ISSN 1453 - 7303
"HIDRAULICA" (No. 3-4/2012)
Magazine *of Hydraulics, Pneumatics, Tribology, Ecology, Sensorics, Mechatronics*

EDUCATION IN DEVELOPMENT OF ELECTRONIC MODULES USING FREE AND OPEN SOURCE SOFTWARE TOOLS

Andrei DRUMEA[1]

[1] "Politehnica" University Bucharest, 313 Spl. Independentei, 060042 Bucharest, andrei.drumea@cetti.ro

Abstract: *An important topic in electronics education is the design and development of electronic modules with emphasis on embedded systems, complex activity that combines hardware and software design. Hardware development chain consists of analog and digital design, schematics capture and circuit simulation, printed circuit board design and generation of manufacturing information. Software development requires tools like code editor, compiler and debugger, usually grouped together in an Integrated Development Environment (IDE). Some key aspects of educational software are accessibility, simplicity, quality of documentation and lower cost for a working licence. Unfortunately, typical commercial software is expensive and too complex so, for educational purposes, free and/or open source software can be a better option. Present paper analyses some free and open source software tools for electronic modules development: gEDA, Eagle Light edition and KiCad for schematics capture and printed circuit design, LTSpiceIV and Ngspice for circuit simulation, and Code::Blocks and Eclipse for software development for embedded systems.*

Keywords*: electronic modules, CAD, education, open source*

1. Introduction

Education in the field of mechatronics includes mechanics courses, informatic courses and electronics courses, usually related to embedded systems development. The typical mechatronic systems maybe found in the articles of the specialists in hydraulics [3], [4]. This topic covers many important areas like software development, analog and digital circuits design and simulation, CAD/CAM activities like printed circuit board design and preparing for manufacturing; all these activities require software tools coupled in a chain (fig.1).

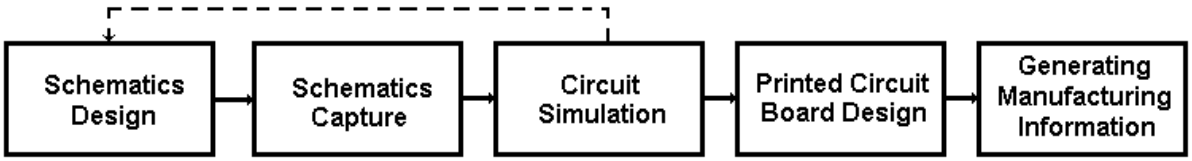

Fig.1 Typical hardware development tool chain

CAD/CAM software can be proprietary, with higher or lower license costs, or free/open source software, usually developed by academia or enthusiasts/hobbyists. In last years, a lot of proprietary software for electronics evolved into complex packages, hard to manage and too complicated to use in education environment where focus is on solving simple problems and explain principles. Free and open source software, on the other side, is developed with these aspects in mind, and in last years it reached a performance level almost similar with commercial software. Free software is released as binary (already compiled code, ready for run) and as source code (that must be compiled first by the user) and this means that source code is always available for improvements for free and open source software. A quote from GNU General Public License, version 3, 29 June 2007, developed by Free Software Foundation Inc., describes the essence of free software ([5]): "When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you

want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things."

2. Tools for hardware design of electronic modules

CAD software for electronic modules design consists, at least, in schematic capture program and printed circuit board (PCB) layout editor. Extra modules, like circuit simulator and Gerber viewer program for preparing manufacturing files, can be available.

Commercial software for electronic design is represented by tools like Mentor Graphics, Cadence with Orcad ([1]) and Allegro suites, Altium Designer or EAGLE.

Among these programs, an interesting option is freeware version of EAGLE (Easily Applicable Graphical Layout Editor) ([8]), that combines all features of commercial version (same simplicity and productivity, access to all component libraries and ability to run ULPs- user language programs to automate some tasks) . The limits of freeware version are acceptable for education – schematic must fit on one A4 page, printed circuit board design with no more than 2 layers and 80x100mm board size, but student projects usually fit in these limits. A GPS module board ([2]) was easy developed within these constrains.

