

A/D Converter

The signal obtained after the Signal Conditioning circuit is an analogic signal. Despite it could be possible to use analogic methods to detect the different frequencies; this could lead to an expensive and bulky device. The new digital signal processing tools allow to analyze digital signals with a few hardware and software resources.

This processing can be carried in a pc or embedded in a microprocessor IC. As it has been discussed before, in this case the processing is going to be carried by computer software. The computer needs to be fed with a digital signal, so it is necessary to convert the analogue signal to a digital one with an Analogue to Digital Converter. The IC ADS8506 from Texas Instrument (*See datasheet in the appendix*) has been chosen for its characteristics and competitive price.

12 bit Analog to Digital Converter	±0.45 LSB
5-V Single Analog Supply	28 pin SO package
40 kHz Min Sampling Rate	Full parallel interface
4V, 5V and ±10V Input Ranges	

Being interested in the ±5V input, the 12 bit Successive Approximation Register gives:

$$2^{12} = 4096 \text{ states.}$$
$$10\text{V}/4096 = 0.00244\text{V} \text{ -----} \rightarrow 2.44 \text{ mV resolution}$$
$$2.44 * 0.45 = 1 \text{ mV} \text{-----} \rightarrow 1\text{mV error}$$

Signal processing and storing of data.

Once the signal has been converted into digital, it can be 'acquired' by the processor of the pc. A software tool is going to be used for this. MATLAB. The data is read from the A/D converter, via the parallel port of the PC. A simple software samples this digital data, and stores it in a file. It is important here to discuss the sampling frequency of the data. The signal from the light source is believed to be flickering in frequencies from 0 to 500 Hz. The basic rule for being able to reproduce a signal from its samples determines that the sampling frequency has to be at least double of the highest frequency to be detected (Nyquist Frequency).

As this software is embedded in the microprocessor of a computer, and it is controlling other tasks at the same time, it is difficult to establish a cycle time for this process. Some decision have been taken based on experience. The data acquisition time has been adjusted to 1 second (+- 0.01), and, for getting proper measurements in different pc's and different conditions, the sampling frequency has been set to 4 times. Briefly, 2048 samples are going to be taken in one second (Fs=2000 Hz aprox.).

```
base = 888; %Initialises Bytronics board
keep_going = 1;
disp('Hello Bytronics')
while keep_going
    tic;%set internal clock running to time processing;
    for i=1:2048 %number of samples
        t(i)= i;
        adc = PPIInpval(base); %reads from the parallel port
        pc_value = (100.0/255.0)*(adc);
        value(i)= double(pc_value);
        for k = 1:50000; %delay between samples
            end;
        end;
    time_taken = toc
    fs = 2048/time_taken; %f=num of samples/time lasted
    plot(t,value);
    keep_going=input('Press 1 to continue; Press 0 to exit');
end
```

Shanon Theorem: A countinuous signal $x(t)$ with frequencies lower than F_{max} can be reconstructed exactly from its samples $x(n)=x(t)$ if the sampling frequency is $F_s > \text{or equal than } 2 * F_{max}$

Each of this samples is being stored in a file, agains the time. So, for each sample, we have the timing and the data of itself. Using this data, we can reproduce the signal in the time domain (value against time).

Fig 1: Program run in Matlab

Display of data.

Once the data has been sampled, it can be plotted. The most common way of displaying a wave is in the time domain, where the value is displayed against the time. Using the command plot from matlab we would have a view of the sampled signal. Matlab has also a Signal Processing Tool, named 'sptool.' With this tool it is possible to reconstruct a signal from the containing data in a file. This is an example of the signal first detected and sampled via our software in Matlab, and reconstructed by SPTOOL based in a data file

