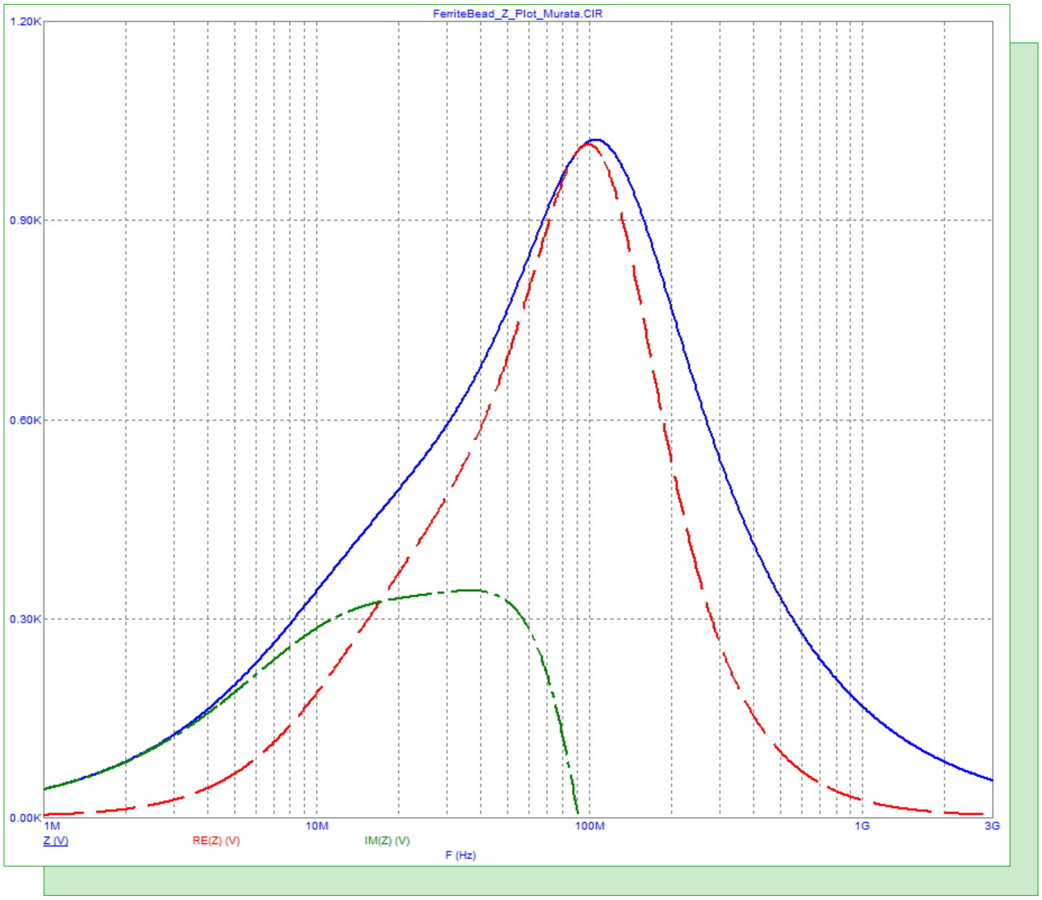
spectrum news

Applications for Micro-Cap™ Users

Spring 2015

News



Ferrite Impedance Plot

Featuring:
- Subcircuit Variables and Passed Parameters
- A Simple Way to Add Parts to the Component Library
- Plotting Ferrite Core Impedance and S Parameters

News In Preview

This newsletter's Q and A section describes how to name nodes. Named nodes have the advantage over system-assigned node number names in that they do not change. The Easily Overlooked Feature section describes the Localize command.

The first article describes how to plot variables in subcircuits and macros.

The second article describes a method for quickly adding parts modeled by .Model statements to the Component library.

The third article describes how to model ferrite cores and plot their impedance and S parameter curves.

Contents

Book Recommendations

General SPICE

• *Computer-Aided Circuit Analysis Using SPICE*, Walter Banzhaf, Prentice Hall 1989. ISBN# 0-13-162579-9

• *Macromodeling with SPICE*, Connelly and Choi, Prentice Hall 1992. ISBN# 0-13-544941-3

• *Inside SPICE-Overcoming the Obstacles of Circuit Simulation*, Ron Kielkowski, McGraw-Hill, 1993. ISBN# 0-07-911525-X

• *The SPICE Book,* Andrei Vladimirescu, John Wiley & Sons, Inc., 1994. ISBN# 0-471-60926-9

MOSFET Modeling

• *MOSFET Models for SPICE Simulation, William Liu, Including BSIM3v3 and BSIM4*, Wiley-Interscience, ISBN# 0-471-39697-4

Signal Integrity

• *Signal Integrity and Radiated Emission of High-Speed Digital Signals*, Spartaco Caniggia, Francescaromana Maradei, A John Wiley and Sons, Ltd, First Edition, 2008 ISBN# 978-0-470-51166-4

Micro-Cap - Czech

• *Resime Elektronicke Obvody,* Dalibor Biolek, BEN, First Edition, 2004. ISBN# 80-7300-125-X

Micro-Cap - German

• *Simulation elektronischer Schaltungen mit MICRO-CAP,* Joachim Vester, Verlag Vieweg+Teubner, First Edition, 2010. ISBN# 978-3-8348-0402-0

Micro-Cap - Finnish

• *Elektroniikkasimulaattori,* Timo Haiko, Werner Soderstrom Osakeyhtio, 2002. ISBN# 951-0-25672-2

Design

• *High Performance Audio Power Amplifiers,* Ben Duncan, Newnes, 1996. ISBN# 0-7506-2629-1

• *Microelectronic Circuits,* Adel Sedra, Kenneth Smith, Fourth Edition, Oxford, 1998

High Power Electronics

• *Power Electronics,* Mohan, Undeland, Robbins, Second Edition, 1995. ISBN# 0-471-58408-8

• *Modern Power Electronics,* Trzynadlowski, 1998. ISBN# 0-471-15303-6

Switched-Mode Power Supply Simulation

• *SMPS Simulation with SPICE 3,* Steven M. Sandler, McGraw Hill, 1997. ISBN# 0-07-913227-8

• *Switch-Mode Power Supplies Spice Simulations and Practical Designs*, Christophe Basso, McGraw-Hill 2008. This book describes many of the SMPS models supplied with Micro-Cap.

