

WEB BASED APPLICATION FOR ELECTRONIC CIRCUITS SIMULATION AND SIGNALS ANALYSIS

Tiliute Doru Eugen

“Stefan cel Mare” University of Suceava, str. Universitatii nr. 9, 720225, Suceava, Romania

Abstract. The aim of this paper is to present an application destined for electronic circuits simulation and for signals spectra analysis. The particularity of this application is that it is conceived to use the Web technologies which involve some benefits: it doesn't require any special software on the client side, any recent browser being suitable; users can access it from any computer in the world which has an Internet connection; the graphics are generated “on fly” using the results that specialized programs return. These applications software are running on a web server which can grant public access or restricted access for members or a limited functions pack. Using open-source software for HTTP server and CGI interfaces the effective costs of system becomes very attractive. The application is suitable for the studies of electronic circuits in schools and for research in the field as well. It is possible to verify the behavior of some circuits, in a wide range of conditions, from any Internet connected computer without needing any specialized software.

Keywords. Web, circuits, analysis, simulation, Fourier

INTRODUCTION

Day by day, the Internet becomes more and more a part of our life. From entertainment to the latest news, from e-mail to advertising and from e-commerce to e-learning, the Internet confirms to be a very useful instrument for the human modern life. This has become possible due to the development of the Internet infrastructure, on the one hand and inventing and developing protocols and programming languages for Internet, on the other hand. Among the programming languages, PHP, which is actually a scripting language, has reached a large popularity due to the simplicity and its powerful functions. The latest versions of PHP (version 5 is the newest at the moment of editing this work), include a graphic library, which allows the creation and manipulation of various images in JPEG, GIF and PNG formats. Thus, it is possible to render, on a web page, images that don't exist, actually, anywhere. This is a good point in generating dynamic-content Web pages, which should look different, according to the client's options. So, it is possible to display the graphics of a mathematical function according to the input data supplied by client.

Also it is possible to generate charts, e.g. as those created by MS Excel, using data stored in a database or supplied by the client through a form [1], fig. 1-b.

Without getting into details about using the PHP graphic functions [2], we present in the table 1 the most used ones in our application - among over 100 available functions - grouped by category.

Table 1. Most important graphic function used in the present application

No.	Category	Function	Description
1	creating, loading and saving images	imagecreate (x_size, y_size)	Returns a resource representing an image in memory. The resulting image has a size of <i>x_size</i> pixels by <i>y_size</i> pixels.
2		<ul style="list-style-type: none"> • imagecreatefromgif (filename) • imagecreatefromjpeg (filename) • imagecreatefrompng (filename) 	Return an image resource that is equivalent to the pixel size of filename on disk. Each function is associated with the corresponding image file type.
3		<ul style="list-style-type: none"> • imagegif (img_resource) • imagejpeg (img_resource) • imagepng (img_resource) 	Send an image corresponding to <i>img_resource</i> to the browser. Each function is associated with the corresponding image file type.
4	drawing	imagesetpixel (img_resource, x, y, col)	Draws a pixel in the position <i>X, Y</i> in color <i>col</i> .
5		imageline (resource, x1, y1, x2, y2, col)	Draws a line in a graphic resource image from point (<i>x1, y1</i>) to (<i>x2, y2</i>) in color <i>col</i>
6		imagerectangle (resource, x1, y1, x2, y2, col)	Draws outlined rectangles. The top-left corner of the rectangle is (<i>x1, y1</i>) and the bottom right is (<i>x2, y2</i>) in color <i>col</i>
7		imagepolygon (resource, points, num_points, col)	Draws an outlined polygon specified by coordinates <i>points</i> and the points number <i>num_points</i> in color <i>col</i>
8	filling	imagefilledrectangle (resource, x1, y1, x2, y2, col)	Fills a rectangle specified by points (<i>x1, y1</i>) and (<i>x2, y2</i>) with color <i>col</i> , in a graphic resource <i>resource</i>
9		imagefilledpolygon (resource, points, num_points, col)	Fills a polygon specified by coordinates <i>points</i> and the points number <i>num_points</i> in color <i>col</i> , in a graphic resource <i>resource</i>
10	color allocation	imagecolorallocate (resource, R, G, B)	Returns a color id for the image in resource <i>resource</i> . The color is specified by 3 bytes, <i>R, G, B</i> , in decimal representation (0 to 255).
11	text	imagestring (resource, font_size, x, y, string, col)	Draws the string <i>string</i> in the resource image <i>resource</i> , with font having the size <i>font_size</i> and color <i>col</i> . The string begins in the position <i>x, y</i> .

