The Heart of the Matter

The University of Notre Dame Department of Electrical Engineering
Senior Design Final Report

Brendan Galloway
Ana Guzman
Allison Walker
Nicole Wardeberg

Table of Contents

1. Introduction

1.1 Problem and Solution

Many heart issues in patients go undiagnosed. Because of this, unrecognized poor cardiac health reveals itself in the forms of cardiac arrest, heart attacks, and other life-threatening symptoms before it can be treated. However, these events can be prevented. While an EKG (electrocardiogram) could likely identify these problems, most otherwise healthy people have no reason to undergo the rather involved diagnostic tests required. In contrast, the non-invasive recording and analysis of a pulse waveform, from anywhere on the body that blood passes through, could reveal abnormalities that prompt patients to seek further tests from physicians before the situation becomes an emergency. Photoplethysmography measures changes in blood volume by the absorption and reflection of red and infrared light in hemoglobin. When light is shined on human flesh, deoxygenated and oxygenated blood absorbs mostly red and infrared wavelengths, respectively. Any light not absorbed by the blood is reflected and captured by a photodiode. This signal is then used to generate a waveform reflecting changes in blood volume and concentration, which can be analyzed to detect concerning heart characteristics.

A doctor can extrapolate several heart characteristics from the PPG waveform that can indicate relative levels of heart health. It functions much like the more well-known EKG, but instead of using electrical pulses it uses volume as detected by changes in either reflected or transmitted light to glean information about how the heart is functioning. By analyzing a PPG waveform, a doctor can first determine heart rate, or the number of beats per minute. Secondly, a PPG waveform provides information about heart rate variability (HRV), the variability in the time interval between beats. Ideally, PPG waveforms from a person should not be perfectly periodic, showing that the heart is able to adjust to changes in the body and fluctuations in environment. A low HRV reading indicates higher risk of sudden cardiac death.

Additionally, by locating the dicrotic notch, systolic, and diastolic peaks of each pulse, the inflection point area can be calculated. This serves as an indicator of peripheral resistance. Similarly, the augmentation index reflects the elasticity of the artery. With the additional information of a patient's height, the PPG waveform can be analyzed to provide the arterial stiffness index. Taking the first and second derivatives of the PPG waveform can be utilized to locate parameters such as relative peaks and the dicrotic notch, which can make the analysis more robust for patients that do not have a clear dicrotic notch.

The project aims at creating a PPG instrument to observe detailed changes in the pulse wave using the properties of light, providing a doctor with a noninvasive way to gain insight into cardiac health. The transmitter of the device consists of two LEDs, one red (660 nm) and one infrared (940 nm), which are connected antiparallel to each other and driven by a timer module contained within the TI AFE4490, a high resolution analog-to-digital converter designed for pulse oximetry applications. The AFE4490 is controlled by a microcontroller. As each LED flashes, light illuminates hemoglobin at that particular wavelength. Photons, either reflected or transmitted through the skin, strike the receiver to generate an analog current, which is measured by the AFE. This analog current is converted to a 22-bit digital value representing the voltage of the signal. This digital value is sent back to the microcontroller by SPI communication. Within the microcontroller, these values are mapped to meaningful voltages, and then sent to the SD card for storage via a secondary SPI module.

On the software level, the microcontroller is programmed as the master with the AFE as the slave in the SPI communication system. By implementing successive SPI “write” commands, the AFE’s timer module is programmed by setting values for dedicated registers, which govern the duration of functional time blocks. It is during these time blocks that the LEDs are flashed at a specified frequency, the light incident upon the receiver is sampled, and the analog current values from the receiver are converted to corresponding digital voltages. Once these timers are set by write instructions from the microcontroller to the AFE, a “read” command is sent and the cycle of flashing, sampling, and converting repeats continuously, sending the voltage values back to the microcontroller, which are ultimately sent to the SD card as noted above. Finally, a MATLAB program analyzes the data sent from the microSD card and outputs the PPG waveform and other useful heart-health data onto a user-friendly GUI.

1.2 Expectations vs. Reality

The initial intention of the project was to analyze a PPG waveform specifically for cardiomyopathy indicators. Cardiomyopathy, enlarged heart syndrome, causes the walls of the heart to thicken to an abnormal degree. While symptoms of the disease do not exist for many carriers of cardiomyopathy, this obstruction often presents itself as heart arrhythmias. The project was designed use a PPG waveform to accurately catch the “twice beating” heart arrhythmias that might exist among patients with the difficult-to-diagnose cardiomyopathy.

However, as development of the project progressed and research ensued, the team found that the PPG waveform can present a myriad of cardiac health indicators and characteristics. Because of the usefulness of the PPG waveform, the project became more broad than simply analyzing the waveform data for cardiomyopathy symptoms. Instead, the data obtained from the system is used to analyze heart rate, heart rate variability, inflection point area, augmentation index, and arterial stiffness index. These parameters are all indicators of cardiac health.

The project successfully measures a person’s blood concentration to output a PPG waveform. However, the system is hard coded with a set number of samples for the sensor to detect. In future improvements, the device would be able to continuously measure data without having to change the number of samples collected. Overall, the project was successfully completed. The device is able to extract real data from a person, the subsystems communicate, and the MATLAB program plots and analyzes the PPG waveform.

2. Detailed System Requirements

2.1 Overall System Block Diagram

The functionality of the system as a whole depends on the communication between several subsystems. A block diagram of the overall system is described below. The PIC32MX270F256D microcontroller uses SPI communication to control the TI AFE4490 high resolution analog-to-digital converter. Used for pulse oximetry purposes, the TI AFE4490 device sets the flashing rate of the red and infrared LEDs. The red and infrared LEDs are connected in antiparallel.

The flashing red and infrared LEDs are positioned on a board 7 millimeters next to the photodiode. As the LEDs blink, oxygenated and deoxygenated blood in the human body absorbs mostly infrared and red light, respectively. The photodiode captures the light that is reflected back (i.e. not absorbed by the blood) and sends the analog current signal back to the TI AFE4490, which converts the analog signal to a digital signal. This digital signal is sent back to the microcontroller, which sends the data to the microSD card using SPI communication. Once the microSD card captures the data, it is ejected from the hardware and inserted into an external computer for data processing and analysis. The MATLAB program outputs useful data onto a user-friendly GUI. Using the PPG waveform, the program calculates heart rate, heart rate variability, inflection point area, augmentation index, and arterial stiffness index.

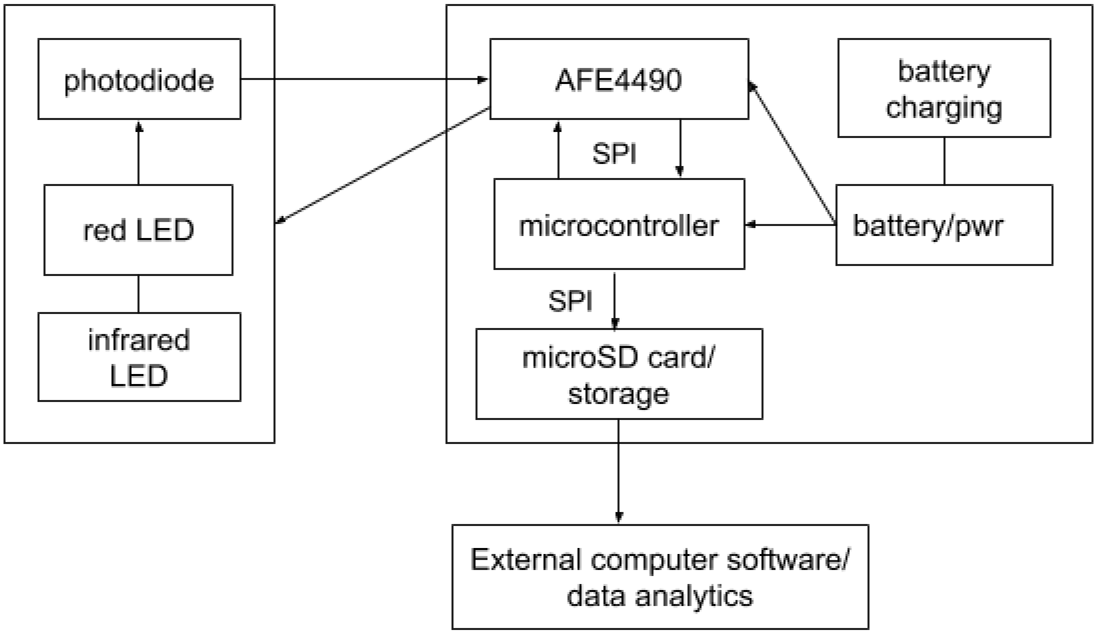


Figure 1. Overall System Block Diagram

2.2 Subsystem Block Diagram

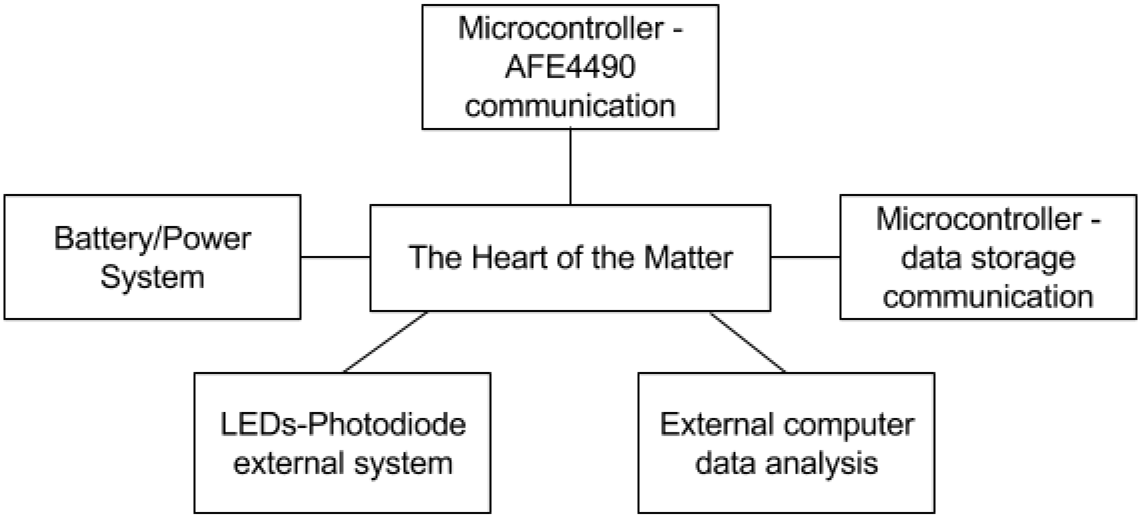


Figure 2. Subsystem Block Diagram

2.3 Hardware Component Requirements Descriptions

Microcontroller
The PIC32MX270F256D operates at 2.3-3.6 V and has 5 V tolerant pins. The microcontroller runs off of a 50 MHz clock. A 10 mA source/sink current can run through all the I/O pins. Additionally, the PIC32MX270F256D has two SPI modules, which allows for SPI communication with both the AFE4490 and the microSD card. This microcontroller has 44 pins, which is enough pins to make connections with the AFE4490, microSD card, PICKIT3 programmer, and power system.

AFE4490
The AFE4490 is an integrated analog front-end for pulse oximeters. It consists of a low-noise receiver channel with a 22-bit analog-to-digital converter, an LED transmit section, and diagnostics for LED and photodiode fault detection. Thus, the AFE4490 provides a suitable functionality for medical pulse oximetry applications. The AFE4490 communicates with an external microcontroller via SPI communication and is able to do so with the PIC32MX270F256D. The LED drive current is programmable to 50, 75, 100, 150, and 200 mA. The continuous input current to any pin is 7 mA. The device is small at 6.00 mm x 6.00 mm. The Rx analog supply pin requires a supply voltage of 2.0 V to 3.6 V. The transmit control supply pin (Tx) requires a supply voltage of 3.0 V to 5.25 V.

microSD card
The microSD card operates between 2.7 and 3.6 V. The voltage supply needs to be kept below 5 V so that the card is not damaged. The storage capacity of the SD card is 16GB. The microSD card size is 11 mm x 15 mm. The microSD card communicates with the microcontroller using SPI protocol, which is possible with the PIC32MX270F256D's second SPI module.

LEDs/Photodiode system

The LEDs/Photodiode system requires a red LED and infrared LED in anti-parallel. These specific wavelengths of about 660 nm (red) and 940 nm (infrared) are ideal because deoxygenated hemoglobin absorbs mostly red light and oxygenated hemoglobin absorbs mostly infrared light. The reflected or transmitted light is the signal used for the pulse oximetry analysis. In order to capture the signal to use as data, a photodiode captures the light and converts it to current. Vishay's VSMD66694 dual color emitting diode has a red and infrared LED package, emitting at 660 nm and 940 nm respectively, in anti-parallel. The maximum forward currents of both the red and infrared LEDs are 70 mA, but they typically operate at 20 mA. The package is small at 2 mm x 2 mm. The TEMD5080X01 is a silicon PIN photodiode suitable for visible and infrared radiation, which is necessary for the capturing of the reflected or transmitted light from the dual color emitting diode. The package is small at 5.00 mm x 4.24 mm.

