C = 25°C, I_R = 1$ mA	2.97	2.98	2.99	2.95	2.98	3.01	V
Uncalibrated Temperature Error	$T_C = 25°C, I_R = 1$ mA		0.5	1		1	3	°C
Uncalibrated Temperature Error	$T_{MIN} \le T_C \le T_{MAX}, I_R = 1$ mA		1.3	2.7		2	5	°C
Temperature Error with 25°C Calibration	$T_{MIN} \le T_C \le T_{MAX}, I_R = 1$ mA		0.3	1		0.5	1.5	°C
Calibrated Error at Extended Temperatures	$T_C = T_{MAX}$ (Intermittent)		2			2		°C
Non-Linearity	$I_R = 1$ mA		0.3	0.5		0.3	1	°C

Temperature Accuracy LM335, LM335A (Note 1)

Parameter	Conditions	LM335A			LM335			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$T_C = 25°C, I_R = 1$ mA	2.95	2.98	3.01	2.92	2.98	3.04	V
Uncalibrated Temperature Error	$T_C = 25°C, I_R = 1$ mA		1	3		2	6	°C
Uncalibrated Temperature Error	$T_{MIN} \le T_C \le T_{MAX}, I_R = 1$ mA		2	5		4	9	°C
Temperature Error with 25°C Calibration	$T_{MIN} \le T_C \le T_{MAX}, I_R = 1$ mA		0.5	1		1	2	°C
Calibrated Error at Extended Temperatures	$T_C = T_{MAX}$ (Intermittent)		2			2		°C
Non-Linearity	$I_R = 1$ mA		0.3	1.5		0.3	1.5	°C

Electrical Characteristics (Note 1)

Parameter	Conditions	LM135/LM235 LM135A/LM235A			LM335 LM335A			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage Change with Current	$400\ \mu A \le I_R \le 5$ mA At Constant Temperature		2.5	10		3	14	mV
Dynamic Impedance	$I_R = 1$ mA		0.5			0.6		Ω
Output Voltage Temperature Coefficient			+10			+10		mV/°C
Time Constant	Still Air 100 ft/Min Air Stirred Oil		80 10 1			80 10 1		sec sec sec
Time Stability	$T_C = 125°C$		0.2			0.2		°C/khr

Note 1: Accuracy measurements are made in a well-stirred oil bath. For other conditions, self heating must be considered.

Note 2: Continuous operation at these temperatures for 10,000 hours for H package and 5,000 hours for Z package may decrease life expectancy of the device.

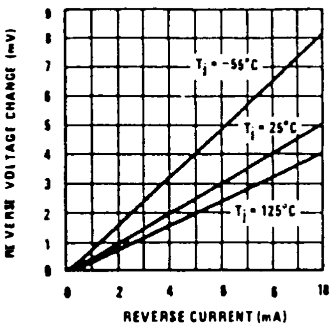
Note 3: Thermal Resistance

	TO-92	TO-46
θ_{JA} (junction to ambient)	202°C/W	400°C/W
θ_{JC} (junction to case)	170°C/W	N/A

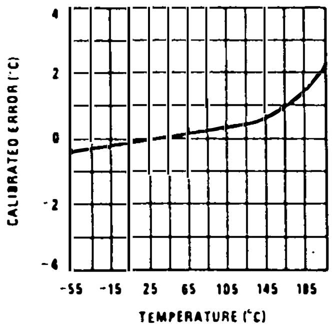
Note 4: Refer to RETS135H for military specifications.

Typical Performance Characteristics

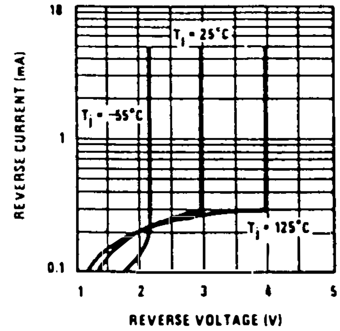
Reverse Voltage Change



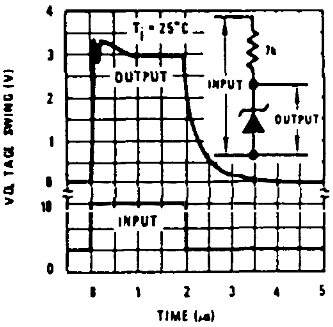
Calibrated Error



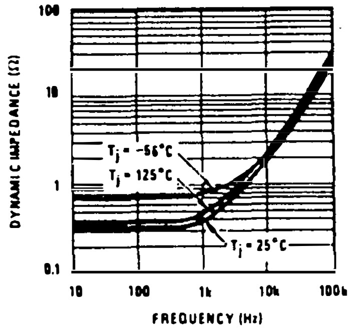
Reverse Characteristics



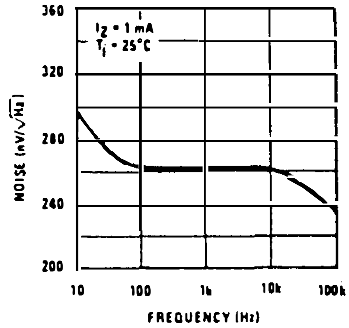
Response Time



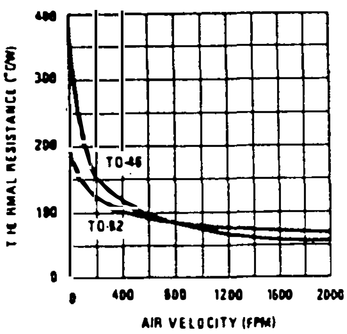
Dynamic Impedance



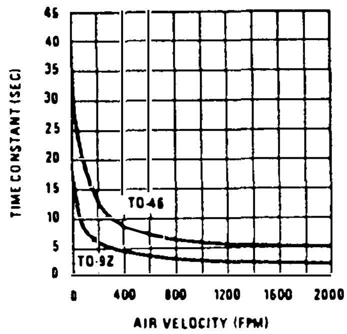
Noise Voltage



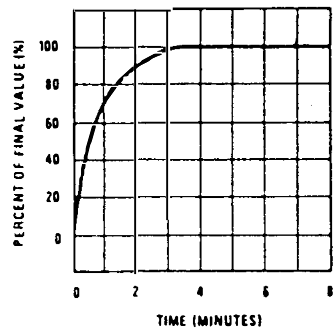
Thermal Resistance Junction to Air



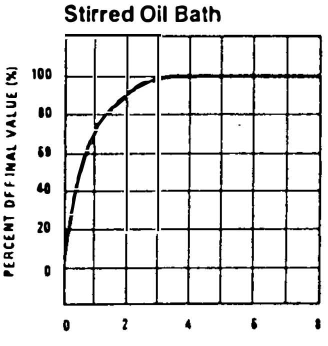
Thermal Time Constant



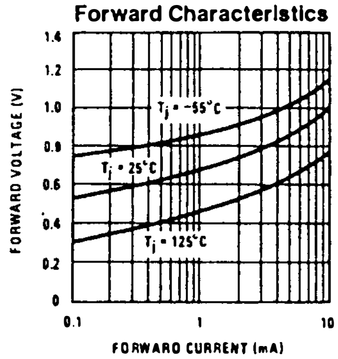
Thermal Response in Still Air



Thermal Response in Stirred Oil Bath



Forward Characteristics



TL/H/5698-3

National Semiconductor

LM124/LM224/LM324, LM124A/LM224A/LM324A, LM2902 Low Power Quad Operational Amplifiers

General Description

The LM124 series consists of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, DC gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124 series can be directly operated off of the standard $+5\ V_{DC}$ power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional $\pm 15\ V_{DC}$ power supplies.

Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.
- The unity gain cross frequency is temperature compensated.
- The input bias current is also temperature compensated.

Advantages

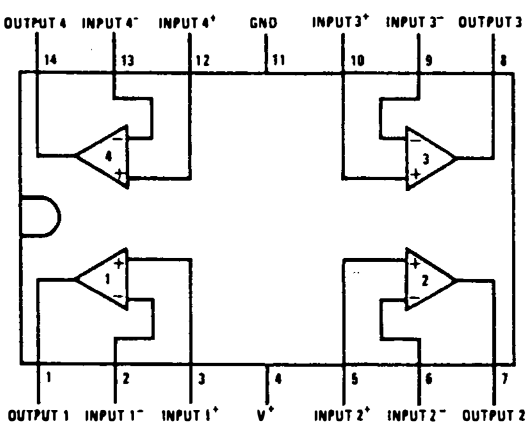
- Eliminates need for dual supplies
- Four internally compensated op amps in a single package
- Allows directly sensing near GND and V_{OUT} also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

- Internally frequency compensated for unity gain
- Large DC voltage gain — 100 dB
- Wide bandwidth (unity gain) — 1 MHz (temperature compensated)
- Wide power supply range:
  - Single supply — $3\ V_{DC}$ to $32\ V_{DC}$
  - or dual supplies — $\pm 1.5\ V_{DC}$ to $\pm 16\ V_{DC}$
- Very low supply current drain (700 μA)—essentially independent of supply voltage
- Low input biasing current — $45\ nA_{DC}$ (temperature compensated)
- Low input offset voltage — $2\ mV_{DC}$
  and offset current — $5\ nA_{DC}$
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing — $0\ V_{DC}$ to $V^+ - 1.5\ V_{DC}$

Connection Diagram

Dual-In-Line Package

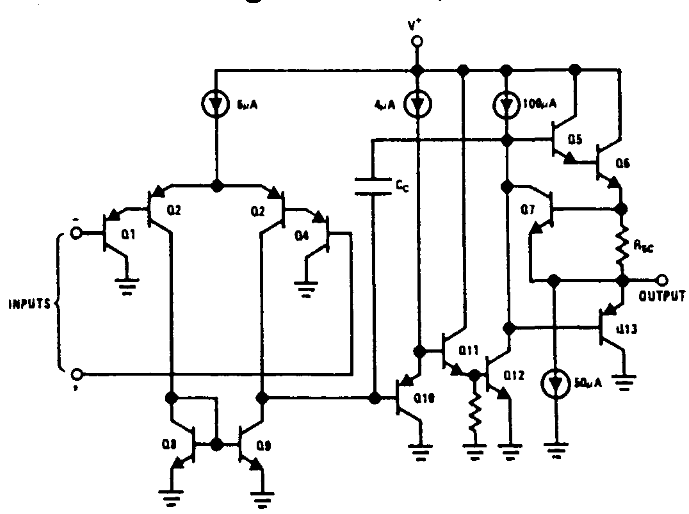


TL/H/9299-1

Top View

Order Number LM124J, LM124AJ, LM224J, LM224AJ, LM324J, LM324AJ, LM324M, LM324AM, LM2902M, LM324N, LM324AN or LM2902N
See NS Package Number J14A, M14A or N14A

Schematic Diagram (Each Amplifier)



TL/H/9299-2

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications. (Note 9)

	LM124/LM224/LM324 LM124A/LM224A/LM324A	LM2902
Supply Voltage, V^+	$32\ V_{DC}$ or $\pm 16\ V_{DC}$	$26\ V_{DC}$ or $\pm 13\ V_{DC}$
Differential Input Voltage	$32\ V_{DC}$	$26\ V_{DC}$
Input Voltage	$-0.3\ V_{DC}$ to $+32\ V_{DC}$	$-0.3\ V_{DC}$ to $+26\ V_{DC}$
Input Current		
($V_{IN} < -0.3\ V_{DC}$) (Note 3)	50 mA	50 mA
Power Dissipation (Note 1)		
Molded DIP	1130 mW	1130 mW
Cavity DIP	1260 mW	1260 mW
Small Outline Package	800 mW	800 mW
Output Short-Circuit to GND		
(One Amplifier) (Note 2)		
$V^+ \leq 15\ V_{DC}$ and $T_A = 25°C$	Continuous	Continuous
Operating Temperature Range		−40°C to +85°C
LM324/LM324A	0°C to +70°C	
LM224/LM224A	−25°C to +85°C	
LM124/LM124A	−55°C to +125°C	

	LM124/LM224/LM324 LM124A/LM224A/LM324A	LM2902
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	260°C	260°C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)	260°C	260°C
Small Outline Package		
Vapor Phase (60 seconds)	215°C	215°C
Infrared (15 seconds)	220°C	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

	LM124/LM224/LM324 LM124A/LM224A/LM324A	LM2902
ESD Tolerance (Note 10)	250V	250V

Electrical Characteristics $V^+ = +5.0\ V_{DC}$, (Note 4), unless otherwise stated

Parameter	Conditions	LM124A			LM224A			LM324A			LM124/LM224			LM324			LM2902			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 5) $T_A = 25°C$		±1	±2		±1	±3		±2	±3		±2	±5		±2	±7		±2	±7	mV_{DC}
Input Bias Current (Note 6)	$I_{IN(+)}$ or $I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		20	50		40	80		45	100		45	150		45	250		45	250	nA_{DC}
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		±2	±10		±2	±15		±5	±30		±3	±30		±5	±50		±5	±50	nA_{DC}
Input Common-Mode Voltage Range (Note 7)	$V^+ = 30\ V_{DC}$, (LM2902, $V^+ = 26\ V_{DC}$), $T_A = 25°C$	0		$V^+-1.5$	0		$V^+-1.5$	0		$V^+-1.5$	0		$V^+-1.5$	0		$V^+-1.5$	0		$V^+-1.5$	V_{DC}
Supply Current	Over Full Temperature Range $R_L = \infty$ On All Op Amps $V^+ = 30V$ (LM2902 $V^+ = 26V$) $V^+ = 5V$		 1.5 0.7	 3 1.2		 1.5 0.7	 3 1.2		 1.5 0.7	 3 1.2		 1.5 0.7	 3 1.2		 1.5 0.7	 3 1.2		 1.5 0.7	 3 1.2	mA_{DC}
Large Signal Voltage Gain	$V^+ = 15\ V_{DC}$, $R_L \geq 2\ k\Omega$, ($V_O = 1\ V_{DC}$ to $11\ V_{DC}$), $T_A = 25°C$	50	100		50	100		25	100		50	100		25	100		25	100		V/mV
Common-Mode Rejection Ratio	DC, $V_{CM} = 0V$ to $V^+ - 1.5\ V_{DC}$, $T_A = 25°C$	70	85		70	85		65	85		70	85		65	85		50	70		dB
Power Supply Rejection Ratio	DC, $V^+ = 5\ V_{DC}$ to $30\ V_{DC}$ (LM2902, $V^+ = 5\ V_{DC}$ to $26\ V_{DC}$), $T_A = 25°C$	65	100		65	100		65	100		65	100		65	100		50	100		dB

Electrical Characteristics $V^+ = +5.0\ V_{DC}$ (Note 4) unless otherwise stated (Continued)

Parameter		Conditions	LM124A			LM224A			LM324A			LM124/LM224			LM324			LM2902			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Amplifier-to-Amplifier Coupling (Note 8)		f = 1 kHz to 20 kHz, T_A = 25°C (Input Referred)		−120			−120			−120			−120			−120			−120		dB
Output Current	Source	$V_{IN}^+ = 1\ V_{DC}$, $V_{IN}^- = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$, $V_O = 2\ V_{DC}$, T_A = 25°C	20	40		20	40		20	40		20	40		20	40		20	40		mA_{DC}
	Sink	$V_{IN}^- = 1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$, $V_O = 2\ V_{DC}$, T_A = 25°C	10	20		10	20		10	20		10	20		10	20		10	20		mA_{DC}
		$V_{IN}^- = 1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$, $V_O = 200\ mV_{DC}$, T_A = 25°C	12	50		12	50		12	50		12	50		12	50		12	50		μA_{DC}
Short Circuit to Ground		(Note 2) $V^+ = 15\ V_{DC}$, T_A = 25°C		40	60		40	60		40	60		40	60		40	60		40	60	mA_{DC}
Input Offset Voltage		(Note 5)			±4			±4			±5			±7			±9			±10	mV_{DC}
Input Offset Voltage Drift		$R_S = 0\Omega$		±7	±20		±7	±20		±7	±30		±7			±7			±7		μV/°C
Input Offset Current		$I_{IN(+)} - I_{IN(-)}$, $V_{CM} = 0V$			±30			±30			±75			±100			±150		±45	±200	nA_{DC}
Input Offset Current Drift		$R_S = 0\Omega$		±10	±200		±10	±200		±10	±200		±10			±10			±10		pA_{DC}/°C
Input Bias Current		$I_{IN(+)}$ or $I_{IN(-)}$		40	100		40	100		40	200		40	300		40	500		40	500	nA_{DC}
Input Common-Mode Voltage Range (Note 7)		$V^+ = +30\ V_{DC}$ (LM2902, $V^+ = 26\ V_{DC}$)	0		$V^+ - 2$	0		$V^+ - 2$	0		$V^+ - 2$	0		$V^+ - 2$	0		$V^+ - 2$	0		$V^+ - 2$	V_{DC}
Large Signal Voltage Gain		$V^+ = +15\ V_{DC}$ (V_O Swing = 1 V_{DC} to 11 V_{DC}) $R_L \geq 2\ k\Omega$	25			25			15			25			15			15			V/mV
Output Voltage Swing	V_{OH}	$V^+ = +30\ V_{DC}$, $R_L = 2\ k\Omega$	26			26			26			26			26			22			V_{DC}
		$R_L \geq 10\ k\Omega$ (LM2902, $V^+ = 26\ V_{DC}$)	27	28		27	28		27	28		27	28		27	28		23	24		V_{DC}
	V_{OL}	$V^+ = 5\ V_{DC}$, $R_L \geq 10\ k\Omega$		5	20		5	20		5	20		5	20		5	20		5	100	mV_{DC}

Electrical Characteristics $V^+ = +5.0\ V_{DC}$ (Note 4) unless otherwise stated (Continued)

Parameter		Conditions		LM124A			LM224A			LM324A			LM124/LM224			LM324			LM2902			Units
				Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Output Current	Source	$V_O = 2\ V_{DC}$	$V_{IN}^+ = +1\ V_{DC}$, $V_{IN}^- = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$	10	20		10	20		10	20		10	20		10	20		10	20		mA_{DC}
	Sink		$V_{IN}^- = +1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$	10	15		5	8		5	8		5	8		5	8		5	8		

Note 1: For operating at high temperatures, the LM324/LM324A, LM2902 must be derated based on a +125°C maximum junction temperature and a thermal resistance of 88°C/W which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM224/LM224A and LM124/LM124A can be derated based on a +150°C maximum junction temperature. The dissipation is the total of all four amplifiers—use external resistors, where possible, to allow the amplifier to saturate of to reduce the power which is dissipated in the integrated circuit.

Note 2: Short circuits from the output to V^+ can cause excessive heating and eventual destruction. When considering short circuits to ground, the maximum output current is approximately 40 mA independent of the magnitude of V^+. At values of supply voltage in excess of +15 V_{DC}, continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.

Note 3: This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action on the IC chip. This transistor action can cause the output voltages of the op amps to go to the V^+ voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than −0.3 V_{DC} (at 25°C).

Note 4: These specifications are limited to $-55°C \le T_A \le +125°C$ for the LM124/LM124A. With the LM224/LM224A, all temperature specifications are limited to $-25°C \le T_A \le +85°C$, the LM324/LM324A temperature specifications are limited to $0°C \le T_A \le +70°C$, and the LM2902 specifications are limited to $-40°C \le T_A \le +85°C$.

Note 5: $V_O \cong 1.4\ V_{DC}$, $R_S = 0\Omega$ with V^+ from 5 V_{DC} to 30 V_{DC}; and over the full input common-mode range (0 V_{DC} to $V^+ - 1.5\ V_{DC}$) at 25°C; for LM2902, V^+ from 5 V_{DC} to 26 V_{DC}.

Note 6: The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.

Note 7: The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V (at 25°C). The upper end of the common-mode voltage range is $V^+ - 1.5V$ (at 25°C), but either or both inputs can go to +32 V_{DC} without damage (+26 V_{DC} for LM2902), independent of the magnitude of V^+.

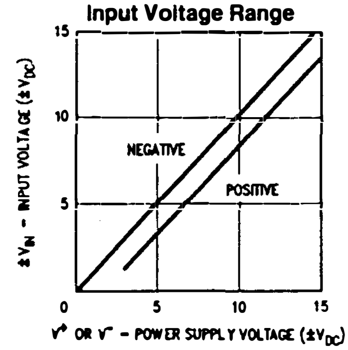
Note 8: Due to proximity of external components, insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequencies.

Note 9: Refer to RETS124AX for LM124A military specifications and refer to RETS124X for LM124 military specifications.

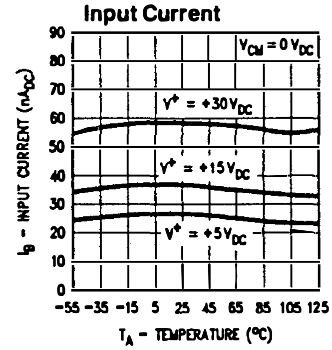
Note 10: Human body model, 1.5 kΩ in series with 100 pF.

Typical Performance Characteristics

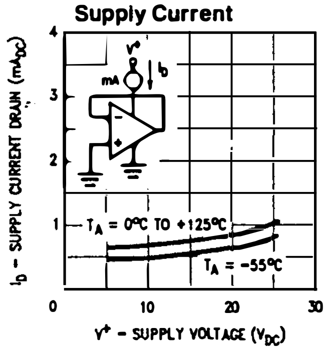
Input Voltage Range



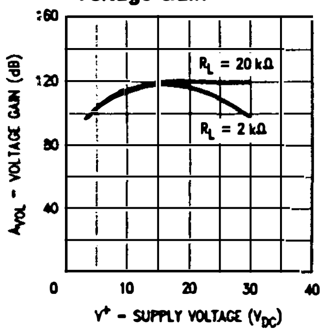
Input Current



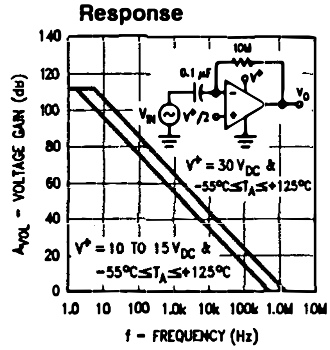
Supply Current



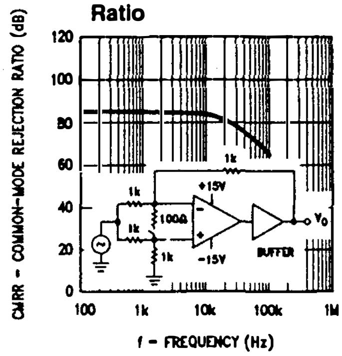
Voltage Gain



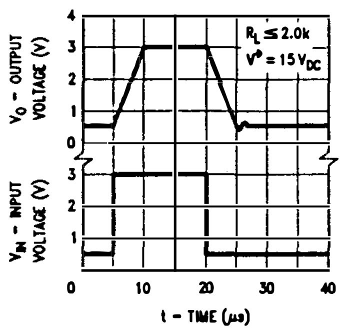
Open Loop Frequency Response



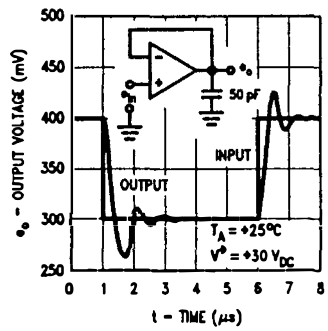
Common Mode Rejection Ratio



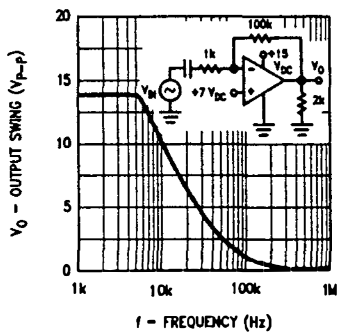
Voltage Follower Pulse Response



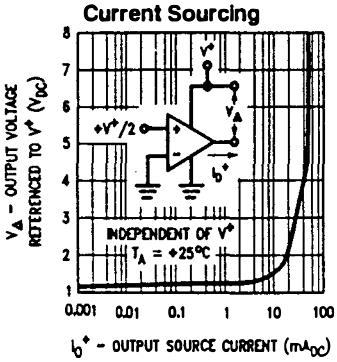
Voltage Follower Pulse Response (Small Signal)



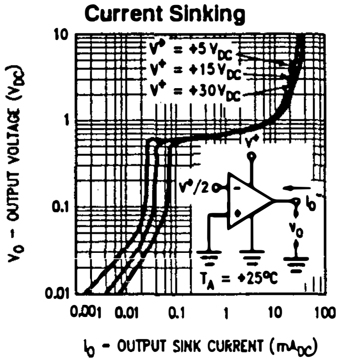
Large Signal Frequency Response



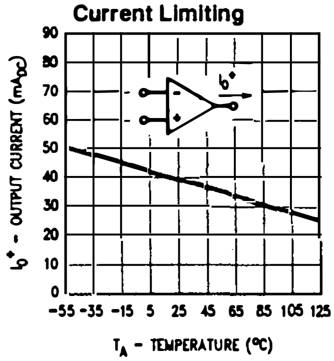
Output Characteristics Current Sourcing



Output Characteristics Current Sinking



Current Limiting



TL/H/9299-3

National Semiconductor

LM158/LM258/LM358, LM158A/LM258A/LM358A, LM2904 Low Power Dual Operational Amplifiers

General Description

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM158 series can be directly operated off of the standard +5 V_{DC} power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional ±15 V_{DC} power supplies.

Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.
- The unity gain cross frequency is temperature compensated.
- The input bias current is also temperature compensated.

