Automotive
Control
Solutions

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

April 13, 2005

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

Re: ENSC 440 Project Post Mortem for the AF Optimizer

Dear Mr. One:

Please find the attached document, *Automotive Control Solutions' AF Optimizer –Post Mortem* which outlines the end of our project for ENSC 440. The included document contains information about where the project stands, and how it ran for the last three months. The ACS team is beyond the prototype development stage of the AF Optimizer, a low-cost, flexible air/fuel controller. The next stage for the ACS team will be working in conjunction with the University's Industry Liaison Office to approach marketing this product.

The purpose of this document is to provide an overview of the current state of the device, problems encountered during development, and future plans for the device. In addition, we compared our estimated and actual project schedule and financial plans, as well as the marketability of our product. Each group member also discussed the technical experienced gained through this project.

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 –Post Mortem

Automotive Control Solutions'
AF Optimizer
Post-Mortem

Version 1.0 (April 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

Abstract

This document outlines in further detail the current status of the ACS AF Optimizer Project, difficulties encountered, as well as possible improvements to the project. Details will be presented as to where our project is currently sitting with respect to the software, the hardware, and the in-car integration and testing.

Problems of software debugging and product integration were encountered throughout the AF Optimizer's development. These issues will be reviewed, as well as providing the solutions that were implemented to overcome the problems.

ACS has received some non-technical user feedback, which has provided the team with future development plans. The future development of the software, the circuitry, and the casing will be discussed so that we will be able to deliver a respectable product to the market.

This document will conclude with a comparison of our projected budget and timeline with the actual figures, and it will close with the final thoughts of each of the team members: Russell Potter, Alex Gutica, and Brian Nelson.

Table of Contents

List of Figures

List of Terms

ACS	Automotive Control Solutions
DAC	Digital-to-analog converter
DSP	Digital signal processing
Dynamometer	Dyno for short, this is a machine that measures a vehicles output power, while logging information such as air fuel ratios
ECU	A vehicle's electronic control unit
PCB	Printed circuit board
PIC	Used in this document to refer to the PIC16F877 microcontroller
RPM	Revolutions per minute, engine speed
TPS	Throttle position sensor
VFD	Vacuum fluorescent display used in our user interface

1 Introduction

Automotive Control Solutions has successfully designed, implemented, built, and tested our product: the AF Optimizer. The AF Optimizer is a production-ready solution that allows automobile enthusiasts to modify their vehicle's air/fuel ratios. Over the past four months our group composed of an electronics and two systems engineers has been devoted to the completion of a device that not only served the purpose of modifying a vehicle's air/fuel ratios, but did so reliably, and furthermore provided additional vehicle sensor monitoring capabilities.

This document describes the features and functionality of our completed prototype, the problems encountered during the entire course of the project, as well as our future plans for the AF Optimizer. Team performance is evaluated as a whole along with a detailed budget and timeline analysis after which each member of our group shares his personal experiences gained through being part of this group and bringing this project to completion.

Our goals for the prototype of the AF Optimizer have been met in spite of the obstacles that had to be overcome along the way. The fundamental firmware architecture has undergone many design revisions before meeting our proposed requirements. The hardware, although not entirely refined, serves its intended purpose, and the overall product, although not in its final stage of development, has performed within specifications in the testing stages. Overall, we have delivered a fully functional prototype that fulfills all the requirements we proposed and that can easily be transformed into a marketable product.

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

2 Current Project Status

The AF Optimizer prototype has successfully proven itself through its various abilities to its designers and potential users. With the firmware code written to implement a 2-dimensional interpolation of engine speed and throttle position, the AF Optimizer is able to successfully recalibrate a vehicle's airflow sensor. These recalibrations have been proven on a dynamometer to be able to change the power output characteristics of an engine. The data gathered by these tests have shown that our AF Optimizer is fully functional in its use.

This section will discuss in detail the status of the various project components, including the firmware, the hardware and the vehicle integration.

2.1 Firmware Status

The firmware integrated into the AF Optimizer is composed of three main conceptual modules. The first and most important module performs the Digital Signal Processing (DSP) functions of our unit. The second module is responsible for running the user interface along with the display functionality of our unit. A third module is used to relate various vehicle sensor data to the driver and is known as the monitoring module. All three modules were then integrated to ensure that the DSP functionality of our unit always maintained the highest priority and that we met all requirements set out in our functional specifications.

2.1.1 DSP Firmware Module

The functionality of this module entails sampling the vehicle's airflow meter along with the TPS and RPM signals. The airflow meter along with the TPS signals pass through the built-in 10bit A/D converters of the PIC microcontroller while the RPM signal is simply attached to one of the inputs of the PIC since it is conditioned into a square wave signal. We were able to read all three signals accurately, and it is these three signals that are used to perform all the interpolating calculations that result in the output signal returned as the recalibrated airflow signal to the vehicle's ECU. The interpolating algorithm returns the recalibrated airflow signal based on the real-time values of the TPS and RPM signals cross-referenced with the user-stored calibration values in the microcontroller's EEPROM. The interpolating algorithm is written to both be efficient and ensure a smooth transition between all user-preset calibration values for airflow. A graphical representation of the function performed by the interpolating algorithm is shown in Figures 2-1 and 2-2 below. Note that both RPM and throttle position values are successfully used to perform the two-variable interpolation for every 250 RPM interval between 2000 and 8750 RPM.