Power system

Based on the voltage requirements of the subsystems above, the power system outputs a VDD of 3.3 V to the entire board. To do this, a Li-Ion battery of about 3.7-4 V is regulated by the TPS61201, a low input voltage synchronous boost converter which takes the battery as its input and outputs a constant 3.3 voltage.

3. Detailed Project Description

3.1 System Theory of Operation

From a high level view, the heart monitor system operates by the blood's absorption of light. The microcontroller uses SPI communication to control the AFE4490 digital-to-analog converter. After the microcontroller communicates with the AFE4490, the AFE4490 controls the flashing rate of the red and infrared LEDs, which are connected in antiparallel. The red and infrared light shines on the human body, and oxygenated hemoglobin absorbs mostly infrared light while deoxygenated hemoglobin absorbs mostly red light. Light not absorbed by the blood is then reflected back into a photodiode, which converts the signal to an analog current and sent to the AFE4490 digital-to-analog converter. The AFE4490 converts the signal to digital and sends it to the microcontroller using SPI communication. The microcontroller then sends the data to the microSD card using SPI communication. Finally, the microSD card is physically extracted from the board system and inserted into an external computer. A MATLAB program then analyzes the data to output the PPG waveform and calculates heart rate, heart rate variability, inflection point area, augmentation index, and arterial stiffness index.

3.2 System Block Diagram

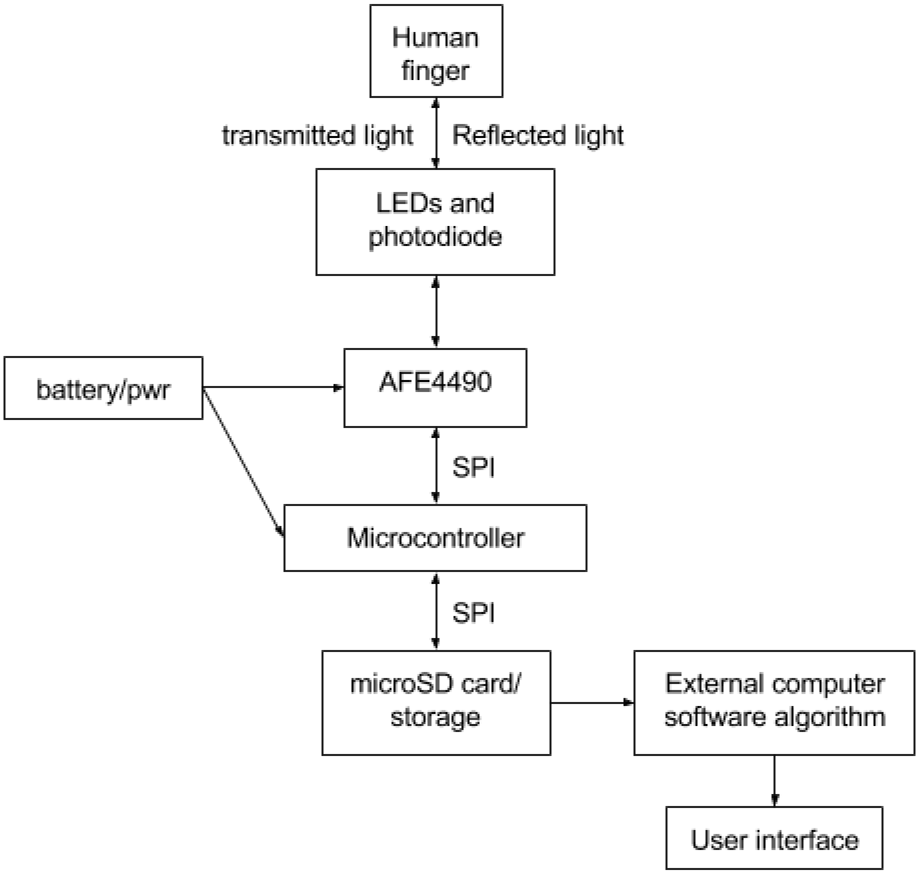


Figure 3. System Block Diagram

3.3 Detailed Design/Operation of LEDs and Photodiode Hardware

Flow Chart

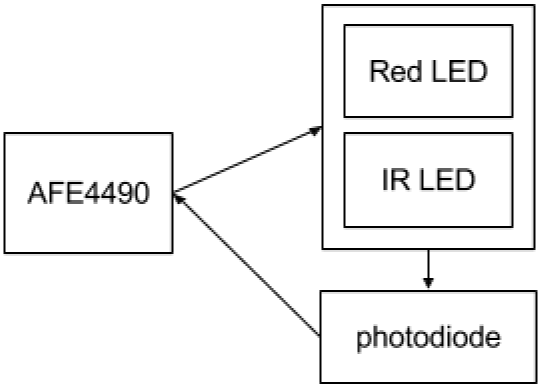


Figure 4. LEDs and Photodiode Flow Chart

Schematic

The dual light emitting diodes, red and infrared, and the photodiode are placed on a small board 7 millimeters apart. The TI AFE4490 digital-to-analog converter sends the flash rate to the red and infrared LEDs. The red and infrared LEDs are connected to the AFE4490 at pins 14 and 15. The red and infrared LEDs send light to the human body (the finger), and the deoxygenated and oxygenated blood absorbs the red and infrared wavelengths, respectively. Light is reflected from the body and then captured by the photodiode, which is coupled with a 0.5 uF capacitor for noise control. The photodiode captures the reflected light and converts it into current. This signal is then sent back to the AFE4490 digital-to-analog converter. The anode of the photodiode is connected to the AFE4490 at pin 1 and the cathode of the photodiode is connected to the AFE4490 at pin 2.

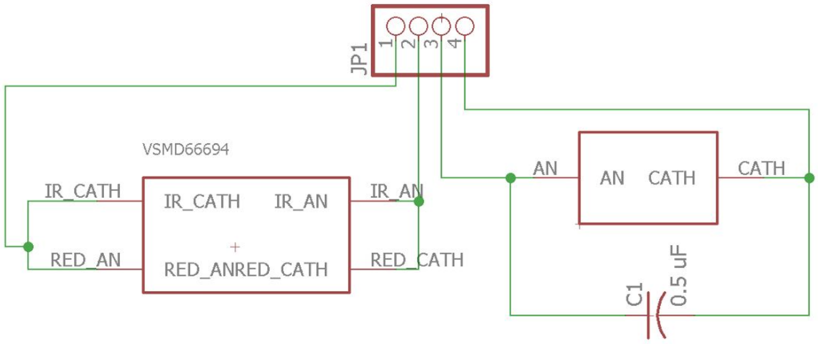


Figure 5. LED and Photodiode Schematic

Function and Connection to Other Subsystems

The external hardware consists of two LEDs, red and infrared, and a photodiode. The LED part used has the part number VSMD66694. These dual color emitting diodes, red and infrared, are connected in antiparallel, as shown below from the Vishay Semiconductors datasheet. The VSMD66694 contains both LEDs in the package.

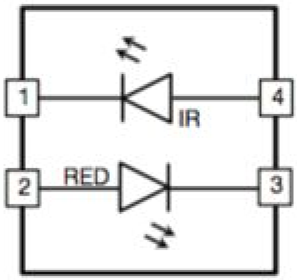


Figure 6. Dual Emitting Diode Block Diagram

The VSMD66694 part provides a strong enough signal for the photodiode to catch the light from both reflectance and transmittance. Additionally, the package includes both LEDs in one package, which saves space when constructing the wearable. The IR light emits at 990 nm and the red light shines at 660 nm, which correspond to the wavelengths needed for deoxygenated and oxygenated blood to absorb. Finally, the VSMD66694 is a surface mount part, making it easy to use solder paste on a small board.

The photodiode, coupled with a 0.5 uF capacitor for noise control, exists separately from the dual light emitting diodes. The photodiode is placed 7 millimeters from the LED package on a small board, capturing the reflected light. The photodiode part number is TEMD5080X01. This silicon PIN photodiode is sensitive from 350 to 1100 nm, which enables a single photodiode to capture both the red and infrared wavelengths of the LEDs. The surface mount package allows for relatively easy soldering on the small board.

Externally, the LED and photodiode system is connected directly to the AFE4490. The AFE4490 controls the flashing rates of the red and infrared LEDs. The blood absorbs some light, and some light reflects back to the photodiode. The photodiode then captures the signal and sends a current to the AFE4490, which then converts the signal from analog to digital for the microcontroller to process. The analog LED driver, pin 14, controls the red LED, while the analog LED driver, pin 15, controls the infrared LED. Finally, the photodiode anode and cathode sends its signal to the receiver input pins 1 and 2, respectively.

Subsystem Testing

An evaluation board mounted with the TI AFE4490 was used for familiarization and testing. This incorporated the dual red/infrared LED and photodiode for accurate collection of data of the pulse waveform. The dual red/infrared LED and photodiode are connected in antiparallel orientation. For the initial design review, small wires connected to the dual red/IR LED surface mount part VSMD66694. The anode of the infrared LED (cathode of the red LED) and cathode of the infrared LED (anode of the red LED) were wired with two wires to a DB9 connector, which fed into the evaluation board. On a separate board, the photodiode, coupled with a 5 μF capacitor, was also connected to the DB9, which fed into the evaluation board. The following diagrams from the AFE4490 datasheet show the dual LED and photodiode systems:

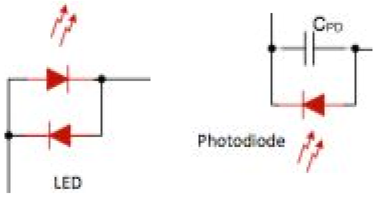


Figure 7. AFE4490 Datasheet LEDs and Photodiode

The evaluation board came with software enabling the manipulation of certain operating conditions and monitoring the data being collected in live time. The TI setup included an oximetry clip commonly found in hospitals for a patient's fingers. The AFE4490 controls the flashing rate of the LEDs. Light shines through the finger, and the finger absorbs some of the red and infrared light. Using reflection geometry, the photodiode, on the same side of the finger as the LEDs, captures the uninterrupted photons, which generate a current to be read as a voltage by the AFE4490. This signal is ultimately passed through the microcontroller to the microSD card. These values can be observed on the computer as a high resolution pulse waveform in MATLAB. After testing several current values, setting the current at 10 mA provides a clear pulse signal. Driving the current too high leads to saturation, and readable pulse waveforms fail to output on the computer.

3.4 Detailed Design/Operation of MATLAB Data Processing

System Flow Chart

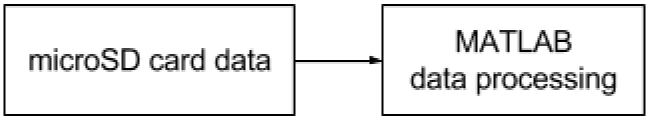


Figure 8. Data Processing Flow Chart

Function and Connection to Other Subsystems

The recorded data is exported to a text file on the SD card. The SD card is plugged into a computer and the file is exported to an Excel document which is then read into MATLAB for data processing. An example of the raw data produced by the system, in terms of voltages across the photodiode, is shown below:

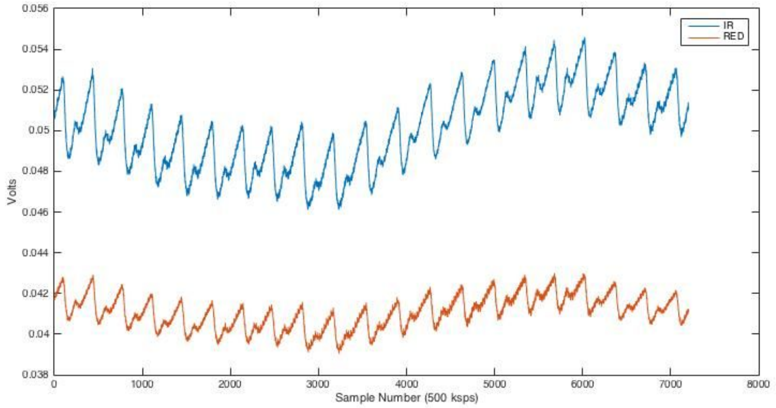


Figure 9. Pulse waveforms generated from photodiode and LEDs

The first step in processing this data requires changing these voltages to relative concentrations of deoxygenated and oxygenated hemoglobin utilizing equations 1 and 2, shown below. They take into consideration the molar extinction coefficient of blood at both wavelengths to determine how a change in light attenuation is related to change in concentration of HbO_2 and Hb.

