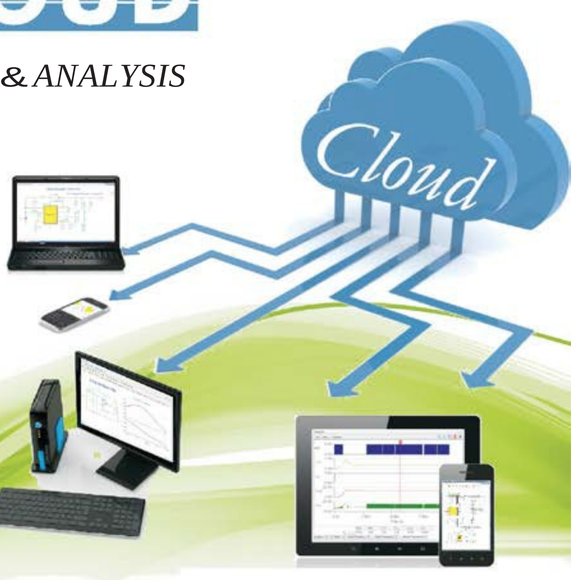
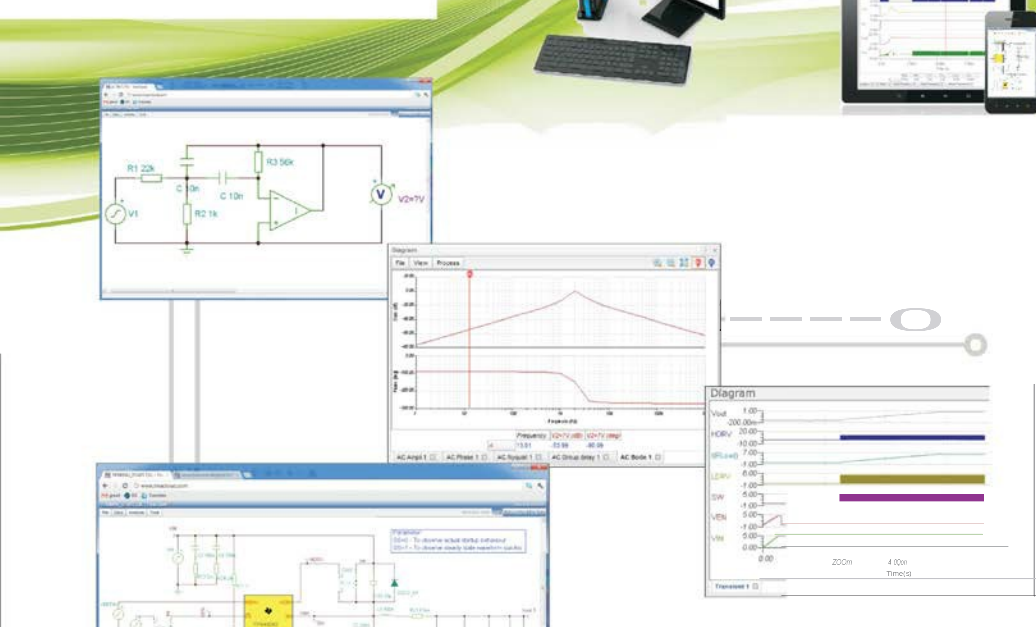
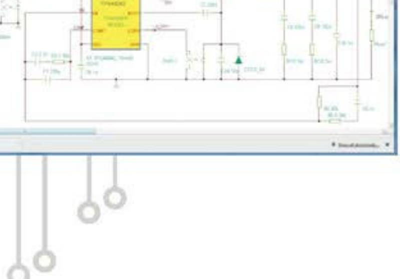
TINA CLOUD

WEB-BASED CIRCUIT DESIGN & ANALYSIS



USERS MANUAL





www.tina.com
www.tinacloudcom

DesignSoft
www.designsoftware.com

TINA Cloud

WEB-BASED CIRCUIT DESIGN & ANALYSIS

USERS MANUAL

DesignSoft
www.designsoftware.com

COPYRIGHTS

LIMITED LIABILITY

TINA, *TINACloud* together with all accompanying materials, is provided on an "as is" basis, without warranty of any kind.

DesignSoft, Inc., its distributors, and dealers make no warranty, either expressed, implied, or statutory, including but not limited to any implied warranties of merchantability or fitness for any purpose.

In no event will DesignSoft Inc., its distributor or dealer be liable to anyone for direct, indirect, incidental or consequential damages or losses arising from the purchase of *TINA* or from use or inability to use *TINA.*

TRADEMARKS

IBM PC/AT, PS/2 are registered trademarks of *International Business Machines Corporation*

Windows, *Windows 9x/ME/NT/2000/XP/Vista / Windows 7/ 8/ 10* are trademarks of *Microsoft Corporation.*

PSpice is a registered trademark of *Cadence Design Systems, Inc.*

TINA is a registered trademark of *DesignSoft, Inc.*

TINACloud is a registered trademark of *DesignSoft, Inc.*

** English version*

TABLE OF CONTENTS

1. INTRODUCTION 7

1.1 What are TINA and TINACloud? 7
1.2 Available Program Versions 12
1.3 Optional supplementary hardware 13
3.1.1 LabXplorer Multifunction Instrument for Education and Training with Local and Remote Measurement capabilities 13

2. COMPARISON OF TINACLOUD 15

3. RUNNING AND START-UP 17

3.1 Hardware and software requirements 17
3.1.1 Minimum hardware and software requirements 17
3.2 Experimenting with Example Circuits, avoiding common problems 20

4. GETTING STARTED 23

4.1 Measurement Units 23
4.2 Opening Circuits 24
4.3 Schematic Editing 24
4.4 Placing the Circuit Components 25
4.4.1 Wire 26
4.4.2 Input and Output 27
4.5 Exercises 27
4.5.1 Editing an RLC Circuit Schematic 27
4.6 Analyses 31
4.6.1 Analyzing an RLC Circuit (DC, AC Transient and Fourier analysis) 32
4.6.2 Creating and analyzing an Op-Amp circuit 39
4.6.3 Calculating DC Transfer Characteristic 42
4.6.4 Analysis of SMPS Circuits 43
4.6.5 Stress Analysis 46
4.6.6 Network Analysis 48
4.6.7 Analyzing a Digital Circuit with TINA's Digital Engine 49
4.6.8 Analyzing a Digital Circuits using Digital HDL Simulation Mode 51
4.6.8.1 Analyzing a Digital Circuit Using Digital VHDL Simulation 51
4.6.8.2 The HDL Debugger 54
4.6.8.3 Analyzing a Digital Circuit Using Digital Verilog Simulation 55
4.6.8.4 Analyzing Circuits Using Verilog-A models 56
4.6.8.5 Analyzing Circuits Using Verilog-AMS models 57
4.6.9 Mixed Mode Simulation (Spice, VHDL, MCU co-simulation) 59

4.6.9.1 Waveform generation with a VHDL and Spice subcircuit ... 59
4.6.9.2 MCU controlled SMPS circuit ... 63
4.6.10 Testing your Circuit in Interactive mode ... 64
4.6.10.1 Digital Circuit with a Keypad ... 65
4.6.10.2 Light Switch with Thyristor ... 65
4.6.10.3 Ladder Logic networks ... 66
4.6.10.4 HDL Circuits ... 67
4.6.10.5 Microcontroller (MCU) Circuit ... 68
4.6.10.6 Using the ASM Debugger ... 69
4.6.10.7 Example PIC Interrupt handling ... 70
4.6.10.8 Making a Breakpoint in ASM ... 73
4.6.10.9 Debugging C code in MCUs ... 73

5. USING SCHEMATIC SUBCIRCUITS AND SPICE MACROS 77

5.1 Making a Macro from a schmatic ... 77
5.2 Making a Macro from a Spice subcircuit ... 81

6. CREATING A PRINTED CIRCUIT BOARD (PCB) 85

6.1 Setting and checking footprint names ... 86
6.2 Invoking TINACloud PCB ... 88
6.3 Advanced editing functions of TINACloud PCB's Layout editor ... 93

7. CREATING A TWO-LAYER, DOUBLE-SIDED, SURFACE-MOUNT TECHNOLOGY BOARD 94

CHAPTER 1

INTRODUCTION

1.1 What are TINA and TINACloud?

TINA Design Suite is a powerful yet affordable software package for analyzing, designing and real time testing of circuits with analog, digital & microcontroller components and components defined in various HDLs (Hardware Description Languages) and for designing their PCB layouts. You can also analyze RF, communication, optoelectronic circuits and mechatronics applications with 3D interface. TINA is a Windows application which you should install on your computer or LAN server.

There is now an online version of TINA called TINACloud. If you have licenses for both products, you can store your designs on the web and run anytime on any platforms without installation, including PCs, Macs, thin clients, tablets, smart phones, smart TVs and e-book readers. The program will run on our powerful web server with the same high speed whether you use a laptop, tablet or just a smartphone. You can then smoothly download your design from the web to your PC. Should you change something while you are on the road, continue the development off-line and upload your design again.

TINACloud supports most of the features of TINA however some features are possible in the offline TINA only at the time of writing this manual. A more detailed description of the differences can be found chapter 2 of this manual.

A unique feature of TINACloud permits you to bring circuits to life with the remote controlled LabXplorer hardware turn your computer into a powerful, multifunction T&M instrument. With LabExplorer, you can carry out remote measurement which is great tool for distance education.

TINACloud can also be used in the training environment. It includes unique tools for testing students' knowledge, monitoring progress and introducing troubleshooting techniques. With optional hardware it can be used to test real circuits for comparison with the results obtained from simulation. Of great importance to educators, the package includes all the tools needed to prepare educational materials.

Let's overview the most important features of TINACloud

Schematic Capture. Circuit diagrams are entered using an easy to use schematic editor. Component symbols chosen from the Component bar are positioned, moved, rotated and/or mirrored on the screen by the mouse. TINACloud's semiconductor catalog allows the user to select components from a user-extendible library. An advanced "rubber wire" tool is provided allowing easy modification of the schematic diagrams.

Electrical Rules Check (ERC) will examine the circuit for questionable connections between components and display the results in the Electrical Rules Check window. ERC is invoked automatically, so missing connections will be brought to your attention before analysis begins.

Text Editor. TINACloud includes a Text Editor for annotating schematics, calculations, includes graphic output, and measurement results. It is an invaluable aid to teachers preparing problems and examples.

Report. The circuit diagrams and the calculated or measured results are automatically save into a report file in pdf format which can be downloaded to your computer.

DC analysis calculates the DC operating point and the transfer characteristic of analog circuits. The user can display the calculated and/ or measured nodal voltages at any node by selecting the node with the cursor.

Transient analysis. In the transient and mixed mode of TINACloud you can calculate the circuit response to the input waveforms that can be selected from several options (pulse, unit step, sinusoidal, triangular wave, square wave, general trapezoidal waveform, and user-defined excitation) and parameterized as required. For digital circuits, programmable clocks and digital signal generators are available.

Fourier analysis. In addition to the calculation and display of the response, the coefficients of the Fourier series, the harmonic distortion for periodic signals, and the Fourier spectrum of non-periodic signals can also be calculated.

Digital Simulation. TINACloud also includes a very fast and powerful simulator for digital circuits. You can trace circuit operation step-by- step, forward and backward, or view the complete time diagram in a special logic analyzer window. In addition to logic gates, there are ICs and other digital parts from TINACloud's large component library.

HDL simulation. TINACloud now includes all major analog, digital and mixed Hardware Description Languages: VHDL, Verilog, Verilog-A and Verilog AMS to verify designs in analog, digital and mixed-signal analog-digital environments. Your circuits can contain editable HDL blocks from the libraries of TINACloud and Xilinx or other HDL components created by yourself or downloaded from the Internet. TINACloud compiles HDL into highly efficient machine code for speed optimization. You can freely combine HDL and Spice macros and the schematic components of TINACloud. Also you can edit the HDL source of any HDL components then simulate and see the result instantly. With the built in HDL debugger you can execute the HDL code step-by-step, add breakpoints, watchpoints, display variable infor- mation, etc.

Microcontroller (MCU) simulation. TINACloud includes a wide range of microcontrollers (PIC, AVR, 8051, HCS, ARM) which you can test, debug and run interactively. The built in MCU assembler allows you to modify your assembler code and see the result promptly. You can also program and debug MCUs in C, using external C compilers.

AC analysis calculates, complex voltage, current, impedance, and power can be calculated. In addition, Nyquist and Bode diagrams of the amplitude, phase and group delay characteristics of analog circuits can be plotted. You can also draw the complex phasor diagram. For non-linear networks, the operating point linearization is done automatically.

Network analysis determines the two-port parameters of networks (S, Z, Y, H). This is especially useful if you work with RF circuits. Results can be displayed in Smith, Polar, or other diagrams. The network analysis is carried out with the help of TINACloud's network analyzer. The RF models of the circuit elements can be defined as SPICE subcircuits (SPICE macros) which contain parasitic components (inductors, capacitors) or as an S-parameter model defined by its S (frequency) function. S functions are normally provided by the component manufacturers (based on their measurements) and can be downloaded from the Internet and inserted into TINACloud either manually or by using TINACloud's library manager.

Noise analysis determines the noise spectrum with respect to either the input or the output. The noise power and the signal-to-noise ratio (SNR) can also be calculated.

Symbolic analysis produces the transfer function and the closed form expression of the response of analog linear networks in DC, AC, and transient modes. The exact solution, calculated through the symbolic analysis, can also be plotted and compared to the numerically calculated or measured results.

Monte-Carlo and Worst-case analysis. Tolerances can be assigned to the circuit elements for use in Monte-Carlo and/or worst-case analyses. The results can be obtained statistically, and their expected means, standard deviations and yields can also be calculated.

Design Tool This powerful tool works with the design equations of your circuit to ensure that the specified inputs result in the specified output response. The tool offers you a solution engine that you can use to solve repetitively and accurately for various scenarios. The calculated component values are automatically set in place in the companion TINACloud schematic and you can check the result by simulation. This feature is also very useful for semiconductor and other electronics component manufacturers to provide application circuits along with the design procedure.

Optimization. TINACloud's enhanced optimization tool can tweak one or more unknown circuit parameters to achieve a predefined target response. The target circuit response (voltage, current, impedance, or power) must be "monitored" by meters. For example, you can specify several working point DC voltages or AC transfer function parameters and have TINACloud determine the values of the selected components.

Post-processor. Another great new tool of TIN A is its post-processor. With the post-processor, you can add new curves of virtually any node and component voltage or current to existing diagrams. In addition, you can post-process existing curves by adding or subtracting curves, or by applying mathematical functions to them. You can also draw trajectories; i.e., draw any voltage or current as a function of another voltage or current.

Interactive mode. When everything is in order, the ultimate test of your circuit is to try it in a "real life" situation using its interactive controls (such as keypads and switches) and watching its displays or other indicators. You can carry out such a test using TINACloud's interactive mode. You can not only play with the controls, but you can also change component values while the analysis is in progress. In addition, you can assign hotkeys to component values and switches to change them simply by pressing a key. You will immediately see the effect of the change. You can also test MCU applications in TINACloud's interactive mode. You can not only run and test them using the several lifelike interactive controls e.g., keyboards, but you can also debug them while the MCU executes ASM code step by step, And displays the register contents and TINACloud's outputs in each step. If necessary you can modify the ASM code on the fly and test your circuit again without using any other tool.

Real-time Test & Measurements. TINACloud can go beyond simulation when LabXplorer hardware is installed on the server. With this hardware, TINACloud's powerful tools can make real-time measurements on real circuits.

Training and Examination. TINACloud has special operating modes for training and for examination. In these modes, under TINACloud's control, the students solve problems assigned by the teacher. The solution format depends on the types of problems: they can be selected from a list, calculated numerically, or given in symbolic form. The interpreter - providing a number of solution tools - can also be used for problem solving. If the student cannot solve the problem, he/she can turn to the multilevel Advisor. The package includes all the tools needed to produce educational materials. Another special educational function of TINACloud is the software or hardware simulation of circuit faults to practice troubleshooting. Using TINACloud, you can transform existing PC classrooms and even home computers into contemporary electronics training labs at low cost.

1.2 Available Program Versions

Different program versions, tailored to meet various needs, are available.
Both versions are available with the following features:

- **Industrial version:** Includes all of TINACloud's features and utilities.
- **Educational version:** It has most features of the Industrial version but parameter stepping and optimizations are allowed for one parameter only, Stress Analysis is also not included.
- **Classic Edition:** It has the same features as the Educational version above, except that Network Analysis is not allowed, TINACloud's large S-parameter component library and the Parameter Extractor, Stress Analysis and the Steady State Solver are not included.
- **Student Version:** Has the same features as Classic Edition version except that the circuit size is limited to 100 nodes including internal Spice macro nodes. Global Parameters and HDL extensions are not allowed.
- **Basic version:** Has the same features as Classic Edition except that the circuit size is limited to 200 nodes including internal Spice macro nodes. Global Parameters and HDL extensions are not allowed.

1.3 Optional supplementary hardware

1.3.1 LabXplorer: Multifunction Instrument for Education and Training with Local and Remote Measurement capabilities

LabXplorer turns your desktop, laptop, tablet or smart phone into a powerful, multifunction test and measurement instrument for a wide range of applications. Instruments, whatever you need, are at your fingertips. LabXplorer provides multimeter, oscilloscope, spectrum analyzer, logic analyzer, programmable analog and digital signal generator, impedance analyzer and also measures characteristics of passive electronic components and semiconductor devices.

LabXplorer can be used with its virtual instruments both stand-alone or remotely through the Internet or LAN.

It also supports the TINACloud circuit simulation program and its cloud based version TINACloud for comparison of simulation and measurements as a unique tool for cir cuit development, troubleshooting, and the study of analog and digital electronics.

In remote mode Labexplorer’s virtual instruments run on most OSs and computers, including PCs, Macs, thin clients, tablets—even on many smart phones, smart TVs and e-book readers. You can use LabXplorer remotely in the classroom, computer lab, at home, and, in fact, anywhere in the world that has internet access. LabXplorer comes with various, remotely programmable, plug-in analog, digital and mixed circuit experiment boards.

CHAPTER 2

COMPARISON OF FEATURES IN DIFFERENT VERSIONS OF TINACLOUD

In this chapter we provide a comparison of features in different versions of TINA and TINACloud.

