QND Medical Devices

School of Engineering Science
Burnaby, BC
V5A 1S6
qnd-medical@sfu.ca
http://qndmedical.cjb.net

December 18, 2000

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 340 Process Report for the POSSESS

Dear Dr. Rawicz:

The attached document, *Process Report for the POSSESS* (**PO**rtable **S**afety and **S**leep **E**nhancement **S**y**S**tem), essentially summarizes what we learned and experienced during our product's development. The POSSESS allows a user to awaken after he or she has completed an REM stage of sleep in a user-defined time window. Doing so allows a user to consistently awaken refreshed and alert. As more people trade-off sleep for work, the market for the POSSESS only grows larger.

Our process report details the current state of the POSSESS. An overview of the project, the technical implementation of our solution, what we would do differently given another chance, and various problems we encountered during project development are also discussed. Additionally, we present possible enhancements to the POSSESS and each of QND's members delivers a statement regarding the valuable lessons learned during ENSC 340.

QND Medical Devices is comprised of five talented, dedicated, and emotionally drained fourth-year engineering students: David Lee, Stephen Liu, Hiten Mistry, Roch Ripley, and Matt Ward. Should you have any comments, questions, or suggestions, please feel free to contact me at (604) 421-6126 or via the e-mail address listed above. Thank you for your time.

Regards,

Roch Ripley

Roch Ripley
President and CEO
QND Medical Devices

Enclosure: Process Report for the POSSESS

Process Report for the

POSSESS

Portable Sleep and Safety Enhancement System

Project Team: David Lee
Stephen Liu
Hiten Mistry
Roch Ripley
Matt Ward

Contact Person: Matt Ward
(604) 929-7889
E-mail: qnd-medical@sfu.ca

Submitted to: Dr. Andrew Rawicz (ENSC 340)
School of Engineering Science
Simon Fraser University

Steve Whitmore (ENSC 305)
School of Engineering Science
Simon Fraser University

Date Submitted: December 18, 2000

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The POSSESS represents more than a semester of hard work by five engineering students. The POSSESS is an integration, or synthesis, of numerous ideas that the members of QND Medical Devices first started discussing in mid-summer, 2000. To create the POSSESS, we had to take into account numerous biomedical, societal, and technological considerations. By doing so, we were able to create a device that, to the best of our knowledge, is the *first brainwave-monitoring alarm clock in the history of humanity*.

Predictably, creating the *first brainwave-monitoring alarm clock in the history of humanity* has taught each of the members of QND Medical devices a few lessons. Such lessons are summarized in this report, and are perhaps the most important part of it.

2 PROJECT OVERVIEW

The POSSESS is an alarm clock that awakens a sleeping person based on his/her brainwaves and eye movements. By performing signal processing on this data, the POSSESS determines which stage of sleep the user is experiencing. Figure 1 is a detailed block diagram that gives an overview of how the POSSESS acquires data, conditions and processes it, and generates an alarm tone.

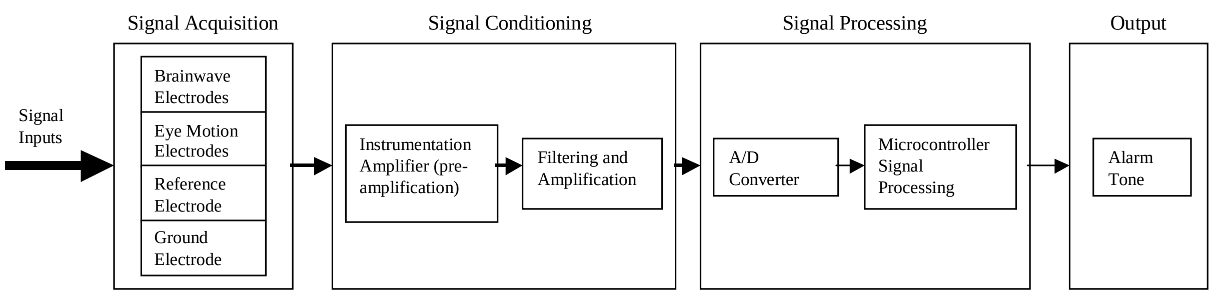


Figure 1: Detailed POSSESS Block Diagram

For normal operation, the user will use the soft menu to specify a time window in which he or she would like to awaken. Via four electrodes (one for brainwaves, one for eye movements, one for signal ground, and one reference electrode), the POSSESS then acquires the necessary brainwaves and eye movements from the user.

In the signal conditioning stage, an instrumentation amplifier compares the acquired signals and any common-mode signals (e.g.: 60 Hz noise) are rejected. The important differential signals are amplified. The amplified differential signal continues through multiple filtering and gain stages and is eventually amplified to the level required by the signal processing stage.

The signal processing stage converts the analog signal it receives to a digital format for use in our Atmel AVR AT90S8535 microcontroller. The custom software running on our microcontroller determines if a rapid-eye movement, or REM, stage of sleep terminates while the user is in his or her own defined time-window. If appropriate to do so, the user is awakened upon completion of the REM stage. If not, the user is awakened at the end of the time-window. For medical safety reasons (i.e.: in order to achieve electrical isolation), all components of the POSSESS run off power supplied by batteries.

Figure 2 is a high-level flowchart of the software algorithm we use in the POSSESS.

Main Program Flowchart (assuming all user input dealt with via appropriate interrupts):