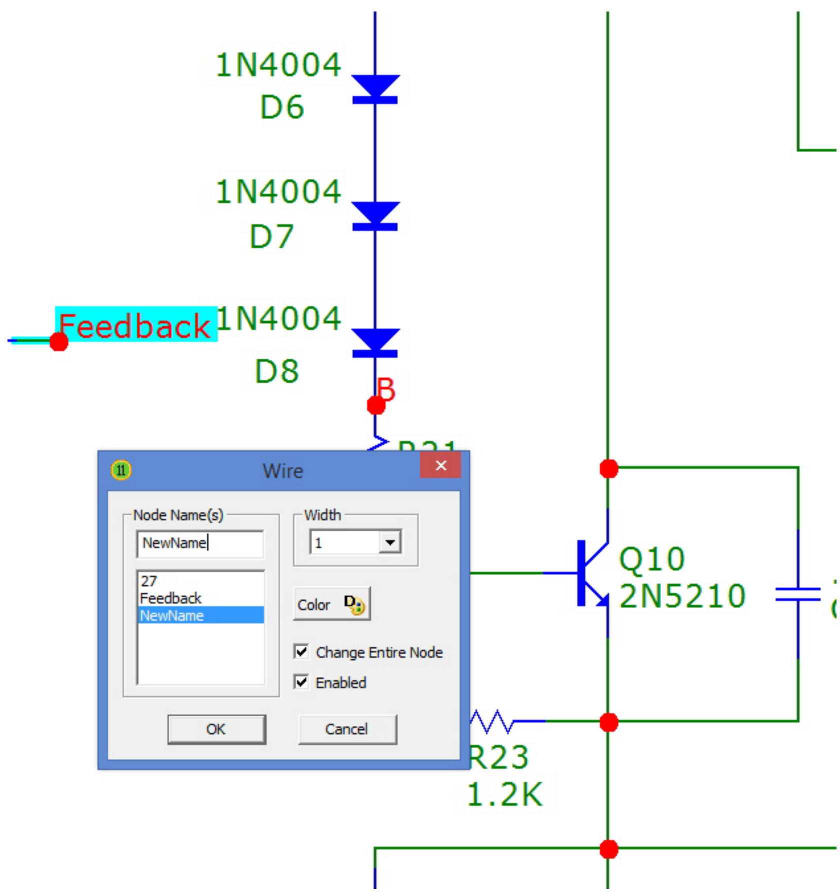
Micro-Cap Questions and Answers

Question: I was just wondering if there was a way of having node numbers stay fixed so when I added a component they would not change.

Answer: Not directly. The node numbers must be able to adapt to a changing circuit. What you can do though is to assign a node label like IN or OUT or even simpler A, or B, or C.

To assign a node label simply add text to any part of the node. Click on the Text icon, then click on the node and type in a node name. Once the OUT node label is assigned to a node, you can plot V(OUT) instead of say, V(10), avoiding the concern that the numeric name will change.

You can also add, edit, or change the node name by double-clicking on it while in Select mode. When you do this a dialog box comes up showing the names the node already has. Every node has a numeric name assigned by the program. It may have as many names as you like. To add a new one, type the name into the Node Name(s) field. To edit an existing node name, select it from the list and then type in the changes you want. Here is what the dialog box looks like:



Easily Overlooked Features

This section is designed to highlight one or two features per issue that may be overlooked among all the capabilities of Micro-Cap.

Localize command:

Ever had to send a circuit to a colleague and had trouble sending the necessary files? Well there's an easy way to do it using the Localize command.

First some background: When you select an analysis, Micro-Cap accesses electrical modeling information for any parts in the circuit that require models. How does it do this? It searches for the modeling information (model parameter lists, model statements, subcircuits, or macro statements) in the following order:

Local Search within the circuit:

- In the grid text or text area
- In the file named in the File attribute (if the device has one).
- In one or more files named in a .LIB filename statement.
- In any files named in the NOM.LIB of the folder specified for the last used LIBRARY path.
- In any files named in the NOM.LIB statement found in any path from a .PATH LIBRARY command within the circuit.

Global Search using library paths:

• In any files named in the NOM.LIB statement found in any path from File menu / Paths / Model Library and Include Files. If more than one path is specified, it searches them in left to right order. For example, consider this path:

C:\Micro-Cap\LIBRARY ; D:\OTHER ; E:\ELSEWHERE

In this example, the search starts in C:\Micro-Cap\LIBRARY, then proceeds to D:\OTHER. Finally E:\ELSEWHERE is searched.

Because Micro-Cap first searches locally in the circuit, you can use the Localize command to place the needed model data within the circuit making it easier to send the circuit in a single file to your colleague.

Subcircuit Variables and Passed Parameters

Have you ever needed to plot a macro or subcircuit variable like the voltage several levels deep in a subcircuit or to see the value of a calculated, passed parameter? It's easy and here is how it can be done; access the variable as X1.X2.*variable_name*, where *variable_name* is the name you'd use at the normal circuit level.

For instance, suppose you have this circuit. The main circuit calls the XYZ macro and it calls the DEF macro.

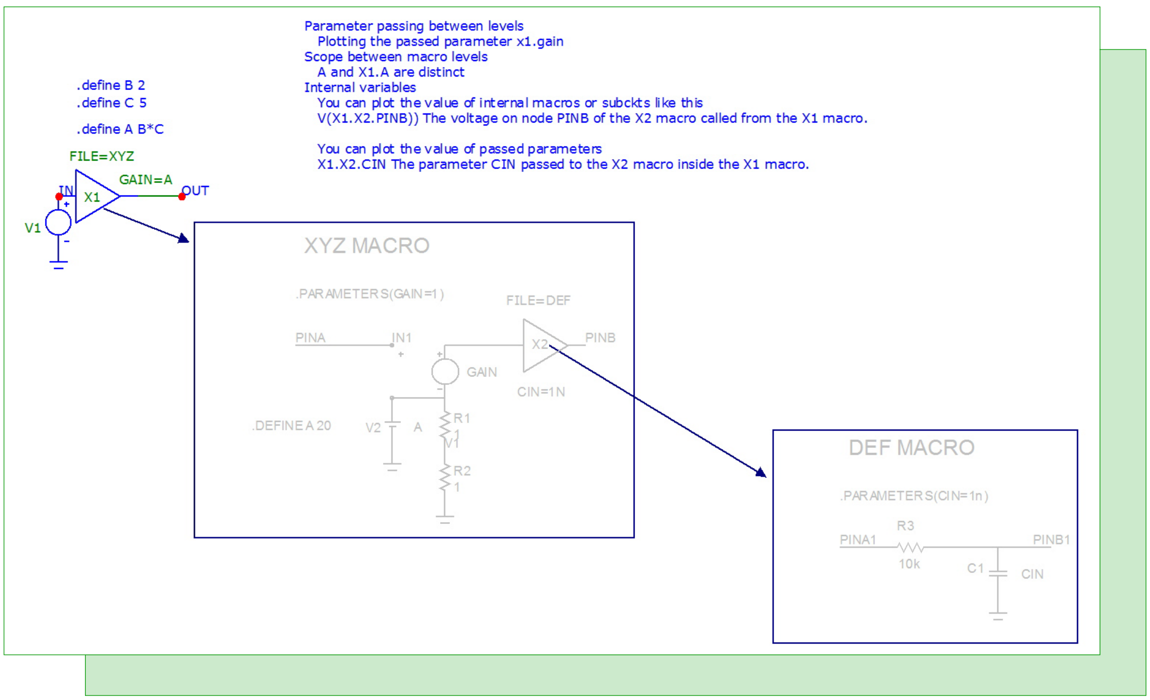


Fig. 1 - Sample Circuit

To plot the voltage at the node PINB in the X2 subcircuit called by the X1 circuit, use this:

V(X1.X2.PINB)

Each reference to a lower level subcircuit or macro use the subcircuit names (X1 X2) and a period to separate them.