Equations 1 and 2[2]

$$\Delta c_{\mathrm{HbO_2}} = \frac{\alpha_{\mathrm{Hb}}^{\lambda_1}\frac{\Delta A^{\lambda_2}}{L^{\lambda_2}} - \alpha_{\mathrm{Hb}}^{\lambda_2}\frac{\Delta A^{\lambda_1}}{L^{\lambda_1}}}{\alpha_{\mathrm{Hb}}^{\lambda_1}\alpha_{\mathrm{HbO_2}}^{\lambda_2} - \alpha_{\mathrm{Hb}}^{\lambda_2}\alpha_{\mathrm{HbO_2}}^{\lambda_1}},$$

$$\Delta c_{\mathrm{Hb}} = \frac{\alpha_{\mathrm{HbO_2}}^{\lambda_1}\frac{\Delta A^{\lambda_2}}{L^{\lambda_2}} - \alpha_{\mathrm{HbO_2}}^{\lambda_2}\frac{\Delta A^{\lambda_1}}{L^{\lambda_1}}}{\alpha_{\mathrm{HbO_2}}^{\lambda_1}\alpha_{\mathrm{Hb}}^{\lambda_2} - \alpha_{\mathrm{HbO_2}}^{\lambda_2}\alpha_{\mathrm{Hb}}^{\lambda_1}}.$$

This generates waveforms that more closely resemble the expected Blood Pulse Volume (BPV) waveform. Noise is reduced by MATLAB's smooth function with the filter type 'loess'. An example of a smoothed HbO_2 concentration waveform is displayed in the figure below:

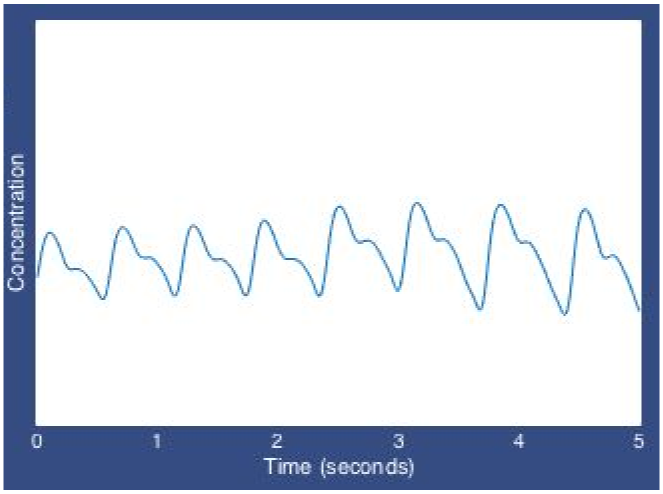


Figure 10. Smoothed HbO_2 concentration waveform

The next step separates each pulse using the *findpeaks* function and solves for parameters such as the pulse wave duration, pulse wave amplitude, pulse propagation time, and time differences between the systolic and diastolic phases. These parameters are averaged over a given time period to create an "average pulse" from which indicators of cardiac health are examined. The following figure describes the BVP waveform:

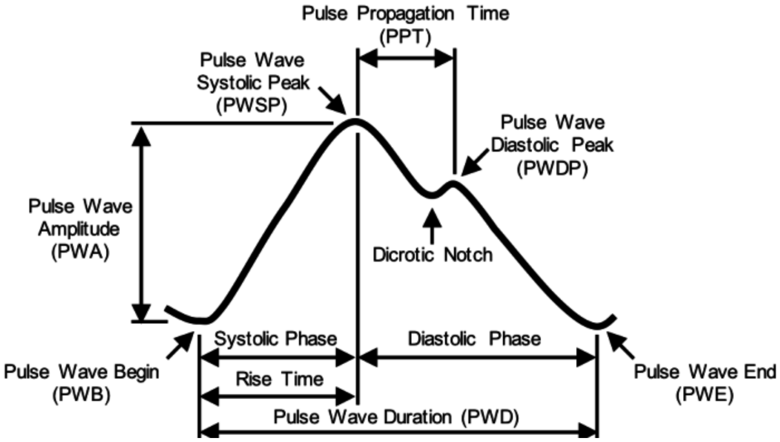


Figure 11. Blood Volume Pulse (BVP) waveform and parameters[1]

After heart rate, one of the first indicators to look at is the heart rate variability (HRV). This is the standard deviation of the Pulse Wave Duration of 5 minutes to 24 hours worth of pulse data. It measures how well the heart responds to the autonomic nervous system. High values indicate a healthy response. Another indicator of heart health is the Augmentation Index, which is a ratio of the PWSP and PWDP as labeled on the figure above. This indicates arterial stiffness. The inflection point area ratio is calculated. This is the ratio of the integral of the signal before the dicrotic notch divided by the integral of the signal after the dicrotic notch. It is an indicator of total peripheral resistance. The stiffness index (SI) is a ratio of the pulse propagation time and the height of a patient. The index is calculated in meters/second and is a reflection of the stiffness of major arteries. [3]

Interacting with GUI

The MATLAB GUI functions to make the raw data more accessible and useful to the doctor or patient reviewing the results. It does not require significant knowledge of programming to operate.

An important note is that the current GUI is optimized to accept inputs from the Evaluation Board. To make sure the GUI is functioning properly in this mode, ensure that the "Utilizing Evaluation Board" checkbox is appropriately marked at the bottom of the yellow "User Inputs" box. When this box is unchecked, the GUI accepts inputs from our custom board but the scaling is not set for the calculations section to be reliable.

Software Flow Chart

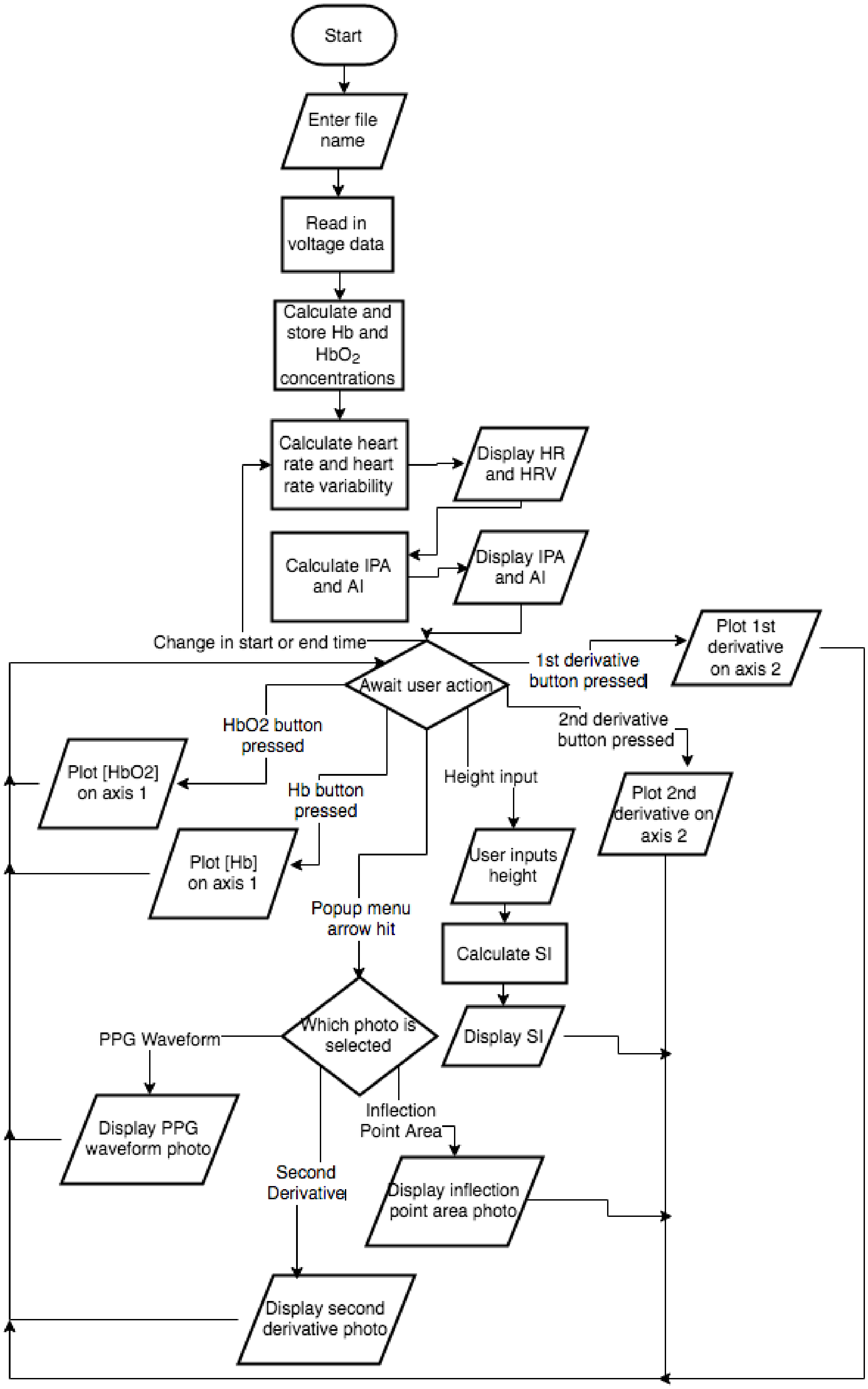


Figure 12. MATLAB GUI Flow Chart

Explanation of the GUI

When the GUI is opened, the user must first input the file name. As a check for scaling accuracy as the GUI is being developed, it first plots the IR waveform with marks at the top of each detected peak. An example of this is shown in the figure below:

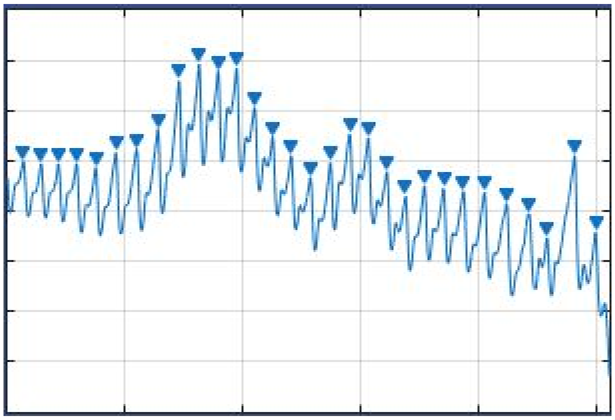
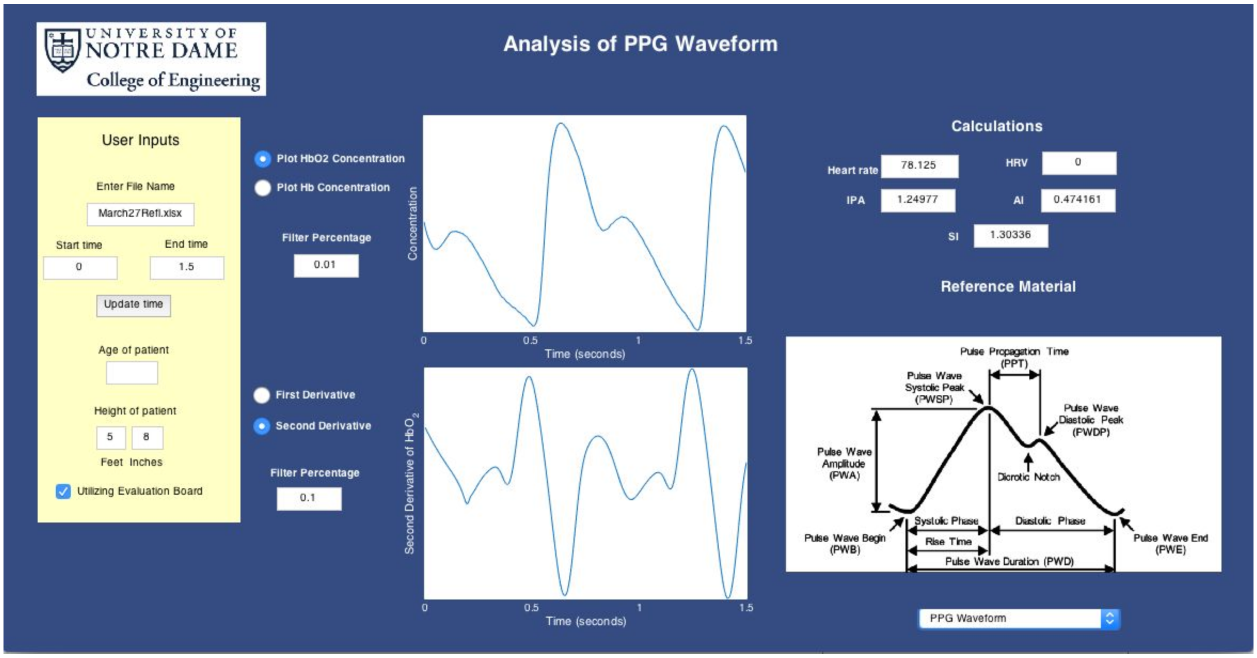


Figure 13. Example of image to check for findpeaks accuracy

Next, the user enters the height of the patient. They will then select the desired plots for each graph and the desired reference image to be displayed, if any.