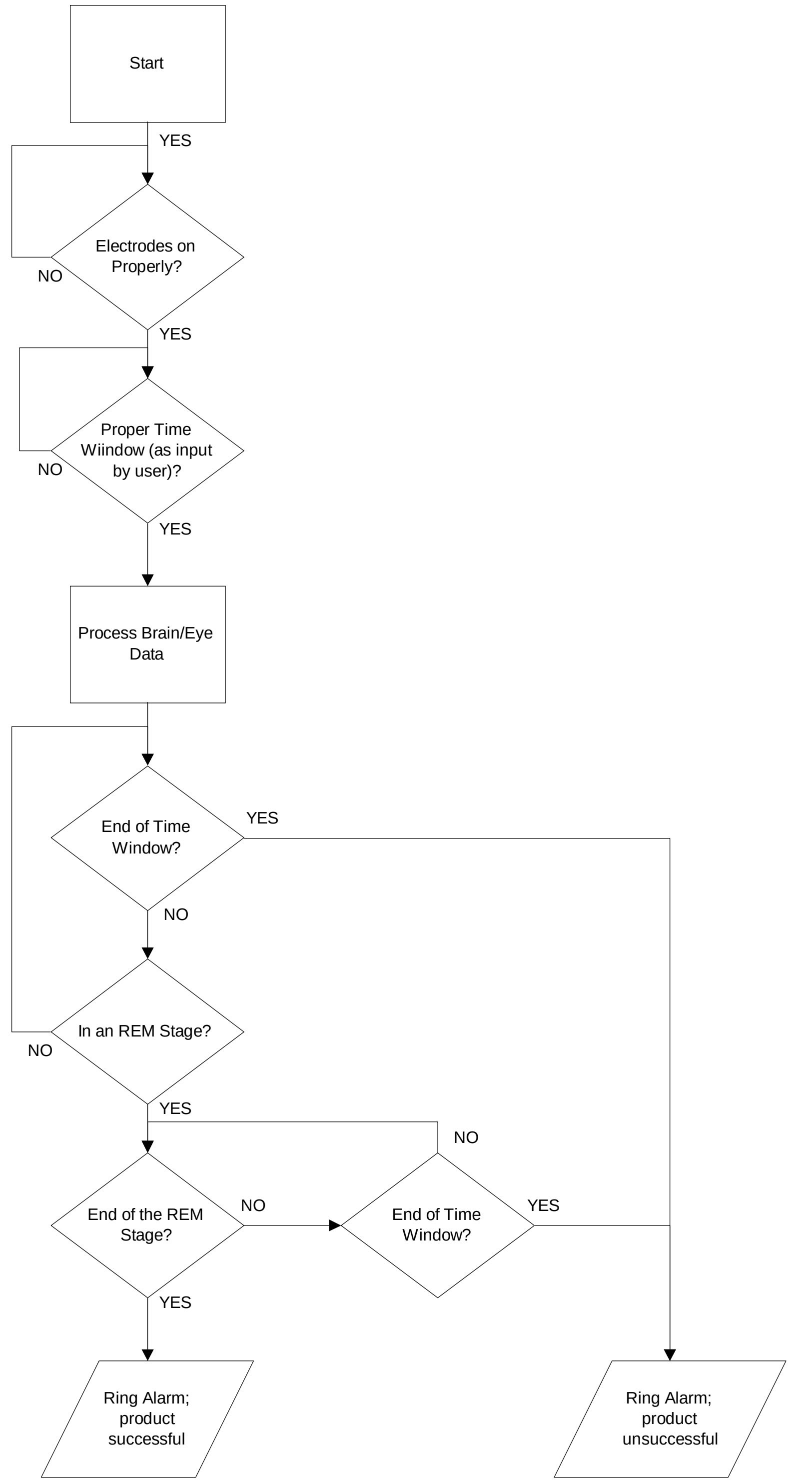


Figure 2: Flowchart of Signal Analysis and Output Generation

3 TECHNICAL IMPLEMENTATION OF SOLUTION

The description of our technical implementation will follow the signal block diagram of Figure 3. We shall explain the technical implementation of each block proceeding from left to right.

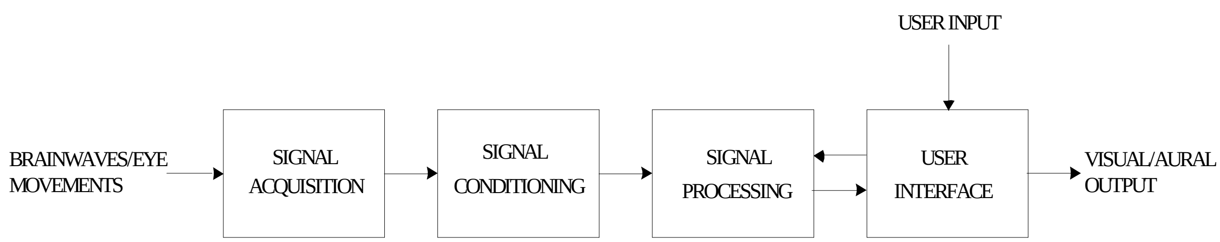


Figure 3: POSSESS Block Diagram

3.1 Signal Acquisition

3.1.1 Electrode Selection

The electrodes chosen for our final prototype are reusable silver/silver chloride electrodes that are 9 mm in diameter. Our decision to use reusable silver/silver chloride electrodes is based primarily on their reusability and on their small size.

The reusability of the electrodes allows our user to be able to purchase one set of electrodes, albeit for a relatively high initial price, and then be able to reuse the same electrodes indefinitely. Should the silver chloride coating on the electrodes be worn off, the user can simply dip the electrodes into chlorine bleach, causing the silver chloride to re-form by a redox reaction. Thus, the long-term cost of reusable electrodes to the user is lower than that of disposable electrodes.

The small size of reusable electrodes (9 mm in diameter relative to ~ 25 mm → 40 mm in diameter for disposable electrodes) helps us to place the electrodes through the barrier of hair that is unfortunately present on many, many heads. Smaller electrodes adhere better to the skin than larger electrodes and thus help us to attain higher quality signals, especially when the electrodes must collect good data for multiple hours at a time.

3.1.2 Electrode Placement

Table 1 and Figure 4 elaborate on the proper location of our electrodes.

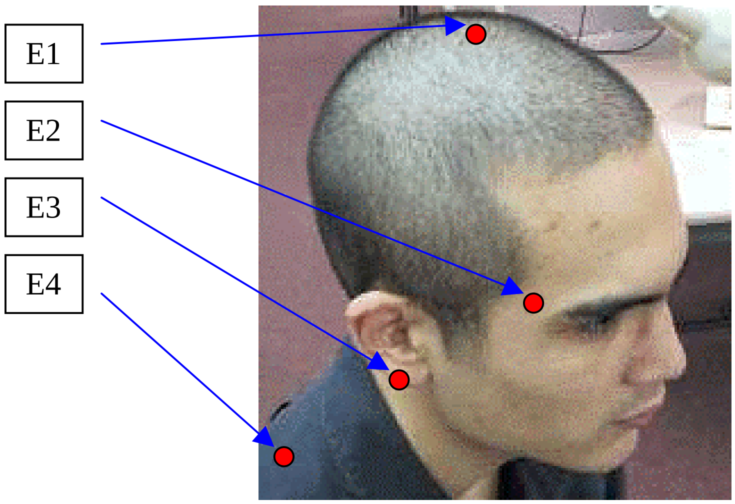


Figure 4: Location of Electrodes on Body

Table 1: Summary of Electrode Placement and Usage

Electrode Reference Designator	Electrode Use
E1	Brainwave acquisition electrode
E2	Eye movement acquisition electrode
E3	Signal ground electrode
E4	Common-mode driver, or reference, electrode

3.1.3 Electrode Adhesion and Conduction

To ensure proper electrode adhesion over a full night, especially over the hair, a substance called collodion is used. Collodion is a skin-friendly biomedical adhesive designed especially for electrode application and adhesion. Conductive gel is applied along with the collodion in order to improve the signal acquisition abilities of the electrode. A simple hair dryer can be used to dry the collodion/gel mixture after application.