Advantages

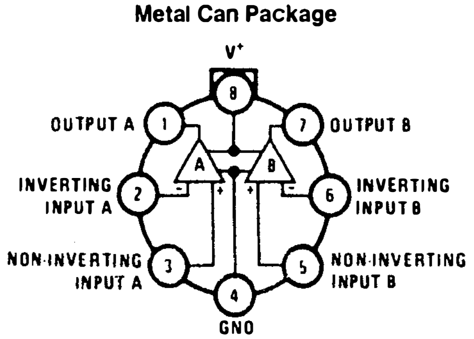
- Eliminates need for dual supplies
- Two internally compensated op amps in a single package
- Allows directly sensing near GND and V_{OUT} also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation
- Pin-out same as LM1558/LM1458 dual operational amplifier

Features

- Internally frequency compensated for unity gain
- Large dc voltage gain — 100 dB
- Wide bandwidth (unity gain) (temperature compensated) — 1 MHz
- Wide power supply range:
  - Single supply — 3 V_{DC} to 32 V_{DC}
  - or dual supplies — ±1.5 V_{DC} to ±16 V_{DC}
- Very low supply current drain (500 μA)—essentially independent of supply voltage
- Low input biasing current (temperature compensated) — 45 nA_{DC}
- Low input offset voltage — 2 mV_{DC}
  and offset current — 5 nA_{DC}
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing — 0 V_{DC} to $V^+ - 1.5$ V_{DC}

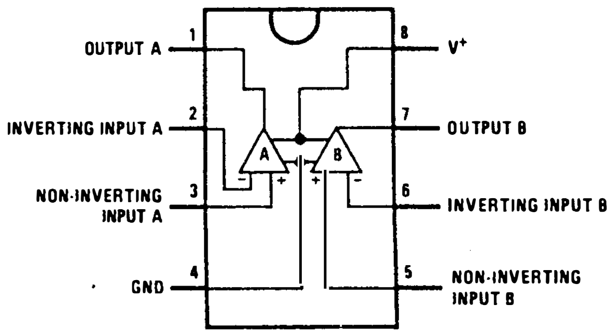
Connection Diagrams (Top Views)

Metal Can Package



TL/H/7787-1

Order Number LM158AH, LM158H, LM258AH, LM258H, LM358AH or LM358H
See NS Package Number H08C



TL/H/7787-2

Order Number LM158J, LM158AJ or LM358J
See NS Package Number J08A
Order Number LM358M, LM358AM or LM2904M
See NS Package Number M08A
Order Number LM358AN, LM358N or LM2904N
See NS Package Number N08E

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications. (Note 9)

	LM158/LM258/LM358 LM158A/LM258A/LM358A	LM2904
Supply Voltage, V^+	32 V_{DC} or ±16 V_{DC}	26 V_{DC} or ±13 V_{DC}
Differential Input Voltage	32 V_{DC}	26 V_{DC}
Input Voltage	−0.3 V_{DC} to +32 V_{DC}	−0.3 V_{DC} to +26 V_{DC}
Power Dissipation (Note 1)		
Molded DIP (LM358N)	830 mW	830 mW
Metal Can (LM158H/ LM258H/LM358H)	550 mW	
Small Outline Package	530 mW	530 mW
Output Short-Circuit to GND (One Amplifier) (Note 2) $V^+ \leq 15\ V_{DC}$ and $T_A = 25°C$	Continuous	Continuous
Input Current ($V_{IN} < -0.3\ V_{DC}$) (Note 3)	50 mA	50 mA

	LM158/LM258/LM358 LM158A/LM258A/LM358A	LM2904
Operating Temperature Range		
LM358	0°C to +70°C	−40°C to +85°C
LM258	−25°C to +85°C	
LM158	−55°C to +125°C	
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C
Lead Temperature, DIP (Soldering, 10 seconds)	260°C	260°C
Lead Temperature, Metal Can (Soldering, 10 seconds)	300°C	300°C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)	260°C	260°C
Small Outline Package		
Vapor Phase (60 seconds)	215°C	215°C
Infrared (15 seconds)	220°C	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

ESD Tolerance (Note 10)	250V	250V

Electrical Characteristics $V^+ = +5.0\ V_{DC}$, unless otherwise stated

Parameter	Conditions	LM158A			LM258A			LM358A			LM158/LM258			LM358			LM2904			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 5), $T_A = 25°C$		±1	±2		±1	±3		±2	±3		±2	±5		±2	±7		±2	±7	mV_{DC}
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$, $T_A = 25°C$, $V_{CM} = 0V$, (Note 6)		20	50		40	80		45	100		45	150		45	250		45	250	nA_{DC}
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		±2	±10		±2	±15		±5	±30		±3	±30		±5	±50		±5	±50	nA_{DC}
Input Common-Mode Voltage Range	$V^+ = 30\ V_{DC}$, (Note 7) (LM2904, $V^+ = 26V$), $T_A = 25°C$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	V_{DC}
Supply Current	Over Full Temperature Range $R_L = \infty$ on All Op Amps $V^+ = 30V$ (LM2904 $V^+ = 26V$)		1	2		1	2		1	2		1	2		1	2		1	2	mA_{DC}
	$V^+ = 5V$		0.5	1.2		0.5	1.2		0.5	1.2		0.5	1.2		0.5	1.2		0.5	1.2	mA_{DC}

Electrical Characteristics (Continued) $V^+ = +5.0\ V_{DC}$, Note 4, unless otherwise stated

Parameter	Conditions	LM158A			LM258A			LM358A			LM158/LM258			LM358			LM2904			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain	$V^+ = 15\ V_{DC}$, $T_A = 25°C$, $R_L \geq 2\ k\Omega$, (For $V_O = 1\ V_{DC}$ to $11\ V_{DC}$)	50	100		50	100		25	100		50	100		25	100		25	100		V/mV
Common-Mode Rejection Ratio	DC, $T_A = 25°C$, $V_{CM} = 0V$ to $V^+ - 1.5\ V_{DC}$	70	85		70	85		65	85		85	85		65	85		50	70		dB
Power Supply Rejection Ratio	DC, $V^+ = 5\ V_{DC}$ to $30\ V_{DC}$ (LM2904, $V^+ = 5\ V_{DC}$ to $26\ V_{DC}$), $T_A = 25°C$	65	100		65	100		65	100		65	100		65	100		50	100		dB
Amplifier-to-Amplifier Coupling	f = 1 kHz to 20 kHz, $T_A = 25°C$ (Input Referred), (Note 8)		−120			−120			−120			−120			−120			−120		dB
Output Current Source	$V_{IN}^+ = 1\ V_{DC}$, $V_{IN}^- = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$, $V_O = 2\ V_{DC}$, $T_A = 25°C$	20	40		20	40		20	40		20	40		20	40		20	40		mA_{DC}
Sink	$V_{IN}^- = 1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$ $V^+ = 15\ V_{DC}$, $T_A = 25°C$, $V_O = 2\ V_{DC}$	10	20		10	20		10	20		10	20		10	20		10	20		mA_{DC}
	$V_{IN}^- = 1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$ $T_A = 25°C$, $V_O = 200\ mV_{DC}$, $V^+ = 15\ V_{DC}$	12	50		12	50		12	50		12	50		12	50		12	50		μA_{DC}
Short Circuit to Ground	$T_A = 25°C$, (Note 2), $V^+ = 15\ V_{DC}$		40	60		40	60		40	60		40	60		40	60		40	60	mA_{DC}
Input Offset Voltage	(Note 5)			±4			±4			±5			±7			±9			±10	mV_{DC}
Input Offset Voltage Drift	$R_S = 0\Omega$		7	15		7	15		7	20		7			7			7		μV/°C
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$			±30			±30			±75			±100			±150		±45	±200	nA_{DC}
Input Offset Current Drift	$R_S = 0\Omega$		10	200		10	200		10	300		10			10			10		pA_{DC}/°C
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$		40	100		40	100		40	200		40	300		40	500		40	500	nA_{DC}

Electrical Characteristics (Continued) $V^+ = +5.0\ V_{DC}$, Note 4, unless otherwise stated

Parameter	Conditions	LM158A			LM258A			LM358A			LM158/LM258			LM358			LM2904			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Common-Mode Voltage Range	$V^+ = 30\ V_{DC}$, (Note 7) (LM2904, $V^+ = 26\ V_{DC}$)	0		V^+-2	0		V^+-2	0		V^+-2	0		V^+-2	0		V^+-2	0		V^+-2	V_{DC}
Large Signal Voltage Gain	$V^+ = +15\ V_{DC}$ ($V_O = 1\ V_{DC}$ to $11\ V_{DC}$) $R_L \geq 2\ k\Omega$	25			25			15			25			15			15			V/mV
Output Voltage Swing																				
V_{OH}	$V^+ = +30\ V_{DC}$, $R_L = 2\ k\Omega$	26			26			26			26			26			22			V_{DC}
	$R_L \geq 10\ k\Omega$ (LM2904, $V^+ = 26\ V_{DC}$)	27	28		27	28		27	28		27	28		27	28		23	24		V_{DC}
V_{OL}	$V^+ = 5\ V_{DC}$, $R_L \geq 10\ k\Omega$		5	20		5	20		5	20		5	20		5	20		5	100	mV_{DC}
Output Current Source	$V_O = 2\ V_{DC}$, $V_{IN}^+ = +1\ V_{DC}$, $V_{IN}^- = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$	10	20		10	20		10	20		10	20		10	20		10	20		mA_{DC}
Sink	$V_{IN}^- = +1\ V_{DC}$, $V_{IN}^+ = 0\ V_{DC}$, $V^+ = 15\ V_{DC}$	10	15		5	8		5	8		5	8		5	8		5	8		mA_{DC}

Note 1: For operating at high temperatures, the LM358/LM358A, LM2904 must be derated based on a +125°C maximum junction temperature and a thermal resistance of 120°C/W which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM258/LM258A and LM158/LM158A can be derated based on a +150°C maximum junction temperature. The dissipation is the total of both amplifiers—use external resistors, where possible, to allow the amplifier to saturate or to reduce the power which is dissipated in the integrated circuit.

Note 2: Short circuits from the output to V^+ can cause excessive heating and eventual destruction. When considering short cirucits to ground, the maximum output current is approximately 40 mA independent of the magnitude of V^+. At values of supply voltage in excess of $+15\ V_{DC}$, continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.

Note 3: This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action on the IC chip. This transistor action can cause the output voltages of the op amps to go to the V^+ voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than $-0.3\ V_{DC}$ (at 25°C).

Note 4: These specifications are limited to $-55°C \leq T_A \leq +125°C$ for the LM158/LM158A. With the LM258/LM258A, all temperature specifications are limited to $-25°C \leq T_A \leq +85°C$, the LM358/LM358A temperature specifications are limited to $0°C \leq T_A \leq +70°C$, and the LM2904 specifications are limited to $-40°C \leq T_A \leq +85°C$.

Note 5: $V_O \cong 1.4\ V_{DC}$, $R_S = 0\Omega$ with V^+ from $5\ V_{DC}$ to $30\ V_{DC}$; and over the full input common-mode range ($0\ V_{DC}$ to $V^+ - 1.5\ V_{DC}$) at 25°C. For LM2904, V^+ from $5\ V_{DC}$ to $26\ V_{DC}$.

Note 6: The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.

Note 7: The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V (at 25°C). The upper end of the common-mode voltage range is $V^+ - 1.5V$ (at 25°C), but either or both inputs can go to $+32\ V_{DC}$ without damage ($+26\ V_{DC}$ for LM2904), independent of the magnitude of V^+.

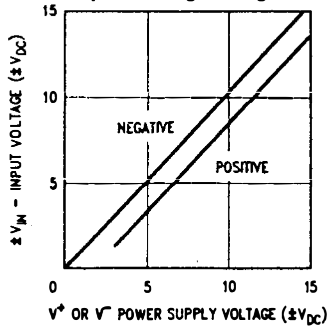
Note 8: Due to proximity of external components, insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequencies.

Note 9: Refer to RETS158AX for LM158A military specifications and to RETS158X for LM158 military specifications.

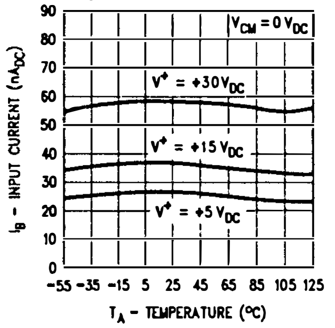
Note 10: Human body model, 1.5 kΩ in series with 100 pF.

Typical Performance Characteristics

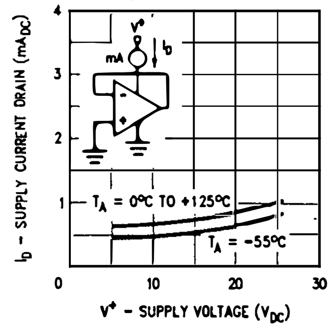
Input Voltage Range



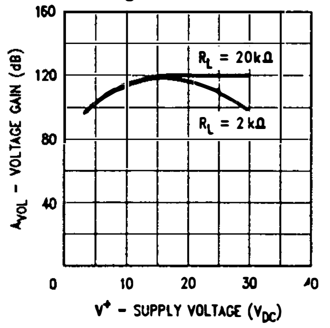
Input Current



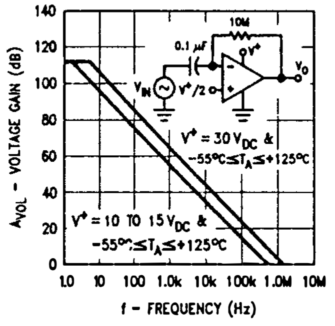
Supply Current



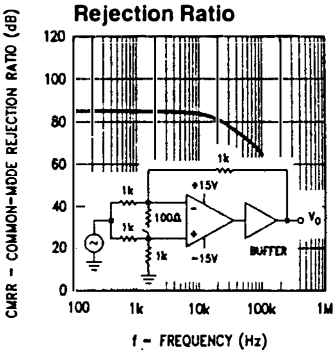
Voltage Gain



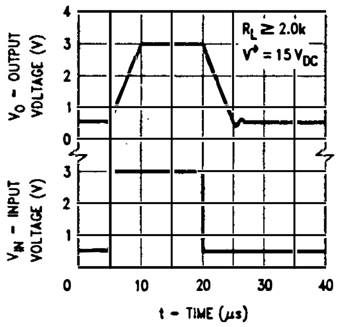
Open Loop Frequency Response



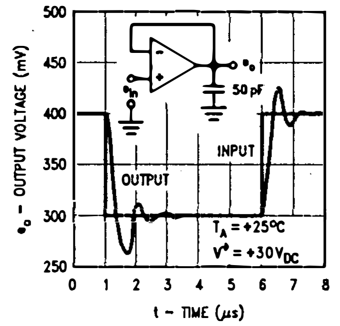
Common-Mode Rejection Ratio



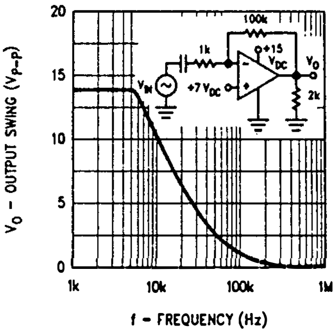
Voltage Follower Pulse Response



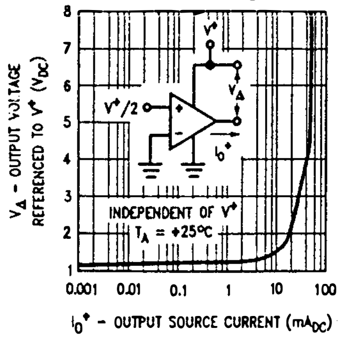
Voltage Follower Pulse Response (Small Signal)



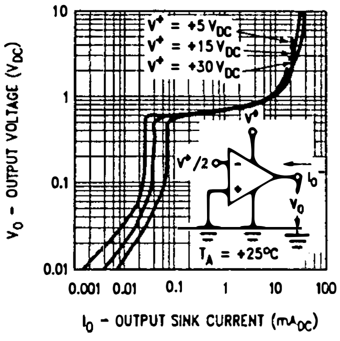
Large Signal Frequency Response



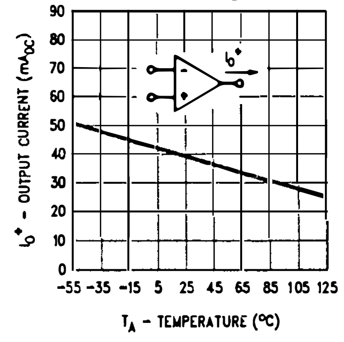
Output Characteristics Current Sourcing



Output Characteristics Current Sinking



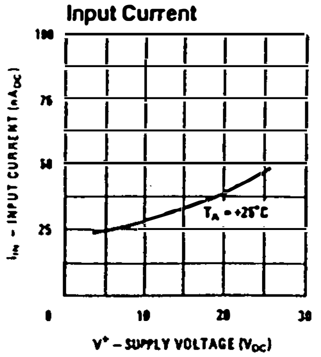
Current Limiting



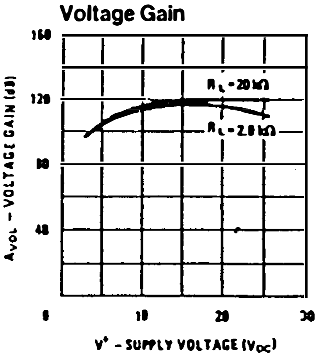
TL/H/7787–4

Typical Performance Characteristics (Continued) (LM2902 only)

Input Current



Voltage Gain



TL/H/7787–5

Application Hints

The LM158 series are op amps which operate with only a single power supply voltage, have true-differential inputs, and remain in the linear mode with an input common-mode voltage of 0 V_{DC}. These amplifiers operate over a wide range of power supply voltage with little change in performance characteristics. At 25°C amplifier operation is possible down to a minimum supply voltage of 2.3 V_{DC}.

Precautions should be taken to insure that the power supply for the integrated circuit never becomes reversed in polarity or that the unit is not inadvertently installed backwards in a test socket as an unlimited current surge through the resulting forward diode within the IC could cause fusing of the internal conductors and result in a destroyed unit.

Large differential input voltages can be easily accomodated and, as input differential voltage protection diodes are not needed, no large input currents result from large differential input voltages. The differential input voltage may be larger than V+ without damaging the device. Protection should be provided to prevent the input voltages from going negative more than −0.3 V_{DC} (at 25°C). An input clamp diode with a resistor to the IC input terminal can be used.

To reduce the power supply current drain, the amplifiers have a class A output stage for small signal levels which converts to class B in a large signal mode. This allows the amplifiers to both source and sink large output currents. Therefore both NPN and PNP external current boost transistors can be used to extend the power capability of the basic amplifiers. The output voltage needs to raise approximately 1 diode drop above ground to bias the on-chip vertical PNP transistor for output current sinking applications.

For ac applications, where the load is capacitively coupled to the output of the amplifier, a resistor should be used, from the output of the amplifier to ground to increase the class A bias current and prevent crossover distortion. Where the load is directly coupled, as in dc applications, there is no crossover distortion.

Capacitive loads which are applied directly to the output of the amplifier reduce the loop stability margin. Values of 50 pF can be accomodated using the worst-case non-inverting unity gain connection. Large closed loop gains or resistive isolation should be used if larger load capacitance must be driven by the amplifier.

The bias network of the LM158 establishes a drain current which is independent of the magnitude of the power supply voltage over the range of 3 V_{DC} to 30 V_{DC}.

Output short circuits either to ground or to the positive power supply should be of short time duration. Units can be destroyed, not as a result of the short circuit current causing metal fusing, but rather due to the large increase in IC chip dissipation which will cause eventual failure due to excessive function temperatures. Putting direct short-circuits on more than one amplifier at a time will increase the total IC power dissipation to destructive levels, if not properly protected with external dissipation limiting resistors in series with the output leads of the amplifiers. The larger value of output source current which is available at 25°C provides a larger output current capability at elevated temperatures (see typical performance characteristics) than a standard IC op amp.

The circuits presented in the section on typical applications emphasize operation on only a single power supply voltage. If complementary power supplies are available, all of the standard op amp circuits can be used. In general, introducing a pseudo-ground (a bias voltage reference of V+/2) will allow operation above and below this value in single power supply systems. Many application circuits are shown which take advantage of the wide input common-mode voltage range which includes ground. In most cases, input biasing is not required and input voltages which range to ground can easily be accommodated.

National Semiconductor

LM627/LM637 Precision Operational Amplifiers

General Description

The LM627/LM637 series feature extremely low noise and excellent precision along with high speed. Voltage noise is a low 3 nV/√Hz in the flat band and rises to only 3.5 nV/√Hz at 10 Hz. The A grades offer guaranteed specifications of 25 μV offset voltage and 0.3 μV/°C drift, and their *guaranteed* 126 dB CMRR, 120 dB PSRR and voltage gain of 5 Million ensure an ultra-low V_{OS} under all conditions.

The unity-gain stable LM627 is nearly twice as fast as the OP-27 with a slew rate of 4.5 V/μs and a 14 MHz gain-bandwidth product. Stable at gains of 5 or more, the decompensated LM637 is considerably faster.

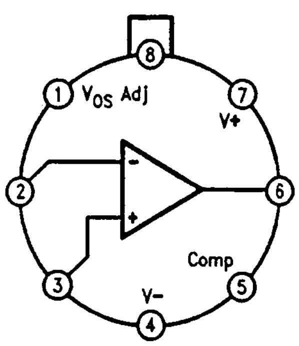
Other enhancements of the LM627/LM637 include a guaranteed 600Ω load drive capability over temperature: ±10V output swing at voltage gains over one million. Bias current has been reduced to 10 nA for the A and B grades and 25 nA for the C grade. Furthermore the LM627 may be overcompensated to allow it to drive capacitive loads up to 2000 pF while maintaining its superb dc specs.

Features

- Low Noise — 3 nV/√Hz@1 kHz; 3.5 nV/√Hz@ 10 Hz
- Low V_{OS} — 25 μV Max
- Low Drift — 0.3 μV/°C Max
- Offset Drift 100% Tested (A and B grades)
- Noise Voltage 100% Tested (A and B grades)
- High Gain — 5 Million Min
- High CMRR — 126 dB Min
- High PSRR — 120 dB Min
- High Speed
  - LM627: 14 MHz Gain-Bandwidth; 4.5 V/μs Slew Rate
  - LM637: 65 MHz Gain-Bandwidth; 14 V/μs Slew Rate
- *Guaranteed* 600Ω drive over temperature
- Wide Power Supply Range — ±3.5V to ±18V
- Overcompensation Pin — Allows driving high C_L

Connection Diagrams

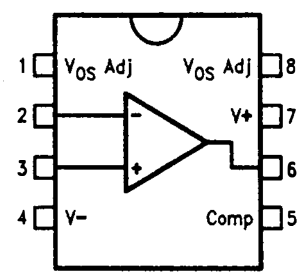
Metal Can Package



TL/H/9212-1

Top View

DIP Packages



TL/H/9212-2

Ordering Information

LM627

Package	Temperature Range		NSC Drawing
	Military	Commercial	
TO-99	LM627AMH LM627BMH	LM627ACH LM627BCH LM627CH	H08C
8-Pin Cerdip	LM627AMJ LM627BMJ	LM627ACJ LM627BCJ LM627CJ	J08A
8-Pin Molded DIP		LM627ACN LM627BCN LM627CN	N08E

LM637

Package	Temperature Range		NSC Drawing
	Military	Commercial	
TO-99	LM637AMH LM637BMH	LM637ACH LM637BCH LM637CH	H08C
8-Pin Cerdip	LM637AMJ LM637BMJ	LM637ACJ LM637BCJ LM637CJ	J08A
8-Pin Molded DIP		LM637ACN LM637BCN LM637CN	N08E

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Differential Input Overdrive Current (Note 7)	±25 mA
Suppy Voltage	44V
Input Voltage	Supply Voltage
Output Short Circuit to Gnd	Continuous
Power Dissipation (Note 8)	
Molded DIP	1300 mW
Ceramic DIP	1190 mW
Metal Can	830 mW
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 5 sec.)	260°C
Maximum Junction Temperature	150°C
ESD Rating	3 kV
C_{ZAP} = 100 pF, R_{ZAP} = 1.5 kΩ	

Operating Ratings

Temperature Range (Note 8)	
AM and BM grades	$-55°C \le T_J \le +125°C$
AC, BC, and C grades	$-25°C \le T_J \le +85°C$

Electrical Characteristics

All limits guaranteed for T_J = 25°C, V_{CM} = 0, V_O = 0 and ±15V supplies unless otherwise specified. **Boldface limits apply at operating temperature extremes.**

Parameter	Conditions	Typ	LM627AM LM637AM Tested Limit (Note 5)	LM627AM LM637AM Design Limit (Note 6)	LM627BM LM637BM Tested Limit (Note 5)	LM627BM LM637BM Design Limit (Note 6)	Units
Input Offset Voltage	(Note 2)	15	25 **55**		50 **110**		μV Max
Input Offset Voltage Drift	(Note 3)	0.2	**0.3**		**0.6**		μV/°C Max
Input Offset Voltage Long Term Stability	(Note 4)	0.2					μV/mo Max
Input Bias Current		3	10 **20**		10 **20**		nA Max
Input Offset Current		2	10 **20**		10 **20**		nA Max
Input Noise Voltage	0.1 to 10Hz	0.08		0.18		0.18	μV p-p Max
Input Noise Voltage Density	f = 10Hz f = 30Hz f = 1kHz	3.5 3.1 3.0	5.5 4.5 3.8		5.5 4.5 3.8		nV/√Hz Max
Input Noise Current Density	f = 10Hz f = 30Hz f = 1kHz	1.7 1.0 0.4					pA/√Hz Max
Input Resistance	Common-Mode	20					GΩ
Input Voltage Range		±12	±11.5 **±10.5**		±11.5 **10.5**		V Min
Common-Mode Rejection Ratio	V_{CM} = ±11.5V **V_{CM} = ±10.5V**	140	126 **120**		126 **120**		dB Min
Power Supply Rejection Ratio	V_S = ±3.5V to ±18V	140	120 **117**		120 **117**		dB Min
Large-Signal Voltage Gain	V_O = ±12V $R_L \ge 2$ kΩ V_O = ±10V $R_L \ge 1$ kΩ $R_L \ge 600Ω$	10000 7000 6000	5000 **3000** 4000 **2000** 3000 **1500**		5000 **2000** 3500 **1500** 2000 **1000**		V/mV Min

