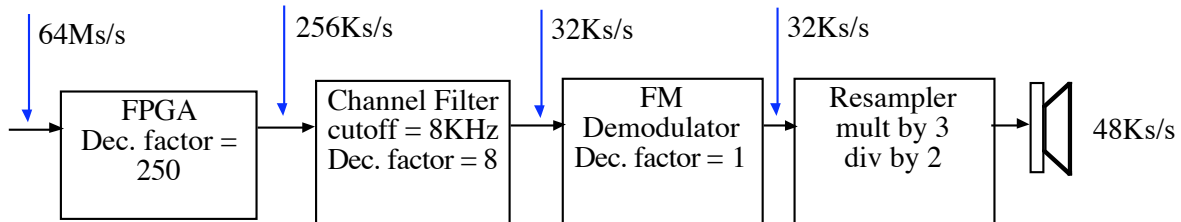


Waveforms and Spectra in NBFM Receiver

GNU radio was used to create the following NBFM receiver. The USRP with the RFX400 daughterboard was used to capture the signal.



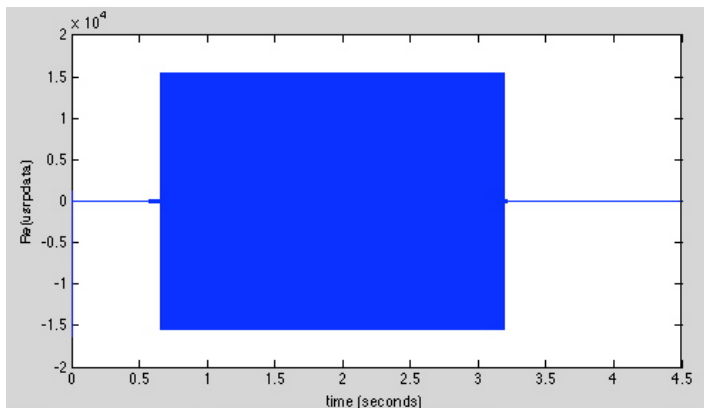
Output of the USRP

The ADC samples at 64Ms/s and sends the samples to the FPGA which decimates by a factor of 250 resulting in a data rate at the output of USRP of $64M/250 = 256Ks/s$. Thus, the time between samples is $1/256K$ or .00391 mS. The file usrp.dat is converted to MATLAB readable form and placed into a matrix called usrpdata. This array contains 1148928 complex values and covers $(1/256K)*1148928 = 4.488$ seconds. This data represents the I and Q values of a 446MHz NBFM signal that has been mixed down to DC (approximately).

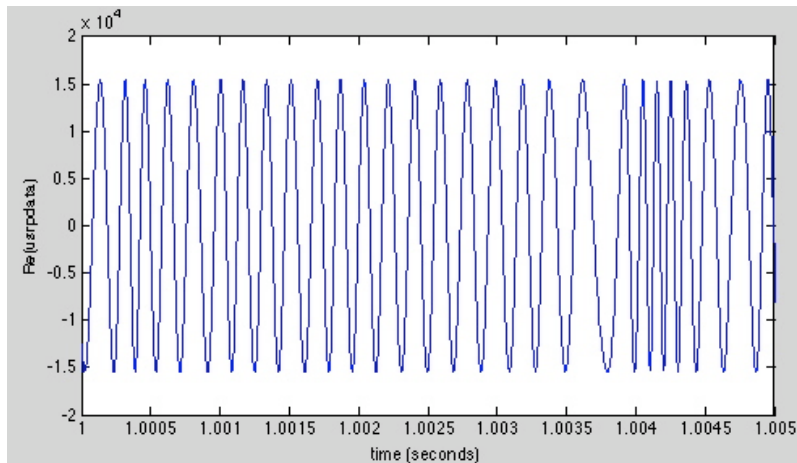
Using the following MATLAB code:

```
>> fs=256000;  
>> ts=1/fs;  
>> plot([1:1148928]*ts, real(usrpdata))
```