Analysis of PPG Waveform

User Inputs

Enter File Name: March27Refl.xlsx

Start time: 0

End time: 1.5

Age of patient:

Height of patient: 5 Feet 8 Inches

☑ Utilizing Evaluation Board

● Plot HbO2 Concentration

○ Plot Hb Concentration

Filter Percentage: 0.01

○ First Derivative

● Second Derivative

Filter Percentage: 0.1

Calculations

Heart rate: 78.125

HRV: 0

IPA: 1.24977

AI: 0.474161

SI: 1.30336

Reference Material

PPG Waveform

Figure 14. Screenshot of operating MATLAB GUI

The x-axis of the plotted data can be changed by entering a new start and stop time, in seconds, and clicking the “Update time” button. The graphs will adjust accordingly, as will the values of the heart rate and heart rate variability (HRV) calculations. If the user would like more or less filtering of either plot, they can adjust this with the Filter Percentage editable text boxes.

MATLAB Processing Testing

Due to the time limitations of the project, there was not comprehensive testing of the data processing for the data from our custom board. This is hoped to be continued into the next variation of this project.

There was substantial testing of the data collected from the evaluation board. There was significant trial-and-error involved in filtering the data enough that the findpeaks function would not find false peaks while also ensuring that the necessary information was not removed. Ultimately, the "loess" smooth function was deemed more reliable than resampling the signal and gave the added advantage of not changing the size of the data array.

3.5 Detailed Design/Operation of the Microcontroller Communication with the AFE4490

System Flow Chart

Figure 15. Microcontroller to AFE4490 Flow Chart

Software Flow Chart

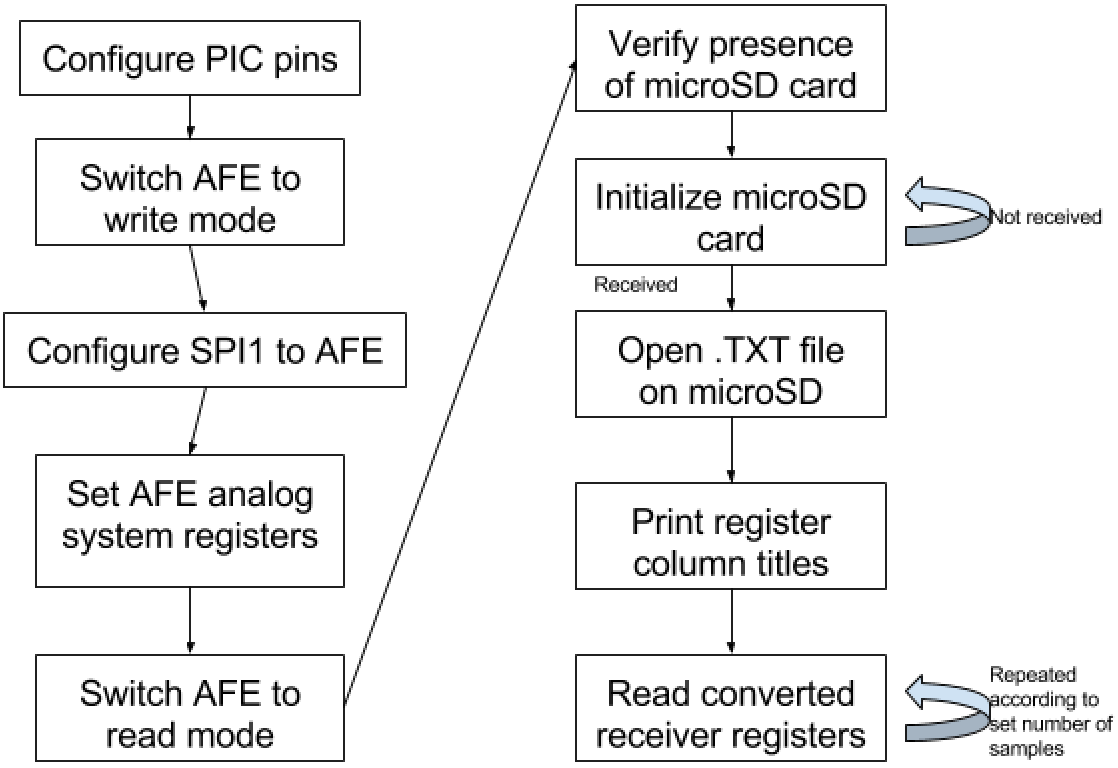


Figure 16. Microcontroller to AFE4490 Software Flow Chart

Schematic

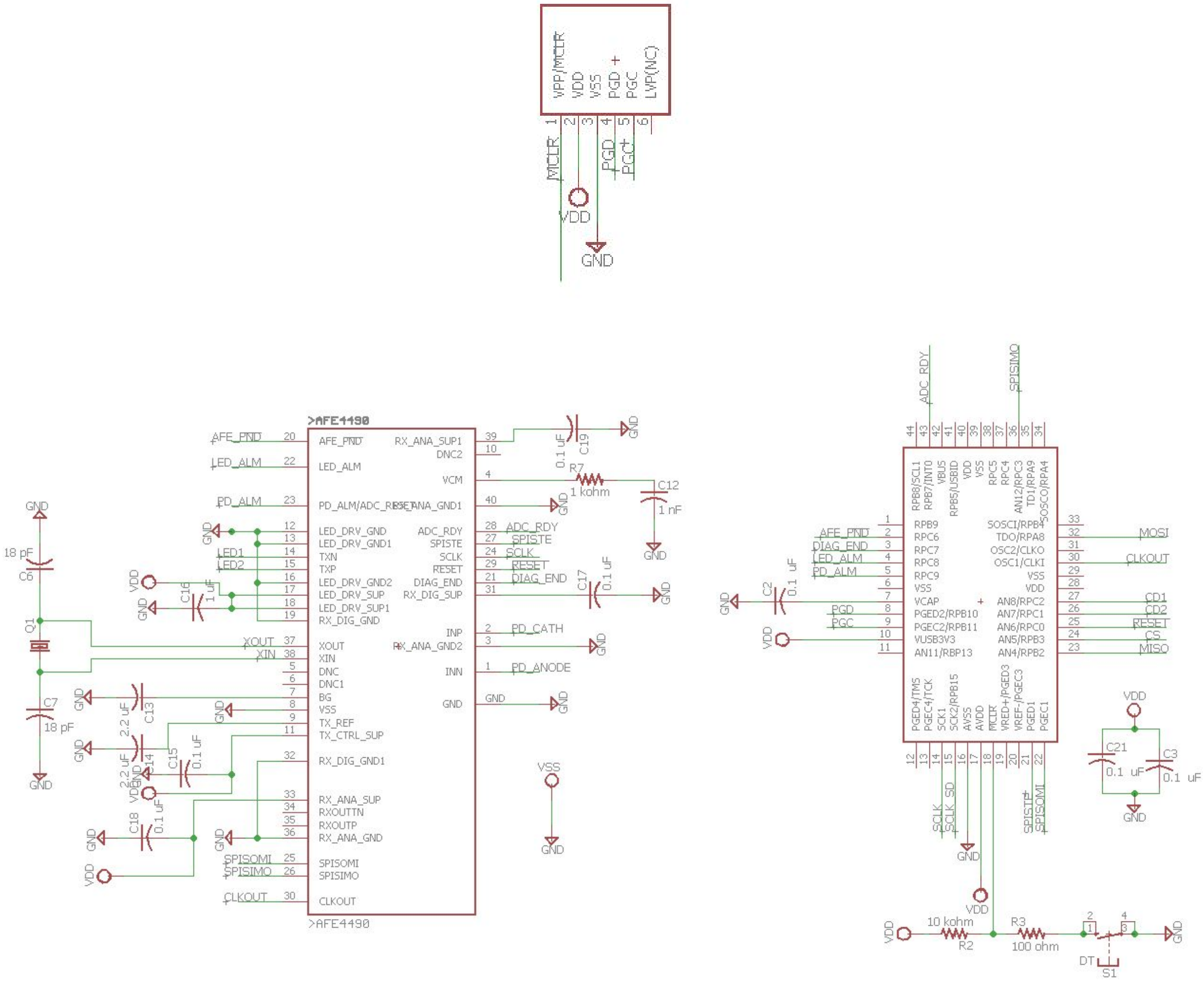


Figure 17. PICKIT to Microcontroller to AFE4490 Schematic

Function and Connection to Other Subsystems (Hardware)

The PIC32MX270F256D microcontroller is the master to the AFE4490 digital-to-analog converter. The microcontroller dictates the clock rate of the AFE4490, which then controls the LED flashing rate. Once the microcontroller (the master) communicates with the AFE4490 (the slave), the LEDs and photodiode subsystem send a signal back to the AFE4490, which converts the signal to analog and sends it to the microcontroller.

The specific pin connections are shown on the schematic above. First, the SPI connections between the microcontroller and AFE were designated to remappable pins. The SPI serial interface enable, pin 27 on the AFE4490, connects to the remappable pin 21 on the microcontroller. The microcontroller sends a clock signal to the AFE via SPI communication. This SCLK is pin 24 on the AFE and is connected to the SCLK pin 14 on the microcontroller. On the AFE, the SPI serial in master out (SPISIMO) receives signals from the microcontroller. These connections are made via pin 26 on the AFE and the remappable pin 36 on the microcontroller. The SPI serial out master in (SPISOMI) sends the data received from the LEDs and photodiode subsystem to the microcontroller. These connections are made via pin 25 on the AFE and the remappable pin 22 on the microcontroller.

The clock out (CLKOUT) signal from the AFE runs at 4 MHz. This signal from pin 30 of the AFE connects to the clock input (CLKI) pin 30 on the microcontroller. The RESET input of the AFE connects to port pin 25 of the microcontroller. This reset of the AFE is active low. The AFE power down input (AFE_PND), set as active low, at pin 20, connects to port pin 2 of the microcontroller. The ADC_RDY pin 28 of the AFE is an output signal that indicates if the analog-to-digital conversion is complete. The output is the input to the interrupt input pin 43 of the microcontroller. The DIAG_END pin 21 of the AFE is an output signal that indicates completion to diagnostics and connects to port pin 3 of the microcontroller. The AFE4490 have output signals that indicate photodiode and LED faults, PD_ALM and LED_ALM. PD_ALM (pin 23) and LED_ALM (pin 22) are connected to microcontroller port pins 5 and 4, respectively.

The microchip PICKIT3 programmer is connected to the microcontroller to program the device. Output pin 1 of the PICKIT3, !MCLR, is connected to the !MCLR pin 18 of the microcontroller. PGD and PGC, which ultimately program the microcontroller, of the PICKIT3 send signals to pins 8 and 9, respectively, of the microcontroller.

The block diagram for the inner workings of the AFE4490 is shown below:

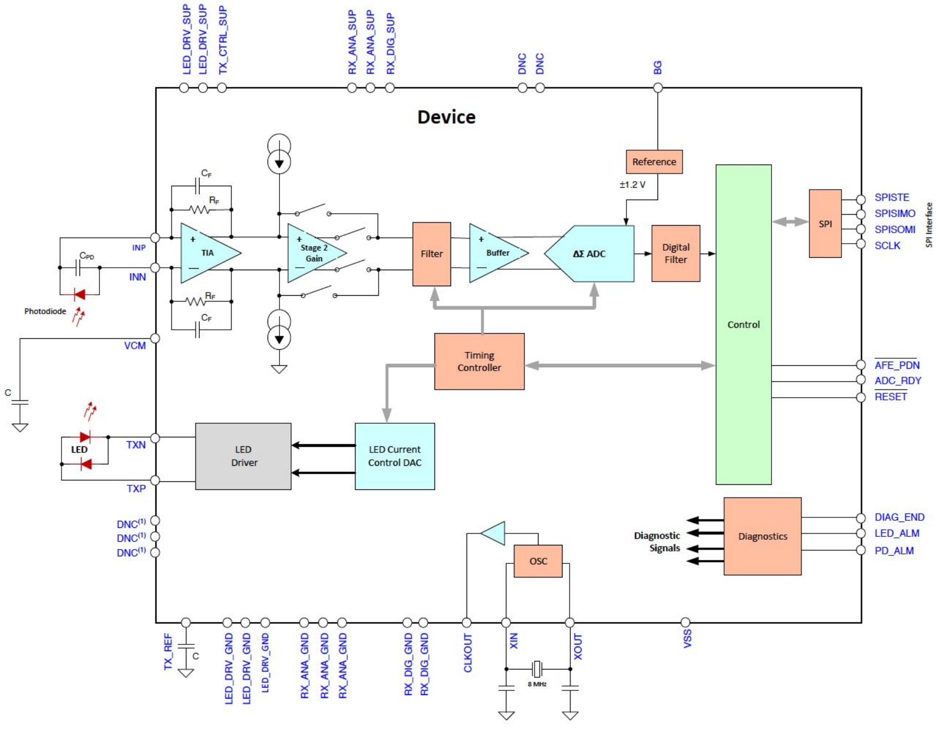
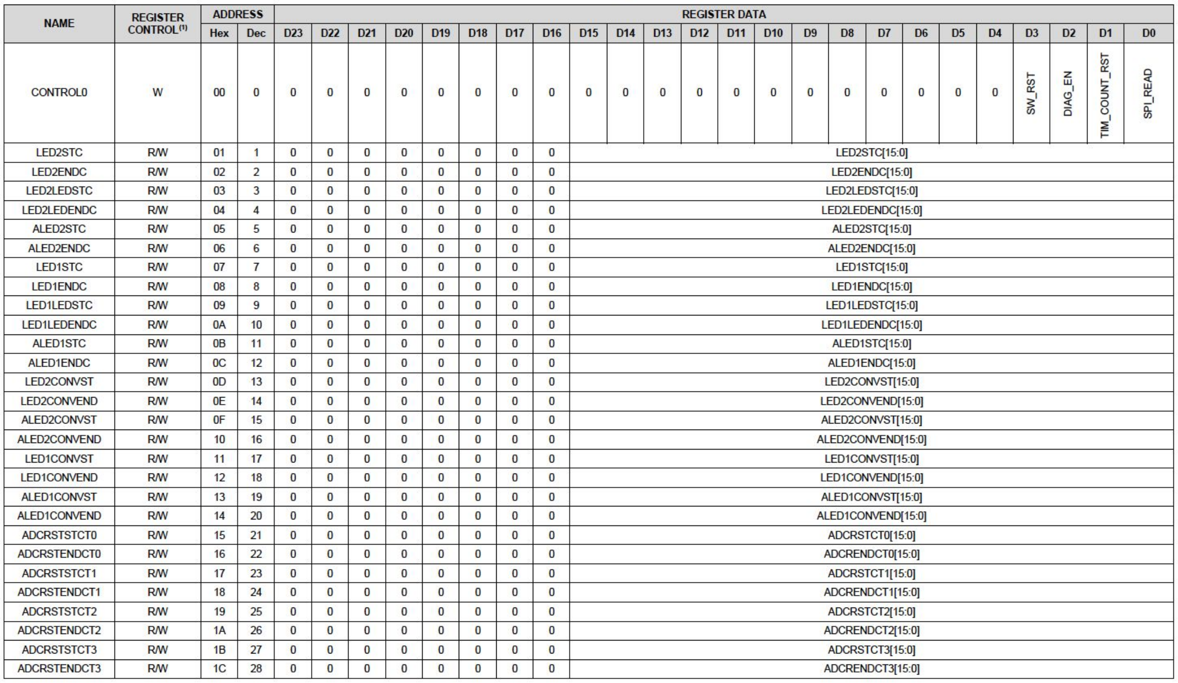


Figure 18. AFE Internal Block Diagram

Function and Connection to Other Subsystems (Software)

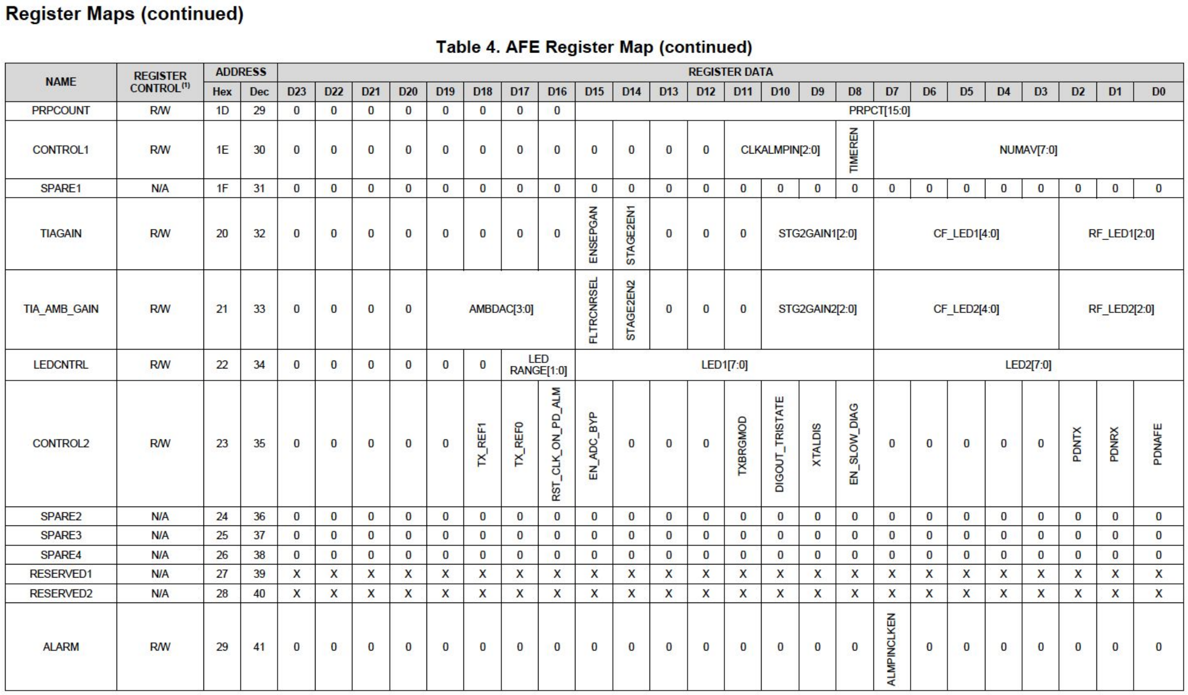
The digital subsystems of the project communicate via serial peripheral interface (SPI) bus. The microcontroller and the AFE exchange data over master-out-serial-in (MOSI) and master-in-serial-out (MISO) connections with the PIC as the master and the AFE as the slave. There is an 8 MHz clock signal input to the AFE from a crystal oscillator. The AFE then drops the clock rate to its operating frequency of 4MHz. This clock signal is then routed into the microcontroller so that the two components can run on a synchronized clock; this greatly reduces communication errors often caused by misaligned clock signals, especially when operating near top capacity. An initial series of write instructions sets up the extensive timing module on the AFE which controls the LED flashing, photodiode sampling sequence, and analog-to-digital conversion, as detailed in the AFE4490 datasheet (see Appendix A). Also in this series of write instructions are the specifications for the gain stages, control registers (delineating various operational modes and reference voltages), and the current values for the LEDs to manipulate brightness.

Table 4. AFE Register Map



NAME	REGISTER CONTROL[1]	ADDRESS		REGISTER DATA																							
		Hex	Dec	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CONTROL0	W	00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SW_RST	DIAG_EN	TIM_COUNT_RST	SPI_READ
LED2STC	R/W	01	1	0	0	0	0	0	0	0	0	LED2STC[15:0]															
LED2ENDC	R/W	02	2	0	0	0	0	0	0	0	0	LED2ENDC[15:0]															
LED2LEDSTC	R/W	03	3	0	0	0	0	0	0	0	0	LED2LEDSTC[15:0]															
LED2LEDENDC	R/W	04	4	0	0	0	0	0	0	0	0	LED2LEDENDC[15:0]															
ALED2STC	R/W	05	5	0	0	0	0	0	0	0	0	ALED2STC[15:0]															
ALED2ENDC	R/W	06	6	0	0	0	0	0	0	0	0	ALED2ENDC[15:0]															
LED1STC	R/W	07	7	0	0	0	0	0	0	0	0	LED1STC[15:0]															
LED1ENDC	R/W	08	8	0	0	0	0	0	0	0	0	LED1ENDC[15:0]															
LED1LEDSTC	R/W	09	9	0	0	0	0	0	0	0	0	LED1LEDSTC[15:0]															
LED1LEDENDC	R/W	0A	10	0	0	0	0	0	0	0	0	LED1LEDENDC[15:0]															
ALED1STC	R/W	0B	11	0	0	0	0	0	0	0	0	ALED1STC[15:0]															
ALED1ENDC	R/W	0C	12	0	0	0	0	0	0	0	0	ALED1ENDC[15:0]															
LED2CONVST	R/W	0D	13	0	0	0	0	0	0	0	0	LED2CONVST[15:0]															
LED2CONVEND	R/W	0E	14	0	0	0	0	0	0	0	0	LED2CONVEND[15:0]															
ALED2CONVST	R/W	0F	15	0	0	0	0	0	0	0	0	ALED2CONVST[15:0]															
ALED2CONVEND	R/W	10	16	0	0	0	0	0	0	0	0	ALED2CONVEND[15:0]															
LED1CONVST	R/W	11	17	0	0	0	0	0	0	0	0	LED1CONVST[15:0]															
LED1CONVEND	R/W	12	18	0	0	0	0	0	0	0	0	LED1CONVEND[15:0]															
ALED1CONVST	R/W	13	19	0	0	0	0	0	0	0	0	ALED1CONVST[15:0]															
ALED1CONVEND	R/W	14	20	0	0	0	0	0	0	0	0	ALED1CONVEND[15:0]															
ADCRSTSTCT0	R/W	15	21	0	0	0	0	0	0	0	0	ADCRSTCT0[15:0]															
ADCRSTENDCT0	R/W	16	22	0	0	0	0	0	0	0	0	ADCRENDCT0[15:0]															
ADCRSTSTCT1	R/W	17	23	0	0	0	0	0	0	0	0	ADCRSTCT1[15:0]															
ADCRSTENDCT1	R/W	18	24	0	0	0	0	0	0	0	0	ADCRENDCT1[15:0]															
ADCRSTSTCT2	R/W	19	25	0	0	0	0	0	0	0	0	ADCRSTCT2[15:0]															
ADCRSTENDCT2	R/W	1A	26	0	0	0	0	0	0	0	0	ADCRENDCT2[15:0]															
ADCRSTSTCT3	R/W	1B	27	0	0	0	0	0	0	0	0	ADCRSTCT3[15:0]															
ADCRSTENDCT3	R/W	1C	28	0	0	0	0	0	0	0	0	ADCRENDCT3[15:0]															

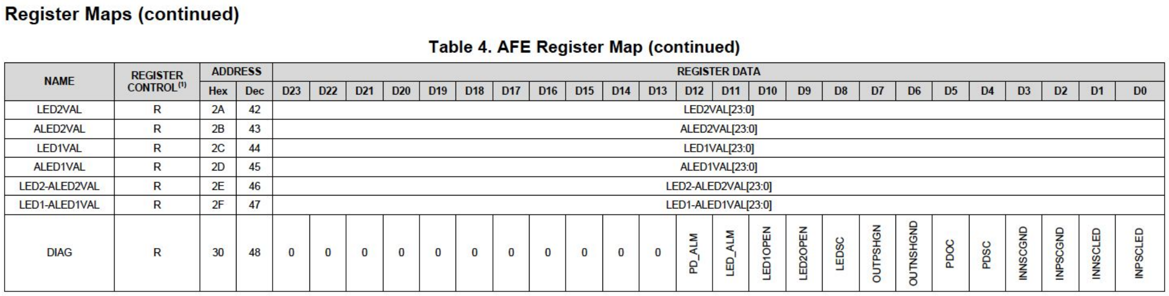
(1) R = read only, R/W = read or write, N/A = not available, and W = write only.