Comparison of TINACloud versions

TINACloud version comparison	TINACloud Industrial	TINACloud Educational	TINACloud Classic	TINACloud Basic	TINACloud Student
Circuit Entry	+	+	+	+	+
Schematic Editor	+	+	+	+	+
Open designs and macros from Web	+	+	+	+	+
Undo	+	+	+	+	+
Redo	+	+	+	+	+
Automatic/manual wire routing and drag support	+	+	+	+	+
Instruments as standard schematic symbols	+	+	+	+	+
Subcircuits	+	+	+	+	+
Bus	+	+	+	+	+
Integrated Netlist Editor	+	+	+	+	+
Equation Editor	+	+	+	+	+
Excitation Editor for arbitrary waveforms	+	+	+	+	+
Analyses					
Multi core support	+	+	+	+	+
Enhanced analysis speed & convergence	+	+	+	+	+
Max. number of external nodes and nodes in macros	unlimited	unlimited	unlimited	200	200
DC, AC, Transient, Digital, Mixed mode Simulation	+	+	+	+	+
RF Simulation	+	+	+	+	+
RF models given by S-parameters	+	+	-	-	-
Network Analysis	+	+	-	-	-
Number of components and models	**20000**	**20000**	**10000**	**10000**	**10000**
Digital Simulation	+	+	+	+	+
VHDL Simulation	+	+	+	max 5000 lines	max 5000 lines
Verilog, Verilog A&AMS	+	max. 1000 lines	max. 1000 lines	-	-
HDL debugger	+	+	+	+	+
MCU simulation and debugging	+	+	+	+	+
Interactive Mode	+	+	+	+	+
Symbolic Analysis (closed formulas)	+	+	+	+	+
Fourier Analysis (harmonics)	+	+	+	+	+
Fourier Analysis (spectrum)	+	+	+	+	+
Noise, Monte Carlo, Worst Case	+	+	+	+	+
Stress (Smoke) Analysis	+	-	-	-	-
Group Delay	+	+	+	+	+
Number of Optimization Targets & Parameters	1	1	1	1	1
Number of Parameters in Parameter Stepping	any	1	1	1	1
Parameter Sweeping	+	+	+	+	+
Output Capabilities					
Scaled Diagrams	+	+	+	+	+
Nyquist Diagram	+	+	+	+	+
Tools to enhance diagrams	+	+	+	+	+
Excel export	+	+	+	+	+
Educational Features	-	+	-	-	+

CHAPTER 3

RUNNING AND START-UP

3.1 Hardware and software requirements

A great feature of TINACloud that is runs on all widely used platforms including both hardware and operating systems. A c ommon requirement that all platforms must have internet connection and a compatible browser.

3.1.1 Minimum hardware and software requirements

Windows (desktops and laptops)
Windows

- Microsoft Windows XP / Vista / Windows 7 / 8 / 10
- Browser IE 9, Chrome 34.0, Firefox 29.0 or later

OS X (Mac desktops and laptops)

- OS X
- Safari, Chrome

Linux (desktops and laptops)

- Linux
- Chrome, Firefox

iOS (Apple iPad tablets and iPhones)

- iOS 5.x, 6.x, 7.x
- Safari, Chrome

Android (Android tablets, ebook readers, smart phones & smart TVs)

- Android 2.x, 3.x, 4.x
- Chrome, Firefox

Windows Phone

- Windows Phone 8
- IE 11 or later

A great feature of TINACloud that it does not need any installation. Once you receive your credentials (username and password) and you have a good internet connection you just need to login and can start working right away. You will normally get your email information in an email like this:
Thank you for your registration. Your remote login credentials are as follows.

Registration credentials:

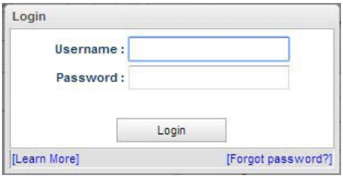
Remote application URL:	http://www.tinacloud.com
Username:	Joe
Password:	123456

Registered product:

Name	Version	Expires	Price
TinaCloud	Educational	201x-xx-xx	n/a

When you click http://www.tinacloud.com you will be redirected to TINACloud's homepage. Click the 

button and the TINACloud will open in your browser. On a desktop or laptop the following screen will appear:



Enter your credentials and press Login. If your credentials are valid, the opening screen of TINACloud will appear.

NOTE:

It may happen that on some phone the Desktop interface and vice versa on some tablets the phone interface appears and you want to switch to Mobile or Desktop interface. After Login you can do that using the View menu under the Desktop and the Eye icon under the mobile interface.

NOTE:

Under the Mobile interface text notes are disabled by default, so only the schematics and the diagrams are displayed. To see the above mentioned instructions select the Eye icon and then Show-> Circuits, Texts and Pictures.

3.2 Experimenting with Example Circuits, avoiding common problems

Start the program and click the *File* menu item in the top line of the screen or tap the icon at the bottom left corner of the mobile interface to get to the *File* menu. Select the *Open* command and the open file dialog box appears Select the *EXAMPLES* folder, and a list of files with .*TSC* extensions will appear. After selecting a file, the circuit schematic will appear.

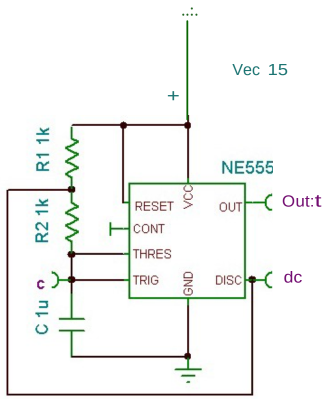
Now you can execute an analysis, modify or expand the circuit, and evaluate the results. Keep in mind that every command may be aborted by pressing the *[Esc]* key or clicking on the *Cancel* button.

We recommend that you load the following circuits and follow the instructions on the screen for the circuit types listed below.

This will avoid some common problems.

Oscillator circuit	EXAMPLES*colpitts.tsc*
555 Oscillator	EXAMPLES*555_AST.tsc*
Rectifier circuit	EXAMPLES*Bridge Rectifier1.tsc*

For example at 555.tsc you will see:



IMPORTANT

Make sure to set the "Zero Initial values"
option in Transient Analysis dialog, as shown
bellow,
otherwise you might get an "Operating point not found"
or "Irregular circuit" error message.
This is because in the this mode the 555
has no DC operating point.

CHAPTER 4

GETTING STARTED

In this chapter, we present a step by step introduction given using examples.

4.1 Measurement Units

When setting parameters for electronic components or specifying numerical values, you may use standard electronic abbreviations. For example, you can enter 1k (ohm) for 1000 (ohm). The multiplier abbreviations should follow the numeric value, e.g., 2.7k, 3.0M, 1u, etc.

The following characters indicate multiplier factors:

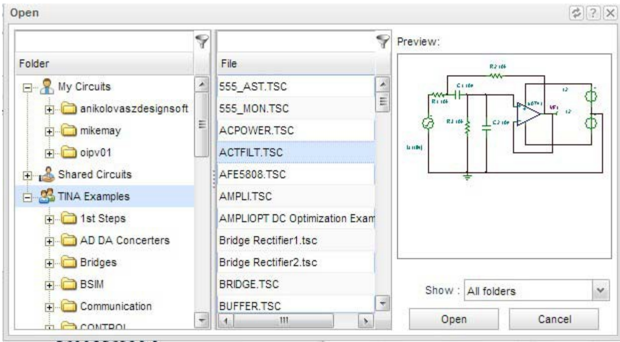
p = pico = 10^{-12}	**T** = tera = 10^{12}
n = nano = 10^{-9}	**G** = giga = 10^{9}
u = micro= 10^{-6}	**M** = mega = 10^{6}
m= milli = 10^{-3}	**k** = kilo = 10^{3}

NOTE:

Upper and lower cases must be carefully distinguished (e.g., M = m), and the selected letter must follow the numeric characters without a space (e.g., 1k or 5.1G), or TINA will indicate an error.

4.2 Opening Circuits

To open an existing circuit, stored under your account, start the program and click the *File* menu item in the top line of the screen or tap the icon at the bottom left corner of the mobile interface to get to the *File* menu. Select the *Open* command and the open file dialog box appears. Select the *TINA EXAMPLES* folder to the examples provided with TINACloud, and a list of files with *.TSC* extensions will appear. In the desktop version you can also see a preview of the selected circuit.



After pressing the Open button or in the mobile version simply tapping the file name the circuit schematic will appear in your browser.

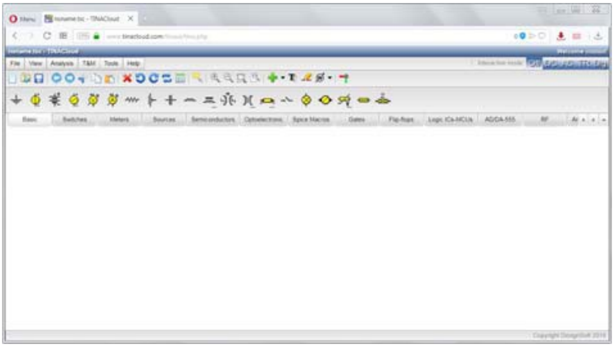
4.3 Schematic Editing

In TINACloud the Schematic Editor is is similar to the offline version. However, apart from other parts of TINACloud it runs on your computer to ensure a fast and smooth editing.

When you log in to TINACloud, the default circuit — noname.tsc from your My Circuits folder — will be automatically loaded into the editor. You can start editing it or running analyses on it as if you have opened it with the open dialog. Everything you change in the circuit will be automatically uploaded to the server and saved in your account. If you wish to undo a change you made, you can either use the undo menu/button or use the File|Revert to original menu which will reload the last saved version of the circuit.

When you are ready with editing you can save your circuit in your "My Circuits" folder with your chosen name.

If you want to make a new circuit you can press the New button or the File|New menu. In this case you will need to specify a name and folder for your new circuit. If the circuit already exists on the server it will be deleted, so you can start editing a new, empty circuit.



4.4 Placing the Circuit Components

Components are selected from the Component bar of the schematic editor and their symbols are moved by the mouse to the required position. When you click the left mouse button, the program locks the pins of the component symbol to the nearest grid dots.

Components can be positioned vertically or horizontally and rotated by 90-degree steps by pressing the  buttons. In addition, some components (like transistors) can also be mirrored around their vertical axis by pressing the  button on the toolbar.

On touch sensitive platforms when placing a component tap the component symbol on the toolbar then tap the approximate location on editing surface to place the symbol. The component will be placed in the editing area. You can then touch the symbol and drag to the final location with your finger or with a positioning device if available.

After a component symbol has been selected and positioned, you may double click on it to enable a dialog window where you can enter parameter values and a label.

When entering numeric values, abbreviations of integral powers from 10^{-12} to 10^{12} can be used. For example, 1k is understood as 1,000. See 4.1 for more information.

NOTE:

Upper and lower cases must be carefully distinguished (e.g., M = m), and the selected letter must follow the numeric characters without a space (e.g., 1k or 5.1G), or TINA will indicate an error.

TINACLOUD will automatically assign a label for each component you place on the schematic. It will also display the numerical value of the main component parameter (for example: R4 10k). Note that the value is shown only if the Values option of the View menu is checked. For files from the older versions of TINACLOUD, the Values option is turned off by default. The first part of the label, e.g., R4, is required for symbolic analysis modes. You can also display the units of the capacitors and inductors (for example: C1 3nF) if both the Values and the Units options of the View menu are checked.

4.4.1 Wire

A wire establishes a simple short (zero ohm connection) between two component pins.

To place a wire, move the cursor to the component terminal point where you want to begin. The cursor will change into a drawing pen. You can draw a wire in two different ways:

1) Select the starting point of the wire with a left mouse click, then move the pen with the mouse while TINACloud draws the wire along the path. While drawing the wire, you can move in any direction and the wire follows. At the end point of the wire, click the left button of the mouse again.

2) Hold down the left mouse button while positioning the pen; release it at the end point.

While drawing a wire, you can delete previous sections by moving backwards on the same track.

You can easily modify existing wires by selecting and dragging sections or edges.

For short wire sections, you may need to hold down the shift key while drawing.

Be sure not to leave any component nodes unconnected. If there are unconnected components or terminals, TINACloud's Electric Rule Check tool (ERC) will issue a warning (unless you have disabled it).

Wire segments made by the Wire tool are always vertical or horizontal. However, you can add angled wire segments using the components made for bridges, Y and D circuits under the Special component toolbar.

4.4.2 Input and Output

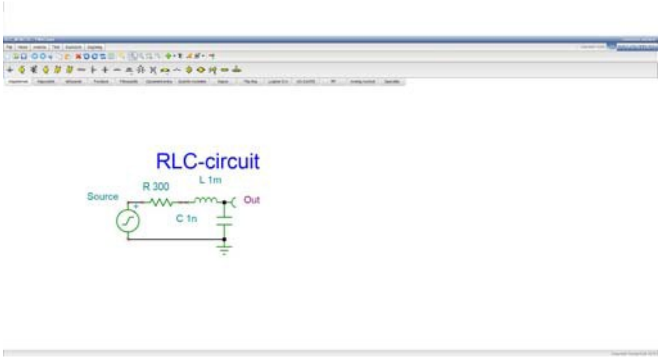
Certain types of analysis (DC Transfer characteristic, Bode diagram, Nyquist diagram, Group delay, Transfer function) cannot be executed until both input and output have been selected. These establish where the excitation is applied and where the circuit response is taken. The output(s) chosen also determine which curve(s) will be displayed in the chosen analysis mode. Sources and generators can be configured as inputs, while meters can be configured as outputs. However, meters can also serve to determine the location of the input quantity that will be used when computing AC Transfer curves and functions.

4.5 Exercises

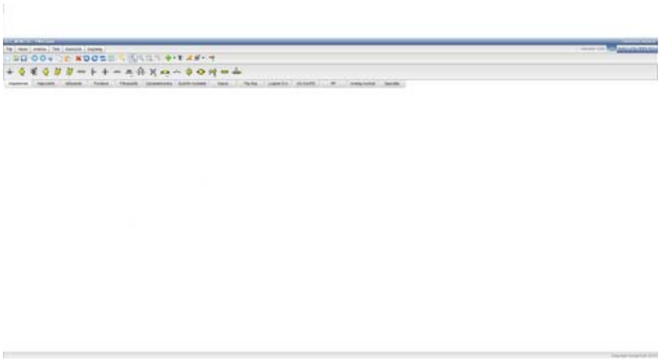
These exercises will help you build upon and integrate what you've learned from the manual so far.

4.5.1 Editing an RLC Circuit Schematic

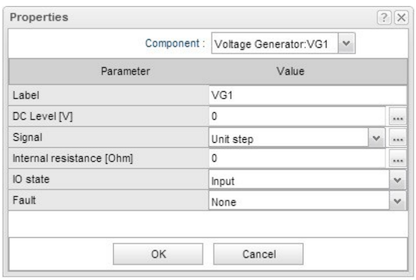
Create the circuit diagram of a series RLC network as shown in the following figure.



First clear the schematic window with the **File|New** command.

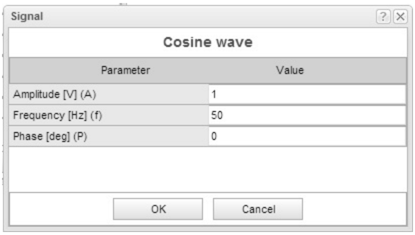


Now start adding components. Click on the voltage generator icon, then release the mouse button. The cursor will change into the generator symbol. Position it using the mouse (or by pressing the  buttons for rotation or the  button for mirroring) somewhere in the middle of the screen, then press the right mouse button; the schematic editor's popup menu will come up. Select *Properties*. The following dialog box will appear:



Leave the *DC level* and the *IO state* parameters unchanged. Note that by accepting *Input* for the *IO state* parameter you have selected the output of this generator to be the input for the Bode diagram.

In the the *Signal* menu line you can choose from several input waveforms and with the button set their parameters. Select Cosine wave and press the button the same line. The default parameters of the Cosine wave will appear:



Change the frequency to 200k (200kHz). Click on *OK* and return to the previous dialog box and click on *OK* again. The program will automatically place the label near the component and you will be able to position and place the component and the label together. If the default label position is not satisfactory, you'll be able to drag the label to the desired position later on. When the component is where you want it, press the left mouse button to drop it. This completes the placement of the generator.

Note that alternatively you can drop the components anywhere on the editing space and then rotate, mirror and drag to the final place afterwards. You can also set the properties later by double-clicking the components.

Note: On touch sensitive devices e.g. tablets the placement is slightly different. Tap the required component on the toolbar then tap the approximate place on the editing space to place the component. It remains selected (red color) but if not you can select again by tapping on it. Now you can rotate or mirror with the and buttons if necessary and drag to any position with your fingers or a stylus.

Now click on the *Basic* tab on the Component bar and choose the **Resistor** icon (your cursor will automatically change when you are over the tabs or the icons and the name of the components will be displayed). After the symbol of a resistor has appeared in the schematic window, press the right button of the mouse and then select *Properties* from the popup menu, or drop the symbol then double-click on it.

When the property dialog box appears, change the Resistance to 100.

Continue circuit entry with the **L** and **C** components as indicated in the figure above. Set the parameters to *L 1*= 1 m and *C1* = 1 n. Note the default values of the parallel resistive losses for the capacitor and the series resistive losses of the capacitor. Add the Voltage Pin (chosen from the Meters component group) on the upper pin of the capacitor (or you can add a voltmeter in parallel with the capacitor). Place a ground below the generator.

Note that placing a ground is very important when you use voltage pins, otherwise a ground is automatically added to one node of the circuit, but not necessarily to the bottom.

Connect the generator and capacitor and the other components as shown in the figure. To do this, move the cursor over the appropriate pin node until the small drawing pen appears. When the pen appears, click the left button of the mouse, draw the wire, and left click again at its end point. You can do the same on touch sensitive platforms by tapping and drawing by hand or stylus.

Note that you can draw the wire through components and TINACloud will automatically insert the components into the wire.

Finally, add the title to the schematic. Click the T button and the text editor will appear. Type in: RLC Circuit. Click the Properties button and set size 14. The Property editor also lets you choose another font, style, color etc. Now click OK to retutrt to the text editor and press OK to return the the schematic editor to position and drop the text on the schematic editor Window.

Now save the circuit with by pressing Save as... from the file menu. Name the circuit as **RLC_NEW.TSC** (the *.TSC* extension is added automatically). The circuit will be save into the My Circuits folder on the server. You can also Download it you your computer with the File.Download, upload to Google Drive or Dropbox with the File.Export and Share with other users with the File.Export commands.

4.6 Analyses

TINACloud has a variety of analysis modes and options:

The analysis method is analog when a circuit contains only analog components; then the components are modeled with their analog models.

The analysis method is mixed when a circuit contains both analog and digital components. TINACloud will analyze the analog parts in analog, the digital parts in digital, and will automatically create the interfaces among the components. This ensures synchronization and fast convergence.

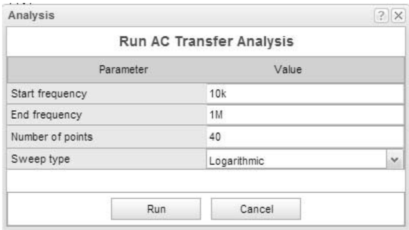
The analysis method is digital when a circuit contains only digital components; then the components are modeled with their fast digital models.

4.6.1 Analyzing an RLC circuit (DC, AC, Transient and Fourier analysis)

Execute AC and transient analyses on the RLC circuit you have just entered.

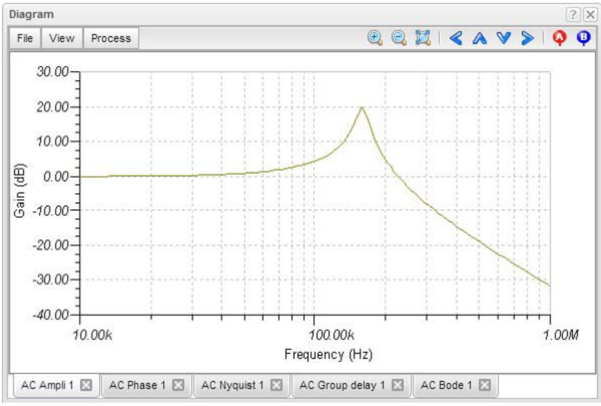
First perform an AC nodal analysis. Select **Analysis | AC analysis | Nodal voltages**. Your cursor will turn into a test probe which you can connect to any node. In a separate window the nodal voltages will be displayed. If you have placed any meters on the schematic, clicking on them with the probe will present detailed information from that instrument. Note that you can acquire DC nodal voltages in a similar fashion through DC Analysis.