4250 4500 4750 5000

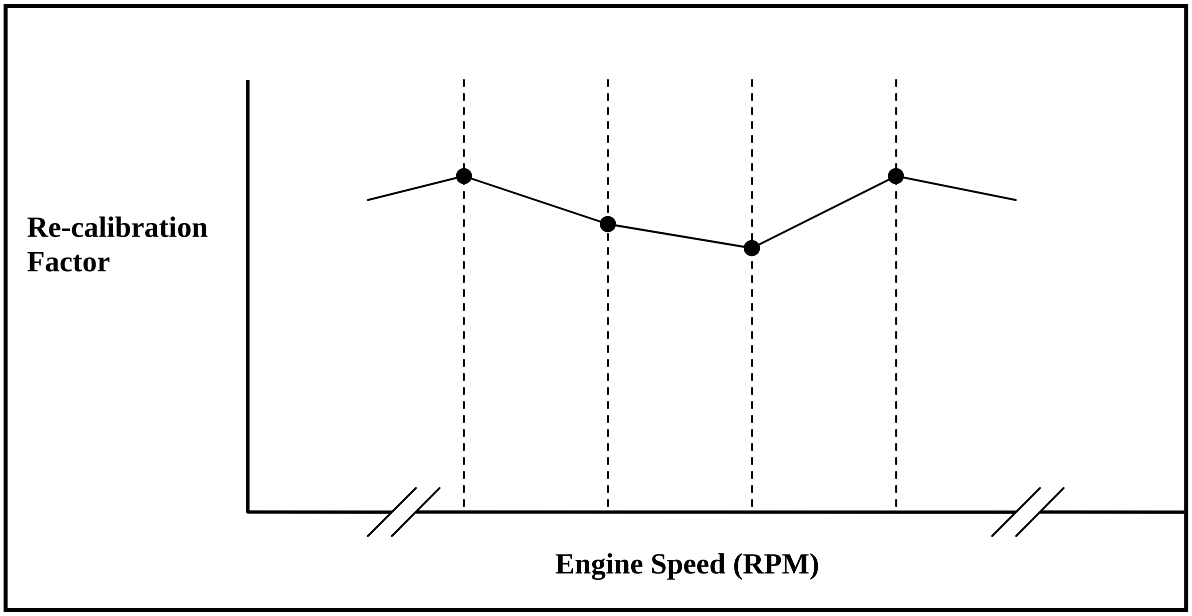

Figure 2-1: Interpolated Re-calibration Factors as a Function of RPM

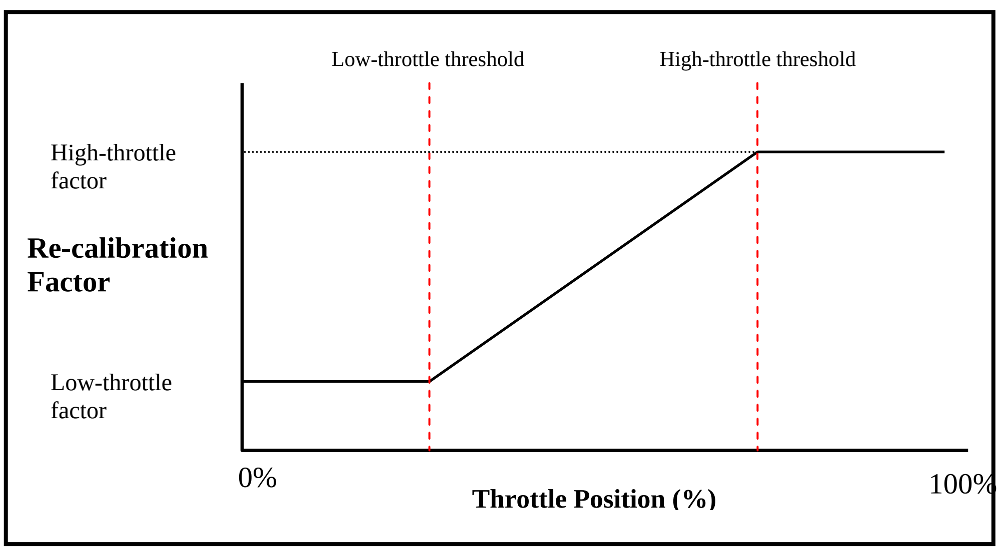

Figure 2-2: Interpolated Re-calibration Factors as a Function of Throttle Position

The value of the interpolating algorithm is serially fed to an external DAC that sends its output to the vehicle's ECU. The total latency of our system from input to output is 1.5ms which meets our requirements and proves to be more-than-adequate during in-vehicle testing.