Register Maps (continued)

Table 4. AFE Register Map (continued)

NAME	REGISTER CONTROL(1)	ADDRESS		REGISTER DATA																							
		Hex	Dec	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
PRPCOUNT	R/W	1D	29	0	0	0	0	0	0	0	0	PRPCT[15:0]															
CONTROL1	R/W	1E	30	0	0	0	0	0	0	0	0	0	0	0	0	CLKALMPIN[2:0]			TIMEREN	NUMAV[7:0]							
SPARE1	N/A	1F	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TIAGAIN	R/W	20	32	0	0	0	0	0	0	0	0	ENSEPGAN	STAGE2EN1	0	0	0	STG2GAIN1[2:0]			CF_LED1[4:0]					RF_LED1[2:0]		
TIA_AMB_GAIN	R/W	21	33	0	0	0	0	AMBDAC[3:0]				FLTRCNRSEL	STAGE2EN2	0	0	0	STG2GAIN2[2:0]			CF_LED2[4:0]					RF_LED2[2:0]		
LEDCNTRL	R/W	22	34	0	0	0	0	0	0	LED RANGE[1:0]		LED1[7:0]								LED2[7:0]							
CONTROL2	R/W	23	35	0	0	0	0	0	TX_REF1	TX_REF0	RST_CLK_ON_PD_ALM	EN_ADC_BYP	0	0	0	TXBRGMOD	DIGOUT_TRISTATE	XTALDIS	EN_SLOW_DIAG	0	0	0	0	0	PDNTX	PDNRX	PDNAFE
SPARE2	N/A	24	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SPARE3	N/A	25	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SPARE4	N/A	26	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESERVED1	N/A	27	39	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RESERVED2	N/A	28	40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ALARM	R/W	29	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ALMPINCLKEN	0	0	0	0	0	0	0



Register Maps (continued)

Table 4. AFE Register Map (continued)

NAME	REGISTER CONTROL(1)	ADDRESS		REGISTER DATA																							
		Hex	Dec	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
LED2VAL	R	2A	42	LED2VAL[23:0]																							
ALED2VAL	R	2B	43	ALED2VAL[23:0]																							
LED1VAL	R	2C	44	LED1VAL[23:0]																							
ALED1VAL	R	2D	45	ALED1VAL[23:0]																							
LED2-ALED2VAL	R	2E	46	LED2-ALED2VAL[23:0]																							
LED1-ALED1VAL	R	2F	47	LED1-ALED1VAL[23:0]																							
DIAG	R	30	48	0	0	0	0	0	0	0	0	0	0	0	PD_ALM	LED_ALM	LED1OPEN	LED2OPEN	LEDSC	OUTPSHGN	OUTNSHGND	PDOC	PDSC	INNSCGND	INPSCGND	INNSCLED	INPSCLED

Figure 19. AFE Register Maps

After all of the registers above are initialized, the final bit of the Control 0 register is switched from 0 to 1 in order to put the AFE into a continuous read mode during which it outputs data to the microcontroller. The instruction and data flow for the SPI communication between the microcontroller and the AFE are shown below:

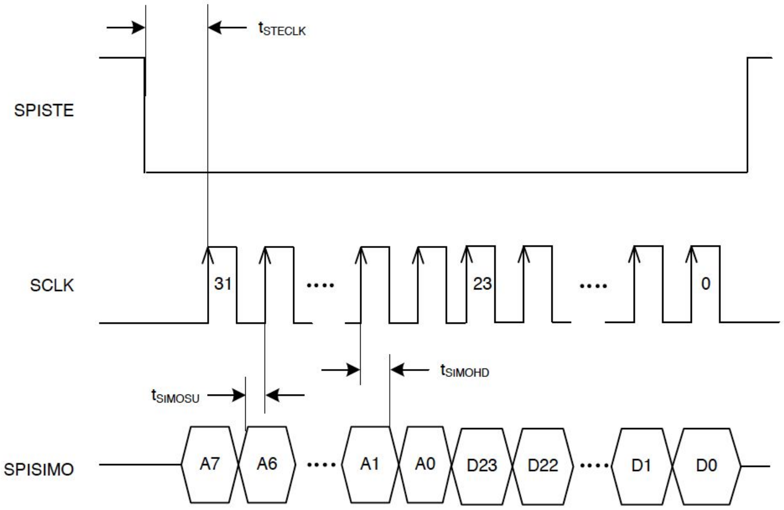
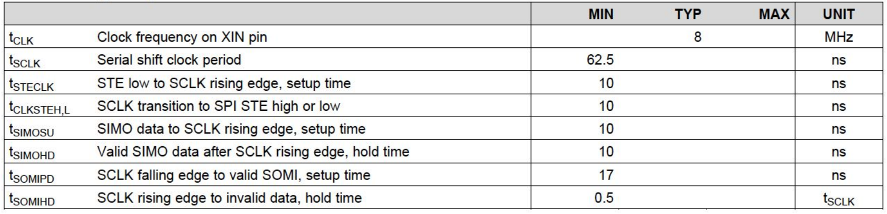
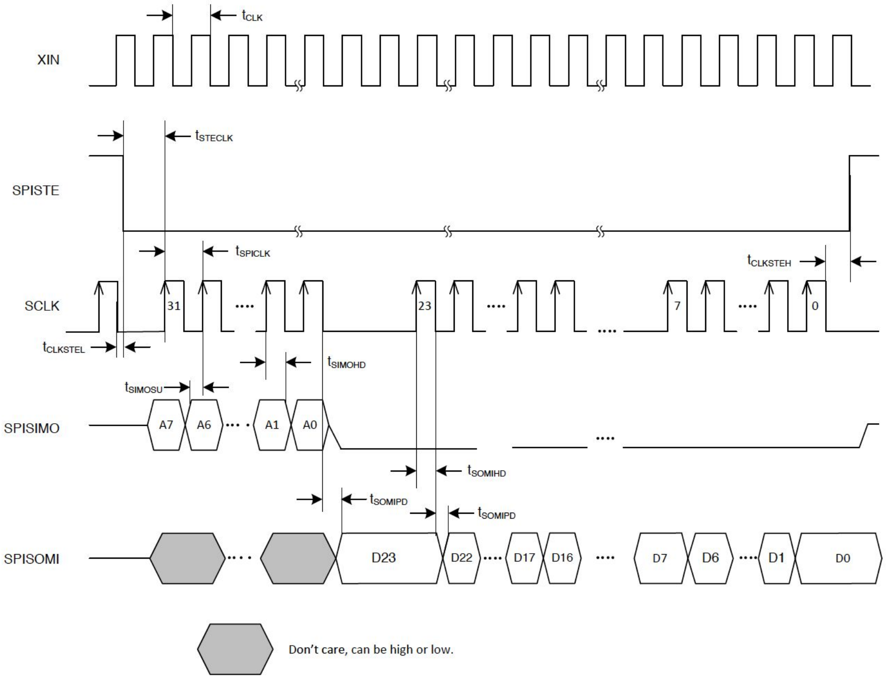


Figure 20. SPI Timing Diagram, Write Operation

7.6 Timing Requirements: Serial Interface



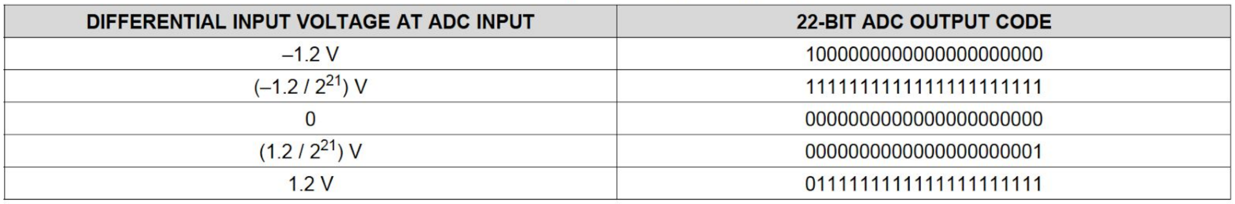
		MIN	TYP	MAX	UNIT
t_{CLK}	Clock frequency on XIN pin		8		MHz
t_{SCLK}	Serial shift clock period	62.5			ns
t_{STECLK}	STE low to SCLK rising edge, setup time	10			ns
$t_{CLKSTEH,L}$	SCLK transition to SPI STE high or low	10			ns
t_{SIMOSU}	SIMO data to SCLK rising edge, setup time	10			ns
t_{SIMOHD}	Valid SIMO data after SCLK rising edge, hold time	10			ns
t_{SOMIPD}	SCLK falling edge to valid SOMI, setup time	17			ns
t_{SOMIHD}	SCLK rising edge to invalid data, hold time	0.5			t_{SCLK}



(1) The SPI_READ register bit must be enabled before attempting a register read.
(2) Specify the register address whose contents must be read back on A[7:0].
(3) The AFE outputs the contents of the specified register on the SOMI pin.

Figure 21. SPI Timing Diagram, Read Operation

The digital values are sent back in twos complement format in order to simplify the presentation of negative values. One of the software functions deployed to the microcontroller will convert this value to a functional voltage value falling within the range of acceptable values (+/- 1.2V) as noted by the microcontroller data sheet below:



DIFFERENTIAL INPUT VOLTAGE AT ADC INPUT	22-BIT ADC OUTPUT CODE
–1.2 V	1000000000000000000000
(–1.2 / 2^{21}) V	1111111111111111111111
0	0000000000000000000000
(1.2 / 2^{21}) V	0000000000000000000001
1.2 V	0111111111111111111111

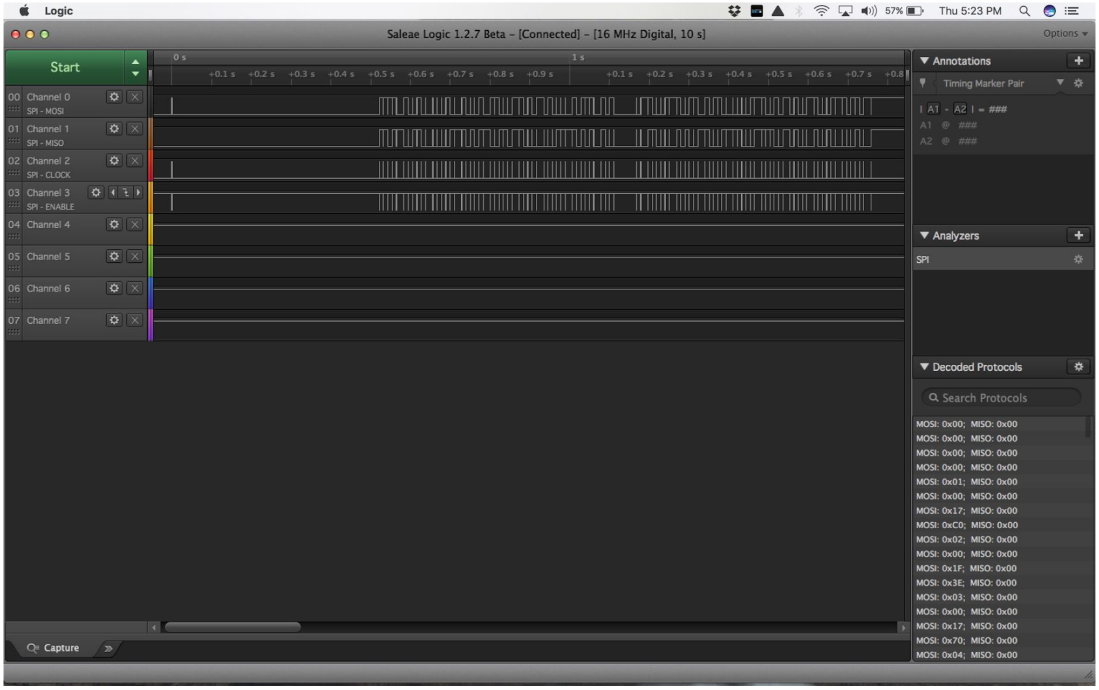
Figure 22. Input Voltage Mapping

Subsystem Testing

A Saleae USB-connected logic analyzer was used in order to test the communication on each branch of the system (between the PIC and AFE and between the PIC and microSD card). The logic analyzer was set up with for an SPI configuration matching the necessary settings of the hardware components as written into our software--most significant bit first, 8 bits per transfer, clock set to idle low, data valid on leading clock edge, and enable set to active low. The MOSI, MISO, SPI Enable, and SCK lines were connected to their corresponding wires on the board with the provided clips, and the system was set to trigger on the falling edge of the enable. The sample time window was set corresponding to the amount of samples being taken (as hardcoded into the main function of the driving software file. After programming the PIC, the reset button was pushed and the Saleae collected the data.

For several iterations, the logic analyzer made clear that all of the output lines coming from the microcontroller to the AFE, and later, to the microSD card were functioning perfectly well. However, no response from either of these secondary components was being sent. This facilitated isolating the problem in communication to the slave in both branches. It was quickly determined that sufficient power was not being supplied to either of the slaves. Once that was repaired, communication in both directions on both branches could be readily observed.