Electrical Characteristics (Continued)

Parameter	Conditions	Typ	LM627AM LM637AM Tested Limit (Note 5)	LM627AM LM637AM Design Limit (Note 6)	LM627BM LM637BM Tested Limit (Note 5)	LM627BM LM637BM Design Limit (Note 6)	Units
Output Voltage Swing	$R_L \geq 2\ k\Omega$ $R_L \geq 600\Omega$	±13.8 ±12.5	±13 **±12.5** ±11 **±10.5**		±13 **±12.5** ±11 **±10.5**		V Min
Slew Rate	LM627 $R_L = 2k$ LM637	4.5 14		3 10		3 10	V/µs Min
Gain-Bandwidth Product	LM627 $f = 10\ kHz$ LM637	14 65		10 45		10 45	MHz Min
Output Resistance	Open Loop	50					Ω
Supply Current		3	4.5 **5.5**		4.5 **5.5**		mA Max
Offset Adjust Range	$R_P \geq 10\ k\Omega$	±2					mV

Electrical Characteristics

All limits guaranteed for $T_J = 25°C$, $V_{CM} = 0$, $V_O = 0$ and ±15V supplies unless otherwise specified. **Boldface limits apply at operating temperature extremes.**

Parameter	Conditions	Typ	LM627AC LM637AC Tested Limit (Note 5)	LM627AC LM637AC Design Limit (Note 6)	LM627BC LM637BC Tested Limit (Note 5)	LM627BC LM637BC Design Limit (Note 6)	LM627C LM637C Tested Limit (Note 5)	LM627C LM637C Design Limit (Note 6)	Units
Input Offset Voltage	(Note 2)	15	25 **50**		50 **110**		100	**210**	µV Max
Input Offset Voltage Drift	(Note 3)	0.2	**0.6**		**1.0**			**1.8**	µV/°C Max
Input Offset Voltage Long Term Stability	(Note 4)	0.2							µV/mo Max
Input Bias Current		3	10	**20**	10	**20**	25	**50**	nA Max
Input Offset Current		2	10	**20**	10	**20**	25	**50**	nA Max
Input Noise Voltage	0.1 to 10 Hz	0.08		0.18		0.18		0.25	µV p-p Max
Input Voltage Noise Density	f = 10 Hz f = 30 Hz f = 1 kHz	3.5 3.1 3.0	5.5 4.5 3.8		5.5 4.5 3.8			8.0 5.6 4.5	$nV/\sqrt{Hz}$ Max
Input Noise Current Density	f = 10 Hz f = 30 Hz f = 1 kHz	1.7 1.0 0.4							$pA/\sqrt{Hz}$ Max
Input Resistance	Common Mode	20							GΩ
Input Voltage Range		±12	±11.5	**±11**	±11.5	**±11**	±11.5	**±11**	V Min
Common-Mode Rejection Ratio	$V_{CM} = \pm 11.5V$ $\mathbf{V_{CM} = \pm 11V}$	140	126	**120**	126	**120**	120	**116**	dB Min
Power Supply Rejection Ratio	$V_S = \pm 3.5V$ to $\pm 18V$	140	120	**117**	120	**117**	110	**108**	dB Min

Electrical Characteristics (Continued)

Parameter	Conditions	Typ	LM627AC LM637AC Tested Limit (Note 5)	LM627AC LM637AC Design Limit (Note 6)	LM627BC LM637BC Tested Limit (Note 5)	LM627BC LM637BC Design Limit (Note 6)	LM627C LM637C Tested Limit (Note 5)	LM627C LM637C Design Limit (Note 6)	Units
Large-Signal Voltage Gain	$V_O = \pm 12V$ $R_L \geq 2\ k\Omega$	10000	5000	**3000**	5000	**3000**	4000	**2500**	V/mV Min
	$V_O = \pm 10V$ $R_L \geq 1\ k\Omega$	7000	4000	**2500**	3500	**2000**	2500	**1500**	
	$R_L \geq 600\Omega$	6000	3000	**2000**	2000	**1500**	1500	**1000**	
Output Voltage Swing	$R_L \geq 2\ k\Omega$	±13.8	±13	**±12.5**	±13	**±12.5**	±13	**±12.5**	V
	$R_L \geq 600\Omega$	±12.5	±11	**±10.5**	±11	**±10.5**	±10.5	**±10**	Min
Slew Rate	LM627 $R_L = 2k$	4.5		3		3		3	V/µs
	LM637	14		10		10		10	Min
Gain-Bandwidth Product	LM627 $f = 10\ kHz$	14		10		10		10	MHz
	LM637	65		45		45		45	Min
Output Resistance	Open Loop	50							Ω
Supply Current		3	4.5	**5**	4.5	**5**	4.8	**5.2**	mA Max
Offset Adjust Range	$R_P \geq 10\ k\Omega$	±2							mV

Note 1: Absolute maximum ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed.

Note 2: Input offset voltage for A and B grades is tested and guaranteed with the device fully warmed up. See *Figure 1* in the Application Hints for test circuit. Warmup drift is typically 5 µV settling out in 5 minutes. The LM627C/LM637C offset voltage is measured by automated test equipment within 200 ms of applying power.

Note 3: Input offset voltage drift is defined as (V_{OS}(85°C) − V_{OS}(−25°C))/110°C for the industrial temperature range. For the military temperature range, the input offset voltage drift is measured from room temperature to both extremes: both (V_{OS}(25°C) − V_{OS}(−55°C))/80°C and (V_{OS}(125°C) − V_{OS}(25°C))/100°C.

Note 4: Input offset voltage long term stability refers to the average trend line of V_{OS} vs. time over extended periods of time after the first 30 days of operation. Excluding the initial hour of operation, changes in V_{OS} during the first 30 days are typically 2 µV.

Note 5: Guaranteed and 100% production tested. These limits are used to calculate outgoing quality levels.

Note 6: Guaranteed but not 100% production tested. These limits are not used to calculate outgoing quality levels.

Note 7: Inputs are protected by back-to-back diodes to prevent zener breakdown of the input transistors. Series limiting resistors have not been included since they degrade noise performance. Excessive current may flow if a differential voltage in excess of 0.7V is applied.

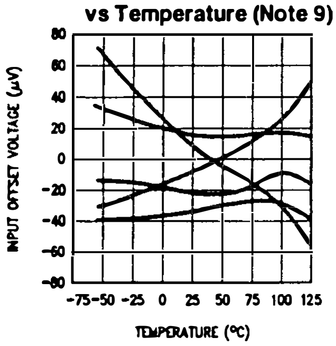
Note 8: For operation above 25°C, the maximum power dissipation specification must be derated. Typical junction-to-ambient thermal resistance of the molded DIP and the ceramic DIP are 95°C/W and 105°C/W respectively. The metal can package has a typical junction-to-ambient thermal resistance of 150°C/W and a typical junction-to-case thermal resistance of 17°C/W.

Note 9: These units selected to illustrate the types of variations that may be encountered. (This note refers to particular curves within the Typical Performance Characteristics.)

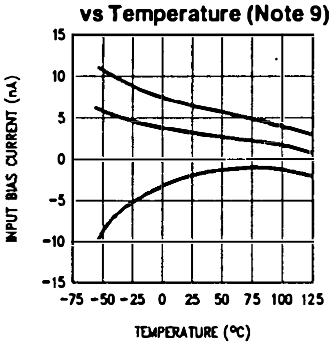
Typical DC Performance Characteristics (LM627, LM637)

$V_S = \pm 15V$, $T_A = 25°C$, $R_L = 2k$ unless othewise indicated.

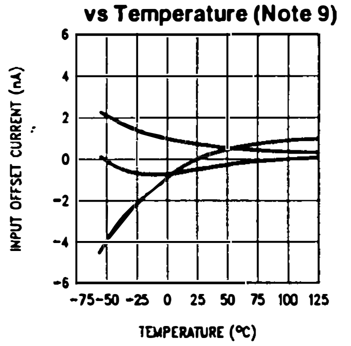
Input Offset Voltage of 5 Representative Units vs Temperature (Note 9)



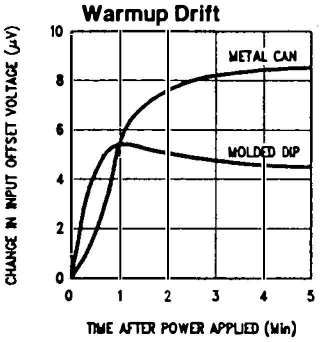
Input Bias Current of 3 Representative Units vs Temperature (Note 9)



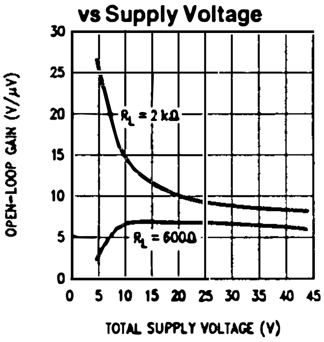
Input Offset Current of 3 Representative Units vs Temperature (Note 9)



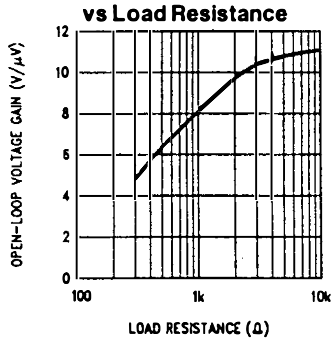
Warmup Drift



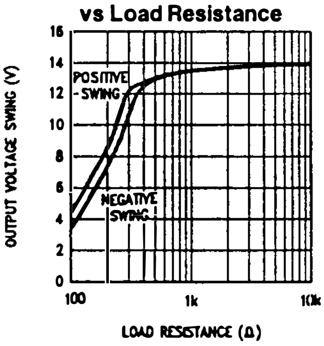
Open Loop Gain vs Supply Voltage



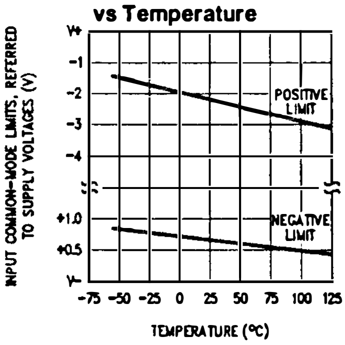
Open Loop Gain vs Load Resistance



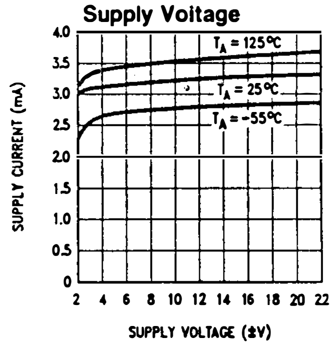
Output Voltage Swing vs Load Resistance



Input Common-Mode Limits vs Temperature



Supply Current vs Supply Voltage

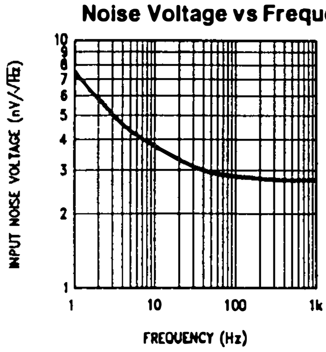


TL/H/9212-9

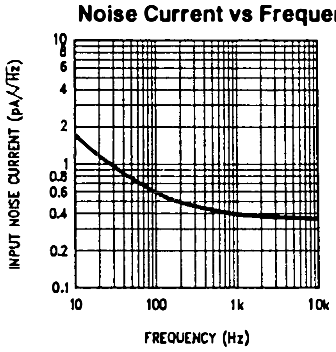
Typical Noise Characteristics (LM627, LM637)

$V_S = \pm 15V$, $T_A = 25°C$

Noise Voltage vs Frequency



Noise Current vs Frequency

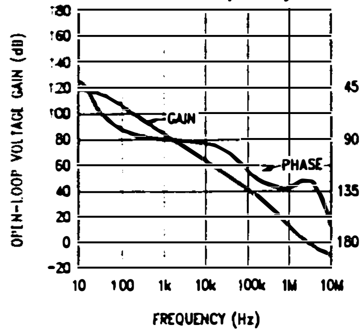


TL/H/9212-10

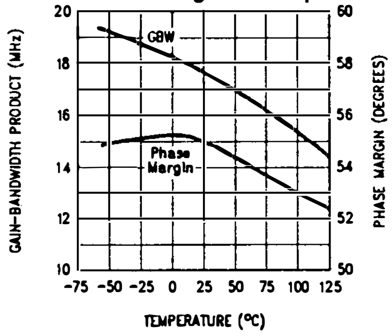
Typical AC Performance Characteristics (LM627)

$V_S = \pm 15V$, $T_A = 25°C$, $R_L = 2k$

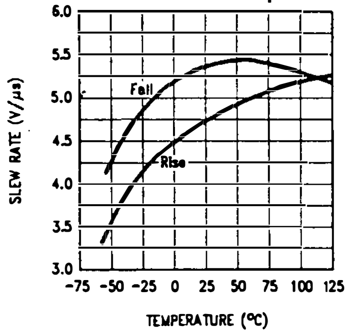
Open-Loop Gain and Phase vs Frequency



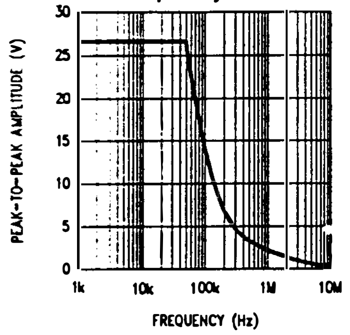
Gain-Bandwidth Product, Phase Margin vs Temperature



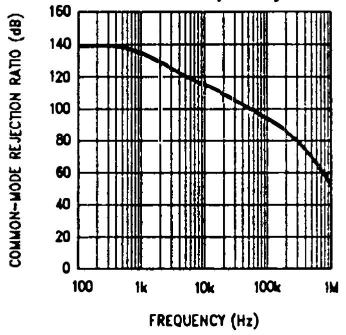
Slew Rate vs Temperature



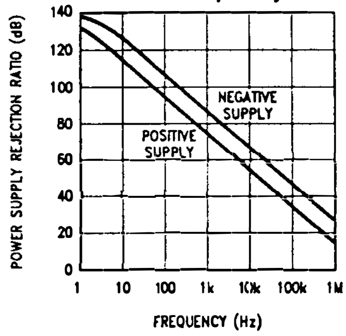
Maximum Output Swing vs Frequency



Common-Mode Rejection Ratio vs Frequency

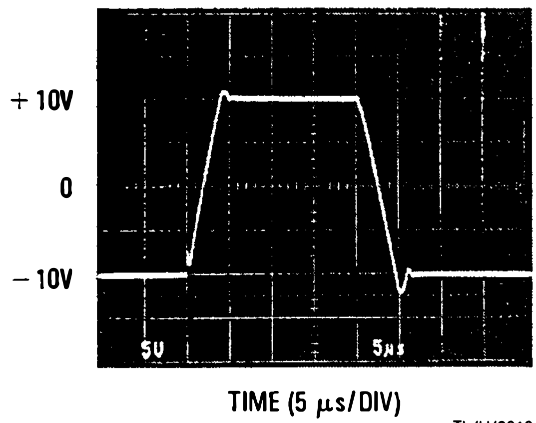


Power Supply Rejection Ratio vs Frequency



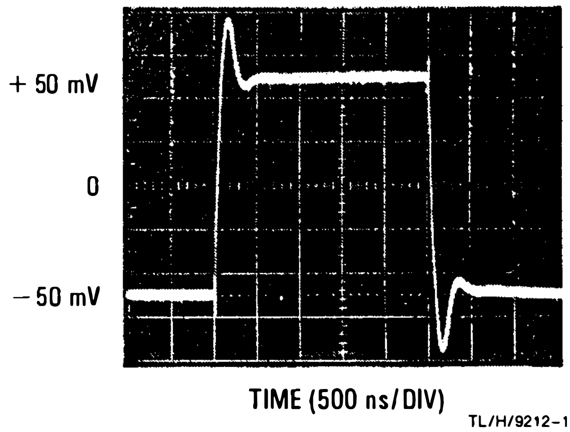
TL/H/9212-11

Large-Signal Pulse Response, $A_V = +1$



TL/H/9212-12

Small-Signal Pulse Response, $A_V = +1$

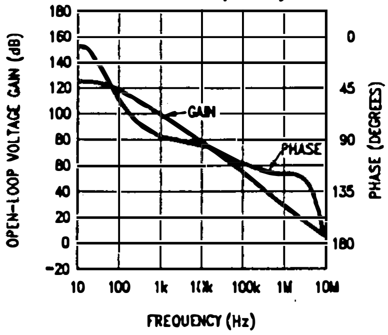


TL/H/9212-13

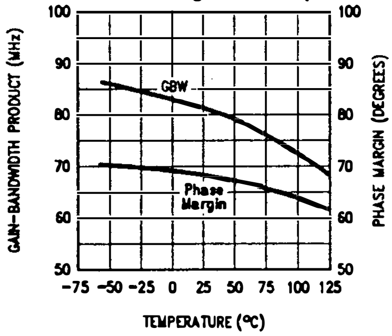
Typical AC Performance Characteristics (LM637)

$V_S = \pm 15V$, $T_A = 25°C$, $R_L = 2k$

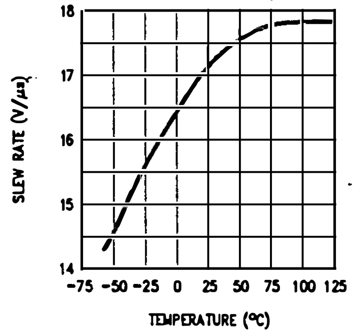
Open-Loop Gain and Phase vs Frequency



Gain-Bandwidth Product, Phase Margin vs Temperature

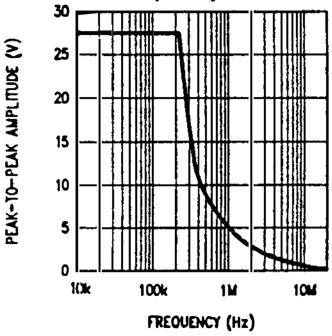


Slew Rate vs Temperature

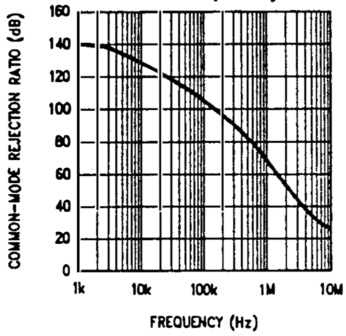


TL/H/9212–14

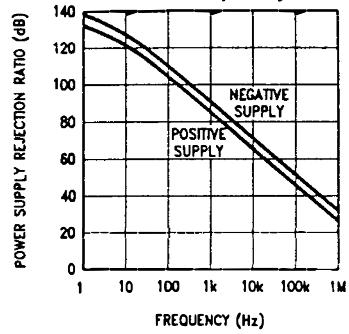
Maximum Output Swing vs Frequency



Common-Mode Rejection Ratio vs Frequency

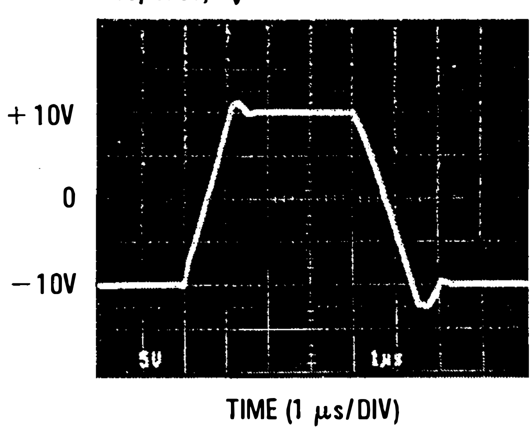


Power Supply Rejection Ratio vs Frequency



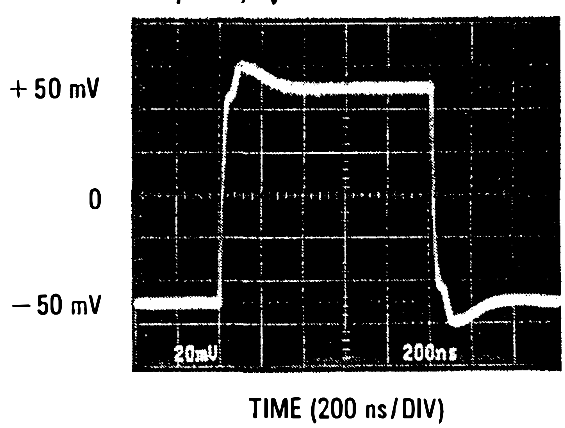
TL/H/9212–15

Large-Signal Pulse Response, $A_V = +5$



TL/H/9212–16

Small-Signal Pulse Response, $A_V = +5$



TL/H/9212–17

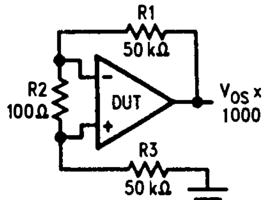
Application Hints

OFFSET VOLTAGE

Offset voltage of the LM627/637 is internally trimmed to a very low value. The data sheet V_{OS} specification applies at $T_J = 25°C$, $V_{CM} = 0$ and ±15V supplies. For other temperatures, common-mode voltages, and supply voltages, temperature drift, common-mode rejection and power-supply rejection errors must be taken into account.

Since the LM627/LM637C offset voltage is measured within 200 ms of applying power, the 5 μV typical warmup drift is not accounted for in the measurement. Fortunately, the warmup drift is a small fraction of its 100 μV max offset. For the 25 μV A and 50μV B grades, the offset voltage is measured with the circuit of *Figure 1* approximately 5 minutes after applying power.

To measure V_{OS} with high accuracy, V_{OS} must be amplified right at the device as shown; otherwise the offset voltage can be obscured by noise and thermoelectric voltages. Thermocouples occur in the devices, the IC socket and the resistor across the device inputs (R2), all of which must be held isothermal. Usually best results are obtained by placing the circuit in a box or chamber to minimize airflow and employing a long thermal soak time. R2 should be mounted symmetrically with respect to potential thermal gradients: e.g. *not* perpendicular to the board but instead parallel to the board and the device socket. In addition, R2 should have low thermal EMF. Cermet or nichrome metal film types are acceptable; avoid tin-oxide resistors.

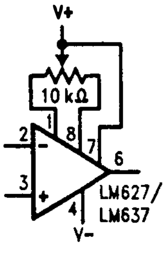


TL/H/9212-3

FIGURE 1. Offset Voltage Test Circuit

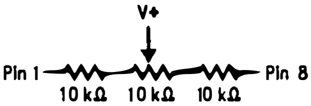
OFFSET NULLING

This is usually not required on the LM627/637 family since its offset voltage is internally trimmed. An offset adjust range of approximately ±2 mV is available using a single 10 or 20 kΩ potentiometer as shown in *Figure 2*. With these values, the adjustment is relatively linear over the entire range. If a 100 kΩ potentiomer is used, the adjustment becomes very coarse at the extremes (above 700 μV) but fine in the center, which makes it easier to precisely null the offset. For even more sensitivity, employ a pot in conjunction with two fixed resistors. The circuit of *Figure 3*, which uses this technique, has an adjustment range of ±200 μV. Because adjusting the offset voltage of an LM627/637 will alter its offset voltage temperature drift, caution is advised.



TL/H/9212-4

FIGURE 2. Offset Adjust Circuit



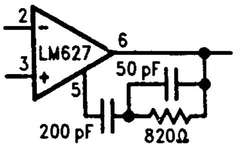
TL/H/9212-5

FIGURE 3. Improved Sensitivity Offset Adjust

Every 100μV of offset will produce a 0.33 μV/°C drift component. For this reason the offset adjust potentiometer should not be used to null out a sensor offset if system temperature drift is important; rather a stable voltage reference must be added to the sensor voltage. Offset voltage drift is guaranteed by design for the LM627C either with or without external nulling. The higher precision A and B grades are 100% drift tested and guaranteed without nulling only.

OVERCOMPENSATION

Without any external compensation, the LM627 is stable at unity gain and up to 500 pF load capacitance. It has a slew rate of 4.5 V/μs and a gain-bandwidth product of 14 MHz. If desired, the amplifier may be overcompensated by adding external components as shown in *Figure 4*. This increases maximum capacitive loading to 2000 pF while decreasing slew rate to 1.5 V/μs and bandwidth to 1.5 MHz. If overcompensation of the LM627 (or the LM637) is not desired, pin 5 should be left open.

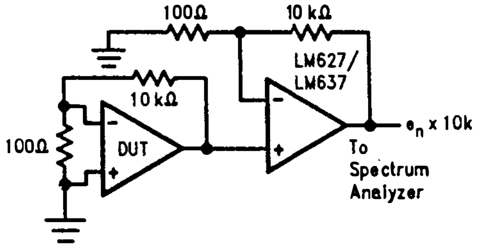


TL/H/9212-6

FIGURE 4. Overcompensation

NOISE

When measuring spot noise voltage, a circuit as shown in *Figure 5* is recommended. The DUT running at a gain of 100 will not roll off until approximately 140 kHz. Adding the second gain of 100 amplifier brings total DUT-input-referred gain up to 10,000, which minimizes to minimize sensitivity to EMI in the environment. When measuring spot noise at 30 Hz, it is recommended that the spectrum analyzer bandwidth be 20 Hz or less to minimize pickup at line frequency.