2.1.2 User Interface, Display and Monitoring Modules

The user interface and display modules allow the user to interact with our unit. The user interface is composed of five buttons that allow intuitive navigation through our menu systems as shown in Figure 2-3. The left and right buttons are used to scroll through particular parts of a menu, whereas the up and down buttons are used to modify values such as the calibration percentages in the calibration menu. Finally, the centre button is used to scroll between the main setup, calibration, and two monitoring menus. Button bouncing was eliminated through the implementation of interrupts which allowed the user interface to be both responsive and efficient while not delaying the processor with lengthy interrupt service routines.

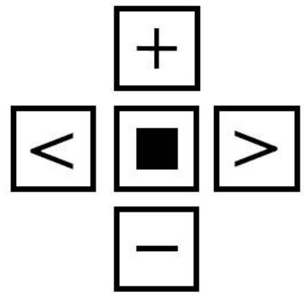

Figure 2-3: Navigation Buttons

The display features of our unit were carried out through a serial interface with the firmware imbedded in our VFD display. Since displaying to the VFD introduced delays to our microcontroller, the maximum error-free baud rate of 115,200 bps was employed.

The menu structure was divided into three main modules: setup, calibration, and monitoring. Figure 2-4 shows the general structure of each of these modules. The setup module is utilized to input first-time parameters to the system such as the calibration of the TPS, the low/high throttle thresholds, as well as the shift light RPM value. The calibration menu allows the modification of the airflow signal for each 250 RPM interval between 2000 and 8750 RPM for both low and high throttle calibration curves. To ensure a trouble-free operation of the system while driving, the user can only modify values in both the calibration and the setup menus while the car is either turned off or idling.

```
Zero Throttle:    0mV

Press UP to Enter
Press RIGHT to CONT
```

```
 3000 3250 3500 3750

>102% 110% 109% 107%
 98%  100%  97%  98%
```

```
RPM:          3250rpm
Throttle:       80%
Battery:        14V
O2 Sensor:     220mV
```

Figure 2-4: Sample display screens showing Setup, Calibration, and Monitor modules

The first of the two monitoring menus display real-time RPM, TPS, battery voltage, and oxygen sensor signal values. The second monitoring menu displays the current calibration factor as a percentage as well as the pre and post calibrated voltage value of the airflow signal in millivolts. The monitoring menus accurately keep the driver informed of the status of all these vehicle sensors as proposed in our functional specification of our product.

Finally, the five LED sequential shift light alerts the driver of the shift point 500 RPM before that point is reached by lighting up the first LED. Every 100 RPM after that, an additional LED is lit in sequence until the gear shift should occur.

2.1.3 Integration of the Modules

All three firmware modules described above met all the functional requirements proposed for our product. Furthermore, the entire firmware code fit into 92% of the PIC's 8k of internal program memory. An area that was optimized, but not completely is the delays caused by the display and user interfacing functions. These delays caused minor interruptions in the airflow signal output of our unit which should be eliminated entirely if possible. In spite of such delays, our unit's response time did not negatively affect the driveability of the vehicle during our testing phase.

All the firmware code was written in the C language and then converted to assembly and HEX code using a commercial C compiler for the PIC microcontroller. Using C for development allowed for easy portability of the code, a quick learning curve of the microcontroller and reduced development time. However, memory usage could have been reduced and more efficient code could have been written if the firmware code was written directly in assembly.

2.2 Hardware Status

Currently, the hardware is in a complete prototype state. That is, all of our input circuitry is fully functional, as is our output circuitry. We constructed a case to house the prototype, and in it we placed our custom fabricated printed circuit boards for the PIC/mainboard and one for the buttons. We followed all of the designs as laid out in our design specification. The following list outlines the hardware components that were successfully designed and implemented in a real world scenario:

1. 5 Volt, 3A power regulator
2. PIC Microcontroller @ 20 MHz clock
3. Noritake VFD Display via 115.2kbaud serial connection
4. All input circuitry
5. The 10-bit Maxim Semiconductor DAC output circuitry
6. 5 buttons and their required circuitry
7. 5 LED sequential shift light
8. Hardware failsafe / Calibration off switch

The only hardware design that received no mention in our design specification was that of the hardware failsafe / calibration off switch. This switch allows the user to bypass the AF Optimizer's recalibration ability in the event of a malfunction of the AF Optimizer, or a desire to "turn off" the tuned version of the airflow sensor.

2.3 Vehicle Operation Status

The AF Optimizer's specific use is to change the air fuel ratios in a vehicle's engine, while displaying monitoring information to the user. In order to do this successfully, we needed to design specific input circuitry capable of conditioning the specific vehicle sensor inputs. Signal research and input signal testing allowed us to easily design the needed circuitry for required operation within a vehicle.

Currently the AF Optimizer has been integrated to a 1988 Honda Prelude 2.0Si, which is a fuel injected vehicle. Vehicle testing, through daily driving, has shown that the AF Optimizer is a reliable unit and has caused no running problems in the car. Once the driving testing proved that the unit was reliable, we took the vehicle to an automotive tuning shop where we had access to a dynamometer. Dynamometers give readings of a vehicle's output horsepower and torque, and also provide information about the operating air fuel ratios. Our team was able to use this tuning tool to prove that the AF Optimizer provided a means for changing the characteristics of a motor. More specifically, we were able to prove that our unit was able to change the air fuel ratios in a predictable and desired manner, which in turn changed the horsepower output and efficiency of the motor. Although repeated measurements showed slight variances between dyno tests, our unit was performing in an acceptable manner for its prototype stage. Hardware improvements will be made to reduce these variances.