Below is a screenshot of the SPI communication between the PICKIT3 and AFE4490, as captured by the Saleae logic analyzer. The first, very brief set of signals occurring immediately after 0s is the series of “writes” to the AFE setting up the timer and other analog system registers. The clearer string of signals on all four lines beginning after 0.5 s corresponds to the reading of the six data registers--the MOSI line repeatedly transmits their addresses separated by provocative non-operation commands and the MISO line carries the returned value held within those six registers at the time of their individual provocation.



```
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x01;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x17;  MISO: 0x00
MOSI: 0xC0;  MISO: 0x00
MOSI: 0x02;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x1F;  MISO: 0x00
MOSI: 0x3E;  MISO: 0x00
MOSI: 0x03;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x17;  MISO: 0x00
MOSI: 0x70;  MISO: 0x00
MOSI: 0x04;  MISO: 0x00
```

Figure 23. Logic Analyzer Capture of SPI Communication

3.6 Detailed Design/Operation of the MicroSD card communication and Data Storage

System Flow Chart

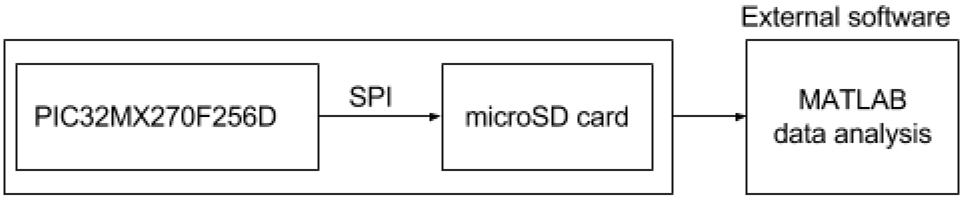


Figure 24. Data Storage Flow Chart

Software Flow Chart

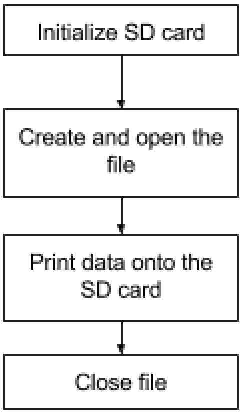


Figure 25. SD Card Code Flow Chart

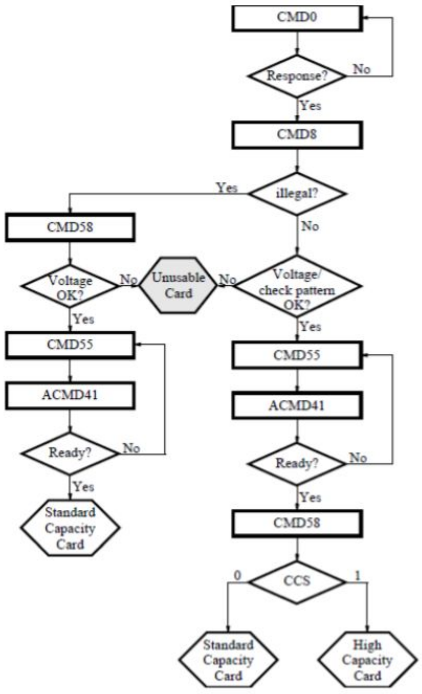


Figure 26. Initialization with SPI protocol for SD card

Schematic

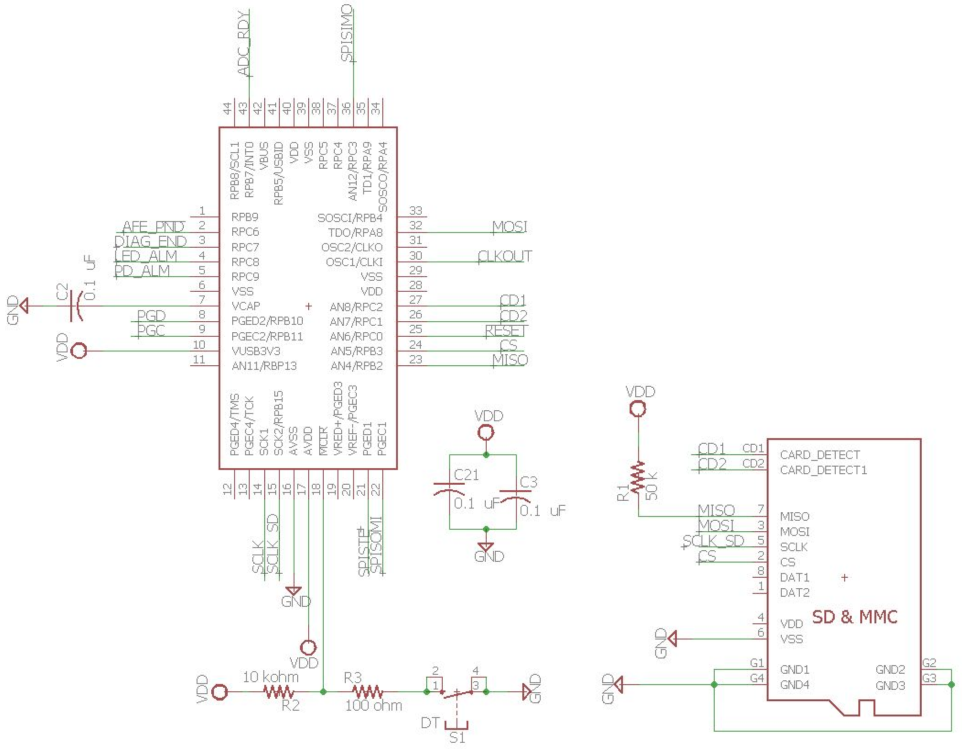


Figure 27. Microcontroller to microSD Card Schematic

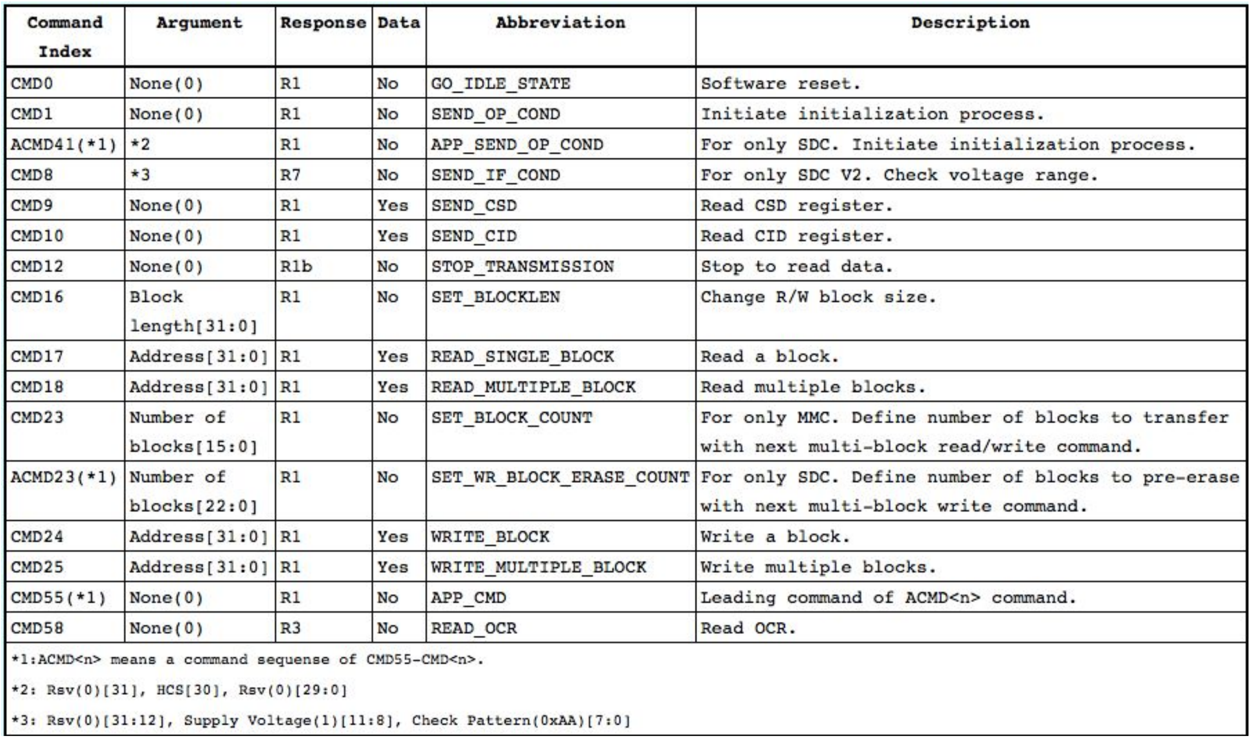
Function and Connection to Other Subsystems (Hardware)

The microcontroller sends data to the microSD for storage. The card is then physically extracted from the board and inserted into a microSD card adapter to plug into the computer. The external MATLAB software program then analyzes the data from the microSD card.

The microSD card connects its two card-detect signals to port pins 26 and 27 of the microcontroller. The chip-select of the microSD card, pin 2, is also connected to a port pin of the microcontroller at pin 24. The microSD card extracts data from the microcontroller via SPI communication. The SCLK of the microSD card is controlled by the second SPI clock of the microcontroller, SCLK2 at pin 15. The master-out-serial-in (MOSI) of the microSD card connects to the remappable pin 32 of the microcontroller. The master-in-serial-out (MISO) of the microSD card connects to the remappable pin 23 of the microcontroller and the board's power system, VDD.

Function and Connection to Other Subsystems (Software)

When the microcontroller receives data back from the AFE, it sends the information to the microSD card so it can later be analyzed on the computer. In order to use the data, the SD card needs to have a file system so that the information can then be read from a file. The microSD card has a FAT32 (file allocation table) file system on it. In order to access the file system and open a file, the microcontroller uses SPI protocol to communicate with the SD card. SPI communication writes to the microSD card and creates a .TXT file that can be read and used for data analysis. Using the Microchip Memory Disk Drive File System (MDD), specific commands, shown in the table below, communicate with the SD card in simplified functions. The functions from the MDD File System set up a file in the SD card and write data onto the card. The originally tested code used for communicating with the SD card came from a sample demonstration code from Microchip. This code was adapted for usage by the PIC32MX270F256D.



Command Index	Argument	Response	Data	Abbreviation	Description
CMD0	None(0)	R1	No	GO_IDLE_STATE	Software reset.
CMD1	None(0)	R1	No	SEND_OP_COND	Initiate initialization process.
ACMD41(*1)	*2	R1	No	APP_SEND_OP_COND	For only SDC. Initiate initialization process.
CMD8	*3	R7	No	SEND_IF_COND	For only SDC V2. Check voltage range.
CMD9	None(0)	R1	Yes	SEND_CSD	Read CSD register.
CMD10	None(0)	R1	Yes	SEND_CID	Read CID register.
CMD12	None(0)	R1b	No	STOP_TRANSMISSION	Stop to read data.
CMD16	Block length[31:0]	R1	No	SET_BLOCKLEN	Change R/W block size.
CMD17	Address[31:0]	R1	Yes	READ_SINGLE_BLOCK	Read a block.
CMD18	Address[31:0]	R1	Yes	READ_MULTIPLE_BLOCK	Read multiple blocks.
CMD23	Number of blocks[15:0]	R1	No	SET_BLOCK_COUNT	For only MMC. Define number of blocks to transfer with next multi-block read/write command.
ACMD23(*1)	Number of blocks[22:0]	R1	No	SET_WR_BLOCK_ERASE_COUNT	For only SDC. Define number of blocks to pre-erase with next multi-block write command.
CMD24	Address[31:0]	R1	Yes	WRITE_BLOCK	Write a block.
CMD25	Address[31:0]	R1	Yes	WRITE_MULTIPLE_BLOCK	Write multiple blocks.
CMD55(*1)	None(0)	R1	No	APP_CMD	Leading command of ACMD<n> command.
CMD58	None(0)	R3	No	READ_OCR	Read OCR.

*1:ACMD<n> means a command sequense of CMD55-CMD<n>.

*2: Rsv(0)[31], HCS[30], Rsv(0)[29:0]

*3: Rsv(0)[31:12], Supply Voltage(1)[11:8], Check Pattern(0xAA)[7:0]

Figure 28. SD Card SPI Commands

Subsystem Testing

The microcontroller-microSD card subsystem testing used breakout boards that connected the SD card to the class kit boards. Communication was established between the microSD card and the microcontroller using SPI protocol, and a file was created on the microSD card. Initially, the FATF system was tested, but failed to provide adequate data. In order to further test the subsystem, Microchip's Memory Disk Drive File System, which uses the FAT32 on the SD card, outputted a signal, proving communication.

3.7 Detailed Design/Operation of the Power System

System Flow Chart

The power system consists of a two part battery system. The first is a charging system and the second is a DC to DC booster.