3.2 Signal Conditioning

This section will outline the actual functionality of the signal conditioning unit of the POSSESS. Parts are explained in some detail because numerous things have changed since the original Design Specification was written. Most generally, two separate channels are acquired from the user's head, conditioned separately, and sent via a ribbon cable to the signal processing unit.

3.2.1 INA128 Instrumentation Amplifiers

Two INA128 instrumentation amplifiers take differential inputs from two distinct signal electrodes with respect to a shared reference electrode.

The INA128 was selected due to its excellent power, bias current, noise, and supply voltage-requirement characteristics and its good offset voltage and common mode rejection ratio (CMRR) properties. These characteristics, along with its relatively low cost, offer a combination that is more attractive than any other instrumentation amplifier considered. Table 2 quantifies these characteristics.

Table 2: Characteristics of INA128 Instrumentation Amplifier

Parameter	***INA128***
Power Consumption	750 uA
CMRR	100 @ G=100
Supply Voltage	±3V
Input Offset Voltage	50 uV
Input Bias Current	5 nA
Input Voltage Noise	10 nV / √Hz

The gain of the INA128 can be set from 1 to 1000 V/V using an external resistor network. A gain of 100 V/V using a standard resistor size of 500 Ω was used. This level of gain was selected to ensure that the differential signal is adequately amplified before any filtering is performed, but that any unwanted noise is not significantly amplified.

3.2.2 AC Coupled Reference

Connected between the output and the reference terminal of each instrumentation amplifier is one quarter of an AD4251 micro-power quad op-amp. A capacitor connected between the output and the inverting terminal turns the device into an inverting integrator. Hence, for DC inputs, the output of the op-amp is negative, lowering the integration value, and creating a low cut-off high-pass filter. Therefore, the reference terminal of each instrumentation amplifier receives all frequency components from its output above approximately 0.75 Hz.

3.2.3 RC Low Pass Filter

A simple RC low pass filter is cascaded to the output of each instrumentation amplifier. The time constant of the filter was set to be 4.7 ms, and hence its frequency response rolls off starting at approximately 34 Hz.

3.2.4 Compensated Voltage Amplifier

To generate a signal on the order of volts, as required by the signal processing stage, a voltage amplifier was used as the last stage of signal conditioning. Although the INA128 provides a modest amplification of about 100 V/V, the original brain wave signals are on the order of tens of microvolts and the filter stage slightly attenuates the signal. As a result, a non-inverting amplifier is used to amplify the signal by an additional factor of approximately 100 V/V. A 47 nF capacitor was also connected in parallel with the feedback resistor to provide an additional low pass characteristic. The time constant formed by the feedback resistor and capacitor is once again 4.7 ms, to make for a 34 Hz roll off. In effect, combining the simple RC filter with the roll off of the final gain stage makes for a second-order low pass filter in the signal conditioning unit.

3.2.5 Voltage Regulator and Rail Splitter

To power the elements of the signal conditioning stage, a voltage regulator was used in tandem with a rail splitter integrated circuit. The voltage regulator takes the system power from the main batteries, and steps it down to approximately 6 volts. The rail splitter then creates an intermediate signal at 3 volts from the 6 volt regulator output and ground. Now, the elements of the gain stage use the 3 volt constant line as a reference, and are powered in single ended fashion by the 6 volt regulated line as the positive supply, and the real ground signal as the negative supply.

3.2.6 Butterworth Filter (Not Used in Final Implementation)

Prior to the implementation of the relatively simple RC networks used above, we used a second-order Butterworth Filter to pass only the signals about which we cared (2 Hz – 20 Hz). The implementation of digital filters eventually made the complexity of the second-order Butterworth unnecessary (see Section 3.3), but for the sake of completeness we include the information on this former section of our analog front-end. Figure 5 shows the schematic for our Butterworth filter.

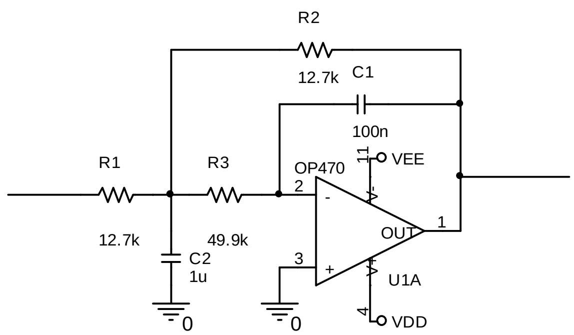


Figure 5: Schematic of a Second-Order Butterworth Filter

The small bandwidth requirement was achieved with this filter while providing the smoothest frequency response in the pass band. By cascading a second-order filter with a simple RC filter, we were able to achieve an overall filter of the third-order. The values of the components used in the filter were obtained by using a computer program intended to be applicable only to filter design.

3.3 Signal Processing

After the brain wave signals have been acquired and conditioned, the signal-processing unit analyzes them and determines whether or not to wake up the user. We apply the method of digital signal processing (DSP), where analog input signals are digitized to increase efficiency, accuracy, and practicality. In the implementation of our prototype, the signal processing unit was constructed on a single prototype board that receives signals from the signal conditioning board via a ribbon cable. Because the microcontroller resides on this board, the pushbutton and LCD interfacing, and main power connections are made on this board.

3.3.1 Digital Low Pass Filter and Decimation

Spectral analysis has revealed that we have undesirable signal content around half of our sampling rate (~ 64Hz). Therefore, we filter it out using a low pass digital filter implemented in DSP. We then use a digital decimation that leaves us with our frequencies of interest. Decimation is commonly used in DSP when oversampling the analog input signal to effectively decrease the sample rate and to prevent aliasing. Unfortunately, decimation filters tend to introduce some signal power distortion at half of the sampling rate (~ 64Hz). Therefore, the digital low pass filter is doubly important.

3.3.2 Fast Fourier Transform

Fast Fourier Transform is an important tool employed in many DSP applications. Basically, information is converted from the time domain to the frequency domain. While there are different methods to implement a FFT, the radix-2 method is often applied because it reduces the number of processor instruction cycles required. FFT analysis results in the division of the power spectrum into a set of frequency “bins”.

Our original plan was to use frequency bins for frequency spectrum analysis, but due to processing restraints we simplified our analysis to viewing signal power at discrete frequency components only. This allowed a quick analysis to be performed without losing significant resolution.

3.3.3 Timing

Due to the limited random access memory (RAM) size of the processor (512 bytes), it was desirable to use the smallest number of data points that produced an optimal frequency spectrum and maximized efficiency. A radix-2 FFT restricts the sequence size to a power of two, and a minimum of 128 Hz (just above the Nyquist frequency of 120 Hz) was needed to avoid aliasing with the noise at 60 Hz. A sampling rate of 128 Hz was chosen because it provided adequate resolution. A rate of 128 samples per second for 128 sampling points results in a one second sampling duration.

