Automotive
Control
Solutions

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

March 3, 2005

Mr. Lakshman One
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 440 Project Design Specification for the AF Optimizer

Dear Mr. One:

Please find the attached document, *Automotive Control Solutions' AF Optimizer –Design Specification*, which outlines our project for ENSC 440. The included document contains all of the functional requirements of the AF Optimizer, as determined by the design team. The ACS team is currently well into the prototype development stage of the AF Optimizer, a low-cost, flexible air/fuel controller.

The purpose of this document is to specify design requirements to be met by the AF Optimizer. These include detailed descriptions of the design of hardware components, software, as well as packaging and user interface. These requirements will meet or exceeded at the prototype stage, with further design optimization before the final product is created. Together, these specifications will define the system's overall design, including all pertinent information for the design engineers.

Automotive Control Solutions is a company that is interested in providing electronic performance solutions for car enthusiasts. ACS is comprised of intelligent individuals who are in their concluding year of engineering science at Simon Fraser University; Alex Gutica, Brian Nelson, and Russell Potter. If you have any concerns with this project proposal, please feel free to contact us at acs-ensc440@sfu.ca.

Sincerely,

Russell Potter
President and CTO
Automotive Control Solutions

Enclosure: Automotive Control Solutions' AF Optimizer –Design Specification

Automotive Control Solutions' AF Optimizer

Design Specification

Version 1.0 (March 2005)

Team Members:

Alex Gutica
Russell Potter
Brian Nelson

Team Contact:

Russell Potter
rjpotter@sfu.ca

Group Contact:

ACS-ENSC440@sfu.ca

Submitted To:

Mr. Lakshman One (ENSC440)
School of Engineering Science
Simon Fraser University

Mr. Mike Sjoerdsma (ENSC305)
School of Engineering Science
Simon Fraser University

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

Executive Summary

This document outlines in further detail the design decisions of the AF Optimizer project. This product fills a unique and expansive niche in the after-market automobile part industry. It will give users the ability to inexpensively install/uninstall an air-flow recalibration unit, with the result being consistent and accurate control of the air/fuel mixture. Current competitors offer similar functionality to the AF Optimizer, at a premium price. The here-contained specifications will be comparable to competitors' products, satisfying more than the basic needs of our customer.

The project itself is comprised of two stages: the prototype stage and the production ready stage. The prototype stage will be comprised of the following modules:

1. Input hardware to interface vehicle signals to the microcontroller
2. Programmed micro-controller to read, process, and output information to all other subsystems
3. Output circuitry to interface the micro-controller to the car's ECU
4. User-interface hardware and software including input buttons and a VFD Display

The final product will include prototype features, as well as:

1. A small plastic enclosure and mounting hardware.
2. A detailed user manual

The prototype design and construction will be complete by the end of March, 2005.

Table of Contents

List of Figures

List of Terms

ACS	Automotive Control Solutions
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
IC	Integrated Circuit
ISR	Interrupt Subroutine function that is triggered by an interrupt
MSB/LSB	Most Significant Bit / Least Significant Bit
O2 Sensor	Oxygen sensor in a vehicle used to measure exhaust gas for oxygen content
PIC	Used in this document to refer to the PIC16F877
USART	Universal Synchronous Asynchronous Receiver Transmitter used in serial communication
UART	Universal Asynchronous Receiver Transmitter used in serial communication
VFD	Vacuum Fluorescent Display

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

1 Introduction

The AF Optimizer product is intended to appeal to a wide variety of amateur auto enthusiasts. Once the product is designed and completed to specification, it will perform at or above the expectations of customers who are looking to optimize their air/fuel control. The product is currently in the design phase. The first milestone, the prototype, will be completed by the end of March, 2005. After that time, the design cycle will continue at the next level, bringing to fruition all those requirements intended for the final product.

1.1 Scope

This document outlines the AF Optimizer's design decisions and is organized into the various subcomponents of the project. From these specifications, design engineers will successfully meet all the functional requirements [2] set out earlier in the project. The content of this document will explain how the ACS team arrived at the current set of design decisions, and also detail how these decisions will be implemented in the prototype product.

1.2 Intended Audience

Design engineers will use this document to create design specifications for the components that make up the air-fuel optimizer.

Managerial staff will use this document to track progress of each subsystem, as well as correctness of implementation. Also, the design specification will offer insight into which parts will be needed to assemble the prototype, and in what quantity.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

2 System Overview

The AF Optimizer is an electronic controller that will be used as a tool to optimize a vehicle's air-fuel ratios. By giving the automotive enthusiast the ability to have a customized recalibration of their vehicle's airflow sensor, their vehicle can be made to run more efficiently with either increased horsepower output or increased fuel efficiency. We give the automotive enthusiast the capacity to easily and reliably recalibrate the air-fuel ratios throughout their entire RPM range and at different throttle positions. Figure 2-1, shown below, identifies where the AF Optimizer fits in its intended application and installation into a vehicle [1].

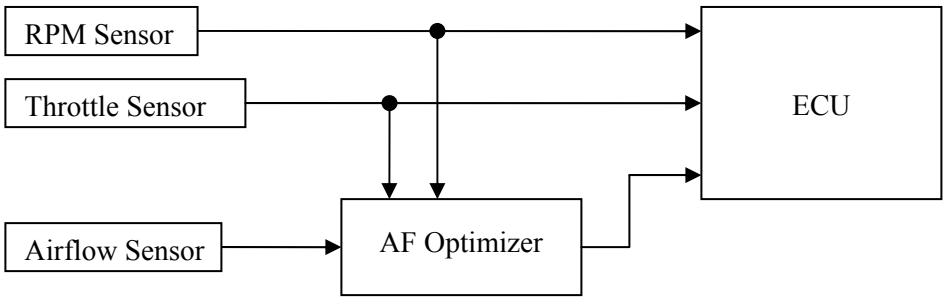

Figure 2-1: AF Optimizer System Layout

2.1 Functional Modules of the AF Optimizer

The AF Optimizer itself can also be broken down into various components. Figure 2-2 outlines those components, and how they interconnect in a broad sense.

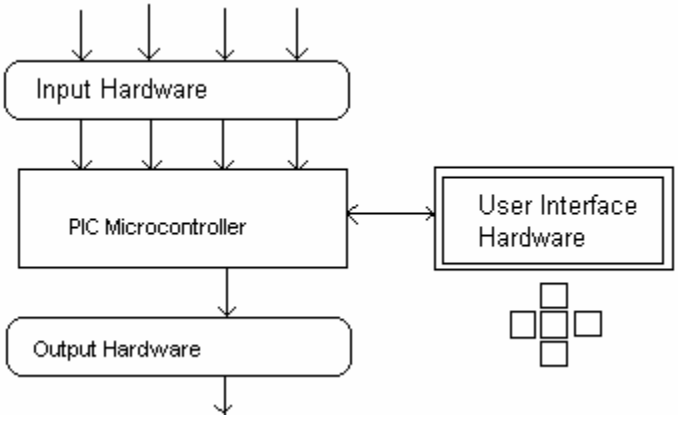

Figure 2-2: Modular Overview

Each of the various subsystems carries with it a set of design specifications. The following sections of this document detail the group of specifications associated with each module.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

3 Software Design

A very important aspect of the AF Optimizer is the software that implements its vital and auxiliary functions. The microcontroller is programmed to handle the DSP functionality of the AF Optimizer as well as control all user interface requirements and engine monitoring features. The software must allow the AF Optimizer to perform properly at all times, while meeting all conditions outlined in our functional specification [2] document. All the software functional blocks of our product are thoroughly discussed throughout the remainder of this section.

3.1 User Interface Software

The user interface is the most critical portion of the AF Optimizer since users need to be able to understand how to use its particular features. The user interface software is primarily concerned with its output communication with the VFD but must be able to understand input through the five navigation buttons.

Firstly, we must understand the three modules of the display menu system that are used in our AF Optimizer. A Setup module is used to allow the user to setup various attributes like the low and high throttle threshold percentages, or the shift light rpm. A Configure module is used to allow the user to manipulate and change the values of the recalibration curves. This module allows gives the user the ability to recalibrate the airflow sensor based on rpm (in 250 rpm increments) and throttle position (high or low throttle). Lastly, a Monitor module presents the user with real time information from engine speed to throttle position or pre and post calibrated airflow values.

These modules are navigated in a sequential manner, such that the user can scroll between each of the three modules in sequence by hitting the "ENTER" input button. A diagram of this structure is shown below in Figure 3-1.

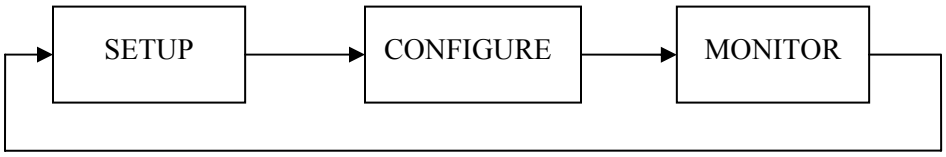

Figure 3-1: Display Module Sequence

In addition to these output display modules, software must be designed to handle the input button portion of the user interface. The following sections will discuss the software design issues for the aforementioned modules and for the input buttons.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

3.1.1 Setup Module

The Setup Module guides the user through the various setup procedures that must be performed in order for the AF Optimizer to operate correctly. In order for the AF Optimizer to perform recalibration interpolations based on throttle position, it must be able to correlate a certain throttle position sensor voltage with a throttle percentage. Thus, knowing what voltage (assuming it is not 0.0 Volts) is associated with zero-throttle and what voltage (assuming it is not 5.0 Volts) is associated with full throttle is imperative to our system's performance. The Setup module makes certain that the user has input valid voltage levels for the throttle position.