Now select **AC Analysis | AC Transfer Characteristic...** from the main menu. The following dialog box appears:

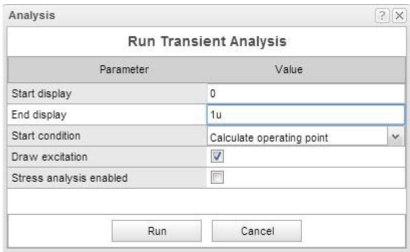


All the usual diagrams (Amplitude, Phase, will be calculated and displayed under different tabs (Amplitude, Phase, Nyquist, Group Delay, Bode) will be calculated. Set the Start frequency to 10k and then press *OK*. A progress bar will appear while the program is calculating. After the calculations are finished, the Bode amplitude characteristic will appear in the Diagram Window. You can easily switch to Nyquist or Amplitude & Phase and the other diagrams by using the Tabs at the bottom of the Diagram Window.

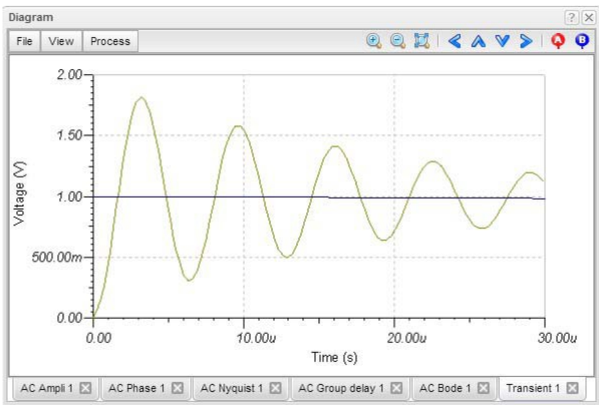
You can read exact input/output values by enabling one or more of the cursors.



Now perform a transient analysis. First, make sure your cursor is the selection arrow, then double-click on the voltage generator and change t he wavefor m to the default unit step. After selecting **Analysis | Transient Analysis**, the following dialog box appears:

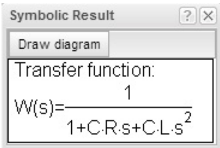


Change the *End Display* parameter to 30 u then press OK. In a separate window the transient response will appear.



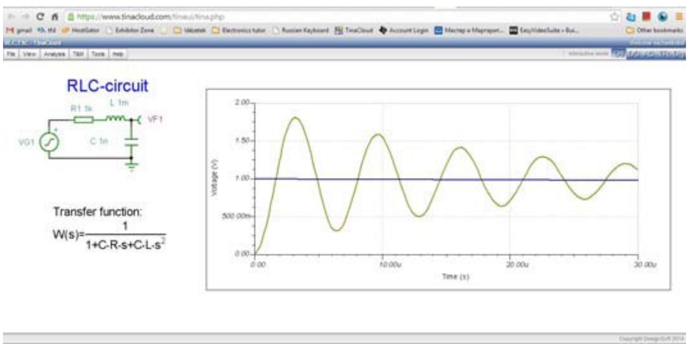
As expected, the RLC circuit exhibits a response of damped oscillation. Exact input/output data pairs can be read by enabling the A and/or A graphic cursors.

Now select **Analysis | Symbolic** or **Analysis | Semi-symbolic Transient** from the menu. The closed form expression of the circuit response appears in a window.



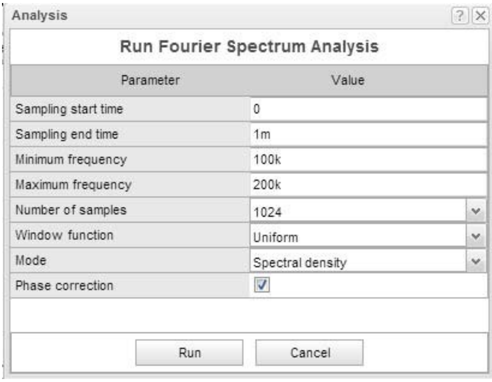
Let's add this formula and the latest diagam to the schematic design. Double-click at an empty space on the main window to invoke the Schematic Editor.

Press the green  Insert button and select "Insert Symbolic Result". The small window with the symbolic formula will appear and you can move a drop it anywhere on the schematic diagram. Similarly you can select Insert Diagram and you can add the last calculated diagram to the schematic. After saving again the resulting window should look like this.

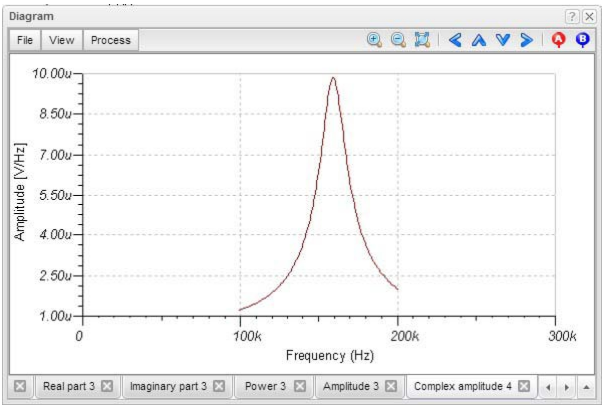


Fourier Spectrum

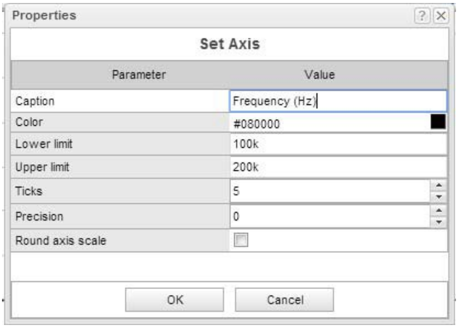
To demonstrate a more advanced feature of TINACloud, examine the **Fourier Spectrum** of the non-periodic transient response just obtained. First, in order to get a finer curve, select Analysis Select Analysis Parameters… and change the "TR maximum time step" parameter to 10n. Next run Fourier Analysis/Fourier Spectrum from the Analysis menu and set the parameters as shown on the dialog below.



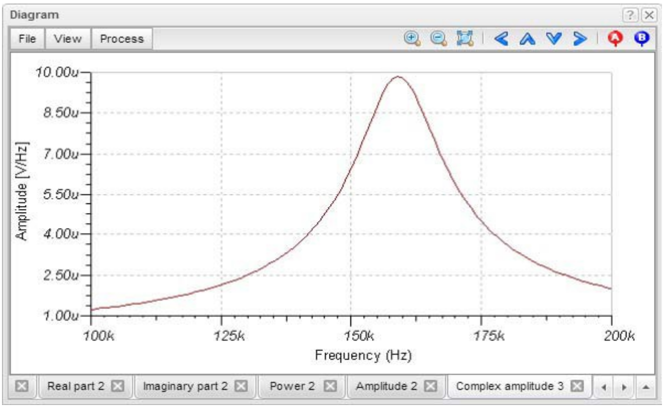
After pressing the Run button the following diagram will appear.



To get a better picture double-click the horizontal axis, set the Lower limit to 100k, the upper limit to 200k, and uncheck the Round axis scale checkbox, as shown on the dialog below.



Press OK and then the following diagram will appear:

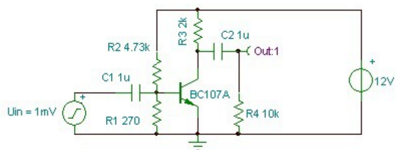


You might be surprised that the unit of the frequency spectrum is in V/Hz. That is because the continuous Fourier spectrum is a density function versus frequency. If you want to know the approximate amplitude in a narrow frequency band, you should multiply the average amplitude (given in V/Hz or Vs) with the bandwidth (given in Hz or 1/s).

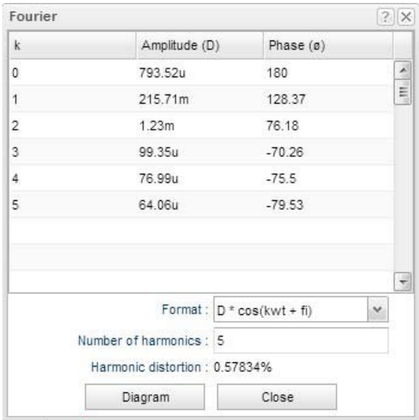
You can also find the Amplitude in V directly, if you select Spectrum in the Mode field of the Fourier Analysis dialog. In this case the applied bandwidth is 1/DT, where DT is the length of the Transient analysis (End display - Start display). This feature is especially useful if your signal contains both non-periodic and periodic components. If your signal contains periodic components, you can display them in the diagram more accurately if you select a suitable *Window function* in the Frequency Spectrum dialog. For reading the amplitude from the diagram it is the best to use the Flattop window function.

Fourier Series

Fortunately, Fourier analysis is not so complicated for clearly periodic signals. Periodic signals can be represented by Fourier Series or in other words as a sum of cosine and sine waves at the base (fundamental) frequency and integer multiples of the base frequency. To try out this kind of Fourier analysis in TINACloud, load **AMPLI.TSC** from the *EXAMPLES* folder.



Run a Fourier analysis/Fourier Series. and then select the output curve with the largest amplitude. Note that you can choose the waveform (Output) to be analyzed at the bottom of the Fourier Analysis dialog. Set Sampling start time to 1ms and the Number of samples to 2048. Note that for best accuracy, it is very important to set the starting time for the Fourier Series analysis to after the initial transient has died away. Now press *Run*. The list of Fourier components will appear.

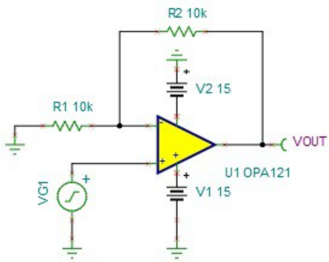


Note that you can from choose various formats (B*cos(kwt+fi), A*cos(kwt)+B*sin(kwt+fi), etc.) at the bottom of the Fourier dialog.

If you press *Diagram* you can also draw a diagram showing the amplitudes in V (volts) at integer multiples of the base frequency.

4.6.2 Creating and analyzing an Op-Amp circuit

Create the circuit diagram using an OPA121 operational amplifier from Texas Instruments as shown in the following figure:



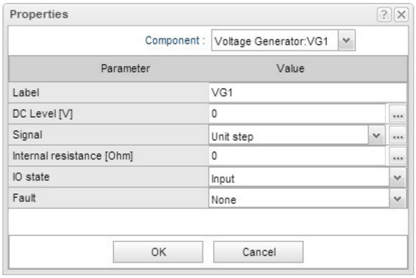
Start a new design with the File|New command.

Now start adding components. Left click on the voltage generator icon then release the mouse button. The cursor will change into the generator symbol. Position it using the mouse and click again to place it. Both while positioning or after placement you can rotate or mirror the symbol with the  and  buttons. Also if necessary you can still drag the symbol to any position with the mouse.

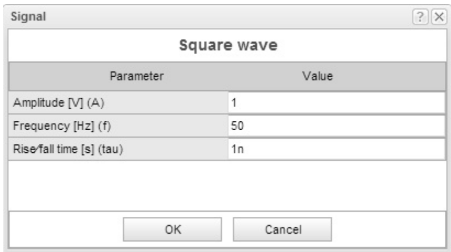
NOTE:

On touch sensitive devices you should first tap on the Symbol on the component toolbar and then tap on the editor area to place the symbol. After this you can rotate or mirror the selected symbol (displayed in red) with the  *and*  *buttons or drag with your finger or stylus.*

We still need to set the properties of this generator. Double-click on the generator and the following dialog box will appear:

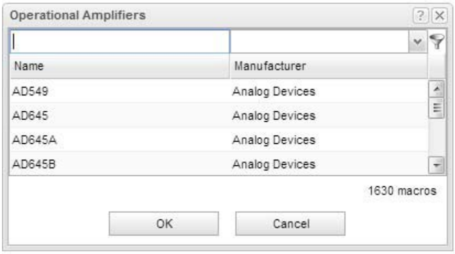


Leave the DC level, and the IO state parameters unchanged. Note that by accepting Input for the IO state parameter you have selected the output of this generator to be the input for this analysis (a Bode diagram in this example). In the Signal menu line, using the button select Square Wave from the menu and then press the button to set its parameters. In the case of the Square Wave signal, these are:



Change the Amplitude to 500m (this represents 500mV peak), the frequency to 100k (100kHz), and the Rise/Fall time to 1p (1ps). Click on OK and return to the previous dialog box and click on OK again. The program will automatically place the label (VG1) near the component and you will be able to position and place the component and the label together. If the default label position is not satisfactory, you'll be able to drag the label to the desired position later on.

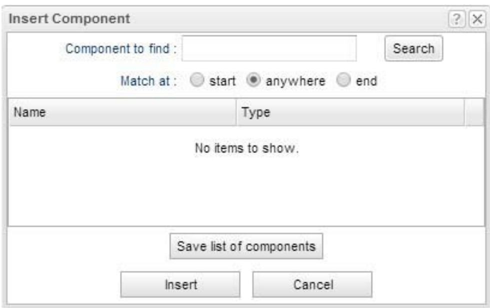
Now click on the Spice Macros tab and press the left Operational Amplifiers button. The following dialog box will appear:



To find the IC we want, scroll down the list until you find OPA121. You can narrow the list if you select the manufacturer (Texas Instrument in our example) from the Manufacturer listbox. Or you can also simply type OPA121 and the list will automatically jump to the IC and show the narrowed selection immediately as you are typing. Click on the line (OPA121) and press the OK button. The schematic symbol of this opamp will appear and be attached to the cursor. By moving the mouse, you fingers or stylus position the opamp as shown on the schematic at the beginning of the section and then press the left mouse button to place the opamp into your schematic.

You can also select a part using the  Insert button and the Find & Insert Compoment tool at the top-right corner of the Schematic Editor. If you type the part number into the "Component to find" field and press the Search button, the list of available component(s) will appear. You can enter just part of the name if you are not sure of the entire name.

Press the Insert button to place the component. With the List Component button you can create the list of all available components in a text file.

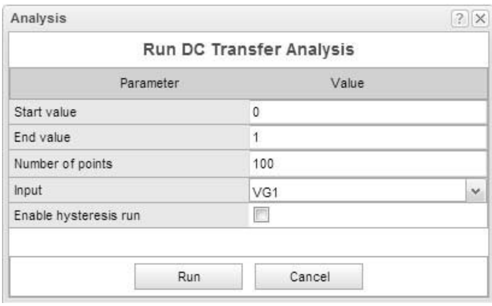


Note that other types of ICs are available under the buttons next to the Operational Amplif iers: Diffe rence Amplifiers, Fully-Differential Amplifiers, Comparators, Voltage Regulators, Buffers, Current Shunt Monitors, and Other Components). You can bring all of these various components into the dialog box for any of the buttons if you set the Show All Components checkbox. In addition to selecting an IC on the list, you can also find it by clicking on any item on the list and then typing in the name of the IC.

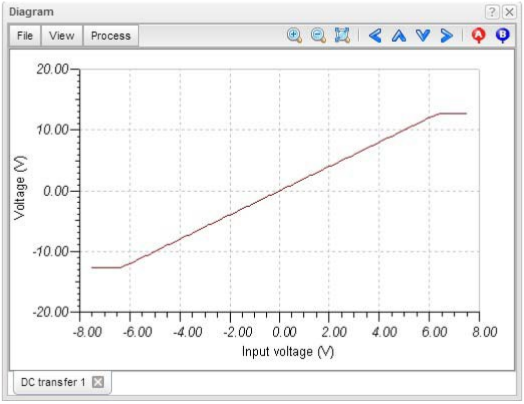
Add the remaining components and connect them with wires similarly as decribed at the previous RLC example.

4.6.3 Calculating DC Transfer Characteristic

We have already seen several of TINA's analysis modes. But so far we have not used the DC analysis mode to calculate the DC transfer characteristic of this circuit. Select DC Analysis|DC Transfer Characteristic… from the Analysis menu. The following dialog box will appear:

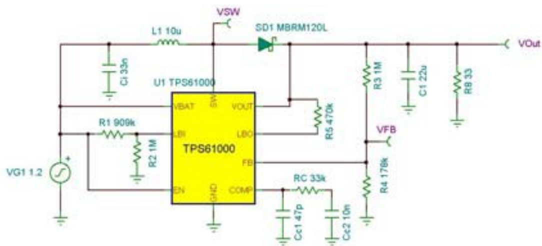


Set the Start value to –7.5, the End value to 7.5, and then press OK. After a short running time, a Diagram Window will appear as shown below. This displays the circuit's transfer curve-output voltage vs. input voltage.



4.6.4 Analysis of SMPS Circuits

SMPS or Switching-Mode Power Supply circuits are an important part of modern electronics. The heavy transient analysis needed to simulate such a circuit may take a lot of time and computer storage. In order to support the analysis of such circuits TINA provides powerful tools and analysis modes. In this Chapter we will demonstrate these through examples.

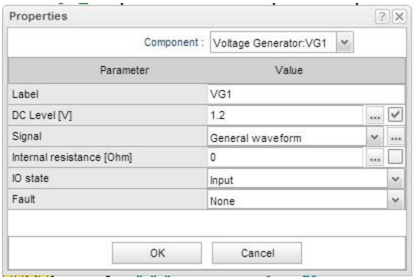


Input step analysis

One of the standard analyses for SMPS circuits is the calculation of the response to an input change to test the capability of the SMPS design to regulate the output with step changes in the input line. This can be accomplished by adding a pulse to the input voltage and checking the output and other voltages. Since the input change is relative to the steady state, we can start it from the steady state initial values calculated by TINA's steady state solver.

Load the Input Step Transient TPS61000.TSC Boost Converter circuit file from the EXAMPLES\SMPS\QS Manual Circuits folder. The schematic design is the same as above.

To see the input step waveform, double-click on the VG1 voltagegenerator on the left. The following dialog box will appear:



Properties

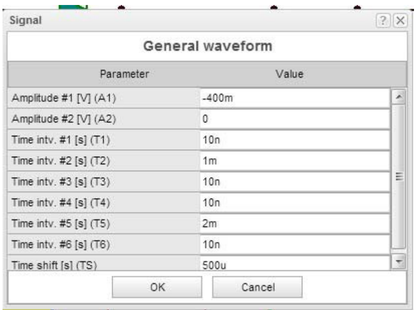
Component : Voltage Generator:VG1

Parameter	Value
Label	VG1
DC Level [V]	1.2
Signal	General waveform
Internal resistance [Ohm]	0
IO state	Input
Fault	None

OK Cancel

According this, the input voltage is 1.2V. This is converted by the SMPS circuit to 3.3V.

Now click on the Signal line of the above dialog and then the button. The following signal in the Signal Editor will appear:



Signal

General waveform

Parameter	Value
Amplitude #1 [V] (A1)	-400m
Amplitude #2 [V] (A2)	0
Time intv. #1 [s] (T1)	10n
Time intv. #2 [s] (T2)	1m
Time intv. #3 [s] (T3)	10n
Time intv. #4 [s] (T4)	10n
Time intv. #5 [s] (T5)	2m
Time intv. #6 [s] (T6)	10n
Time shift [s] (TS)	500u

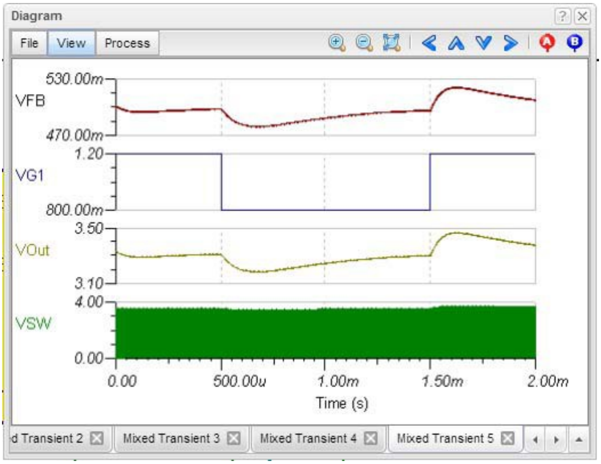
OK Cancel

According to the waveforms, the input voltage will decrease from 1.2V to 0.8V for a T2=1ms time; and the starting edge (T1) and the ending edge (T3) of the pulse are 10us.

