# INTRODUCTION TO PSPICE

**Rahul Behl**

MAY 18, 2014

# Preface

Introduction to PSpice consists of a set of tutorials which would guide the user to simulate basic analog and digital circuits using PSpice. The tutorials would provide step by step guide from installing PSpice to modeling and simulating circuits on PSpice. I hope you learn something valuable from these tutorials and get a better understanding of how different circuits work and what the characteristics of these circuits. For any other information feel free to contact me on raulbehl@gmail.com . I'll be happy to help.

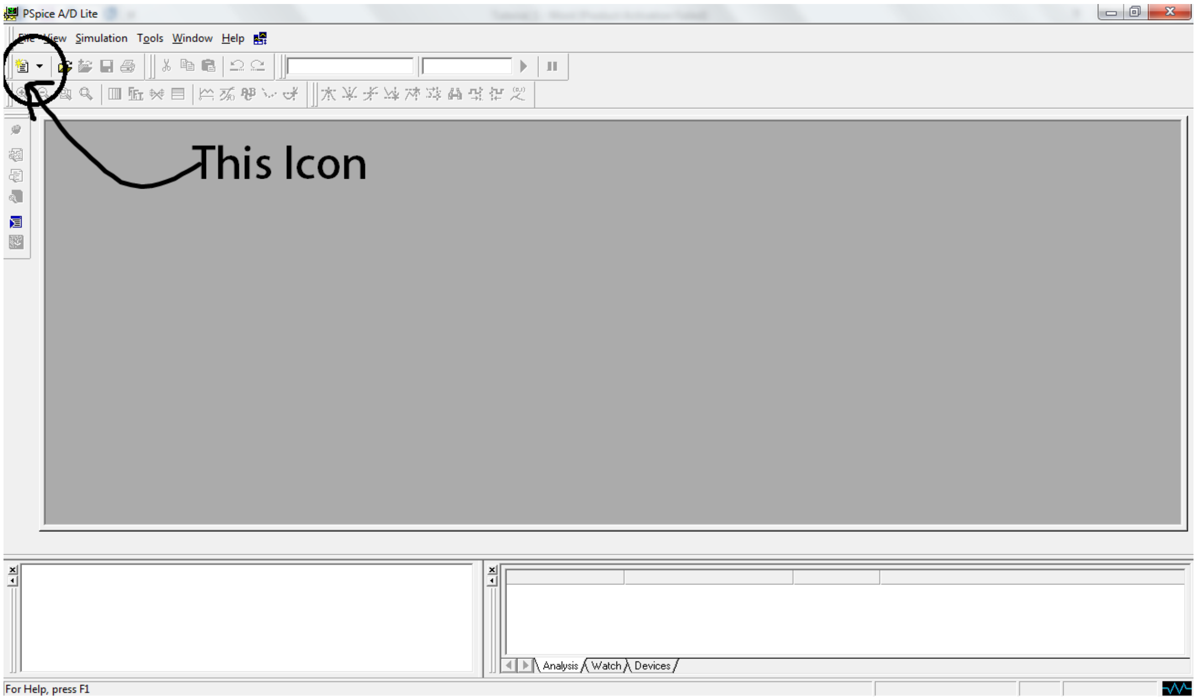
Happy Coding!

# Tutorial 2

## *Writing Your First Code*

Welcome to your second tutorial on PSpice. This tutorial would guide you towards your first PSpice code. Get ready for some hands on experience with PSpice.

Fire up PSpice and click on the new file icon just below the File button. The figure below shows the icon.



This Icon

But before we start writing the code we need to follow certain conventions which PSpice generally follows.

We are using PSpice for simulating digital and Analog circuits. Therefore we need to find a way in which we can express the circuit diagrams as a piece of code. PSpice generally uses the concept of “node” to convert any circuit diagram into a piece of code. To illustrate this consider the example below

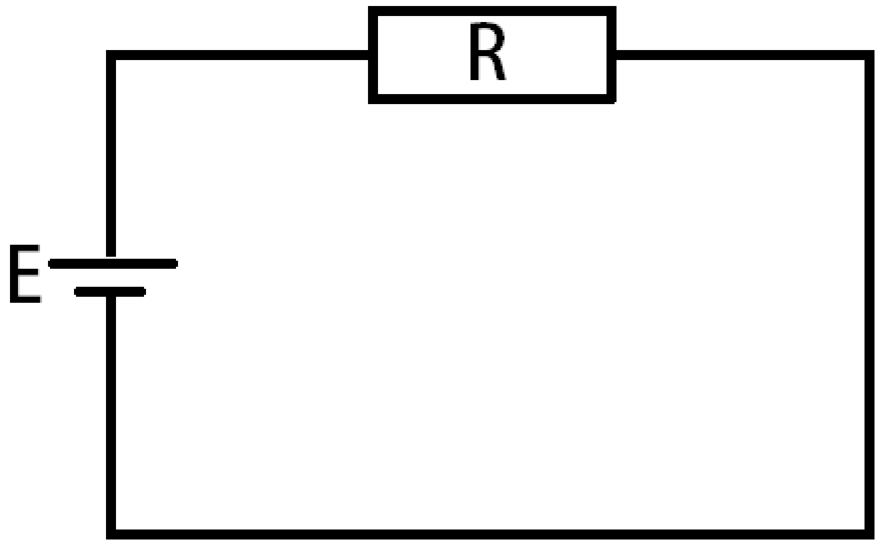


Fig. Circuit Diagram

The above figure shows the schematic of a battery source E connected to a resistor R in series. Now, we want to model this in PSpice. To do this we need to define "nodes". Therefore we'll model each of the nodes with a number as shown in the figure below