X1 is the name of the highest level subcircuit or macro.
X2 is the name of the next lowest level subcircuit or macro.
PINB is the name of the node in the X2 circuit whose voltage we want to plot.

The same method can be applied to other variables, like the resistance of the resistor R1 in X1 would be V(X1.R1) .

X1 is the name of the highest level subcircuit or macro.
R1 is the name of a resistor in X1
V(X1.R1) is the voltage across the resistor named R1.

Here, for example, is the capacitance of the capacitor C1 in X2, called from X1.

X1 is the name of the highest level subcircuit or macro.
X2 is the name of the subcircuit or macro called from X1.
C1 is the name of a capacitor C1 in X2
V(X1.X2.C1) is the voltage across the capacitor C1.

What about passed parameters? Here is how is you'd plot the CIN parameter passed to the X2 macro.

X1 is the name of the highest level subcircuit or macro.
X2 is the name of the subcircuit or macro called from X1.
V(X1.X2.CIN) is the capacitance parameter passed to the X2 macro.

How about locally defined parameters? Here is how is you'd plot the value of the A variable in the X1 macro.

X1 is the name of the highest level subcircuit or macro.
V(X1.A) is the value of the local A parameter in the X1 macro.

Note that there is an A variable defined in the main circuit that is distinct from the A variable defined in the X1 subcircuit. The variables A and X1.A are separate and distinct.

This plot shows the value of several of these variables

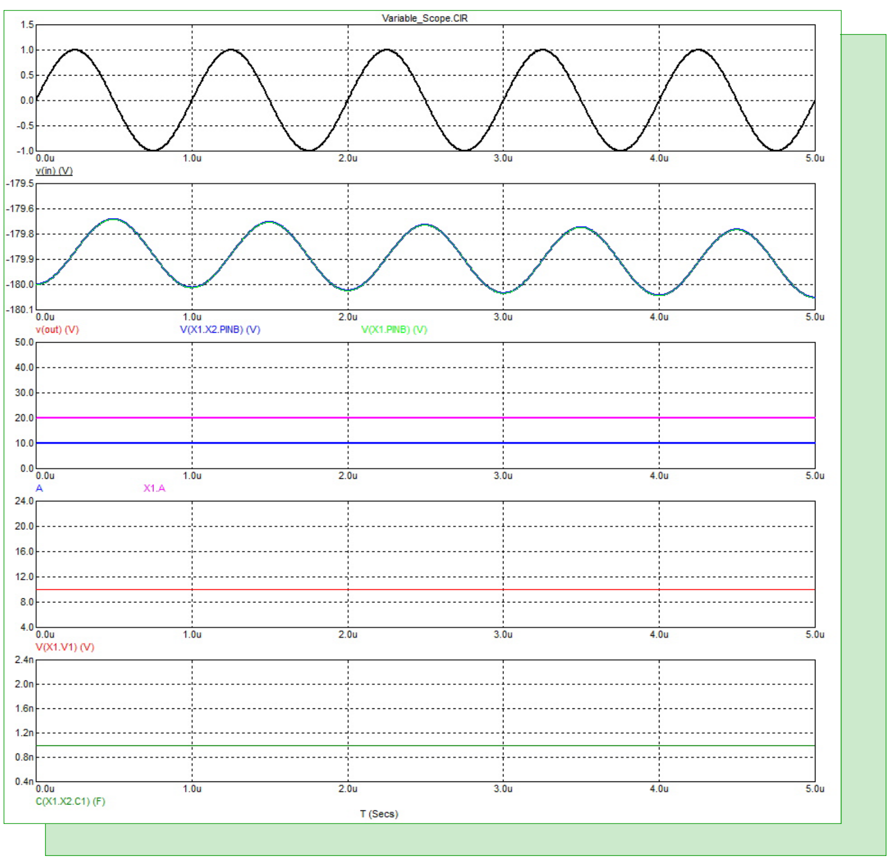


Fig. 2 - Transient analysis showing variable values

A Simple Way to Add Parts to the Component Library

Here is a simple method to add parts to the Component Library.

If your part is a primitive (e.g. diode, NPN, NMOS, etc.) that is modeled with a .model statement, then there is an easy way to add the part to the library.

Assume that you have an NPN called 2NK1730 and its .model statement. Here is what you do:

1) Create a file from the File New command.
2) Copy and paste the .model statement into the file.
3) Save the file in the LIBRARY folder. Call it MYLIB.LIB.
4) Open the NOM.LIB file from your LIBRARY folder and add this line anywhere in that file.

```
.LIB "MYLIB.LIB"
```

This adds the contents of the MYLIB.LIB file, including your 2NK1730 part to the list of candidates when you place a new part.

As you create new model statements for other parts, add them to the file. They can be of any primitive type so long as they are modeled with a .model statement.

The 2NK1730 part name will not appear on the Component Menu. That is because when you add the part using the Component editor's Add Part command, the first step specifies where in the Component menu hierarchy you want the part name to appear. That step is missing in this simpler procedure. The name, however, will appear as an item in the part name list in the Attribute dialog box when you place an NPN part. Here are the steps again, this time in more detail.

1) Create a file from the File New command.

