Abstract — The ECG (electrocardiogram) is a non-invasive system that provides critical data of a patient’s heart activity. The adaptive ECG filter will be designed to reduce noise caused by MRI (magnetic resonance imaging) machines to produce a diagnostic quality ECG signal. This project aims to help MRI operators make fewer mistakes by making some of the choices for them. This project also intends to deliver better results than the linear filters that are commonly employed. Some of the most common examples of noise that the ECG filter would need to remove in order to provide useful results includes power line interference, electrode contact noise, motion artifacts, muscle contraction, and baseline drift. Unfortunately with the large number of MRI machines it is almost impossible to teach everyone how to use each of them. This requires us to create a system that works for each user regardless of their experience level.

Index Terms — Filter, Data Acquisition, Analog-Digital Conversion, Digital-Analog Conversion, Microprogramming

I. INTRODUCTION TO ECG FILTERING

The ECG test is a non-invasive system that provides critical data of a patient’s heart activity. While a patient is undergoing a MRI scan, the MRI machine radiates electromagnetic energy that interferes with the electronic ECG signal received from the patient. This electromagnetic radiated energy causes some of the ECG data received during a MRI to be corrupted and therefore useless. Our adaptive filter removes the need to manually select a filter and allows the medical professional to better concentrate on the patient and less on the equipment. The filter system is a unit that attaches to the ECG monitor and is connected inline between the ECG sensors and the ECG monitor. The filter also has an optional connection to a computer via a USB port. The box has four input ports from the ECG sensors, four filtered output connections (1 for each electrode), and 1 USB port. The internal part of the unit consists of an analog front end with an amplifier and filter, an analog to digital converter, microprocessor, and a digital to analog converter. All of these components are soldered to a printed circuit board manufactured exclusively for the adaptive filter. This design allows the unit to be attached without any modification to existing ECG monitoring equipment. Since there is no modification to the monitoring equipment, the device will be universal to many different monitoring equipment manufacturers.

II. PROJECT COMPONENTS

The analog front end provides patient protection, amplifies ECG electrode signals, and filters the ECG electrode signals. The analog to digital converter takes the signals received from the electrodes and makes the signals usable for the microprocessor. After receipt of the digital data, the microprocessor will then perform the Least Mean Square filter on the incoming data. The Least Mean Square Filter was chosen due to its common use in medical imaging and the inclusion of it within ARM’s CMSIS library. The incoming data is processed by the Least Mean Square Filter, sends the processed data by UART to a computer. MATLAB is then used to the results on nine different graphs. Three graphs are the original signal with noise, three graphs are just the noise from the gradient leads, and the last three is the filtered signal.

Figure 1 - Overall System Diagram
III. PCB CIRCUIT DESIGN

The Adaptive ECG Hardware consists of a 6 layer PCB (Printed Circuit Board). The board contains 417 components and measures 3.7 x 5.4 inches. The device is powered by a 5V external battery. The Circuit was designed using Cadence OrCad and was went through a Quality Assurance review at Philips. The Quality Assurance review consisted of 2 electrical engineers reviewing the design for incorrectly connected components, ensuring that correct component values were used, and reviewing datasheets to ensure correct components were selected. After the design was approved it was transferred onto a PCB using Cadence Allegro. Once the board layout was completed the Gerber file was sent to a PCB manufacturer that printed the circuit board with a 5 day turnaround. Once the PCB arrived components were cataloged and send with the PCB to a PCB assembly house that populated the board for us. A schematic of the PCB can be observed below.

![Figure 2- PCB Schematic of adaptive ECG Filter](image)

The Device has 4 inputs that come from the ECG Electrodes attached to a patients body and 6 inputs that come from a gradient antenna. The 6 inputs are then reduced to three signals using a differential amplifier. The device then filters, amplifies, and biases these signals while providing ESD/Defibrillation protection for the patient and the board. After the front end signal processing is completed the signals are received by the ARM Cortex M4 processor that converts the signals to digital and performs the digital filtering process. The processor then transmits the results of the filtering to an attached computer via a USB port.

A. PCB Isolation

Isolation is required on the PCB to prevent the high frequency noise caused by the digital input from coupling with the analog ECG input and corrupting the ECG signal. Isolation is achieved by separating the analog and digital power supplies and ground planes with a capacitor-input filter commonly referred to as a CLC pi filter. The filter works by shunting noise to the digital ground before the power reaches the analog ground. A figure of the CLC pi filter configuration can be observed in the following figure.