Less significant setup variable can also be changed in the Setup module. These variables include setting the high and low throttle threshold percentages, associated with the two airflow calibration curves, and the shift light RPM. The reason these variables are less significant is because the AF Optimizer will initialize with values that are initially stored in the unit's memory. Thus, the AF Optimizer will work properly without the user having to input new values.

Initially, when the unit is turned on for the very first time, the low throttle threshold percentage is set at 10% throttle, and the high throttle threshold percent is set at 70% throttle. Moreover, the initial shift light RPM is set to 5000 RPM. Unless these variables are changed by the user in the Setup module, the AF Optimizer will use these predefined values. The user is able to change the predefined values with the use of the "+" and "-" input buttons. Naturally, hitting the "+" button will increase the value, and hitting the "-" will decrease the value that is being manipulated.

Certain constraints must be implemented in the Setup module. For example, ask yourself what would happen if the user set the low throttle threshold percentage to a value that is greater than the high throttle threshold percentage. Obviously, this scenario does not make any sense with regards to how the calibration curves will be implemented. Thus, a constraint must be implemented that ensures the low throttle threshold percentage is always lower than the high throttle threshold percentage.

Our design will be implemented in such a way that the user will be able to increment these threshold percentages in 5% increments. The low throttle threshold percentage is not permitted to be less than 5% throttle and not more than 90% throttle. Conversely, the high throttle threshold percentage is constrained to the values with 10%and 95%. These boundary restrictions maintain that the throttle thresholds can always be achieved. For example, if the high throttle threshold was set to 100% throttle, due to integer calculations, there may be a chance that the high throttle calibration curve will never be used completely. This scenario occurs because the calculations within the DSP Software (to be discussed in the next section) may only yield 99% throttle due to truncations. So, in order to guarantee that the high throttle calibration curve can always be achieved, the AF Optimizer limits the range of the threshold percentage to 95%. The same argument is used to justify why the lowest percentage for the low throttle calibration curve is 5% and not 0%. Shown in Figure 3-2 below are the specified ranges for the threshold percentages that have just been described.

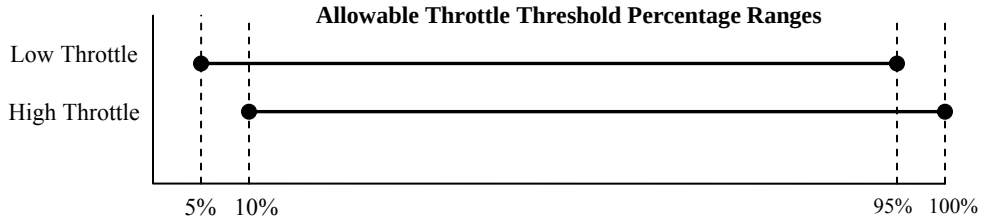

Figure 3-2: Throttle Threshold Percentage Ranges

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

In order the maintain the fact that the high throttle threshold percentage will always be higher than the low throttle threshold percentage, the AF Optimizer software will adjust the threshold percentage that is not being adjusted by the user. In the case where the user wants to increase the low throttle threshold percentage from 50% to 55%, but the high throttle threshold percentage is currently set at 55%, the AF Optimizer will allow the user to change the low throttle percentage to 55%, and it will automatically increase the high throttle threshold percentage to compensate; thus, ensuring that the threshold percentages are always separated by 5%. Again, this 5% separation is needed to eliminate discontinuities in the AF Optimizer's operations.

From below, which again shows how the throttle threshold percentages will be used by the interpolating DSP software, we can see that if the low and high throttle threshold percentage values were allowed to be the same, a discontinuity would appear as a vertical step between the two calibration curves. In the overall system, this means that if the driver were to continually increase their throttle position, the AF Optimizer would instantaneously switch between the low throttle calibration curve to the high throttle calibration curve. Potentially, this problem could manifest itself as a jolt in the continuous operation of the car's engine as seen in Figure 3-3.

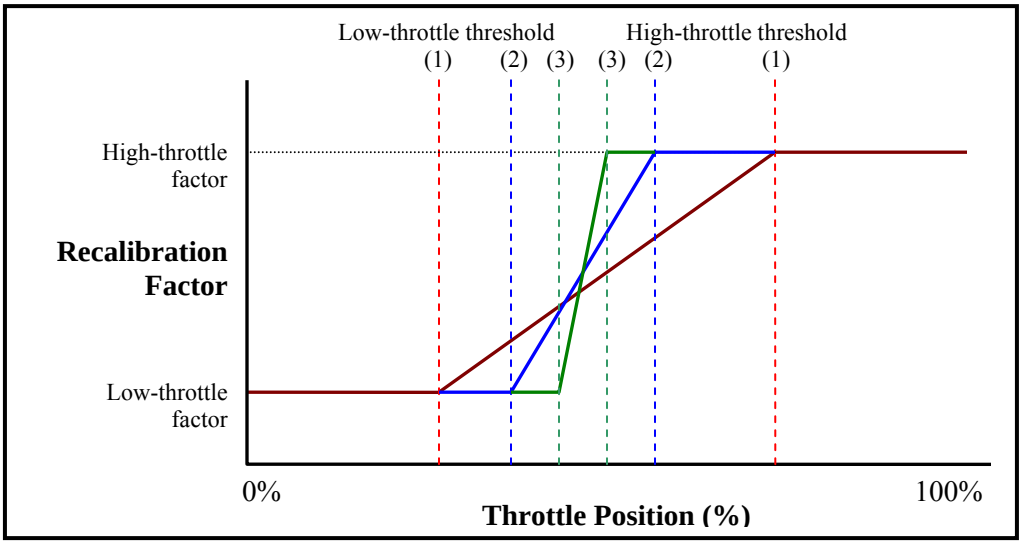

Figure 3-3: Throttle Position Calibration Interpolation

One can see, from the above figure, that as the throttle threshold levels become closer together, as they are in setting (3), the interpolation will be come much steeper. Eventually, if these threshold levels were allowed to be equal, the interpolation becomes discontinuous, and the recalibration factor would essentially jump from the low throttle factor to the high throttle factor at the particular threshold percentage.

The shift light RPM is the following item to be setup in this module. Initially, the AF Optimizer has the predefined shift light RPM of 5000 RPM. If the user so wishes, they can either increase this RPM up to 10000 RPM, or decrease it to 1500 RPM by using the "+" and "-" input buttons. The minimum resolution for the increments is 100 RPM, to allow a wide range of possible shift light RPM values.

In the final product, the Setup module is also where the user will define how many cylinders their particular vehicle has.

The next module in sequence is the Configure module, which allows manipulation of the calibration curves.

3.1.2 Configure Module

As previously stated, the Configure module is the module that gives the user the ability to actually change the calibration curves of the vehicles airflow sensor. From a functionality perspective, the AF Optimizer must allow for this recalibration to be a function of engine speed and throttle position. As a result, the user is given two calibration curves to manipulate; both of which are a function of RPM. Each calibration curve is throttle-dependent: a high throttle calibration curve and a low throttle calibration curve, where each curve will be used based on the throttle threshold percentages that were previously setup.

Shown in Figure 3-4 is the screen that allows the user to manipulate the calibration percentages in the engine speed range from 2000 RPM to 2750 RPM. The AF Optimizer software allows for both of the high and low throttle calibration curves to be viewed and manipulated at the same time. These curves can be seen as the 3rd and 4th lines on the display, respectively. This screen shot is the menu referred to as the "2000RPM Configuration Menu".

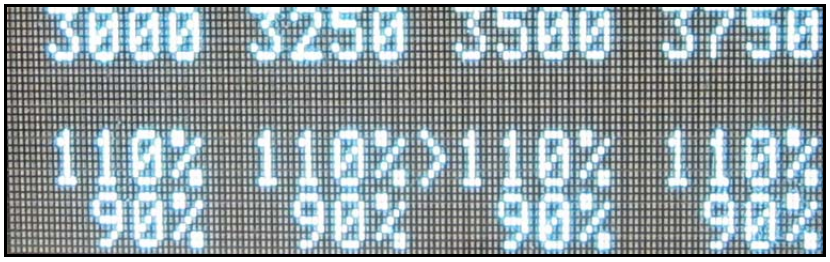

Figure 3-4: Setting the Calibration Curves Screenshot

The Configure module is structured as a continuous loop, allowing the user to recalibrate the airflow sensor from 2000 RPM up to 8750 RPM. A block diagram of the display screens is shown in Figure 3-5 below, which shows that the user can scroll through the different menus in a continuous loop. In this structure, the user is able to rotate through the menus in either a clockwise or counter-clockwise fashion.