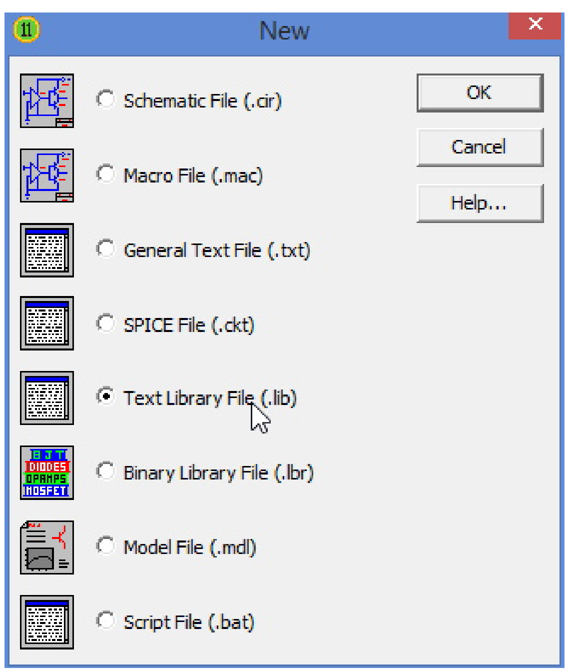


Fig. 3 - The File New dialog box

Click on the Text Library File (.LIB) item. Click on the OK button. MC11 will create a file called LIBRARY1.LIB and place it on the screen. Go to the circuit or other place where you have your model statement. Select the model statement and copy it. Paste the model statement text into the library file. The screen should now look something like this:

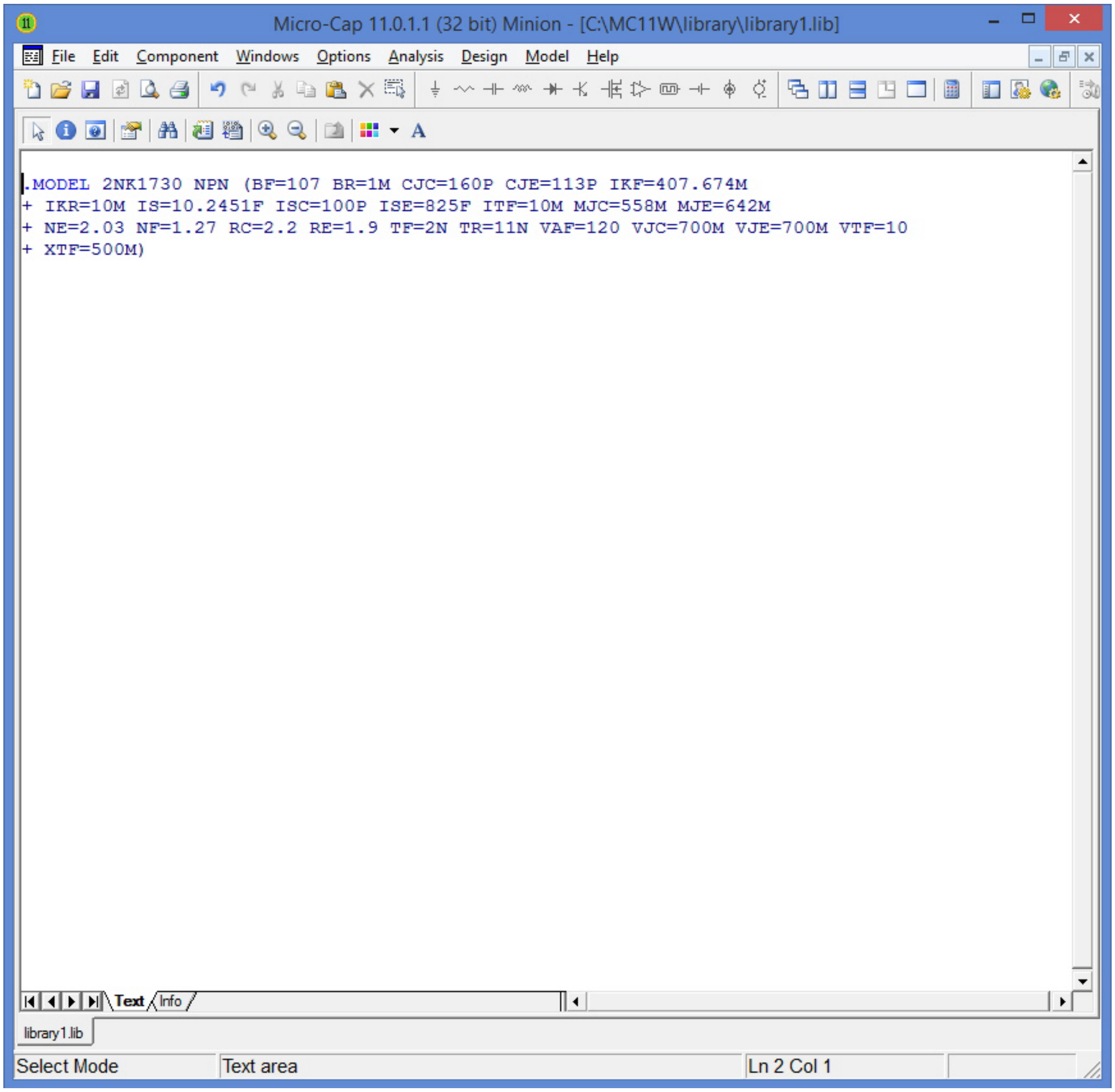


Fig. 4 - The text library file

Here the .model statement has been pasted into the library file.

Now we're ready for the next step.

3) Save the file in the LIBRARY folder. Call it MYLIB.LIB.

Use the File menu / Save As command and enter

Windows XP: MYLIB.LIB

Windows 7 or later: MYLIB with the File Type selected as SPICE library (*.LIB)