To see the response of the circuit, let's invoke and run the Transient analysis from the Analysis menu.

NOTE:

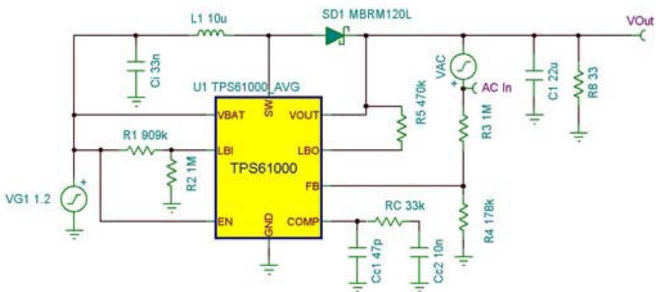
To draw each curve in a separate coordinate system you many need to click View and then Separate curves in the Diagram tool.



AC analysis

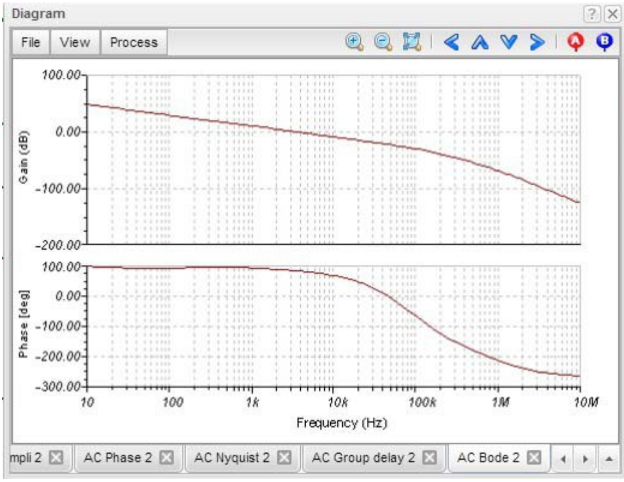
For AC analysis and stability analyses you can use the so called Average models provided in TINA. The average models represent a method, based on averaging the effects during the switching process. The resulting equations are linear therefore the method is extremely fast in order to draw Bode and Nyquist plots needed for stability analysis. Note that for using the AC analysis function of TINA you need an average model, the transient models are not applicable and will give improper results.

To demonstrate this tool, let's load the Average model TPS61000.TSC circuit file from the EXAMPLES\SMPS\QS Manual Circuits folder.



Note the VAC generator which is providing signal for the AC analysis., and the AC In Voltage pin which is the Input of the AC analysis (its IO state parameter is set to Input).

Let's run AC Analysis/AC Transfer Characteristic… from the Analysis menu and see the result.



4.6.5 Stress Analysis

Stress Analysis can check parts for stress conditions such as maximum power dissipation and maximum voltage and current limits. You can set these parameters in the property window of the parts or in the catalog. This kind of analysis is also called Smoke analysis, because overloaded parts often emit smoke.

Note: Stress Analysis is available in the Industrial version TINA and TINACloud .

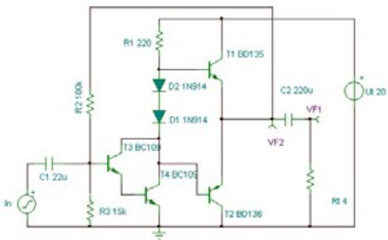
In TINACloud you can use Stress Analysis during DC or Transient analyses.

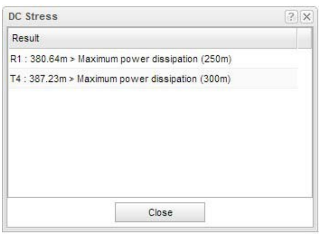
For DC analysis you can start Stress Analysis by selecting the Stress Analysis command on the Analysis menu. The program will calculate the DC operating point with Stress Analysis enabled.

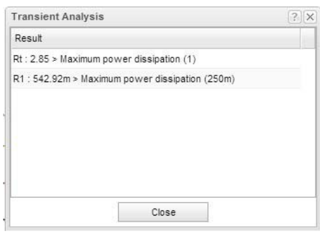
When running DC or Transient Analysis from the Analysis menu, a list of components will appear, along with the parameters exceeding maximum limits.

The maximum values of the components can be set in the component property dialogs or in the component catalog parameter dialogs. Both can be entered by double-clicking on the components. Before running an analysis, check and set the maximum values of the components in your circuit.

As an example of Stress Analysis, open the file Stress Analysis.TSC from TINA's EXAMPLES folder and run DC Analysis / Stress Analysis and Transient Analysis from the Analysis menu. At Transient Analysis make sure that the "Stress analysis enabled" checkbox (visible in the Industrial version only) is checked. In the following figures, you can see the result of a DC and a Transient Stress Analysis.



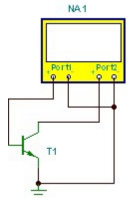




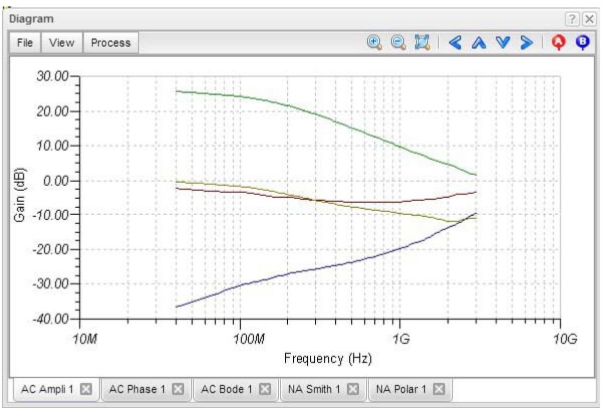
Apparently the power dissipation of T4, Rt and R1 exceed maximum limits allowed for these parts.

4.6.6 Network Analysis

TINA helps you perform network analysis and determine the two-port parameters of networks (S, Z, Y, H). This is especially useful if you work with RF circuits. Results can be displayed in Smith, Polar, or other diagrams. You can assign the two ports needed for Network Analysis with the Network Analyzer component to be found on the Meters component toolbar. As an example open the circuit EXAMPLES\RF\SPAR_TR.TSC.



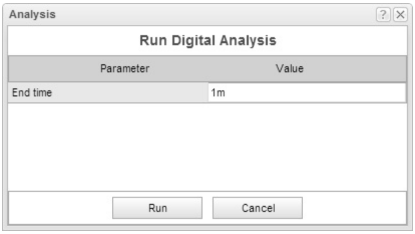
To analyze this circuit run Analysis/AC Analysis/Network Analysis. The amplitude diagram is as follows:



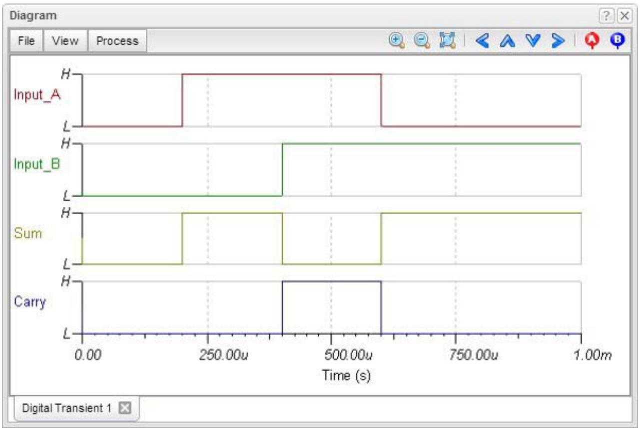
Note that we have added the labels to the curves using the  Auto label tool of the diagram window. For more details on the Network Analysis see the "Network Analysis and S-parameters" chapter of the advanced topics manual.

4.6.7 Analyzing a Digital Circuit with TINA's Digital Engine

Let's test a digital circuit using timing Analysis.
Selecting the *Analysis|Digital…* command, brings up this menu:



The result is shown on the timing diagram below.



You could also select *Transient...* instead of *Digital Timing Analysis*, in which case the program would carry out an analog analysis, giving the detailed continuous waveforms and voltages instead of idealized logic levels. Note that circuits which contain only digital components can be analyzed by both digital and analog methods.

NOTE:

You can set the order of the curves by simply appending a colon (:) character and a number to the output name. This is particularly important when presenting the results of digital analysis, where each output is displayed as a separate diagram. For example, if you have outputs named OutA, OutB, Carry, and Sum, you can ensure that they will be displayed in the order given by using the labels OutA:1, OutB:2, Carry:3, and Sum:4.

The results of a purely analog analysis normally appear in one diagram: however, you can force TINA to display the results as separate diagrams, in the order you desire, by using the labeling method described above. You must use the View | Separate Curves command in the Diagrams window to separate the curves. If you don t use this labeling method, TINA presents the curves in alphabetical order.

4.6.8 Analyzing a Digital Circuits using Digital HDL Simulation Models

Hardware Description Languages (HDL) are standard text-based modeling languages used by electronic designers to describe and simulate their chips and systems prior to fabrication.

TINACloud now includes the four most widely used Hardware Description Languages defined by IEEE standards: VHDL, Verilog, Verilog-A and Verilog-AMS.

VHDL and Verilog are used for modeling digital circuits. The two languages are comparable in modeling digital hardware. However the behavioral capabilities of VHDL are more powerful, while Verilog is easier to learn and understand. In TINACloud you can use and mix models of both languages.

Verilog-A is an easy to read high-level behavioral language for modeling analog electronic circuits and devices (e.g., bipolar and MOS transistors).

Verilog-AMS is an extension of Verilog for modeling analog and mixed signal circuits allowing both Verilog and Verilog-A instructions, connect modules, and rules.

A full presentation of HDLs in TINACloud is beyond the scope of this manual. We refer the interested reader to the detailed standards, manuals and infor mation on the Internet:

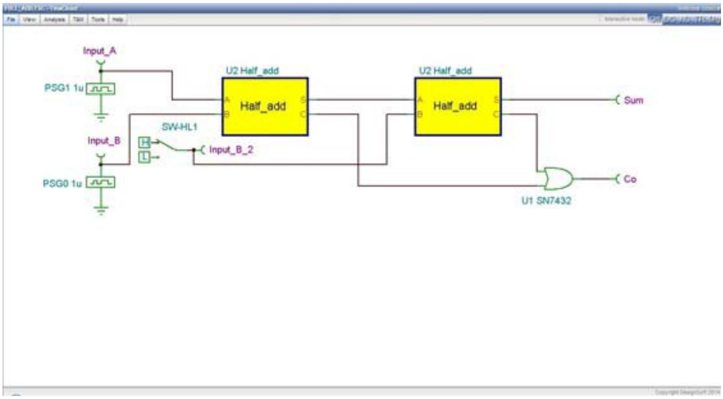
www.hdl.org and www.verilog.org

In the following sections we will demonstrate the use of these languages through examples.

4.6.8.1 Analyzing a Digital Circuit Using Digital VHDL Simulation

TINACloud includes a powerful digital VHDL simulation engine. Any digital circuit in TINACloud can be automatically converted into VHDL code and analyzed as a VHDL design. In addition you can analyze a wide range of hardware available in VHDL and define your own digital components and hardware in VHDL. The great advantage of VHDL is not only that it is an IEEE standard hardware description language, but also that it can be realized automatically

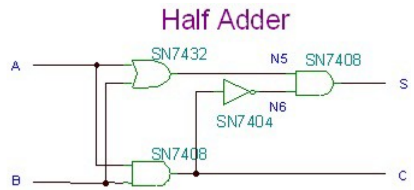
in programmable logic devices such as FPGAs and CPLDs.
Before realizing a VHDL design, either with discrete components or with FPGA, verify it with simulation using TINACloud's *Analysis|Digital…* command. Let's examine some aspects of the VHDL simulation.
To do our first VHDL analysis, Open the FULL_ADD.TSC circuit from the EXAMPLES\VHDL folder. The following circuit will appear:



This circuit is a combination of two VHDL half adder blocks (macros) and a discrete OR gate.
Note that the essential VHDL code of the half adder is at the bottom and it is only

```
S <= ( N5 AND N6 );
N6 <= NOT ( C );
C <= ( A AND B );
N5 <= ( A OR B );
```

At first glance, the code may look a bit strange, but it in fact is a machine translation of our half adder, assembled from gates in 4.6.1. Introducing the node names N5 and N6 as shown on the figure below, it is clear that




```
C <= ( A
AND B );
N6 <= NOT
( C );
N5 <= ( A OR B );
```

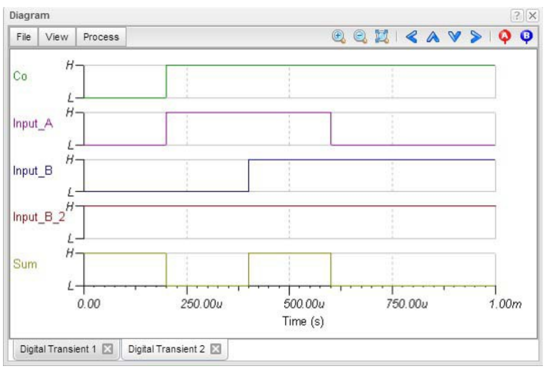
and therefore

```
S <= ( N5 AND N6 );
```

You might find it odd that in the VHDL code in the box, S appears to be calculated from N5 and N6 even before N5 and N6 have been calculated. This is valid, however, because VHDL is a concurrent language, and the order of the lines does not mean the order of execution.

The delays are taken from the given discrete values. But if the target hardware is an FPGA the synthesizer program will use the delay values of the FPGA data sheet.

Now select Digital VHDL Simulation from the Analysis menu and press OK. The following diagram will appear:



A great feature of TINACloud's VHDL is that you can not only view the VHDL code of each component, but you can edit and run them immediately. Let us replace the 4 line VHDL code -

```
S <= ( N5 AND N6 ); N6 <= NOT ( C );
C <= ( A AND B ); N5 <= ( A OR B );
```

with this simpler 2 line code

```
S <= (A xor B); C <= (A and B);
```

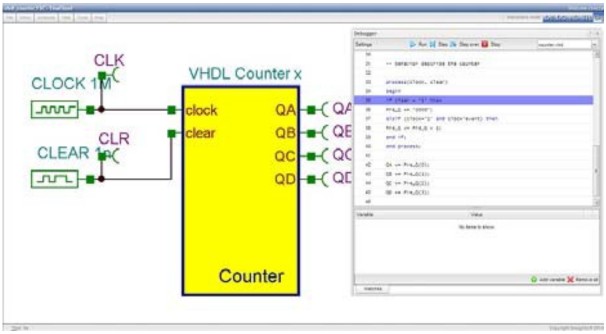
This is easier to understand. In fact, if one of the A or B inputs is true, S is True. (A and B). We recognize this as an Xor function. Now select *Digital…* from the Analysis menu, and press OK. The diagram that is drawn will be practically identical to the previous diagram.

4.6.8.2 The HDL Debugger: Debugging VHDL and Verilog codes

Debugging HDL programs is especially hard because of the concurrent processes in these languages.
A great feature in TINACloud is that the HDL debugger is now integrated. You can:

- Execute VHDL and Verilog codes statement-by-statement (Step)
- Execute subprograms as a single statement (Step Over)
- Add breakpoints (Toggle Breakpoint), running continuously (Start) and stopping at the breakpoints.
- Place variables, signals and other objects under the Watches tab and see their value during debugging.
- View all breakpoints and objects under the Breakpoints and Locals tabs at the bottom of the HDL debugger window.

To practice the use of the HDL debugger in TINACloud, open the vhdl_counter.TSC file from the EXAMPLES\VHDL folder with the Open command of the File menu. Next, click the Analysis menu and enable the debugger by clicking the Enable HDL Debugger line. Finally press, the DIG button on the toolbar at the top of the screen or click Start on the Interactive menu. The HDL Debugger will appear. Click the counter.vhd tab at the bottom of the code.
You should see the following screen:



You will see two modules on the toolbar: *counter.vhd* and *vhdl_counter_comp.vhd*. The first is the contents of the *VHDL Counter* macro, while the second file is the VHDL conversion of the whole circuit including the sources.

This macro implements a counter. The counter entity consist of five processes, and all processes run in parallel. The first process is sensitive to the clock and the clear signal. So when one of these signals is changing, this process is triggering and executing. The other processes are sensitve to the Pre_Q signal. When Pre_Q(i) changes, the *i+1th* process is triggering and executing.

There is a similar example in Verilog called verilog_counter.tsc in the Examples\Verilog folder.

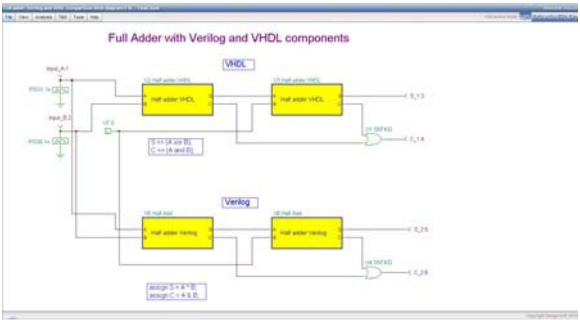
4.6.8.3 Analyzing a Digital Circuit Using Digital Verilog Simulation

TINACloud also includes a powerful digital Verilog simulation engine. The advantage of Verilog compared to VHDL that it is easier to learn and understand, however there are more features in VHDL.

Verliog -similarly to VHDL- can also be realized automatically in programmable logic devices such as FPGAs and CPLDs.

Before realizing a Verilog or any other digital HDL design, either with discrete components or FPGA, you need to verify it with simulation with TINACloud's Analysis|Digital… command.

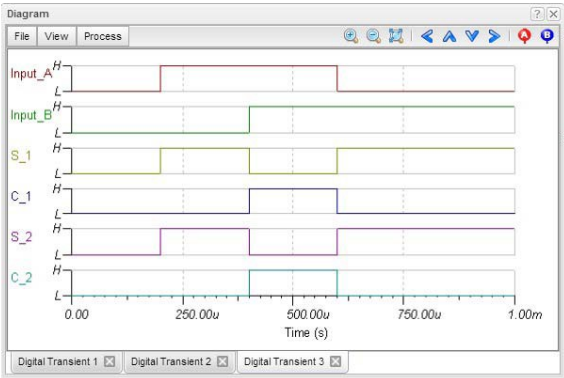
Let's run the previous VHDL circuit along with its Verilog model. the Full adder Verilog and VHDL comparison time diagram.TSC circuit from the EXAMPLES\Verilog folder. The following circuit will appear:



You can see realization of the half adder function in both languages, they are very similar. You can click the VHDL or the Verilog macros and to see all the details.

Now run the Digital Timing Analysis from the Analysis menu. The following diagram will appear:

You can see that the output signals from both models are exactly the same.



4.6.8.4 Analyzing Circuits Using Verilog-A models

Today the most widely used language to describe electronics circuits and device models is the Spice netlist format (1973). However the Spice netlists are often hard to read and understand, and they lack a lot of the functionalities of prog ramming languages which engineers would need while creating models and simulation.

