



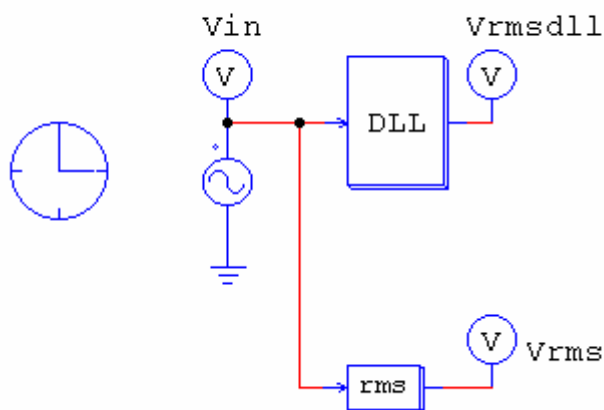
PSIM SIMULATION SOFTWARE

<p>TUTORIAL <i>How to use the DLL Block</i></p>
--

The objective of this tutorial is to help you use DLL Blocks to link C/C++ code with PSIM.

The example circuit is the following (this example can be run with the demo version) :

RMS Calculation from the Built-in Block and from DLL

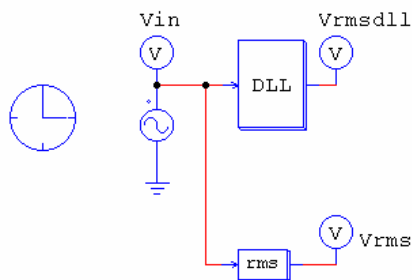


For this tutorial you need a C code that calculates the RMS value of an entry. This code is given with your PSIM version, with the name "msvc_dll.c".

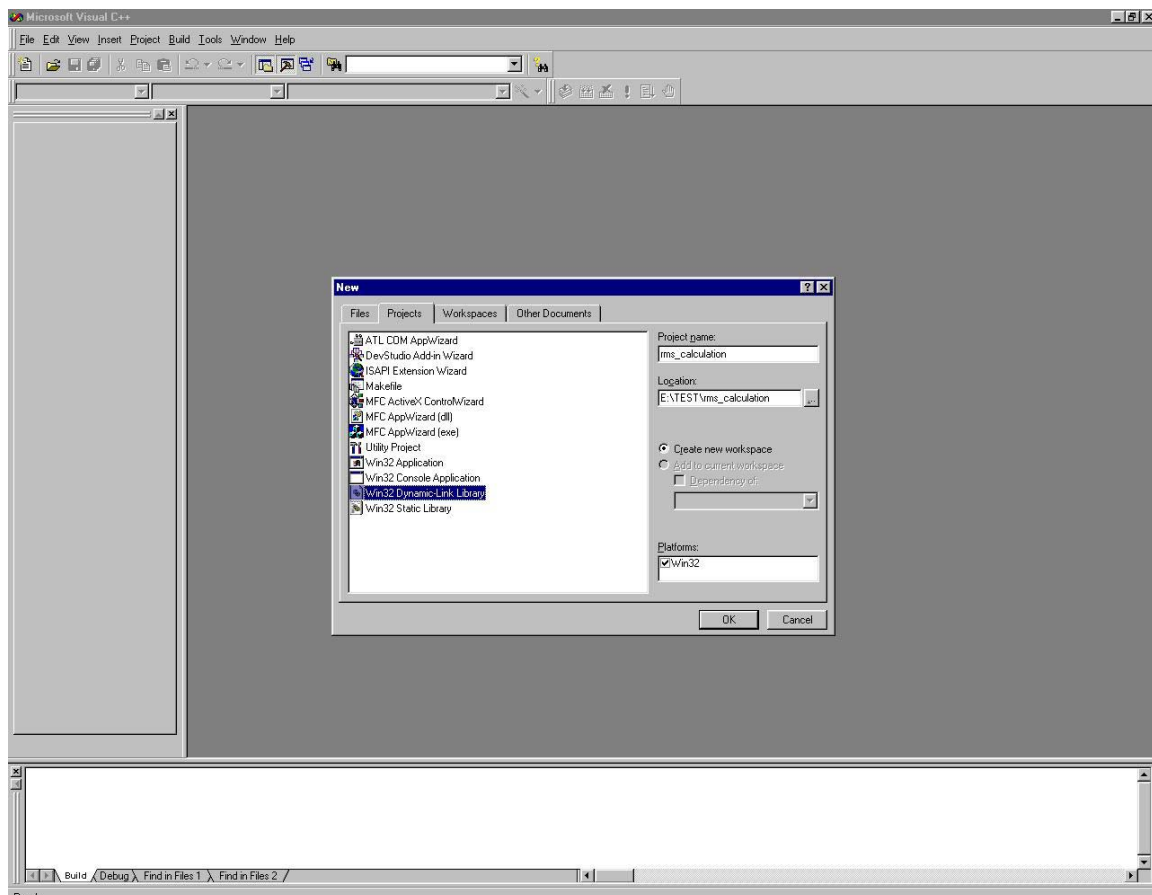
Hereafter are the different steps to use DLL Block :