TL/H/9212-7

FIGURE 5. Spot Noise Test Circuit

National Semiconductor

LM741/LM741A/LM741C/LM741E Operational Amplifier

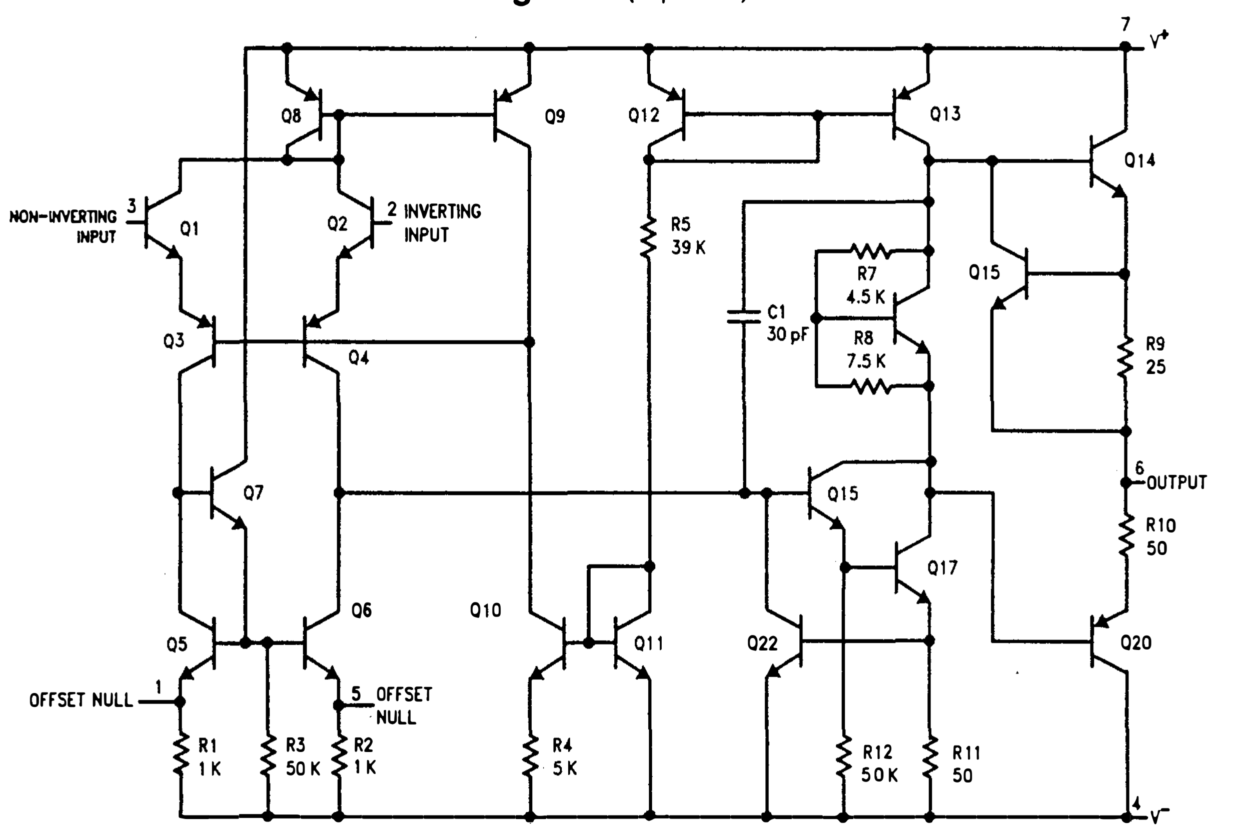
General Description

The LM741 series are general purpose operational amplifiers which feature improved performance over industry standards like the LM709. They are direct, plug-in replacements for the 709C, LM201, MC1439 and 748 in most applications.

The amplifiers offer many features which make their application nearly foolproof: overload protection on the input and output, no latch-up when the common mode range is exceeded, as well as freedom from oscillations.

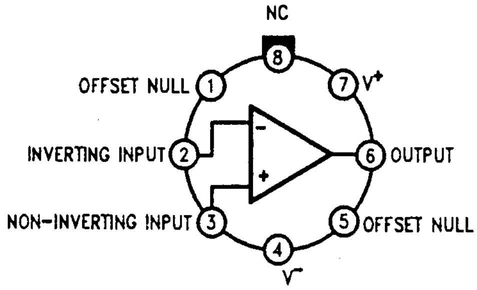
The LM741C/LM741E are identical to the LM741/LM741A except that the LM741C/LM741E have their performance guaranteed over a 0°C to +70°C temperature range, instead of −55°C to +125°C.

Schematic and Connection Diagrams (Top Views)



TL/H/9341-1

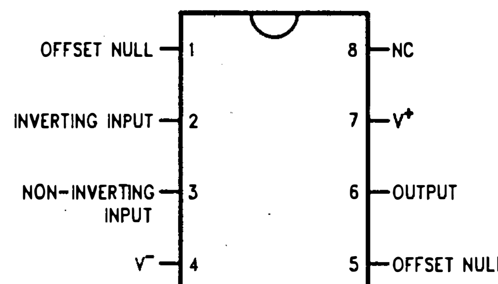
Metal Can Package



TL/H/9341-2

Order Number LM741H, LM741AH, LM741CH or LM741EH
See NS Package Number H08C

Dual-In-Line or S.O. Package



TL/H/9341-3

Order Number LM741J, LM741AJ, LM741CJ, LM741CM, LM741CN or LM741EN
See NS Package Number J08A, M08A or N08E

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

(Note 5)

	LM741A	LM741E	LM741	LM741C
Supply Voltage	±22V	±22V	±22V	±18V
Power Dissipation (Note 1)	500 mW	500 mW	500 mW	500 mW
Differential Input Voltage	±30V	±30V	±30V	±30V
Input Voltage (Note 2)	±15V	±15V	±15V	±15V
Output Short Circuit Duration	Continuous	Continuous	Continuous	Continuous
Operating Temperature Range	−55°C to +125°C	0°C to +70°C	−55°C to +125°C	0°C to +70°C
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C	−65°C to +150°C	−65°C to +150°C
Junction Temperature	150°C	100°C	150°C	100°C
Soldering Information				
N-Package (10 seconds)	260°C	260°C	260°C	260°C
J- or H-Package (10 seconds)	300°C	300°C	300°C	300°C
M-Package				
Vapor Phase (60 seconds)	215°C	215°C	215°C	215°C
Infrared (15 seconds)	215°C	215°C	215°C	215°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

ESD Tolerance (Note 6)	400V	400V	400V	400V

Electrical Characteristics (Note 3)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	$T_A = 25°C$										
	$R_S \leq 10 k\Omega$					1.0	5.0		2.0	6.0	mV
	$R_S \leq 50\Omega$		0.8	3.0							mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										
	$R_S \leq 50\Omega$			4.0							mV
	$R_S \leq 10 k\Omega$						6.0			7.5	mV
Average Input Offset Voltage Drift				15							μV/°C
Input Offset Voltage Adjustment Range	$T_A = 25°C, V_S = \pm 20V$	±10				±15			±15		mV
Input Offset Current	$T_A = 25°C$		3.0	30		20	200		20	200	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			70		85	500			300	nA
Average Input Offset Current Drift				0.5							nA/°C
Input Bias Current	$T_A = 25°C$		30	80		80	500		80	500	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			0.210			1.5			0.8	μA
Input Resistance	$T_A = 25°C, V_S = \pm 20V$	1.0	6.0		0.3	2.0		0.3	2.0		MΩ
	$T_{AMIN} \leq T_A \leq T_{AMAX}$, $V_S = \pm 20V$	0.5									MΩ
Input Voltage Range	$T_A = 25°C$							±12	±13		V
	$T_{AMIN} \leq T_A \leq T_{AMAX}$				±12	±13					V
Large Signal Voltage Gain	$T_A = 25°C, R_L \geq 2 k\Omega$										
	$V_S = \pm 20V, V_O = \pm 15V$	50									V/mV
	$V_S = \pm 15V, V_O = \pm 10V$				50	200		20	200		V/mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$, $R_L \geq 2 k\Omega$,										
	$V_S = \pm 20V, V_O = \pm 15V$	32									V/mV
	$V_S = \pm 15V, V_O = \pm 10V$				25			15			V/mV
	$V_S = \pm 5V, V_O = \pm 2V$	10									V/mV

Electrical Characteristics (Note 3) (Continued)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Output Voltage Swing	$V_S = \pm 20V$										
	$R_L \geq 10\,k\Omega$	±16									V
	$R_L \geq 2\,k\Omega$	±15									V
	$V_S = \pm 15V$										
	$R_L \geq 10\,k\Omega$				±12	±14		±12	±14		V
	$R_L \geq 2\,k\Omega$				±10	±13		±10	±13		V
Output Short Circuit Current	$T_A = 25°C$	10	25	35		25			25		mA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$	10		40							mA
Common-Mode Rejection Ratio	$T_{AMIN} \leq T_A \leq T_{AMAX}$										
	$R_S \leq 10\,k\Omega$, $V_{CM} = \pm 12V$				70	90		70	90		dB
	$R_S \leq 50\Omega$, $V_{CM} = \pm 12V$	80	95								dB
Supply Voltage Rejection Ratio	$T_{AMIN} \leq T_A \leq T_{AMAX}$,										
	$V_S = \pm 20V$ to $V_S = \pm 5V$										
	$R_S \leq 50\Omega$	86	96								dB
	$R_S \leq 10\,k\Omega$				77	96		77	96		dB
Transient Response	$T_A = 25°C$, Unity Gain										
Rise Time			0.25	0.8		0.3			0.3		µs
Overshoot			6.0	20		5			5		%
Bandwidth (Note 4)	$T_A = 25°C$	0.437	1.5								MHz
Slew Rate	$T_A = 25°C$, Unity Gain	0.3	0.7			0.5			0.5		V/µs
Supply Current	$T_A = 25°C$					1.7	2.8		1.7	2.8	mA
Power Consumption	$T_A = 25°C$										
	$V_S = \pm 20V$		80	150							mW
	$V_S = \pm 15V$					50	85		50	85	mW
LM741A	$V_S = \pm 20V$										
	$T_A = T_{AMIN}$			165							mW
	$T_A = T_{AMAX}$			135							mW
LM741E	$V_S = \pm 20V$										
	$T_A = T_{AMIN}$			150							mW
	$T_A = T_{AMAX}$			150							mW
LM741	$V_S = \pm 15V$										
	$T_A = T_{AMIN}$					60	100				mW
	$T_A = T_{AMAX}$					45	75				mW

Note 1: For operation at elevated temperatures, these devices must be derated based on thermal resistance, and T_j max. (listed under "Absolute Maximum Ratings"). $T_j = T_A + (\theta_{jA} P_D)$.

Thermal Resistance	Cerdip (J)	DIP (N)	HO8 (H)	SO-8 (M)
θ_{jA} (Junction to Ambient)	100°C/W	100°C/W	170°C/W	195°C/W
θ_{jC} (Junction to Case)	N/A	N/A	25°C/W	N/A

Note 2: For supply voltages less than ±15V, the absolute maximum input voltage is equal to the supply voltage.

Note 3: Unless otherwise specified, these specifications apply for $V_S = \pm 15V$, $-55°C \leq T_A \leq +125°C$ (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to $0°C \leq T_A \leq +70°C$.

Note 4: Calculated value from: BW (MHz) = 0.35/Rise Time(µs).

Note 5: For military specifications see RETS741X for LM741 and RETS741AX for LM741A.

Note 6: Human body model, 1.5 kΩ in series with 100 pF.

Appendix B

A table of Thermocouple Power Series is shown below (TC LTD, 1997, p. 9)

POWER SERIES EXPANSIONS AND POLYNOMIALS:

For computer applications the following expressions are given for the commonly used thermocouple conductor combinations.
Resultant errors from their use will be less than the last significant digit as per the thermocouple reference tables included in this publication.

K

Temperature range −270°C to 0°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 3.945\,012\,802\,5 \times 10^{1}$
$a_2 = 2.362\,237\,358\,6 \times 10^{-2}$
$a_3 = -3.285\,890\,678\,4 \times 10^{-4}$
$a_4 = -4.990\,482\,877\,7 \times 10^{-6}$
$a_5 = -6.750\,905\,917\,3 \times 10^{-8}$
$a_6 = -5.741\,032\,742\,8 \times 10^{-10}$
$a_7 = -3.108\,887\,289\,4 \times 10^{-12}$
$a_8 = -1.045\,160\,936\,5 \times 10^{-14}$
$a_9 = -1.988\,926\,687\,8 \times 10^{-17}$
$a_{10} = -1.632\,269\,748\,6 \times 10^{-20}$

0°C to 1372°C

$$E = b_0 + \sum_{i=1}^{n} b_i (t_{90})^i + c_0 \exp[c_1 (t_{90} - 126.9686)^2] \ \mu V$$

where

$b_0 = -1.760\,041\,368\,6 \times 10^{1}$
$b_1 = 3.892\,120\,497\,5 \times 10^{1}$
$b_2 = 1.855\,877\,003\,2 \times 10^{-2}$
$b_3 = -9.945\,759\,287\,4 \times 10^{-5}$
$b_4 = 3.184\,094\,571\,9 \times 10^{-7}$
$b_5 = -5.607\,284\,488\,9 \times 10^{-10}$
$b_6 = 5.607\,505\,905\,9 \times 10^{-13}$
$b_7 = -3.202\,072\,000\,3 \times 10^{-16}$
$b_8 = 9.715\,114\,715\,2 \times 10^{-20}$
$b_9 = -1.210\,472\,127\,5 \times 10^{-23}$

$c_0 = 1.185\,976 \times 10^{2}$
$c_1 = -1.183\,432 \times 10^{-4}$

T

Temperature range −270°C to 0°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 3.874\,810\,636\,4 \times 10^{1}$
$a_2 = 4.419\,443\,434\,7 \times 10^{-2}$
$a_3 = 1.184\,432\,310\,5 \times 10^{-4}$
$a_4 = 2.003\,297\,355\,4 \times 10^{-5}$
$a_5 = 9.013\,801\,955\,9 \times 10^{-7}$
$a_6 = 2.265\,115\,659\,3 \times 10^{-8}$
$a_7 = 3.607\,115\,420\,5 \times 10^{-10}$
$a_8 = 3.849\,393\,988\,3 \times 10^{-12}$
$a_9 = 2.821\,352\,192\,5 \times 10^{-14}$
$a_{10} = 1.425\,159\,477\,9 \times 10^{-16}$
$a_{11} = 4.876\,866\,228\,6 \times 10^{-19}$
$a_{12} = 1.079\,553\,927\,6 \times 10^{-21}$
$a_{13} = 1.394\,502\,706\,2 \times 10^{-24}$
$a_{14} = 7.979\,515\,392\,7 \times 10^{-28}$

0°C to 400°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 3.874\,810\,636\,4 \times 10^{1}$
$a_2 = 3.329\,222\,788\,0 \times 10^{-2}$
$a_3 = 2.061\,824\,340\,4 \times 10^{-4}$
$a_4 = -2.188\,225\,684\,6 \times 10^{-6}$
$a_5 = 1.099\,688\,092\,8 \times 10^{-8}$
$a_6 = -3.081\,575\,877\,2 \times 10^{-11}$
$a_7 = 4.547\,913\,529\,0 \times 10^{-14}$
$a_8 = -2.751\,290\,167\,3 \times 10^{-17}$

J

Temperature range −210°C to 760°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 5.038\,118\,781\,5 \times 10^{1}$
$a_2 = 3.047\,583\,693\,0 \times 10^{-2}$
$a_3 = -8.568\,106\,572\,0 \times 10^{-5}$
$a_4 = 1.322\,819\,529\,5 \times 10^{-7}$
$a_5 = -1.705\,295\,833\,7 \times 10^{-10}$
$a_6 = 2.094\,809\,069\,7 \times 10^{-13}$
$a_7 = -1.253\,839\,533\,6 \times 10^{-16}$
$a_8 = 1.563\,172\,569\,7 \times 10^{-20}$

760°C to 1200°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = 2.964\,562\,568\,1 \times 10^{5}$
$a_1 = -1.497\,612\,778\,6 \times 10^{3}$
$a_2 = 3.178\,710\,392\,4$
$a_3 = -3.184\,768\,670\,1 \times 10^{-3}$
$a_4 = 1.572\,081\,900\,4 \times 10^{-6}$
$a_5 = -3.069\,136\,905\,6 \times 10^{-10}$

N

Temperature range −270°C to 0°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 2.615\,910\,596\,2 \times 10^{1}$
$a_2 = 1.095\,748\,422\,8 \times 10^{-2}$
$a_3 = -9.384\,111\,155\,4 \times 10^{-5}$
$a_4 = -4.641\,203\,975\,9 \times 10^{-8}$
$a_5 = -2.630\,335\,771\,6 \times 10^{-9}$
$a_6 = -2.265\,343\,800\,3 \times 10^{-11}$
$a_7 = -7.608\,930\,079\,1 \times 10^{-14}$
$a_8 = -9.341\,966\,783\,5 \times 10^{-17}$

0°C to 1300°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 2.592\,939\,460\,1 \times 10^{1}$
$a_2 = 1.571\,014\,188\,0 \times 10^{-2}$
$a_3 = 4.382\,562\,723\,7 \times 10^{-5}$
$a_4 = -2.526\,116\,979\,4 \times 10^{-7}$
$a_5 = 6.431\,181\,933\,9 \times 10^{-10}$
$a_6 = -1.006\,347\,151\,9 \times 10^{-12}$
$a_7 = 9.974\,533\,899\,2 \times 10^{-16}$
$a_8 = -6.086\,324\,560\,7 \times 10^{-19}$
$a_9 = 2.084\,922\,933\,9 \times 10^{-22}$
$a_{10} = -3.068\,219\,615\,1 \times 10^{-26}$

E

Temperature range −270°C to 0°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 5.866\,550\,870\,8 \times 10^{1}$
$a_2 = 4.541\,097\,712\,4 \times 10^{-2}$
$a_3 = -7.799\,804\,868\,6 \times 10^{-4}$
$a_4 = -2.580\,016\,084\,3 \times 10^{-5}$
$a_5 = -5.945\,258\,305\,7 \times 10^{-7}$
$a_6 = -9.321\,405\,866\,7 \times 10^{-9}$
$a_7 = -1.028\,760\,553\,4 \times 10^{-10}$
$a_8 = -8.037\,012\,362\,1 \times 10^{-13}$
$a_9 = -4.397\,949\,739\,1 \times 10^{-15}$
$a_{10} = -1.641\,477\,635\,5 \times 10^{-17}$
$a_{11} = -3.967\,361\,951\,6 \times 10^{-20}$
$a_{12} = -5.582\,732\,872\,1 \times 10^{-23}$
$a_{13} = -3.465\,784\,201\,3 \times 10^{-26}$

0°C to 1000°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 5.866\,550\,871\,0 \times 10^{1}$
$a_2 = 4.503\,227\,558\,2 \times 10^{-2}$
$a_3 = 2.890\,840\,721\,2 \times 10^{-5}$
$a_4 = -3.305\,689\,665\,2 \times 10^{-7}$
$a_5 = 6.502\,440\,327\,8 \times 10^{-10}$
$a_6 = -1.919\,749\,550\,4 \times 10^{-12}$
$a_7 = -1.253\,660\,049\,7 \times 10^{-15}$
$a_8 = 2.148\,921\,756\,9 \times 10^{-18}$
$a_9 = -1.438\,804\,178\,2 \times 10^{-21}$
$a_{10} = 3.596\,089\,948\,1 \times 10^{-25}$

R

Temperature range −50°C to 1064.18°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 5.289\,617\,297\,65$
$a_2 = 1.391\,665\,897\,82 \times 10^{-2}$
$a_3 = -2.388\,556\,930\,17 \times 10^{-5}$
$a_4 = 3.569\,160\,010\,63 \times 10^{-8}$
$a_5 = -4.623\,476\,662\,98 \times 10^{-11}$
$a_6 = 5.007\,774\,410\,34 \times 10^{-14}$
$a_7 = -3.731\,058\,861\,91 \times 10^{-17}$
$a_8 = 1.577\,164\,823\,67 \times 10^{-20}$
$a_9 = -2.810\,386\,252\,51 \times 10^{-24}$

1064.18°C to 1664.5°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = 2.951\,579\,253\,16 \times 10^{3}$
$a_1 = -2.520\,612\,513\,32$
$a_2 = 1.595\,645\,018\,65 \times 10^{-2}$
$a_3 = -7.640\,859\,475\,76 \times 10^{-6}$
$a_4 = 2.053\,052\,918\,24 \times 10^{-9}$
$a_5 = -2.933\,596\,681\,73 \times 10^{-13}$

1664.5°C to 1768.1°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = 1.522\,321\,182\,09 \times 10^{5}$
$a_1 = -2.688\,198\,885\,45 \times 10^{2}$
$a_2 = 1.712\,802\,804\,71 \times 10^{-1}$
$a_3 = -3.458\,957\,064\,53 \times 10^{-5}$
$a_4 = -9.346\,339\,710\,46 \times 10^{-12}$

S

Temperature range −50°C to 1064.18°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = 5.403\,133\,086\,31$
$a_2 = 1.259\,342\,897\,40 \times 10^{-2}$
$a_3 = -2.324\,779\,686\,89 \times 10^{-5}$
$a_4 = 3.220\,288\,230\,36 \times 10^{-8}$
$a_5 = -3.314\,651\,963\,89 \times 10^{-11}$
$a_6 = 2.557\,442\,517\,86 \times 10^{-14}$
$a_7 = -1.250\,688\,713\,93 \times 10^{-17}$
$a_8 = 2.714\,431\,761\,45 \times 10^{-21}$

1064.18°C to 1664.5°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = 1.329\,004\,440\,85 \times 10^{3}$
$a_1 = 3.345\,093\,113\,44$
$a_2 = 6.548\,051\,928\,18 \times 10^{-3}$
$a_3 = -1.648\,562\,592\,09 \times 10^{-6}$
$a_4 = 1.299\,896\,051\,74 \times 10^{-11}$

1664.5°C to 1768.1°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = 1.466\,282\,326\,36 \times 10^{5}$
$a_1 = -2.584\,305\,167\,52 \times 10^{2}$
$a_2 = 1.636\,935\,746\,41 \times 10^{-1}$
$a_3 = -3.304\,390\,469\,87 \times 10^{-5}$
$a_4 = -9.432\,236\,906\,12 \times 10^{-12}$

B

Temperature range 0°C to 630.615°C

$$E = \sum_{i=1}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_1 = -2.465\,081\,834\,6 \times 10^{-1}$
$a_2 = 5.904\,042\,117\,1 \times 10^{-3}$
$a_3 = -1.325\,793\,163\,6 \times 10^{-6}$
$a_4 = 1.566\,829\,190\,1 \times 10^{-9}$
$a_5 = -1.694\,452\,924\,0 \times 10^{-12}$
$a_6 = 6.299\,034\,709\,4 \times 10^{-16}$

630.615°C to 1820°C

$$E = \sum_{i=0}^{n} a_i (t_{90})^i \ \mu V$$

where

$a_0 = -3.893\,816\,862\,1 \times 10^{3}$
$a_1 = 2.857\,174\,747\,0 \times 10^{1}$
$a_2 = -8.488\,510\,478\,5 \times 10^{-2}$
$a_3 = 1.578\,528\,016\,4 \times 10^{-4}$
$a_4 = -1.683\,534\,486\,4 \times 10^{-7}$
$a_5 = 1.110\,979\,401\,3 \times 10^{-10}$
$a_6 = -4.451\,543\,103\,3 \times 10^{-14}$
$a_7 = 9.897\,564\,082\,1 \times 10^{-18}$
$a_8 = -9.379\,133\,028\,9 \times 10^{-22}$

Appendix C
MCH68C11E9 Features

- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- 12 Kbytes of ON-Chip ROM
- 512 Bytes of On-Chip EEPROM with Block Protect for Extra Security
- 512 Bytes of ON-Chip RAM (All Saved During Standby)
- 16-Bit Timer System
  - Four Output Compare Channels
  - Three Input Capture Channels
  - One Input Capture or Output Compare Channel (software Selectable)
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System
- Synchronous Serial Peripheral Interface (SPI)
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
- 8-Channel 8-Bit Analog-to-Digital (A/D) Converter
- 38 General-Purpose Input/Output (I/O) Pins
  - 16 Bidirectional I/O Pins
  - 11 Input-Only pins, 11 Output-Only Pins

- Available in a 52-Pin Plastic Leaded Chip Carrier (PLCC) and 64-Pin Quad Flat Pack (QFP)

HC11 MC68HC11E9 Technical Data (1991, p..1-1)