Figure 29. Battery Charging System Flow Diagram

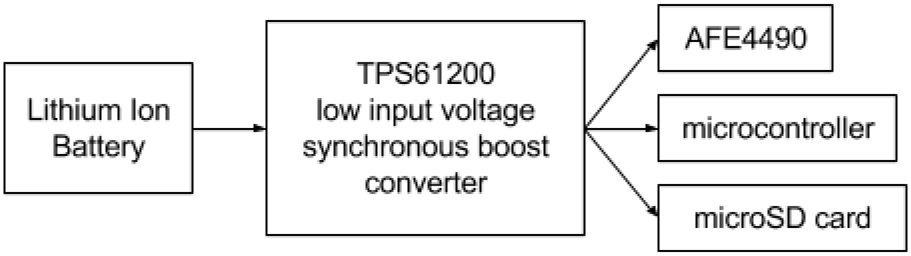


Figure 30. Power Distribution Flow Diagram

Schematic

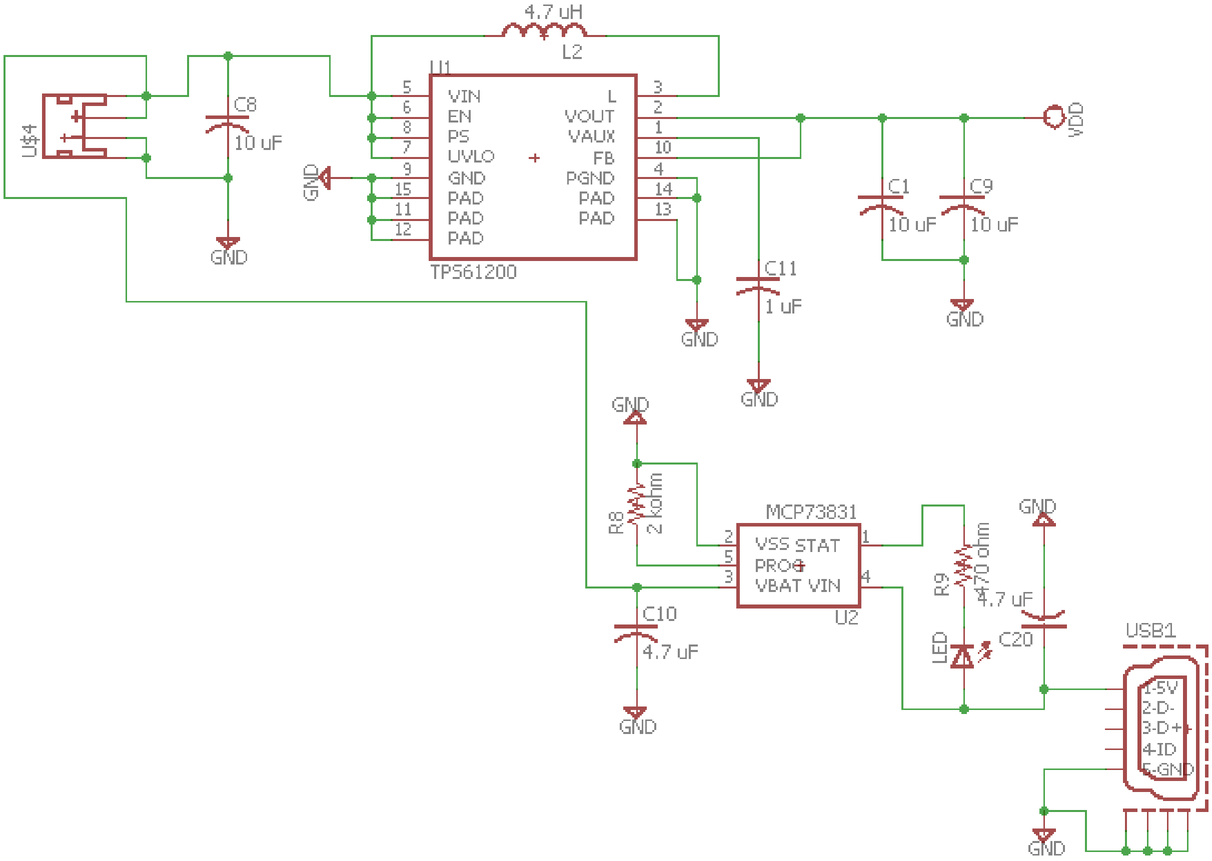


Figure 31. Power System Schematic

Function and Connection to Other Subsystems

The battery power charging system uses an external power supply via a miniUSB to charge a 3.7 V lithium ion battery. The MCP73831 is a miniature single-cell, fully-integrated Li-Ion, Li-Polymer, charge management controller by Microchip. The device employs a constant voltage, constant charge algorithm to charge the Li-Ion battery. The MCP73831 was suitable for a small space and is able to charge the 4 V Li-Ion battery in use via a USB input power.

The charged Li-Ion battery then becomes the source of power to the entire board system. The TPS61200 is a low input voltage synchronous boost converter. The TPS61201, a version of the TPS61200, takes the Li-Ion battery as its input and outputs a fixed 3.3 voltage. The 3.3

output voltage becomes the power source which powers the microcontroller, AFE4490, and microSD card. The microcontroller, AFE4490, and microSD card can all be powered by a 3.3 V input voltage.

Subsystem Testing

The charging circuit requires two tests. First, when the external power source is connected via the miniUSB port, the illumination of the red diode light indicates that the MCP73831 is receiving an input voltage. If the red diode does not light up, then the charging system is not receiving enough input voltage from the external power supply. Next, the voltage of the battery can be tested by measuring the voltage across the battery at the negative and positive terminals. A fully charged Li-Ion battery should reach about 3.7 V to be used as an input to the battery regulating circuit system.

The battery power subsystem testing requires testing the voltages at the VDD points on the board. The input voltage of the battery should be around 3.7 V. After regulation by the TPS61201, the output voltage should be 3.3 V. Testing the the VDD points, which should be 3.3 V, on the components on the main board (AFE4490, microcontroller, and microSD card) will test whether the power regulating system is working.

4. System Integration Testing

Initial testing made use of several "breakout boards." The first breakout board included the AFE4490 to allow for subsystem testing of the the microcontroller SPI communication with the AFE4490. However, the breakout board, due to a design mistake, did not include power to the board and was unable to be used for testing. The second breakout board included the dual red/IR LED with pins on the end of the board to connect to the evaluation board. The final breakout board included the photodiode coupled with a 5 uF capacitor, for noise cancelling purposes, with pins on the side of the board to connect the terminals to the evaluation board. The red/IR LED board and photodiode board were connected to the DB9 connector of the evaluation board. This allowed for the testing of the sensor. Reflection and transmission were tested, and reflection of light proved to provide the clearest and most accurate signal.

Several subsystems had to be tested before the final system integration testing, as described in section 3 above. First, the microcontroller communication to the AFE4490 was confirmed via a logic analyzer. Then, communication back from the AFE4490 to the microcontroller was tested. In order to test this microcontroller to AFE4490 and AFE4490 to microcontroller SPI communication, a logic analyzer was hooked up to the CLK, SPI enable, MISO, and MOSI pins of the microcontroller. The MOSI line was first used to write values to registers within the AFE, controlling the timer, gain, LED current, etc. After setting these registers, the MOSI line continually sends the addresses of the six sampled registers (separated by non-operation bytes to provoke each byte of the value in that register) whose values are taken in by the detector. The MISO responded to the register call of the MOSI line with the converted digital values contained in the register being read. The debugger, particularly the part that allows the tracking of variables, was used to make sure the two's complement conversion of the data was done correctly.

Next, the SPI communication between the microcontroller and microSD was confirmed.The communication was tested through the use of the logic analyzer and the debug mode in MPLAB. The most recurring issue that was found in the communication between the SD card and microcontroller was that it would fail to move on from the initialization of the SD card, so the logic analyzer would show the initialization commands being sent in an unending loop. When this happened, the debugger was used to help find at what point in the code the program was running into a fault, which led to the conclusion that the SD card was not being found. The SD card was also checked to see if it had the correct text file name with data that seemed correct.

The sensor was then connected to the processing board. The lighting up of the LED was the first test of successful communication. Using an oscilloscope, the LEDs were confirmed to run at 500 Hz. This indicated that the communication between the AFE4490 and LEDs was working correctly. Next, the photodiode needed to capture the reflected light from the LEDs. A voltmeter confirmed that current was being passed through the photodiode to the AFE4490.

The system integration testing relied on the testing of these points between each subsystem. The SD card was ejected and inserted into the computer. The opening of a text file with data ensured that the processing board and detector system were working and communicating correctly. To test that the data was usable, the six columns, which represented voltages for the light and ambient light of the red and IR LEDs and the differences between light and ambient light for the red and IR LEDs, in the text file were quickly plotted on excel. Correct data was represented by a PPG waveform. Finally, the data was imported into the MATLAB GUI and processed by the GUI.

5. User's Manual/Installation manual

5.1 How to install and setup the product

First, use external wire to connect the photodiode and LED board to the main board, which has the microcontroller, AFE chip, power and SD card. The four through pins on the edge of the board indicate the connections needed for the LEDs and photodiode. The anode of the red LED (cathode of the infrared LED) connects to hole one of the LED-labeled through pins and the cathode of the red LED (anode of the infrared LED) connects to hole two. The anode of the photodiode connects to hole one of the PD-labeled through pins and the cathode of the photodiode connects to hole two. Before inserting the SD card into its connector in the board, plug it into a computer and verify that the 16GB card has adequate memory space. Ideally, the card should be clear. Next, charge the Li-Ion battery by connecting the battery to the JST-Molex connector and connecting an external power source (a computer) using a mini USB cord. Once the battery is charged, keep the battery in the JST-Molex connector as a power source to the board.

On the software side, open AFE_SD_code.c on MPLAB. Using the PICKIT3, load and run the the program file onto the microcontroller. Once the download is complete and verified, unplug the PICKIT3 from the board. Place the board in the arm band. Place the LED/photodiode board in the finger band. Once the boards are placed in the bands, strap the arm and finger bands on the body. Note that between each use of the device, the SD card file name in the AFE_SD_code.c must be changed from the previous version.

The external software program uses MATLAB. Load SD_GUI.m on MATLAB and check "Utilizing Eval Board" to run the GUI. The MATLAB GUI analyzes the data. Enter the filename of the data that is stored in excel from the SD card. Input the time range that the user would like to view. For example, inputting 0 as the start time and 5 as the end time provides data from 0 to 5 seconds. Enter the user's age and height. A PPG waveform will appear in the top graph. Additionally, the calculations of heart rate, HRV, IPA, AI, and SI update on the right side of the GUI. On the top graph, selecting "Plot HbO2 Concentration" or "Plot Hb Concentration" will update the Concentration vs. Time plot. On the bottom graph, selecting "First Derivative" or "Second Derivative" will update the Derivative of Concentration vs. Time plot. Changing the filter percentage to the left of each graph will update the plots. Finally, a dropbox below the box in the lower right hand corner of the GUI allows the user to select specific reference materials related to the PPG waveform.

5.2 How to use the product

1) Place the wearable PPG device on the arm and finger. Verify that the SD card is connected and battery is inserted.
2) Wear the device, click the reset button, and collect an ample amount of data.
3) Remove the SD card from the device and plug it into the computer.

4) Manually load the data from SD card into Excel so it can be used for analysis in the MATLAB program.
5) Use the MATLAB GUI.
    a) Type in the File Name.
    b) Set a start and end time.
    c) Enter the age and height of patient.
    d) Select the plots for the Concentration vs. Time plot and Derivative of Concentration vs. Time plot.
    e) Use the drop down box on the right side of the GUI to select reference materials related to the the PPG waveform.
    f) View the calculations section to see heart rate, heart rate variability (HRV), inflection point area (IPA), augmentation index (AI), arterial stiffness index (SI).

5.3 How the user can tell if the product is working

The first indication that the product is working is the illumination of the LED of the sensor. Checking on an external computer that data was retrieved on the SD card confirms that the sensor sent a signal through the processing board. After the user has the data on the SD card, he or she can see whether the GUI accepts the excel filename that is typed into the input box. After the filename is inputted, the GUI should output a PPG waveform, derivative graphs, and parameter calculations on the right.

5.4 How the user can troubleshoot the product

If the user runs into issues when trying to use the product, the user can make sure that the cables from the main board to the smaller board, with the LEDs and photodiode, are connected. Also, the user should also make sure that the SD card is well inserted into its slot. If the light on the sensor does not illuminate, measure the voltage on the Li-ion battery and make sure that it is adequately charged above 3.3 V. The user can also attempt renaming the filename of the .txt file in the MPLAB code, re-downloading the program onto the microcontroller, and hitting the reset button on the board. If none of these things work, then there is probably a bigger hardware error that will be difficult for the user to fix.

6. To-Market Design Changes

A more advanced prototype of the device for the market would have several significant improvements. Physically, the main board would be slightly smaller for a more comfortable fit on the forearm. This change could be made by simply rearranging the board layout to save more physical space. Next, an improved device would have a more direct flow of data from the board to the computer without having to manually transfer the microSD. A solution to manual data transfer could be solved using wifi or bluetooth communication from the data storage to the external computer. Wifi or bluetooth could also allow for a live transfer of data to the computer. Finally, a phone application to livestream the heart data would allow the user or other people to follow the information in real time. The app would essentially replace the external computer and analyze the data from the main board. An app would also use wifi or bluetooth to retrieve the data from the main board.

7. Conclusion

Overall, the senior design project was a success. The goal was to create a photoplethysmography device which uses optics to measure blood volume. The project successfully charges a Li-ion battery, regulates power across the board, flashes red and IR LEDs, captures the reflected light on a photodiode sensor, processes the data via the board processing system, stores the data, and analyzes the PPG waveform created from the data. While improvements to the design could be made, this was the first prototype of a wearable PPG device. The overall functionality of the device was effective and successful.

8. Appendices (Attached/ Found on Website)

Appendix A: Hardware Schematics and Component Datasheets
Appendix B: Complete Software Listings
Appendix C: Research Documents and Links