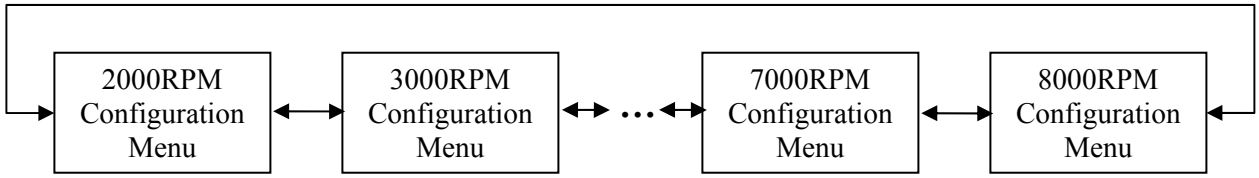

Figure 3-5: Configure Module Screen Navigation Structure

As seen in Figure 3-4, there is an arrow, designated by the ASCII character ">", which points at which configuration percentage is currently active. Each potential calibration point acts as one system state. Within this module, there are 56 system states since there are two calibration curves each having 28 calibration points between 2000 RPM and 8750 RPM. As we have implemented in the software, only one system state can be active at any particular time. The software uses these states to determine which calibration percentage to manipulate whenever the user hits the "+" or "-" input buttons. By keeping track of the current system state, the AF Optimizer is able to place the arrow ">" at the correct position on the screen and is able to read and write directly to the correct memory address once any calibration percentage changes have been made.

Our software has set limitations to the range of recalibration. To prevent the airflow sensor's voltage value from being changed drastically from what the car's ECU might be expecting, the AF Optimizer's limits of recalibration have been set from 75% to 125% of the original sensor's value.

In summary, this module is used by the user to scroll though and view the current calibration factors for each RPM range and throttle position. The user is also able to manipulate the factors by either incrementing them or decrementing them. Once all the factors have been set as desired, the user can choose to monitor their vehicle by toggling to the next module: the Monitor module.

3.1.3 Monitor Module

The third module of the display portion of the user interface is the Monitor module. This module gives the use real-time visual feedback of various vehicle systems. This module has been designed with two separate screens that can easily be toggled between. One of these screens displays information including the vehicle's engine speed, the current throttle position percentage, the vehicles battery voltage, and the O_2 sensor voltage. The other screen of this module shows the user the pre and post calibrated airflow sensor values as well as the actual calibration percentage that relates the two values.

An image of each of the screens from the Monitor module is shown below in Figure 3-6 and Figure 3-7. As it can be seen, the screens are laid out nicely so that the information can easily be obtained.

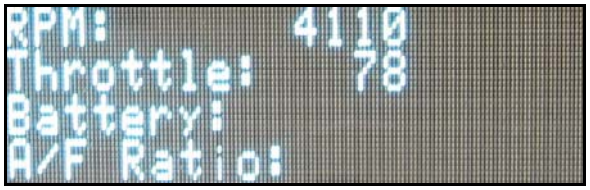

Figure 3-6: Monitor Menu One

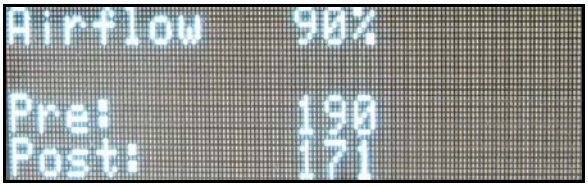

Figure 3-7: Monitor Menu Two

The information that is provided to the user in the Monitor module will prove to be useful when the car is being tuned, as it will allow the user to see the actual data that the PIC microcontroller is using in its calculations.

The previous subsections described the output software, and now we will describe the software for the input push buttons of the user interface.

3.1.4 Input Buttons

The input push buttons are a vital part of the user interface, since they allow the user to interact with the AF Optimizer. The buttons that will be used are designed to be active low momentary switches acting on the upper four bits of the PIC's PortB.

The PIC's PortB can be configured to trigger interrupts whenever any of the four most significant bits change state from either a low to high or high to low. This means that the PortB ISR is triggered each time a button is either pressed or released.

Since there is no need to run the ISR when the button is released, the software determines if the PortB has any low bits before determining which button has been pressed. If it has been determined that a button has indeed been pressed, the AF Optimizer logically tests the four bits to determine the actual button that is pressed. To the right is the flowchart, described in Figure 3-8, that describes the software to determine that button that was pressed.

This logical testing must be done since there are five buttons configured to switch only the four most significant bits of PortB. The actual circuitry needed for this configuration is described in the Hardware section below.

The momentary push buttons used in the AF Optimizer have a characteristic that they bounce when they are released. This bounce could be interpreted by the AF Optimizer as multiple button pushes, which could result in a value being incremented more than is desired, for example.

The software implementation of the AF Optimizer uses internal timers to delay the reading from PortB. This delay is designed to wait for a specified period of time, which will give the button enough time to stop bouncing. After which time, the PortB value is then read from the port, and the logical bit testing is then performed. To understand what the specific amount of time must be, refer to Figure 4-10 in the Hardware section, which shows the button bounce behaviour on an oscilloscope screen capture.

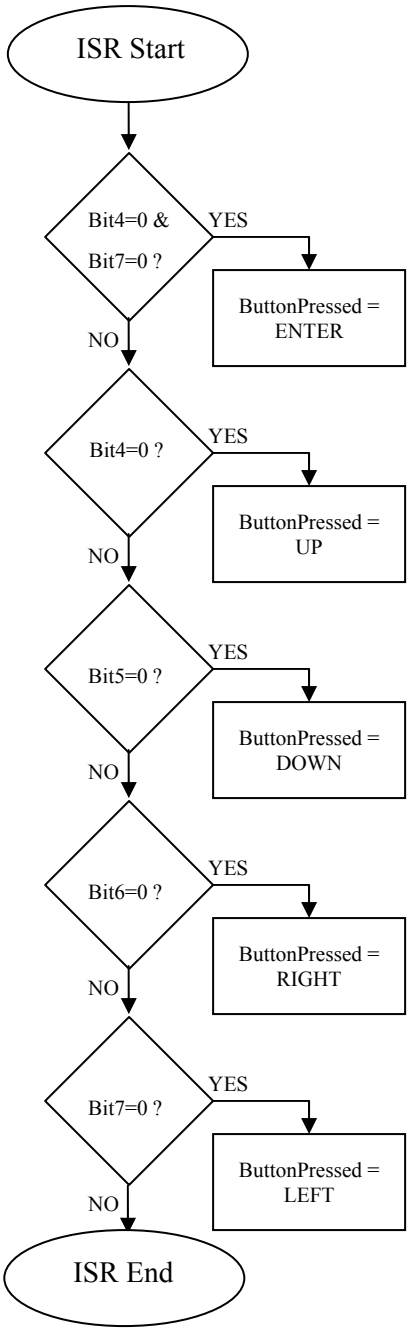

Figure 3-8: Button Bit Test Flowchart

3.2 DSP Software

The digital signal processing capabilities of our product are fundamental to its function. Since our product essentially introduces a non-linear, user-adjustable amplifier between the analog output of the vehicle's airflow sensor and the vehicle's ECU, we must perform an analog-to-digital conversion (ADC) on that signal, modify it accordingly, and finally return the signal through a digital-to-analog conversion (DAC). This entire process occurs quickly and accurately enough such that it meets and exceeds all timing requirements indicated in our functional specifications document. We have chosen to use digital signal processing (DSP) to meet the functional requirements of our product because it gives us a flexible working environment where the components required for implementation are both competitively priced as well as readily available. Furthermore, the software required to run our DSP hardware components is specifically written to function on the PIC16F877 microcontroller [3].

3.2.1 Input Signals

The input signals considered in the DSP software are the RPM signal, the Throttle Position signal, and the Airflow signal.

3.2.1.1 Determining the RPM

The RPM signal enters our microcontroller via PORTB0 where it is regulated to be a 5V peak-to-peak square wave signal generated by the pulsing ignition coil of the vehicle. Our software initiates an internal timer every time it senses a rising edge on the PORTB0 line. This timer is incremented by 1 for every 8 processor clock cycles, and its value is recorded and re-initialized to 0 on the next rising edge seen on PORTB0. Knowing the number of cylinders the vehicle has (in our prototype this is limited to 4), and knowing the number of coil pulses per engine revolution as well as the clock speed of our microcontroller, our product is able to calculate the RPM of the vehicle.

3.2.1.2 Determining the Throttle Position

The analog throttle position sensor signal enters our microcontroller via the built-in 10bit ADC [3] located on PORTA0. The 10bit value obtained is then adjusted to obtain a range of 0% -100% for our throttle position.

3.2.1.3 Acquiring the Airflow Sensor

The analog voltage of the airflow sensor enters our microcontroller via PORTA1 and it is then converted using another internal 10bit ADC. It is this signal that is then modified through our DSP software algorithm.