The relatively new Verilog-A language (1995) provides an alternative method with an easy to read programming language style C like syntax. Thus Verilog-A is a suitable successor of the SPICE netlists for describing circuit topologies.

Most of the device libraries of TINACloud are in Spice netlist format. However you can already create and import models and place TINACloud macros in Verilog A format. You can find several language examples, device models, and circuits in the Examples\Verilog A folder of TINACloud.

For a demonstration of Verilog-A in TINACloud, load the examples in the EXAMPLES\Verilog-A folder.

We suggest that you start with the Opamp Model Comparison.TSC file where a simple opamp model is realized in three different ways: Verilog-A, Spice, and the schematic diagram.

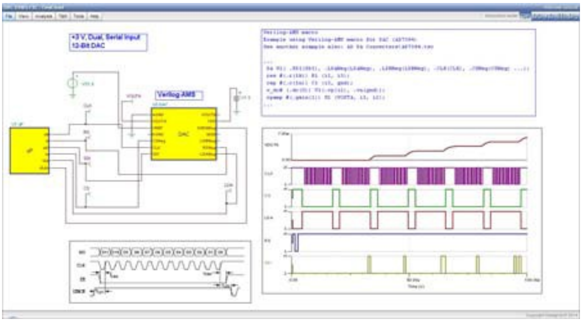
You can also study nonlinear device models in Verilog-A and their characteristics in the other examples: diode.TSC, JFET.TSC etc.

4.6.8.5 Analyzing Circuits Using Verilog-AMS models

An even more sophisticated method of describing electronics circuit, containing both analog and digital components is the Verilog-AMS language. As we observed earlier, Verilog-AMS is a derivative of the purely digital Verilog extended with the purely analog Verilog A and an interface for the connection of the analog and digital parts.

In TINACloud you can also create or import Verilog AMS macros for modelling mixed signal devices.

Let's see the structure of such a model. Open the DAC VAMS.TSC circuit from the EXAMPLES\Verilog AMS folder. The following circuit will appear.



This circuit contains a Digital Analog Converter (DAC) macro with Serial Peripheral Interface (SPI) and a test bench macro, generating the digital SPI signal.The DAC model is defined in Verilog AMS. Interestingly, test bench on the left side is written in VHDL which is an example of mixing different HDLs but here we will concentrate on the Verilog AMS macro on the right.

To see the Verilog AMS code of the model click the DAC macro. The following window will appear.

We will not go into a detailed analysis of the code. We just want to show that in the first part shown above, the DA Verilog module converts the serial signal into an analog signal (VOUTA).

At the end of the macro shown below (you can scroll down there), the DA module is called and the signal is smoothed by a simple opamp and an RC filter using Verilog A instructions. You can also see the definition of the capacitor in the code fragment below.

4.6.9 Mixed Mode Simulation (Spice - VHDL - MCU co-simulation)

TINACloud include a new powerful mixed mode simulation engine. It is based on the XSPICE mixed mode algorithm, extended with MCU and VHDL components. In your circuits you may freely mix any analog or digital components of TINACloud, including microcontrollers (MCUs) and macros with Spice or VHDL content.

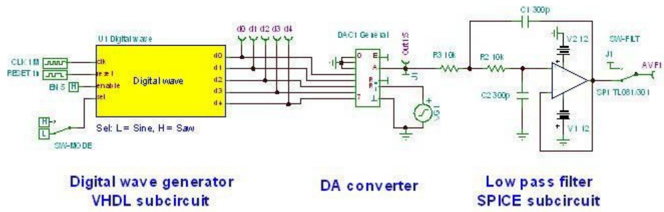
You can modify these components on the fly along with the code in the MCUs. TINACloud will analyze the analog parts in analog, the digital parts in digital, and will automatically create the interfaces among the components. This ensures synchronization and fast convergence.

Let's explore some of the uses of this mode through a few examples.

4.6.9.1 Waveform generation with a VHDL and Spice subcircuits

The following circuit (EXAMPLES\VHDL\Mixed\Wave generator.TSC) generates an analog sine or sawtooth signal depending on the status of the left SW-MODE switch.

GETTING STARTED



The Digital Wave box on the left of the circuit includes a VHDL code with a lookup table *Sine_LUT* for the sine wave and a counter for the sawtooth signal. The essential part of the VHDL code is:

```
process(Reset, Clk)
begin
      if (Reset = '1') then
      Wave <= (others => '0');
      LUT_index <= 0;
   elsif rising_edge(Clk) then
      if (Enable = '0') then
      Wave <= (others => '0');
   elsif (Sel = '0') then
      Wave <= Sine_LUT(LUT_index);
  else
      Wave <= conv_std_logic_vector(LUT_index,5);
  end if;
             if (LUT_index = LUT_index_max) then
             LUT_index <= 0;
      else
             LUT_index <= LUT_index + 1;
      end if;
      end if;
end process;
  d0  <=  Wave(0);
  d1  <=  Wave(1);
  d2  <=  Wave(2);
  d3  <=  Wave(3);
  d4  <= Wave(4);
```

You can see all the details of the code and modify it if necessary by clicking the Digital Wave box.

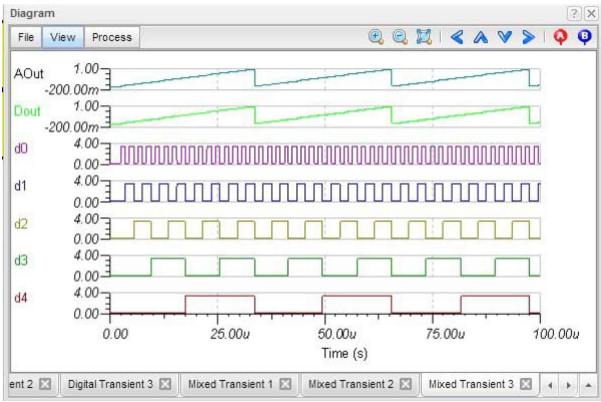
Note that the model is set to TTL in this dialog, but you may select from various other models (CMOS, LS, HC, HCT etc.).

The digital output of the counter is converted into an analog signal in the 5 bit DA converter of TINACloud shown in the middle of the circuit.

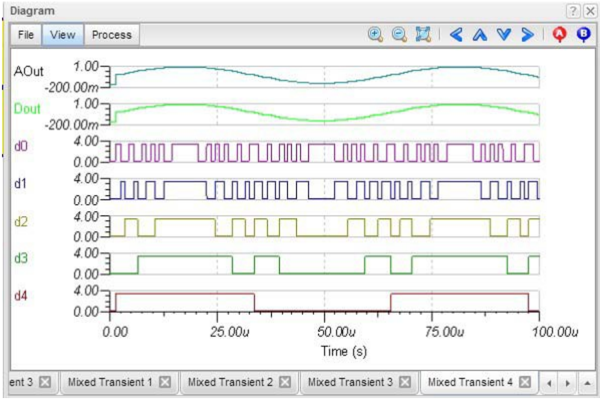
The DAC sine wave output needs to be cleaned up with a low pass filter. We will use a Spice opamp model of the TL081 in a Sallen and Key low pass filter configuration. Press the Enter Macro button on the property dialog and TINACloud will open the macro. You can review and, if necessary, modify the Spice code inside the macro.

The sawtooth signal from the counter output (on pin J1) does not need to be filtered, so we will connect it directly to one terminal of switch SW_FILT. The sine wave developed at the DAC output (DAEX) does, in fact, require filtering, so we will pass it through the low pass filter and connect the filtered Aout analog output to the other terminal of SW_FILT. A jumper (J1) connects the DAEX output to the switch. Although it's not obvious in the schematic, the switches SW_FILT and SW_MODE are synchronized as though they were a DPDT switch. We cause them to be synchronized by assigning both switches to be controlled by the Hotkey A.

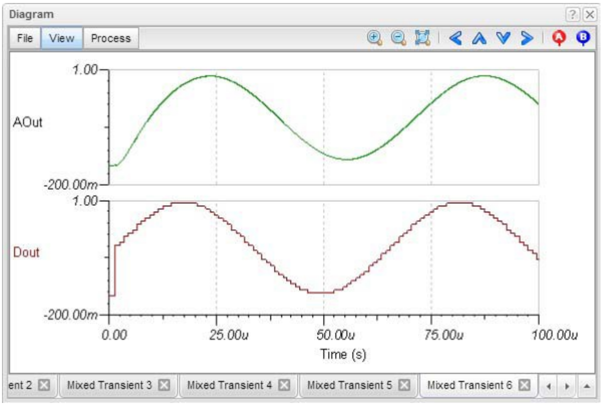
Here are the final waveforms of the full circuit, including the five counter output wavefor ms. SW_MODE is in the High state, selecting the sawtooth signal.



If we change the SW-MODE switch to Low and run Transient analysis again, the waveforms are:



To see the effect of the analog filter, delete curves d0 to d4 from the diag ram. Alternatively, you can delete outputs d0 to d4 temporarily and run Transient Analysis again.

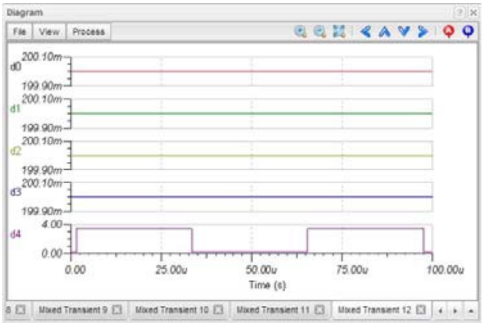


To demonstrate the flexibility of TINACloud's VHDL features, we'll modify the VHDL code to generate a square wave instead of the sawtooth waveform. Simply set Wave(0) to Wave(3) to zero in the VHDL code.

Double-click the Digital Wave macro and press the Enter Macro button. Locate the Wave <= conv_std_logic_vector(LUT_index,5) line and insert the following statements:

```
Wave(0) <= '0';
Wave(1) <= '0';
Wave(2) <= '0';
Wave(3) <= '0';
```

Now you can run Transient… from the Analysis menu to get the following waveforms.

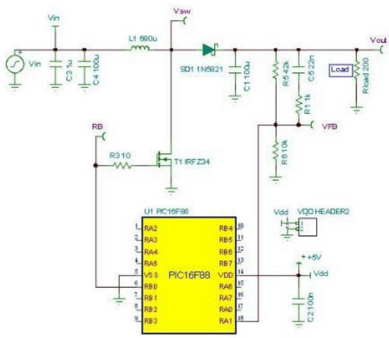


You can check out a more complex version of this circuit under EXAMPLES\VHDL\Mixed\Wave generator dipsw.TSC. There you can select all the three waveforms we discussed using a dip switch.

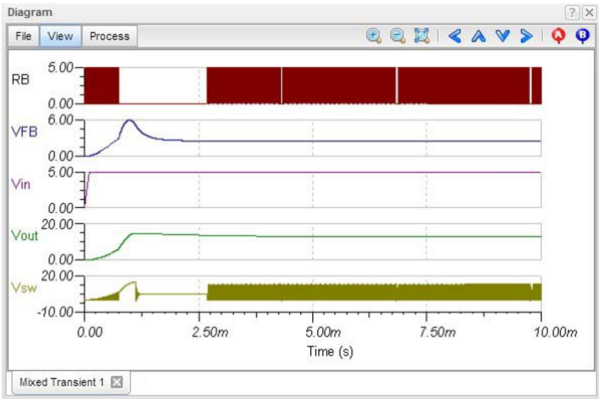
Note that you can download the VHDL portion of the code into an FPGA and use hardware form.

4.6.9.2 MCU controlled SMPS circuit

The mixed mode simulator of TINACloud not only allows MCUs, but also any linear or nonlinear parts in TINACloud's libraries. As an example, let's study the following circuit, which realizes a DC-DC converter, converting 5V DC to 13V DC, and operating in boost mode. You can find this circuit in TINACloud under EXAMPLES\Microcontrollers\Pic\ Boost_converter.TSC .

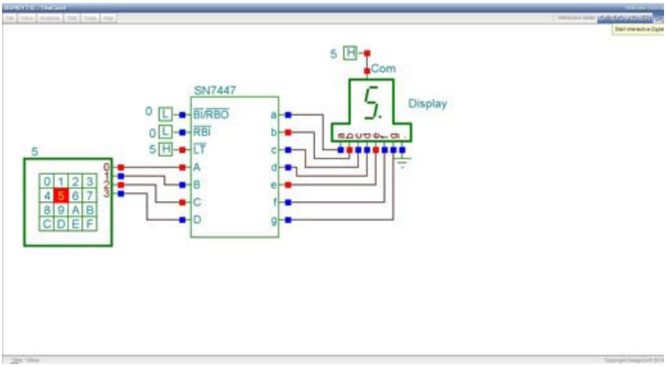


The PIC MCU in the circuit produces a PWM output at pin RB0 that controls the switching FET. The interrupt routine of the code in the PIC compares the feedback voltage at VFB (connected to pin RA1 of the PIC), with a built in threshold voltage. If the voltage is lower than the threshold defined in the code, the duty cycle of the PWM output waveform is increased. You can study the ASM code in the PIC by clicking on the PIC, and finally select the ASM code field. You can see and debug the code on the fly here. Click on the Enable MCU code debugger line under the Analysis menu, press the TR interactive transient analysis button.
The waveforms below demonstrate how the analog parts and the MCU interact in TINACloud.



4.6.10 Testing your Circuit in Interactive mode

When everything is in order, the ultimate test of your circuit is to try it in a “real life” situation using its interactive controls (such as keypads and switches) and watching its displays or other indicators. You can carry out such a test using TINACloud’s interactive mode. Not only can you play with the controls, but you can also change component values and even add or delete components while the analysis is in progress. The interactive mode is also very useful for educational and demonstration purposes, for tuning circuits interactively and for interactive circuits which you cannot test otherwise, e.g., circuits with switches, relays, or microcontrollers. You can start the interactive simulation with the pressing the DC, AC, TR or DIG button. Press the Off button to stop the interactive simulation. Now the displays and indicators in your schematic will reflect whatever you do with the controls. In addition to displays, TINACloud has special multimedia components (light bulb, motor, LED, switch, etc.) which respond with light, motion and sound. Let's see a few examples.

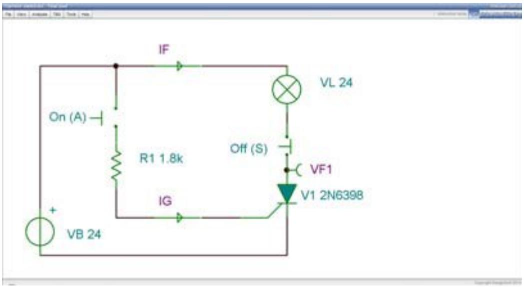


4.6.10.1 Digital Circuit with a Keypad

To try out the interactive mode, load the **DISPKEY.TSC** circuit from the *EXAMPLES\Multimedia* folder. The circuit is shown below. Press the DIG button (the button will turn light green).

4.6.10.2 Light Switch with Thyristor

Open the T hyristor switch example, TSC circuit from the EXAMPLES folder and press the TR button. You will see the following screen:

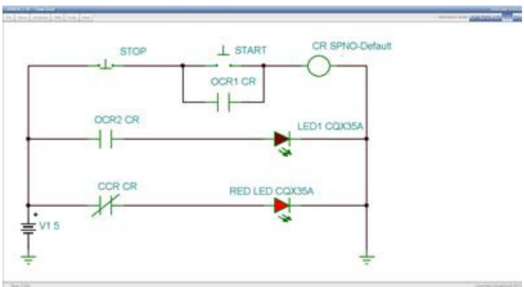


Press the On push button (Wait until the cursor turns into a vertical arrow) to turn on the light. The Thyristor will turn on and remain on even after the push button is released. So will the light. You can turn off both the Thyristor and the light bulb by clicking on the Off push button . In both states of the circuit, you will see the currents shown by the two ammeters.

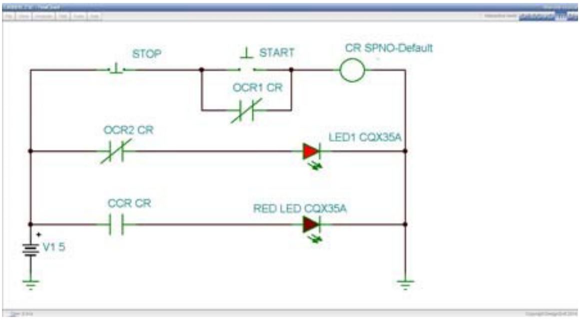
4.6.10.3 Ladder Logic networks

Another version of a self holding circuit, this one based on ladder logic, can be found in the LADDERL.TSC circuit file in the EXAMPLES/Multimedia folder.

Initially the red LED will light. If you click on the START button (click when the cursor changes into a vertical arrow), OCR1 will close and stay closed (since the current flowing through OCR1 will keep magnetizing the relay coil CR). Now the LED1 will light, OCR2 will open, and the red LED will turn off. If you now click on the STOP button, you will break the self holding circuit and the relay CR will release, the red LED will light again, and the LED1 will turn off.



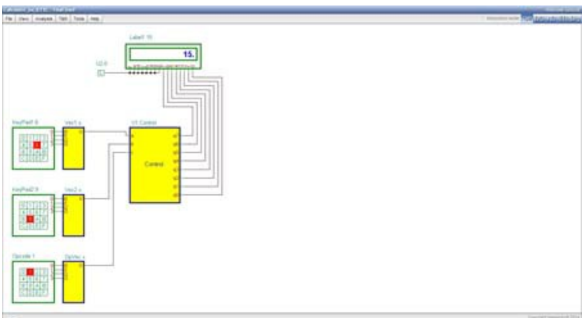
Ladder logic: Initial state or after clicking the STOP button.



State after clicking the START button

4.6.10.4 HDL Circuits

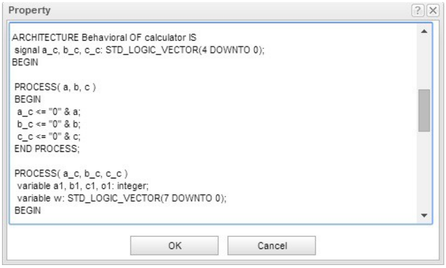
A great feature of TINACloud is that you can not only test but also modify
HDL circuits on the fly including, the HDL code itself. Let's see this with the example Calculator_ex_8.TSC in TINACloud's Examples\VHDL\Interactive folder.



This is a special calculator circuit controlled by the Opcode keypad. For the Operation codes 1, 2 , 3 and 4, it realizes a basic four func- tion calculator, complete with +, -, /, and * basic arithmetic opera- tions. Further operations can be added through modifying the VHDL code inside the Control unit. First press the DIG button; as the Opcode is 1, you should see 4+2=6 on the LCD display. Try the other Opcodes with different settings on KeyPad1 and KeyPad2.

Now let's implement the op1*op1 operation. Click on the Control box and press on HDL code. The VHDL code of the component will appear.

```
Add when 5 => o1 := a1*a1; line before WHEN
OTHERS => o1 := 0; line
```



4.6.10.5 Microcontroller (MCU) Circuit

To test circuits with prog rammable devices requires special development software that permits a high degree of interactivity. This calls for debugging software that can test the code running in the device step-by-step.

You can see, modify, and debug the program running in any of the supported processors, and, of course, you can make and run your own code.