Choosing the Resistor tolerances for the 12 Volt Battery Interface

Resistor	Value	+5%	-5%	+1%	-1%
R1	1000Ω	1050Ω	950Ω	1010Ω	990Ω
R2	2200Ω	2310Ω	2090Ω	2222Ω	2178Ω

$$\text{Scale Factor} = \frac{R_1}{R_1 + R_2}$$

<u>Case 1</u> $R_1 = +5\%$, $R_2 = -5\%$

$$= \frac{1050}{1050 + 2090} = 0.3344$$

<u>Case 2</u> $R_1 = -5\%$, $R_2 = +5\%$

$$= \frac{950}{950 + 2310} = 0.2914$$

<u>Case 3</u> $R_1 = +1\%$, $R_2 = -1\%$

$$= \frac{1010}{1010 + 2178} = 0.3168$$

<u>Case 4</u> $R_1 = -1\%$, $R_2 = +1\%$

$$\frac{990}{990 + 2222} = 0.3082$$

Worst Case Error at 16 Volts

Case 1 $\dfrac{16 \times 0.3344}{0.3125} = 17.1213$ => Error 1.12 V

Case 2 $\dfrac{16 \times 0.2914}{0.3125} = 14.9100$ => Error 1.09 V

Case 3 $\dfrac{16 \times 0.3168}{0.3125} = 16.2202$ => Error 0.22 V

Case 4 $\dfrac{16 \times 0.3082}{0.3125} = 15.7798$ => Error 0.22 V

Appendix E

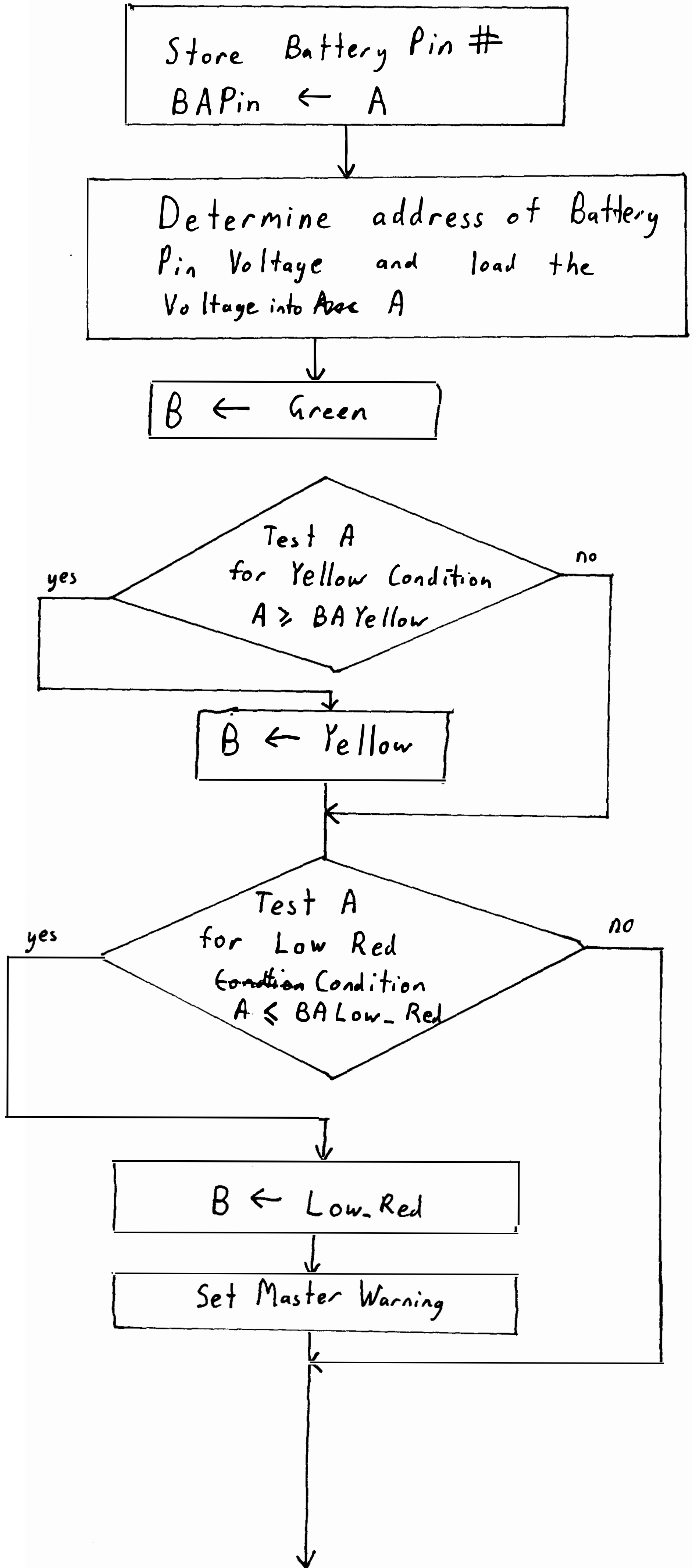
This Appendix contains the algorithms, in flow diagram format, for;

- Battery_Volts
- KType
- LM335_Temp
- RPM Procedure

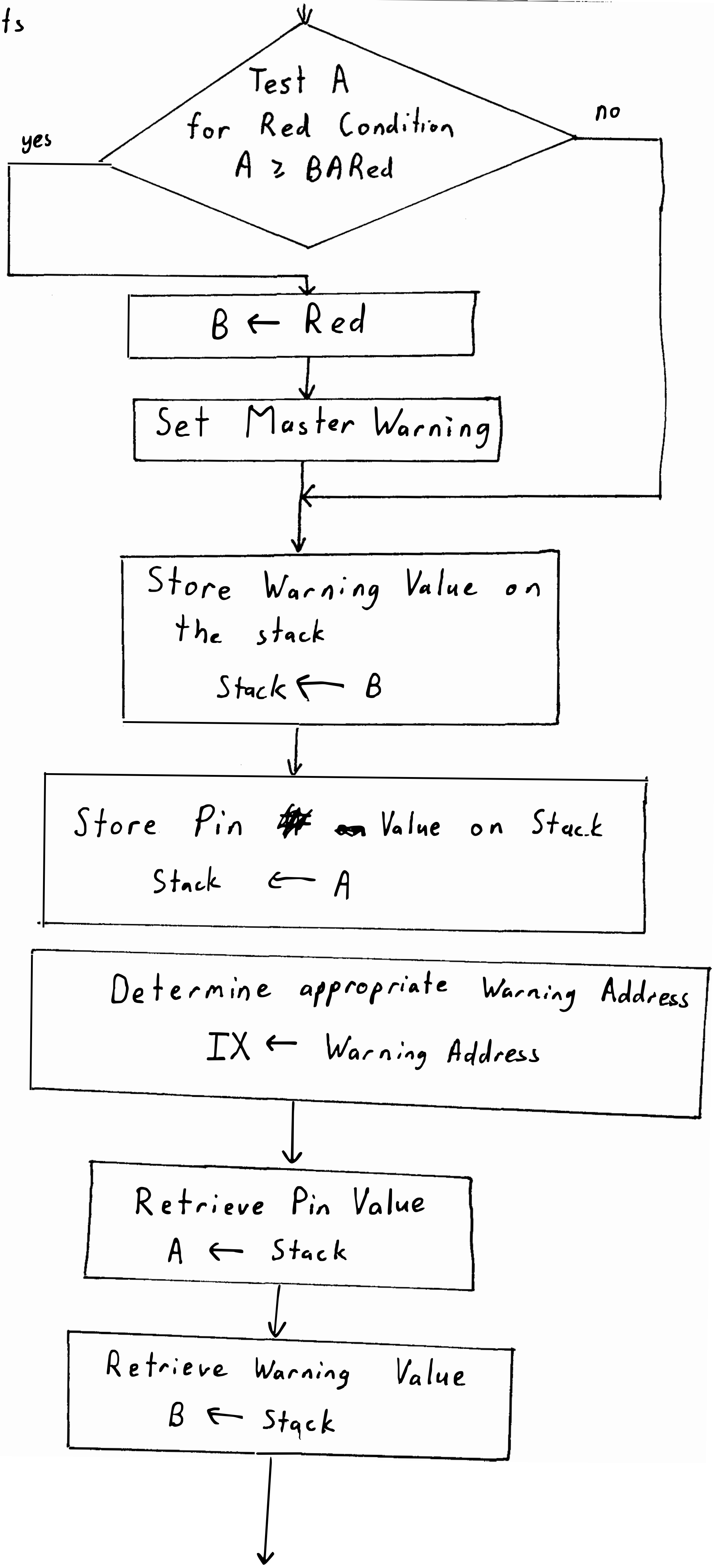
attery_ Volts

gorithm

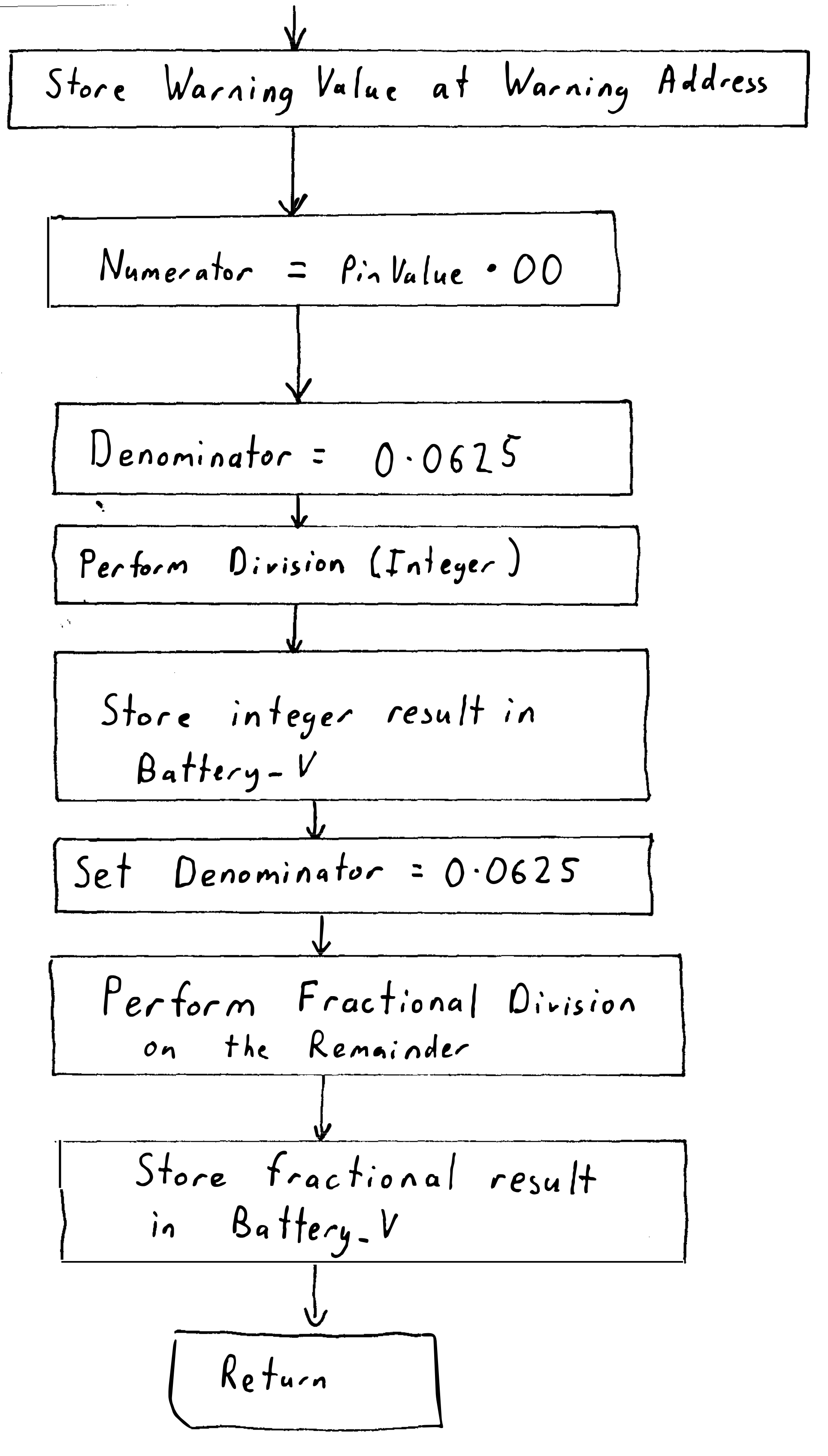
e I

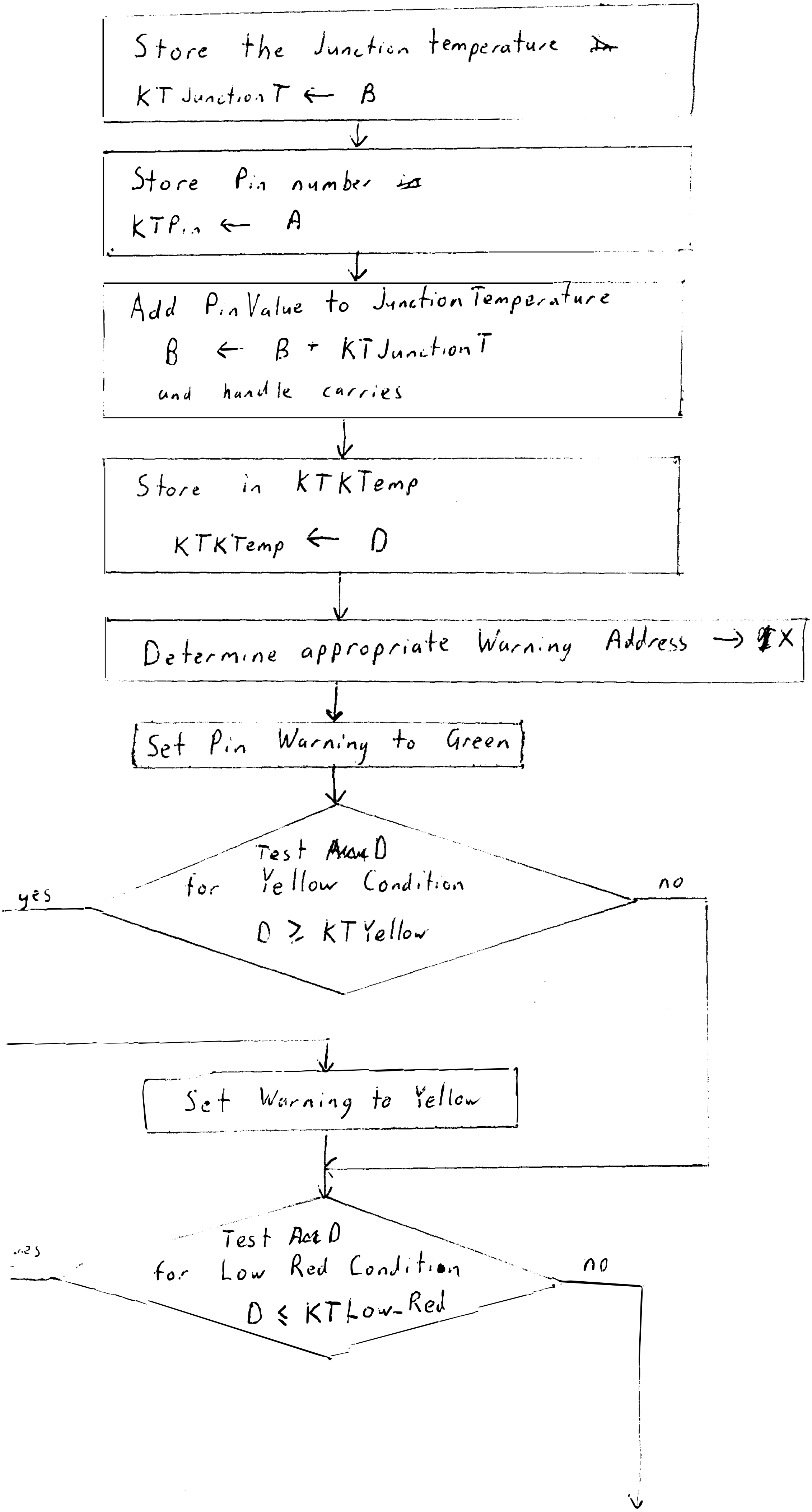


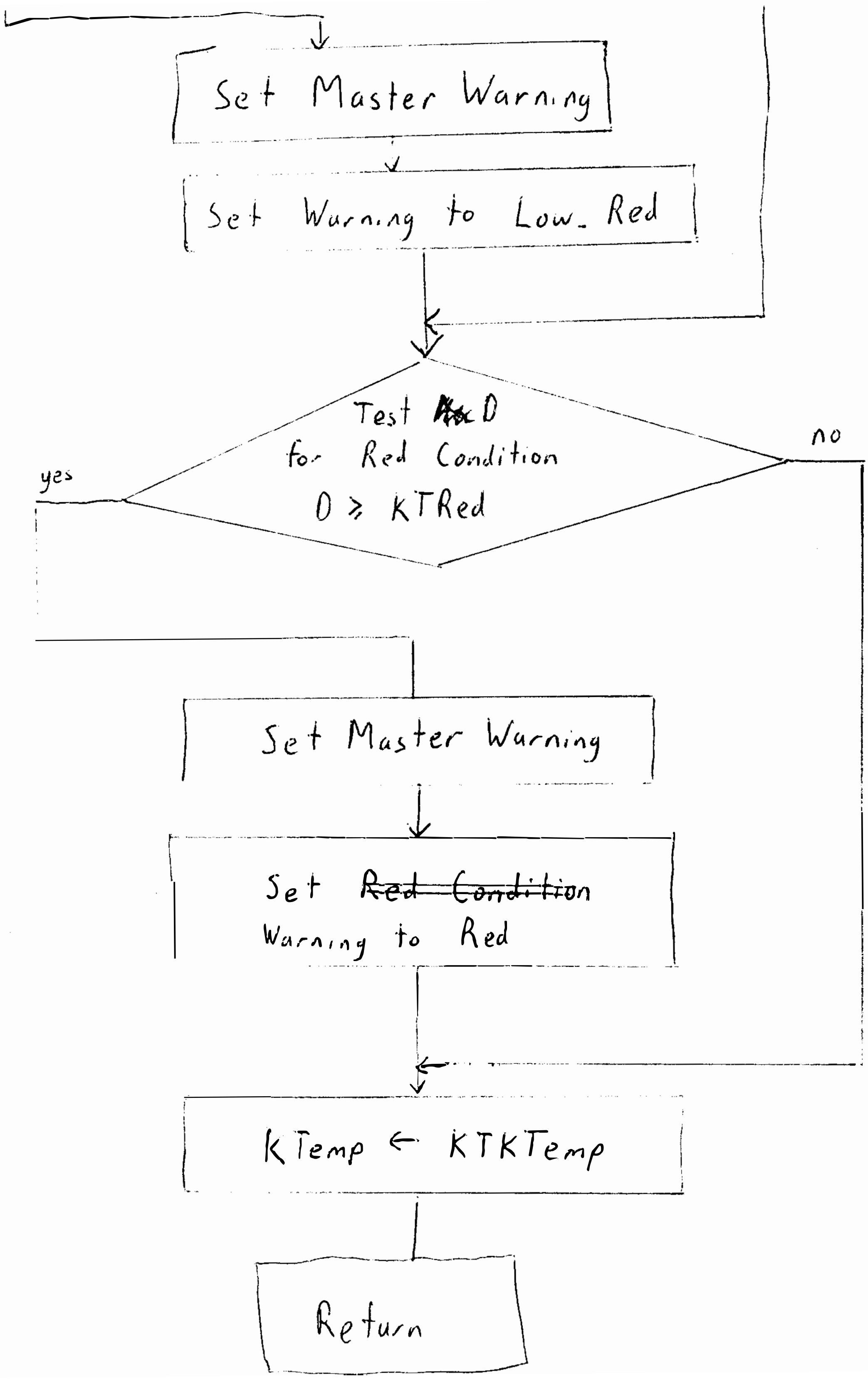
ttery_Volts
orithm
e 2



ter_Volts
rithm
e :

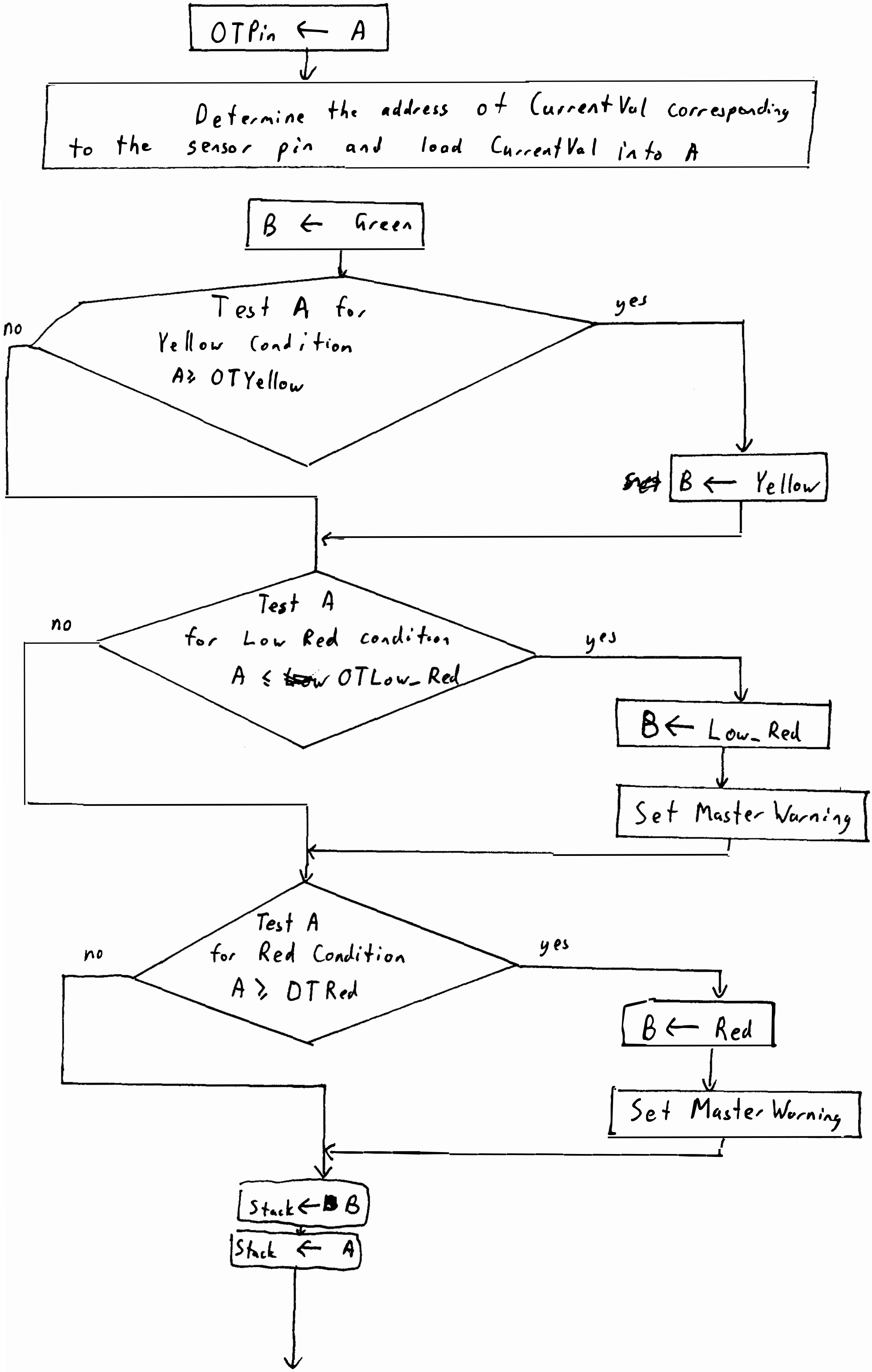



Store the Junction temperature
KTJunctionT ← B
Store Pin number
KTPin ← A
Add Pin Value to Junction Temperature
B ← B + KTJunctionT
and handle carries
Store in KTKTemp
KTKTemp ← D
Determine appropriate Warning Address → X
Set Pin Warning to Green
Test D
for Yellow Condition
D ≥ KTYellow
yes
no
Set Warning to Yellow
Test D
for Low Red Condition
D ≤ KTLow-Red
no

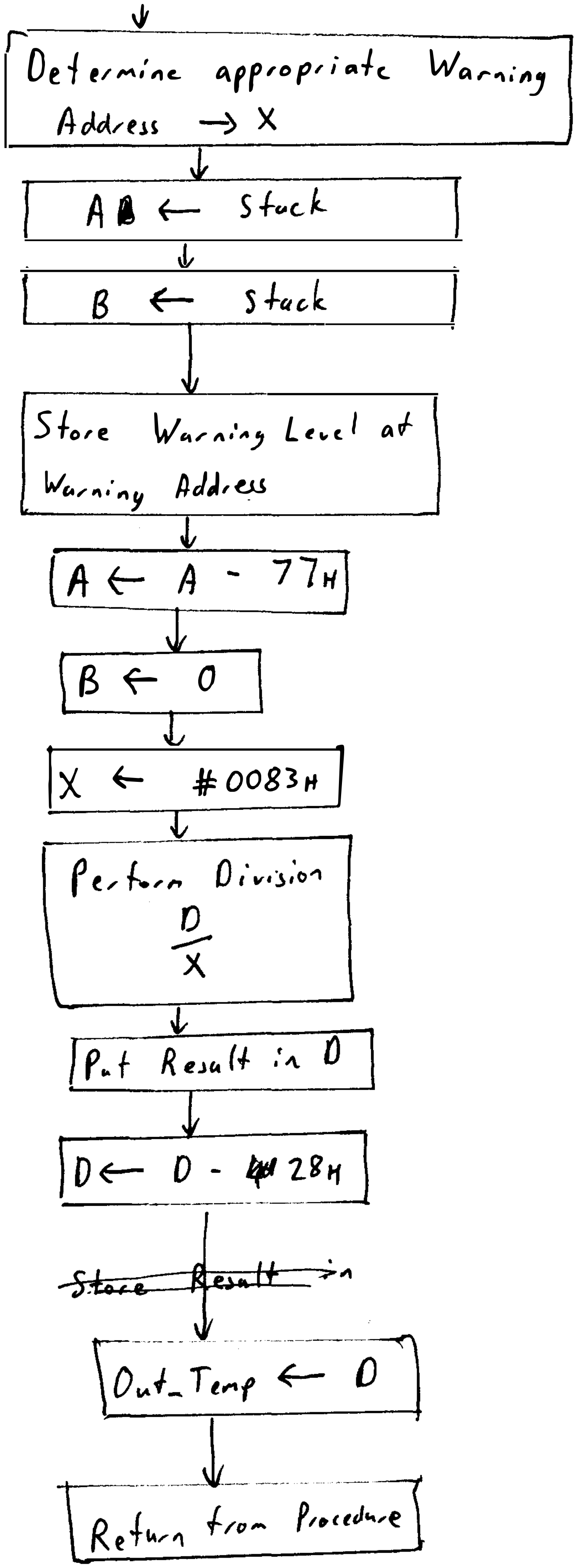

Set Master Warning
Set Warning to Low_Red
Test D
for Red Condition
D ≥ KTRed
yes
no
Set Master Warning
Set Red Condition
Warning to Red
KTemp ← KTKTemp
Return