![Figure 3-CLC π Filter (Removed due to NDA)](image)

B. Power distribution

The PCB consists of components that need both analog and digital power sources of different values. This requirement is achieved through the use of linear voltage regulators and a voltage reference. The PCB is powered by a battery that provides 5VDC. The 5VDC is connected to the CLC pi filter which provides a 5VA output. Both the 5VDC and 5VA are connected to linear voltage regulators to step down their voltages to accommodate the needs of various components. A voltage reference is also connected to the 3.3VA supply to prevent any variation in supply voltage to the analog to digital converter.

C. ESD/Defibrillation Protection

The adaptive ECG Filter is designed to be used on a person in a clinical environment and because of this, it must be able to withstand an electrical impulse from an Automated External Defibrillator. The device satisfies this requirement by providing ESD/Defibrillation protection with a combination of diodes and glass discharge tubes. Clamping diodes are connected after the ECG input and before the first OPAMP to short any currents outside the range of +/- 3.3V to provide basic protection. There is also a Littelfuse glass discharge tube connected after each ECG input to short the current in the event that an Automated External Defibrillator is used. A schematic of how the diodes and discharge tube are oriented can be observed in the figure below.

![Figure 4 - ESD/Defib Protection (Removed due to NDA)](image)

D. Analog Filtering

The device performs front end filtering to eliminate as much noise as possible before the analog signal is converted into a digital signal. The front end filtering is performed using a 3rd order unity gain Sallen-Key filter with a cutoff frequency of approximately 55,000 rad/s or 9kHz. 9kHz was the chosen cutoff frequency because all signals above
that frequency are unwanted noise and it falls within the Nyquist rate of 250ksps. The configuration and the frequency response of the Sallen-Key filter can be observed in the figures below.

Figure 5 - Sallen Key Filter (Removed due NDA)

![Frequency Response of 3rd Order Sallen Key Filter](image)

Figure 6 - Frequency Response of Sallen Key Filter

E. Op-Amp Biasing and Amplification

In order for the analog to digital converter to work properly, the Op-Amp going into the processor is biased. The biasing is achieved by applying a 5V source and using a voltage divider with 2 resistors to step down the voltage to 1.2V. The same voltage divider process is used to reduce the input signal from the ECG from +/- 3.3V to +/-0.55V. These two signals are added together and go into the Op-Amp with a value between 0.65V and 1.75V. The Op-Amp is configured to have a gain of 1.5 which amplifies the signal values to be between 0.975V and 2.625V. The signal with a range between 0.975V and 2.625V is then transmitted to the processor for analog to digital conversion.

IV. ANALOG TO DIGITAL CONVERSION

The ARM Cortex M4 processor has an integrated delta-sigma analog to digital converter (ADC). The processor contains two identical converter modules that share the same 24 analog input channels. Each ADC module operates independently, supports up to 1MSPS sampling rate with 12 bit precision, and operates on a 16 MHz clock rate. The main advantage of this converter is that it operates using a delta-sigma encoding method. The delta-sigma method is preferred because it balances speed with precision. Other conversion methods such as SAR (successive approximation register) or Flash tend to favor either speed or precision over the other.

The ADC collects sample data by using a programmable sequence based approach. A sample sequence is a series of consecutive samples that allows the ADC to sample multiple times at once. There are 4 different types of sequencers. Sequence 3 allows for 1 sample and a FIFO size of 1. Sequence 2 and 1 allows for 4 samples and a FIFO size of 4. Sequence 0 allows for 8 samples and a FIFO size of 8. The sequencers can be applied on multiple pins or on the same pin. Once the sample has been collected the result is stored in a FIFO. The FIFO are simple circular buffers and each FIFO entry is a 32 bit word, with the lower 12 bits containing the data. When a sequence completes the resulting data can be retrieved from the FIFO and then transferred into main program memory.