We will be redesigning some of the software to include more features from the existing AF Optimizer. So the current design will eventually be revamped, but it will be continually used for testing purposes in the current test vehicle. These tests will hopefully reveal specific user operations that can be redesigned for more optimal use.

3 Problems Encountered

This section will turn the focus from the current status of the AF Optimizer to the problems that were encountered throughout its development. The AF Optimizer relies almost entirely on its well-written software. Without well-written code, the AF Optimizer would not be able to run a car with its recalibrated airflow sensor value. Thus, we will be discussing the specific software design problems that needed to be resolved throughout our software development.

The software would not be able to do its job correctly without being integrated with the correct hardware. The AF Optimizer needed to be designed with the correct hardware so that it could be integrated properly into car. We needed hardware to provide reliability, accuracy, and protection for the AF Optimizer's continued use in an automotive environment.

3.1 Software Problems

The firmware of the AF Optimizer was the aspect of the project that our group devoted the most time to developing and integrating. Throughout this time, we had several obstacles which we overcame, and the most notable of these are outlined below.

The first firmware component of the AF Optimizer that was developed was the menu system. We had several goals for this menu system: it had to be memory efficient and had to exhibit good usability characteristics. This was a challenge as we were dealing with 8k of memory, code development was done in the C language, and our display area was limited to 20x4 characters. However, through the elimination of redundancies when refreshing information to the screen by reusing characters that were already written and keeping track of which menu was displayed through variables, we were able to dedicate less than 50% of our total available memory to the menu system; a great achievement. Although our usability survey from non-technical people has shown that the user interface could use improvements, they were able to navigate the menu system with little prior instruction.

Another challenge that came as part of the user interface was the implementation of the button de-de-bouncing algorithm. At first, this essential algorithm employed a delay to ensure that the button signal had settled which kept the microcontroller in an interrupt subroutine for much too long. In order to remove this unnecessary processor delay, button de-bouncing was implemented using a timer internal the microcontroller that would overflow when the required delay time had elapsed.

The actual display functionality of our unit was carried out through a serial interface established between the VFD display and the PIC microcontroller. This was very simple to setup using the C language, but required some attention to the fact that writing to the screen delayed the DSP functionality of our unit. To overcome this difficulty, we used the fastest baud rate available and optimized the refresh rate in our sensor monitoring menus such that it was adequately fast while slow enough not to overburden the capabilities of the processor.

The most noteworthy issues encountered during the development of our DSP code revolved around the resolution, accuracy, repeatability, and latency of our output. In order to maintain the first three of those aspects within specifications, we had to use floating point calculations at times which hurt the response time and used large amounts of memory. This is where the efficiency of our code suffered, especially since it was written in the C language and not directly in assembly.

In spite of these shortcomings, we met our memory and latency requirements with very little room for further improvements to functionality due to memory limitations.

3.2 Hardware Problems

Naturally, there were problems encountered in this ambitious project. The most prominent problem occurred when we first tried out the prototype circuit in a breadboard in a vehicle. We found that the 5 Volt output of our regulator was experiencing brown out conditions, and that caused the VFD to turn off, and back on again with several seconds of initializing required. This left the PIC in limbo, waiting for an acknowledgement from the VFD, and the system crashed. This problem was solved by using a rectifier circuit on the 12V side of the regulator, consisting of a series diode and a large capacitor. We also improved high frequency impedance of the voltage rail by using a 10uF Tantalum capacitor.

Another issue encountered was using 5 buttons, when we had only 4 edge-triggered interrupt pins. We solved this by encoding the buttons such that the 5th button sends an edge to two pins, and the software decoded this. It was necessary to use two diodes so that the buttons which use those two pins did not cross-talk through the 5th button.

The final issue we encountered was that of how to make a stable 2.5V reference for the DAC. Our attempts at using small surface mount references turned out poorly, so instead we used a voltage divider network tuned to account for the input resistance of the V_{ref} input pin of the DAC. This solution was not only very simple, but also cost effective and accurate enough for this project. Plans to use a more stable 2.5V reference are in the future.

3.3 Integration/ Vehicular Issues

Once initial power supply and connection issues were resolved within our circuitry, the AF Optimizer performed as expected in the test vehicle. Largely due to the research completed beforehand, signals were read and output by our AF Optimizer with no problems. Our initial research and signal testing revealed to us how we should implement input circuitry for our AF Optimizer. Thus the potential issues of integration into a vehicle quickly became non-issues. Comprehensive lab simulation also provided our team with an accurate approximation to an actual vehicle, so we were able to test many conditions beforehand.

As previously stated, there were initial issues with the power supply and wiring connections for our circuitry. These problems will later be discussed in detail. Such problems arose because of unforeseen car battery behaviour. When the vehicle's starter is turning, it is drawing up to a few hundred amps of current from the battery. This current then causes the car's battery voltage to drop substantially. This voltage drop was enough to cause brown-outs for our AF Optimizer, which essentially rendered our unit useless and caused the vehicle to operate in an unpredictable manner. Power supply upgrades implemented within our input circuitry enabled us to overcome these issues and produce a unit that no longer suffered the effects of brown-outs.