There are over 100 image functions available in PHP, which make it very powerful for image handling. Detailed information can be found on Web resources [2], [3].

A domain where the graphic functions fit very well is that of electronic circuits' simulation one. Such an application is presented bellow.

APPLICATION DESCRIPTION

The application consists in two main PHP modules: first module is dedicated to electronic circuits analysis since the second for the output signal spectra analysis. First module supplies the pages requested by client, compute the circuit responses (e.g. AC, DC and transient), generate and send the graphic to the client. The main process phases are explained in the figure 1.

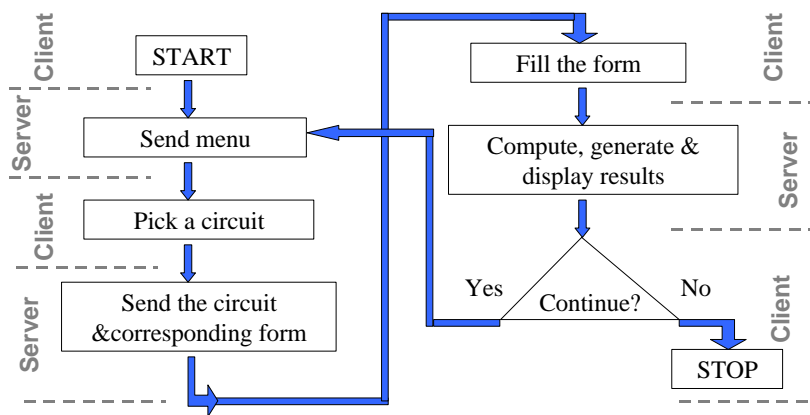


Figure 1. Functioning diagram of circuit analysis module

When the client connects to the application, a menu containing some basic circuit topologies is send to the client, figure 2.

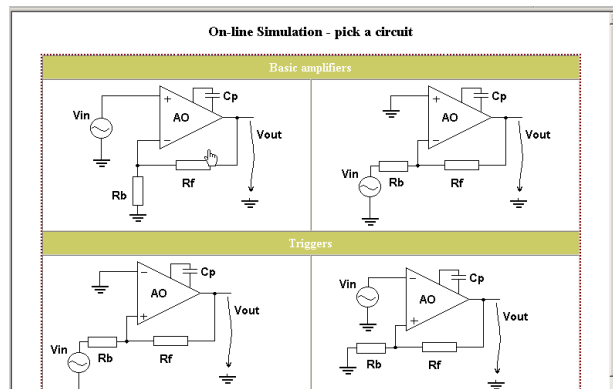


Figure 2. A menu with available circuit topologies

In the next steps the client chooses the desired circuit and send his option to the server, which replays with the corresponding form to be filled with the component values, and some test conditions as supply voltages, signal source parameters, etc. All these parameters are sent to the PHP script, which will process data, using the GET method. When completed, the script sends to the browser a new empty form, giving the client the possibility to change the conditions, the graphic

and a table with the summary of conditions the circuit behavior was tested in. In order to include the generated graphic in the HTML page containing the results, the image is inserted by setting the argument of *src* attribute with the URL of the script that effectively generates the image, as follows:

```
<img src='raspuns.php?Vc=$_GET[Vc]&f=$_GET[f]&Vi=$_GET[Vi]&Rb= $_GET[Rb]&Rf=$_GET[Rf]' border='1' style='position: absolute; left:450; top:60'>
```

Note that the data from the form are passed to the script that generates the image using the GET method as well.

In the present application, pressing the “Simulate” button in the form, the data are send to the same script that generates the form (the script appeals to itself); the data being set, the new form includes the generated image based on these data and the summary mentioned above. The result is shown in figure. 3.

The scripts that perform these operations are given in reference [4].