3.3.4 The Algorithm

Our algorithm consists of a digital form of sleep staging. As explained in both our Functional Specifications and our Design Specifications, we believe that awakening a person after they have just experienced a stage of rapid eye movement, or REM, sleep will allow the person to awaken with a high level of alertness. Thus, our algorithm relies on detecting the transition from REM to non-REM sleep.

Five different stages of sleep exist: REM, and Stages 1 through 4. REM sleep consists of relatively low voltage, mixed frequency EEG activity and phasic, or episodic, rapid eye movements. Stages 3 and 4 consist of relatively high amplitude, slow frequency (2 Hz – 4 Hz) brainwaves. Thus, the differentiation between REM and stages 3 and 4 is not a difficult one; amplitude of brainwaves alone can be used. While the differentiation of stages 3 and 4 themselves would be much more difficult, for the POSSESS, we do not have to worry about such a problem because we do not need to know which of these stages the user is experiencing.

Sleep stage 2 involves a lower voltage signal than that seen in stages 3 or 4, but whose frequency is in the range of 12 Hz – 14 Hz. Such a narrow frequency range combined with a large voltage, relative to the voltage seen in REM signals, means that differentiating between stage REM and stage 2 is fairly straightforward.

Determining whether or not a person is in stage REM or stage 1 is where the algorithm is put to the test. When a user leaves REM sleep, he/she is more likely to enter stage 1 sleep than any other stage. Thus, the ability of our algorithm to determine whether or not the user is in stage 1 or REM is critical. Stage 1 sleep is comprised of low voltage, mixed frequency brainwaves with a prominence of activity in the 2 Hz – 7 Hz frequency range. Slow, rolling eye movements that can last several minutes are also prominent in stage 1. REM sleep is comprised of low voltage, mixed frequency brainwaves with episodic rapid eye movements. Clearly, these two stages highly resemble each other.

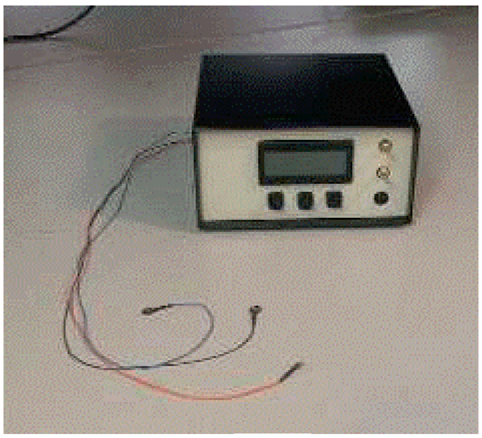
The use of eye movements is what is primarily used to determine whether a user is in sleep stage 1 or stage REM. Slow, rolling eye movements can last for several minutes in stage 1. Thus, a rise in the low frequency components of the FFT derived from the eye movement channel while reading mixed frequency, low amplitude waves from the brainwave channel results in the confirmation of sleep stage 1. Should we be lucky enough to observe phasic rapid eye movements and mixed frequency, low voltage brainwaves, stage REM will be confirmed. Because of the episodic nature of eye movements during REM sleep, however, we are able to more easily confirm stage 1 sleep using the presence of eye movements than REM sleep. A lack of slow, rolling eye movements over several sleep epochs (refer to our Design Specifications for information on how we use sleep epochs during sleep staging) while observing mixed frequency, low voltage brainwaves is regarded as REM sleep by the POSSESS. I.e., the POSSESS will assume that due to the intermittent nature of the rapid eye movements during REM sleep and the short 1 second sampling period of the POSSESS that REM sleep is occurring and that the eye movements were simply not detected.

3.4 User Interface

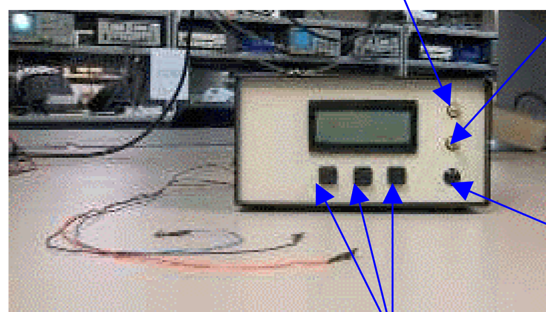
3.4.1 Menu Structure

The user interface is responsible for accepting time, alarm window, alarm on/off, and snooze commands from the user. The user interface makes this data available to the firmware of the POSSESS and allows for proper operation. Additionally, the user interface provides the aural alarm output to the user. Without this output, the POSSESS would be about as useful as a mute auctioneer.

Photos showing the still under-construction case are shown in Figure 6a) and 6b). The majority of information is conveyed to the user via our 4x20 Hitachi 44780 LCD placed on the front of our case. Four push buttons are present on the device; three control the LCD soft menu, while the fourth is an alarm test button. The final version of the case is an improved version of what is shown below.



a)



b)

Figure 6: Mechanical Construction of the Unit

The bottom line of the LCD Display shows an icon above each button that indicates its function in the current soft menu. These icons are programmed into the CGRAM memory of the LCD.

The function of the buttons for the main menu is shown in Table 3.

Table 3: Main Menu Buttons

Button 1	Set Time
Button 2	Set Alarm Window
Button 3	Alarm ON/OFF

If button 1 or button 2 is pushed, the system will enter a sub-menu in which the user can choose their desired settings. For example, if the user pushes the Set Time button, the system will enter the Set Time sub-menu. The icons are be updated to indicate the operation of each button in the sub-menu. The Set Time sub-menu functions are summarized in the following table.

Table 4: Set Time Sub-Menu

Button1	Toggle Between Hours and Minutes
Button2	Increment Time
Button3	Return to Main

The Set-Alarm Window sub-menu is similar to that of the Set Time sub-menu, except that two specific times will be required. Table 5 shows the button layout in this sub-menu.

Table 5: Set Alarm Window Sub-Menu

Button1	Toggle Between Hours, Minutes, and Start/End Times
Button2	Increment Time
Button3	Return to Main

The Alarm ON/OFF button is used to both turn the alarm on and off, and disable the waking tone should it be activated (i.e.: act as the all important "Snooze" button). A volume control knob to adjust the volume of the alarm, to be used in conjunction with the alarm test button, is also provided. No sub-menu is required for this button; when pushed the main menu is updated instantly.

The Power ON/OFF switch has an obvious function. Its operation will be independent of the microprocessor. The Alarm Test button is used to manually generate a tone for the user to audibly assess the current alarm volume level.

3.4.2 Display Selection

In the end, we decided on a 20 x 4-character backlit LCD display. A display of this type meets all of the requirements outlined in our Design Specifications. The majority of LCD displays use the industry standard display controller – the Hitachi HD44780. It has a track record of reliability and is known to have good usability.