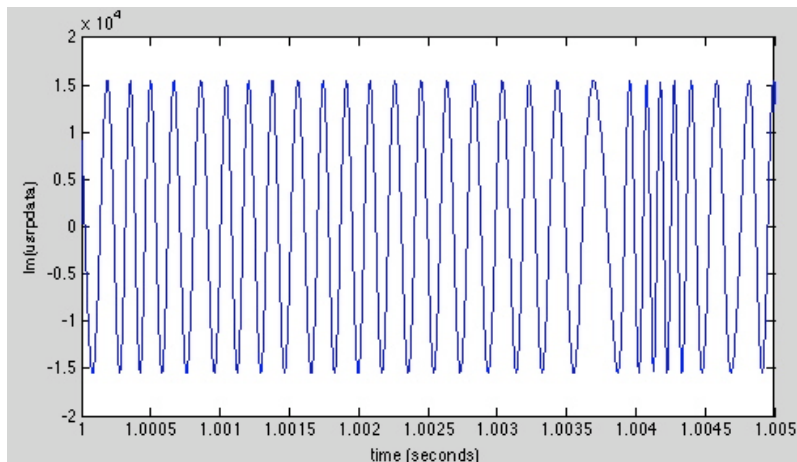
we observe the I component of the modulated FM waveform:



Expanding the time axis to observe a 5 mS portion:



Similarly the Q component:



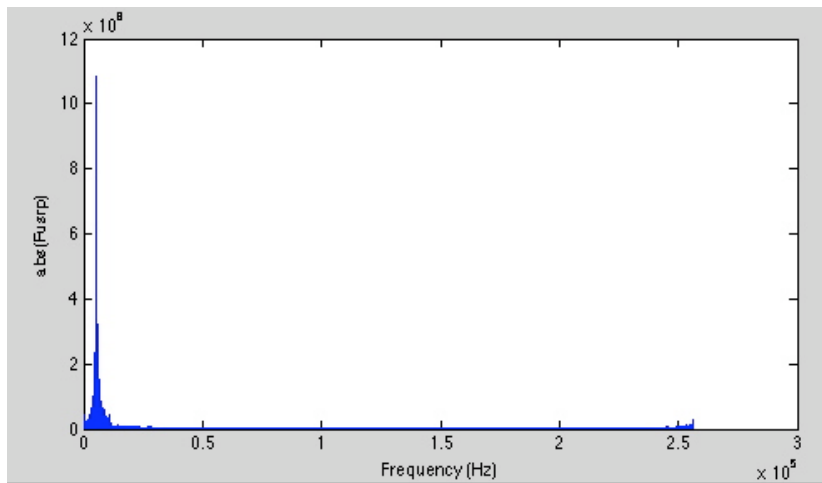
Note that the frequency of the signal with no modulation is approximately 5 KHz, which seems to be an error. This will be explained below.

Using the fft command in MATLAB to take the DFT, we have $N=1148928$ values with a foldover frequency at 128KHz. The frequency increment between values is $1/NT = 0.223$ Hz or after $N/2 = 574464$ samples we are at 128 KHz. The frequencies can be labeled as $n*(1/NT)$. If we find and plot the spectrum using the following MATLAB commands:

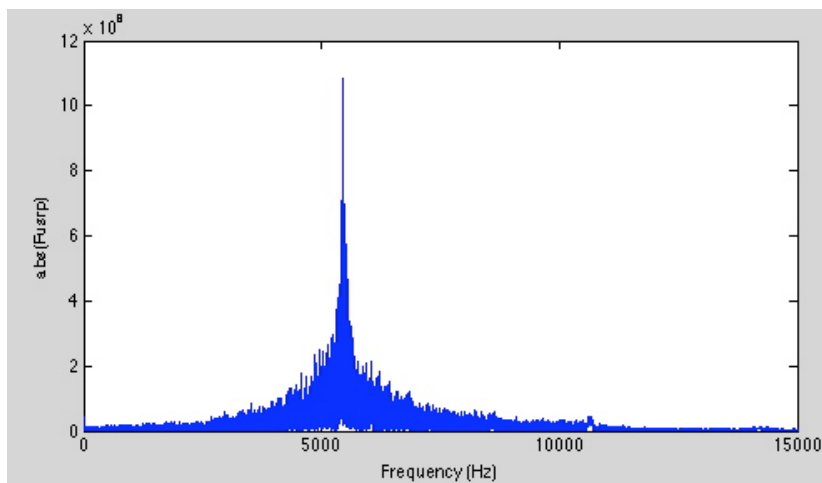
```
>> Fusrp=fft(usrpdata);  
>> N=1148928;  
>> deltaf=1/(N*ts);  
>> plot([1:N]*deltaf, abs(Fusrp))
```

```
>> plot([0:N-1]*deltaf, abs(Fusrp))
```