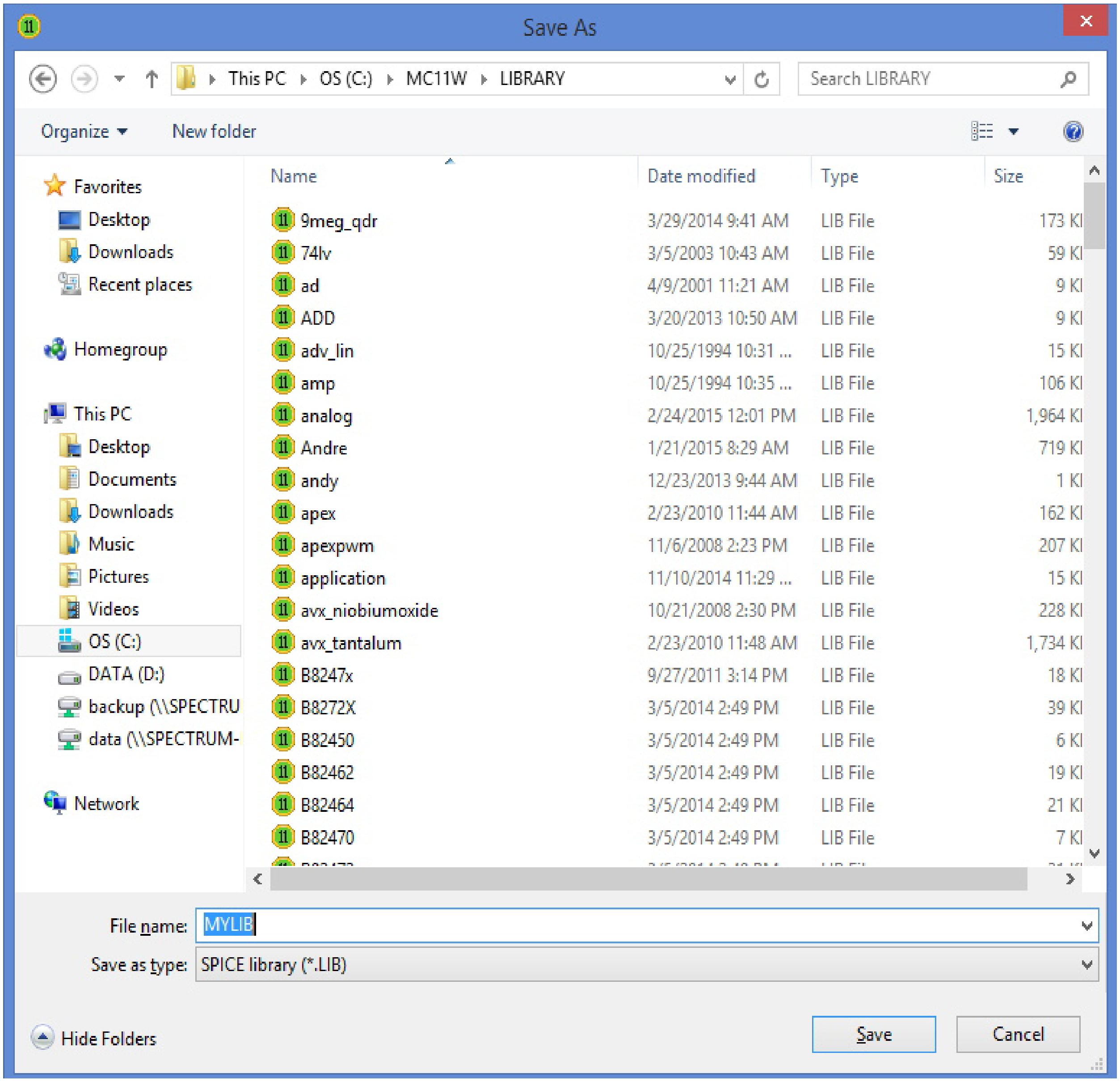


Fig. 5 - Saving the file as MYLIB

Here we are saving the file as MYLIB and type SPICE library and we are saving it in the LIBRARY folder as that is where it must be.

Now we're ready for step 4.

4) Open the NOM.LIB file from your LIBRARY folder and add this line anywhere in that file.

.LIB "MYLIB.LIB"

Use the File menu / Open command to open the NOM.LIB file located in the LIBRARY folder.

Pick any line location in the file and add this line

    .LIB "MYLIB.LIB"

Your screen should now look like the following.

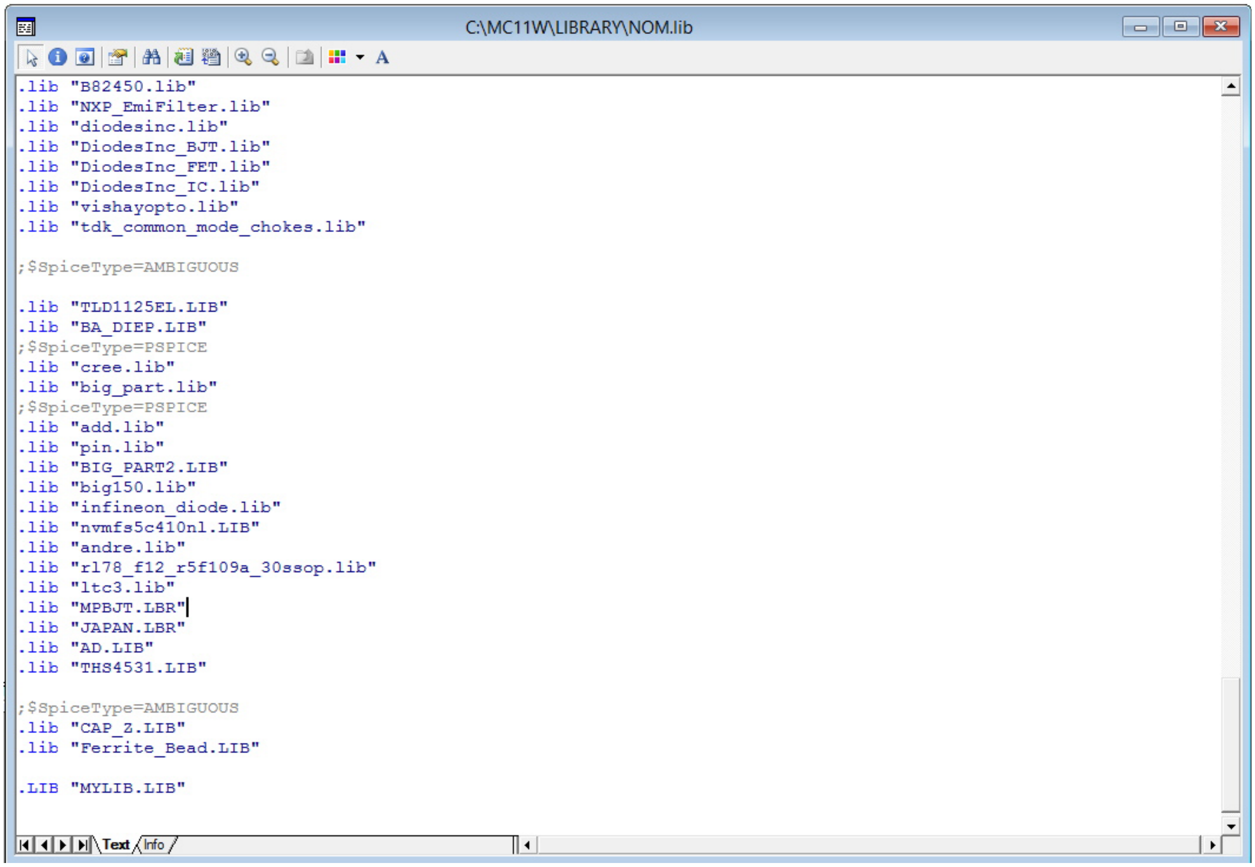


Fig. 6 - Adding the .LIB "MYLIB.LIB" line to the NOM.LIB file

After adding the .LIB "MYLIB.LIB" line, use the File menu Save command to save the modified NOM.LIB file.

Now the MYLIB.LIB contents, including the 2NK1730 NPN, will be added to the library and thus available for placement and use in a schematic.

Plotting Ferrite Core Impedance and S Parameter Curves

An earlier newsletter showed how to model ferrite beads using a double parallel RLC plus series resistor model. The model parameters for the seven passive components were easily found by Micro-Cap's Model program optimizer.

So the question is do the S parameter curves match as well? The short answer is yes. This article will show you how to plot those S parameter curves and how to compare them with those provided by the manufacturer.