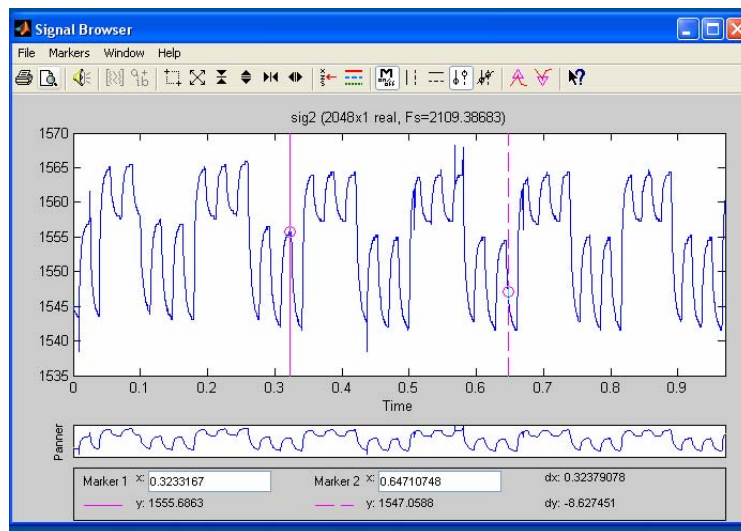


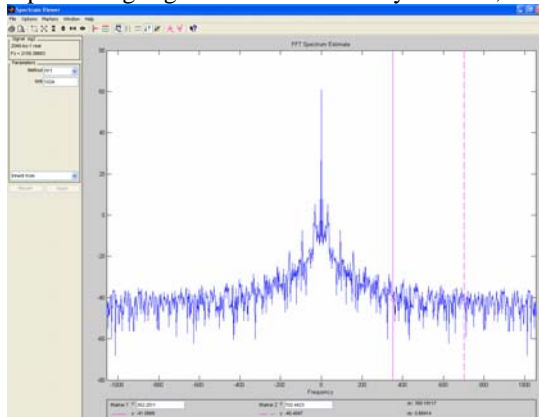
FIG 2: Row signal obtained from the sensor

A fast analysis of the picture indicates that this is a typical modulated signal. In this kind of signals, there are two different waves. Despite the objective of this report is not to analyze the characteristics of modulation or modulated signals, here are some details.

- The info, is being carried by a high frequency signal (aprox 30Hz)
- The carrier is a signal with a lower frequency, and in this case digital. (aprox 6Hz)

Frequency analysis

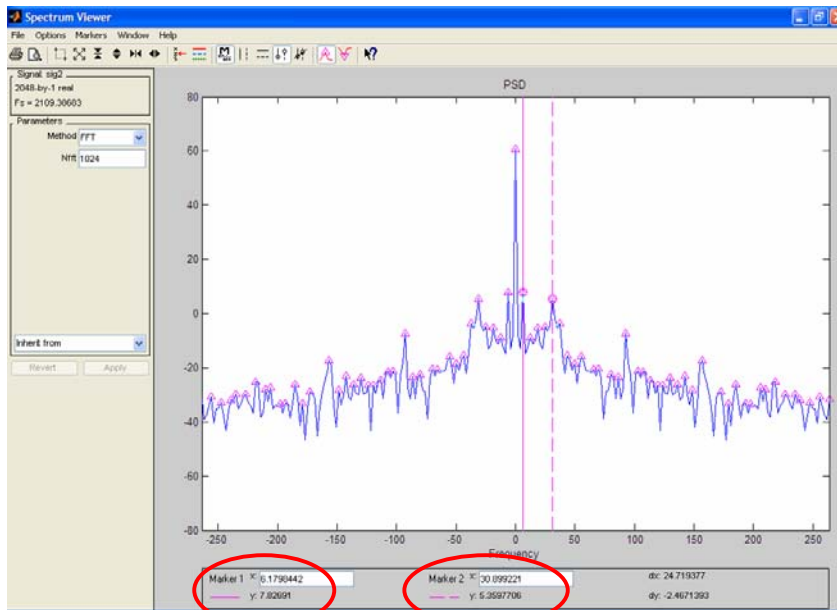
Once the signal has been obtained, it is desirable to do a frequency analysis, with the objective of identifying the different frequencies content in the signal. To do this, the before discussed SPTOOL has another function for determining the Fourier Transform. The Fourier transform is a mathematical tool for representing signals in the frequency domain, or spectrum of frequencies. The different signals can be