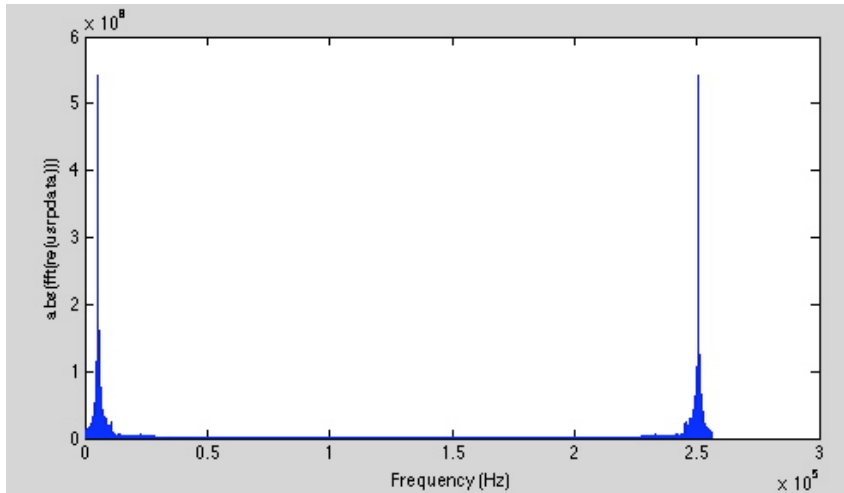
The following plot results:



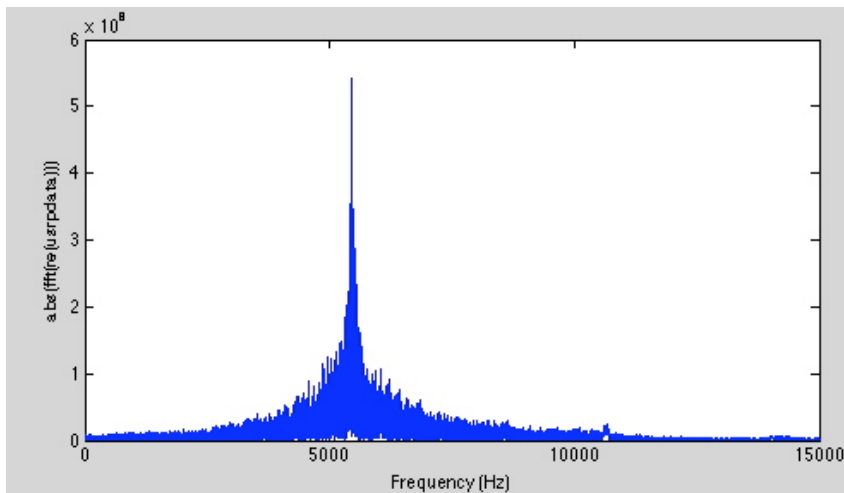
or expanding the frequency axis:



If we take the DFT of the real part of the usrp data, or the in phase component only, the result is:



OR:

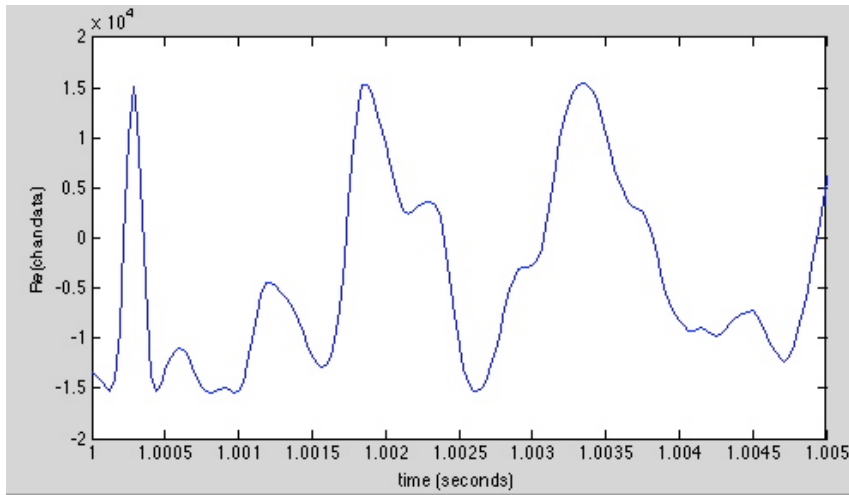


Again, we see the unexpected shift of 5 KHz. This is explained by the fact that the oscillator on the daughterboard is inaccurate. Measurements showed an approximately 5 KHz error on the 446 MHz oscillator frequency. Thus, the motherboard did not actually remove the entire carrier frequency component.

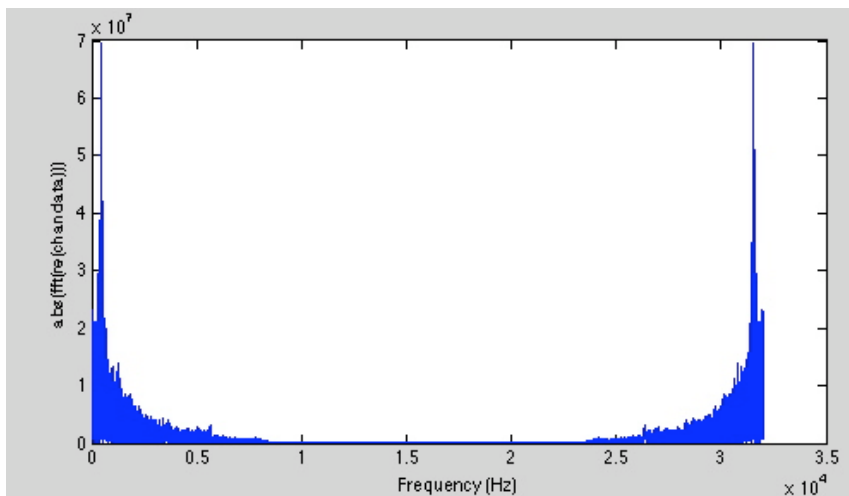
Output of the Channel Filter

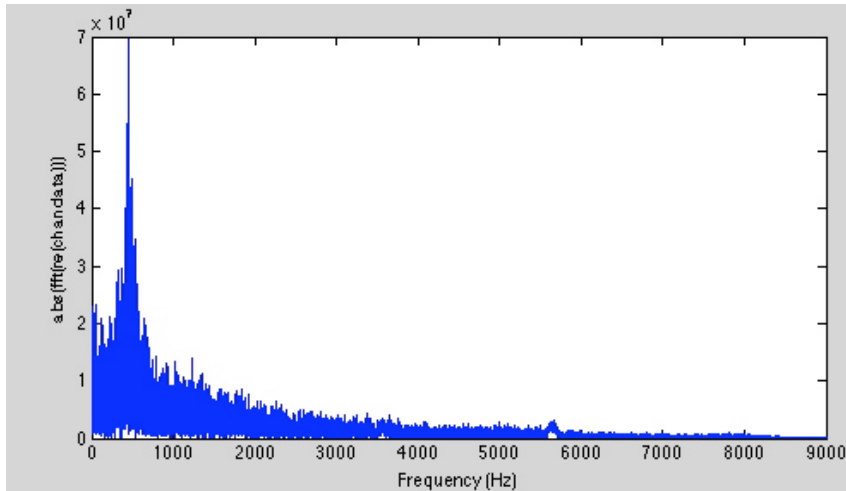
The output rate of the Channel Filter is 32Ks/s. Thus, the time between samples is $1/32K$ or .031 mS. The file chan.dat is converted to MATLAB readable form and placed into a matrix called chandata. This file contains 143616 complex values ($1148928/8$) and covers $(1/32K) * 143616 =$