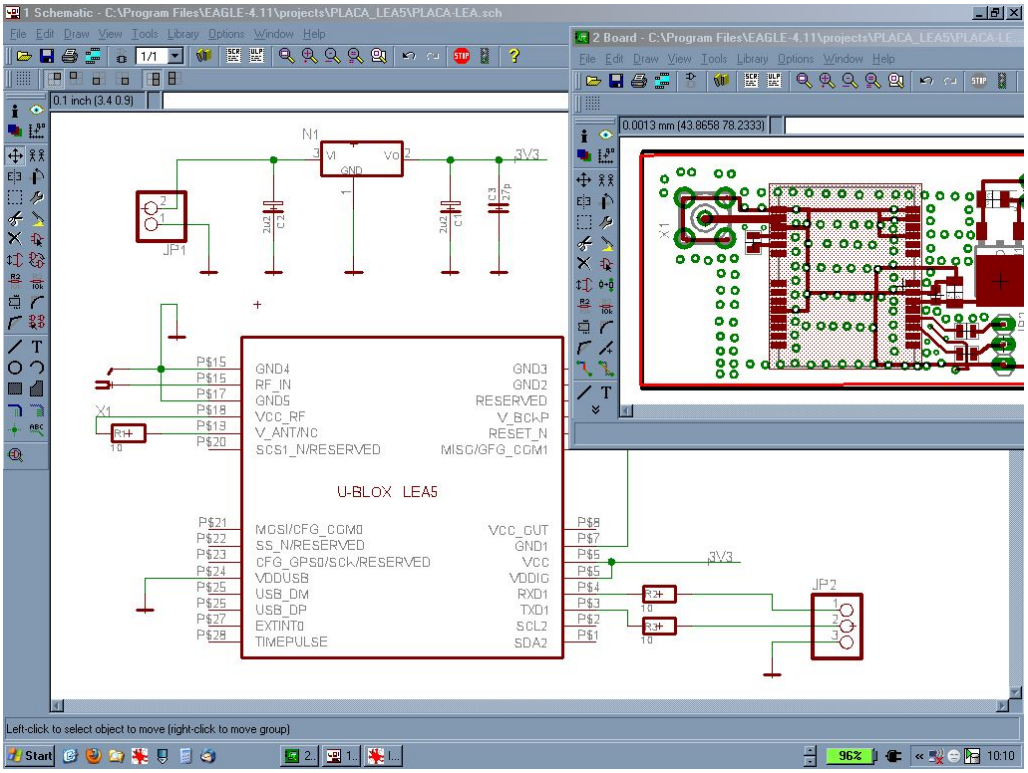

Fig.2 Schematic capture and layout design in EAGLE framework

Unfortunately EAGLE package offers no circuit simulation tool. An excellent option is to use another free program, LTSpiceIV, from semiconductor vendor Linear Technology. The program has a powerful SPICE engine (Simulation Program with Integrated Circuit Emphasis, de facto standard in electronic circuit simulation) and offers modern device models; it is obviously biased on Linear Technology components but extra devices can be added into the libraries. Other advantages are frequent updates and releases and a large and internet-active user base. Its installation is small and the program is able to use multi-core processors for faster simulations. In

some simple examples of circuits, the program performs much better and reaches convergence without any tweaks compared to commercial Spice implementations. In figure 3 is shown the simulation of an astable circuit with LTSpiceIV program.