To start with let's consider the BLM21AG102SH1D part from Murata. Here is its Z vs. F plot.

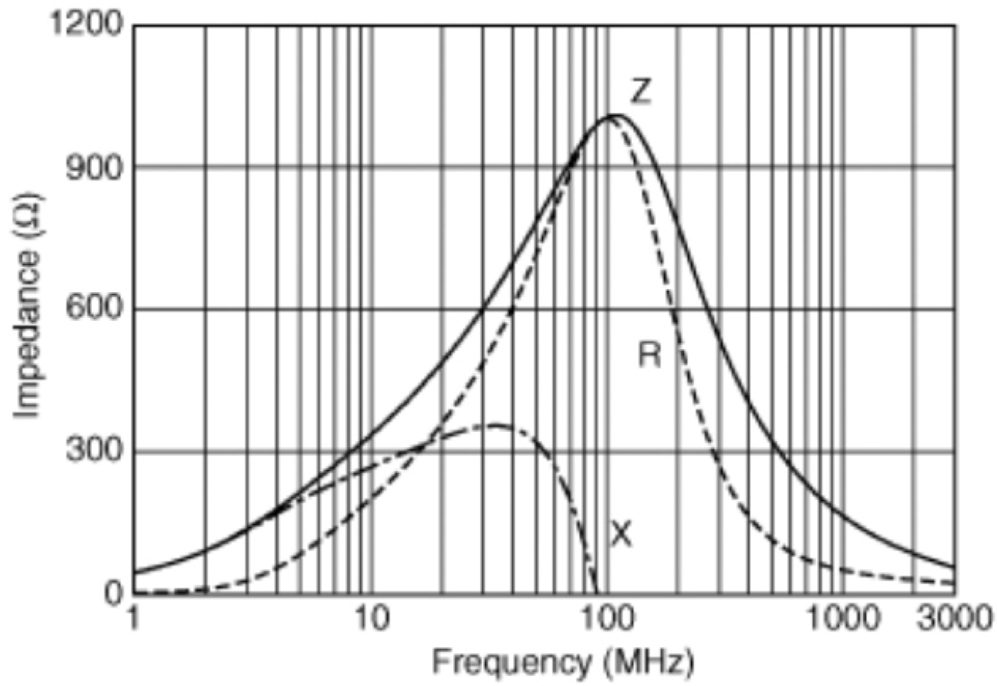


Fig. 7 - The Murata Z vs. Frequency Curve

To model this device we load the C:\MC11W\LIBRARY\FerriteBead_Murata.mdl file from the Model program. We then enter the Z and F values from the above curve.

The more data points you enter the more closely the curve matches the plot above. One thing to remember is that to get a good fit you need a balance of data points over the plot. The optimizer treats each data point with equal weight so it will fit more closely where there are more data points.

After entering a suitable number of points and optimizing, the plot looks like the following.

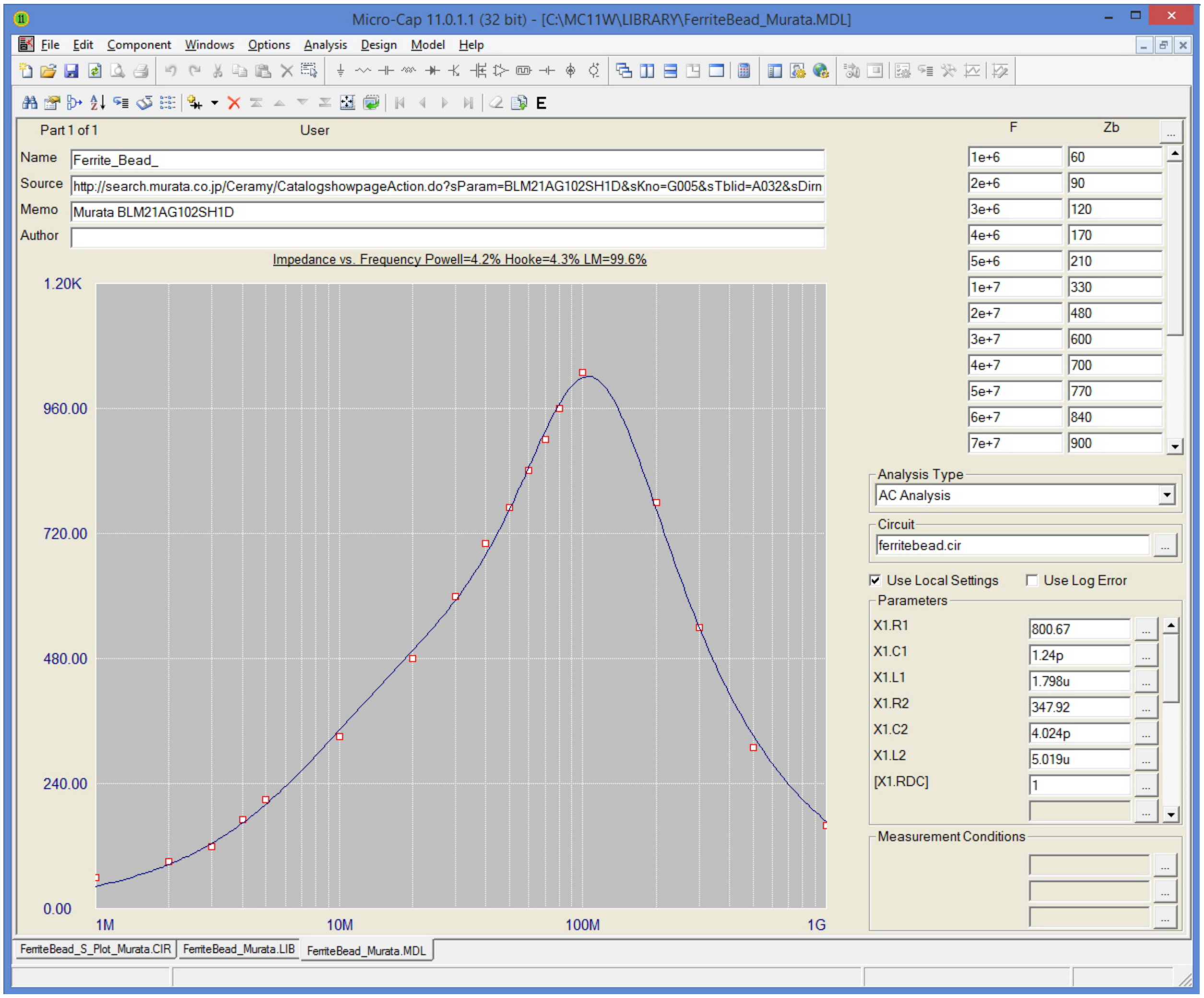


Fig. 8 - Modeling the BLM21AG102SH1D Z vs. Frequency Curve

The optimizer has fitted the seven RLC values to the Z vs. F curve.