S_Temp

1

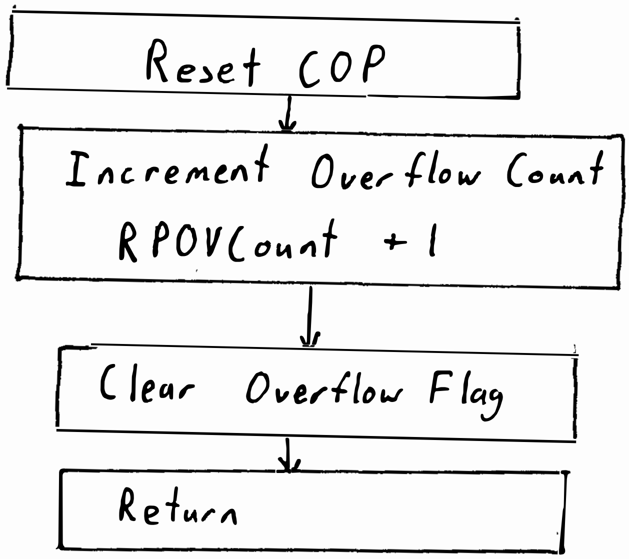

OTPin ← A
Determine the address of CurrentVal corresponding to the sensor pin and load CurrentVal into A
B ← Green
Test A for Yellow Condition A ≥ OTYellow
no
yes
B ← Yellow
Test A for Low Red condition A ≤ OTLow_Red
no
yes
B ← Low_Red
Set Master Warning
Test A for Red Condition A ≥ OTRed
no
yes
B ← Red
Set Master Warning
Stack ← B
Stack ← A

Page 2

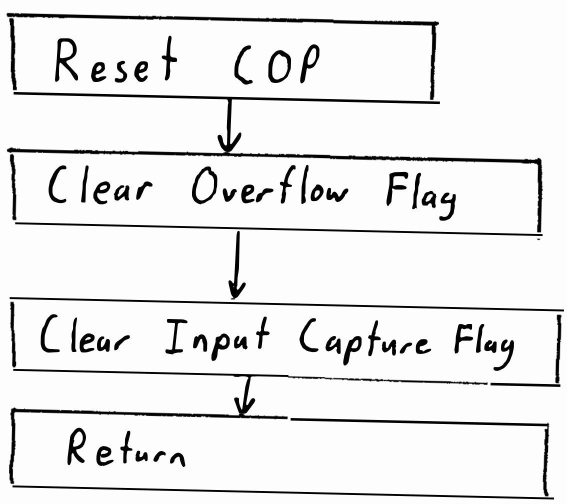

Determine appropriate Warning Address → X
AB ← Stack
B ← Stack
Store Warning Level at Warning Address
A ← A - 77H
B ← 0
X ← #0083H
Perform Division D/X
Put Result in D
D ← D - 28H
Store Result in
Out_Temp ← D
Return from Procedure