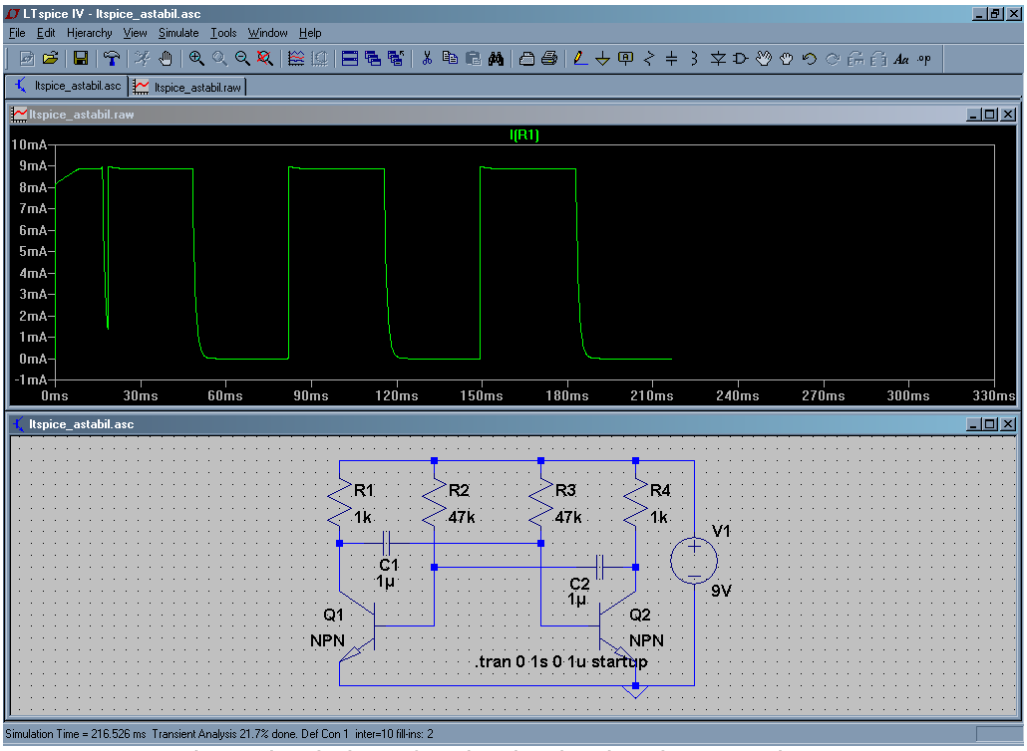

Fig.3 Simulation of a simple circuit using LTSpiceIV

Another software tool for electronics design is KiCad. It consists of Eeschema schematic editor (fig.4), PCBnew board editor, Gerber viewer (fig.5) and library management tools. It is open source program, so free to use and all sources available (Eagle and LTSpiceIV sources are not available). Its interface is not as efficient as Eagle, but is intuitive and easy to learn.

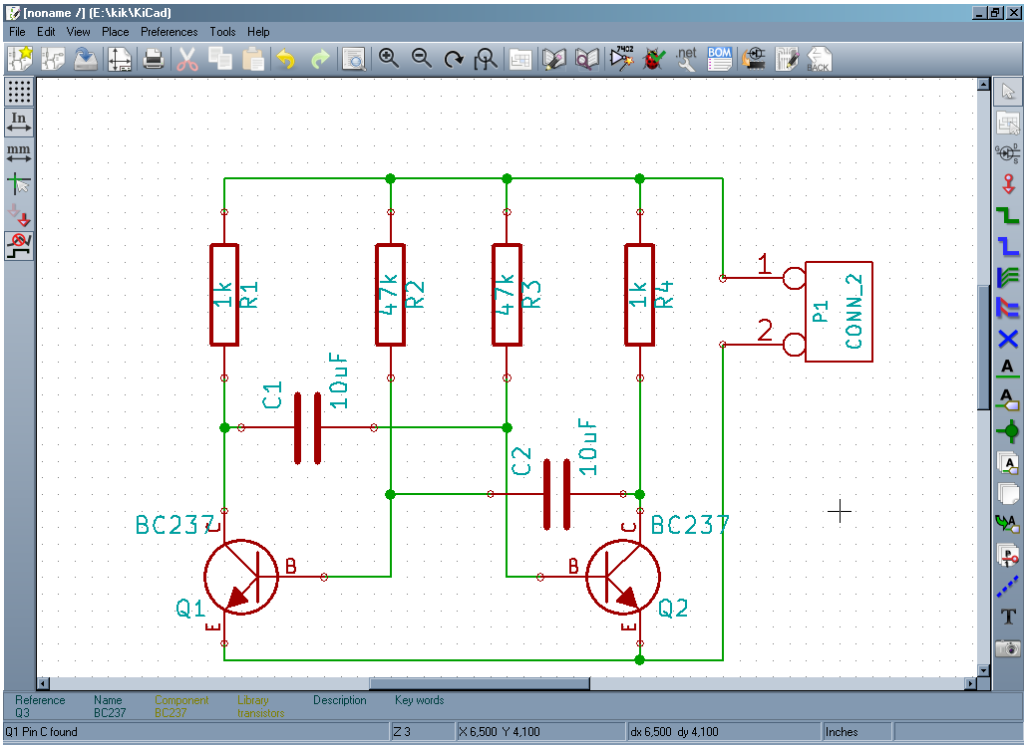

Fig.4 Schematic capture in KiCad framework

The board editor can process boards with maximum 16 electrical layers and 12 technical layers (like solder mask, silkscreen) and has a large component database.