As explained in our Design Specifications, we chose an ACM2004D display from AZ Displays. This selection was based on pricing and availability.

3.4.3 Enclosure

As per the requirements outlined in our Functional Specifications and elaborated on in our Design Specifications, we decided to use a Hammond 1458 instrument case as the case for the POSSESS. While somewhat costly (~ $50 CDN), we believe the case adds to the overall marketability and functionality of our product and is thus worth the money. We could have obtained a plastic case for a smaller amount of money, but a metal case is important for shielding purposes.

3.4.4 Buttons

The soft-menu push buttons used on the POSSESS are hooked up to the two hardware interrupts of our Atmel microcontroller. Buttons one and two use a single hardware interrupt while the second hardware interrupt handles the other buttons. The buttons, wired in an active low configuration, are wired through AND logic to the proper hardware interrupts. Additionally, some buttons are wired to additional non-interrupt input pins on the Atmel. Thus, whenever a button is pushed, one of the two active low hardware interrupts is triggered. The microcontroller then checks the non-interrupt pin that is paired with the hardware interrupt that was triggered. The hardware interrupt itself determines that a button was pressed; the additional input pin is used to determine exactly which button was pressed.

4 PROBLEMS ENCOUNTERED

4.1 Technical Issues

Numerous unexpected issues presented themselves as problems during the development of the POSSESS. These issues are outlined below.

- ***Isolation Required in the Front-End***

  A crucial component of our system that was not initially specified is electrical isolation. Electrical isolation breaks ground loops and isolates the user from potentially hazardous voltages. The obvious need to protect our user from hazards such as a surge from one section passing through to the signal conditioning stage necessitates isolation as a crucial component of our system.

  Since analog isolators are expensive and imperfect, digital isolation was chosen. The user can be isolated at two possible points in the signal chain, right after the instrumentation amplifier or before any external connection. Since the analog-digital converter is on the microcontroller, the earliest possible point of digital isolation is between the serial port and the microcontroller. If a separate analog-digital converter were used, simple digital isolation could be placed before the analog-digital converter.

  Instead of opto-isolators, we used the Burr-Brown ISO150, a bi-directional digital capacitive barrier. The ISO150 is placed in the signal chain between the microcontroller and the RS232 transceiver.

- ***Difficulty Completing Our Fast Fourier Transform (FFT) Algorithm***

  While FFT algorithms are readily available, the limited amount of memory and computation power on the 8-bit microcontroller required a less processor-intensive FFT at the price of dynamic range. Floating point numbers on a small microcontroller such as ours were avoided due to the large amount of program memory required. Hence we used 16-bit integers to simplify arithmetic while avoiding overflow. Sine and cosine lookup tables had to be implemented to facilitate reasonable execution times.

- ***Difficulty Acquiring Brainwave Data***

  Waking brainwave data is drastically different from sleeping brainwave data. Sleeping brainwave data was relatively difficult to attain because of the relatively late completion date of our FFT algorithm. To gather sleeping brainwave data, a subject should to sleep in a lab overnight while data is collected. Because of issues regarding electrode stability and reliability that were not solved until relatively late in the semester, this ideal situation did not prove possible. Eventually, we solved our technical issues and convinced Roch to sleep in Lab 1 for a few days.

- ***Unreliable Breadboards***
  The breadboards we used for the project were riddled with faults (e.g.: open connections) that made early testing very difficult. This was solved after we transferred our circuits from breadboards to soldered prototyping boards.

- ***Limited Program Memory Available in the Atmel 8535***
  While we originally thought 8 kilobytes of Flash RAM would be *more than* sufficient for our application, in the end it turned out that 8 kilobytes of Flash RAM was *barely* sufficient for our application. This lack of memory forced us to spend several days optimizing our code and it limited the complexity of our sleep staging algorithm.

- ***Difficulty Completing Our Sleep-Staging Algorithm***
  Limitations in our Flash RAM, difficulty in acquiring consistent brainwaves, difficulty completing portions of the project critical to staging algorithm development, and the inherent unpredictability of brainwaves all made our sleep-staging algorithm very difficult to refine.

4.2 Financial Issues

Table 6 is the original budget forecast submitted by QND Medical Devices in our Design Proposal. Table 7 is the actual amount of money we spent.

Table 6: Original Budget

Equipment		**Cost ($)**
Bio-Medical	Electrodes	300
	Sensor Hat	120
	Skin Conditioners	20
	Cabling	10
Electronic Components	Amplifiers	20
	Microprocessor EVB	250
	LCD Screen	40
	Power Supply	40
	Miscellaneous Electronics	50
Hardware	Instrumentation Case	50
	Circuit Board	20
	User Interface Peripherals	30
Software	Microprocessor Development Tools	85
	Electronic Simulation Tools	100
Estimated Total		**1135**

Table 7: Actual Money Spent

Equipment		**Cost ($)**
Bio-Medical	Electrodes	120.00
	Sensor Hat	Sewn by Dave's Mom
	Skin Conditioners	71.63
	Cabling	Included with Electrodes
Electronic Components	Amplifiers	81.72
	Microprocessor EVB	Borrowed from Spectrum Signal Processing
	LCD Screen	Donated by iZ Technologies
	Power Supply	19.38
	Miscellaneous Electronics (opto-isolators, voltage regulators, digital-to-analog converters, UART IC's, etc.)	209.63
Hardware	Instrumentation Case	62.70
	Circuit Board	21.38
	User Interface Peripherals	7.50
Software	Microprocessor Development Tools	12.25
	Electronic Simulation Tools	Downloaded
Miscellaneous	Library Book Overdue Fees	30.00
	Shipping Fees	85.00
	Customs Fees	30.26
Total		**751.45**

Even with the addition of a fairly expensive group of miscellaneous items, we still came in significantly under budget. This is a testament to the equipment we were able to coax out of friends in industry, to the skills of Dave's mother, and to the generous software authors who post Freeware on the Internet.

4.3 Scheduling Issues

Now this is where it all blew up in our faces.

Figure 7 is the Gantt chart submitted with our Design Proposal late September, 2000.