The data retrieved from the ADC depends on how many bits of resolution the ADC has, described by Equation 1. The ADC has 12 bits of resolution so the value retrieved will be an integer from 0 to 4096. Converting this ADC code into the voltage value it represents is done through Equation 2. \( V_{REFP} \) is the positive reference voltage and \( V_{REFN} \) is the negative reference voltage. In our case we are using the internal reference voltage of 3.3V and ground for the negative reference voltage. Analog input voltages above 3.3V saturate at 4096 while those below 0V saturate at 0.

\[
0 \leq \text{ADC code} \leq 2^{12} \text{ bits of resolution}
\]

Equation 1 – Range of ADC code

\[
V_{ADC} = \frac{V_{REFP} - V_{REFN}}{4096} \times \text{ADC code}
\]

Equation 2 – Convert ADC code to voltage

The device has 3 inputs for ECG signals and 3 inputs for gradient signals which gives a total of 6 different signals to run through the ADC. The sample sequence 0 is able to convert these 6 signals all at once into a digital value with no speed reduction. To achieve enough resolution on the ECG signals, each signal must be sampled at least 500 MSPS. For a 50 kHz input signal that is 10 samples per period. Each sequence sample is triggered by a processor interrupt, waits for conversion to complete, copies data from FIFO into main memory, and then repeats continuously.

V. LEAST MEAN SQUARES FILTER

The Normalized Least Mean Squares Filter is an adaptive filter that takes in two different signals. These are the input
signal and the interference signal. After processing the signals the algorithm outputs an “error” signal. The error signal is the difference between the output and the reference signal. Despite its name, it is actually the filtered data from processing the input signal and the interference signal.

As the Least Mean Squares Filter is adaptive, the results will steadily improve and the coefficients become closer to their optimal values. The coefficients change after every set of input points. This is why the data near the start of the program is rather noisy and improves drastically soon after.

In the attempt to deliver accurate results a step size which allowed an optimal amount of learning was necessary. This requirement was solved by using the Normalized Least Mean Squares filter as opposed to the Least Squares Filter. The Normalized Least Mean Squares Filter is normalized as per the name and the optimal step size for this algorithm is one. This is the case if the interference and input signal are not related each other which is the case for this project.

As shown in Figure 7, the Least Mean Squares Filter is able to remove a lot of noise from an inputted signal and delivers fairly accurate results. The inputted signal in this example is a very noisy sin wave and the output is displayed under “Signal”.

Due to the excellent results received in the Figure 7 and the common use of the Least Mean Squares Filter, it seemed to be the best choice. Also, since it is an adaptive filter its coefficients will change to filter out the signal based on the inputs. Although the Least Mean Squares Filter does do an excellent job with respect to filtering the data; it unfortunately does require a bit of memory to utilize.

While overall the choice of the Least Mean Squares Filter was a good one; when implementing the filter with the ARM Cortex M4 there were problems with the amount of memory stored. To resolve this issue it was necessary to decrease the quality of the algorithm decrementing the number of blocks recorded and remove any excess code in our application.

VI. CMSIS

CMSIS stands for Cortex Microcontroller Software Interface Standard. CMSIS is used to allow Cortex-M silicon vendor products to migrate over to other products without a large change in source code. The main proponent to using CMSIS is the ability for software re-use.

The CMSIS libraries are optimized with regards to the Cortex M4 processor. It does this by using the SIMD instruction set that the Cortex M4 has. Many benefits were found by using the CMSIS libraries. The primary benefit being the Least Mean Square algorithm include within it.

Inside the CMSIS libraries is an optimized Least Mean Square library. The Least Mean Square library has been implemented for 16 bit and 32 bit systems. The 32-bit float system was chosen over the 16 bit integer and 32-bit integer systems due to us not needing to scale coefficients or having to deal with Overflow and Saturation. Also, our system while needing a decrease in orders was usable by our microcontroller despite the changes.

The implementation of the Least Mean Squares Filter with CMSIS required an initialization; in which the coefficients can be pre-weighted to a selected value. After initialization, the coefficients will converge to a final results. As such, the Least Mean Square Filter is required to remember multiple data points for obtaining the filtered signal.

Using the CMSIS library will hopefully allow other research teams to further our current development and continue to improve the quality received from ECG Signals.

VII. MICROPROCESSOR IMPLEMENTATION

The implementation of the Least Mean Squares Filter was done using the ARM Cortex CMSIS library and embedded C++. Embedded C++ was chosen to allow a more object oriented approach than straight C. This allows
the use of three instances of the filter without a lot of extra code. As such, if more signals were inputted this project would be able to filter more of them at once.

The filter is attached to an Interface, called IFilter. This interface has a filter attached to it that processes the data. Performing the filtration in this manner allows changes to easily be done between different filters or to add changes on top of the existing system.

The architecture of the system utilized a Top-Down approach so that the underlying code importing data, the code filtering the data and the code sending the data were separated. One such way this was done was with a Publish/Subscribe system.

The Publish/Subscribe System currently only sends data to MATLAB, but, it was written in such a way to allow it to send data to any waiting programs that the device has access to. This will allow the program to also send the data by way of a digital to Analog device. This will allow further processing of the signal and the display in legacy devices that Philips currently has.