Wiring connections were a minor issue that had to be overcome as well, since the breadboard's connections were not perfect. Initial vehicle testing revealed inconsistent reading of input signals, which caused erratic output from our AF Optimizer. A simple solution entailed designing and creating a printed circuit board to secure all the wiring connections.

Since these early integration issues were solved, there have been no problems with the AF Optimizer's operation within the test vehicle. We are able to successfully read in all vehicle sensors necessary for fluent operation, and we are able alter the vehicles air fuel ratios through the change of desired calibration percentages.

3.4 Group Dynamics and Collaboration

Our design team was structured in such a way that different modules were designed and worked on separately to start. Namely, these modules were the software/firmware module, the output circuitry module, and the input circuitry. Then, as each module progressed to a level where it could be integrated with other modules, we combined them.

Alex and Russell worked closely together designing, developing, and testing the system firmware upon which the AF Optimizer was to run. This process entailed designing the efficient yet understandable user interface with the button interrupts and the monitoring information displayed on the screen, designing the signal processing functions, and outputting a steady recalibrated value back to the car's ECU. This firmware development dealt with optimizing the source code to be efficient, and reliably accurate. As discussed in previous sections, the firmware was a large part of this project, and it needed to be tested and thoroughly developed to provide a product that the end users (automotive enthusiasts) would be easily capable of using to tune and monitor their vehicles.

The other sections, dealing with the input and output circuitry, were handled separately by the entire group. Russell and Alex spent many hours researching and measuring the actual signals from the test vehicle in order to understand what kind of input circuit would be needed. Brian then used the signal research to design the needed input conditioning circuitry to provide suitable input signals to the microcontroller.

Brian spent time researching and sourcing the digital-to-analog converter (DAC) that would be needed as part of the output circuitry. This piece of equipment was essential in delivering an accurate voltage level back to the car's ECU, being the recalibrated airflow sensor voltage. A short piece of firmware code was written that easily implemented serial communication with the DAC to output the recalibrated voltage level. Once the full functionality of the circuitry was completed, Alex and Russell were able to completely test the AF Optimizer software to ensure everything was operating smoothly and as optimally as possible before testing it in a vehicle.

Once the system firmware was fully functional and ready to be tested in a vehicle, a printed circuit board (PCB) providing strong electrical connections was needed to be designed. Russell and Alex designed the prototype PCB to be used in the prototype casing. They then took their design and etched the final PCB used in the prototype. At the same time, the prototype casing needed to be designed and build to hold the entire AF Optimizer. Russell and Alex designed and machined the prototype product casing to encase the main PCB, the buttons and display interface, and the shift light.

Throughout this final prototype completion stage, Brian was primarily involved in developing the company and product website: www.sfu.ca/~bnelson/. This website provides a company overview as well as a product overview outlining its potential for tuning yours car's fuel delivery. Additionally, the website provides browsers with documents such as the Function Specification and the Quick Start User Guide. Finally, the website gives email addresses so potential customers can obtain more information about our product or company. This website contribution to ACS' AF Optimizer project really completed the product by introducing a marketing tool that can be used to attract customers. It really caps off this company and its AF Optimizer product.

4 Future Design Plans

As stated earlier, the AF Optimizer is currently in its prototype form, where it was used to prove that the concept of tuning your car's air fuel ratios was viable. This being said, there are still improvements that should be made to both the software and the hardware before the AF Optimizer is released to the market. For the future of our product to be successful, we must continually implement improvements and feature additions to stay competitive. These improvements to the software, the circuits, and the casing will now be discussed in further detail.

4.1 Firmware Improvements

Although the AF Optimizer is fully functional as a prototype, there are various firmware improvements that would improve both its performance as well as add features to the existing product. The improvements and features suggested in this section are a result of in-car testing and user feedback.

4.1.1 Platform Improvements

The AF Optimizer is a product that is built around the functionality of its software. This software resides in the memory of the microcontroller and therefore the choice of microcontroller employed is a very important one. We have chosen the PIC16F877, and although this choice is a very good one due to its low price, availability and the feature set, an application-specific solution would be better suited for our purposes. If we were to move away from the PIC family, we would be looking at a microcontroller that has an independent serial or graphics interface controller. We would also look for a microcontroller with a smaller PCB footprint, a better ability to sink current, and a larger internal memory size. These requirements would raise the cost of production for the AF Optimizer, however, if we want to add more features to our product, the current microcontroller is not sufficient.

A second option to improving the processing capabilities of our unit is the implementation of dedicated DSP and user interface/display PIC microcontrollers. This solution would eliminate any interruptions of the DSP functionality by the user interface and display activity while maintaining a low product cost and 100% code portability. However, the shortcomings of this solution are the increased footprint needed to house the two PICs and an overall less-elegant solution.