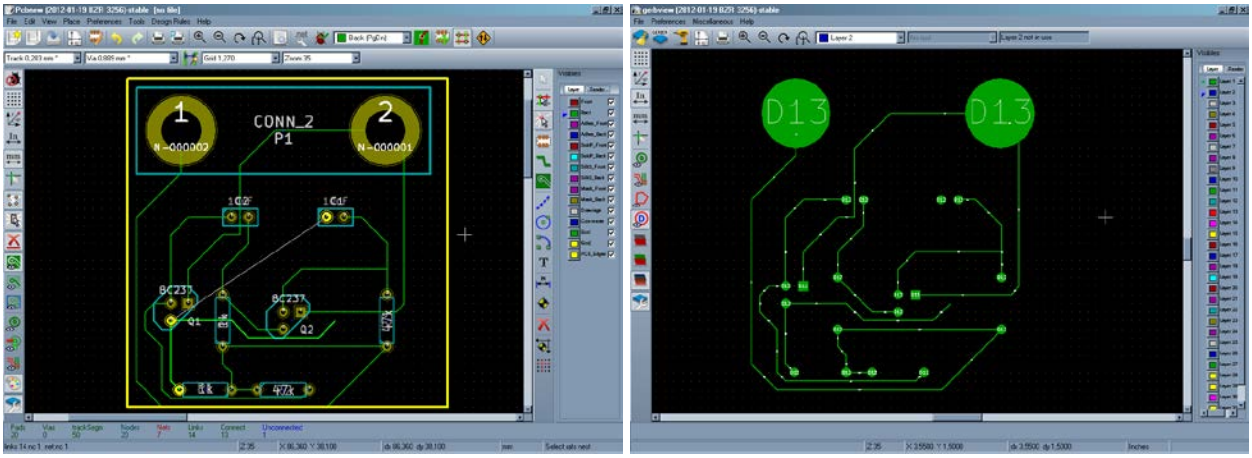

Fig.5 Layout editor and Gerber viewer in KiCad framework

Another powerful CAD suite for electronics is gEDA (GPL EDA), also open source and with large user database ([6]). It consists of gschem schematic capture, pcb board editor, Gerber viewer and integrates also other open source tools like spice. gEDA suite is developed especially for GNU/Linux open source operating systems. In figure 6 is shown a simple connector design with gschem and gerbv performed by Peter Clifton ([5]).