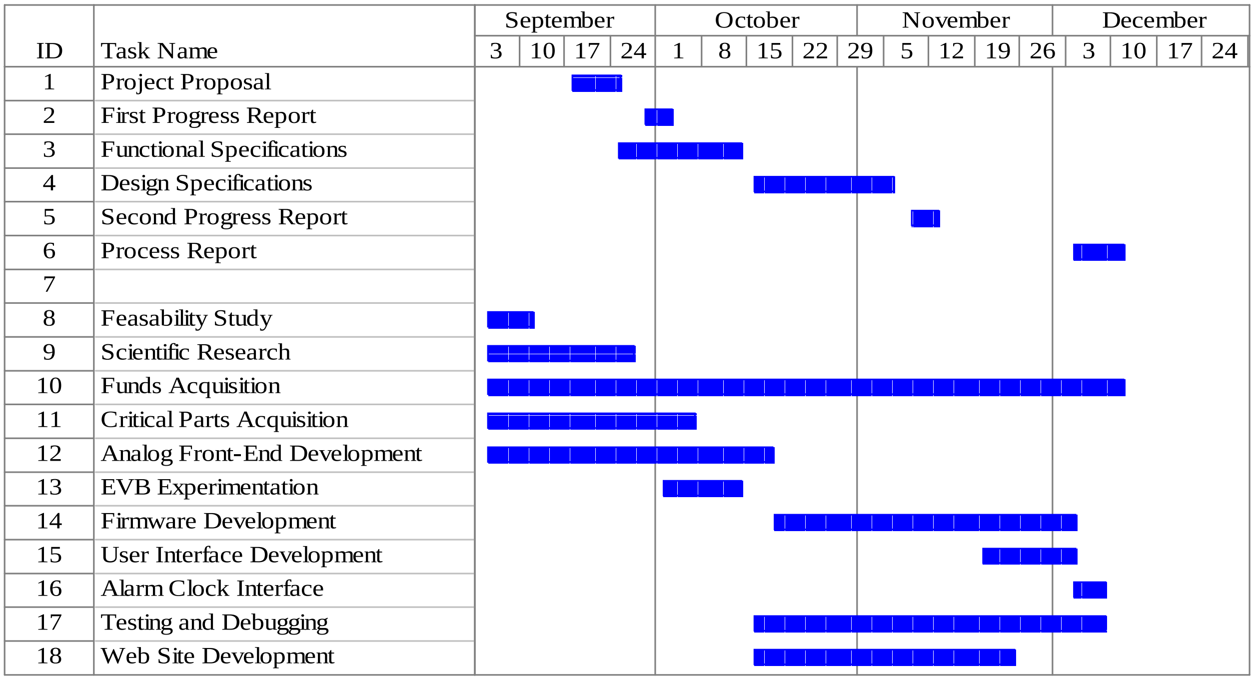


Figure 7: Original Gantt Chart

Figure 8 is what actually happened.

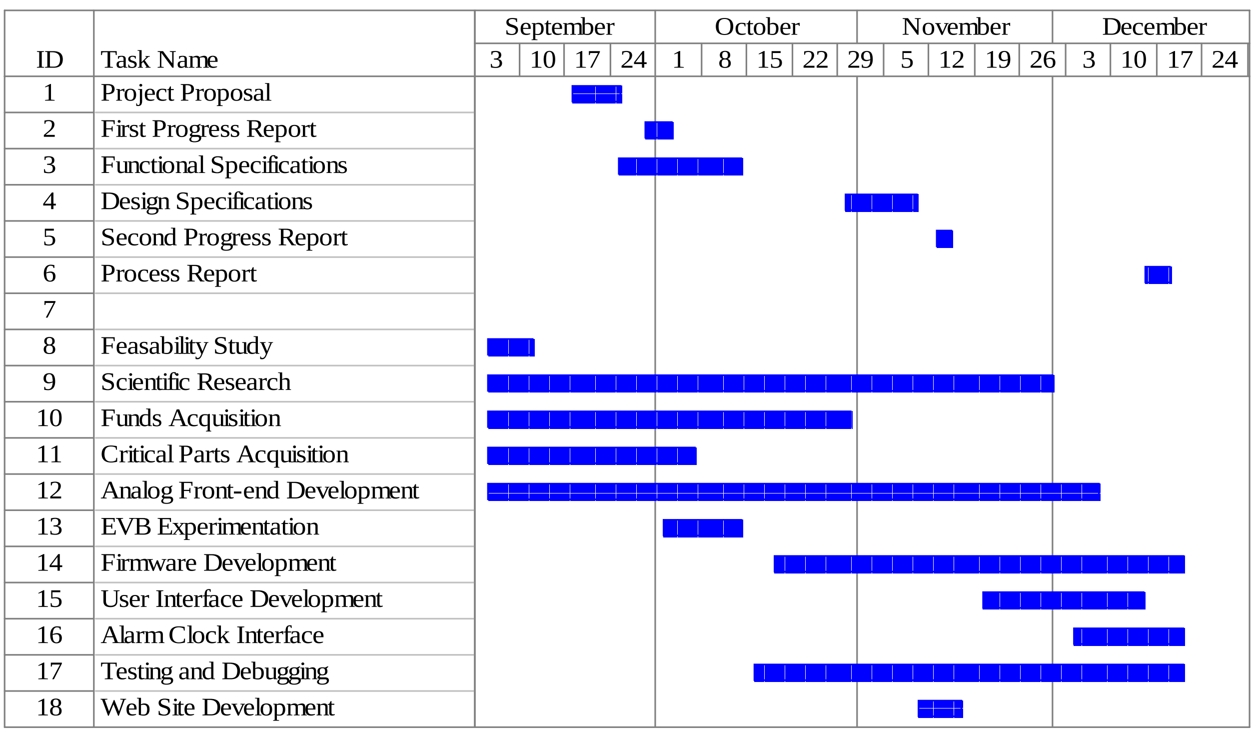


Figure 8: Actual Gantt Chart

Heuristically, two main differences can be observed between Figures 7 and 8. Firstly, the time we actually spent on our documents was slightly less than the time originally allocated for them. This was due mainly to time restrictions placed on group members by other courses. We were, however, still able to create high quality documents in a shorter amount of time than originally thought possible.

Secondly, testing and debugging took longer and affected more aspects of our design than we originally thought it would. We originally budgeted a relatively large amount of time for testing and debugging; clearly, the actual amount of time required was even larger than originally anticipated. Additionally, the large amount of debugging we had to do affected the completion dates of other entries in our Gantt Chart. Specifically, the completion dates for the analog front-end and microcontroller firmware had to be extended.

One thing we noticed regarding testing and debugging is that testing and debugging leads only to more testing and debugging. Thus, we create a sort of paradox. We test and debug in order to solve a problem, but testing and debugging reveals more bugs whose solutions we must test and debug. Eventually we solve all the bugs of which we are capable of solving, but not until we had put in many more hours than we originally thought necessary.

4.4 Group Dynamics Issues

Considering the number of hours the members of QND spent in close proximity together, we think that the number of problems we had regarding group dynamics was actually quite small. The main source of group dynamic problems occurred whenever one person tried to offer "constructive criticism" into an area of the project assigned to another person. Time and time again, we noticed that after one person has spent weeks on one part of a project that he doesn't really appreciate any form of criticism from a person who has spent only five minutes considering the problem. It was only through numerous scoldings that those who had devoted weeks learned tolerance and those offering advice learned tact.

