ESE 519: Real Time Embedded Systems
Final Report:Abdominal Auditory Sensor

Team Members:

1) Jono Sanders
2) Hitali Sheth
3) Sindhu Gurudutt Honnali

Motivation:

A loss of gastrointestinal motility, commonly known as postoperative ileus (POI), occurs after abdominal surgery. Since the 1900s, nurses and other clinicians have been taught to listen for return of bowel sounds to indicate the end of POI.

Research has shown that changes in abnormal intestinal motility led to abnormal sound patterns, and that these can be monitored. Constant monitoring of bowel motility may help in predicting, diagnosing, or monitoring motility disorders. We plan to build a device which senses and collects the data which can be analyzed by doctor to suggest dirt patterns for the patients after abdominal surgery. The purpose of this project is to provide evidence from a randomized clinical trial and rationale supporting evidence-based inquiry concerning return of bowel sounds as an unreliable indicator of the end of POI after abdominal surgery.

Goal:

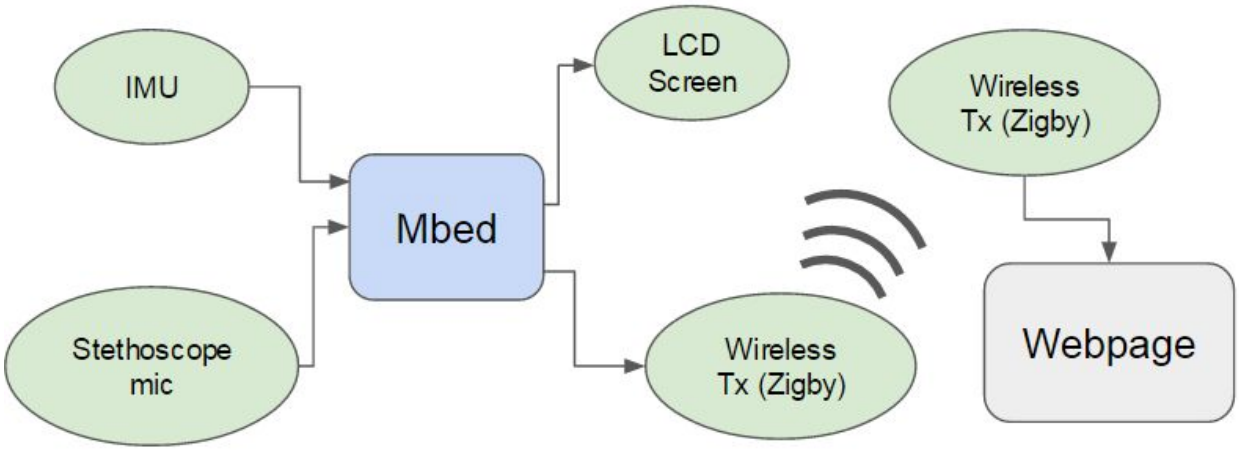
1. Designing a low cost, closed loop device that can be worn across stomach as a belt to capture, detect and classify the sound pattern.
2. Update the relevant details onto a private server so that doctor can easily monitor the actual evidence rather than a random clinical trial.

Methodology:

1. Integrating stethoscope with the microphone to capture the sounds produced by stomach.
2. Designing the filter circuit to capture the low frequency, low amplitude signal.
3. Capturing various sound patterns using the above circuit: bowel sounds, noise produced by rustling, talking, moving around.
4. Using various classifying algorithm on the captured data to come up with a good classifier which is appropriate to our application.
5. Integrating IMU with the mbed to obtain the accelerometer reading.
6. Building the speaker circuit for validation purpose, to make sure that the captured sound are the sounds we are looking for.

7. Mounting all the circuits on laser cut plates to build a compact device and attaching this onto a belt.
8. Simple web application to store and display the status of the patients.