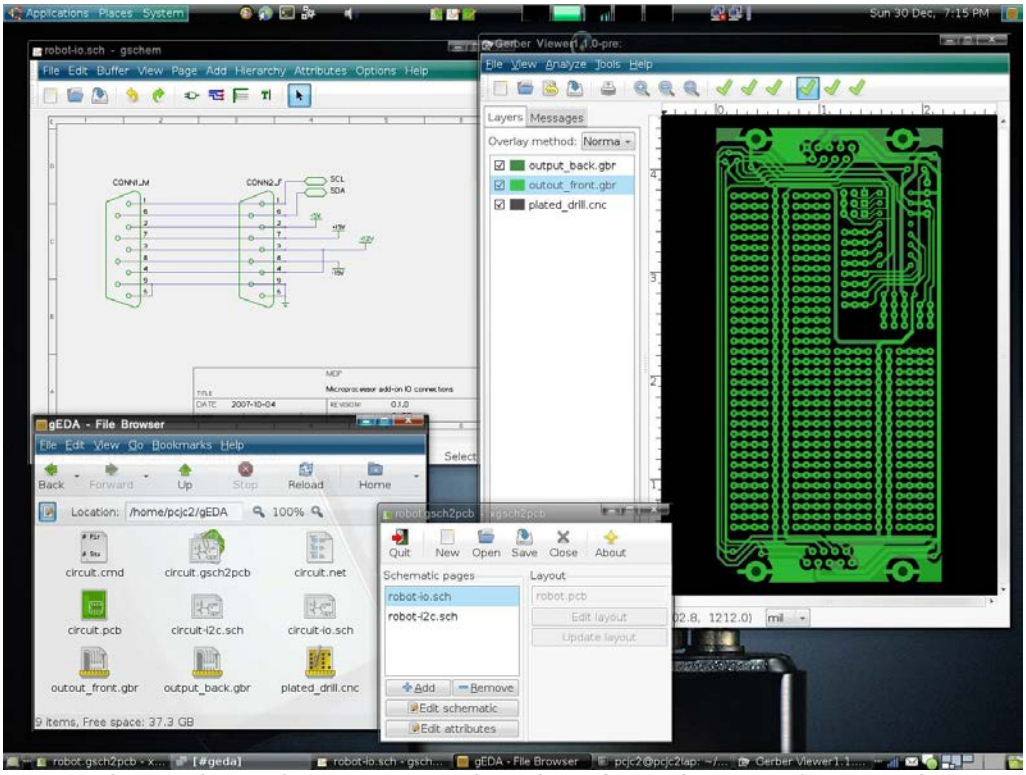

Fig.6 Schematics capture and Gerber viewer in gEDA framework

Board editor is also very powerful and with large component libraries. Lots of electronics projects on Internet are developed using gEDA tools. In fig. 7 is shown such a project, a lightning detector board developed by Kai-Martin Knaak ([5]).

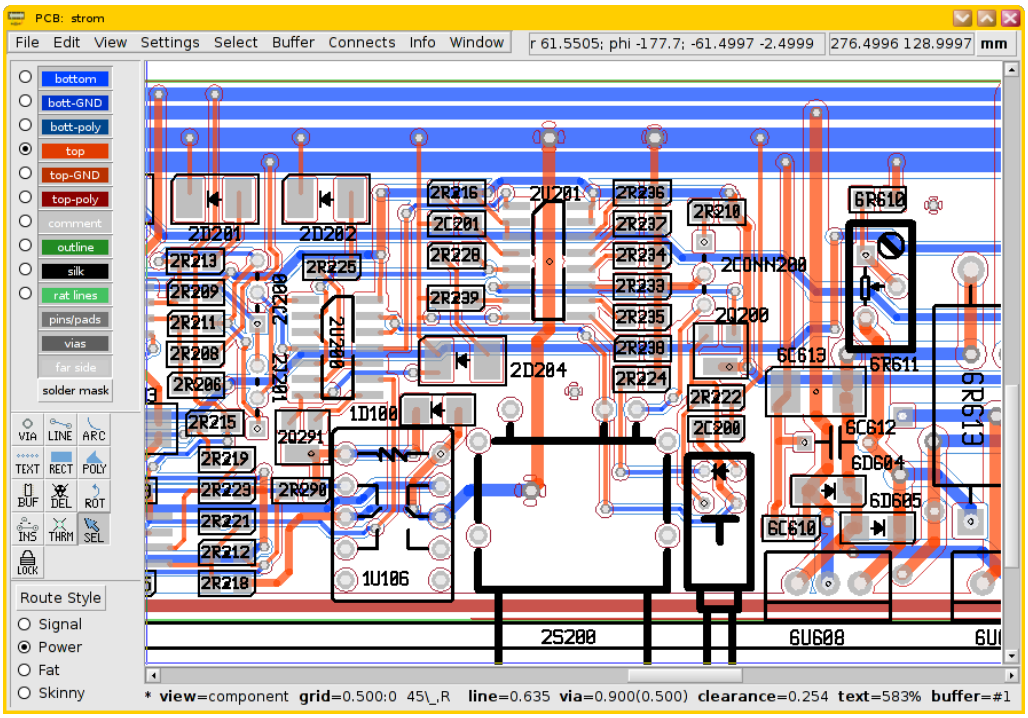

Fig.7 Layout editor in gEDA framework

3. Free and open source tools for software development for embedded systems

Embedded systems development usually requires C code editor, compiler, debugger and programmer. The large availability of open source GNU C Compiler for almost all existing processors made it a good choice in embedded system design. Its only disadvantage is the lack of a user friendly integrated development environment (IDE). Fortunately, in the last years 2 solutions occurred – Eclipse framework, a Java based IDE, very complex and powerful, but hard to manage and with large resource requirements, and Code::Blocks.

Code:Blocks is an open source IDE designed to be very extensible and fully configurable that runs on slow machines and has a consistent look, feel and operation across platforms ([5]). It can be extended with plugins so any extra functions can be implemented using another plugin. For instance, compiling and debugging for different processors or microcontrollers like Intel MCS51, Atmel AVR, Texas Instruments MSP430, ARM, MIPS, PowerPC processors or even Windows, Mac OS X or Android applications.

For a complete embedded systems development framework, user must provide for Code::Blocks the required (cross) compiler and debugger and to make some small configurations. After these steps, development process runs smoothly and can be more productive than commercial solutions. For example, for MSP430 microcontrollers, Code::Blocks IDE, mspgcc C compiler and mspdebug debugger offer an open source alternative more productive than chip manufacturer solutions and without any limitations in code size.