A second group dynamics problem was the disintegration of our weekly team meetings. Over the course of the first half of the semester, we held weekly meetings with set agendas that were useful in terms of project progression. As the semester grew busier with other coursework, however, project progression slowed and with it so did the frequency of our meetings. Without a weekly meeting, however, procrastination became more of a problem. Thus, while meetings may seem like a waste of time during times of slow project progression, they are still valuable in that they do eventually spur project development.

5 WHAT WE WOULD DO DIFFERENTLY

- ***Use EMG in Addition To/In Place of What We Have Already***

  As mentioned above, the main task for our algorithm is to differentiate between REM sleep and stage 1 sleep. An additional channel consisting of an EMG, a reading that accurately detects muscle movements, would have made stage 1 and REM stage differentiation potentially easier and more reliable than what we have now. During REM sleep, muscle movements are non-existent. Thus, if we had detected low amplitude, mixed frequency brainwaves accompanied by muscle movements, which are fairly constant during stage 1 sleep, we could have been virtually guaranteed a confirmation of stage 1.

- ***Reduce Complexity of the Front-End Circuitry***

  Even a brief examination of the schematics for our hardware shows that quite a complex signal conditioning stage exists before the analog-to-digital conversion at the microcontroller. In fact, in earlier stages of development, the analog front-end was much more complex than it is now. We feel that by the use of more powerful software routines we can further simplify the front-end. As results, the device could be made smaller, and less expensive both in terms of component costs and manufacturing time.

- ***Use an External Analog-to-Digital Converter Instead of the On-Board Facilities***

  Although the Atmel AVR8535 has on-board analog to digital conversion facilities, the speed and quality of the conversions are no match for those found in stand-alone conversion integrated circuits. Use of an external A/D converter would also provide more flexibility, in that the microcontroller would no longer have to have the ability to perform conversions.

- ***Use a Digital Signal Processor Instead of an AVR Microcontroller***

  The AVR microcontroller used for this project met our needs quite nicely. It was relatively easy to program via a C compiler, it had good documentation, and several useful resources existed on the Internet. In particular, the UART provision of the AVR that allows communication with any personal computer for debugging purposes was used frequently over the course of the project. As with any device, however, it had a few shortcomings that may lead us to try a different processor in the future. First and foremost, the 512 bytes of on-board random access memory (RAM) didn't prove to be enough. Our signal-processing algorithm requires that we sample data into this memory space for a period of time before performing any calculations, and we don't feel that we are able to gather enough samples to achieve optimum accuracy in our calculations. Using a digital signal processor (DSP) from perhaps Texas

Instruments or Analog Devices would provide a lot more RAM as well as a higher clock speed to us, while cutting down on the number of extra features, such as A/D conversion and an on-board UART.

- ***Use Lower-Cost Parts***
  In trying to achieve optimum results, we have used some component parts that are of an unnecessary level of quality. As a result, should the device go into production in its current design, it would likely have a high cost. Therefore, we would like to experiment with using lower-cost components, and see if the results obtained with a less-expensive design are comparable to those of our original prototype. For the prototype, many of these high-performance integrated circuits were obtained as free samples from their manufacturers.

- ***Improve Testing Procedures***
  Unfortunately, our attempts to get real sleep data from a sleep center failed due to time and financial concerns. As a result, we were forced to use group members as test subjects for the device. Naturally, we would like to rigorously test the POSSESS under real, controlled conditions. We will undertake to do so in the future.

6 WHAT WE LEARNED

David Lee, VP Operations

Ensc 340 provided me with a great opportunity to further my practical knowledge that I gathered from my co-op terms. The technical experience in this course mainly came from hardware assembly, testing and debugging. The testing and debugging skills are essential to getting anything to work and I am glad that I had an opportunity to spend time like this. My other technical contributions included interfacing the microcontroller, the amplifiers and the LCD. This is in addition to building and designing the case.

I had a lot of fun working on the actual case and making it to be an impressive product of QND. I also overlooked our development tools used to ensure that the development process does not have bugs related to equipment and only due to design. Another role I played on is chairing the group meetings. I felt that it be important that someone other than the group leader keep the group meetings going and generate constructive discussion from all group members.

This project allowed me develop my teamwork skills, and learn more about hardware interfacing. I enjoyed contributing and I have learned the important organizational skills required for a successful project. Had we been more organized, more meetings would have taken place and we would never have fallen off schedule.

Stephen Liu, VP Engineering

340 is not a course in the strict sense; it is an exercise in self-discipline, patience and frustration. The end result is very much a reflection of anxiety and obsession. To accomplish something in 340 requires diligence beyond compare; it demands passion. The course load this semester, as demanding as it was, paled in comparison to the time and thought spent agonizing over designs, methods and minutiae. Four months of constant worry, bogging down the everyday struggle wears on a person. Each and every sleepless night tracking down parts and writing algorithms bears on the soul a greater burden. And what is it but engineering to realize that time spent was wasted, not entirely in vain, for the beauty in simplicity is difficult to come by.

It is not the course that is important, it is the self-discovery that it brings out that is valuable and insightful. These trials by fire are not exactly pleasant but then again it is rarely rosy what finds in oneself. One does not create anything from nothing in these four months just as void remains void. It is from within that one must reach to take and bring into existence. And for that I am happy but spent.

Hiten Mistry, Chief Financial Officer (CFO)

The technical experience found in ENSC 340 is extremely valuable. I've familiarized myself with another microcontroller and programming it with its compiler. Learning to use the Atmel, a completely different microcontroller, is refreshing since I have previously used the Motorola HC05, HC11 and HC12. Through this experience I am able to make better comparisons between these brands and apply this insight to comparing a wider selection of microcontrollers. One of the more visible accomplishments with the microcontroller I had was contributing to the user interface (push buttons and LCD menus). Although my focus was on the software side, I have also learned about analog problems and debugging techniques through listening to team members.

Another area of contribution on my part was the research. Going out and finding out about how to read brainwaves with a series of electrodes, finding common problems with such a system (ie. noise from muscles, electrode type, electromagnetic interference), and also learning about some of the safety issues involved with our product if it is to be sold on the market. This research is something that must be collected and analyzed in advance before all the different technical components are pieced together, the objective of our functional specifications. Having a solid understanding of the science behind the application proves to be vital for the complete project to function correctly in the end.

The research and troubleshooting side is what I enjoyed learning about the most. I have always had an interest in engineering used for biomedical applications and I would have liked to spend more time on the project. Unfortunately the course load I had taken turned out to be a little more than I could handle and a lot more that what the engineering department had perceived it to be when they planned it. I regret not having more time for this course. This is a downside of being the guinea pigs of the new curriculum. At the end of it all I believe I would have been able to extract much more from ENSC 340 had it not been for the inattentive curriculum planning of this semester. In the end, ENSC 340 is an important part of our degree and a class we are all proud of taking.