4.1.2 Assembly v.s. C Language

A clear improvement to the performance and reduction of memory usage of our code is its implementation directly in assembler language. However, during the development of the product this would not be the best choice. The main reason for this is because, as mentioned above, a different microcontroller could be chosen. If such a decision is made, assembly code written specifically for the PIC is very limited in portability whereas code written in the C language remains easy to understand and retains a higher level of portability to a different platform, thus making the transition to a different microcontroller much easier. Eventually, when there is no longer any question in regards to what microcontroller should be used, specific assembly code should be written.

4.1.3 Features to Add

There are several features that are not part of our current functional specifications document, but that we have come to believe will improve the functionality of our product as well as its appeal to our customers.

First on this list of proposed features is the addition of the ability to store several calibration curves that can then be retrieved and used as desired by the user. Currently, our product has the storage capabilities for a single set of low and high throttle calibration curves.

Another important feature would be the introduction of the ability to recalibrate mass airflow (MAF) meters that have a frequency signal as apposed to voltage. Although the implementation of such a feature would require a major code and logic revision, its presence would greatly widen the market potential of our product since many newer vehicles employ this type of airflow sensor.

There are several vehicle makes and models that make especially good potential customers for our AF Optimizer. Many of these vehicles employ a system that opens secondary intake valves to allow improved airflow past a certain RPM. With slight modifications, our product can give the driver the ability to change the RPM at which those secondary valves open; once again resulting in more control over the air/fuel ratios and output power of the vehicle.

A final suggested feature is the ability to calibrate the airflow meter signal for RPM values that are lower than 2000. This will result in the improvement of the idling characteristics of vehicles that have difficulties in that respect due to engine modifications.

All the additional features listed above will make the AF Optimizer an improved, more competitive product. The addition of these features is strongly considered for the production model.

4.2 Hardware Improvements

We will now discuss the improvements that are planned for the hardware. In order to optimize PCB spacing to fit the AF Optimizer in smaller casings, the physical circuits must be redesigned to take up less space. Additionally, circuit components will be re-evaluated based on pricing, accuracy, and performance. The prototype casing must be redesigned for its potential sale in the automotive market. Specific improvements covering these issues will now be discussed.

4.2.1 Circuit Improvements

For future manufacturing consideration, this project will require a drastic change in the printed circuit design. To begin, every part will be that can be found in a surface mount version will be implemented in surface mount. This will reduce the cost of each part, as well as the reduce the size and cost of the printed circuit board. It should be noted that a few parts, such as the 2.5V reference part were only available in a surface mount package, and as such we were unable to take advantage of them in our prototype design.

In addition to changing the printed circuit board design, there are also some improvements to be made to that actual circuitry implemented in our design. It was brought to our attention that the DAC could be better filtered by using a different capacitor configuration. Our current design places a capacitive load directly onto the output of the DAC. By placing the capacitor between the output and the feedback pin, we would effectively reduce the load while having no undesired consequences. Additionally, a more stable 2.5 volt reference will be used for the DAC, rather than the current voltage divider off the supply rails. This reference will allow for a more steady

DAC output voltage. Input circuitry will be improved to utilize anti-aliasing filters to reduce the affect of noise on the input signal lines. These improvements will increase the ability of our AF Optimizer to produce repeatable results on the dynamometer.

A final improvement that should be made before this project is manufactured is to allow for a more diverse range of input types. For example, our current design only allows for a 0-1V oxygen sensor, whereas some of our potential customers could have an 02 sensor that has a higher voltage range, and by using it, the input to the PIC would face an over-voltage. Instead, we could have a few different possible connections that an installer would choose from depending on the type of sensor/voltage ranges in that particular vehicle.

4.2.2 Casing Improvements and Modularization

Products in the automotive market have success related to product functionality and product appearance. These issues make sense because consumers want a product that works well, and they want a product that looks good enough that they can show it off to other people. Thus, we need to redesign a casing that is aesthetically pleasing; the product needs to look good on the dashboards of our customers.

The main issue felt by our prototype design is the fact that all the components are cased within a single case. A future design would have separate modules that could be each mounted (or hidden) in different locations within the car. Incorporating separate modules has its advantages being two fold. Firstly, it will allow the individual cases to be physically smaller in size so that the users would be able to mount the modules in a variety of places without becoming an eyesore. Secondly, it would allow the user to mount individual modules in more convenient locations for their specific use. The modules have been already determined to be the brain/main PCB module, the user interface module, and the shift light module. Now the driver could, for example, mount the shift light module in a location that is easily seen while driving, perhaps in a location that would trigger his/her attention. Next, they would be able to mount the user interface (with the display and buttons) in a convenient location enabling them to view the display easily, but not be too distractive while driving. Lastly, the brain unit, consisting of the microcontroller and the other circuitry, could be placed in an inconspicuous place, such as the passenger foot-well.

These improvements to the product casing will prove to be a promising selling feature because it gives the customers flexibility in how they install and integrate the AF Optimizer in their cars.

5 Budget and Scheduling

The budget and scheduling aspects of this project were important in that they helped keep the project on its predicted track. The proposed budget gave our group a target spending amount, while the target timeline kept our group operating on schedule so we could deliver a final product by the deadline.

