DIGITAL SIGNAL PROCESSING PROJECT REPORT

SCHOOL OF ELECTRONICS ENGINEERING

TOPIC: DETECTION OF BREATHING AND SLEEP APNEA CONDITION IN INFANTS

GROUP MEMBERS:

VIDWATH – 17BMD0062.

ABSTRACT

For infants and premature babies and even for adults, sleep apnea is a serious sleeping disorder that occurs when a person's breathing is interrupted during sleep. The thing that motivated us to do this project is to detect breathing and for how long the breathing stops by using devices without direct physical contact with the infants.

WORKING:

The overall design involves acquiring sound from a microphone, this sound is then processed to detect breathing and a timer counts how long between breaths. When an apnea event occurs, longer than 20 seconds without breath, an alarm is sounded.

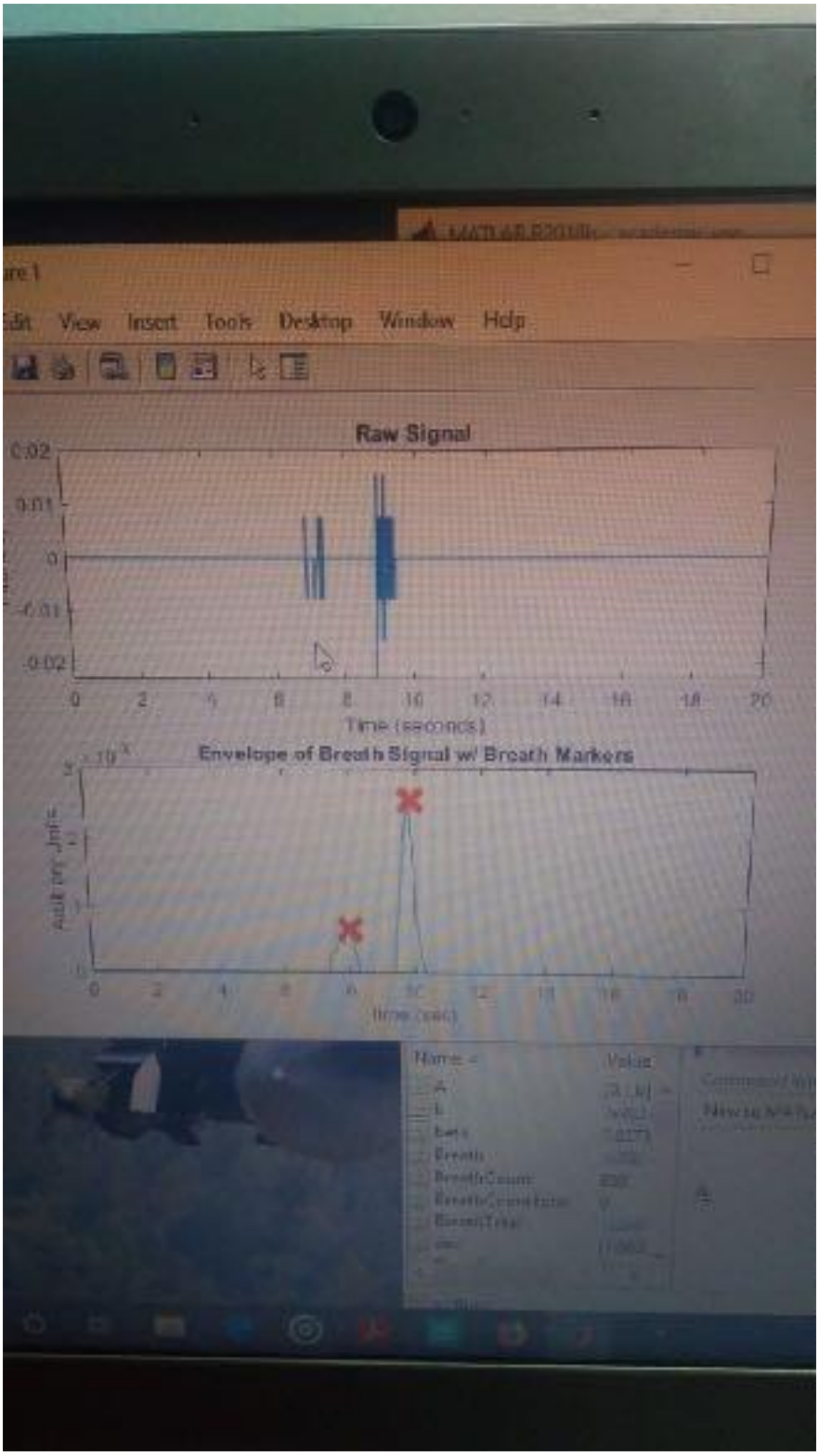
Matlab Prototype

One of the first steps in the project's design was a Matlab model. This served as a proof of concept and a much easier platform to debug than methods used that hardware can easily provide. The program took in audio data from a microphone and detected the peaks in the signal.