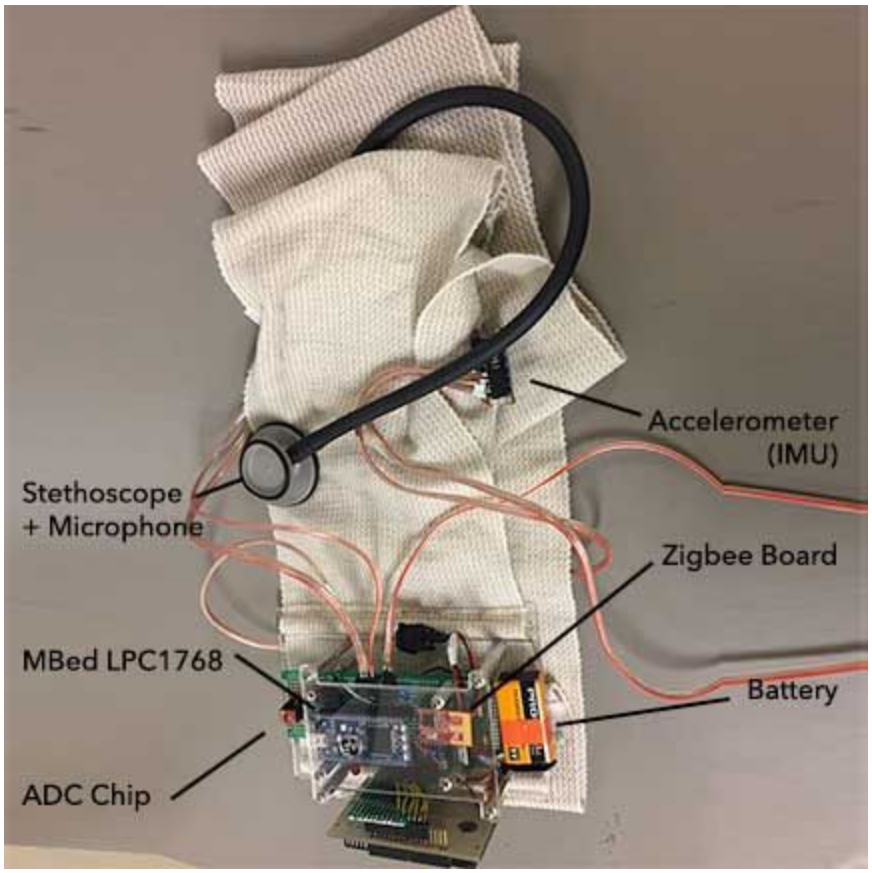
System architecture:



1. Stethoscope and IMU are embedded inside the belt. Stethoscope along with microphone captures the sound along with the noise.
2. Differential circuit followed by sixth order filter circuit will filter out the noise and send clean stomach sounds to mbed.
3. IMU also sends acceleration along x, y and z axes to mbed.
4. MBED performs FFT on the signal. FFT coefficients are provided to the decision tree which classifies it as ileus or bowel sound.
5. MBED also classifies if the person is walking or standing based on IMU values.
6. Both IMU and Stethoscope readings are displayed on the LCD screen on the device.
7. Both IMU and Stethoscope readings are sent to the webpage via zigbee. The webpage displays the status of each patient - if the person is allowed to eat or not and also the duration of time the person has exercised by walking.

Project components:

Hardware



1. MBed LPC1768
2. Stethoscope
3. Microphone breakout board
4. IMU
5. Zigbee
6. Battery
7. Laser cut plates
8. PCB
9. Speakers

Software

1. Matlab
2. Audacity
3. MBed compiler
4. Python flask
5. Zigbee communication

Detailed development steps:

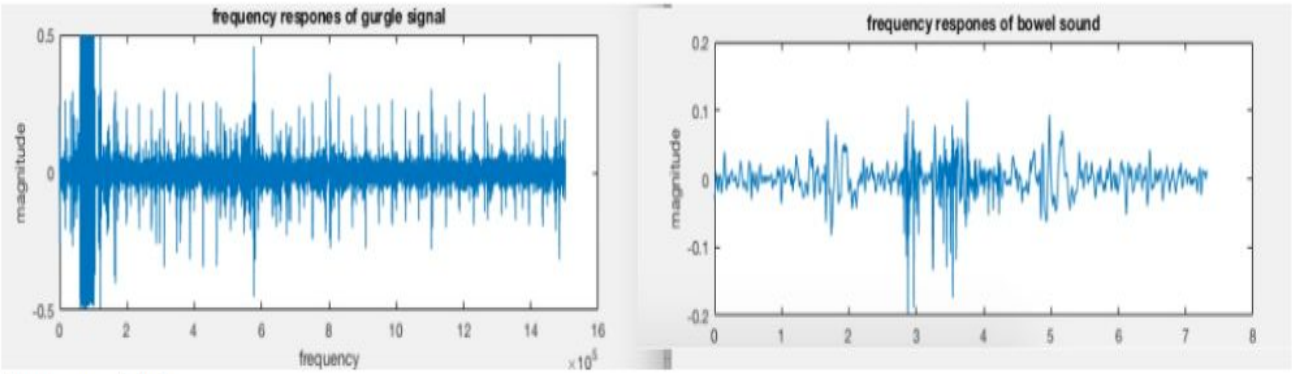
(a) Software Effort :

1. We initially recorded audio sounds from stomach (.wav files) using the software 'Audacity' and processed these files in matlab.
2. Later we replicated software processing steps in hardware (as described in hardware section)
3. The hardware processed signals were then classified on the controller.

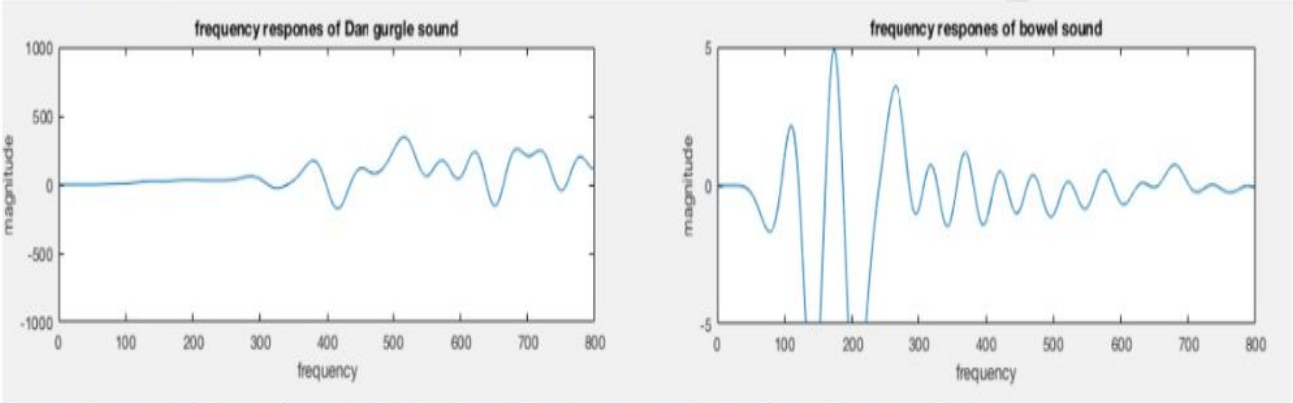
Sound Processing :

The audio files (Stomach sounds for 2/3 seconds) obtained through audacity had very low amplitude and lot of noise. We used the following algorithm to get a better signal.

The analysis was done in a MatLab interface. The algorithm used opens the file, applies an eighth-order low-pass Chebyshev Type I filter with a cutoff frequency of 882 Hz and decimates it at 1/5 of the original sample rate, resulting in 2205 samples per second. Then a high-pass filter (3 dB at 80 Hz) was applied in order to remove heart and breath sounds.