5.1 Budget

In our project proposal, we allowed for a total proposed spending budget of $630. This was designed using our best predictions on what parts would cost, how much we would need to spend on equipment rental and usage, as well as spending time on the dyno. Our actual spending came in at $200. The table below outlines our proposed vs. actual spending for the AF Optimizer project.

Table 5-1 - Budget Analysis

Equipment	**Predicted Cost**	**Actual Cost**
Vacuum Fluorescent Display (VFD)	$100	$0
PIC Microcontroller	$20	$20
Product Casing	$60	$10
Dyno-Time – ½ day	$300	$100
Specialized Equipment Rental	$100	$0
Printed Circuit Board Manufacture	$50	$20
Other expenses (buttons, parking, etc.)	$0	$50
Total	**$630**	**$200**

We conveniently used a VFD that was given as a gift to Russell Potter and did not purchase a second one. It should be noted that the cost of buying one of these displays is around $140 with taxes! We found a case that worked for the prototype AF Optimizer, and the cost was only $10. As for the Dyno time, we allotted $300, which is fair market price for several hours on the machine. However, we had a connection that was able to get us on the dyno and help us one-on-one for a total cost of $100, so we lucked out again.

The SFU Engineering lab staff agreed to allow us to use an old analog scope in Russ's vehicle to characterize the input sensor signals. In addition, there was also a PIC programmer available to us in the engineering lab. As such, we did not need to rent or buy any special equipment to complete the project.

Something we did not account for was the cost of incidental parts such as buttons, connectors, etc. For our demo, we also needed to have the test vehicle parked at SFU, and that cost $10. The total cost of this project to the group was $200.

Our project received only one form of funding and that came from the ESSEF. We would like to thank all the donors to the ESSEF, as well as those who administer the fund. Without the $200 bursary from this endowment fund, we poor students would have become slightly poorer over the course of this project.

5.2 Scheduling

In Figure 5-1, the original Gantt chart showing the target timelines for this project has been updated to include the actual time spent on the various aspects. These actual times are shown in the colour red whereas the original timeline remains in blue. We started this project on time beginning with software development and were able to complete the working prototype before our final demo. However, the individual components of our project such as the firmware design and implementation took longer than we had originally predicted. Learning how to accurately schedule a large project was also new to us and understanding the necessary completion times is something we developed throughout the last four months. Nevertheless, we were not stressed for time near the end of the project which is something we can attribute to our consistent efforts throughout the semester.