Matlab code used:

```
close all, clear all
%% Variables low_stop
= 300; high_stop =
1000;
WindowEnvelope = 0.1; % Length of envelope averaging filter window(seconds)
MaximaWindow = 2.4; % Length of window to find local maxima(seconds)
DownSamp = 200; % Frequency to downsampe envelope to (Hz)
Threshold = 0.2; % Percentage of mean maxima value that a breath must be
above
Fs = 8000; % Sampling Frequency
nbits = 8; % Bits of Precision When Sampling
WindowTime = 5; % Length of recorded time processed in each batch (seconds)
%% Initialize
BreathCountTotal = 0; y =
audiorecorder(Fs,nbits,1);
meanbreath = 0;
%% Create Bandpass Filter
F=[(low_stop-100) low_stop high_stop (high_stop+100)]; % band limits A=[0
1 0]; % band type: 0='stop', 1='pass'
dev=[0.0001 10^(0.1/20)-1 0.0001]; % ripple/attenuation spec
[M,Wn,beta,typ]= kaiserord(F,A,dev,Fs); % window parameters b
= fir1(M,Wn,typ,kaiser(M+1,beta),'noscale'); % filter design
%% Initialize
loopcount = 0; EXIT
= 1;
BreathTotal = []; %% Initial
Recoding signal =
zeros(1,Fs*WindowTime);
record(y) while EXIT == 1  %%
Counter
 loopcount = loopcount + 1;
 %% Filter Signal
 signal_f = fftfilt(b,signal);
23
 %% Find Envelope
 signal_h = hilbert(signal_f); % Hilbert Transform  envelope =
sqrt(signal_h.*conj(signal_h));  envelope_f =
filter(ones(1,round(Fs*WindowEnvelope))/round(Fs*.1),
1,envelope);  envelope_fDS =
downsample(envelope_f,round(Fs*(1/DownSamp)));
 %% Find Local Maxima
```

```
 windowsize = DownSamp * MaximaWindow / 4;
 BreathCount = 0;
Breath = [];
 for ix = 1:length(envelope_fDS) - windowsize
maxima = max(envelope_fDS(ix:ix+windowsize));
if loopcount < 2 % Ignore first 2 loops
     elseif loopcount < 100 % Use next 100 loops to get breath threshold
        if maxima == envelope_fDS(ix+windowsize/2)
BreathCount = BreathCount + 1;
            Breath(BreathCount) = ix + windowsize/2;
end
     else % Begin using and updating threshold
         if (maxima == envelope_fDS(ix+windowsize/2)) && maxima >(Threshold *
meanbreath)
             BreathCount = BreathCount + 1;
             Breath(BreathCount) = ix + windowsize/2;
end      end  end
 BreathTotal = cat(2,BreathTotal,envelope_fDS(Breath));
meanbreath = mean(BreathTotal);  if isempty(Breath) &&
loopcount > 100 == 1      for i = 1:10           beep
pause(.1)      end      EXIT = 0;  end
 %% Keep Recording
stop(y)
 signal_new = getaudiodata(y,'double');
record(y)
 signal(1:end-length(signal_new)) = signal(1+length(signal_new):end);
signal(end-length(signal_new)+1:end) = signal_new;  clear signal_new
 %% Plot  figure(1)
subplot(2,1,1)
signalplot = signal;
 time1 = (0:length(signalplot)-1) / (Fs/4);
refresh  plot(time1,signalplot)
title('Raw Signal')  xlabel('Time
(seconds)')  ylabel('Intensity')
   subplot(2,1,2)
envelopeplot = envelope_fDS;
 time = (0:length(envelopeplot)-1) / (DownSamp/4);
plot(time,envelopeplot)  hold on
plot(Breath/(DownSamp/4),envelope_fDS(Breath),'rx','MarkerSize',
16,'linewidth',4)
title('Envelope of Breath Signal w/ Breath Markers')
xlabel('time (sec)')  ylabel('Arbitrary Units')
hold off end
```


View Insert Tools Desktop Window Help
Raw Signal
0.02
0.01
0
-0.01
-0.02
0 2 4 6 8 10 12 14 16 18 20
Time (seconds)
Envelope of Breath Signal w/ Breath Markers
Amplitude (Units)
Time (sec)
Name
Value

CONCLUSION:

The project was rather successful. As we have aimed to make a matlab prototype was successfully made along with its algorithm. Through this project we have come across lots of new matlab codes which helped us to learn new and potential syntaxes. It could probably require a bit more fit and finish but the system is at least workable. An incoming signal is filtered for the desired range an envelope is created, peaks are detected and a timer is kept.