There are 4 ways of providing the program for microcontrollers in TINACloud. You can:

1) use the binary code and debug file made by any standard compiler (e.g., MPLAB for PICs),

2) load your assembly code to run and debug directly in TINACloud using its built in assembler-debugger,

3) write your MCU code in C, install a C compiler which generates the code for the MCU you want to simulate, (TINACloud will automatically integrate it into its C code debugger),

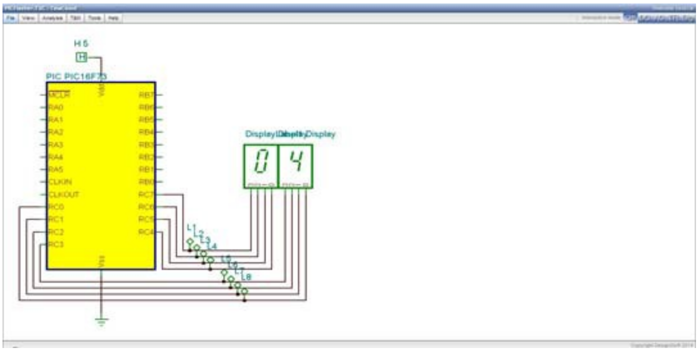
4) or finally use the built in Flowchart editor in TINACloud to generate and debug the MCU code.

3-4) you can do it in offline version of TINACloud. The simulation is available on TinaCloud.

To load the code into the MCU, zip the source files and upload it. To do this, select File|New…, select an MCU from the toolbar. Click Save. Click on the MCU, select MCU-code and select Choose Files, here select your previously zipped file (for example if you have an asm file, zip it and select it in this dialog).

4.6.10.6 Using the ASM Debugger

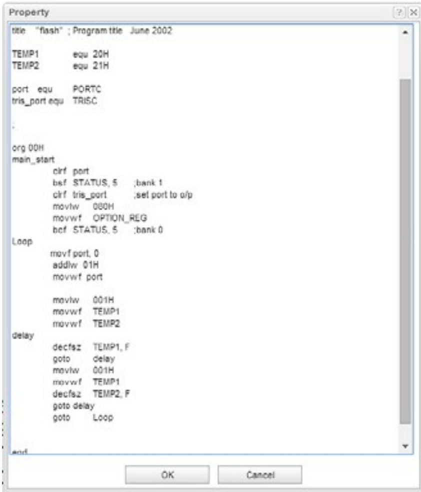
Now let us run a microcontroller application and see how to test and modify its code. Load the PICFlasher.TSC circuit from the Examples\Microcontrollers\PIC folder. The following schematic will appear with the 16F73 PIC microcontroller.



This circuit is simply counting forward one count at a time. Press the DIG button to see how it works. The display should step forward one-by one.

Now let's press the Off button and modify the code to count by 2. Click the MCU, and select MCU code.

The ASM code of the MCU will appear.



Now lets make the following change in the code. Change the instruction (selected above) in line 25 (you can see the line number in the right bottom corner of the code editor window) from

```
addlw 01H
```

to

```
addlw 02H
```

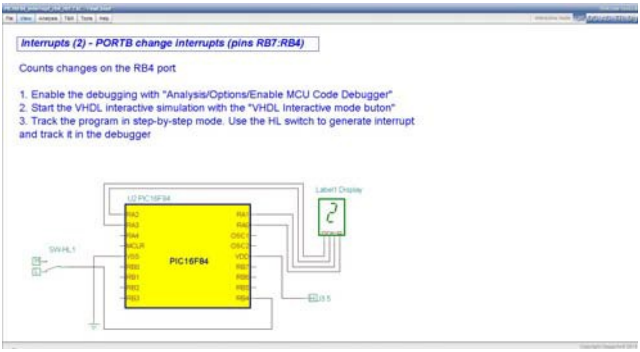
Save the changed code. If you press the DIG button, now the increment will be 2!

Note that the changed code will be automatically saved in the schematic file.

TSC example fr om the Examples\Microcontrollers\PIC folder.

4.6.10.7 Example PIC Interrupt handling

Now let's see another application with some more interactivity. Load the PIC16F84_interrupt_rb4_rb7.TSC example from the Examples\Microcontrollers\PIC folder.



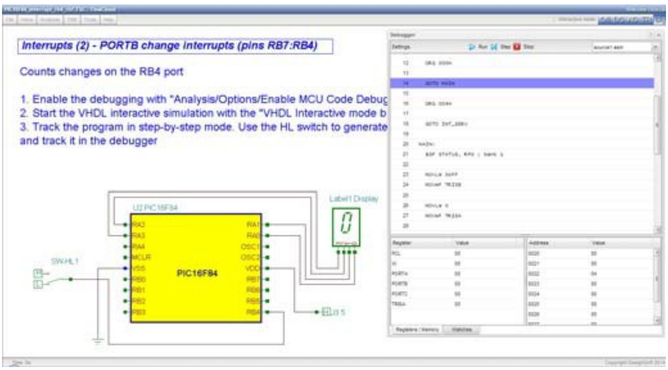
Press the DIG button. At first glance, it appears that nothing is happening.

However, if you click on the SW-HL1 switch, the display will step forward by 1 each time that the switch changes from Low to High. This is realized with the interrupt handling capability of the PIC16F84.

Now let's see the operation in more detail using the interactive ASM debugger.

To activate the debugger, select the "Enable MCU Code debugger" option on the Analysis menu.

The MCU debugger will appear if you press the DIG button:



Here is a short description of the MCU debugger dialog.

On the top line there are the following controlling icons:

 Run the code in the debugger continuously. The lines being executed will be highlighted and the code is scrolled.

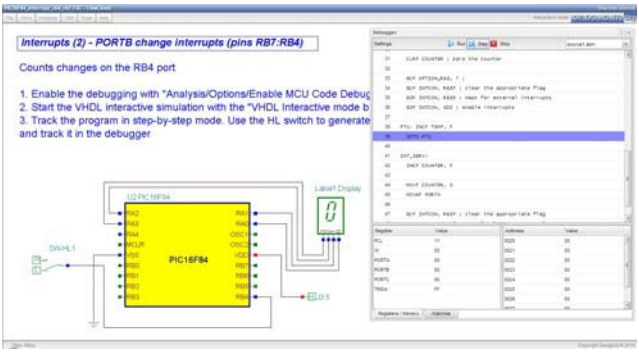
Step forward. Step by step execution. Each time you press this button one command of the program is executed.

Stop Halts program execution.

The Code window (below the control icons) displays the ASM code. The next actual command is highlighted with blue.
The actual content of the registers and memory locations of the MCU are shown in the lower part of the screen.

Let's follow the program execution step-by-step by pressing the Step forward button. After around 14 clicks, we get to the PT1: label, where the program seems to be in an infinite loop.

```
PT1: INCF
TEMP, F GOTO
PT1
```



Now click on the SW-HL1 switch and change it to High. (You should click when the cursor changes into an upward pointing arrow Î).

Return to the Debugger and click the Step forward button twice. The prog ram will recognize the interrupt and jump into the INT_SERV: label.

```
INT_SERV:
INCF COUNTER, F
MOVF COUNTER, 0
MOVWF PORTA
```

increment the COUNTER, and copy it to PORT A. The output will now be 1. After this, the program will return to the "infinite loop" at PT1.

4.6.10.8 Making a Breakpoint in ASM

It is often essentially impossible to get to a certain place in the program since you'd have to single step a thousand times (if the program ever steps there in the first place). To get the program to run to a particular statement and halt there, you can tag the statement as a so called breakpoint. Now run the program in the Debugger's continuous mode using the Run command and the program will stop at the marked space before execution of the marked command. To demonstrate this, click on the increment statement in our interrupt service routine after the INT_SERV: label and click on the left side of the 'INCF COUNTER, F' statement. A breakpoint automatically toggled.

Now press the Run button. The program starts to run and falls into the "infinite loop."
Even though you have set a breakpoint, the code will not stop since it does not pass the breakpoint. However, when you change the switch from Low to High the program will stop at the

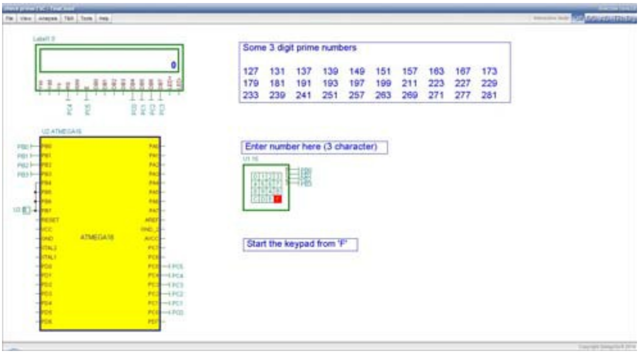
```
INT_SERV:
INCF COUNTER, F
```

statement. Now you can resume execution either step by step or with the Run command again.

4.6.10.9 Debugging C code in MCUs

Just as with ASM and HEX code, you can follow the execution of a C program and even watch the values of the required variables.

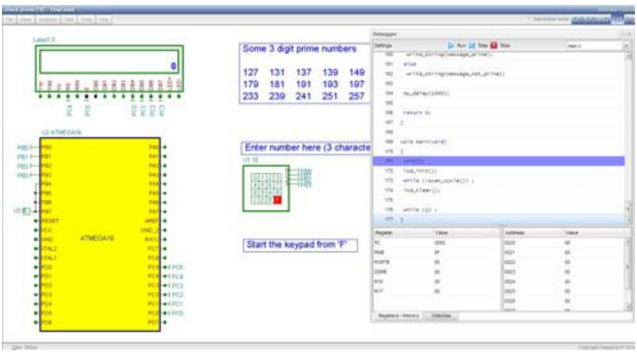
To demonstrate this, let's open the "check_prime.TSC" file in the Microcontrollers\C compiler\AVR folder. The following circuit will appear:



To test this circuit, press the TR button and enter a 3 digit number (each digit must be different). The display will show “Prime number” or “Not prime.”

Now to debug this C program, press the Off button and then click the “Enable MCU Code debugger” in the Analysis menu and then press the TR button again.

The C code debugger window (MCU IDE) will appear.



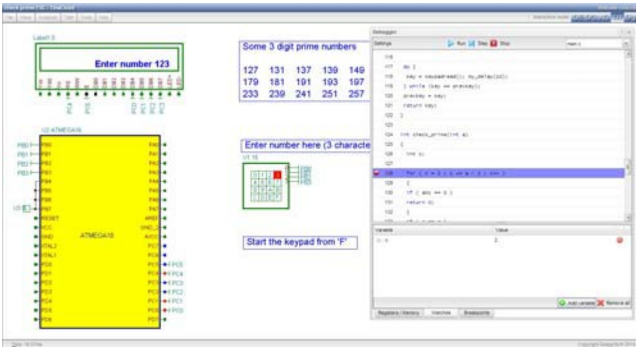
Scroll down the C code until the following function appears (line 128)

```
int check_prime(int a)
{
int c;
for (c=2;c <=a -1; c++)
{
if (   mod == 0 )
```

```
return 0;
}
if ( c == a )
return 1;
}
```

Add a breakpoint here, by clicking left to the for cycle statement (line 128), start the program (Run button). Enter the 123 number (for example) on the keypad. After that the program should stop at the breakpoint.

Now select the Watches tab, and click on Add variable, enter 'c'. The variable c should be appear.



After the stop at the breakpoint, you can continue the execution step by step by pressing the Step button for each step or run continuously by pressing the Run button again.

CHAPTER 5

USING SCHEMATIC SUBCIRCUITS AND SPICE MACROS

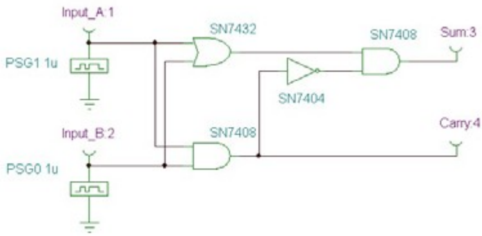
In TINACloud, you can simplify a schematic by turning portions of it into a subcircuit. In addition, you can create new TINACloud components from any Spice subcircuit. In this chapter, we show through text and examples how easy it is to do this in TINACloud.

5.1 Making a Macro from a schematic

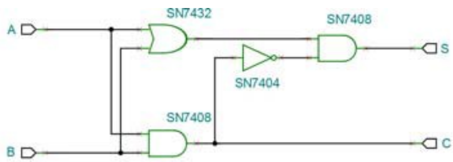
Using TINACloud's macro facility, you can simplify schematics and hide clutter by turning portions of the schematic into a subcircuit. TINACloud automatically represents these subcircuits as a rectangular block.

You can convert any schematic diagram into a subcircuit - called a Macro in TINACloud - simply by adding the terminals and saving the new circuit in the special (*.tsm) format.

Now let's see how to create a macro in TINACloud through an example. Load the Half Adder example (*Half_add.tsc*) from the Examples folder of TINACloud and convert it into a macro.



Delete the old terminals and replace them with subcircuit terminals, called Macro Pins in TINACloud. You can find and select the Macro Pins under the Special component toolbar.

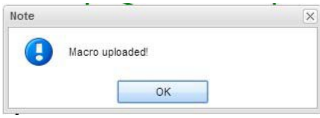


When you place Macro Pins, labels (such as Pin1, Pin2 etc.) are pre-filled in. Double click the Macro Pin and type in the new name in the label field. You can also drag the component, or rotate and mirror it with the buttons.

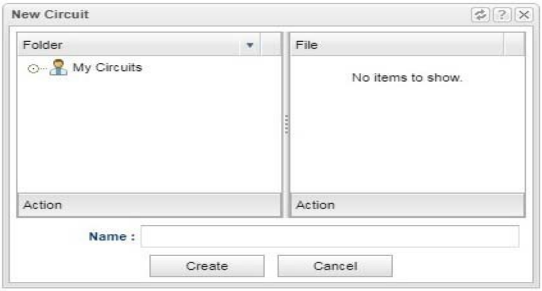
Next, create and save the new macro. Press the green button and select Upload Macro. The following dialog will appear:

Set the label to HA. This label will be displayed as the component label above the component.

When done, press the Upload button. The new Macro will be uploaded into your own Macro area, and the following message will be displayed

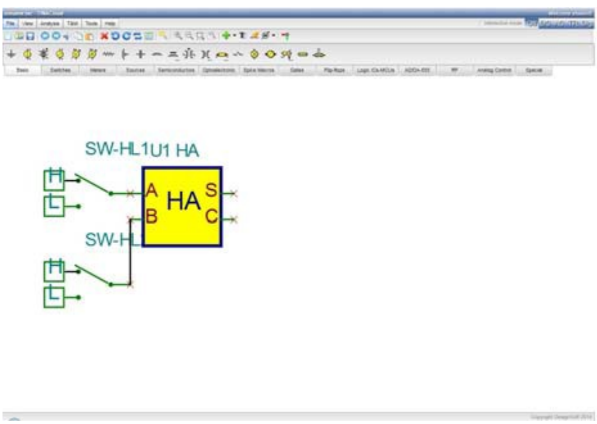


Now let's see how to insert a macro into a schematic and use it. Clear the circuit with **File|New.** The New Circuit dialog will appear.



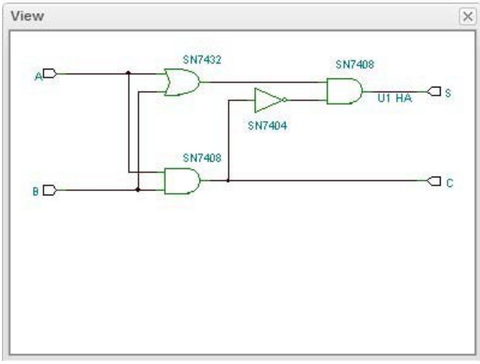
Enter the name of your circuit and press the Create button. Press the green  button and select Insert Macro.. The "User Macros" dialog will appear. Select HA and press OK. Our newly created macro will appear attached to the cursor, and you can place it in the usual way.

To test our macro lets add two digital switches from the Switches toolbar as shown below.

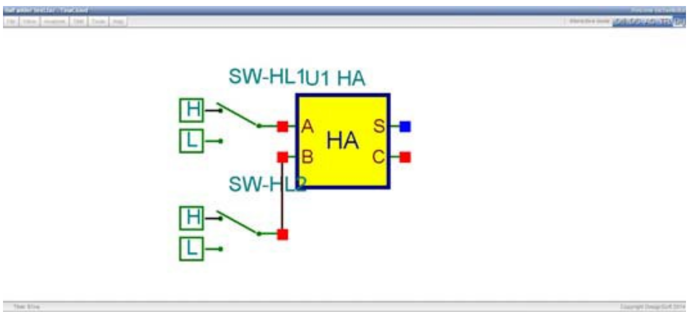


Now press the File | Save as menu and save your circuit under "Half adder test" or similar name.

If you click on the Macro and press the ... button in the SubCkt (Content) line you can see the schematic stored in the Macro as shown below.



Also you can test the circuit with the Macro interactively by pressing the Dig button in the top-right corner. The logic levels will be indicated at each node by a small block (Red for High, Blue for Low) and you can test the circuit response at different input levels by clicking or tapping the switches.

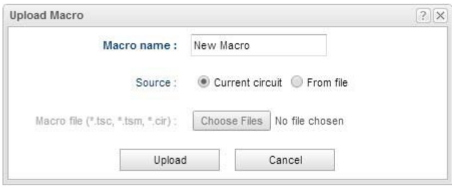


5.2 Making a Macro from a Spice subcircuit

Creating macros from downloaded files

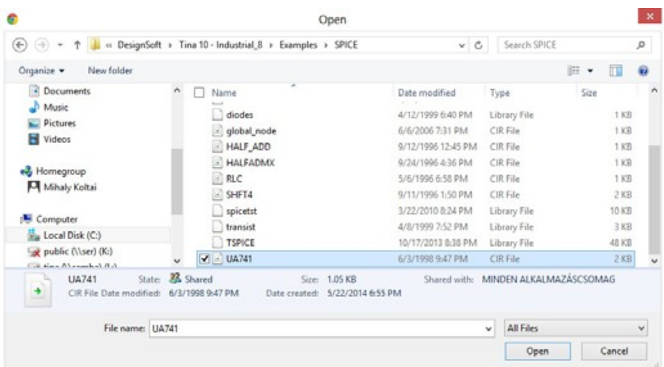
In TINACloud, you can create your own components from any Spice subcircuit that you have made or downloaded from the Internet. Note that there are already many Spice component models in the large and extensible manufacturers' model library provided with TINACloud. The extension of those libraries is described later.

Let's demonstrate this through the well-known uA741 operational amplifier using its Spice subcircuit model. To do this invoke the Schematic Editor and Press the green button and select Upload Macro. The following dialog will appear.

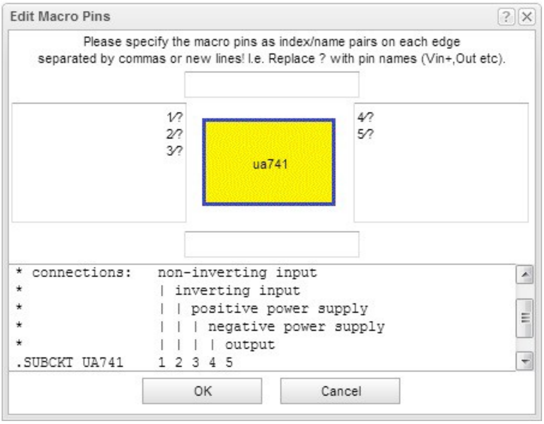


Set the label to ua741. This label will be displayed as the component label above the component. Change the Source settings from "Current Circuit" to "From file" and press the Upload button. An Open dialog box will appear.