decomposed into sinus and cosinus functions, with determined frequency. The display of these frequencies is known as spectrum. SPTOOL and FFT (Fast Fourier Transform) provides the display of the spectrum in a quick processing. Once again, FFT takes samples of the signal. As the signal has already been sampled for the acquisition (2048 HZ), and has been stored in the file with the same timing, the FFT should be done at a frequency higher than this one (in this case 2100 Hz). See the picture of the spectrum.

This representation from 0 to $F_s/2$ also eliminates the aliasing, or repetition of the signals in higher frequencies

FIG 3: Spectrum of the signal (Frequ vs Amplitude) $F(\text{alias})=F(o)+-k*F_s$
 $K=[0,1,2,3,\dots]$



In this magnified representation, it is easy to identify the two signals

1. 6 Hz -> 7.82dB
2. 30Hz-> 5.36dB

The noisy dc signal goes up to 60dB; it is very noisy because its power is even greater than the power of our signals

Fig 4: Magnified vision of the spectrum

Filtering

From the spectrum representation (0 to $F_s/2$) we can see that there is a continuous signal $F_s=0$, and two main frequencies, as well as different noise sources. The relation between the desired signal, and the noise comes by a function SNRQ, and it is important to have as much power as possible in the signal, and reduce the power in the noise (Signal Noise Ratio)

$$\text{SNRQ}(\text{dB}) = 20 * \log_{10} (\text{Pot Signal} / \text{Pot Noise})$$

This noise can be added from different sources, like interferences with the computer hardware, noise in the sensor because of the dc supply for the IC's or the ac supply if it is connected to the net (50Hz or 60Hz noise). Another noise source can be the light. The photodiode not only detects the light from the machine, but also other ambient light like daylight or light from the illumination close to the machine. When one of this interference lights gets to the diode, we have also current flowing through, and an undesired signal.

It is impossible not to capture noise in any system, so filters must be used to reduce the influence of these noises. A good combination of analogue and digital filters will make the signal much more evident, and the frequencies we are interested in. One capacitor is being placed before the operational amplifier in the signal conditioning circuit, aiming to reduce the dc noise

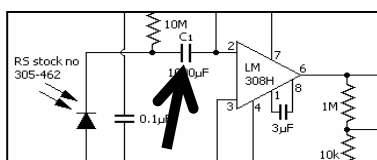


Fig 5: Capacitor to suppress dc noise from external light sources

There are other noises apart of the dc noise that should be reduced or eliminated. To do this, the SPTOOL is going to be used again. There different filters can be created (bandpass, bandstop, highpass and lowpass) for the discrete domain. A lowpass filter is the most basic that can be applied to this system. Only the frequencies up to 500 Hz are important, so, frequencies higher than this must be attenuated. Filter design has to be carefully done, because an excessive order or a very exigent filter can kill our important signals. A bandstop filter must be applied to stop the band between 55 and 65 HZ. This is the frequency of the main network, which means the frequency with what the lighting is working.

With this bandstop the noise of external illumination coming from bulbs, etc will be eliminated. Applying again FFT to the signal after being filtered gives something like this:

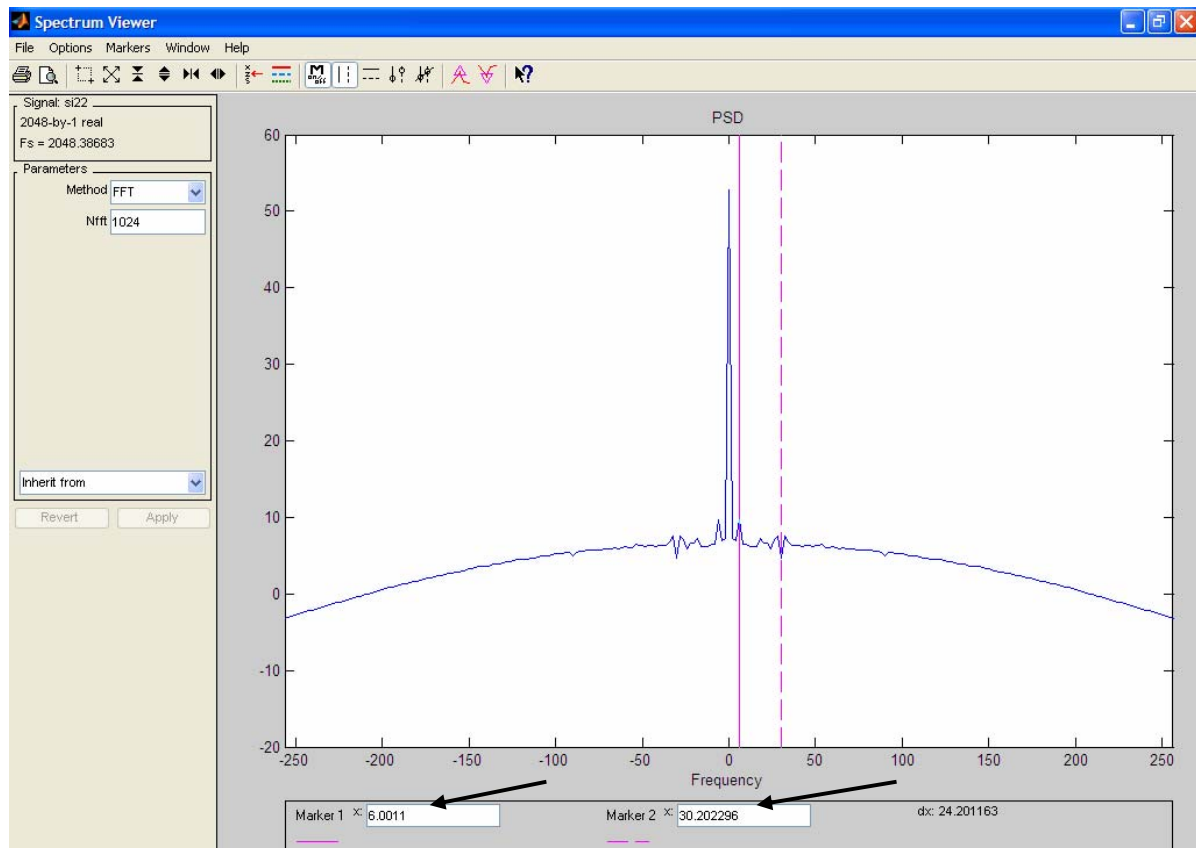


Fig 6: FFT of the filtered signal.

In this picture, the two frequencies can be identified easier, once the noise has been reduced. Still can be seen the dc signal, because this samples were taken with a different signal conditioning circuit, that didn't have the capacitor before the OpAmp.

Finally, the two main signals can be located at **6.00 Hz** and **30.20 Hz**, as it can be read from the markers positioning. From the screen shot can be read that the Sampling Frequency for the FFT is 2048Hz, and the points taken to evaluate it are 1024.

Conclusion

This is a typical Digital Signal Processing scenario, which can be applied to many other areas. In this case, a flickering light with a modulated signal has been analyzed, but the source can be extended beyond visual light, infra red, laser, microwaves, video and audio signals, etc. Nowadays all the information is sent in the form of signals, and DSP becomes more important. The same principles are being used when reading information from a hard disk, receiving and processing the Internet signal from the phone line, or the security and control system in a car that avoids the blocking of the wheel when braking.

Inside this aspect, is embedded the design and implementation of filters. As it has been discussed before, it is impossible to receive a pure signal. Wherever the scenario is, in day to day basis, there are many noise sources interfering with the desired information. The design of filters to discard this undesired noise is a very important discipline, even more than the post processing of the information.

Pablo
Cebrian
File