Raw sound data

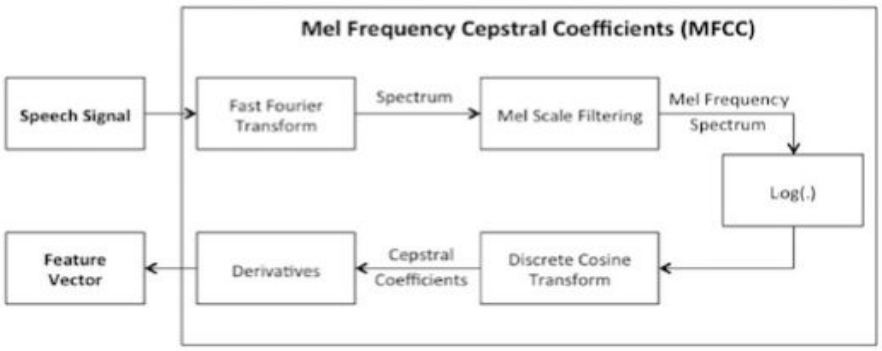


Stomach sounds filtered through Chebyshev 8th order and decimated to reduce amplitude

Since amplitude and frequency are insufficient to adequately represent the sound signals we obtained, we use MFCC coefficients which gave us a feature vector. The MFCC coefficients

basically give us a higher number of features for each sound sample, extracted from its frequency and amplitude.

Mel Frequency Cepstral Coefficients :



The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Having these coefficients extracted from the FFTs helped us to represent audio sounds effectively and be able to classify them.

Machine Learning :

We tried various machine learning algorithms to classify the sounds. SVM, Naive Bayes and Decision tree were implemented in matlab as well as Python to process the signals.

Decision trees gave us the highest accuracy and we decided to implement that on our processor. Fifty samples were used to train the models. These samples considered talking, absolute stomach sounds (after eating), no stomach sounds and walking around scenarios.

Our decision tree classification identified the most important features to use for labeling our data samples. This was helpful when we had to implement the model on ARMLPC 1768, since we faced memory constraints using the onboard RAM of the board.

We then used these decision points and implemented a decision tree on the board. Though decision tree is not the best model choice, with the considered ARM LPC1768 memory constraints and limited training samples, it was the most practical choice for our application.

Decision tree gave us accuracy of around 65%.

IMU data:

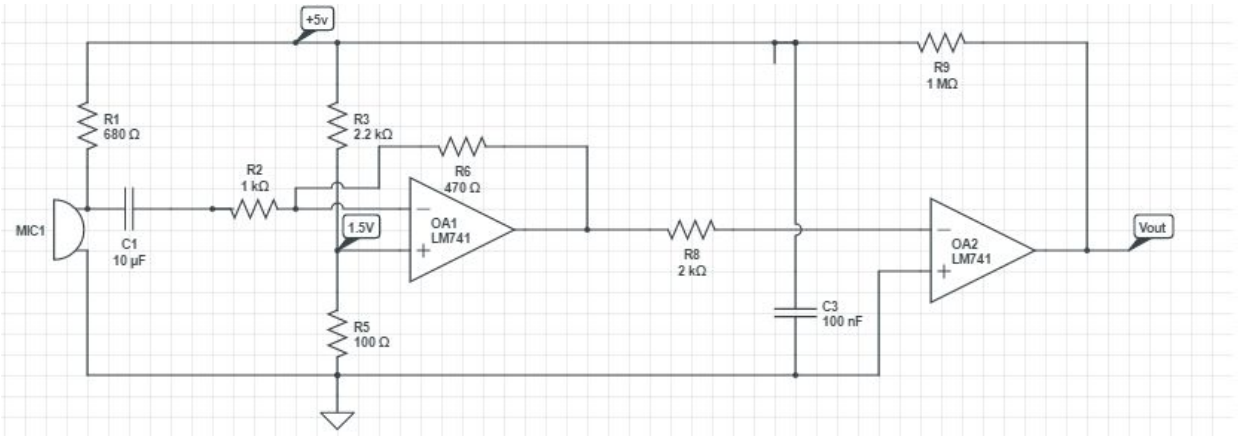
The acceleration in X, Y and Z directions was recorded from the IMU. These values were then calibrated in the initial 5 seconds.

The changes in the acceleration were then used to classify and measure movement (walking/not walking). Thresholding was used in all three directions, to be able to correctly measure movement. Moving average was calculated so that movement over time could be measured.

Once IMU and Stethoscope data was collected, the device transmitted count and noise characterizations over Zigbee to a remote computer; from where it was updated to database and server.

(b)Hardware effort:

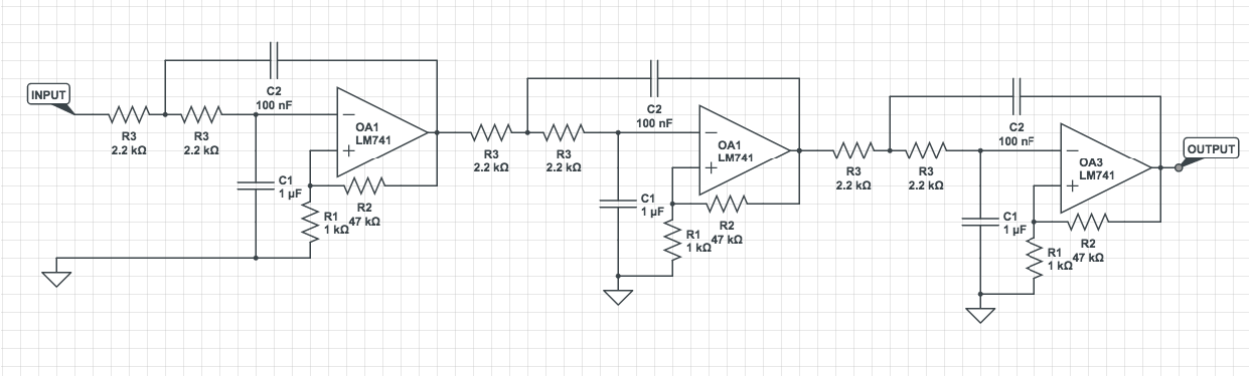
Based on software filtering results, equivalent filters were mimicked with hardware. Since the microphone breakout board was offsetting the captured signal by 1.5V, we used a differential low pass filter circuit to process the actual signal.



Differential low pass filter circuit

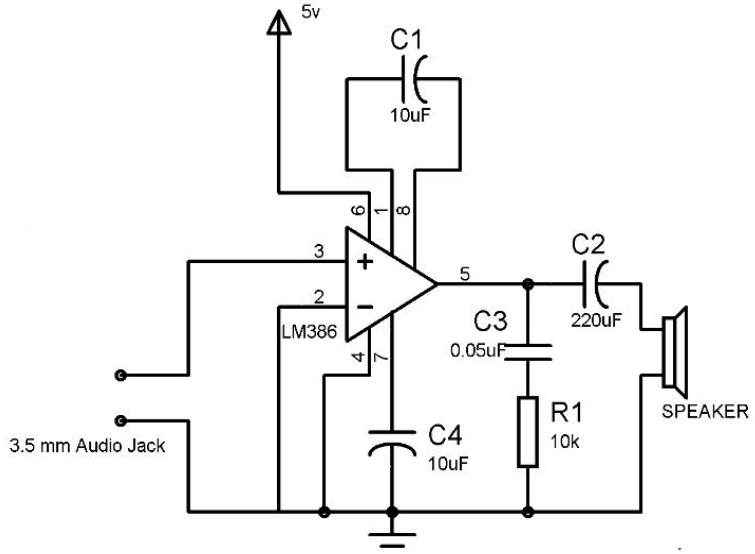
The output of differential circuit is given to the sixth order low pass filter circuit. The circuit is designed to have a cut off frequency at 800 hz, since the bowel sound ranges from 20 - 800Hz.

A second order low pass filter was initially built, the same was replicated and cascaded to get a sixth order low pass filter. The filtered output still contains some white noise in it.



Sixth order low pass filter circuit

Since the sounds we are targeting are low frequency and low amplitude signals which can not be easily felt or heard unlike heart beats, we built a speaker circuit to validate the captured signal. The output of the above filter is given to the speaker circuit to recreate the sound.

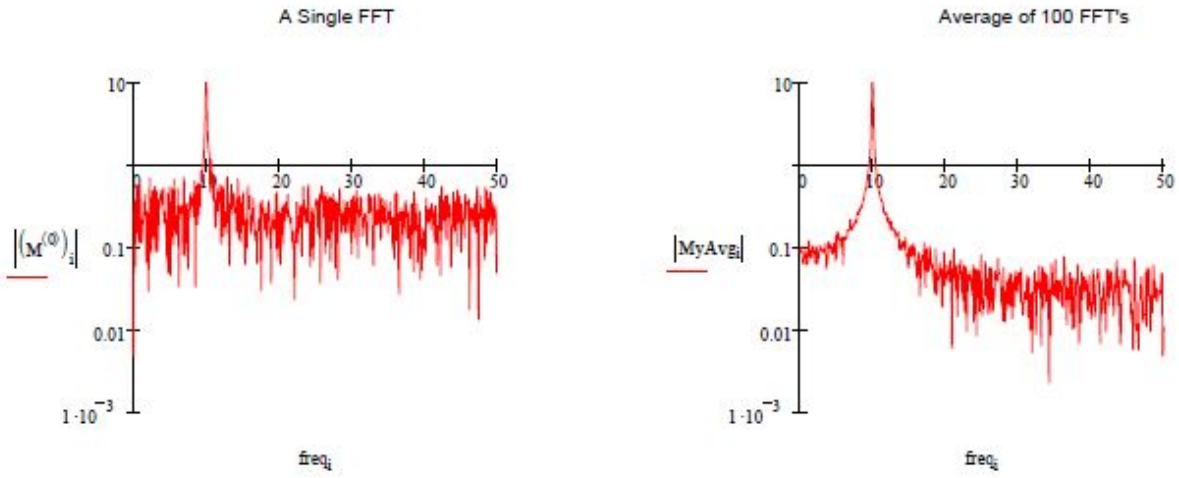


Speaker circuit

The output of the filter circuit is also given to the mbed microcontroller. The controller takes the signal and performs FFT on the signal.

The bowel movement lasts for at least 2- 3 seconds and sampling was done at the rate of 4000 hz, which is greater than four times the highest frequency of the signal (i.e. 800hz). When the signal is sampled at this rate for 2-3 seconds, MBED has no sufficient memory to hold the sampled results. Due to this limitation, FFT averaging was performed on the signal. A window of 256 samples are captured and each of these are summed with the previous window for 2

seconds. The final average of summed FFT coefficients were given to the decision tree in order to classify the captured sound.

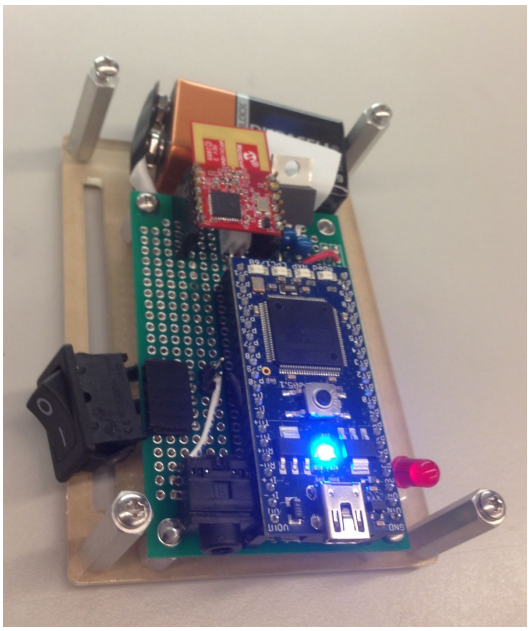


It is also observed that averaging cuts down a lot of noise, which helped us with further filtering.

IMU data is polled for every 3s and is used to classify the activity of the patient. The result of Stethoscope and IMU reading are displayed on attached LCD screen.

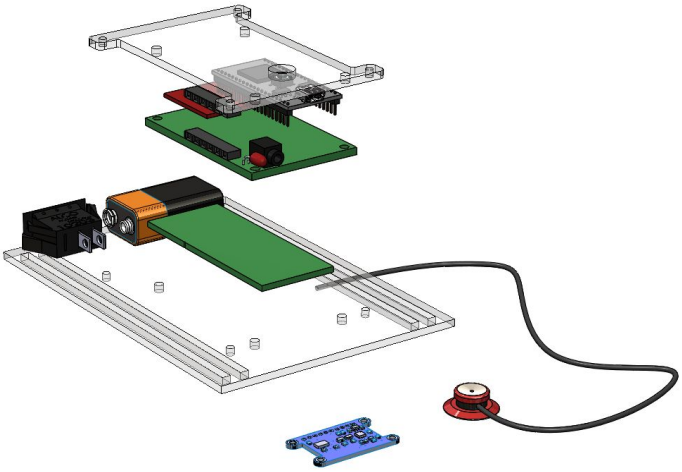
(c) Mechanical effort:

We started out by trying various configurations of microphones and stethoscopes, finally settling on a glued microphone inside a cut segment of the stethoscope tube, to give us isolated sound for processing. We used this microphone in conjunction with an assembly of boards and components, which we built up in an assembly that we could strap around the body. This form factor made it easy to record sounds and also display information on an LCD screen for updates.



Mounting board for Mbed, Zigby and connectors

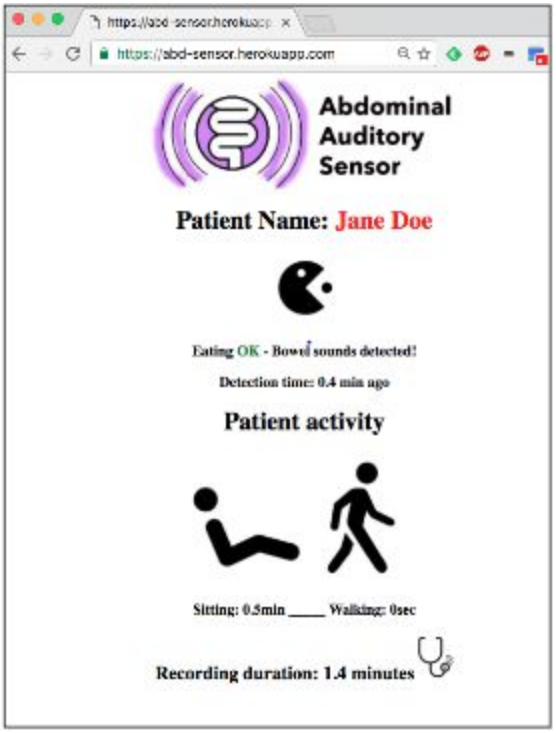
We modeled the assembly in Solidworks and laser cut acrylic pieces for the structure of the device. We then soldered three perforated boards to hold the MBed, the Zigby board, the ADC chip, the sound filtering circuit and all the connectors for the plugged in components.



3D model of the prototype assembly

(d)UI effort:

The web page to monitor patients activities can be accessed at http://abd-sensor.herokuapp.com/. The web page was built using Python Flask. The web page displays each patient's state indicating if the person is allowed to eat or not and also the duration of time the person was standing and walking.



Snapshot of web page

The data is being sent to the server for every 3 seconds.When a bowel movement is detected for 20 times, we are updating the status of each patient allowing him to intake food. The web page server has a counter which counts the number of times it receives can-eat message for each patient and updates it accordingly on the web page.

Re-build project from scratch:

1. To get the stethoscope reading :
   (a) Connect the diaphragm of the stethoscope to the above filter. The filter formed by connecting the differential low pass filter circuit followed by the sixth order low pass filter.
   (b) The output of the filtered circuit is also sent to the speaker circuit and to the mbed controller.

2. To get IMU reading:
(a) Connect IMU to mbed to obtain accelerometer reading along x,y and z directions and calibrate it.
3. The github code can be used to flash the mbed with the program.
4. The result can be seen on the LCD.
5. The data is also logged into the remote site, https://abd-sensor.herokuapp.com

Video link: https://youtu.be/0Z8l1p9MeSc