3.2.2 DSP Algorithms

The software DSP algorithm is shown in Figure 3-9. As it can be seen, the airflow signal is multiplied by a calibration factor that is determined by interpolating between the user-defined RPM calibration factor and present low and high throttle position points.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

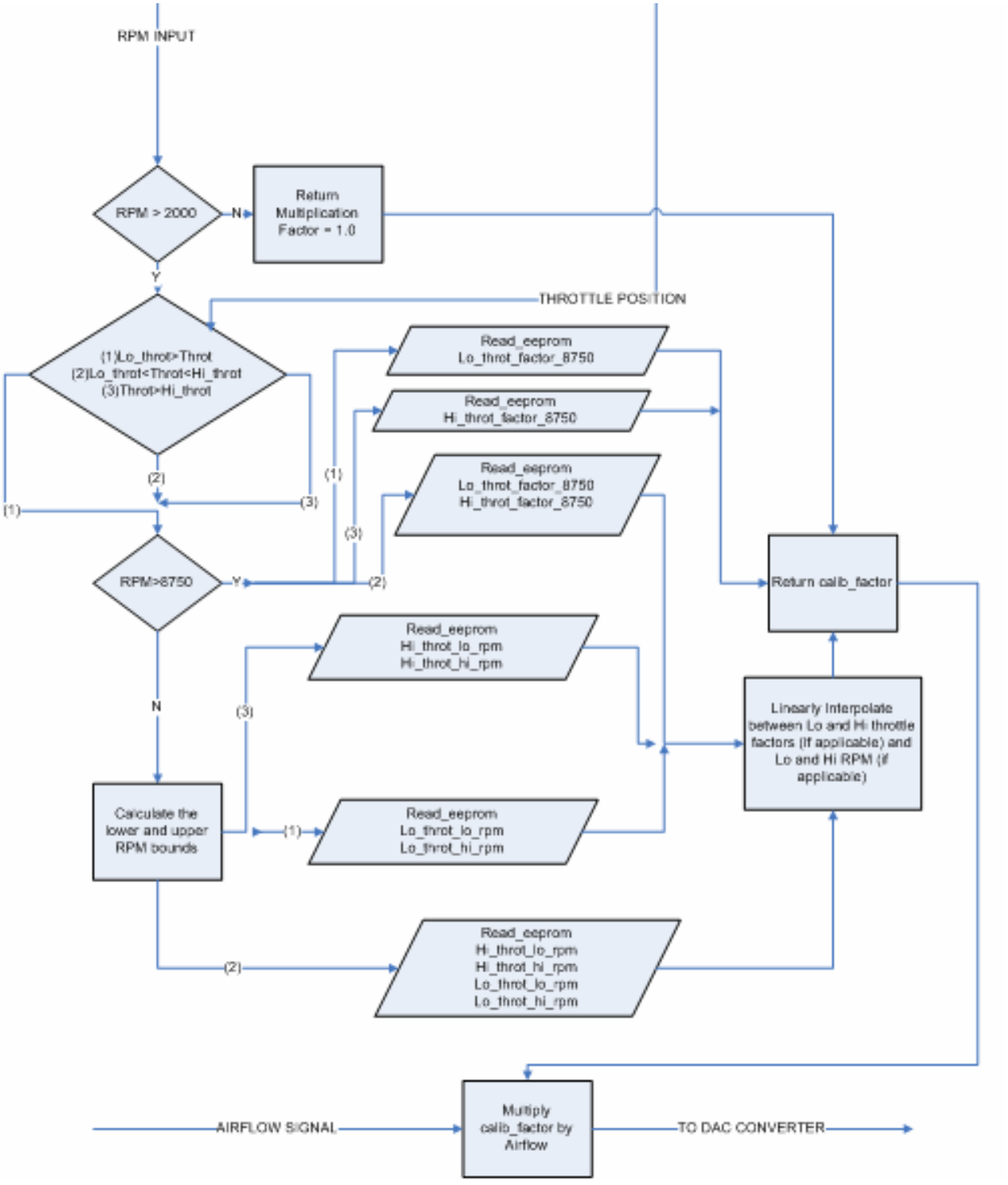

Figure 3-9: DSP System Flowchart

3.2.3 Calibrating According to RPM and Throttle Position factors

The user is able to define a low throttle and a high throttle calibration factor anywhere between 75% and 125% for each 250 RPM interval between 2000 and 8750 RPM. Furthermore, the user is able to define both the low throttle (Lo_throt) and the high throttle (Hi_throt) threshold points. Whenever the vehicles RPM is between 2000 and 8750 RPM, the system determines whether the car's throttle is below the low throttle threshold, above the high throttle threshold, or between the two threshold levels. If the car's throttle is below the low throttle threshold, the Lo_throt user defined calibration point is used to determine the airflow's signal calibration factor. If the car's throttle is above the high throttle threshold, the Hi_throt user defined calibration point is used to determine the airflow's signal calibration factor. If the throttle is between these two thresholds, a linear interpolation occurs between the low throttle and the high throttle thresholds. Figure 3-10 below displays this interpolating algorithm that has just been described.

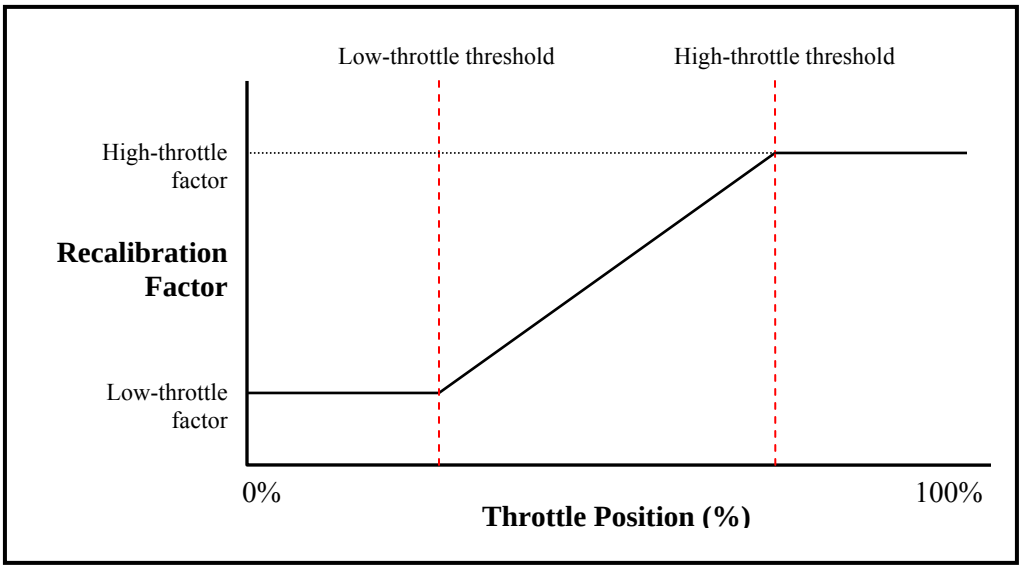

Figure 3-10: Interpolated Recalibration Factors as a Function of Throttle Position

Furthermore, when considering the car's RPM between the values of 2000 and 8750, the algorithm also interpolates between the upper (Hi_rpm) and lower (Lo_rpm) RPM user-defined bounds. For example, for an RPM value of 3111, the algorithm would interpolate between the user-defined calibration points for 3000 and 3250 RPM.

When the vehicle's RPM is below 2000 RPM, the airflow signal is not calibrated, but simply passed through the system unchanged. When the RPM of the vehicle exceeds 8750 RPM, only the Lo_throt and Hi_throt factors for the 8750 RPM user-defined calibration factors are used to calibrate the airflow signal for the rest of the vehicle's operational RPM range. Figure 3-11 graphically illustrates the how the high/low RPM are used to modify the airflow sensor's output by interpolating linearly between the two end values.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

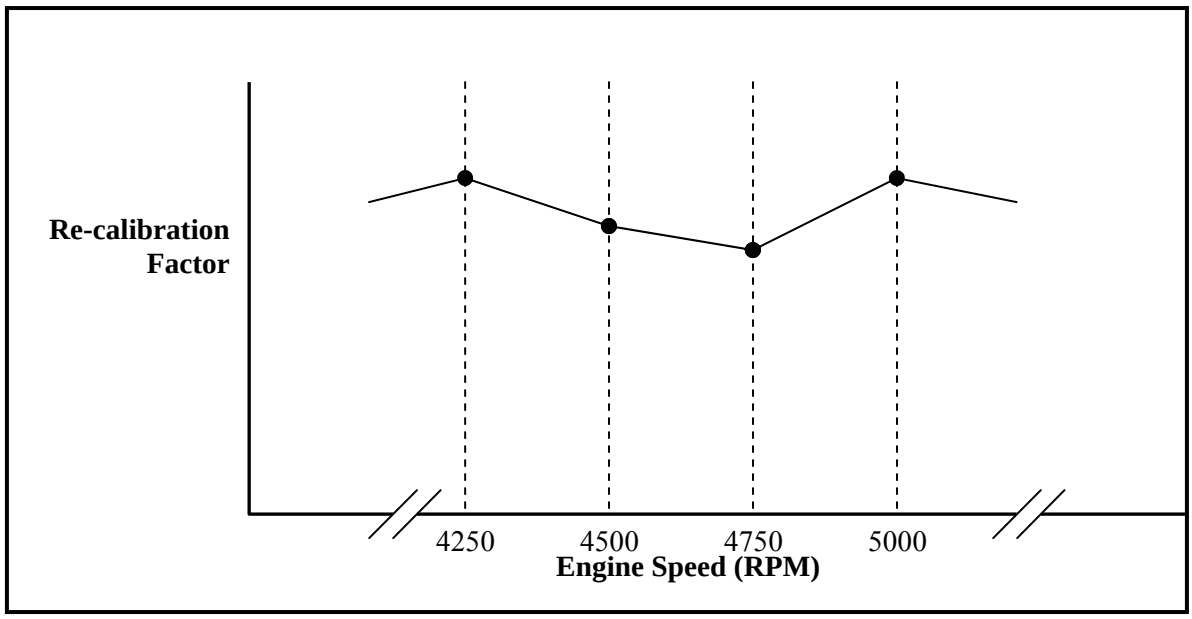

Figure 3-11: Interpolated Re-calibration Factors as a Function of RPP

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

4 Hardware Design

This section outlines design decisions pertaining to the hardware components of the AF Optimizer. The hardware components are grouped into for different sub-sections: Signal Input hardware, Microcontroller itself, User Interface, and Output hardware. From each section, it will be clear to designers how and why to incorporate the selected pieces of hardware.