In figure 8 is shown a small C program for MSP430 microcontroller edited and compiled using Code::Blocks framework.

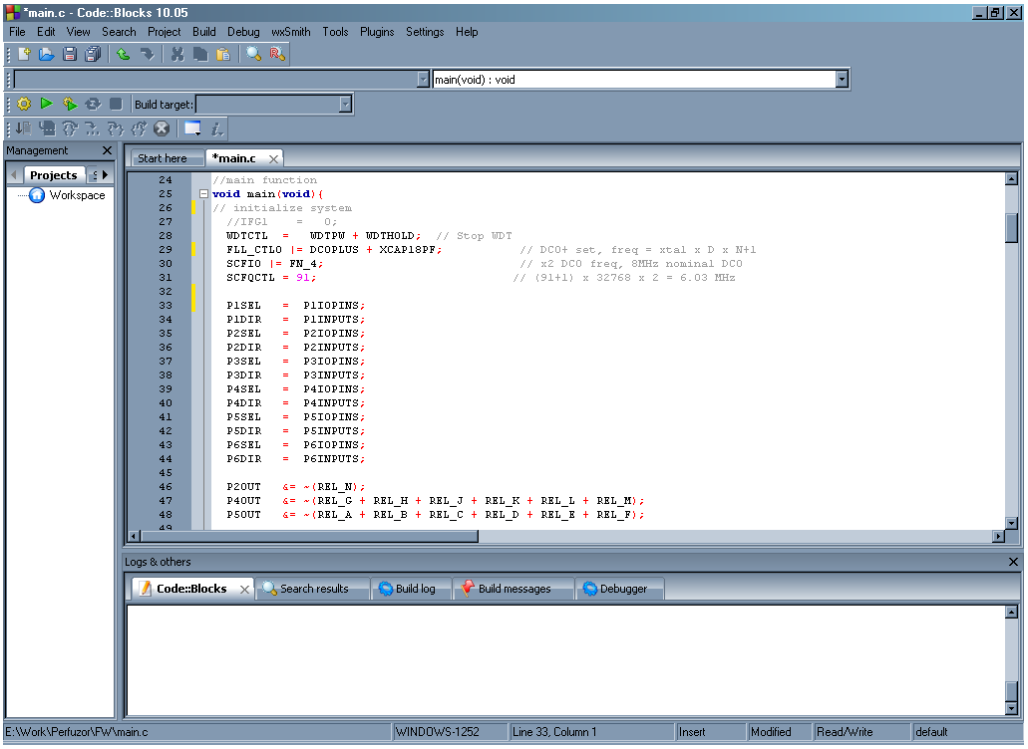

Fig.8 Code::Blocks editor with C program for MSP430 microcontroller

4. Conclusions

Education in electronics, especially in embedded system design, can be performed using only free and open source software tools. The quality of these products greatly improved in the last years and reached almost the same level with commercial software, but their inherent

References

[1] K. Mitzner, "Complete PCB design using OrCad capture and layout", Newnes, Burlington MA, USA, 2007
[2] A.Drumea, N.Irimie, R.Bunea and Al.Vasile, "Communication Module for Laser Rangefinder with Integrated Positioning System", SIITME-2009, Gyula, Hungary; 17-20 September, 2009, ISBN: 978-1-4244-5132-6; pp.301-304
[3] Dragos Ion GUTA, Catalin DUMITRESCU, Ioan LEPADATU, Corneliu CRISTESCU -"Experimental identification of electrohydraulic servomechanisms with virtual instruments technique", HIDRAULICA Magazine, No.3/2010, ISSN 1453-7303, pp.49 – 56
[4] Gabriela MATACHE,Ionel NITA, Alexandra VISAN, Adrian PANTIRU- "High bearing mechanical \ structures made of thin wall steel profile using innovative solutions in the field of hydraulics and resistance calculation methodology", HIDRAULICA Magazine, No.3-4/2011, ISSN 1453-7303, pp.75-80
[5] http://www.gnu.org/licenses/gpl.html
[6] http://en.wikipedia.org/wiki/GEDA
[7] http://www.codeblocks.org/
[8] http://www.geda-project.org/
[9] http://www.kicad-pcb.org
[10] www.cadsoft.de