The next step is to produce the model. Select Model menu / Create Model for This Part. When the dialog box comes up specify a file name for where to place the model. In this example we'll use the default C:\MC11W\LIBRARY\FerriteBead_Murata.LIB.

When you click on the OK button, the program will create a .SUBCKT model using the circuity name like this:

```
*Murata BLM21AG102SH1D
.SUBCKT Ferrite_Bead_  1 2
L1 1 3 1.798u
R1 1 3 800.67
C1 1 3 1.24p
R2 3 4 347.92
L2 3 4 5.019u
C2 3 4 4.024p
RDC 4 2 1
.ENDS Ferrite_Bead_ *
```

Copy the BLM21AG102SH1D text. Change the subcircuit name by selecting the Ferrite_Bead_ text and then pasting in the BLM21AG102SH1D text. This identifies the subcircuit by the Murata part name. Do the same to the .ENDS text name. The circuit we'll use to show the Z vs. F curve is this one, FerriteBead_Z_Plot_Murata.cir, where we've pasted the entire BLM21AG102SH1D.subckt statement into the model page.

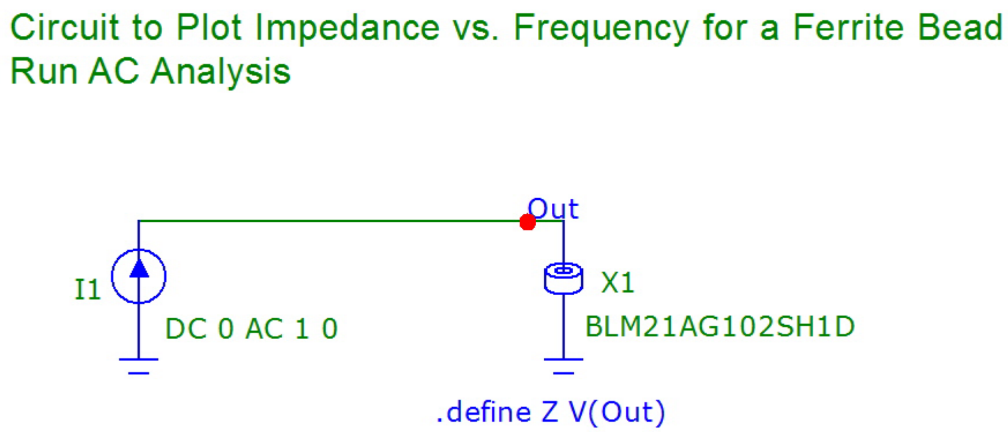


Fig. 9 - The ciruit for plotting the BLM21AG102SH1D Z vs. Frequency Curve

Here is its Z vs. F curve:

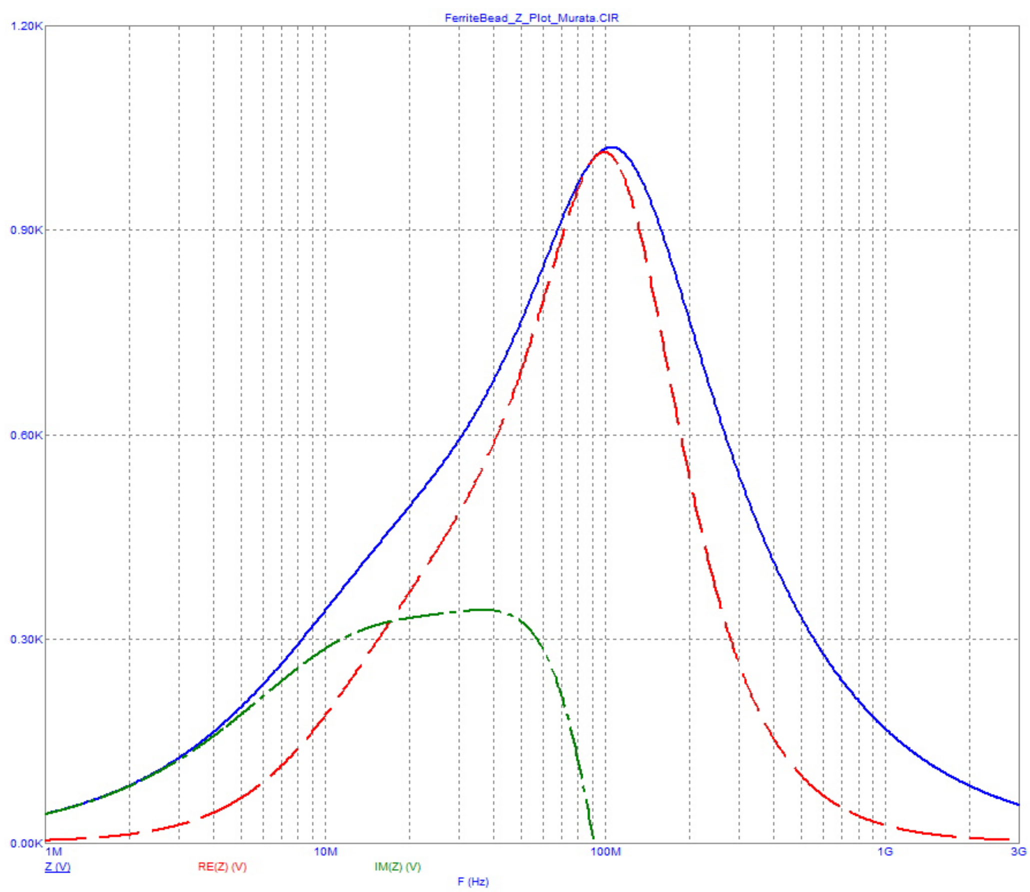


Fig. 10 - The BLM21AG102SH1D Z vs. Frequency Curve

The curve shows a plot of the magnitude of Z, the real part, and the imaginary part. All match well to the original data.

How well will the generated model match the S parameter curves? We'll use this circuit to answer that question.