4.1 Signal Input Hardware

The inputs to the AF Optimizer will require various modifications before they can be fed into the microcontroller. There are two reasons why this is necessary. The first is that the system must appear transparent to the rest of the car's systems, with the exception of the recalibrated airflow output. The second is that the PIC has various requirements on voltage levels and input current that if not adhered to, could result in permanent damage to the IC. The following sub sections will explain in detail the solution for each of the system's inputs.

4.1.1 Airflow input circuit

The airflow sensor is already an analog voltage signal, with a voltage range of 0-5V. Because the system will be interrupted already along this signal wire, only the functionality of the sensor and protection of the PIC must be considered. Because the PIC's input resistance is already very high, we will only consider protecting the PIC from any unwanted high-voltage noise created by the environment of the engine bay. To ensure that the voltage is kept below 5 Volts and at a low current, we will use a Zener diode in reverse bias with a breakdown voltage of 5V, as well as a resistor. Figure 4-1 outlines the necessary input circuitry for the airflow sensor.

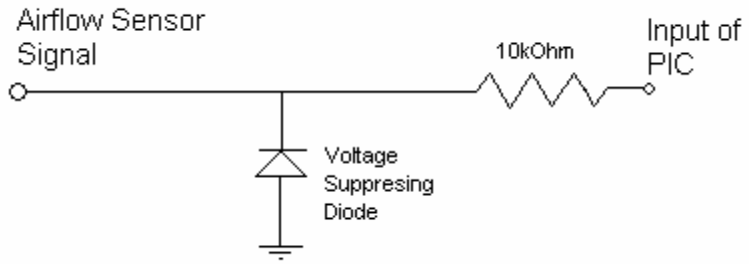

Figure 4-1: Airflow input circuit

4.1.2 Throttle Position Sensor

The throttle position sensor is a potentiometer across a 5V supply. That is, the signal will potentially vary from 0V to 5V. The input circuitry will consist of only a current limiting resistor. Figure 4-2 outlines the circuitry needed for the throttle position sensor.

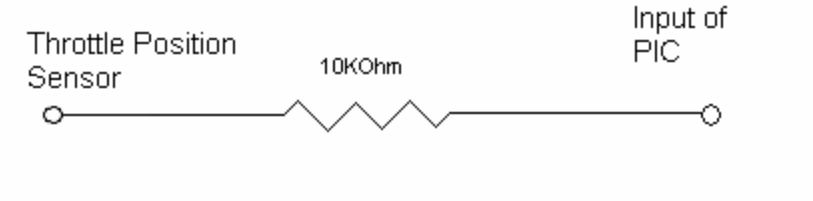

Figure 4-2: Throttle Position Sensor input circuit

4.1.3 Tachometer Input Signal

The signal used to read the RPM into the PIC requires more elaborate input circuitry. This is because the signal can surge to up to 50V as part of an underdamped square wave at a 12V peak . The PIC inputs do have some protection from light surges, but we feel that they are inadequate and should not be relied on for this signal. As such, we will set up two more "barriers" of protection. The first will be a voltage suppressor, and second will be a 10 kΩ resistor. The voltage suppressor SA5.0A can suppress of to 500W of surge power [6]. Unfortunately, our lab results showed that there were still potential voltages of 7 to 8V, higher than the PIC's allowable maximum of VDD+0.3V, or 5.3V. For this reason we will add a 10 kΩ resistor. Figure 4-3 depicts the tachometer sensor input circuit.

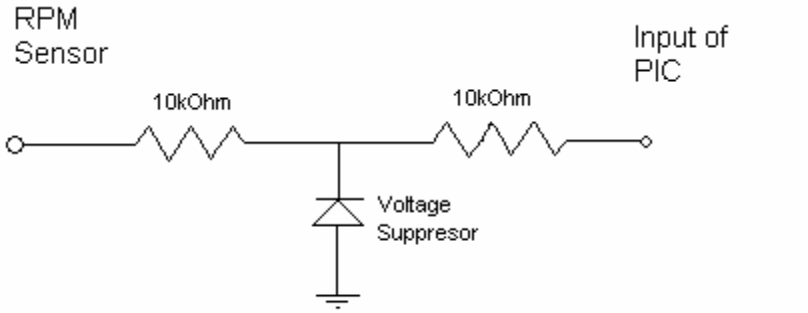

Figure 4-3: RPM Sensor input circuit

It should be noted that the PIC16F877 also has protective clamping diodes on every input. These diodes are capable of sourcing/sinking ±20 mA of current when the input voltage is less than 0, or greater then 5V [3].

4.1.4 Battery Voltage input circuit

The battery voltage will be measured only for the purpose of providing monitor data to the driver. The battery voltage will be measured simply by taking the voltage as an analog input after going through some form of downward voltage scaling. To accomplish this we will use a voltage divider circuit. The battery voltage does not need to be measured to a high degree of accuracy, so a simple voltage divider network with resistors of a 1:3 ratio will suffice. We will use a resistor value of 10 kΩ in series with the battery voltage, and a 3.3 kΩ resistor to measure across. This will result in a simple, accurate circuit to suit the needs of the AF Optimizer. This is shown in Figure 4-4.

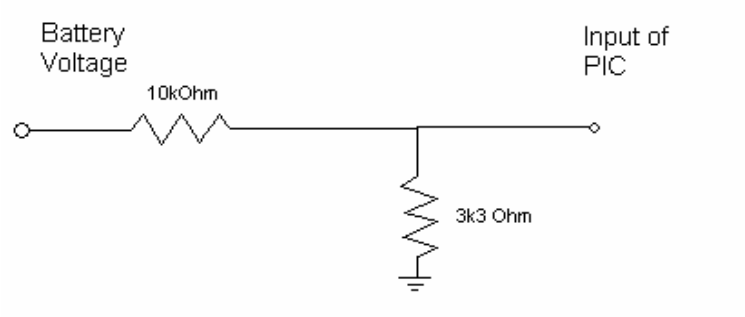

Figure 4-4: Battery voltage input circuit

4.1.5 Oxygen Sensor Input

This circuit will be a low pass circuit; a capacitor (47uF) and a resistor (5M6) to ground. This circuit will prevent unwanted noise in the higher frequencies from reaching the ADC of the PIC. The resistor will allow the capacitor to discharge as the O2 sensor voltage drops, and will slow the charging of the capacitor. Another reason for implementing this circuit is that the oxygen sensor itself works on a very fast switching lambda system. When the air/fuel ratio deviates from the O2 sensor's specified ratio, the voltage swings rapidly up or down. Thus, the sensor allows immediate and obvious feedback to the ECU about how the air/fuel mixture needs to be adjusted. However, since the signal can switch very rapidly, our design will implement the low pass filter to suppress these sensor characteristics. Figure 4-5 details the necessary input circuitry for the oxygen sensor input.

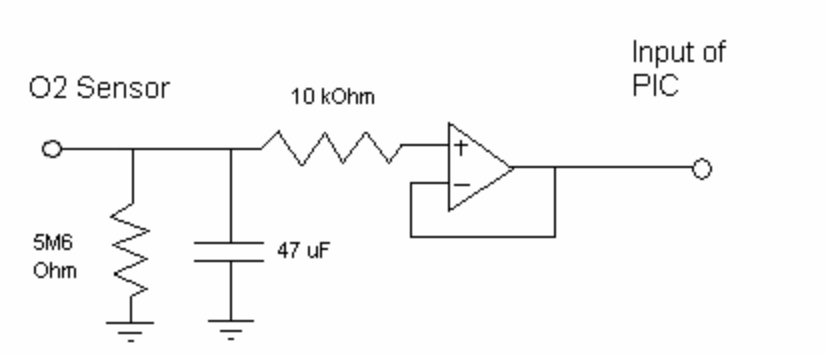

Figure 4-5: Oxygen sensor input circuitry

4.1.6 Power Supply

The entire AF Optimizer will operate off of a single 5V power supply. In the automotive environment, the standard DC operating voltage is 12-14 VDC. Our design will employ a LT323A 5V DC Regulator, capable of delivering 3 Amps of current [5]. This current supply will be more then sufficient for the AF Optimizer, as lab results have shown the maximum current draw to be less than 2 amps, and an average current draw to be around 0.4A. Based on our projected load of 0.4A, we will employ a 1000uF filter capacitor on the 5V line. A problem associated with any voltage regulator is heat dissipation. We will use a large aluminium heat sink, and will locate this component near many ventilation holes through the casing of the AF Optimizer.

The output of the power supply regulator is low in noise and stable over the entire operating voltage of a car (~12V when the car is off, ~14V when the car is running). The only other consideration given to the power supply is the need to run the car's 12V power rail to the battery voltage input circuit (see Section 4.1.4)

4.1.7 Calibration Mode

The AF Optimizer will have two modes of operation: a Calibration ON mode and a Calibration OFF mode. Depending on the mode selected, the AF optimizer will either recalibrate the airflow sensor or pass the sensor value through transparently. By implementing a hardware switch, the system will be able to multiplex the two airflow signals; the original signal and the recalibrated signal. The user will then be able to decide which mode of operation they choose.