Interrupt Routines

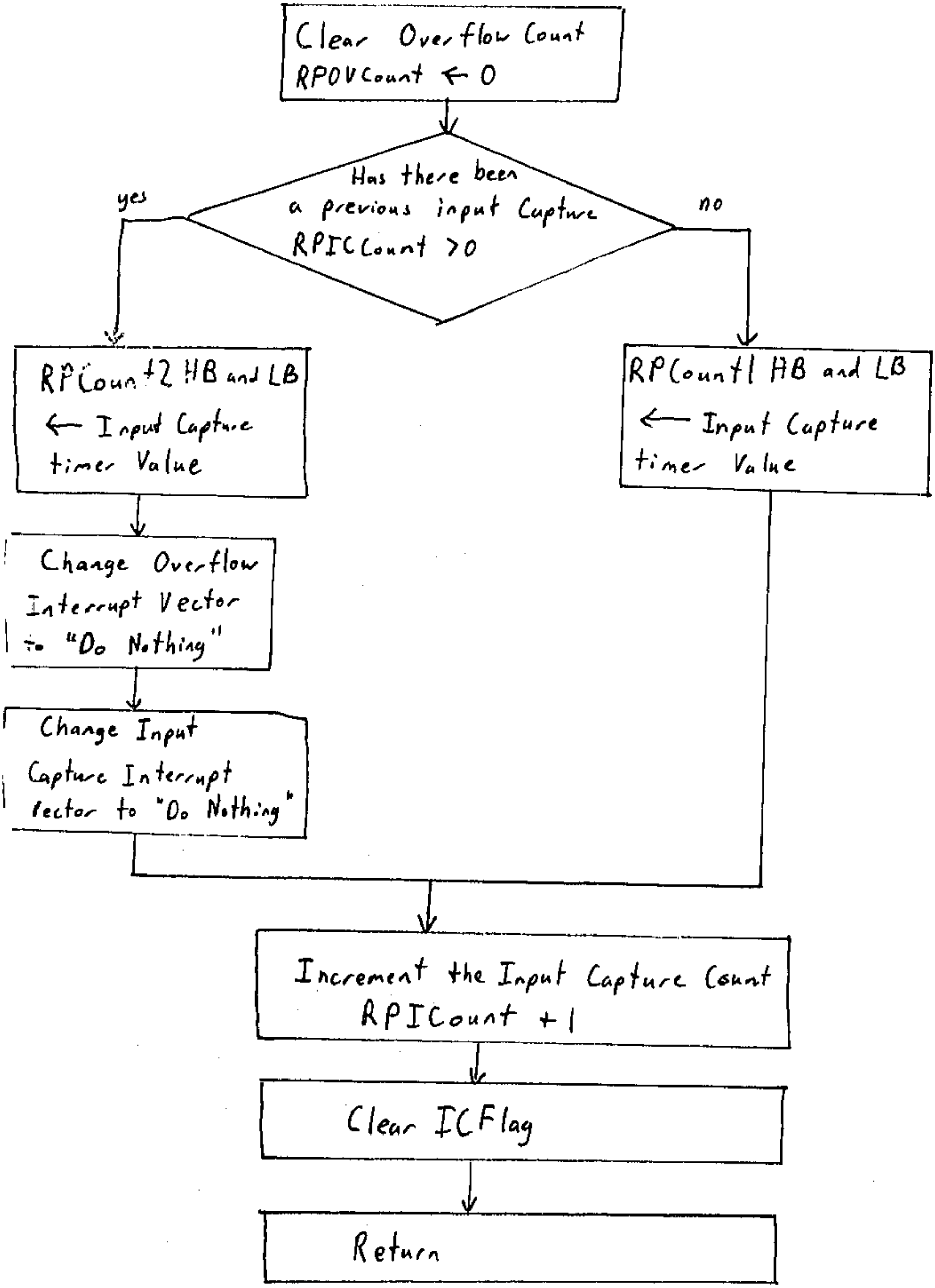
Handle Overflow



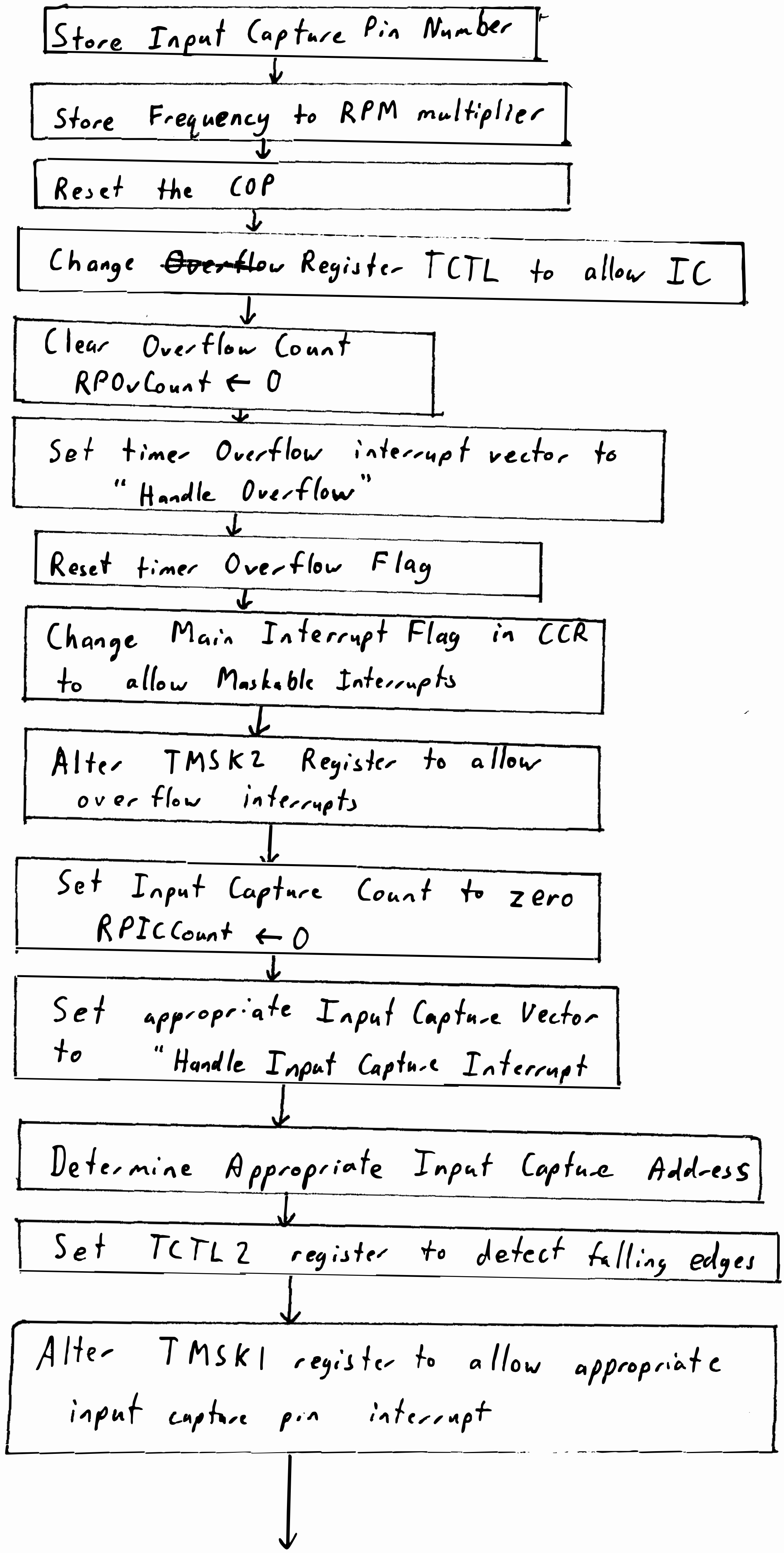
Do Nothing



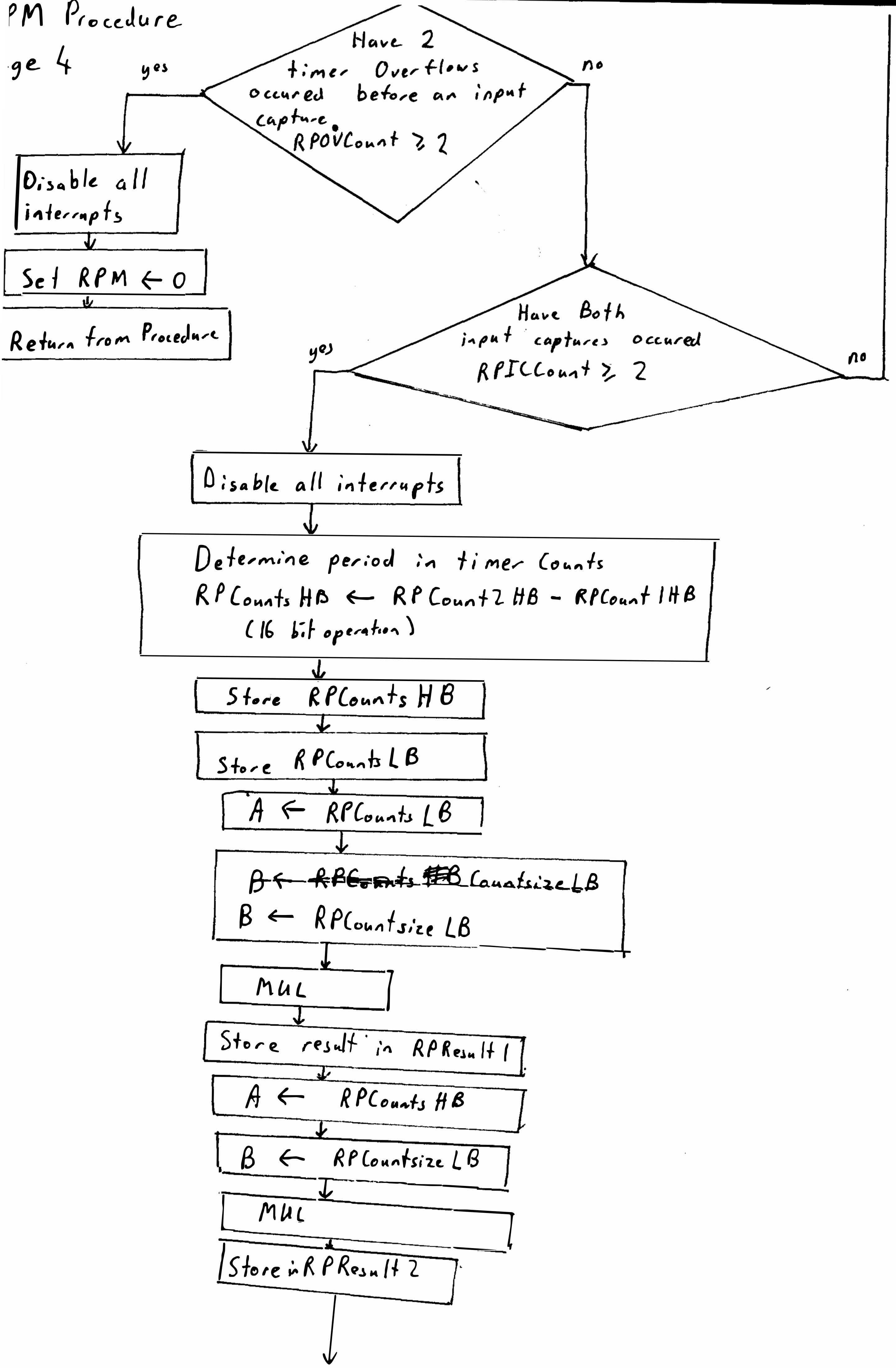
Handle Input Capture

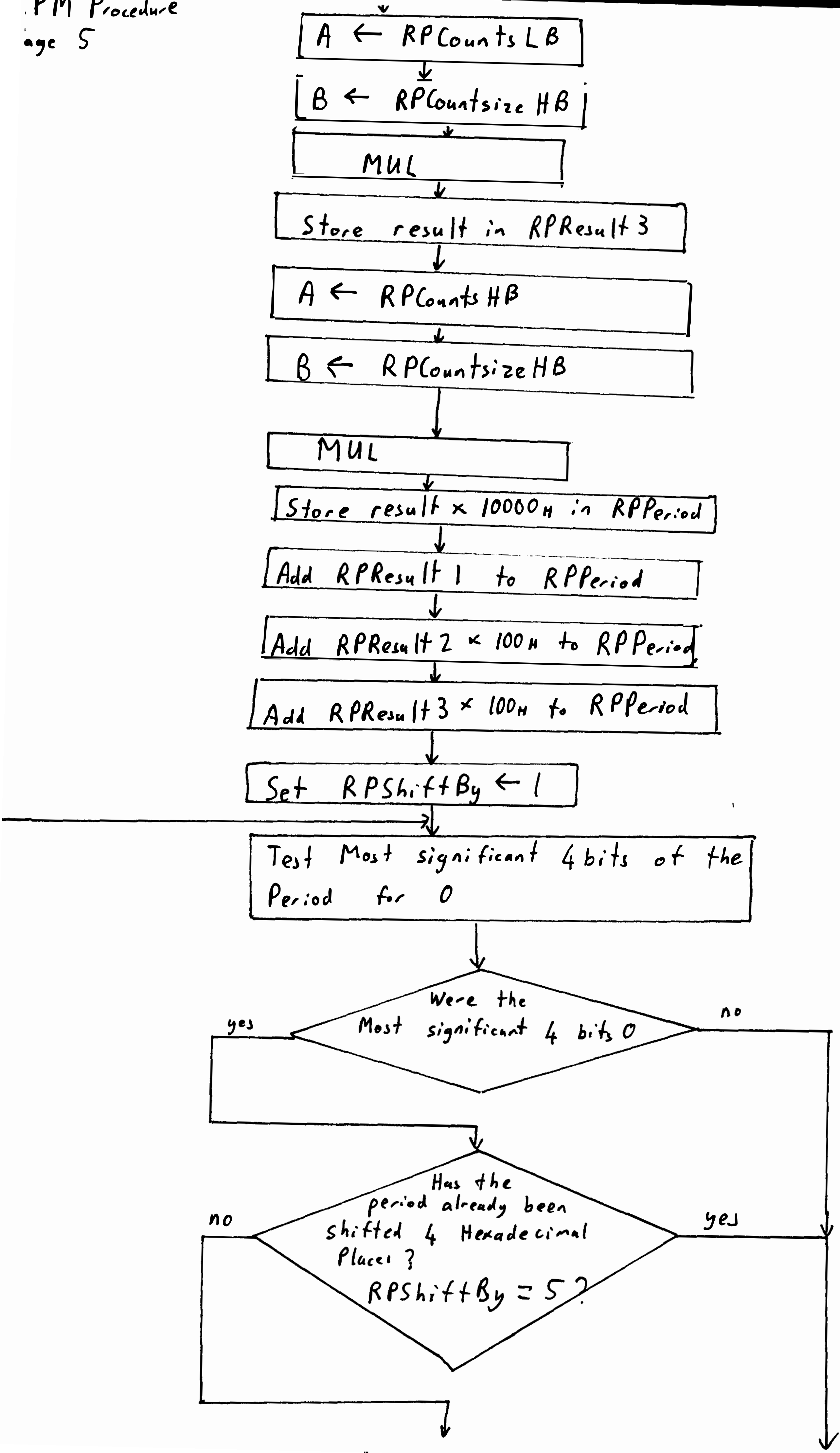


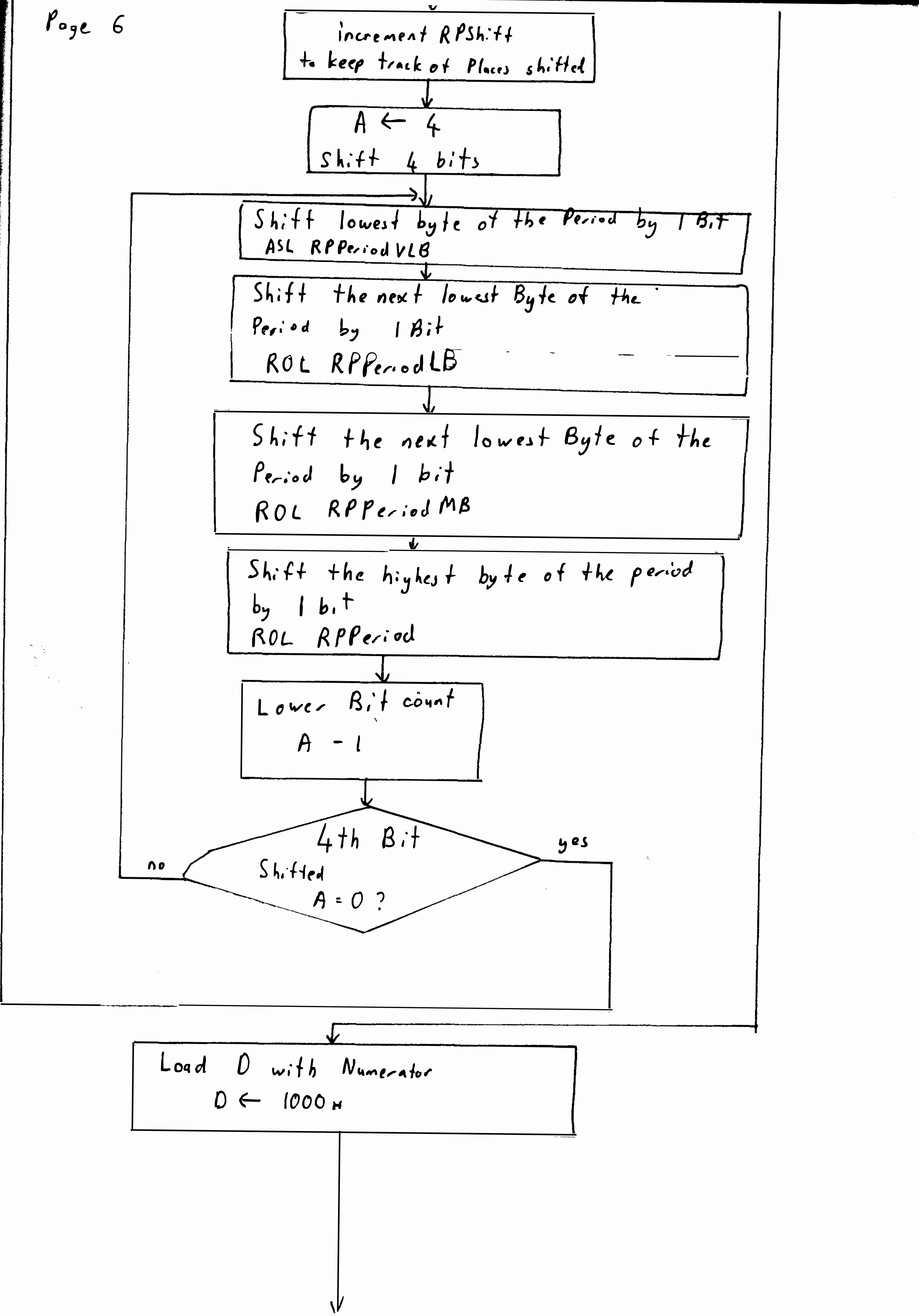
Procedure ~~Cod~~ Algorithm

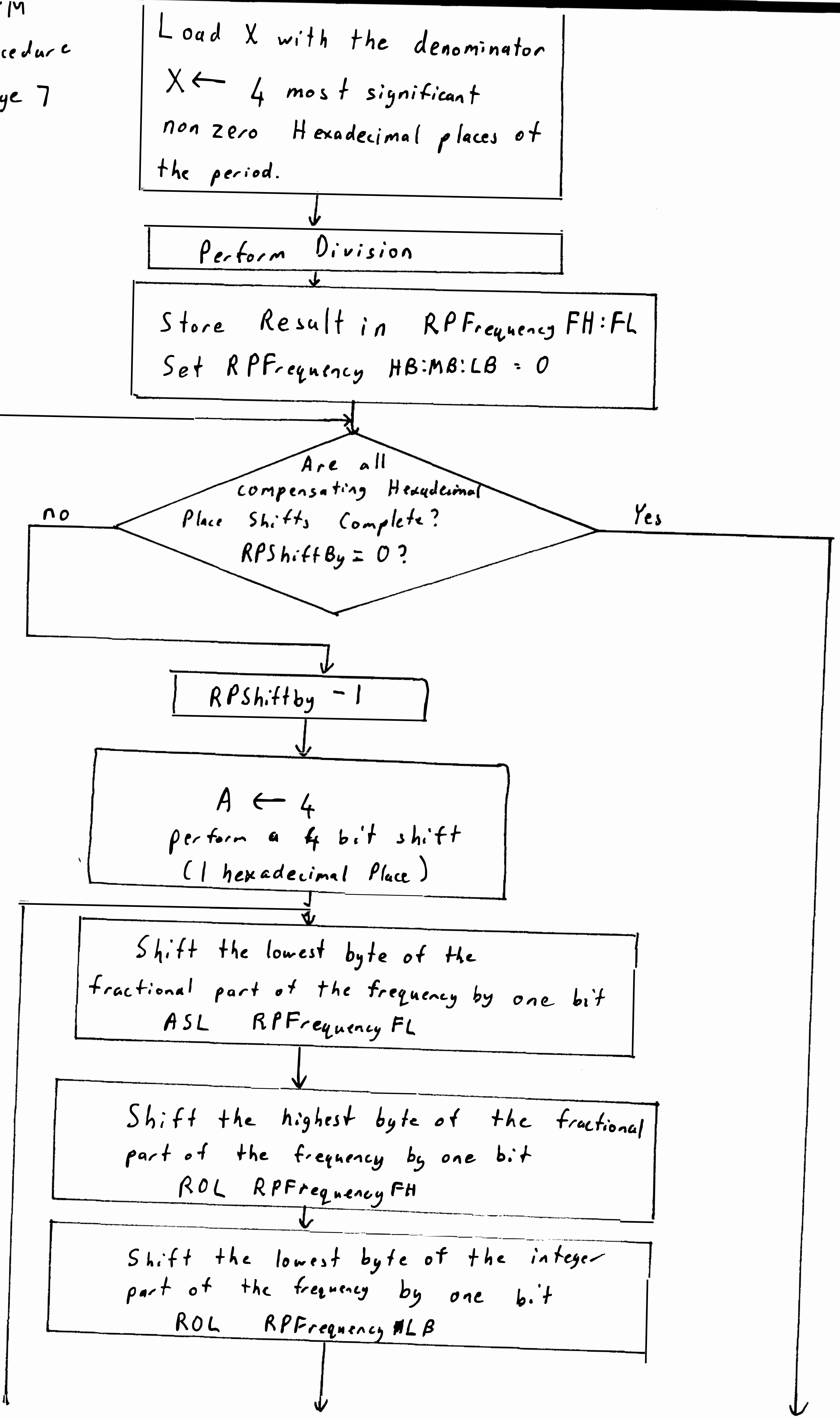


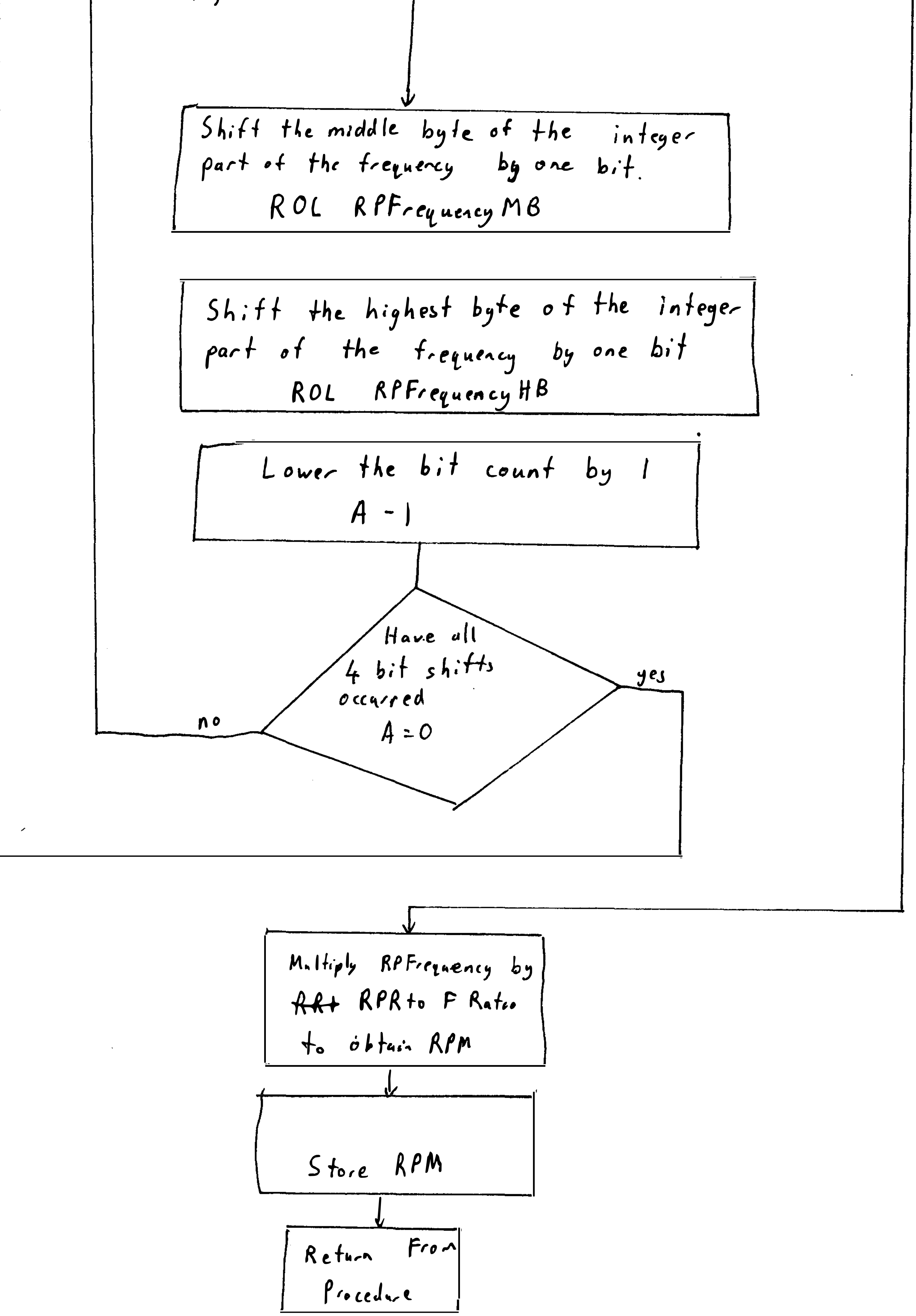
PM Procedure
ge 4




A ← RPCounts LB
B ← RPCountsize HB
MUL
Store result in RPResult 3
A ← RPCounts HB
B ← RPCountsize HB
MUL
Store result × 10000H in RPPeriod
Add RPResult 1 to RPPeriod
Add RPResult 2 × 100H to RPPeriod
Add RPResult 3 × 100H to RPPeriod
Set RPShiftBy ← 1
Test Most significant 4 bits of the Period for 0
Were the Most significant 4 bits 0
yes
no
Has the period already been shifted 4 Hexadecimal Places ?
RPShiftBy = 5 ?
no
yes


increment RPShift
to keep track of Places shifted
A ← 4
Shift 4 bits
Shift lowest byte of the Period by 1 Bit
ASL RPPeriodVLB
Shift the next lowest Byte of the Period by 1 Bit
ROL RPPeriodLB
Shift the next lowest Byte of the Period by 1 bit
ROL RPPeriodMB
Shift the highest byte of the period by 1 bit
ROL RPPeriod
Lower Bit count
A - 1
4th Bit Shifted
A = 0 ?
no
yes
Load D with Numerator
D ← 1000H


Load X with the denominator
X ← 4 most significant
non zero Hexadecimal places of
the period.
Perform Division
Store Result in RPFrequency FH:FL
Set RPFrequency HB:MB:LB = 0
Are all
compensating Hexadecimal
Place Shifts Complete?
RPShiftBy = 0?
no
Yes
RPShiftby -1
A ← 4
perform a 4 bit shift
(1 hexadecimal Place)
Shift the lowest byte of the
fractional part of the frequency by one bit
ASL RPFrequency FL
Shift the highest byte of the fractional
part of the frequency by one bit
ROL RPFrequency FH
Shift the lowest byte of the integer
part of the frequency by one bit
ROL RPFrequency LB


Shift the middle byte of the integer part of the frequency by one bit.
ROL RPFrequencyMB
Shift the highest byte of the integer part of the frequency by one bit
ROL RPFrequencyHB
Lower the bit count by 1
A - 1
Have all 4 bit shifts occurred
A = 0
yes
no
Multiply RPFrequency by ~~RRt~~ RPRto F Ratio to obtain RPM
Store RPM
Return From Procedure

Appendix F

This Appendix contains all of the system Code. There are 4 separate parts and these are;

Version 4.2
Version 4 subroutines
Get Count (The first half of the RPM procedure)
Counts to Frequency (The second half of the RPM procedure)

Version 4.2

CHANGES IN VERSION 4.2
This program still handles the LM335 on pin 2
Battery on pin 3
Ktype circuit on pin 4
The change is that the LED's are driven by the LM335 system

CHANGES IN VERSION 4.1
This program is identical to version 3.1 except that it uses ver4sub.asm
This makes it possible to boot into the program straight after a reset. By switching a jumper on the EVBU.

CHANGES IN VERSION 3.1
Version 3.1 now includes a routine for handling the Ktype thermocouple input on A/D input 4. It still handles the LM335 and the battery circuit input
The master warning LED is controlled by all 3 inputs
The other LEDs are driven only by the Ktype thermocouple input.

VERSION 3.0 NOTES
Version 3-0 is functionally equivilent to Version 2-1.asm The main difference between the two is that version 3-0 requires ver3sub to be in the EEPROM starting at B600. This frees up more RAM for developing new sensor routines.

Version 3 Handles an LM335 sensor connected to A/D pin 2 and;
a battery voltage circuit input at A/D pin 3
The master warning LED value is dependant on both the LM335 and the Battery voltage
The Other LEDs are driven dependant on the warning level generated from the battery voltage circuit

PLEASE NOTE
No stack area has been defined in this program. This is because the EVBU has a predefined stack area. If this program is to be used in real applications a stack area will need to be defined.

```
               *Data for the program
               *Last Value read of the AD input pins is stored in CurrentVal
               *00 to 07.  This is the offset
urrentVal EQU $00
               *Value of each analog input pin after last read
urrentVal0 EQU $00
urrentVal1 EQU $01
urrentVal2 EQU $02
urrentVal3 EQU $03
urrentVal4 EQU $04
urrentVal5 EQU $05
urrentVal6 EQU $06
urrentVal7 EQU $07
               *
               * Defining location of Master warning
aster_Warning EQU 08
               * Defining location of other Warnings
               * Current addresses go from 09 to 12
arning EQU 09
               * Defining location of Outside Temperature in Celsius
ut_Temp EQU $13
               * Defining location of battery Voltage
       * note this is two bytes covering locations 14 and 15
attery_V EQU $14
               * Defining location of the thermocouple temperature
       * note this is two bytes covering locations 16 and $17
Temp EQU       $16
               **** Defining procedure locations in Ver2Sub ****
```

```
LM335_Temp EQU $B60A
Drive_Led  EQU $B649
SCANINPUTPIN    EQU     $B675
Battery_Volts   EQU     $B6D2
************************* The program ************************************
MAIN    ORG     $0100
*               *SETUP THE OPTION REGISTER TO;
*               *     ALLOW THE ANALOG TO DIGITAL CONVERTER TO BE USED
*               *     ACTIVATE THE CLOCK MONITOR
*               *     Set the COP timeout to be 262.14ms
        LDAA #$9A
        STAA $1039

*               *WAIT 200 CYCLES
*               *THIS IS A PRECAUTION TO ENSURE THAT NO READ OF THE ANALOG
*               *TO DIGITAL CONVERTER PINS IS DONE BEFORE THE SYSTEM IS
*               *STABILISED.  GIVES CHARGE PUMP TIME TO CHARGE.
        LDAA #$40
CHARGEAD:
        DECA
        BNE CHARGEAD

*                * Initialize the index register that will be used to
*               * reference memory in procedures
        LDY #$001f
MAINPROGRAMPART:
*               *** Reset the Warnings ***
*               * Set master Warning off
        LDAA #00
        STAA Master_Warning

*               * Set other warnings to 05 = "System Not Handled"
        LDX #Warning
        LDAB #$05
        LDAA #00
Clear_Warnings:
        STAB 0,X
        INX
        INCA
        CMPA #$09
        BLS Clear_Warnings

*                *** Scanning pin 2 ***
*                *Scan pin 2
        LDAA #$02
*                *Perform 4 scans
        LDAB #$04

        JSR SCANINPUTPIN

*                *Store result
        STAA CurrentVal2
*               * Call The LM335 procedure
        LDAA #$02
        JSR LM335_Temp

*               * Scanning pin 3 for the battery system
*               *scan pin 3
        LDAA #$03
*                * perform only 1 scan
        LDAB #$01
        JSR SCANINPUTPIN
*               * Store result
        STAA CurrentVal3
```

```
        LDAA #$03
        JSR Battery_Volts

      * Scanning pin 4 for the thermocouple system
                * scan pin = 4
        LDAA #$04
                * perform only 1 scan
        LDAB #$01
        JSR SCANINPUTPIN
                * store result
        STAA CurrentVal4
        * Call Ktype procedure
                * For pin input pin 4
        LDAA #$04
                * Junction temp = 20 degrees
        LDAB #$14
        JSR KType
                * Run Drive_Led procedure for the LM335 circuit
        LDAB #02
        JSR Drive_Led

                 ** Code for stop left so it can be easily added
         ldaa #00
         tap
         nop
         stop

        JMP MAINPROGRAMPART
************************* Ktype procedure *********************************
 Constants
 Offsets
TPin            EQU     $01
TJunctionT       EQU     $00
 NOte that KTKTemp is 16 bit so it covers location $03 as well
TKTemp          EQU     $02

 Warning triggers
 30 degrees low red
TLow_Red        EQU     $1E
 225 degrees yellow
TYellow         EQU     $E1
 270 degrees red
TRed            EQU     $10E
****************************************************************************
Type:
              Store the junction temperature in KTJunctionT
       STAB KTJunctionT,Y
              Store the AD pin number in KTPin
       STAA KTPin,Y
               *Determine address of CurrentVal corrisponding to the input
               * pin and load CurrentVal into Acc B
       TAB
       LDX #CurrentVal
       ABX
       LDAB 00,X
       Clear Accumulator A
       LDAA #00
               * Add Pin Value to Junction temperature
       ADDB KTJunctionT,Y
               * handle carries
       ADCA #00
               * Store the calculated temperature in KTKTemp
       STD KTKTemp,Y
               * Deterimine appropriate warning address and put in X
       LDX #Warning
```

```
                * Set pin warning level to green = 00
        LDAA #00
        STAA 00,X
                * Test KTKTemp  for Yellow Condition if so set yellow warning
                *  = 02  otherwise move to next test
        LDD KTKTemp,Y
        CPD #KTYellow
        BLO KTNoYellow
                * Set Yellow Warning
        PSHA
        LDAA #02
        STAA 00,X
        PULA

                * Test AccD for Low_Red condition if so set master warning
                * and warning level to low red = 01 otherwise move to next
                * test
KTNoYellow:
        CPD #KTLow_Red
        BHI KTNoLow_Red
                * Set Master Warning
        JSR Set_Master
                * Set Low Red warning
        PSHA
        LDAA #01
        STAA 00,X
        PULA

                * Test AccD for Red condition if so set master warning
                * and pin warning level to red = 03 otherwise do nothing.
KTNoLow_Red:
        CPD #KTRed
        BLO KTNoRed
                * Set Master Warning
        JSR Set_Master
                * Set red warning
        PSHA
        LDAA #03
        STAA 00,X
        PULA

                * Put the temperature (KTKTemp) in KTemp
KTNoRed:
        STD KTemp
                * return from procedure
        RTS
```

```
        This file also contains a jump instruction at address b600 that jumps
        to a program at address 100h.
        This file contains the subroutines required to run all version 4
        programs.  This file is to be assembled and placed in EEPROM
        This file contains;
                Set_Master
                LM335_Temp
                Drive_Led
                SCANINPUTPIN
                Battery_Volts

                * Defining location of first A/D pin value
CurrentVal EQU $00
                * Defining location of Master warning
Master_Warning EQU 08
                * Defining location of other Warnings
                * Current addresses go from 09 to 12
Warning EQU 09
                * Defining location of Outside Temperature in Celsius
Out_Temp EQU $13
                * Defining location of the battery Voltage
Battery_V EQU $14

*       Store in EEPROM
Ver2Sub         ORG $B600
************************* jump to program at address 100 ******************
        JMP $0100
************************* Set Master Warning ******************************
Set_Master:
*               * Stores current value of Acc A so that it may be returned
*               * unaltered at the conclusion of this procedure.
        PSHA
*               * Set master warning
        LDAA #01
        STAA Master_Warning
*                * Restore AccA value
        PULA
*               * return from procedure
        RTS
******************************************************************************
************************* LM335_Temp ******************************
*       *This procedure converts a reading from a LM335 temperature sensor
*       * and converts the reading to a value in degrees Celsius.
*
*      Procedure Constants
OTPin                   EQU 00
OTLow_Red               EQU $94
OTYellow                EQU $96
OTRed                   EQU $98

*               ***Procedure Code***
LM335_Temp:
*                *write Outside temp pin to Address Y + OTPin
        STAA OTPin,Y

*        *Determine Address of CurrentVal corrisponding to the LM335 sensor
*        *pin and load the value at this address into AccA

        LDX #CurrentVal
        TAB
        ABX
        LDAA 00,X

*                *Set AccB to 00 ie "green"
        LDAB #$00
```

```
                *Test AccA for the Yellow condition
      CMPA #OTYellow
                * Set Yellow if Acc A >= OTYellow_Condition
      BLO OTNo_Yellow
      LDAB #$02

TNo_Yellow:
                * Test for Low red condition
      CMPA #OTLow_Red
                * if Acc A <= low red condtion then
                        * set Low red
               *        * Set master warning
      BHI OTNo_Low_Red
      LDAB #$01
      JSR Set_Master

TNo_Low_Red:
                *Test for Red_Condtion
      CMPA #OTRed
                * if AccA >= Red Condtion then
                       *Set low red
                       * Set Master Warning
      BLO OTNo_Red
      LDAB #03
      JSR Set_Master

TNo_Red:
                * store Warning level on stack
      PSHB
                *Store the pin value on the stack
      PSHA
               * Determine the appropriate Warning address and place it in IX
      LDX #Warning
      LDAB OTPin,Y
      ABX

               *Retrieve pin value
      PULA
               * Retrieve Warning level
      PULB
               * Store Warning level at corrisponding Pin Warning address
      STAB 00,X

      * Determine temperature in Celsius
               *Subtract 119 from Pin Value
      SUBA #$77
                *load AccB with 0
      LDAB #00
               *load IX with denminator
      LDX #$0083
               * divide ACC D by 0083h = 0.5117875
      IDIV
               *put result in Acc D
      XGDX
               * Subtract 40
      SUBB #$28
               * Store result
      STAB Out_Temp
               * Return from procedure
      RTS
*************************************************************************
************************** Drive_Led ************************************
ive_Led:
               * Determine Warning address for the LM335
      LDX #Warning
```

```
                * Get the Current Warning Level
        LDAA 00,X
                * Set AccB to Nothing (Do not light any LEDs)
        LDAB #00
                * Test for green Warning level
        CMPA #$00
                * Set AccB to bit code to light green LED if Warning level
                * is equal to Green (PB 0) otherwise do nothing
        BNE DLNo_Green
        LDAB #$01
LNo_Green:
                * Test for Low Red Warning Level
        CMPA #$01
                * If Warning Level is Low_Red set AccB bit code to light the
                * orange LED (PB 1) otherwise do nothing
        BNE DLNo_Low_Red
        LDAB #$02
LNo_Low_Red:
                * Test for Yellow Warning level
        CMPA #02
                * If Warning Level is Yellow set AccB bit code to light the
                * Yellow LED (PB 2) otherwise do nothing
        BNE DLNo_Yellow
        LDAB #$04
LNo_Yellow:
                * Test for Red Warning level
        CMPA #03
                * If Warning Level is Red set AccB bit code to light the
                * Red LED (PB 3) otherwise do nothing
        BNE DLNo_Red
        LDAB #$08
LNo_Red:
                * Load AccA with the Master Warning level
        LDAA Master_Warning
                * test for Master Warning
        CMPA #$01
                * Light the Master Warning LED if the Master Warning is set
                * otherwise do nothing
        BNE DLNo_Master
        ORAB #$10
LNo_Master:
                * send data to port B
        STAB $1004
                * return from procedure
        RTS
************************ Scan Input Pin Procedure ************************
**************************************************************************
                *Data used by the Scan Input Pin Procedure

                *PinNumber is the number of the analog pin to be scanned
                *Valid values for PinNumber are 0 through 7
inNumber EQU $00

                *NumberOfScans determines the number of times a pin is sampled
                *The actual number of times the pin is sampled will be equal to
                *NumberOfScans * 4 since a single scan of a pin results in
                * 4 samples being taken and placed in ADR1 to ADR4
umberOfScans EQU $01
                *Counter is a variable used to keep track of the current scan
                *number and is also used in averaging
ounter EQU $02
                *Total is used for averaging purposes and is 2 byte
otal EQU $03
**************************************************************************
CANINPUTPIN
```

```
        STAB Counter,Y
        STAB NumberOfScans,Y

                *Write to ADCTL
EADLOOP:
        LDAA PinNumber,y
        STAA $1030

        * Wait for valid data
                * set n flag if data valid.  ie if ccf = 1
ALIDLOOP:
        LDAA $1030
                * loop if n flag not set
        BPL VALIDLOOP

        * Read samples average them and store average on the stack
                * Clear ScanTotal
        LDD #0000
                * ScanTotal = ADR1
        LDAB $1031
                * ScanTotal = ADR1 + ADR2
        ADDB $1032
                * Handle any carries
        ADCA #00
                *ScanTotal = ADR1 + ADR2 + ADR3
        ADDB $1033
                * Handle any carries
        ADCA #00
                * ScanTotal = ADR1 + ADR2 + ADR3 + ADR4
        ADDB $1034
                * Handle any carries
        ADCA #00

        * Average 4 results and store the average on the stack
        LDX  #$0004
        IDIV
        XGDX
                *storing on stack. Note AccA should be zero at this point
        PSHB

        * Decrement the counter.  Loop back if Counter is not equal to zero
        DEC Counter,Y
        BNE READLOOP

        * This part of the procedure retrieves the averages that were stored
        * on the stack.  It then averages them to get a final value for the
        * Input Pin.  This final value is placed in ACCA.  ACCB should be
        * equal to zero.

        * Set Counter = NumberOfScans
        LDAA NumberOfScans,Y
        STAA Counter,Y

        * Initialize Total to Zero
        LDD #$0000
        STD Total,Y

INDAVERAGELOOP:
                * Clear Accumulator D
        LDD #0000
                * Pop ScanAverage from stack
        PULB
                * Add current ScanAverage value to Total
        ADDD Total,Y
                * Store in Total
```

```
              * Decrement counter.  The zero flag will be set when all
              * Scan averages have been processed
       DEC Counter,Y
*             * Loop if counter is not zero
       BNE FINDAVERAGELOOP
*      * Set denominator to NumberOfScans
       LDAB NumberOfScans,Y
       LDAA #$00
       XGDX
*            * Set numerator to Total
       LDD Total,Y
*            * Perform division.  This will place result in IX
       IDIV
*             * Store result on Stack
       PSHX
*             * Pull result from stack and place it in ACCA.  NOTE
*             * This should set ACCB to zero also
       PULB
       PULA
*            * Return from procedure

       RTS
*****************************************************************************
*************************** Battery_Volts procedure*************************
* Data used by Battery_Volts procedure
* constants
*      * Battery pin number local offset
BAPin          EQU 00
*      * Warning level triggers
BALow_Red      EQU $00
BAYellow       EQU $77
BARed          EQU $FF
*****************************************************************************
Battery_Volts:
*             * Write Battery Pin number to Address Y + BAPin
       STAA BAPin,Y

*      * Determine the address of Current Val corrisponding to the battery
*      * pin and load this into Acc A
       LDX #CurrentVal
       TAB
       ABX
       LDAA 00,X

*             * Set AccB to Green = 00
       LDAB #$00

*             * Test AccA for yellow condition
       CMPA #BAYellow
*             * Set Yellow condition if A >= BAYellow
       BLO BANo_Yellow
       LDAB #$02
BANo_Yellow:
*             * Test for Low red condition
       CMPA #BALow_Red
*             * if AccA <= low red condition
*                     * set low red
*                     * set master warning
       BHI BANo_Low_Red
       LDAB #$01
       JSR Set_Master
BANo_Low_Red:
*             * Test for red condition
       CMPA #BARed
*             * if AccA >= Red condition then
```

```
      BLO BANo_Red
      LDAB #03
      JSR Set_Master
No_Red:
              * Store Warning level on the stack
      PSHB
              * Store Pin Value on Stack
      PSHA
              * Determine appropriate Warning Address and put in X
      LDX #Warning
      LDAB BAPin,Y
      ABX
              * Retrieve Pin Value
      PULA
              * Pop Warning Level
      PULB
              * Store Warning level at corresponding warning Address
      STAB 00,X
              * Set Numerator = PinValue . 00
      LDAB #00
              * Set denominator == 16 . 00
      LDX #$1000
              * perform integer division
      IDIV
      * Store integer result in Battery_V
      PSHB
      PSHA
      XGDX
      STAB Battery_V
      PULA
      PULB
              * Set denominator = 16.00 for FRACTIONAL division
      LDX #$1000
              * perform Fractional division on the remainder of IDiv
      FDIV
              * Store fractional result in Battery_V
      XGDX
      LDX #$01
      STAA Battery_V,X
              * return from procedure
       RTS
```

```
getcount.asm
This program use the HC11 timer system to determine the number of timer
counts between two consecutive pulses
It is part of the development of the RPM procedure.

Variable offsets
PICPin          EQU     $00
PRtoFRatio      EQU     $01
PDvCount        EQU     $02
PICCount        EQU     $03
PCount1HB       EQU     $04
PCount1LB       EQU     $05
PCount2HB       EQU     $06
PCount2LB       EQU     $07
PICAddress      EQU     $08
PCountsHB       EQU     $02
PCountsLB       EQU     $03

note RPICAddress is 2 byte so it also covers offset $09
AIN org $0100

*** SETING UP THE PRE SCALER this will normally done by the main program
       LDAA #$01
       STAA $1024

Setting the Y index register to 20hex
       LDY #$20
               * To reduce code size and to increase program speed index X
               * will be used for memory offsets in this procedure as this
               * register is not required for most parts of this procedure
               * Setting X to Y.  When the X index register is used for
               * other purposes its value will be temporarily stored on the
               * stack
       PSHY
       PULX

                *Store the Input Capture Pin Number
       STAA RPICPin,X
                * Store the Frequency to RPM Multiplier
       STAB RPRtoFRatio,X
                * reset the COP watchdog to avoid timeout
       LDAA #$55
       STAA $103A
       LDAA #$AA
       STAA $103A
               * Set timer overflow Interrupt Vector to "Handle Overflow"
       LDD #RPHandOv
       STD $00D1
               * Reset timer Overflow Flag in the TFLG2 register
       LDAA #$01
       ORAA $1025
       STAA $1025
               * Change Main Interrupt Flag in CCR to allow maskable
               * interrupts
       TPA
       ANDA #%11101111
       TAP
               * Alter TMSK2 register TOI bit to allow timer overflow
               * interrupts
       LDAA $1024
       ORAA #%10000000
       STAA $1024
               * Set Input Capture Count to zero RPICCount = 0
       LDAA #00
       STAA RPICCount,X
```

```
        LDD #RPHandIC
        STD $00E9
        STD $00E6
        STD $00E3
*       * Determining appropriate input capture timer value address
*       * Could be TIC1 or TIC2 or TIC3
*               * Set address to point to TIC1 (high byte)
        LDD #$1010
        STD RPICAddress,X
*               * Are we using Input Capture 1 if so leave address as TIC1
        LDAA RPICPin,X
        CMPA #1
        BEQ RPAddCorrect
*               * Set offset to point to TIC2 (high byte)
        LDD #$1012
        STD RPICAddress,X
*               * Are we using Input Capture 2 if so leave address as TIC2
        LDAA RPICPin,X
        CMPA #2
        BEQ RPAddCorrect
*               * Otherwise Assume we are using Input Capture 3
*               * Set offset to point to TIC3
        LDD #$1014
        STD RPICAddress,X
RPAddCorrect:
***********

*       * Alter TMSK1 register to allow appropriate pin input capture
*       * interrupt
        LDAA RPICPin,X
*                * Input capture pin 1 case
        LDAB #%00000100
        CMPA #$01
        BEQ RPCorrectPin
*                * Input capture pin 2 case
        LDAB #%00000010
        CMPA #$02
        BEQ RPCorrectPin
*              * Otherwise assume input capture pin 3
        LDAB #%00000001
RPCorrectPin:
        ORAB $1023
        STAB $1023
**********
*       * The input capture loop
RPICLoop:
*               * Have 2 timer overflows occurred before an input capture
*               * (RPOvCount >= 2 ?) ( NO PULSES) if so set the RPM to zero
*               * and quit otherwise continue looping
        LDAA RPOvCount,X
        CMPA #$02
        BNE RPNotTimeout
*       This block of code handles 2 timer overflows without an input pulse
*               * disable all interrupts
*                * input capture
        LDAA #%11111000
        ANDA $1023
        STAA $1023
*               * Timer overflow
        LDAA $1024
        ANDA #% 01111111
        STAA $1024
*               * Set the RPM to 0
*               * Return from procedure
```

```
RNotTimeout:
            * Have both input captures occured (RPICCount >= 2?) ? if so
            * finish looping otherwise continue looping
    LDAA RPICCount,X
    CMPA #$02
    BLO RPICLoop
            * Disable all interrupts
             * input capture
    LDAA #%11111000
    ANDA $1023
    STAA $1023
             * Timer overflow
    LDAA $1024
    ANDA #% 01111111
    STAA $1024

            * Determine period in timer Counts
    LDD RPCount2HB,X
    SUBD RPCount1HB,X
            * Store RPCountsHB/ Store RPCountsLB
    STD RPCountsHB,X
            * Return from procedure ****************
    RTS
** Handle Overflow interrupt *********************************************
RPHandOv:
            *Reset COP watchdog
    LDAA #$55
    STAA $103A
    LDAA #$AA
    STAA $103A

            * Increment Overflow Count (RPOvCount)
    INC RPOvCount,X
            * Clear Overflow Flag in TFLG2 register
    LDAA #%10000000
    STAA $1025
            * Return from interrupt
    RTI

** Handle Input Capture interrupt *****************************************
RPHandIC:
            * Clear Overflow Count (RPOvCount)
    LDAA #$00
    STAA RPOvCount,X
            * if there has been a previous input capture then put timer
            * value into RPICCount2 and change interrupt vectors otherwise
            * put the input capture timer value in RPICCount1
    LDAA RPICCount,X
    CMPA #$00
    BHI RPSecondIC

RPFirstIC:
   * put Input Capture timer value in RPCount1
            * Pushing local variable offset onto the stack
    PSHX
            * Loading X with the correct input capture address
    LDX RPICAddress,X
            * Getting input capture timer value into D
    LDD $00,X
            * Pulling the local variable offset from the stack
    PULX
             * Storing inputcapture timer value in RPCount1HB and LB
    STD RPCount1HB,X
            * Continue the interrupt
```

```
*     * put input capture timer value in RPCount2
RPSecondIC:
*               * Pushing local variable offset onto the stack
        PSHX
*               * Loading X with the correct input capture address
        LDX RPICAddress,X
*               * Getting Input Capture timer value into D register
        LDD $00,X
*               * Pulling the local variable offset from the stack
        PULX
*               * Storing input capture timer value in RPCount2HB and LB
        STD RPCount2HB,X
*               * Change overflow interrupt vector do "Do nothing"
        LDD #RPDoNothing
        STD $00D1
*               * Change input capture interrupt vector to "Do nothing"
        STD $00E3
        STD $00E6
        STD $00E9
*               * Continuing the interrupt
RPContInt:
*                * Increment the input capture count
        INC RPICCount,X
*               * Clear the input capture flag in the TFLG1 register
        LDAA #%00000111
        STAA $1023
*               * return from interrupt
        RTI
** Do Nothing interrupt **************************************************
RPDoNothing:
*               * Clear the overflow flag in TFLG2
        LDAA #%01111111
        ANDA $1025
        STAA $1025
*               * Clear the Input Capture flags in TFLG1
        LDAA #%11111000
        STAA $1023
*               * return from interrupt
        RTI
**************************************************************************
```

```
* The program combines Period.asm and Freq.asm
* The program takes a value for timer counts and determines the frequency

************************** Subprocedure Data and offsets *******************
* Constants
RPCountsizeHB   EQU     $21
RPCountsizeLB   EQU     $8D

* Offsets
RPICpin         EQU     $00
RPRtoFRatio     EQU     $01
RPCountsHB      EQU     $02
RPCountsLB      EQU     $03
RPResult1       EQU     $04
RPResult2       EQU     $06
RPResult3       EQU     $08
*RPResult4              This variable needs no offset and exists in
*                        register D only briefly
RPPeriod        EQU     $0A
RPPeriodLB      EQU     $0C
RPPeriodMB      EQU     $0B
RPPeriodVLB     EQU     $0D
*               * Variables used in the conversion from period to frequency
RPShiftBy       EQU     $02
RPFrequencyHB   EQU     $03
RPFrequencyMB   EQU     $04
RPFrequencyLB   EQU     $05
RPFrequencyFH   EQU     $06
RPFrequencyFL   EQU     $07

MAIN org $0100

* Setting the Y index register to 20hex
        LDY #$20
*               * To reduce code size and to increase program speed index X
*               * will be used for memory offsets in this procedure as this
*               * register is not required for other parts of this procedure
*               * Setting X to Y
        PSHY
        PULX

*               * Store Counts in RPCountsHB and RPCountsLB
        STD RPCountsHB,X
*               load AccumulatorA with RPCountsLB (The low byte of the
*               * time taken in counts
        TBA
*               *load Accumulator B with the low byte of the time per
*               * Count (RPCountsizeLB)
        LDAB #RPCountsizeLB
*               *Perform multiplication
        MUL
*               * Store the result of the multiplication of the two low bytes
*               * in RPResult1
        STD RPResult1,X
*               * load AccA with the high byte of the time taken in counts
        LDAA RPCountsHB,X
*               * load AccB with the low byte of the time per
*               * count(RPCountsizeLB)
        LDAB #RPCountsizeLB
*               * perform multiplication
        MUL
*               * Store the result in RPResult2
        STD RPResult2,X
*               * load AccA with the low byte of the time taken in counts
        LDAA RPCountsLB,X
```

```
*               * perform the multiplication
        MUL
*               * Store the result in RPResult3
        STD RPResult3,X
*               * Load AccA with the high byte of the time taken in counts
        LDAA RPCountsHB,X
*               * Load AccB with the high byte of the time per count
        LDAB #RPCountsizeHB
*               * perform the multiplication answer = RPResult4
        MUL
*       *Determining the period
*               * RPPeriod = RPResult4 * 10000hex
        STD RPPeriod,X
*               * Adding RPResult1 to RPPeriod
        LDD RPResult1,X
        STD RPPeriodLB,X
*               * Adding RPResult2 * 100hex to RPPeriod
        LDD RPResult2,X
        ADDD RPPeriodMB,X
        STD RPPeriodMB,X
        BCC RPNoCarry1
*               *Handle any carries
        INC RPPeriod,X
RPNoCarry1:
*               * Adding RPResult3 * 100hex to RPPeriod
        LDD RPResult3,X
        ADDD RPPeriodMB,X
        STD RPPeriodMB,X
        BCC RPNoCarry2
*               * Handle any carries
        INC RPPeriod,X
RPNoCarry2:
*               *Set RPShiftby = 1
        LDAA #$01
        STAA RPShiftBy,X
RPTestPlace:
*               * Test Most Significant 4 bits (The most significant
*                * Hexadecimal place) of the Period to see if it is zero
        LDAA RPPeriod,X
        ANDA #$F0
*               * Were the Most Significant 4 bits = 0 if not move on to the
*               * division part
        BNE RPDivision
*               * Has the period already been shifted 4 Hexadecimal places?
*               * (RPShiftBy = 5?) If so then move on to the division part
        LDAA RPShiftBy,X
        CMPA #$5
        BEQ RPDivision
*       * This block of code performs a one hexadecimal place shift on the
*       * period

*               * Increment RPShiftBy to keep track of Hexadicimal places
*               * shifted
        INC RPShiftBy,X
*                * load AccA with 4 so a 4 bit shift can take place
        LDAA #$04
RPShiftPeriod:
*               * Shift the lowest Byte of the Period by 1 bit
        ASL RPPeriodVLB,X
*               * Shift the next lowest byte of the period by 1 bit
        ROL RPPeriodLB,X
*               * Shift the next lowest byte of the period by 1 bit
        ROL RPPeriodMB,X
*               * Shift the highest byte of the period by 1 bit
```

```
*                   * Lower the bit count to keep track of how many bits have
*                   * been shifted
        DECA
*                   * Has the period been shifted by four bits?  If not shift by
*                   * another bit else go to the testing of the most significant 4
*                   * bits for zero.
        BNE RPShiftPeriod
        JMP RPTestPlace
*       * This block of code performs the division
RPDivision:
*                   * Load D with the Numerator = 1000h
        LDD #$1000
*                   * Load X with the Denominator = four most significant non zero
*                   * hexadecimal places of the period
*                   * Also storing the value of x for local variable offsets.
        PSHX
        LDX RPPeriod,X
*                   * Perform the Fractional division.
        FDIV
*                   * Store the result in RPFrequencyFH:RPFrequencyFL
        STX RPFrequencyFH,Y
*                   * Recover value of x for local variable offsets
        PULX
*                   * Initialize RPFrequencyHB,RPFrequencyMB and RPfrequencyLB
*                   * to equal 0
        LDAA #00
        STAA RPFrequencyHB,X
        STAA RPFrequencyMB,X
        STAA RPFrequencyLB,X

*       * This block of code cancels out any previous shifting to get the
*       * frequency in the correct Hexadecimal places.
RPCompensate:
*                   * Are all compensating hexadecimal place shifts complete?
*                   * (RPShiftBy = 0?) if so move on to the next part of the
*                   * procedure.  Otherwise perform compensation
        LDAA RPShiftBy,X
        BEQ RPFindRPM
*        * This block performs a 4 bit shift of the frequency
*                   * Decrement the Shift by count to keep track of places
*                   * shifted
        DEC RPShiftBy,X
*                   * load AccA with 4 so we can perform a 1 hexadecimal place
*                   * (4 bit) shift
        LDAA #$04
*                   * Shift the lowest byte of the fractional part of the
*                   * frequency by one bit
RPShiftFreq:
        ASL RPFrequencyFL,X
*                   * Shift the highest byte of the fractional part of the
*                   * frequency by one bit
        ROL RPFrequencyFH,X
*                   * Shift the lowest byte of the integer part of the frequency
*                   * by one bit
        ROL RPFrequencyLB,X
*                   * Shift the middle byte of the integer part of the frequency
*                   * by one bit
        ROL RPFrequencyMB,X
*                   * Shift the highest byte of the integer part of the
*                   * frequency by one bit
        ROL RPFrequencyHB,X
*                   * Lower the bit count by one
        DECA
*                   * Have all the 4 one bit shifts occured?  If not shift
*                   * by another bit otherwise test to see if compensation
```

```
        JMP RPCompensate
;****
FindRPM:
; Infinite loop when finished.  Waiting for a reset.
Finished:
        JMP Finished
;
```