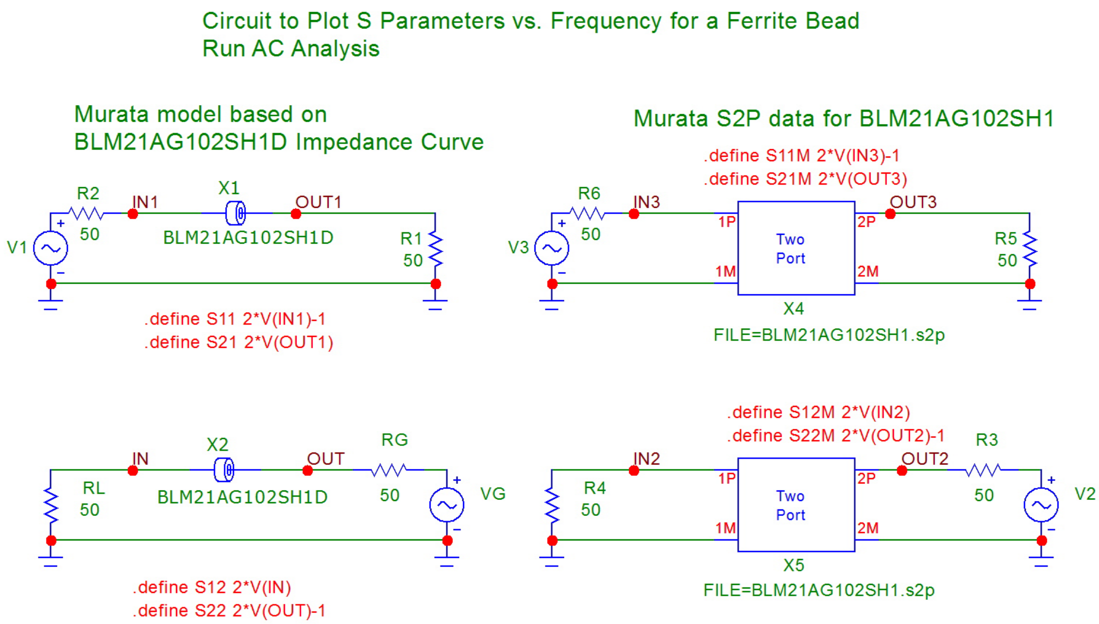


Fig. 11 - Circuit to compare model and measured S parameters

This circuit measures the four S parameters S11, S121, S21, and S22 for both the .subckt model on the left and the measured S parameters provided by Murata as read into the circuit on the right. The .subckt model created earlier has been pasted into this circuit on the Model Page.

The circuit on the right is a collection of 2-ports that model the part by reading in the provided Murata S parameters,

Since the circuit is a linear two port, the S12 and S21 parameters are identical as are the S11 and S22 parameters, so we plot only S11 and S12.

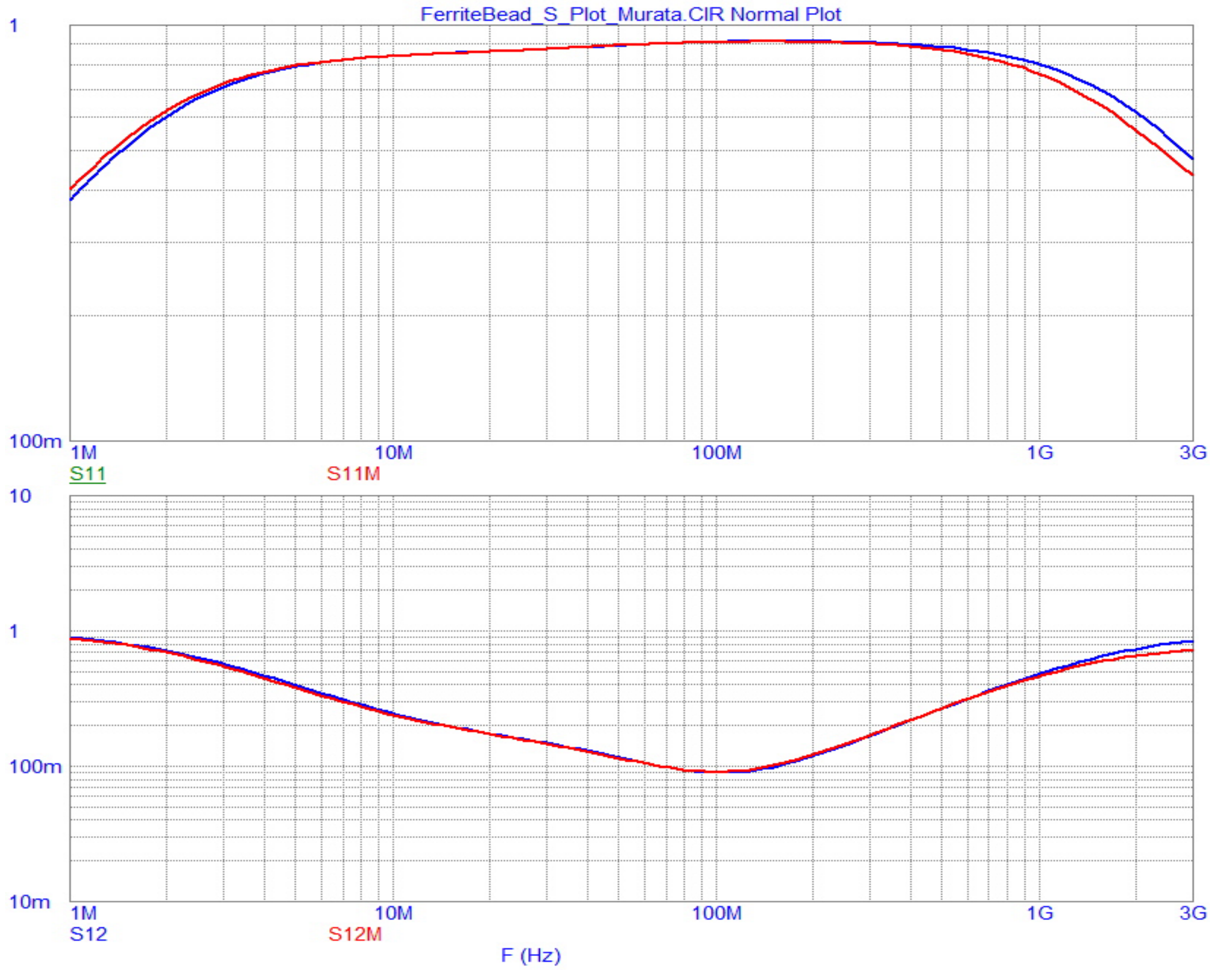


Fig. 12 - Model and measured S parameters

In this plot the blue curves are from the subcircuit model and the red curves are from the Murata S parameter data file.

The match is well within 5% and very similar to the Z vs. F curve match.

What this suggests is that you can match a Z vs. F curve and the resultant model reproduces the S parameter plot as well. That is a direct result of the fact that the model is a linear collection of passive devices, so nonlinearities play no role in the models.

Product Sheet

Latest Version numbers

Micro-Cap 11 .... Version 11.0.1.1
Micro-Cap 10 .... Version 10.1.0.4
Micro-Cap 9 .... Version 9.0.9.1
Micro-Cap 8 .... Version 8.1.4.0
Micro-Cap 7 .... Version 7.2.4

Spectrum's numbers

Sales .... (408) 738-4387
Technical Support .... (408) 738-4389
FAX .... (408) 738-4702
Email sales .... sales@spectrum-soft.com
Email support .... support@spectrum-soft.com
Web Site .... http://www.spectrum-soft.com