The second module uses to compute and display the spectra components of the output signal. Initially conceived as a stand-alone application, the module can be easily included in one general application due to its capability to read the input data from a file. For this the firs module described above must save the samples of the outputs signal into a file or a database. The module uses Forward Discrete Fourier Transformation (FDFT) in order to compute spectra components. The formula of FDFT is given in eqn.1

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n)e^{-j2\pi kn/N} \quad (1)$$

where $n, k = 0 \dots N-1$, N being the number of samples. $F(k)$ is the k^{th} component of spectra since $f(k)$ is the k^{th} sample of time-domain signal.

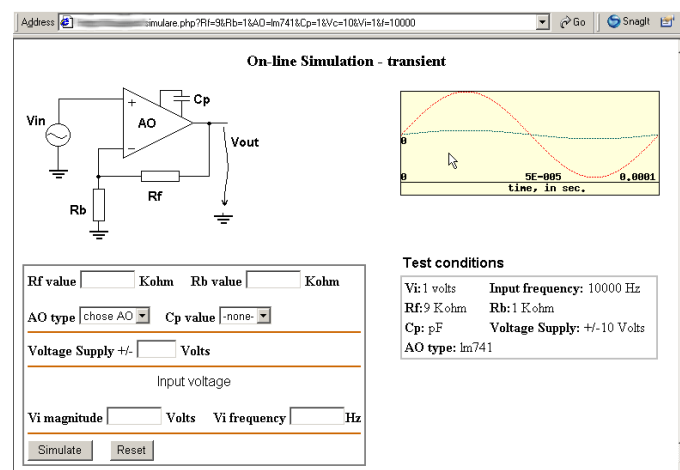


Figure 3. The result of circuit simulation; transient response
Generally $F(k)$ is a complex number characterized by a magnitude and a phase (2):

$$F(k) = \text{Re}(F(k)) + j \text{Im}(F(k)) = \sqrt{(\text{Re}(F(k)))^2 + (\text{Im}(F(k)))^2} e^{j\phi} \quad (2)$$

The values of the N samples of the output signal are read from the file (or database) and processed using the Fast Fourier Transformation (FFT). The PHP script implementing the FFT algorithm is given in reference [5]. The module can also gather data from forms or can compute itself the time-domain samples for a given pre-defined functions or even user functions. The steps followed in order to display the signal spectra are given in figure 4.

The task of computing and image displaying is shared between two scripts. First, entitled “fourier.php” is responsible for reading data from different sources (file, database, HTML forms, etc.), computing the samples if data are given in analytic form, computing the magnitude and the phase of each spectra component, prepare the results to be processed by the next script and passes them to it. The second script, entitled “display.php” takes over the parameters transferred by the first script, generates the image and send it to the browser, in the same mode as described for the first module.

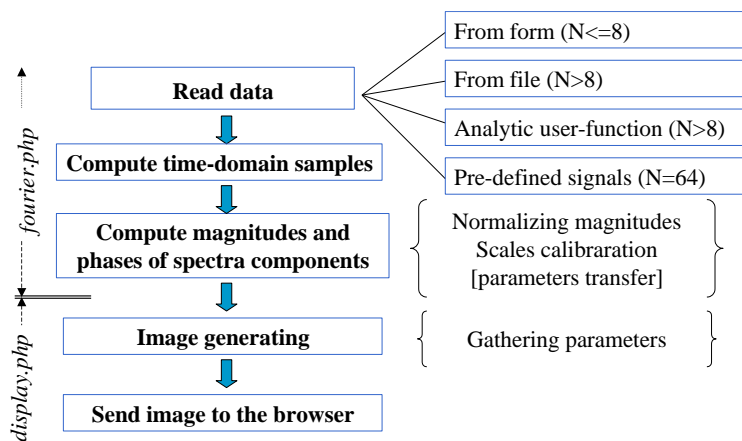


Figure 4 Algorithm representation of module No. 2

Due to the large number of parameters that must be transferred between scripts, the method of session variables is used. The script sources are given in reference [6]. In figure 5 are displayed the magnitudes and the phases of the first 16th spectral components of the function given in eqn. 3.

$$f(n) = \sin \frac{2n\pi}{N} + 1.3 \cos\left(\frac{3n\pi}{N} + \frac{\pi}{4}\right), N=64, T_0=10^{-4}s \quad (3)$$