Appendix G

This appendix contains the Numerical Results from the testing of the various subsystems. There are three parts to this Appendix;

- Testing Results for the Battery Voltage System
- Testing Results for the Cylinder Head Temperature System
- Testing Results for the Ambient Temperature System

Simulated Battery Voltage Vb	HC11 Calculated Voltage Vb*	Real Voltage	error	max error in range		A/ d converte Pin voltage
0	0	0	0	0.1875		
1	1	1	0			0
1.48	1.4375	1.48	0.0425			0.307
2	1.9375	2	0.0625			0.453
2.34	2.25	2.34	0.09			0.609
3.05	2.9375	3.05	0.1125			0.714
3.82	3.6875	3.82	0.1325			0.932
4.23	4.125	4.23	0.105			1.165
4.76	4.625	4.76	0.135			1.291
5.07	4.9375	5.07	0.1325			1.453
5.48	5.3125	5.48	0.1675			1.672
5.8	5.625	5.8	0.175			1.77
6	5.8125	6	0.1875			1.83
6.47	6.3125	6.47	0.1575			1.977
6.93	6.9375	6.93	0.0075			2.17
7.37	7.375	7.37	0.005			2.3
7.45	7.4375	7.45	0.0125			2.33
8.1	8.0625	8.1	0.0375			2.53
8.62	8.625	8.62	0.005			2.69
9.01	9	9.01	0.01			2.82
9.54	9.5	9.54	0.04			2.98
10.28	10.25	10.28	0.03			3.22
10.86	10.875	10.86	0.015			3.4
11.64	11.625	11.64	0.015			3.64
11.75	11.75	11.75	0			3.67
11.8	11.8125	11.8	0.0125			3.69
11.86	11.875	11.86	0.015			3.71
11.9	11.875	11.9	0.025			3.72
11.91	11.875	11.91	0.035			3.73
11.92	11.9375	11.92	0.0175			3.73
11.98	11.9375	11.98	0.0425			3.77
12.05	12.0625	12.05	0.0125			3.79
12.12	12.125	12.12	0.005			3.81
12.17	12.1875	12.17	0.0175			3.83
12.23	12.25	12.23	0.02			3.84
12.28	12.25	12.28	0.03			3.86
12.33	12.3125	12.33	0.0175			3.87
12.38	12.375	12.38	0.005			3.9
12.46	12.375	12.46	0.085			3.91
12.48	12.4375	12.48	0.0425			3.92
12.52	12.5	12.52	0.02			3.93
12.56	12.5625	12.56	0.0025			3.95
12.61	12.625	12.61	0.015			3.96
12.66	12.6875	12.66	0.0275			3.98

Testing results for the Battery Voltage System

	12.71	12.6875	12.71	0.0225			3.99
	12.74	12.75	12.74	0.01			4
	12.79	12.8125	12.79	0.0225			4.03
	12.86	12.875	12.86	0.015			4.04
	12.91	12.875	12.91	0.035			4.06
	12.96	12.9375	12.96	0.0225			4.07
	13.01	13	13.01	0.01			4.27
	13.63	13.625	13.63	0.005			4.51
	14.4	14.3775	14.4	0.0225			4.67
	14.93	14.9375	14.93	0.0075			4.84
	15.44	15.4375	15.44	0.0025			4.92
	15.72	15.6875	15.72	0.0325			5.01
	16	15.9375	16	0.0625			5.9
	16.25	15.9375	16.25	0.3125			

accuracy			
HC11calculated			
A/D	real pin	error	max error
voltage	Voltage		in range
			0.033
0	0	0	
0.3125	0.307	0.0055	
0.449	0.453	0.004	
0.605	0.609	0.004	
0.703	0.714	0.011	
0.918	0.932	0.014	
1.152	1.165	0.013	
1.289	1.291	0.002	
1.445	1.453	0.008	
1.66	1.672	0.012	
1.758	1.77	0.012	
1.816	1.83	0.014	
1.973	1.977	0.004	
2.168	2.17	0.002	
2.305	2.3	0.005	
2.324	2.33	0.006	
2.52	2.53	0.01	
2.695	2.69	0.005	
2.8125	2.82	0.0075	
2.969	2.98	0.011	
3.23	3.22	0.01	
3.398	3.4	0.002	
3.633	3.64	0.007	
3.672	3.67	0.002	
3.691	3.69	0.001	
3.711	3.71	0.001	
3.711	3.72	0.009	
3.711	3.73	0.019	
3.73	3.73	0	
3.77	3.77	0	
3.789	3.79	0.001	
3.809	3.81	0.001	
3.828	3.83	0.002	
3.828	3.84	0.012	
3.848	3.86	0.012	
3.867	3.87	0.003	
3.867	3.9	0.033	
3.887	3.91	0.023	
3.906	3.92	0.014	
3.926	3.93	0.004	
3.945	3.95	0.005	
3.965	3.96	0.005	
3.965	3.98	0.015	

	3.984	3.99	0.006	
	4.004	4	0.004	
	4.023	4.03	0.007	
	4.023	4.04	0.017	
	4.043	4.06	0.017	
	4.063	4.07	0.007	
	4.258	4.27	0.012	
	4.492	4.51	0.018	
	4.668	4.67	0.002	
	4.824	4.84	0.016	
	4.902	4.92	0.018	
	4.98	5.01	0.03	
	4.98	5.9	0.92	

Differential Voltage input to the instrumentation amplifier (milli Volts)	Instrumentation amplifier output Voltage . (Volts)	ideal instrumentation amplifier output . (Volts)	Hexadecimal Value Corresponding to A/D pin voltage .	Voltage as determined by HC11 . .
0	0.005	0	$00	0
0.5	0.24	0.2405	$0D	.253
1	0.51	0.481	$1B	.527
1.5	0.747	0.7215	$25	.723
2	0.966	0.962	$32	.977
2.5	1.221	1.2025	$3F	1.23
3	1.478	1.443	$4B	1.465
3.5	1.714	1.6835	$59	1.738
4	1.953	1.924	$64	1.953
4.5	2.19	2.1645	$70	2.189
5	2.44	2.405	$7F	2.48
5.5	2.66	2.6455	$88	2.656
6	2.93	2.886	$97	2.949
6.5	3.15	3.1265	$A0	3.125
7	3.4	3.367	$AF	3.418
7.5	3.66	3.6075	$BB	3.652
8	3.88	3.848	$C6	3.867
8.5	4.14	4.0885	$D5	4.160
9	4.38	4.329	$DF	4.355
9.5	4.62	4.5695	$EB	4.59
10	4.85	4.81	$F7	4.824
10.5	5.13	5.0505	$FF	4.98

all temperatues assume 20 degree junction temp

Temperature Determined by the HC11	Temperature based on differential input voltage	Temperature based on instrumentation amplifier output	Temperature based on voltage determined by HC11	absolute error between temp based on diff input voltage and HC11 determined temp
20	20	20.25618691	20	0
33	32.3228589	32.29697187	32.9630578	0.677141097
47	44.64571781	46.13106523	47.0021007	2.354282193
57	56.96857671	58.27432495	57.0446278	0.03142329
70	69.29143561	69.49531178	70.058923	0.708564387
83	81.61429452	82.56084439	83.0219808	1.385705484
95	93.93715342	95.72885177	95.0627658	1.06284658
109	106.2600123	107.8208741	109.050571	2.739987677
120	118.5828712	120.0666086	120.066609	1.417128774
133	130.9057301	132.2098683	132.158631	2.094269871
147	143.228589	145.019214	147.068709	3.771410967
156	155.5514479	156.2914382	156.086489	0.448552064
171	167.8743068	170.1255316	171.099042	3.125693161
180	180.1971657	181.3977558	180.116821	0.197165742
195	192.5200246	194.2071015	195.129374	2.479975354
207	204.8428835	207.528821	207.118922	2.157116451
218	217.1657425	218.8010452	218.134959	0.834257548
233	229.4886014	232.1227648	233.147512	3.511398644
243	241.8114603	244.4197366	243.138802	1.188539741
255	254.1343192	256.7167085	255.179587	0.865680838
267	266.4571781	268.5013066	267.169135	0.542821935
275	278.780037	282.8477737	275.162166	3.780036969

absolute error between real and ideal instrumentation amplifier outputs (Volts)
0.005
0.0005
0.029
0.0255
0.004
0.0185
0.035
0.0305
0.029
0.0255
0.035
0.0145
0.044
0.0235
0.033
0.0525
0.032
0.0515
0.051
0.0505
0.04
0.0795

Maximum temperature error across operating range
3.771410967

Maximum error between real and ideal instrumentation amplifier output voltages in operating range
0.0525

Analysis of the accuracy of the A/D converter combined with the LM335_Temp procedure

Voltage at the HC11 A/D pin (VH)	Corresponding temperature in Celsius	HC11 Calculated Temperature in Celsius	absolute error
0	-273	-29	244
0.51	-222	22	244
1.12	-161	83	244
1.67	-106	-119	13
2.22	-51	-64	13
2.32	-41	-54	13
2.34	-39	-39	1.42E-14
2.35	-38	-39	1
2.36	-37	-36	1
2.43	-30	-31	1
2.51	-22	-23	1
2.58	-15	-15	8.88E-15
2.66	-7	-7	1.6E-14
2.75	2	2	1.78E-15
2.85	12	12	1.07E-14
2.92	19	18	1
2.94	21	20	1
2.99	26	26	2.13E-14
3.11	38	38	1.42E-14
3.19	46	45	1
3.27	54	53	1
3.36	63	63	1.42E-14
3.42	69	67	2
3.51	78	77	1
3.59	86	87	1
3.62	89	88	1
3.73	100	100	0
3.79	106	104	2
3.85	112	112	1.42E-14
4.43	170	-87	257

Appendix H

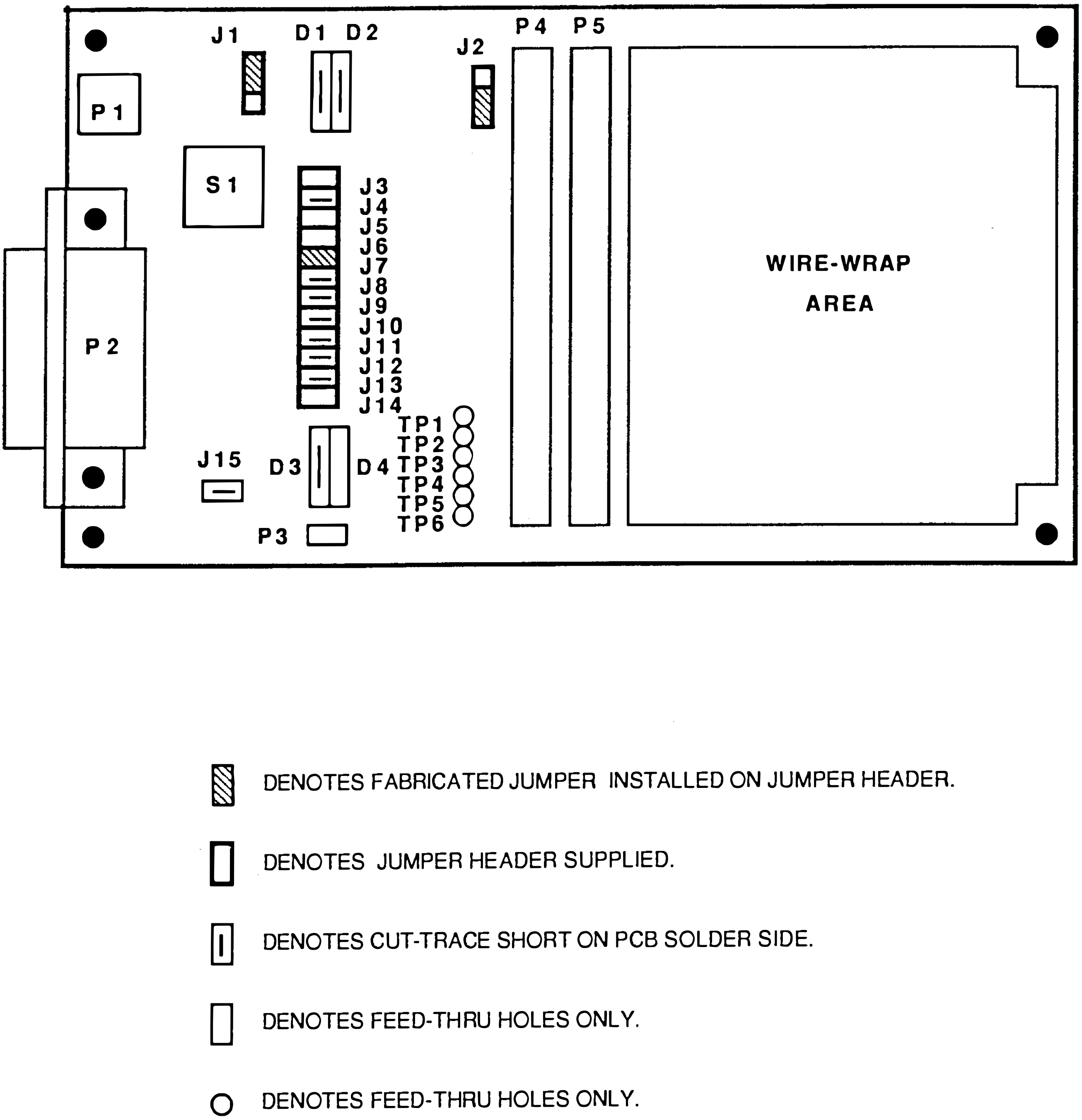


FIGURE 2-1. EVBU Connector, Switch, and Jumper Header Location Diagram