4.488 seconds. This is the FM signal that has been filtered out from the remaining channels. If we plot the same segment as above, we only see 1/8th of the samples or:



Taking the DFT, we have $N=143616$ values with a foldover frequency at 16 KHz. The frequency increment between values is $1/NT = 0.223$ Hz or after $N/2 = 71808$ samples we are at 16 KHz. MATLAB produces the following spectra. Note that the DFT was taken on the real part of the data (or the I component). While the actual phase information of the components will not be complete this does provide information on the frequencies that are present. The Channel Filter included a frequency shift of 5 KHz to compensate for the frequency error in the daughterboard, thus, removing the 5 KHz component present at the output of the USRP. Note that the channel filter has successfully filtering out signals above 8 KHz.

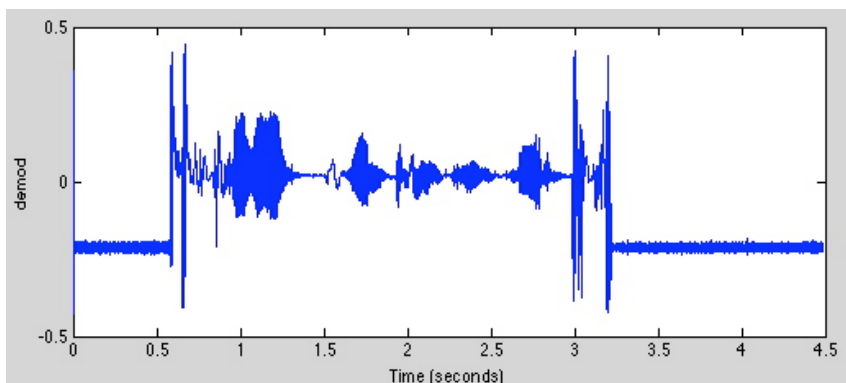


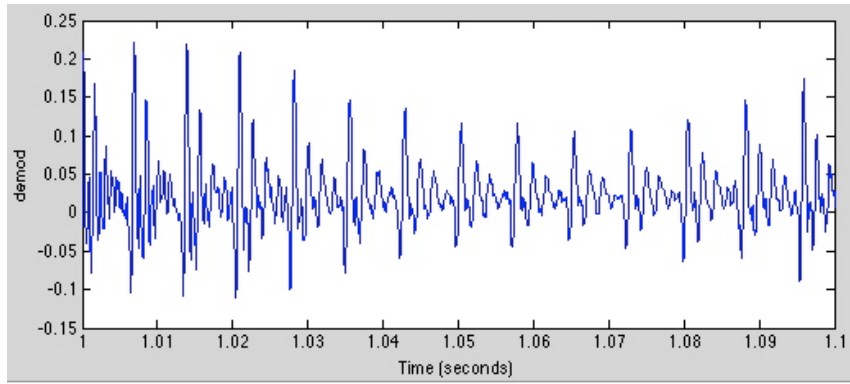


Note that this is still an FM signal.

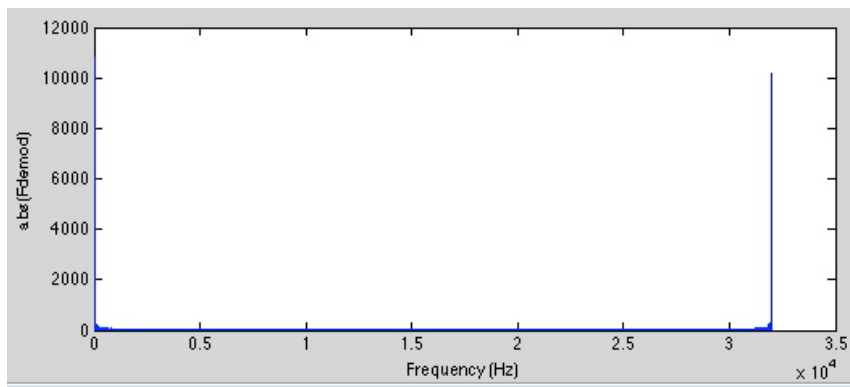
Output of the Demodulator

The output of the demodulator has a data rate of 32Ks/s. The time between samples is $1/32K$ or .031 mS. The values coming out of the demodulator are real, not complex. The demodata file contains 143616 real values since there is no further decimation of the signal in this block. It covers $(1/32K) * 143616 = 4.488$ seconds. A plot of the data produces the following graphs:

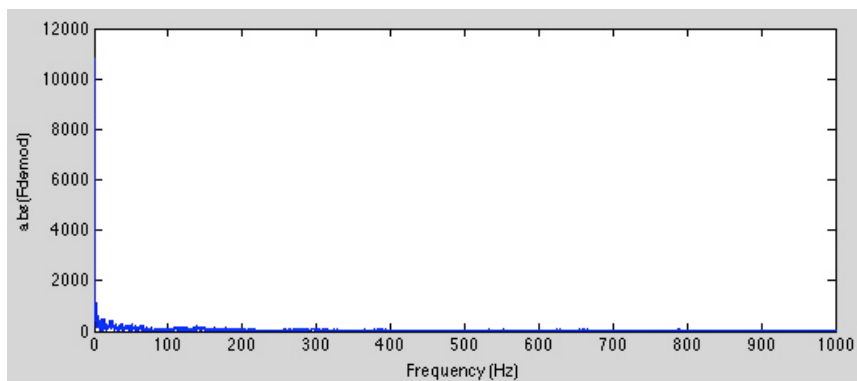


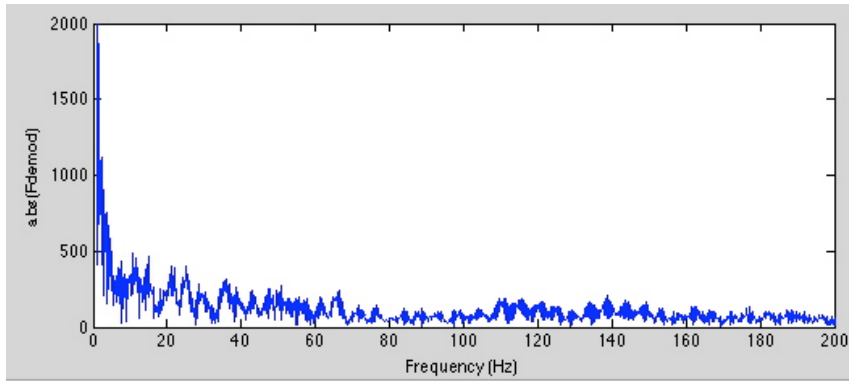


Taking the DFT of the data we get:



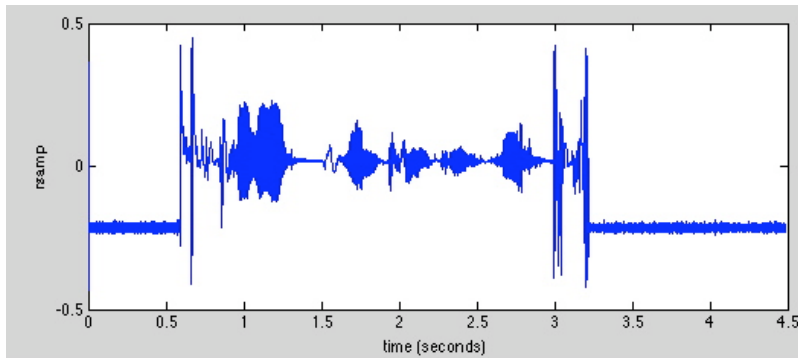
or:





Output of Resampler

The resampler converts the sampling rate from 32 Ks/s to 48 Ks/s to match the required sample rate of the laptop audio card. No additional filtering or processing is done in this block. The data file, rsamp, contains 215424 samples ($143616 * 1.5$), again covering 4.488 seconds. A MATLAB plot of this signal results in:



As expected this is the same waveform that was output from the demodulator. Taking the fft of rsamp results in:

