Digital Real-time Audio Frequency Spectrum Analyzer Development for Audio Devices

G. Adamidis, T. N. Kapetanakis, and I. O. Vardiambasis

Broadband Communications & Electromagnetic Applications (BCEMA) Laboratory,
Department of Electronics
Technological Educational Institute (TEI) of Crete,
Chania Branch, Crete, Greece

Introduction (1/2)

- A digital, real-time audio frequency spectrum analyzer circuit for audio devices is presented in this work.
- This circuit could be useful to anyone who would like to embed it in an audio device or use it as a stand-alone unit.
- Spectrum analyzers are employed in most of the modern signal processing systems.
- Spectrum analyzers measuring the distribution of signal energy in frequency domain.
- An audio spectrum analyzer now is used for measurements in the audible frequency spectrum (from 0 to 20000 Hz about 10 octaves).

Introduction (2/2)

The proposed digital, real-time audio spectrum analyzer circuit for audio devices can be connected to any audio device, as it is :

- Accepting an analogue audio signal as input,
- Digitizing and processing the audio signal using a DSP,
- Computing the distribution of the audio signal energy to 20 specific frequency bands, and
- Displaying the energy distribution on a 20x20 LED display.

Frequency Domain Measurements

As is well known there are two measurement methods in the frequency domain:

- Vector Analysis and Spectrum Analysis.
- The measurements in the frequency domain that required complete information (frequency, amplitude, and phase) about the signal, called <u>vector signal analysis</u>.
- The other large group of measurements can be made without knowing the phase relationship among the sinusoidal components. This type of signal analysis is called <u>spectrum analysis</u>.

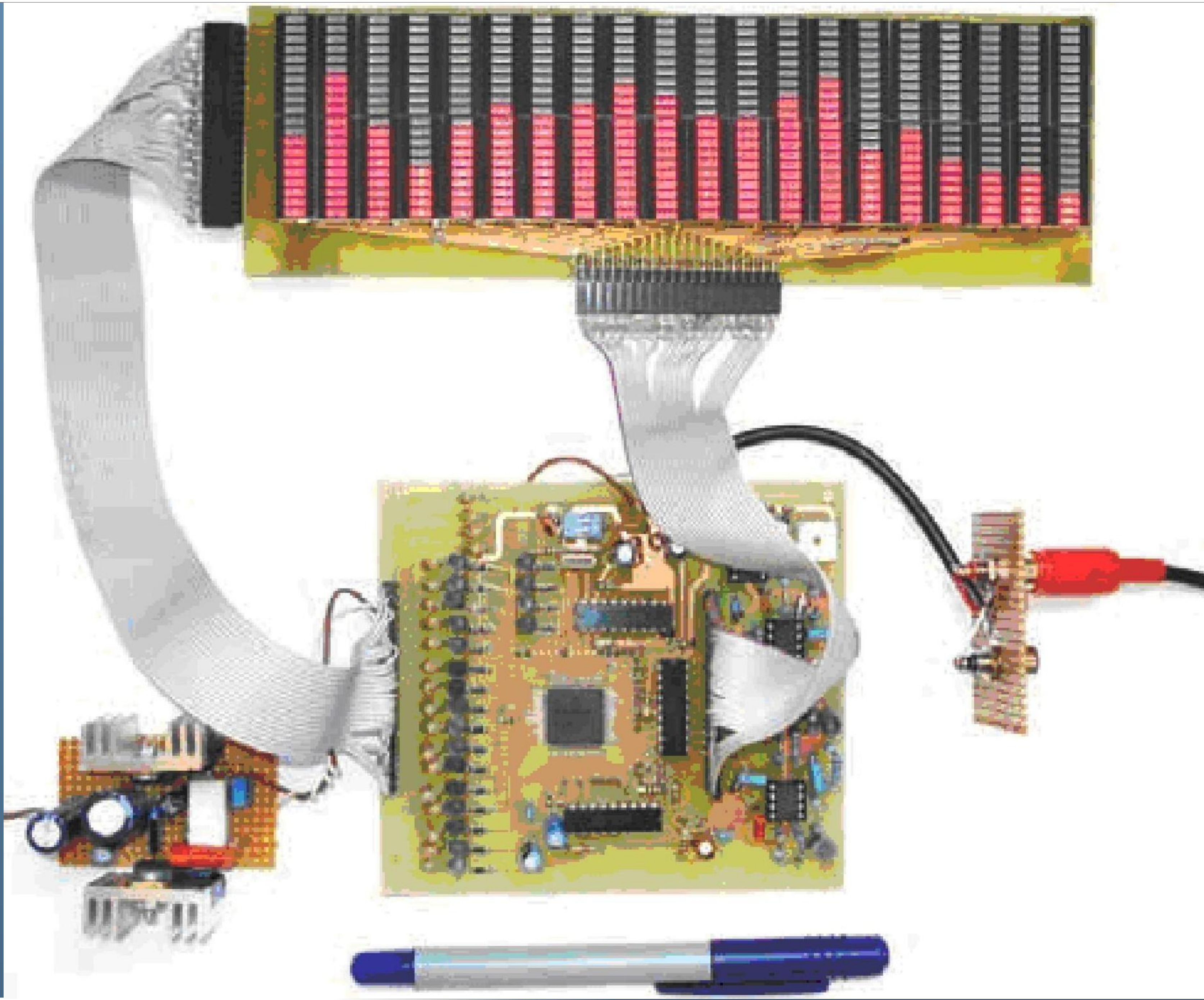
Is Our Design a Spectrum or a Vector Analyzer?

- Our device retrieves complete information about the signal frequency, amplitude, and phase and stores the information in a digital memory.
- Consequently is a vector analyzer ?
- However, for simplicity, it displays only the signal amplitude in the frequency domain.
- That’s why we call it a <u>spectrum analyzer</u>.

Building a Spectrum Analyzer

There are two main approaches in order to build a spectrum analyzer or a vector analyzer:

- the Sweep-Tuned Method, and
- the Fast Fourier Transform (FFT) method.
- For our project, we used the <u>FFT method</u>.
- As such, it digitizes the time domain signal and then uses DSP techniques to perform a Fourier Transform.



Hardware

In order to design our spectrum analyzer we needed:

- a Digital Signal Processor (DSP),
- an Analog to Digital Converter (ADC),
- a microcontroller,
- Operational amplifiers (Op Amps) and
- a Display Unit (DU).

Signal Processing Chip

We have used Microchip Technology's dsPIC 16-bit digital signal controllers.

- The type of this powerful silicon device is both a DSP and a high-performance 16-bit microcontroller in one package.
- So we realized that a dsPIC would be a nice match for our low-cost design and we chose dsPIC30F6012A because it has enough I/O pins and internal data memory.
- dsPIC30F6012A from among the many dsPICs (it has enough I/O pins and internal data memory).
- In addition, dsPIC30F6012A contains a 16-channel, 12-bit internal ADC that further simplifies the design.

Display Selection

The first problem we have faced during the design process was the display. We had to choose between:

- LED Display (low resolution),
- LCD Module (high resolution), and
- Graphics Display (high resolution).

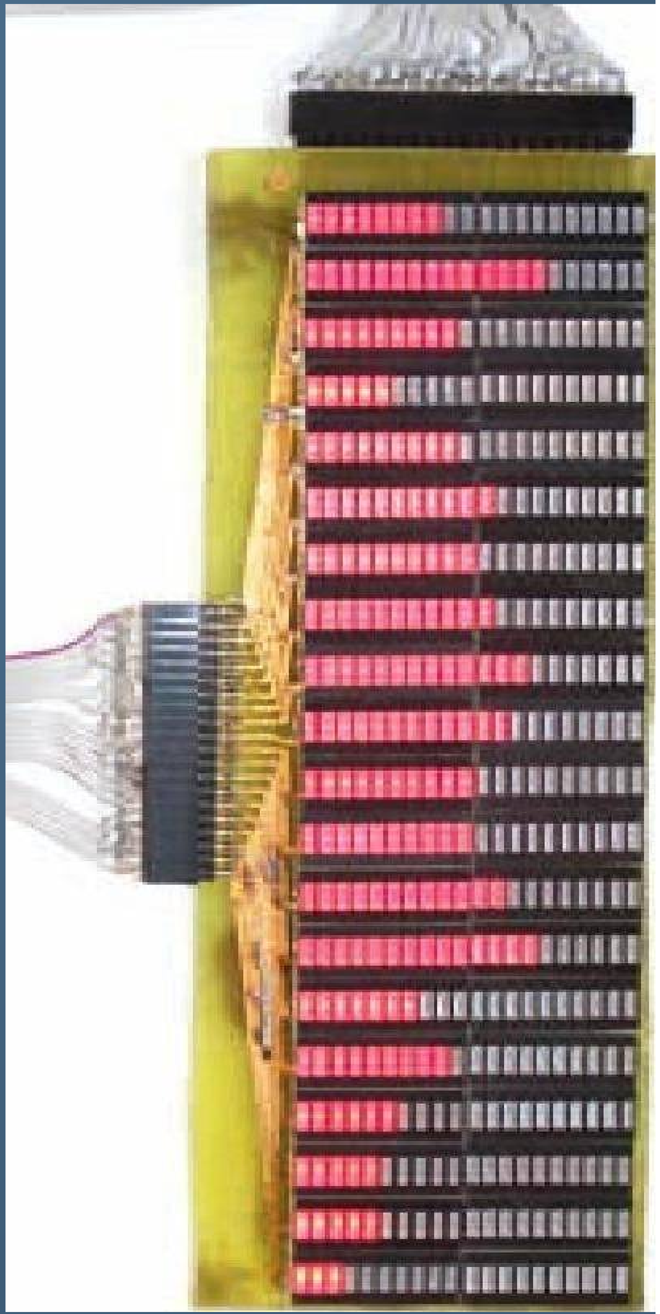
Finally, we chose a LED Display

LED Display Unit

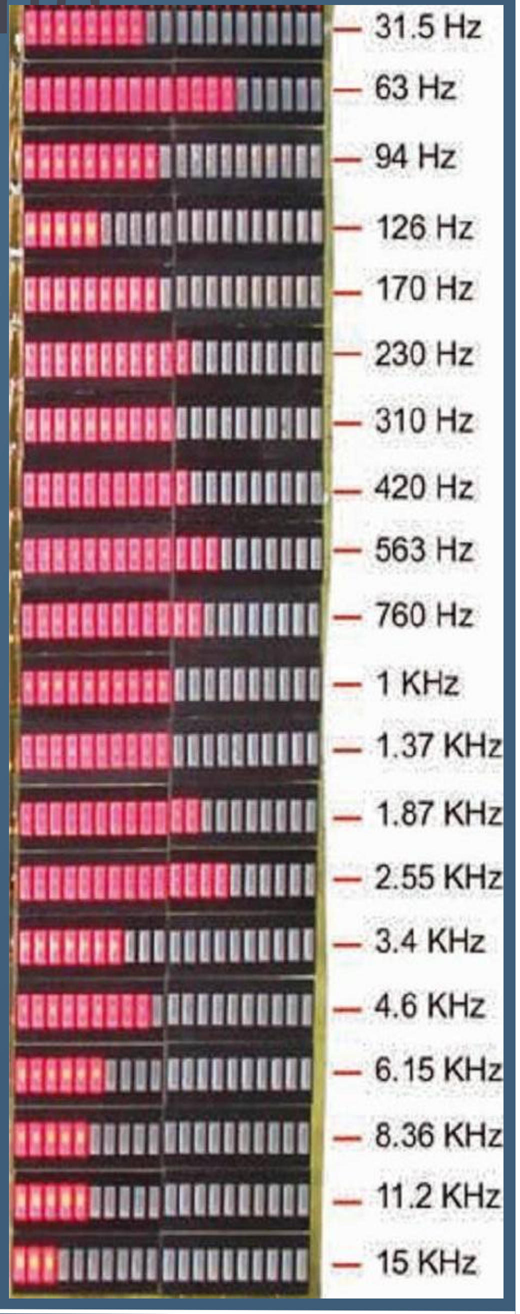
The display has 40 LED bar graphs with a viewing area of about 20.3 cm × 5.1 cm.

In this way we have achieved:

- hardware simplicity,
- software simplicity,
- speed improvement, and
- cost reduction.



Frequency Spectrum Splitting



Our device has 20 frequency bands with central frequencies given by (1) and max frequency resolution 31 Hz.

$$CF(i) = CF(0) \times 2^{i \times a} \qquad (1)$$

where i is an integer $0 \le i \le 19$

The figure on the right presents the display and the aforementioned central frequencies.

FFT Length and ADC's Sampling Frequency

The FFT's length and the ADC's sampling frequency are two crucial parameters in order to perform an efficient spectrum analysis.

It is well known that:

- The frequency resolution depends on the FFT's length and the sampling frequency.
- For an N-point FFT, at a sampling rate SR, the frequency resolution df is:

$$df = \frac{\text{Sampling Rate}}{N}$$

Additionally,

- The sampling rate also must comply with the Nyquist sampling theorem

$$SR > 2 \times f_{MAX}$$

FFT Length and ADC's Sampling Frequency

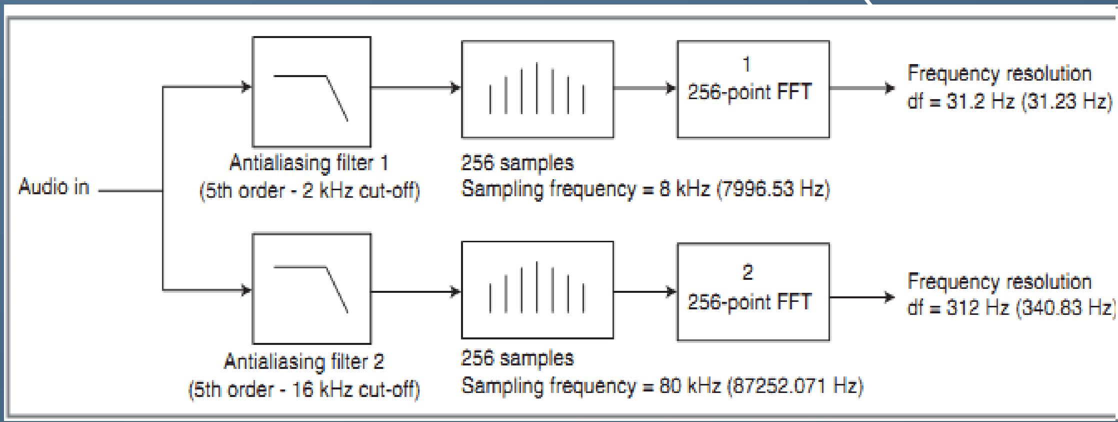
In order to avoid complex anti-aliasing filters (filters of great order)

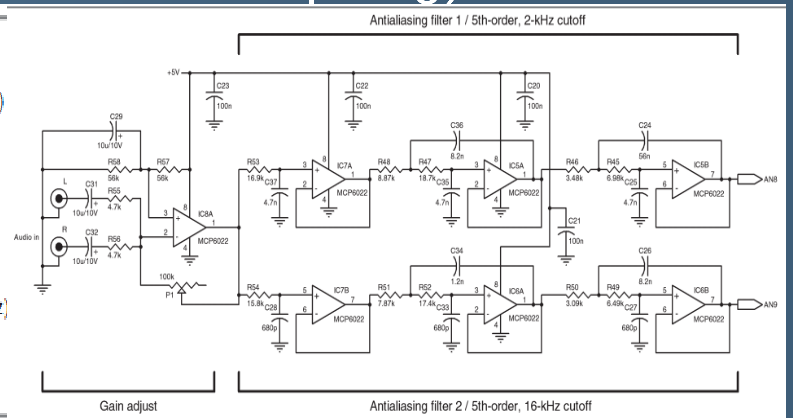
- We chose a sampling frequency much greater than the Nyquist limit, over-sample at 80 kHz and,
- In order to achieve the required 31 Hz resolution N should be at least 4096.

But a 4096-point FFT cannot be implemented using our dsPIC in real time (due to memory, speed, and fractional nature restrictions).

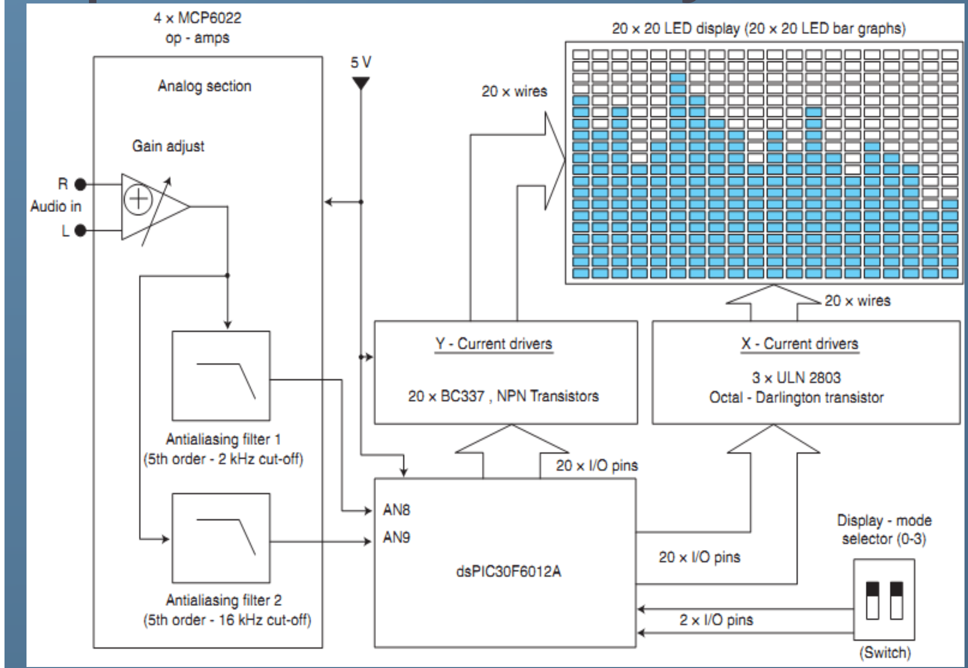
And Now What?

- We used <u>two sampling rates</u> (8 kHz and 80 kHz), two <u>anti-aliasing filters</u>, and <u>two 256-point FFTs</u> because there is no need for the 31-Hz resolution at frequencies above 126 Hz.
- The 1st FFT can provide the resolution needed for the low frequencies, while 2nd FFT provides a very good resolution for frequencies above 1 kHz.
- The filters were designed for a signal-to-noise ratio (SNR) better than the dynamic range (DR) of the spectrum analyzer which is 30 dB. Are quite simple and straightforward because the required filter order is limited to five (due to the over sampling)





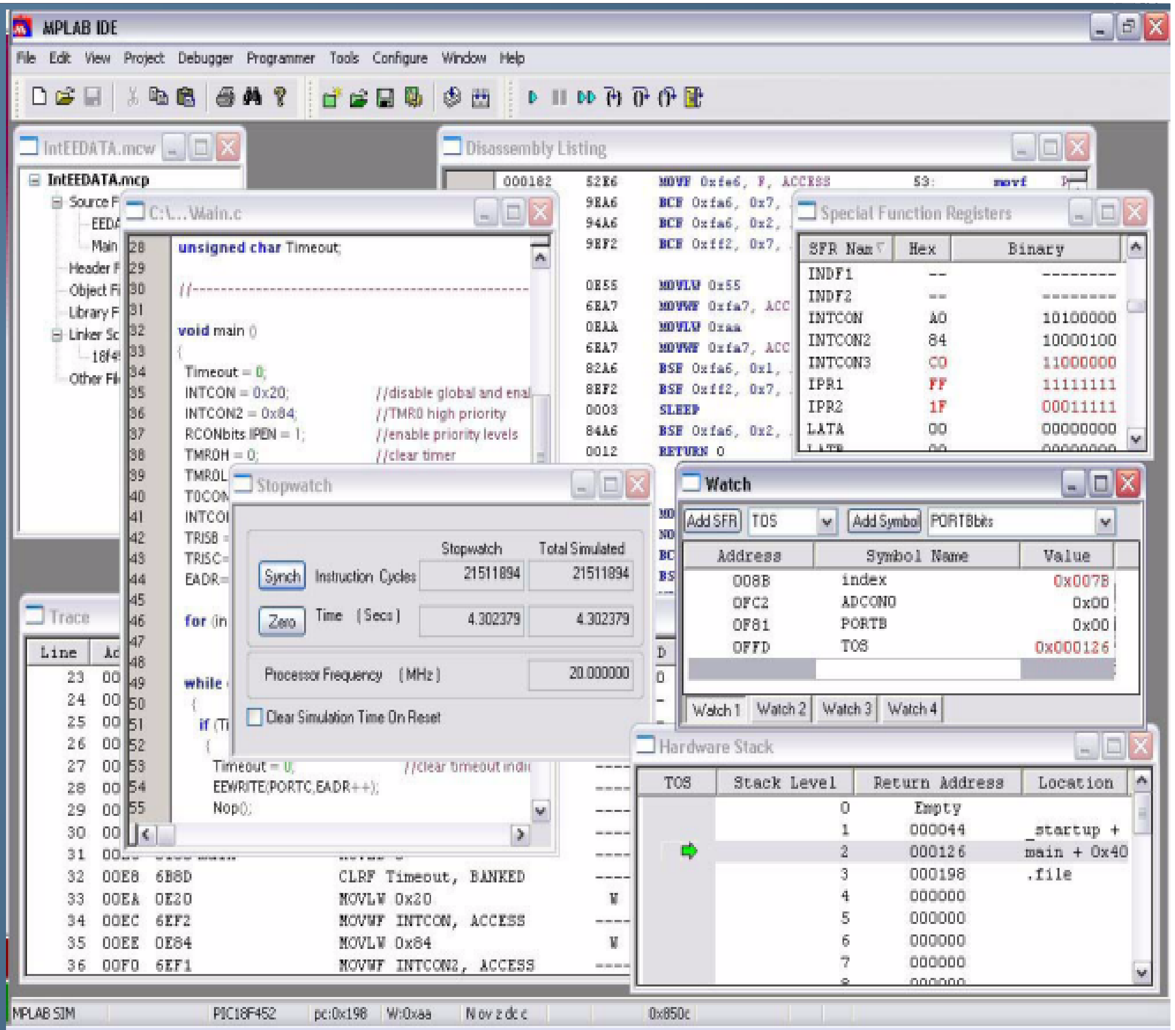
Block Diagram of the Digital Audio Spectrum Analyzer



- Current < 600 mA, because are not powered simultaneously, but are cycled using scanning technique.
- In this figure we can see the complete design of the Digital Audio Spectrum Analyzer as a block diagram.
- Should be noticed that The dsPIC (dsPIC30F6012A) can do both DSP and displaying in real time, but it cannot drive the LEDs with the current needed.
- So current drivers should be used in order to provide the required current to the LED display.

Software

- Software is written in C with Microchip's MPLAB C30 compiler and routines from the compiler's DSP library.



- Additionally implements a frequency analysis technique, which we call a "20-band parallel analysis filter algorithm using FFT."

20-Band Parallel Analysis Filter Algorithm Using FFT (1/3)

- Consider the case when a single tone S, of unity magnitude (0 dB) at frequency f, is applied to the input of the device. This tone will be sampled at a sampling rate SR. This will result in a sampling sequence S(n, f, SR):

$$S(n, f, SR) = \sin\left(2\pi f \frac{n}{SR}\right),$$
$$n = 0, 1, 2, 3, \dots, \pi = 3.14 \dots$$

- In addition, consider a case where you store the first 256 samples of this sequence in memory and apply to them a 256-point, normalized, 0.5 Hamming window w(n). This results in Y(n, f, SR):

$$w(n) = 0.5\left[0.54 - 0.46 \times \cos\left(2\pi \frac{n}{255}\right)\right],$$
$$n = 0, 1, 2, \dots 255$$

$$Y(n, f, SR) = W(n) \times S(n, f, SR)$$

20-Band Parallel Analysis Filter Algorithm Using FFT (2/3)

- It is necessary to use a window function in order to ensure flat-top filter behavior in the system, according to DSP theory. Afterwards, we can apply a 256-point FFT (which is actually the digital version of Fourier's algorithm) to Y(n, f, SR), resulting in X(n,f,SR):

$$X(n, f, SR) = FFT\left[Y(n, f, SR)\right] = \frac{1}{256} \times \sum_{k=0}^{255} Y(k, f, SR)\, e^{i \times 2\pi n \times \frac{k}{256}},$$

where i is the imaginary unit. $i = \sqrt{-1}$ and $e = 2.7182 \ldots$

- Notice that S(n, f, SR), w(n), and Y(n,f,SR) are real signals, while X(n, f, SR) is a complex signal containing real (Re) and imaginary (Im) parts. Computing the square magnitude of X(n,f,SR) we take P(n,f,SR), which is the discrete digital representation of the signal energy (or power) in the frequency domain.

$$P(n, f, SR) = \left\{Re\left[X(n, f, SR)\right]\right\}^2 + \left\{Im\left[X(n, f, SR)\right]\right\}^2$$

20-Band Parallel Analysis Filter Algorithm Using FFT (3/3)

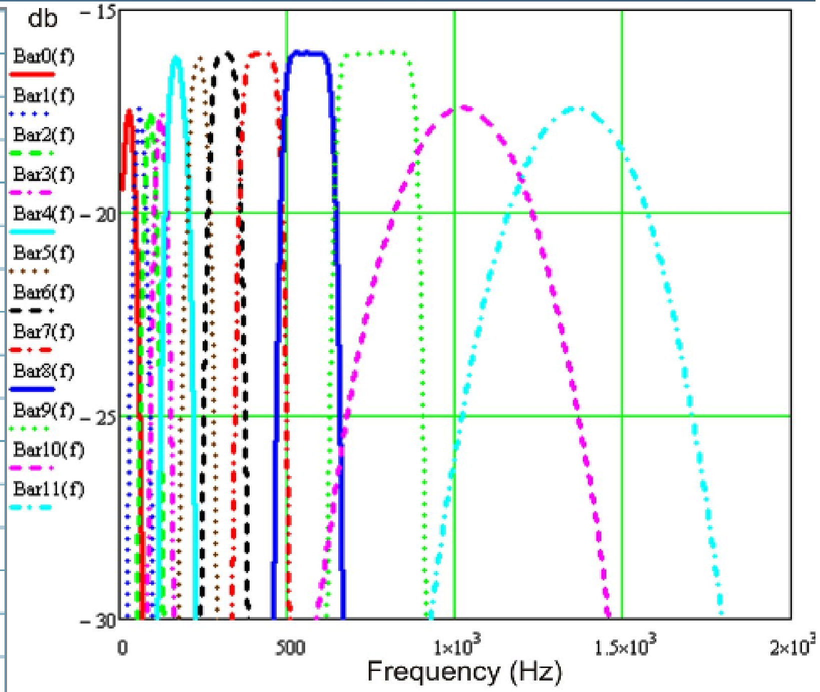
- Next, you define $Bar_0(f)$, $Bar_1(f)$,...,$Bar_{19}(f)$, from $P(n,f,SR)$:

$$\mathrm{Bar}\,i(f) = \sum_{k=u}^{v} P(k, f, SRj)$$

- j, u, v are defined in the Table according to i.
- SR1 (7996.53 Hz) and SR2 (87252.071 Hz) are the sampling frequencies For example:

$$\mathrm{Bar}\,19(f) = \sum_{k=38}^{50} P(k, f, SR2)$$

i	j	u	v
0	1	1	1
1	1	2	2
2	1	3	3
3	1	4	4
4	1	5	6
5	1	7	7
6	1	9	11
7	1	12	15
8	1	16	20
9	1	21	28
10	2	3	3
11	2	4	4
12	2	5	6
13	2	7	8
14	2	9	11
15	2	12	15
16	2	16	20
17	2	21	28
18	2	29	37
19	2	38	50



Bar0(f), Bar1(f), ..., Bar19(f) are 20 bandpass filters, each one having a central frequency CF(i). The 20 filters are overlapping at half-power points (–3 dB below peak). Filters 4–9 and 12–19 are flat-top at –16 dB. Filters 0, 1, 2, 3, 10, and 11 have 1 dB more loss than the others.

Coding

- Step 1: Acquire 256 samples of the audio signal at SR1 = 7996.53 Hz sampling rate.
- Step 2: Multiply by a 256-point Hamming window.
- Step 3: Compute the 256-point FFT of the resulting vector.
- Step 4: Compute the square magnitude vector P with the resulting vector X.
- Step 5: From vector P, compute Bar0, Bar1, Bar2, Bar3, Bar4, Bar5, Bar6, Bar7, Bar8, and Bar9 samples.
- Step 6: Plot each Bar_i on the ith LED bar graph.

Conclusions

The main advantages of the proposed circuit are the following:

- It has low-cost and can be embedded in any audio device,
- It uses a simple, low cost 20x20 LED display,
- It supports four different display modes,
- It provides very good frequency and amplitude resolution (0.431 octaves from 31 to 15,000Hz and 1.3 db from 0 to –28db),
- It can be powered from a single 5V power supply,
- It offers simultaneous monitoring of the entire audio signal band in real time,
- It uses a single processor for both DSP and display controlling.

References

[1] Agilent Technologies, "Spectrum analysis basics", Application Note 150, 2006.

[2] Microchip Technology, "FilterLab active filter software design tool", Product Overview, 2005.

[3] Microchip Technology, "16-Bit language tools libraries", DS51456C, 2005.

[4] L.Di Jasio, *Programming 16-bit Microcontrollers in C*, Newnes, 2007.

[5] M.H. Hayes, *Schaum's Outline of Theory and Problems of Digital Signal Processing*, McGraw-Hill, 1999.

[6] Microchip Technology, "dsPIC30F6011A/6012A/6013A/ 6014A data sheet", DS70143C, 2006.

[7] "MCP6021/1R/2/3/4 data sheet", DS21685C, 2006.

[8] "Reading and using Fast Fourier Transforms (FFT)", AN681, DS000681A, 2001.

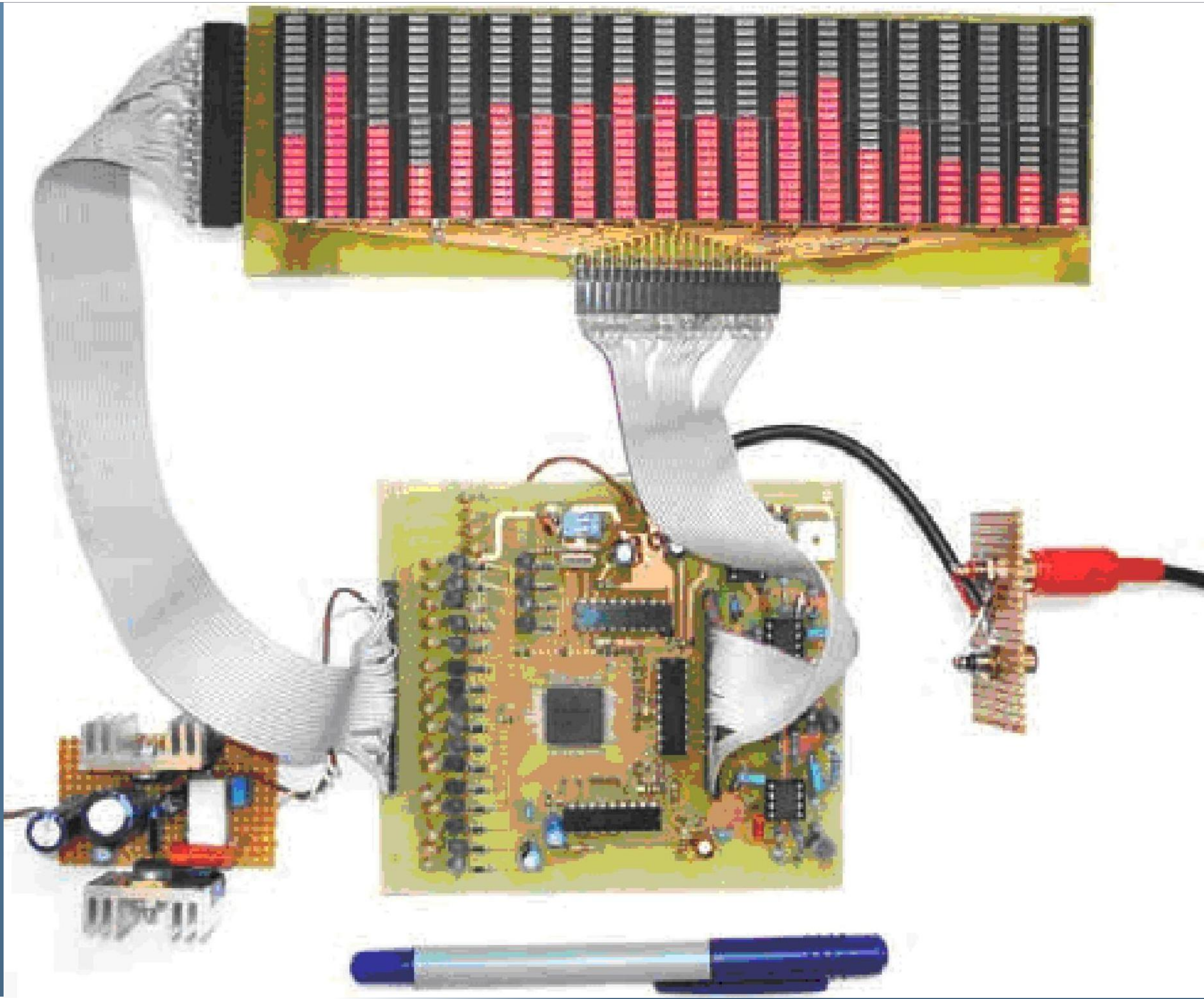
[9] Motorola, "ULN2803/ULN2804: Octal High Voltage, High Current Darlington Transistor Arrays", 1996.

[10] S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd Edition, California Technical Publishing, 1999.

[11] Microchip Technology Inc., dsPIC30F6012A DSC, MCP6022 dual op-amp, and MPLAB ICD 2, www.microchip.com.

[12] Motorola Inc., ULN2803 Darlington transistor array, www.motorola.com.

[13] Parametric Technology Corp., Mathcad engineering calculation software.



Digital Real-time Audio Frequency Spectrum Analyzer Development for Audio Devices

G. Adamidis, T. N. Kapetanakis, and I. O. Vardiambasis

Thank you !!!

Broadband Communications & Electromagnetic Applications (BCEMA) Laboratory,
Department of Electronics
Technological Educational Institute (TEI) of Crete,
Chania Branch, Crete, Greece