VIII. PUBLISH/SUBSCRIBE

A publish/subscribe system was implemented to allow the sending of data to multiple publishers. The publish/subscribe system does not contain a Topic Manager and as such requires the publishers to know about the subscribers. The subscribers are listed as objects inside the publishers.

Essentially, a publisher will contain a MATLAB subscriber which when passed data will send it to MATLAB and any other subscribers attached. Another subscriber is a Digital to Analog subscriber. Essentially creating an analog signal from our filtered data.

To implement the Publish/Subscribe system correctly, a system was initially implemented in Visual Studio C++ prior to receiving the boards that used a Topic Manager. However, this system was not up to the task for the Embedded Device and was later scrapped for a simpler model that took up less memory.

Figure 8 - Class Diagram of Publish/Subscribe

Figure 8 the modified Publish/Subscribe system. Utilizing this method allowed the program to work. However, the subscriber list was not implemented as a list and they were simply declared within the publish class to ensure that memory allocation happened prior to runtime.

IX. MATLAB

MATLAB, which stands for MATrix LABoratory, is a high level language and an interactive environment that lets you focus on the problem at hand rather than worrying about the details in programming. It contains many numerical libraries and prebuilt functions that make writing programs faster than in a low level language like C.

There are no declaring of variables, allocating memory, or pointers, so programming in MATLAB is much less error prone. Therefore, you are able to spend more time developing algorithms to solve your problem than worrying about the low level details. MATLAB is built for technical computing. Thousands of engineering and mathematical functions are available, eliminating the need for you to code them yourself and test them.

These features within MATLAB allowed the initial testing of an LMS filter and further creation of plots to display the filtered values. MATLAB has support for COM which the Microprocessor uses to communicate. With the prebuilt libraries in MATLAB it was a lot simpler to record the data in the MATLAB graphs as opposed to creating the software. It is also rather simple to use and while having less compatibility than a more low level language seems to be a lot easier to implement.

Another consideration for the display of data was to use C++ and the BOOST libraries. However, due to the learning curve of the libraries and the easy to find programs in MATLAB with COM ports and graphs, the choice of using MATLAB was fairly straightforward.

X. MATLAB IMPLEMENTATION

The MATLAB program was built in a modular way to allow multiple graphs to be created. In this project a total of nine graphs were displayed, but it is very simple to decrease or increase the number of graphs displayed within the MATLAB program.

Inside the program, data is received in real time and as such; results must also be displayed in real time. This displays the filtration of the objects immediately and requires a limit to the amount of data objects that can be shown. The data will show a set number of points, currently thirty points were chosen as it seemed that thirty points gave an accurate representation of the data. The number of points is very easy to change and can be used to display more or less data at once.

There are three components to the MATLAB program, the COM port connection, the Graphing of the Plots and the data stored in the plots. The COM port connection needed to be set to the same rate as the UART in the Microprocessor to receive data at the correct rates. The COM port was essentially used to receive the data to plot the different values.

One of the problems that occurred within the data sending of the values was an invalid header file. The program would send correct data initially to MATLAB if MATLAB was started prior to the device running. However, if the device was already running and it sent data
to MATLAB, MATLAB would receive data in the incorrect order and would display the results in an incorrect graph. This solution was remedied by forcing the application to send the graphs position twice prior to plotting.

XI. ARM CORTEX M4

The microprocessor used for the Adaptive ECG filter is the ARM Cortex-M4. The M4 is the most powerful member of the M series of processors which consists of the M4, M3, M1, M0+, and M0. The M series of ARM processors is designed to have low power consumption and a sleep mode to reduce power consumption during times when power is not needed. The ARM M series is also supported by CMSIS. CMSIS contains several DSP functions to aid us in the filtration of the data. The M4 supports 32 bit processing. The Cortex M4 is the last in the line of processors and is recommended for operations such as digital signal control. The ARM Cortex M4 is the least energy efficient of the M series of processors but is also the most powerful. The focus of the Adaptive ECG Filter at this point in time is to perform complex filtering operations therefore power consumption is not the primary focus and the Cortex M4 is the microprocessor for the system.

The ARM Cortex M4 uses RISC (Reduced Instruction Set Computing) Architecture. The RISC architecture works by using simple instructions that take only one clock to execute. Because the RISC architecture requires many simple instructions it takes more space it memory for the program it is running. The memory disadvantage of the RISC architecture is offset by the pipelining advantage it provides by performing one operation for each clock cycle.

The ARM Cortex M4 Also supports Embedded C++ which allowed an object oriented approach to the problem and the ability to change things quickly. In standard C a more procedural approach is required and as such that would have severely complicated the project.

XII. TESTING

The Testing of this project consisted of multiple different parts. The Analog to Digital converter was tested initially by providing a constant signal of .5V, 1V and 2V to ensure that the outputted data was consistent with the inputted signal.

The Least Mean Square Filter from CMSIS was tested by comparison with the MATLAB Least Mean Square Filter. The results of both filters were identical when given the same data.

UART was tested to send the data to MATLAB and initially had an error. The UARTPrintf command sent a new line after a line return when sending a character. This caused an invalid number of bytes to be sent. In an attempt to send 4 bytes for a float, 5 bytes were sent. This error was fixed by using a different UART command to send the data from the microprocessor. UART was used to send a lot of the data for the Least Mean Squares Filter as well as the Analog to Digital Converter. To help remedy this we created a UART library to call of the required function calls. The only required calls were sending floats and a character which held the data for the graph index 0 to 9.

The graphs in MATLAB were tested by sending a range of values to each of the graphs and ensuring that each of them changed correctly. Figure 10 demonstrates this test where the gradient and ECG pins, the first and second column, were provided with the same input. The input consisted on a 1V DC signal which then changes to a 2V DC signal. The first two columns reflects the change in values as expected. The third column is the Least Mean Square algorithm being applied given the two previous signals to the left of it. Since the signals are the same, it is expected and can be seen that the signal converges to 0.
This test demonstrates the working functionality of the ADC, filtering system, and MATLAB plotting.

The other three are the input signals which are the primary signal.

The values for the ECG will be tested by using actual values from an MRI machine as well as testing with artificial values at Philips. This testing has not been completed as of April 3, 2014 but a plan is in motion towards doing so before the presentation on April 17, 2014.

Another method of quality assurance that covered the span of our project is a zero-defect policy with regards to coding. Whenever an error occurred at any point in the code, bad values or an issue at any point all development was halted until said issue was fixed. This is due to the fact that it is much easier to fix an issue early on in the development process as opposed to later on in the project.

XIII. CONCLUSION

This project will hopefully help advance the proficiency and quality of Philips medical equipment with respect to the gathering of ECG signals. Hopefully at a later date in time; the use of Adaptive Filters will become commonplace when recording ECG signals. This project aim is to decrease the errors in the medical field and provide more quality to patients. MRI Scans are routinely done for a variety of reasons and having good results is rather important.

This project has shown an adaptive filter that is able to filter results from an MRI machine. The results are yet to be found with respect to this project and the validity of using the LMS with this configuration will be discussed on April 17, 2014. Unfortunately due to the current time schedule this project has not been completed yet.

The current status of the project includes the ability to process signals on the Development board from six inputs and display the filtered results. Three of the inputs are the noise signals that will need to be processed from the results.

XIV. THE ENGINEERS

Edward Jezisek – CpE
Lead Developer
Verizon Wireless - June 9, 2014
Specializations
Mobile Applications
TCP/UDP Networking
E-mail: ejezisek@yahoo.com
Phone: 520-461-4299

Brandon Autrey – CpE
Assisting Developer
Institute for Simulation and Training
Specializations:
Software Defined Networking
Parallelization
E-mail: bautrey@ieee.org
Phone: 954-829-7950

Renato Ortega - EE
Assisting Electrical Engineer
Siemens Intern
Specializations:
Energy
Solar Cells and Sensors
E-mail: Renato730@knights.ucf.edu
Phone: 561-768-6393

Edward Nowlin - EE
Lead Electrical Engineer
Naval Air Warfare Center - June 16, 2014
Specializations:
Wireless Communication
Navy Veteran
E-mail: ejnowlin@gmail.com
Phone: 407-625-1523

XV. ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Philips in regards to funding and the idea for this project. Philips has also provided valuable insight into the creation of our PCB board and giving advice with regards to patient monitoring of the MRI Environment.
Philips also gave us confidential parts of the circuit board for us to use and modify. This was due to having similar projects in the works.

The authors would also like to acknowledge the assistance provided by Dr. Richie with respect to this project and thanks to the following teachers for agreeing to review this project: Dr. Elena Flitsiyan, Dr. Robert Demara, Dr. Kalpathy Sundaram.

XVI. REFERENCES