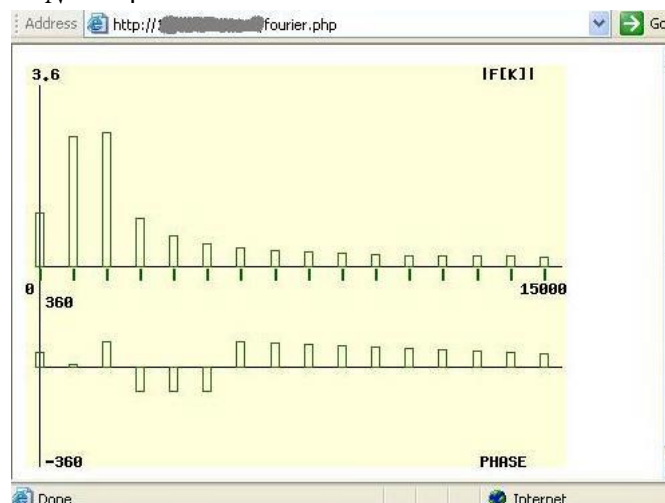


Figure 5. The magnitudes and the phases of spectral components.

For comparison, in figure 6 are given 64 samples of the same signal and the magnitude of spectra components, determined with FOURIER function in MS Excel. Obviously the results are similar, excepting the X-axis scale, which is graduated in frequency units in our application respectively in number of components in MS Excel. In addition, MS Excel starts numbering components from 1 instead from 0, which is the engineering accepted numbering system.

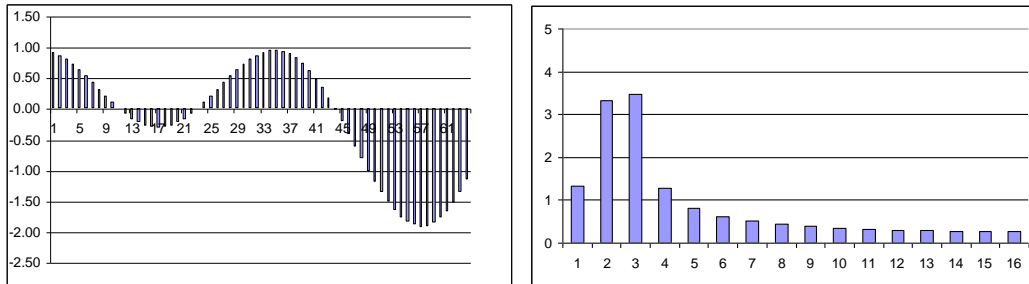


Figure 6. The samples of time-domain signal and its spectra magnitudes

We also have to notice that despite the input samples counts 64 values, only 16 output spectra components are displayed. This happens because FFT algorithm returns 32 pairs of real-imaginary components among which only the first half are reliable.

CONCLUSIONS

The application we present in this paper may be very useful for students when learning electronic circuits analysis and signals analysis. It is not an application like SPICE or other from the same family, because the user can't design its own circuit in a graphical mode, he only can choose among some usual circuits topologies and set different values for components. However, the next step in application development is to allow users to describe their own circuit by mean of a netlist. Combining with other web technologies, as Java, it will be possible to draw simple circuit scheme and simulate its functioning.

REFERENCES

1. BURAGA S. (coord) et. al., 2003, *Aplicatii Web la cheie – studii de caz implementate in PHP (Ready-made Web Applications - Case studies implemented with PHP)*, Ed. Polirom, ISBN:973-681-456-1
2. <http://www.php.net/manual/ro/function.gd-info.php>
3. <http://www.boutell.com/gd/>
4. TILIUTE D. E., 2004, *Web Based Application for Electronic Circuits Simulation* The Fourth International Conference "Internet-Education-Science", Vinnytsa, Ukraine, 5-9 Oct. 2004, Proc. Vol.1, pp: 45-48, ISBN:966-641-102-4
5. <http://www.zend.com/codex.php?id=259&single=1>
6. TILIUTE D. E., 2004, *Aplicatie Web pentru analiza spectrala a semnalelor*, Sisteme distribuite (Distributed Systems), "Stefan cel Mare" University of Suceava 2004, ISBN:973-666-143-1