Additionally, there is an advantage of using a hardware switch rather than a software switch. This advantage is that if the AF Optimizer's microcontroller malfunctions for whatever reason, the user will still be able to change the mode of operation. The switch will be located on the casing to allow the user to switch modes conveniently. Figure 4-6 shows the block diagram of this switching implementation.

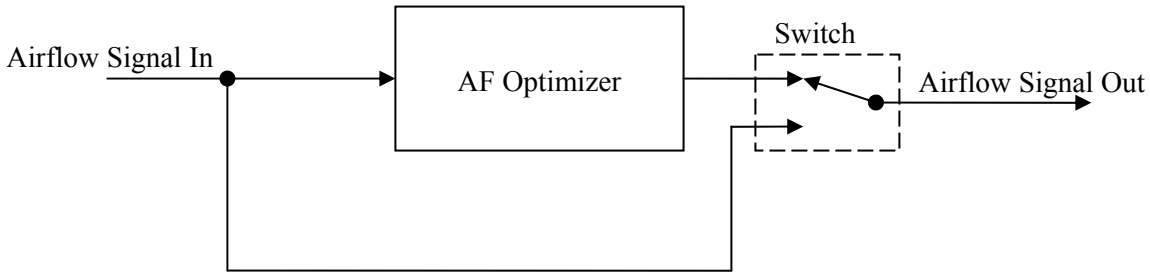

Figure 4-6: Calibration Mode Switching Diagram

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

4.2 Microcontroller

The AF Optimizer has many complex functions that must be handled with one or more microcontrollers. There are many needs of the AF Optimizer that will be considered while choosing a particular microcontroller. These project needs will be discussed, providing strong evidence for making the microcontroller choice we have made.

Firstly, the fundamental purpose of the AF Optimizer is to recalibrate a vehicle's airflow sensor. In order to do this, we must have a controller that is able to determine the vehicles engine speed, throttle position, and current airflow sensor value. We will discuss the needs of a microcontroller to perform this basic, yet fundamental, function.

The AF Optimizer's microcontroller must have analog-to-digital converter capabilities, and it must be able to have edge-triggered interrupts. Ideally, the microcontroller would also have digital-to-analog converter (DAC) capabilities; however, there are many integrated circuit chips that can be bought to perform this function. The microcontroller must also be able to process the signals quickly to maintain as true of an analog signal at the output as possible, meaning it must be able to operate at a fast clock speed.

The other parts of the AF Optimizer include being able to display information to an output VFD. Communication must be done via a serial or parallel connection with the two components. Our microcontroller must be able to handle interrupts that are triggered by the input buttons, and it must be able to turn on LEDs for a shift light.

The PIC16F877 is able to meet these project requirements of a microcontroller; it has 8 analog-to-digital connection pins, edge-triggered and button interrupt capabilities, and it can communicate with our VFD with a UART. Additionally, this PIC microcontroller has 3 internal timers that we will be able to use for various functions in determining the vehicle RPM. We were unable to find a competitive controller that provides a DAC, so the AF Optimizer will use a chip that is specifically designed for this purpose.

Figure 4-7, on the next page, shows the pin-out diagram of the PIC16F877 [3], where the coloured pins show all the connections that are made to external circuitry.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

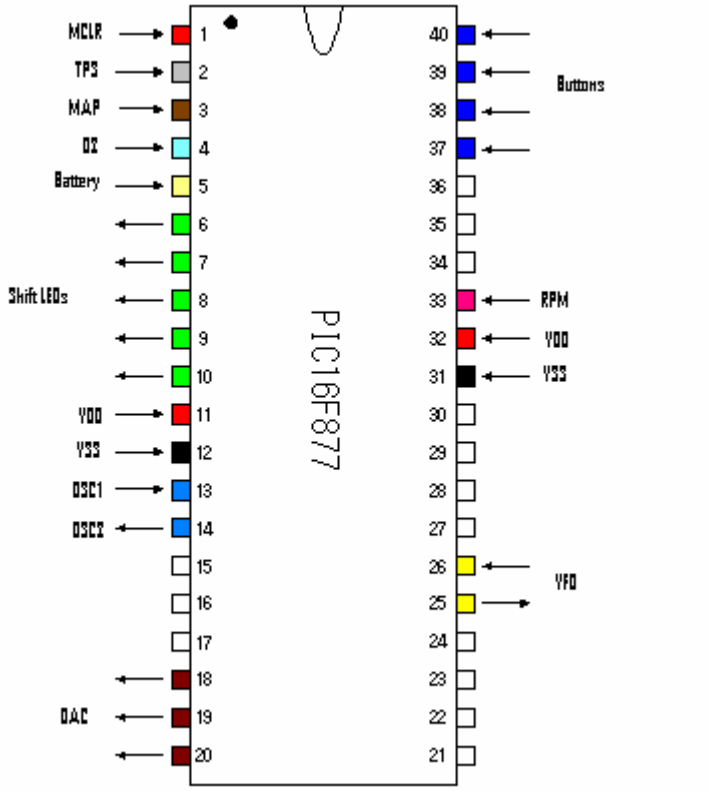

Figure 4-7: PIC Pin-out Diagram for the AF Optimizer

4.3 User Interface Hardware

The user interface is a crucial part of the AF Optimizer that will distinguish this product in its competitive market. Our design for the user interface consists of a vacuum fluorescent display and five menu navigation buttons. The following section details the engineering design choices that were made for the user interface accompanied with the reasons for each decision.

4.3.1 Display Characteristics

The visual display of the user interface needs to be able to relay information to the user with regards to the system it is monitoring. It must also give the user the recalibration information so they can properly recalibrate their vehicles airflow sensor. With these functionality requirements, ACS made a design decision to use a display that had 4 separate lines that could display text.

The monitoring capabilities of the AF Optimizer is limited to monitoring the engine RPM, the throttle position, the oxygen sensor voltage and the battery voltage. The four lines on the display can be used separately and would allow the user to see all four monitoring values on the same screen.

Due to the requirements of having to give the user the information to properly tune the display we designed for has 4 lines of display where each line can display 20 ASCII characters. The Noritake GU140X32F-7002 VFD has been chosen as it gives us the flexibility to communicate with it via a serial connection or a parallel connection. This VFD is shown below in Figure 4-8.

Figure 4-8: Picture of G-7002 VFD

4.3.2 Display Communication

The PIC16F877 has a built in USART, and we have implemented the communication to the VFD with this feature. More specifically, we are implementing communication asynchronously; thus, we are using the UART with a transmit register, Tx, for sending bytes to the display and a receive register, Rx, for receiving bytes from the display.

The transmit byte from the PIC will contain the data to be written to the display screen, while the receive byte received by the microcontroller will be an SBUSY signal from the VFD allowing the microcontroller to know when the VFD is ready to receive the next byte of data. Our software implementation uses these two bytes of data whenever we decide to write information to the display.

4.3.3 Shift Light Output

The shift light output is another output from the user interface that gives the user valuable information for desired shifting points. Through the output of 5 light emitting diodes (LEDs), the AF Optimizer will tell the user when their desired shift point is approaching. By using the RPM engine speed that is calculated by the AF Optimizer and using five output pins on the PIC, the five LEDs are turned on in sequence. Each LED is turned on in increments of 100RPM starting from 500RPM below the desired shift point. As the actual RPM is within 100RPM of the desired shift RPM, all five LEDs will be turned on. Then as the engine speed passes the desired shift RPM, all LEDs will be turned off, allowing the user to easily see that the car is revving higher than desired.

The reason for turning all LEDs off, once the desired shift point has been attained, is because the act of having all LEDs turn off at the same time will easily catch the attention of the driver, even if they are not looking directly at the LEDs.

4.3.4 Button Characteristics

The user interface will contain momentary buttons to allow the user to control both the recalibration factors of the air flow sensor, and to navigate through the menu structure of the display. This implementation will have the use of five momentary push buttons oriented as seen in Figure 4-9 below.

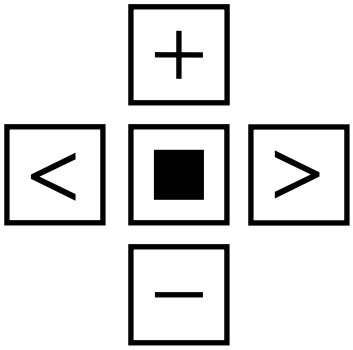

Figure 4-9: User Interface Button Configuration

This layout of the buttons will make the functional purpose of each button intuitive to the user. The leftmost button will allow for navigation to the LEFT, while the rightmost button will allow for navigation to the RIGHT. The buttons on the top and bottom will allow the user to either increase or decrease, respectively, their recalibration factors. Finally the centre button will be used to change the current menu. Since our software has three distinct modules (Setup, Configure, Monitor), the centre button will be used to scroll through the different modules of our system.

From the functional specification of our AF Optimizer, we have said that these buttons will be located on the front of the casing to allow the user to have easy access to their functions [2]. This is the current design implementation of the interface.

4.3.5 Button Bounce

One problem that arises when using momentary push buttons is the transient response that occurs when the button is pressed and released. This transient response follows an underdamped response, which means for an amount of time after the button is initially released, the button will switch on and off many times until it reaches its steady state. A graph showing the output voltage of an active low button that is bouncing is seen below in Figure 4-10.

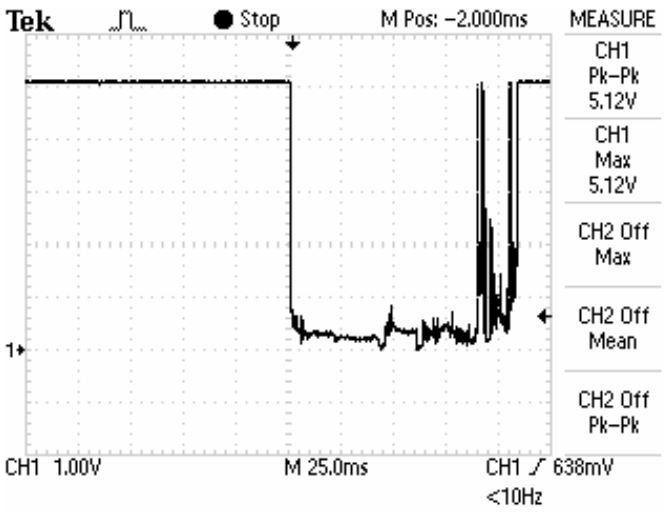

Figure 4-10: Button Bounce Screen Capture

The result of a button bounce is that the program software will detect multiple button presses. So, when the user pushes the button just once, the program will see many pushes, which will result in the program performing the specific button operations many times. As an example, the user could be trying to increase their recalibration factor from 100% to 101%. If the button bounces a few times, then the result could be that the recalibration factor increments past 101% and could count up to 105%, for example. This operation is clearly undesirable, so our software implementation must solve this issue. By using the internal timers of the PIC, we are able to solve this, as was described above.

4.3.6 Button Circuitry

The PIC16F877 has the feature that allows its PortB to be used as a state-change interrupt register [3], which is controlled by the upper four bits: bits 4-7. This means that we can configure an interrupt to occur whenever any of the upper four bits change state from either 1 to 0 or 0 to 1. By implementing this feature, we are able to connect the five momentary navigation buttons to the four upper bits of PortB and have them trigger the start of an interrupt subroutine.

As we have mentioned, there are only four bits from PortB that will trigger this interrupt subroutine; however, our button input interface requires the use of five momentary buttons. So, logic circuitry has been designed to allow each of the five buttons to trigger a combination of the 4 bits from PortB. Then software will decode the bit combination to determine which button was actually pressed.

Using diodes to prevent the connection lines from passing current in both directions, we implemented the button circuitry as seen below in Figure 4-11. Note that this implementation configures the buttons to be active low.

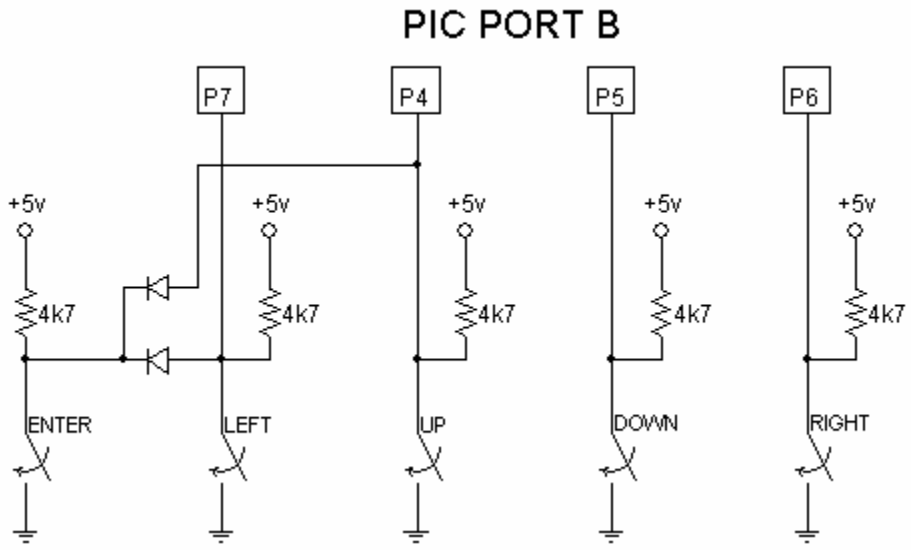

Figure 4-11: Input Button Circuitry

Now we have the circuitry needed to create the button user interface with the PIC microcontroller. As mentioned, the software contained in the interrupt subroutine will logically determine which button was pressed, allowing our program to continue along the desired path of the user.

4.4 Output Signal Hardware

The output signal hardware is built around the Maxim Semiconductor's MAX5254 10-bit Digital to Analog Converter. This DAC was chosen because it has a fast settling time of 10µs, works via serial communication, and it has 10-bits of accuracy between 0 and 5V [4]. This accuracy results in a resolution of (5V – 0V) / 2^{10} = 5mV. This will meet our functional specification of a 20mV resolution ±10mV [2].

Figure 4-12 outlines how the MAC5354 is to be connected. We will use this exactly, with a Reference Voltage of 2.5V supplied by an Analog Devices SO23-5 Voltage Reference IC. Pin 1, the DAC Output voltage, will have a 0.1uF capacitor coupling it to ground. This is smaller then the suggested value of 47uF suggested by the datasheet for the DAC It will improve noise immunity on the output. We chose a smaller value because the output voltage slew rate with a 47uF capacitor resulted in a slew rate of 2V/ms (see Figure 4-15), which is too slow for our application. The DAC Output Amplifier Feedback will be connected to both the output and ground via two 8K2 resistors. Positive Power Supply and Ground will be connected across our 5V Chip-Select, Serial-Data Input, and Serial-Clock Input all are involved in serial communication, and will be connected to three pins of the PIC's output ports.

PIN	NAME	FUNCTION
1	OUT	DAC Output Voltage
2	$\overline{CS}$	Chip-Select Input. Active low.
3	DIN	Serial-Data Input
4	SCLK	Serial-Clock Input
5	FB	DAC Output Amplifier Feedback
6	REF	Reference Voltage Input
7	GND	Ground
8	V_{DD}	Positive Power Supply

Figure 4-12: DAC Connectivity

The serial communication to the DAC requires a 16-bit word. Figure 4-13 outlines the format of the 16-bit word. In our application, the control bits will be "000", because we want the input word to be immediately loaded to the DAC register and updated immediately. The following 10 bits are the data bits, as calculated by the DSP algorithm in software. They are outputted from MSB to LSB. After the 10 data bits have been sent from the PIC to the DAC, the three sub-bits, which are "don't care" bits, complete the 16-bit word. For simplicity, we simply output "000" for these bits.

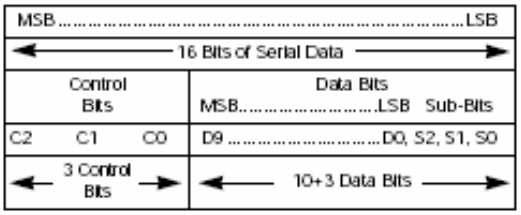

Figure 4-13: DAC Serial Input Word

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

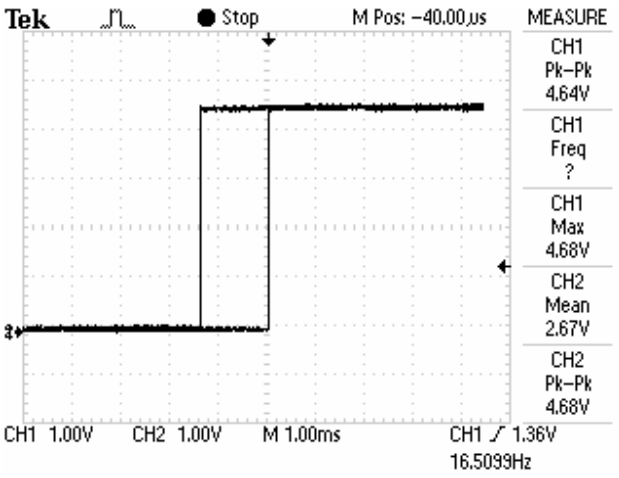

Figure 4-14: Step response of system with DAC ouptut capictor of 0.1uF

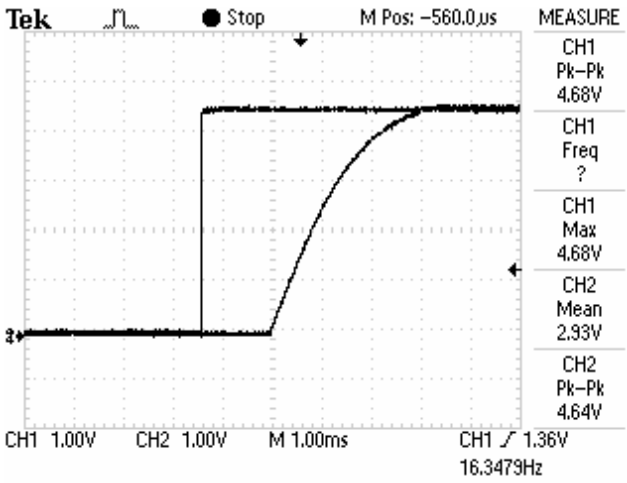

Figure 4-15: Step response of system with a DAC output capacitor of 47uF

Figure 4-14 and Figure 4-15 show a comparison of the step response of the AF Optimizer using different output capacitor values. As can be seen in Figure 4-14, the output settling time meets the requirement of 20us set out in the functional requirement. By measuring this step response, we measured an output slew rate of 0.24 V/µs, which is 240 V/ms. When a capacitor value of 47µF is used as specified by the data sheet, the output slows down to unacceptable levels: 2 V/ms.

5 Optimization – Timing Considerations

The speed with which we are able to accurately respond to a change in the airflow signal value is essential to the proper operation of our unit. Taken into consideration is the response time of the AF Optimizer, the slew rate of the output signal, and finally, how accurately we can follow a sinusoidal input. If we meet acceptable standards for all such figures of merit of our DSP, it meets the timing requirements outlined in our functional specification document. During interruption-free operation of the unit (when the display is not refreshed and no buttons are pressed), the response of the unit to a 16Hz sinusoidal input is shown in Figure 5-1. The response of our unit under these ideal conditions exceeds our proposed output timing requirements for speed.

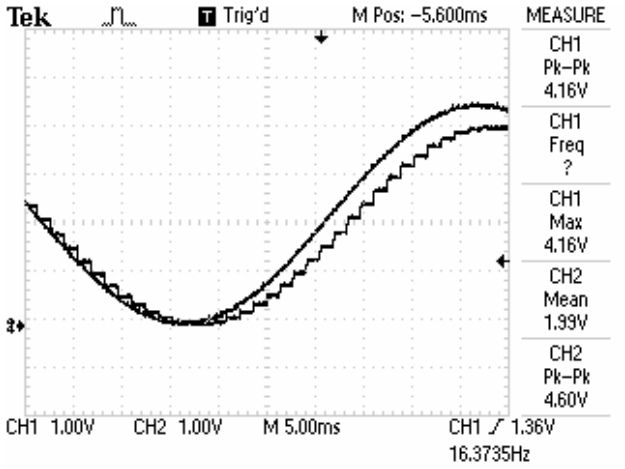

Figure 5-1: AF Optimizer Tracking an Input Sinusoidal Signal

Whenever information is refreshed on the VFD, the function of the microcontroller is interrupted in order to service its serial interface ports. During this time, the output of the unit is held at a constant value during the time of this service. Furthermore, whenever the microcontroller is interrupted by a button being pressed, the output of the DAC will once again be held at a constant value while the microcontroller services the particular interrupt. Such an interruption in the DSP functions of our unit results in less-than-ideal responses to changes in the airflow signal input to the unit. This is a great concern when the AF Optimizer is in its monitoring mode and information such as the vehicle RPM is constantly updated on the VFD.

There are several methods that would improve such a situation. First of all, the microcontroller is run at its highest allowable clock frequency of 20Mhz. Secondly, the baud rate of the serial interface between the microcontroller and the VFD is also run at its highest allowable speed of 115200.

As far as improvements to the software are concerned, several optimizations can further improve the response time of our unit. One of these optimizations entails limiting the refresh rate of the screen in monitoring modes to a rate that does not exceed 10 refreshes per second. A higher refresh rate would harm the timing response of our microcontroller while not giving the user any more information.

However, in order to ensure the interrupt-free operation of our DSP, the implementation of two microcontrollers is required. One microcontroller would solely act as a DSP while the other would be used for all user interface and monitoring requirements. Communication between these two microcontrollers would occur only when the DSP is employed (when the car is off or idling).

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

6 Product Verification Plan

As the product nears completion of each stage, an iterative product verification plan will be executed. This plan will ensure that before the final product is designed, the prototype product met or exceed all requirements contained in this document applicable to the prototype. Testing will focus on output accuracy and reliability once it has been installed in a vehicle.

6.1 Test Overview

The test regime can be broken down into two main categories. The first, accuracy and stability of the output, will ensure that the output of the AF Optimizer is accurate, timely, and suitable for input into the car's ECU. This test is to ensure that the product functions correctly to perform the task of recalibrating the airflow sensor voltage. The second, reliability once installed into a vehicle, will ensure that under all situations foreseeable in a customer's car, the AF Optimizer will perform correctly. This will ensure that any situations encountered in a vehicle that are not encountered in the lab are handled gracefully by the AF Optimizer. As stated in the functional specification, the unit will have an output that performs within a 0-4.99V range, with a resolution of 20mV $\pm$ 10mV. Reliable operation testing will include ensuring the unit will operate under various start-up conditions, and timing.

6.2 Functional Test

This first main test procedure will occur on every unit produced by ACS. It ensures that in a lab setting, the unit behaves exactly as expected. This will prevent product having a manufacturing defect from making it to customers. The following checklist will be performed in the following order on a fully assembled AF Optimizer:

- Ensure proper connectivity of all components by using a visual inspection of the circuit board and enclosure assembly
- Connect a 12V DC power supply and ensure the unit powers on, and that the voltage regulator is producing a clean, stable 5V DC output for the full operating range in a car (11-15VDC)
- Navigate through the menu to test the operation of each button
- Apply a tachometer signal using a function generator, square wave, 100Hz, to simulate an engine running, apply a 0-5VDC supply to the throttle position sensor. Ensure RPM is read correctly, and throttle as a percentage.
- Apply a 0-5VDC airflow sensor signal and observe the output, ensure it tracks the input with the correct multiplicative power from the DSP. Ensure the output is stable, fast, and within the accuracy specification set out in the functional specification

These tests will be performed in order. A failure of one test will end the test procedure, and an appropriate troubleshooting strategy will be employed.

6.3 In-car Test

This test procedure will be performed on the prototype stage, and the early production units, but not on every unit scheduled for sale to customers. It will be performed only on units that have successfully completed the functional test. It is purely to test the design of the AF Optimizer, not the manufactured quality. This test is needed due to the conditions in a car differing from those used to test the unit in the lab. For example, the tachometer signal will be a noisy square-wave with an

abnormal duty cycle. As well, the unit will be tested with the car running to observe any irregularities in the operation of the vehicle. The following test procedure will be employed:

- Connect all sensors and power supply to the AF Optimizer
- Turn car ignition to "on" position. The unit should turn on, with the VFD indicating the vehicle is "off". The output will not be tracking the airflow sensor input and the unit will not output any monitoring signals.
- Start car. Car should start normally, and output should immediately track the value of the airflow sensor and the VFD should be displaying the monitoring values (RPM, throttle %, etc).
- Drive car through conditions of no throttle, maximum throttle, and an intermediate throttle value. The car should operate smoothly without problem.
- Test hardware bypass switch to disable the AF Optimizer output, and allow the original airflow sensor to pass through to the ECU.

If the unit passes all of these tests, in addition to meeting all functional requirements previously, it will be deemed fully operational and ready for manufacture and use.

School of Engineering Science
Burnaby, BC
ACS-ENSC440@sfu.ca
http://www.sfu.ca/~bnelson

7 Conclusion

ACS is poised to attack a massive multi-billion dollar aftermarket automotive parts market. The world of aftermarket parts is youthful by nature, and consumers are demanding quality upgrades with solid engineering design for competitive prices. The AF Optimizer will satisfy all these demands, and will find its way into the dashboards of many enthusiasts. The benefits of the product will make it an easy sale: it can be used to create custom fuel maps; the cost will be a fraction of our competitors; the display provides a usable interface with useful feedback; and it will be a non-invasive install.

This document will be a reference for engineers and management alike. Not only does it define the design specifications to which AF Optimizer will be designed, but in writing it the design team has given themselves a reference from which to design from, as well as an end-target for this exciting product. Adhering to the design choices herein will result in a product engineered to a high standard of quality, functionality, safety, and value in a short period of time. Also, implied in the collection of requirements is the foundation of a product verification plan for the prototype, final product design, and final product manufacture.

Our team members have experience in both the engineering realm and the world of automotive. The project will demand a combination of hardware and software development, as well as a novel integration into a test vehicle. With thorough testing and precise programming, this unit will have proven reliability through its robust design. The aspirations and extensive skill of the ACS team will ensure that this product comes to fruition in a timely manner and within the projected budget.

Car enthusiasts will be able to use our product, through professional tuning, to realize a better performing car. Whether it is a car producing more power, or a car with better fuel economy, this unit will prove its versatility and ease of use to the $29 billion market [1], while maintaining itself at an affordable price point for all consumers.

8 Referenced Documents

[1] The Automotive Control Solutions Team. Proposal for ACS AF Optimizer.

[2] The Automotive Control Solutions Team. Functional Specification for ACS AF Optimizer.

[3] Microchip Technology Inc. "PIC16F87X Data Sheet." <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf> 2001.

[4] Maxim Semiconductor Inc. "MAXIM 10-bit Voltage-Output DACs" <http://rocky.digikey.com/WebLib/Maxim/Web%20Data/MAX5354,MAX5355.pdf>. 1997.

[5] Linear Technology Inc. "LT123A/LT323A 5 Volt, 3 Amp Voltage Regulator." <http://rocky.digikey.com/WebLib/Linear%20Tech%20Web%20Data/LT123A,%20323A,%20LM123,%20323.pdf> 2000.

[6] Fairchild Semiconductor Corp. "SA5.0(C)A - SA170(C)A" <http://www.datasheetarchive.com/datasheet/pdf/78/78124.html> 1998.