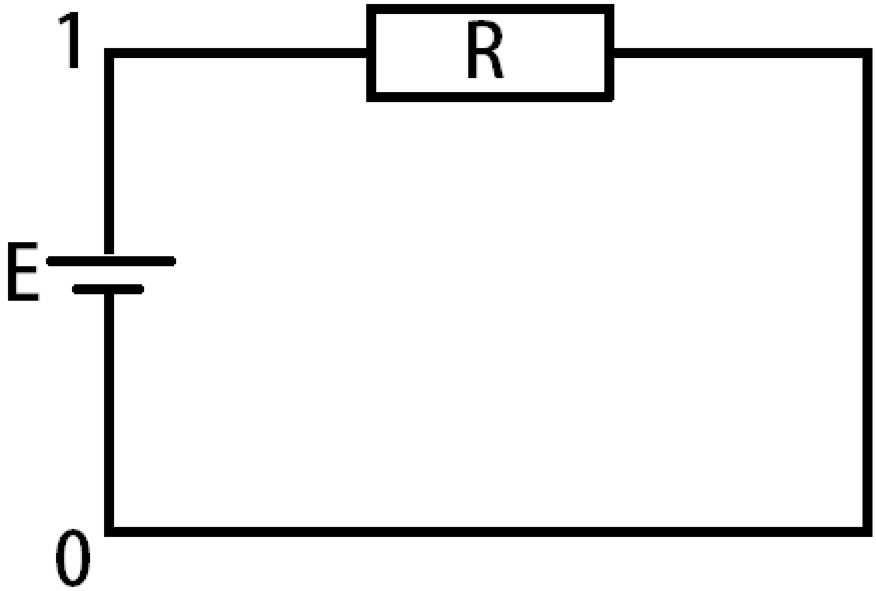


Fig – Circuit Diagram with nodes

We require to define only two nodes to completely convert this circuit into a piece of code.

Note – By default node **0** refers to **Ground.** Therefore, in order to define ground point, we'll just replace that node with 0.

## *Getting Started With Syntax*

After completely defining the nodes for the circuit diagram we can easily begin writing it into a code. As any other coding language, PSpice also has certain sets of syntax.

### Syntax

i) The first line of every PSpice code is not compiled. In general, the first line is used to give the name of the circuit being simulated. For our running example let it be

```
>> First PSpice Code
```

ii) Now we need to model the Voltage source. The syntax for modelling a voltage source is

Name (first letter is V) node1(+ve terminal)  node2 (-ve terminal)  DC/AC  value

For this example we can model the voltage source as

```
>> Vin    1     0     DC    10V
```

You can similarly model a current source. You just need to change the initial V to I as shown

```
>> Iin    1     0     DC    2A
```

Note : For modeling a current source the nodes decide the direction of current. As in the above example the current is flowing from node 1 to node 0. This is roughly opposite in the way a voltage source is defined.

iii) After modeling the sources we need to model the impedance. All the basic passive components can be modelled in the similar way.

```
>> Rin      1      0      1k
```

For modeling a capacitor, just replace R with C as shown,

```
>> Cx       1      0      1uF
```

(Here ‘u’ stands for micro, similarly if it was ‘n’ then it would have meant nano farads, m -> milli, k -> kilo and so on).

Inductor,

```
>> Lout     1      0      5mH
```

iv) With all this done we need to add the library file from where PSpice would get all the complete models of the components. We usually will use NOM.LIB as our library file.

To add the library file to your code, do the following

```
>> .Lib NOM.LIB
```

This line of code would attach the library file.

v) With this done, we need to save the file. For saving the file you need to specify the extension. The extension name of the file is ‘.cir’. So, our file would be saved like  
first_code.cir

The entire piece of code for the above example is:

```
>> First PSpice Code
>> .LIB NOM.LIB
>> Vin      1      0      DC     10
>> Rin      1      0      1K
```

Save the file as first_code.cir . That’s it. You completely converted that circuit into a PSpice compatible code.

With this you come to an end of your second tutorial. Try modeling more complex circuits in PSpice and get yourself familiar with nodes and modeling them into PSpice. In the next tutorial we'll begin with the simulation of these circuits and would learn more commands useful for plotting and obtaining values of currents and voltages at different nodes.