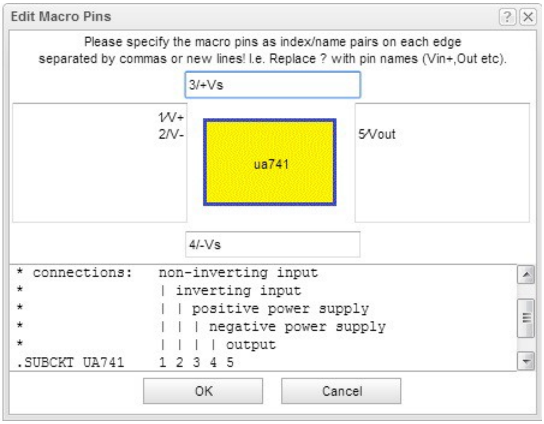
Now you should let's navigate to the file with the Spice model. We assume you already have an offline version of TINA on your computer. In this case find the EXAMPLES\SPICE folder of TINA . Select the UA741.CIR file and press the Open button.



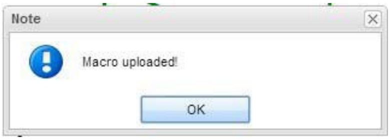
The “Edit Macro Pins” dialog will appear.



Using this dialog you can easily turn any Spice macro into a TINACloud macro since you can edit the pin names while seeing the Spice content of the macro at the same time. You can also choose the location of the pins on the box by simply placing the names at the required side of the box. Note that pin names are given in N/? format where N is the node number in the Spice macro while “?” is the name of the pin to be added. For example in our case “1/?” which according to the comments in the Spice macro is a “non-inverting input” should be turned into 1/V+ or 1/In+ or any similar names, while 5/? which is the positive power supply could be 5/+Vs or 5/+Vcc or similar and located on the top. You can assign any names to the pins. After editing the “Edit Macro Pins” dialog could look like this:

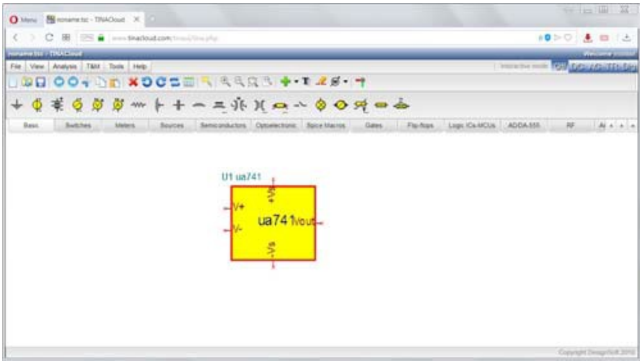


Now press the OK button. The Macro will be uploaded to your macro library and the following confirmation will appear:

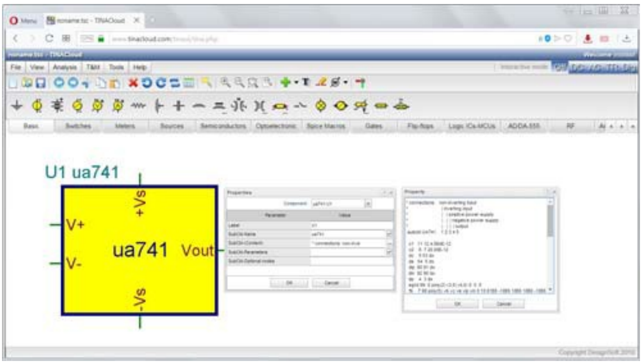


Finally let's see how to insert the macro into a circuit in the Schematic Editor. Fortunately it is the very same as described in the previous chapter in case of schematic macros.

Press the green  button and select Insert Macro… The "User Macros" dialog will appear. Select ua741 and press OK. Our newly created Spice macro will appear attached to the cursor, which you can place it in the usual way and then connect with other components.



In the main window of TINACloud, you can click the macro to see its properties. If in this Property dialog you click the … button in the SubCkt-(Content) line you can see and edit its content too.



CHAPTER 6

CREATING A PRINTED CIRCUIT BOARD (PCB)

Using TINACloud, you've captured the schematic of your circuit and refined the design. It's time to make a prototype of the circuit or to manufacture it. It's good to know that you can continue the process still using TINACloud, since PCB design is now an integral part of the program.

Let's learn how to design a PCB by working through a few examples.

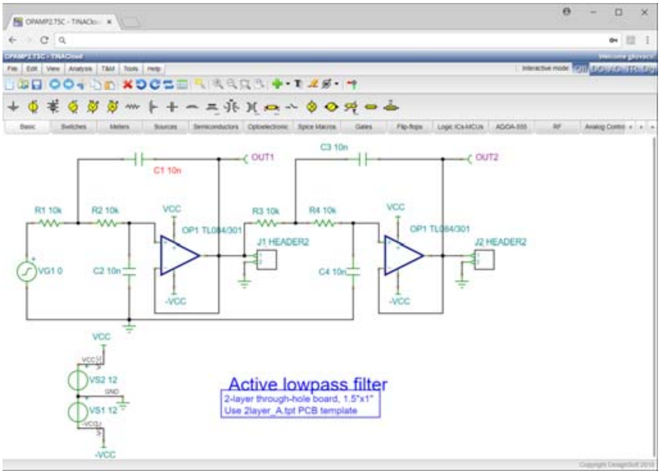
Note that the different phases of the presented design examples have been saved in the TINA'Clouds Examples/PCB directory using the following naming conventions:

* origin.tsc	original schematic file
*.tsc	backannotated schematic file (after pin/gate swapping)renumbering)
* placed.tpc	design parameters set, components placed pcb file
* routed.tpc	net properties set and routed pcb file
* finished.tpc	optionally pin/gate swapped and renumbered, routed, silkscreen adjusted, documentation layers finalized pcb file

If you save your versions of these demo examples, be careful not to overwrite the original files TINACloud installed.

6.1 Setting and checking footprint names

As a first example, open the opamp2.tsc project from TINACloud's Examples/PCB folder. The following schematic will appear:



To achieve an accurate PCB design, one that is easy to build, every part in your schematic must have a physical representation with exact physical size. This is realized through so-called footprints: drawings showing the outline and the pins of the parts.

The footprint naming convention in TINACloud uses as a starting point the IPC-SM-782A (Surface Mount Design and Land Pattern Standard) and the JEDEC standard JESD30C (Descriptive Designation System for Semiconductor-Device Packages, see http://www.jedec.org/download/search/jesd30c.pdf. However, the libraries do not conform to any given set of industry or manufacturer standards because standards have difficulty keeping up to date when technology changes faster than the standards. Generally, standards mirror a fixed set of data at a point in time, while new manufacturing capabilities lead to ever-smaller new footprints with more and more pins.

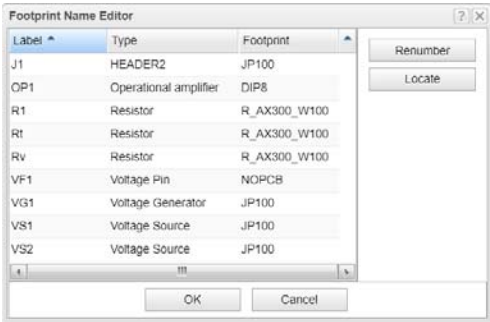
We have already assigned default footprint names to all parts that represent real components.

Note that some parts used for theoretical investigations, controlled sources, for example, do not represent real physical parts and cannot be placed on a PCB. If your design contains such components, you should replace them with real physical parts.

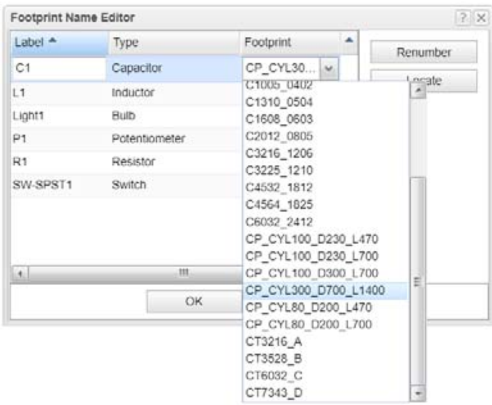
Of course there is no guarantee that the default physical packages of TINACloud's parts are the same as needed by your design.

There are two ways to check this.

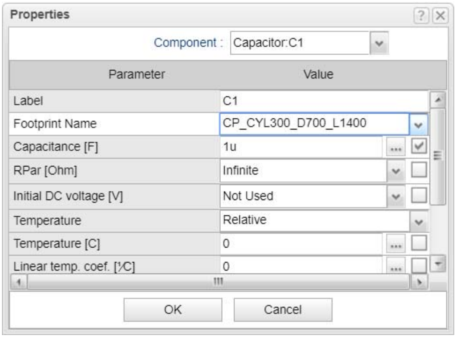
1) You can use TINACloud's "Footprint name editor", which you can invoke from TINACloud's Tools menu. In this dialog you can see all of TINACloud's components and their corresponding footprint names.



Select from the available footprint names by clicking on the footprint name fields. In the dialog box, components with no footprint name association are denoted by empty footprint name field.

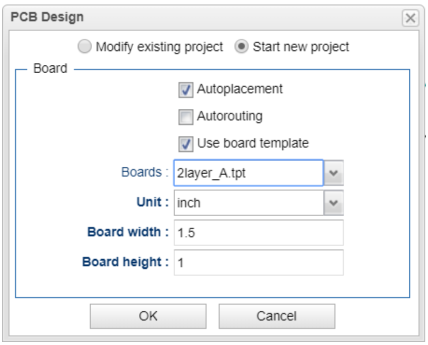


2) The second way to examine the assigned footprints is to double-click on each part and check the Footprint Name in the component property dialog that appears.



6.2 Invoking TINACloud PCB

To begin PCB design, press the  button on TINACloud's toolbar (the last one on the right) or select the "PCB Design" command on the Tools menu. Set the parameters as shown below.



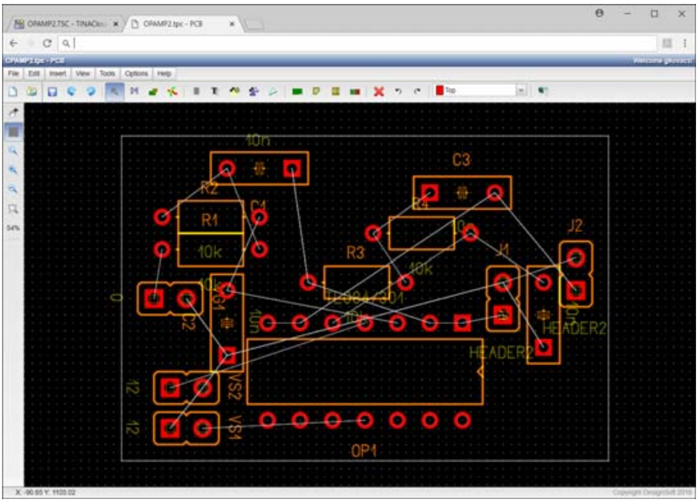
Select "Start New Project," "Autoplacement," and "Use board template." With the Browse button find and select the 2layer_A.tpt template file from TINACloud's Template folder. Using this file will ensure the proper settings for a double-sided PCB.
When starting with a TINACloud template, you are choosing the level of manufacturing complexity of your project. The following three levels of manufacturing technology are defined by the IPC-2221 generic standard:

Level A : General Design Complexity
Level B : Moderate Design Complexity
Level C : High Design Complexity

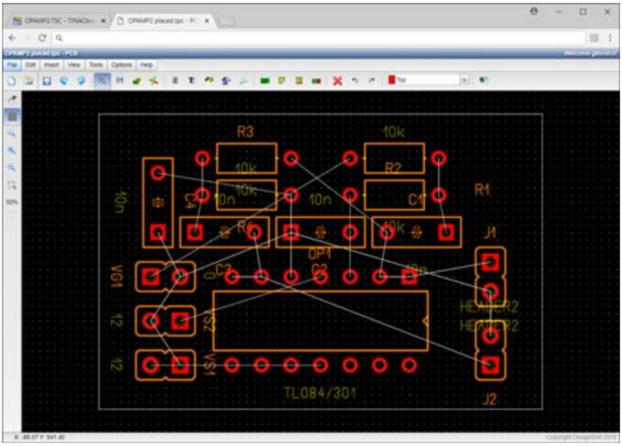
The template file specifies the number of layers, including their properties, system grid size, autorouter settings, spacing and track width. The following templates are included with PCB Designer:

	Level	Routing Layers	Plane Layers	Routing	Spacing	Comments
1layer_A.tpt	A	1	-	25	12 1/2	Allows one track between standard IP IC pin
2layer_A.tpt	A	2	-	25	12 ½	
2layer_B.tpt	B	2	-	8 1/3	8 1/3	Use for SMT or mixed-technoly board
2layer_B_mm.tpt	B	2	-	0.1	0.2	
4layer_C_mm.tpt	C	2	2	0.1	0.15	For moderate and high density

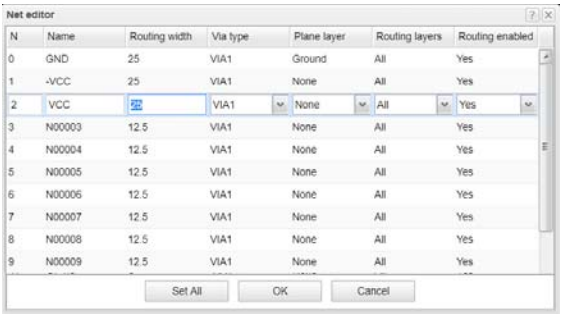
In choosing a PCB template, you should take into consideration technology, density, and package pitch.
To complete the set up, set the PCB size in inches or mm depending on the measurement unit settings in the View/Options dialog of TINACloud.
Now that everything is set properly press the OK button and the PCB layout design will appear with all the components automatically placed on the PCB board.



While all the parts and nets are placed, we need to adjust their positions for good placement and easier routing. Click and drag the parts to the position as shown on the figure below. (Find "opamp2 placed.tpc" to check your results.)



Click on Tools main menu item to reach the Net Editor and set net routing width. First, enter 12.5 into a "Track width" field then press "Set All". Then select power nets (Ground, VCC, -VCC) and set their track width to 25mil.

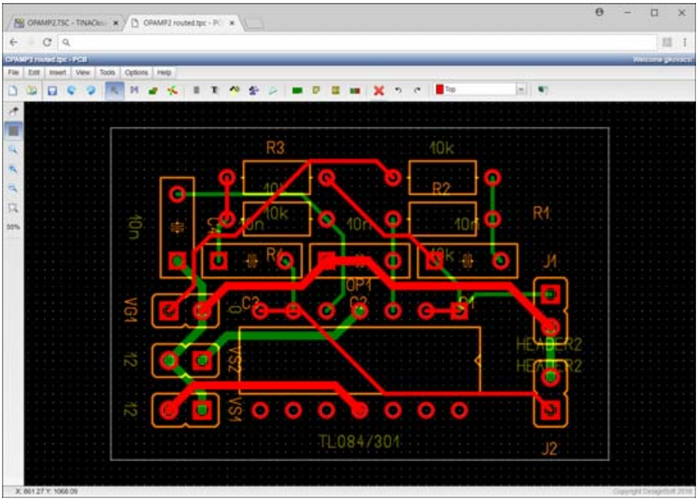


Net editor

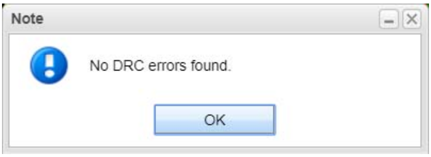
N	Name	Routing width	Via type	Plane layer	Routing layers	Routing enabled
0	GND	25	VIA1	Ground	All	Yes
1	-VCC	25	VIA1	None	All	Yes
2	VCC	25	VIA1	None	All	Yes
3	N00003	12.5	VIA1	None	All	Yes
4	N00004	12.5	VIA1	None	All	Yes
5	N00005	12.5	VIA1	None	All	Yes
6	N00006	12.5	VIA1	None	All	Yes
7	N00007	12.5	VIA1	None	All	Yes
8	N00008	12.5	VIA1	None	All	Yes
9	N00009	12.5	VIA1	None	All	Yes

Set All OK Cancel

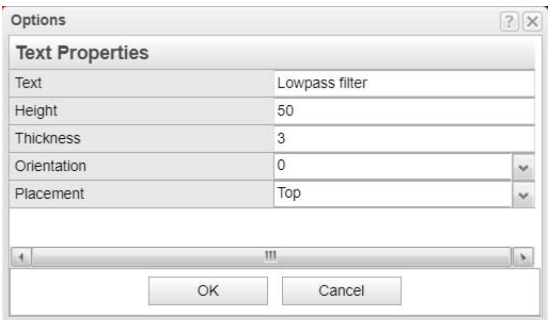
To automatically route the PCB, press select "Autoroute board" command from the Tools menu. The following screen will appear:



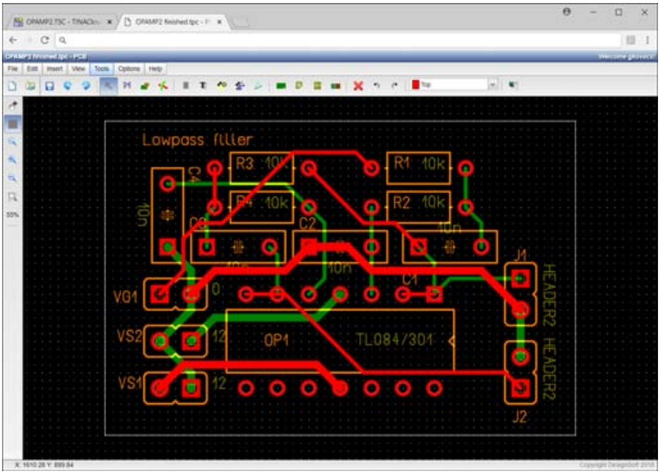
To see of everything is route correctly select DRC (Design Rule Check)/Run DRC from the Tools menu. The following message will appear:



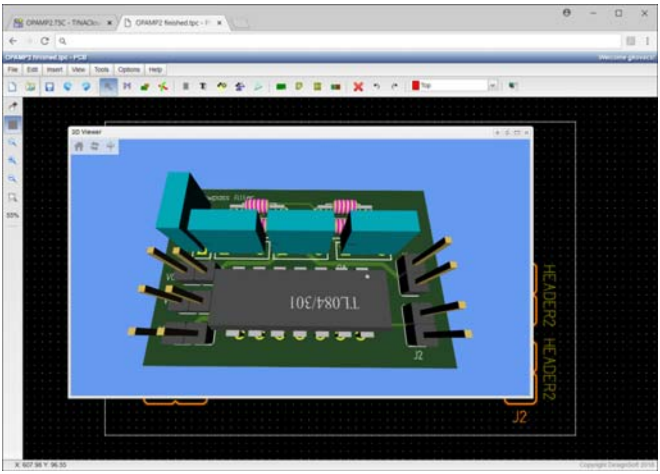
To finish our first simple design, let's add text to the silkscreen/ assembly layer. Click the Text button on the toolbar and make the settings in the dialog that appears:



Enter the text into the empty upper field and press the OK button. The text will be attached to the cursor. Move it to the location shown on the picture below.



Finally, you can check your design in full 3D. Press the 3D View button on the toolbar's right end or select "3D View" from the View menu. The following window will appear.



You can rotate the 3D model to any direction by holding the left mouse button and moving. Hold down the middle button and move the mouse to shift the viewpoint. You can also move the " camera" forward or backward to set a large view showing the complete PCB or to zoom in for more details. Use the mouse wheel for this operation.

At this point, you will probably want to create Gerber files for a manufacturer.

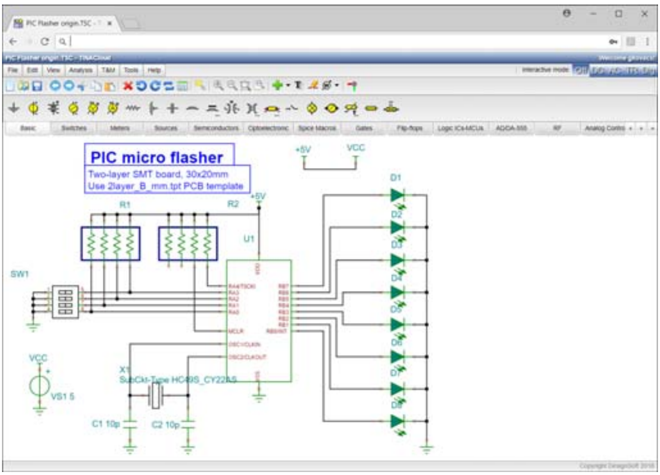
To obtain Gerber (RS - 274 X format) files to direct a photoplotter, choose ExportGerber file from the File menu. (Various Gerber options can be changed using the dialog for Gerber output settings under the Options menu.)

6.3 Advanced editing functions of TINACloud PCB's Layout editor

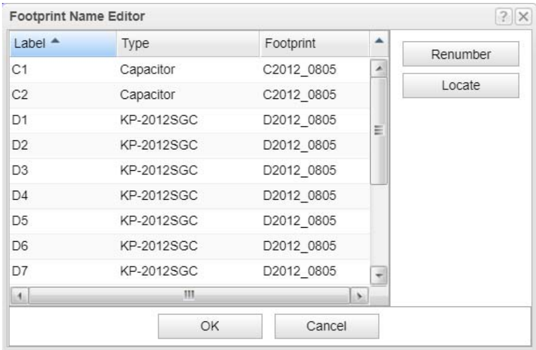
In practice, you may need to do s ome editing using TINACloud PCB's advanced editing features. This is described through examples in the next section.

CREATING A TWO-LAYER, DOUBLE-SIDED, SURFACE-MOUNT TECHNOLOGY BOARD

To get into TINACloud in more detail, open the second example, the file PIC flasher origin.TSC project from TINACloud's Examples/PCB folder.



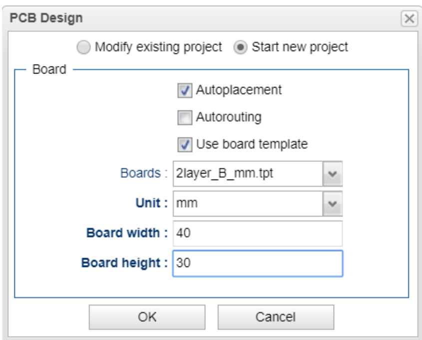
The schematic is PCB-ready, every part has a surface mount device (SMD) physical representation. Now, before we start the PCB wizard, click Tools/Footprint Name Editor… to check the footprints list.



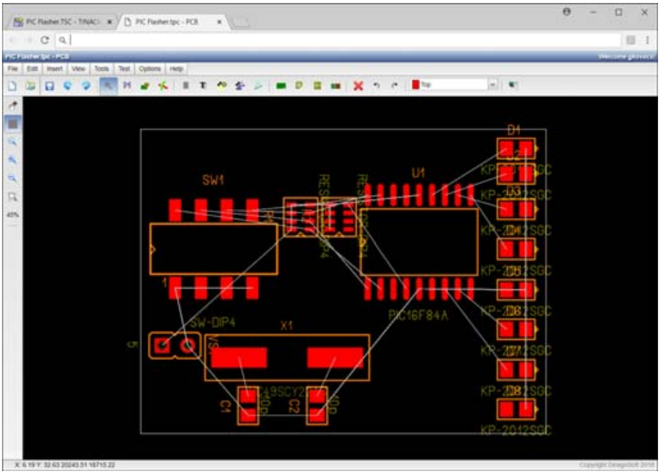
Since all the components appear to have a valid footprint name, we can start using the Tools/PCB Wizard. Click on "PCB Design..." in the Tools menu or on the button on TINACloud's toolbar.

Set the "Start new project," check "Autoplacement" and "Use board template." Select the template file "2layer_B_mm.tpt."

Review actual physical parts, if possible. Be sure to allow for the area of all the components, mounting holes, and keep-away zones and make your best estimate of the values for Board width and Board height. Moreover, it is important to provide enough space between the components to allow for the placement of vias and tracks during routing. Enter 40mm length and 30mm width.



Press the OK button, the PCB Editor window will appear.



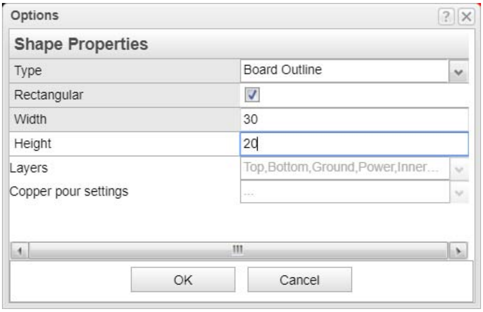
The components are placed in the close proximity to minimize the connection lengths, in a topology similar to that of the schematic, while still respecting the design rule settings. However convenient the result may be for autorouting, the designer usually has to make adjustments to the component layout to satisfy electrical, mechanical and other characteristics. Some of the considerations are -

- the ohmic effect of a long and/or thin power trace the length of a track from the signal source to the load in high-speed digital systems introduces reflections
- in analog situations, poor placement can lead to increased noise coupling
- allowance for automated parts placement clearance
- future serviceability of the PCB
- aesthetic values

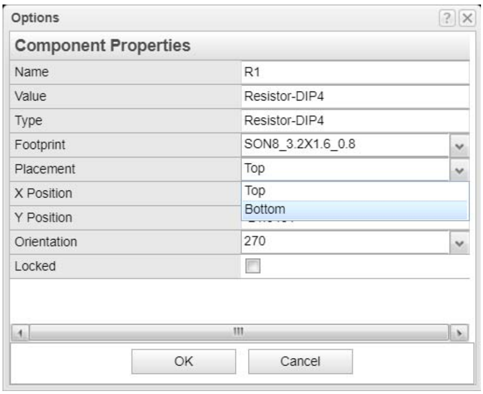
These considerations influence the components' position and could be critical–not only in complex designs–but even in the simplest ones. For these reasons, one must still adjust parts placement manually.

Our circuit, although it is small and not very dense, has a few special requirements, namely to put the crystal closer to the microcontroller, to position the power supply connector, and to adjust the LEDs along the board.

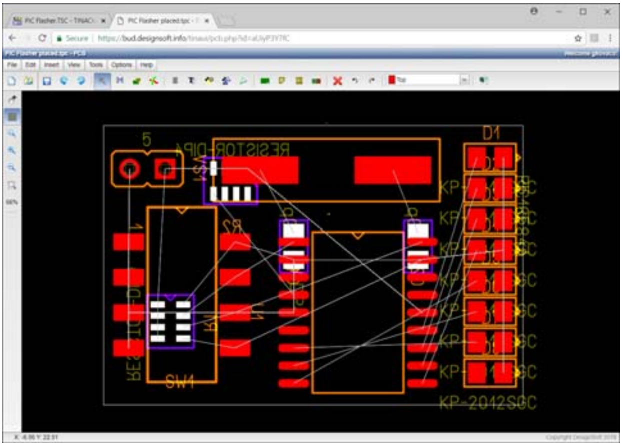
If you want to change the board size, click the "Draw/modify shapes" button  . Let's try making the board smaller. You can double click on the middle of the board and enter the following values into the fields:



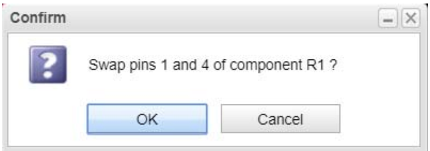
When you press OK, the board outline will shrink.
Now, before we begin routing, let's set the position of the components; place the power connector and the DIP switch on the left hand side and the LED bar on the right hand side. Place the capacitors and resistor networks on the bottom side by double clicking on the components and choosing Bottom Placement side on the dialog.



Compare your result to the file \Examples\PCB\PIC flasher placed.tpc.



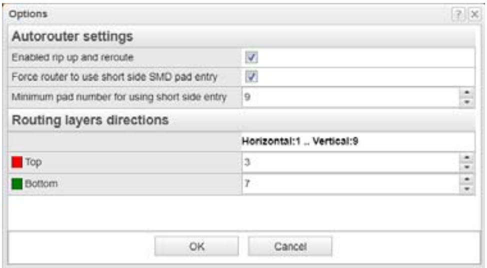
In order to decrease route length, swap R1 pins connected to SW1. Pin swapping is allowable if identically functioning pins are to be exchanged, such as pins of resistors, capacitors, etc. Click on the Pinswap button  on the toolbar to pick up the tool, click on pad 1 of R1 (the upper right), and–finally–pad 4 of R1. The following window should come up:



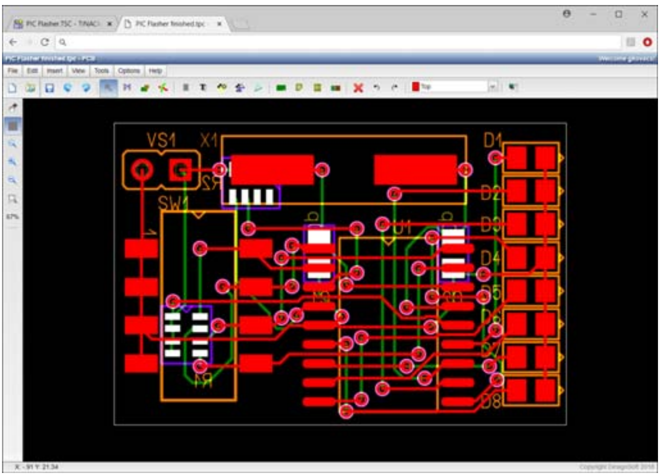
Press the Yes button, then do the same with R1 pad 2,3 and R2 pad 3,4.

Note that a pin swap changes the original connections, so we must update the original schematic later to maintain the correspondence between the schematic and the board taking into account the changes we made to the board while using PCB Designer.

This process, called back annotation, has to be completed whenever a pin/gate swap and/or component renaming have been performed. Signal planes (in our present example, the top and bottom layers) generally are given preferred trace directions up and down in one layer and left and right in the other layer. To set these trace directions, click on Options/Autorouter settings. The Autorouter settings are set to our preferred direction by choosing an integer number from 1 to 9. A value of 1 forces the router to heavily emphasize horizontal lines, while a value of 9 forces it to use vertical lines. 5 tells the router not to care about horizontal or vertical preference. Choosing the extreme values (1 and 9 for a pair) is usually too strict, so we choose to enter 3 for the top layer and 7 for the bottom.



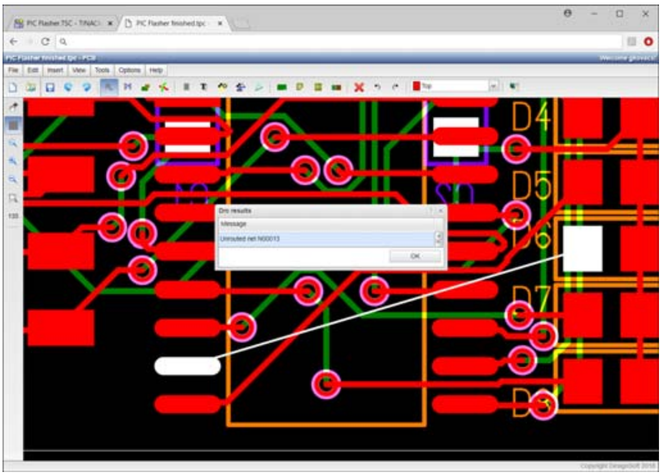
Also on this panel, check “Force router to use short side SMD pad entry” with pad number 9 constraining the router to connect SMD pads on their short side if the component has at least 9 pins. This option can be very used to preserve the gaps between SMD pads for track routing. Because of these choices, R1, R2 and SW1 are allowed to route freely, as half of the pins are connected together, while U1 pads connect on the short side.
After these moves, click on the main menu Tools/Autoroute board to route the PCB automatically. First the power and ground nets are connected, then signals are routed. If necessary, the program will rip-up tracks and reroute the unconnected nets. The following result will appear:



In some cases you might see that there is a net left unconnected. Fortunately, the DRC utility will examine your design and reveal any unconnected traces. Run DRC by clicking on the menu Tools/ DRC/Run DRC.

The DRC Results window will appear, highlighting the corresponding nets.

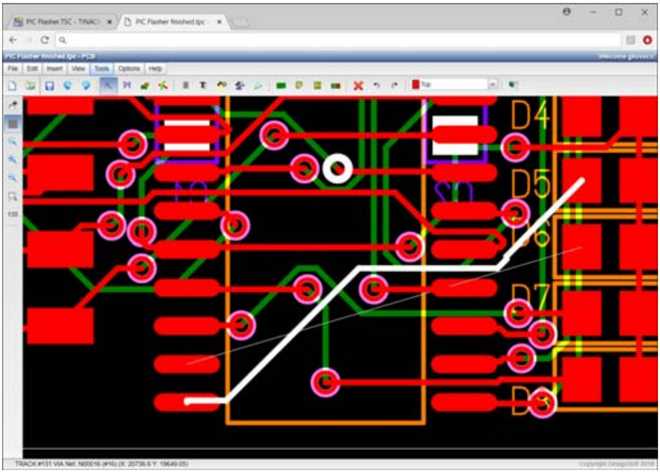
You should browse and correct any errors DRC reports. If you double click on the text 'Unrouted net N00013,' the program magnifies and highlights the selected net while centering the position of the selected objects.



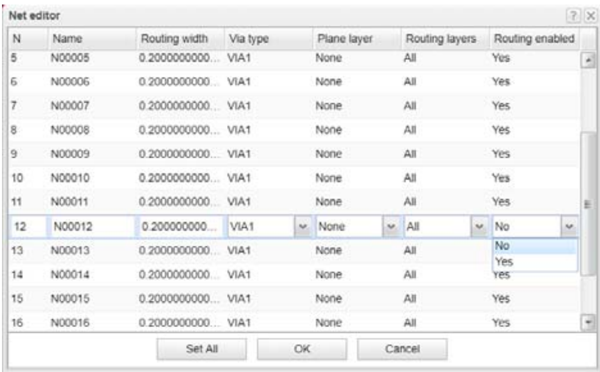
In order to fully route the board, we can use the manual route modes to make space for the unconnected tracks or delete crossing tracks and reroute them in a different order. Working in manual route, you can direct routing wherever necessary. You can move existing segments of tracks, remove segments, or create new segments.

Proceeding step by step, we will use autorouting to route the last trace.

First, press the , the select button on the toolbar, then hold down the Shift key and click the track that connects U1 pad 9 to D5 (see picture below), then press the Delete key to remove the track. This makes room for the unconnected net.

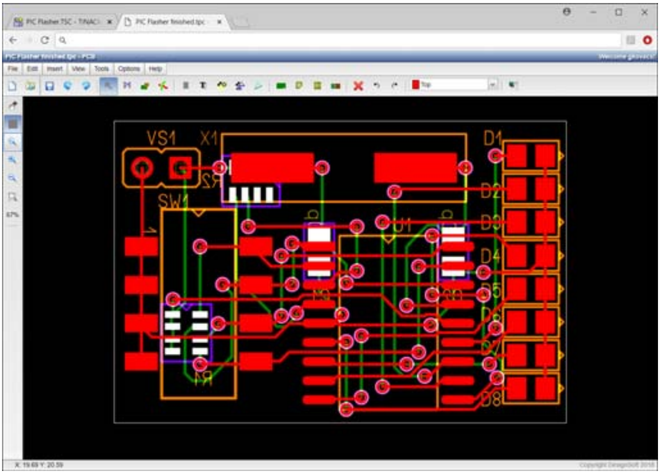


Now click on the menu Tools/Net editor to invoke the net editor. Disable N00012, the previously removed connection. This will prevent the autorouter from reconnecting the net. Leave N00013 routing enabled.



N	Name	Routing width	Via type	Plane layer	Routing layers	Routing enabled
5	N00005	0.2000000000...	VIA1	None	All	Yes
6	N00006	0.2000000000...	VIA1	None	All	Yes
7	N00007	0.2000000000...	VIA1	None	All	Yes
8	N00008	0.2000000000...	VIA1	None	All	Yes
9	N00009	0.2000000000...	VIA1	None	All	Yes
10	N00010	0.2000000000...	VIA1	None	All	Yes
11	N00011	0.2000000000...	VIA1	None	All	Yes
12	N00012	0.200000000...	VIA1	None	All	No
13	N00013	0.2000000000...	VIA1	None	All	No / Yes
14	N00014	0.2000000000...	VIA1	None	All	Yes
15	N00015	0.2000000000...	VIA1	None	All	Yes
16	N00016	0.2000000000...	VIA1	None	All	Yes

Click OK and the menu Tools/Continue autorouting and the autorouter will connect all the enabled nets. Next, enable N00012 in the Net editor, checking Routing Enabled, clicking OK and click again on Continue autorouting menu to autoroutethe last net.
Let's see if our manual intervention has worked. Call again Run DRC to verify that there are no errors.



Now, let's synchronize the PCB and schematic files. First, choose TinaCloud backannotation from the File menu to save the changes back to the schematic Editor. In the other browser window TINACloud will read and update the original schematic.
Take a close look at the resistor networks–the pin order reflects the result of the pinswap.

TINA CLOUD

The Complete Electronics Lab

WEB-BASED CIRCUIT
DESIGN & ANALYSIS

TINAClaud is the cloud based version of the popular TINA circuit simulation software now running in your browser. With this powerful, industrial strength, yet affordable circuit simulation tool you can analyze & design analog, digital, HDL, MCU, and mixed electronic circuits including also SMPS, RF, communication, and optoelectronic circuits and test microcontroller applications in a mixed circuit environment.

Electricai enginears will find TINAClaud an easy to use, high performance tool, while educators will welcome its unique features for the training environment and distance education.

TINAClaud will run on most OSs and computers, including PCs, Macs, thin clients, tablets, even on many smart phones, smart TVs and e-book readers. You can use TINAClaud in the classroom, at home, and, in fact, anywhere in the world that has internet access.

Super-fast multi-core engine running on the server. Every year, electronic circuits become faster and more complex, and therefore require more and more computational power to analyze their operation. To meet this requirement TINAClaud has the ability to utilize the increasingly popular scalable multi-thread CPUs. This is realized on our powerful server so whether you have a desktop, netbook, tablet oreven an e-book reader or mobile phone TINAClaud will run on it with the same high speed up to 20-limes faster execution time for compared to TINA 8 and main competitors.

LabXplorer

Multifunction Instrument for Education and Training with **Local** and **Remote** Measurement capabilities

REAL-TIME MEASUREMENTS

The optional LabXplorer turns your desktop, laptop, tablet or smart phone into a powerful, multifunction test and measurement instrument for a wide range of applications. Instruments, whatever you need, are at your fingertips. LabXplorer provides multimeter, oscilloscope, spectrum analyzer, logic analyzer, programmable analog and digital signal generator, impedance analyzer and also measures characteristics of passive electronic components and semiconductor devices.

DesignSoft
www.designsoftware.com