Roch Ripley, Chief Executive Officer (CEO) and Team Leader

Over the course of ENSC 340, I learned a great deal both about technical and management issues.

Concerning technical issues, I learned a fair deal about programming microcontrollers and analog electronic design. Instead of having to program microcontrollers in assembly language, which can make code integration and the design of large programs very difficult, I learned that numerous hobbyists have set up websites to offer guidance on programming in higher-level languages. Thus, with the help of the Internet (and Steve Liu), I gained valuable experience in programming microcontrollers in C. Doing so entails considering issues such as memory availability and allocation and code efficiency.

From the analog front-end I learned about analog electronic design. The front-end allowed the application of a great deal of “textbook knowledge.” I.e., the design of multiple pole active filters, the removal of 60 Hz noise via active shielding, and the use of devices such as voltage regulators and various types of IC’s. I was also able to apply knowledge from my co-op terms, especially from my most recent term at PMC-Sierra, to help with the project. Finally, I realized that one of the best sources of information is from application notes or datasheets that can be obtained directly from manufacturers; almost every IC has intended application information available from its datasheet.

Regarding management issues, I learned the importance of delegating responsibility. Instinctively, I think that a lot of engineers just want to work on the technical side of a project and leave the planning, documentation, and marketing to someone else. That just doesn't work in a project like this. Five people coding with one computer gets produces headaches. Five people trying to work on one circuit results in yelling. Five people doing five different tasks, albeit not all of them technical, results in a great deal of productivity. In a project as large as ours, egos have to be put aside for the greater good. While we didn't have five people constantly working on the technical aspect of our project continuously, we did have people working on our project's case, marketing, documents, and final presentation. Via the division of labour and the putting aside of egos, I believe we were able to create a better overall product than if we had focused only on circuits and algorithms.

Finally, regarding conflict resolution, I learned that the best way to resolve arguments is to listen to the points of view of all involved. Generally, engineers only become angry when they feel that their opinions, which are generally based on years of hard work and experience, are ignored. If everyone's opinion is taken into consideration and a proper decision made only after all points of view have been heard, then the majority of conflicts can be avoided.

Matt Ward, VP Marketing and Public Relations

Engineering Science 340 has afforded me opportunities to improve my skills in a number of areas. Both technical and inter-personal skills were learned, improved, and tested. Creativity was incorporated to a level not usually found in engineering courses. Certainly, a lot of work was involved, but I feel that the experiences gained are well worth the effort.

Within the group, I performed mainly hardware tasks, together with a few software tasks mixed in for good measure. I was also responsible for parts of the testing and debugging, and many of the component characterization tasks. As such, I have gained experience in debugging microcontroller-based hardware. In addition, I can now add AVR assembly coding to my growing list of low-level software experiences. From the front-end circuitry, I have learned about active filtering circuits. I am now more appreciative of good technical documentation, as I have seen datasheets and application notes ranging from useful to unusable.

Previous courses have given me opportunities to work with microcontrollers, but an evaluation board was always used. I have learned in this course that constructing a single-chip system is not a trivial task. Persons responsible for parts selection and procurement for engineering firms now have my utmost respect, as these tasks were more difficult than originally anticipated. Lastly on the technical side, I have perfected the art of soldering in tight spaces with the large soldering irons in the lab.

As another result of ENSC 340, I have fine-tuned my inter-personal skills. I have learned that planning is often the most important stage of any project, including ours. A cohesive plan, started with the Gantt chart in the Project Proposal document, and continued with the Functional and Design Specifications can make a project go very smoothly. It is also important to note, however, that design is an iterative process, and provisions to allow for minor changes should be made. Good documentation is a clear form of communication. Lack of clear communication is wasteful of time and energy, and can lead to large amounts of unnecessary work being done. With our project, I found that working in a centralized location helped keep the entire group "on the same page", and allowed each member to draw upon the skills of others when he had difficulties. In addition, enlisting help from external experts was paramount to the success of our project. Dr. Weinberg and Dr. Mistlberger provided valuable insight, and numerous experts who posted materials on the Internet were helpful with research and technical implementation advice.

Taking ENSC 340 together with three difficult four-credit engineering courses over the past 13 weeks has been quite an experience. The hours have been long, and brief feelings of exhilaration have interrupted continual periods of frustration. Regardless, looking back, I feel that I am a better person for the experience.

7 Future Enhancements

- ***Prototype with Surface Mount Parts and Custom PC Boards***

  The final prototype for this project consists mainly of many dual in-line package IC's on multiple generic Radio Shack prototyping boards. This solution is cost and time-efficient, but it introduces problems of noise and size, both of which are not desirable for the sensitive nature of our project. Our project depends on having maximum signal integrity at all times. If more time and funds were available, we would definitely use custom-made PC boards with surface-mounted parts for our prototype.

- ***Pay More Attention to Isolation Issues***

  Our prototype uses isolators to separate the RS232 debugging interface from the rest of the hardware, in effect separating any connected personal computer from the device, and hence the user. Because the POSSESS is powered exclusively by batteries, the user is protected from any AC power sources, including those connected to the PC. In the future, we would like to increase the use of isolation in the device, perhaps isolating the user from even the battery-powered section of the circuit. It could also be possible to devise an isolation solution that is cheaper and simpler than our current implementation.

- ***Use Fewer Batteries***

  Eight AA batteries connected in series, providing a total of 12V, currently power the prototype. With a few changes to the circuit, we feel that we could run the device from four AA cells. Also, battery life could be improved by the use of lower-power circuit components. All in all, we could enhance portability and reduce operating costs by designing the hardware to work with lower voltages and lower power.

- ***Explore the Use of Active Electrodes***

  In doing our research for this project, we have discovered that many researchers have achieved very good brainwave readings by using active electrodes, instead of the passive ones we have used for our prototype. We would like to try some active electrodes with the POSSESS, but again, time and financial concerns have prevented us from doing so.

- ***Explore Wireless Electrode-to-Processor Interface***

  If we could some how discover a way to package the electrode and signal conditioning elements into a small package, it could be possible to separate them from the signal processing and user interface sections of the system, transmitting conditioned data over a wireless link. Advantages to this setup would be that the electrodes and signal conditioning stages could be packaged in some sort of a sleeping cap, and no wires would need to travel from the bed to the processing unit. Of course, the safety issues of placing an RF transmitter on a person’s head would have to be carefully considered.

8 GRATUITIES

We would be remiss not to thank the following people who have so kindly assisted us with our project:

Andrew Rawicz, Steve Whitmore, Harold Weinberg, Ralph Mistlberger, Patrick Leung, Jason Rothe, James Balfour, Greg Hall, and Paul Gurney.

Without each them, the POSSESS would not have progressed to its current stage.