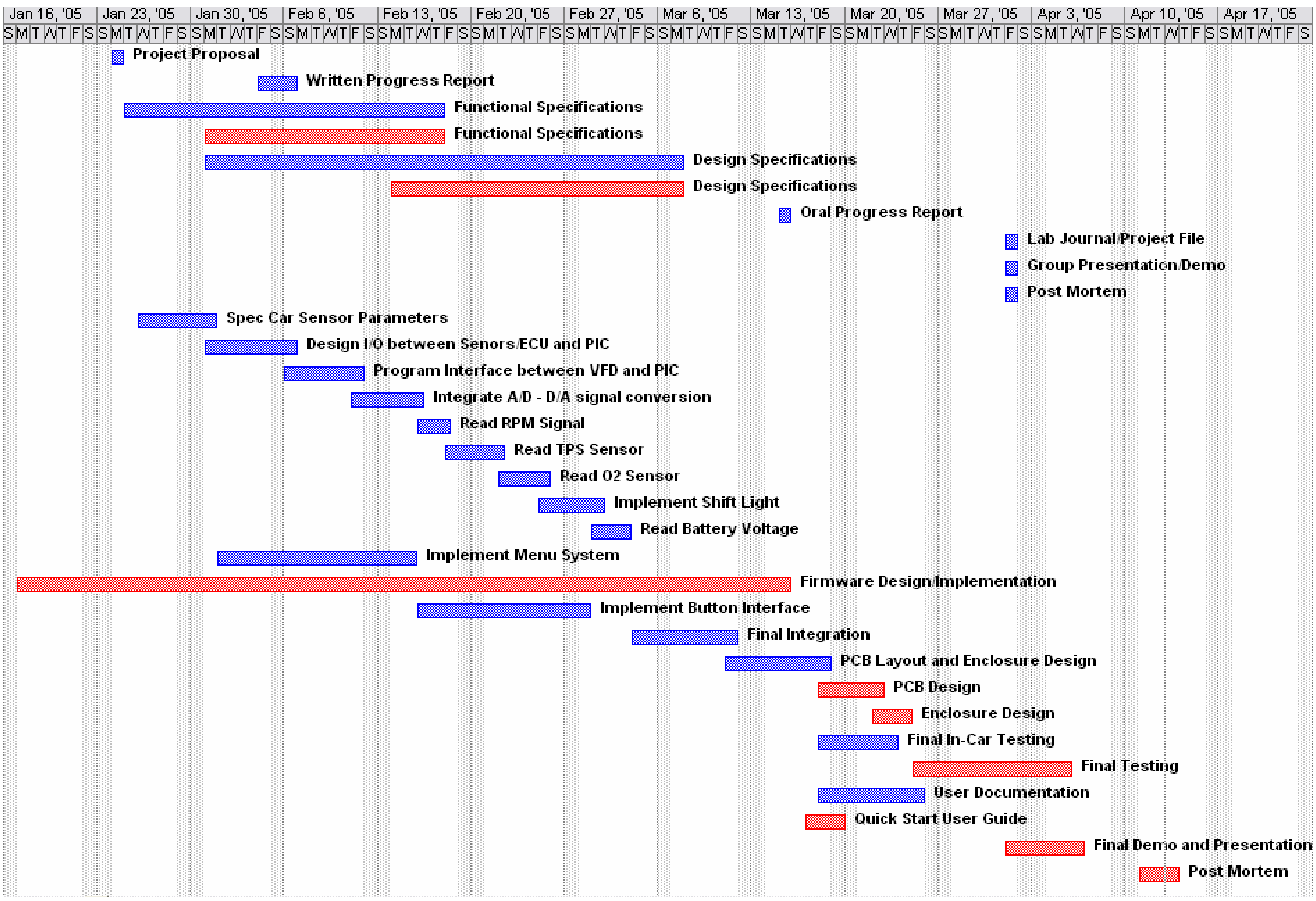

Figure 5-1: AF Optimizer Scheduling Gant Chart (revised with actual completion times)

6 Individual Experiences

There were many individual experiences throughout this entire project. Many late nights in the lab researching, developing, and debugging have given us all very valuable experiences that we can carry forward into our careers as engineering. The following section will provide an overview of the experiences each team member felt during this entire project design and implementation process.

6.1 Russell Potter

I really enjoyed this project from a sense of seeing it come from a concept and design stage through to a working device. I was ecstatic to see our product come to completion through a brief tuning of my car on the dynamometer. To be honest, I had this idea for an ENSC440 project for more than a year before I was finally able to get started on the project this past semester. So when it came time to develop our AF Optimizer, I was full of energy and enthusiasm for doing a great job.

Until now, courses at SFU were mainly theoretical with small lab components that allowed you to slightly see how the theory was used in real life. I found that this project allowed me to use all of my programming skills and electronics education I had received over my past years at SFU to develop a product that I found really interesting.

My entire group found this project interesting and fun, which provided a great point to build from. From researching, to signal testing, to coding and debugging, and integration, I think the common interest we shared in this project combined with our varied skill-sets allowed this product to come together relatively easily and quickly. I am very happy that we were able to produce a prototype product that was fully functional and well within our proposed budget, and I think that all the hard work and late nights provided an avenue for accomplishing this task.

I would like to thank my cousin Dave Atchison for being a large help with getting us started with our PIC microcontroller, and for being the source of our inexpensive dynamometer testing. Above all, I would like to thank my team for the dedication and vision we all shared in creating the AF Optimizer.

6.2 Alex Gutica

The completion of this course and this project brought a sense of accomplishment and completion to the last four years spent in Engineering Science at Simon Fraser University. As a member of a talented team of engineers, I was able to be a part of bringing a product from concept to completion. Being able to apply the skills acquired over the last few years in order to do so was rewarding as I quickly realized how the many hours of study and lab time can effectively be applied in creating something that is both functional and useful.

I am an automobile enthusiast and sharing this enthusiasm with the other members of my group brought a general harmony amongst us that nurtured a strong devotion to bringing this project to successful completion. When Russell proposed the initial idea for the AF Optimizer, I quickly found myself wanting to spend more time on this project than other daily activities; I finally felt truly passionate about a “school assignment.”

Each member of our group applied himself to an area of the project that they felt most knowledgeable in. I found myself being the most involved in the development of the firmware for the AF Optimizer. That being said, due to the small size of our group and common interests, I was able to collaborate with my team members and participate in all aspects of the product development which included hardware design, PCB layout, enclosure design, product assembly, and finally real-world testing.

Overall, I am very satisfied with the results of this project and this course. In fact, I am so satisfied with the final product, that I am assembling a secondary unit that will be used to manage fuel delivery in my own turbocharged Ford Escort.

6.3 Brian Nelson

I found this project to be not only my most successful team project in university, but also the most fun and rewarding. My talented team members complimented my skill-set very well, and allowed me to do what I really wanted to do: hardware design. I credit the team for getting started early, and allowing plenty of time to iteratively design a great product. I learned a great deal about real world hardware design. Some examples are considering noise in a vehicle, loading of sensor outputs, DC/DC conversion, etc. Fortunately I was able to accomplish the tasks that were assigned to me. Naturally there was plenty of collaboration for both the hardware and the firmware. In particular, Russell was a big part of hardware design, and provided a bridge between the firmware team and the hardware team. Besides that, there were some late-night occasions where I was part of the firmware debugging process.

My other contribution to the team came in the form of some organization and documentation planning. For most of the major documents, I would start by assessing what the document should contain. Then I would produce a skeleton document which outlined each section to be filled, what should go into each section, as well as who would be best suited to writing it. By taking on the initial phase of documentation, Alex and Russell had more time to devote to the labour intensive task of writing and debugging code for the PIC Microcontroller. My hat goes off to my team-mates for making this a successful and fun experience.