1. Open your PSIM version.
2. Open the file “dll-rms.sch” given with your PSIM version

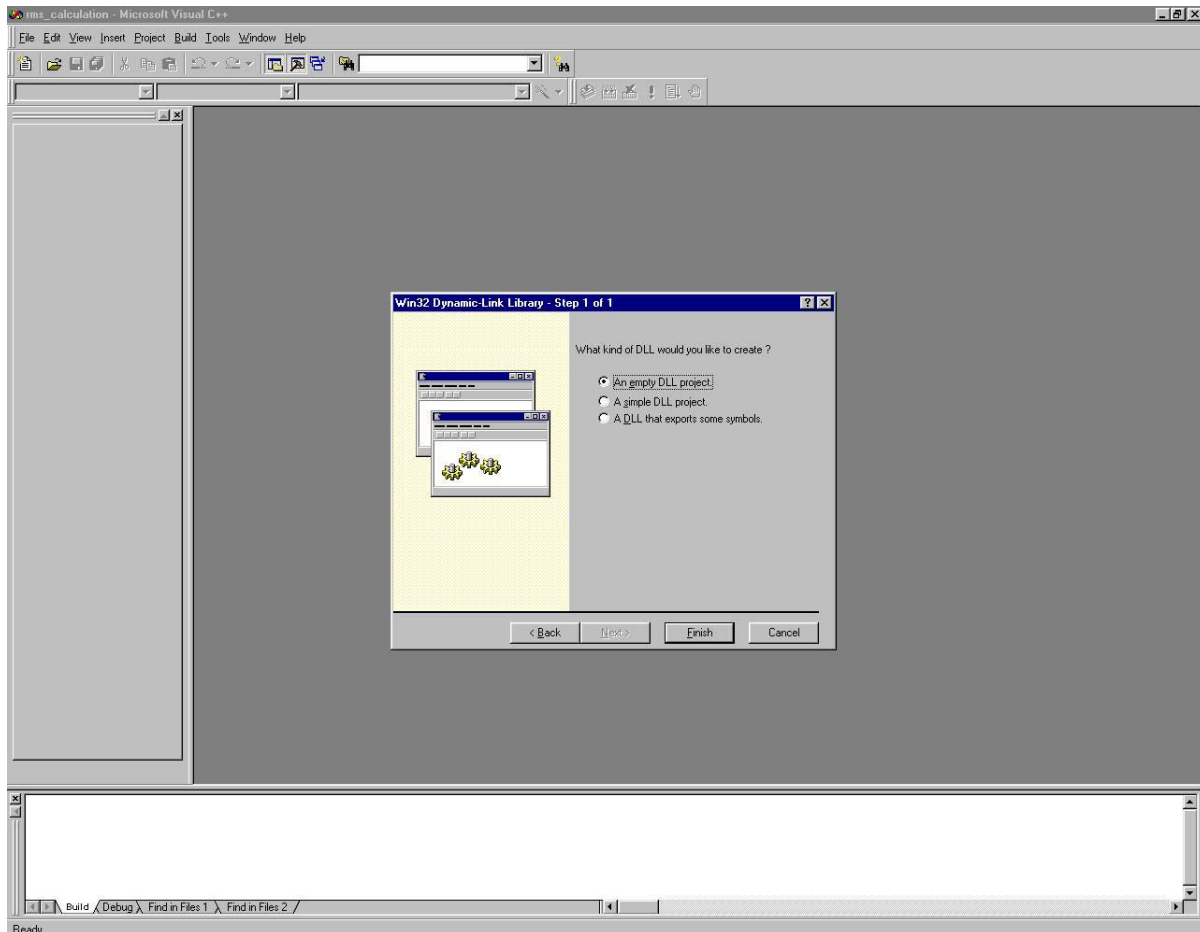
RMS Calculation from the Built-in Block and from DLL



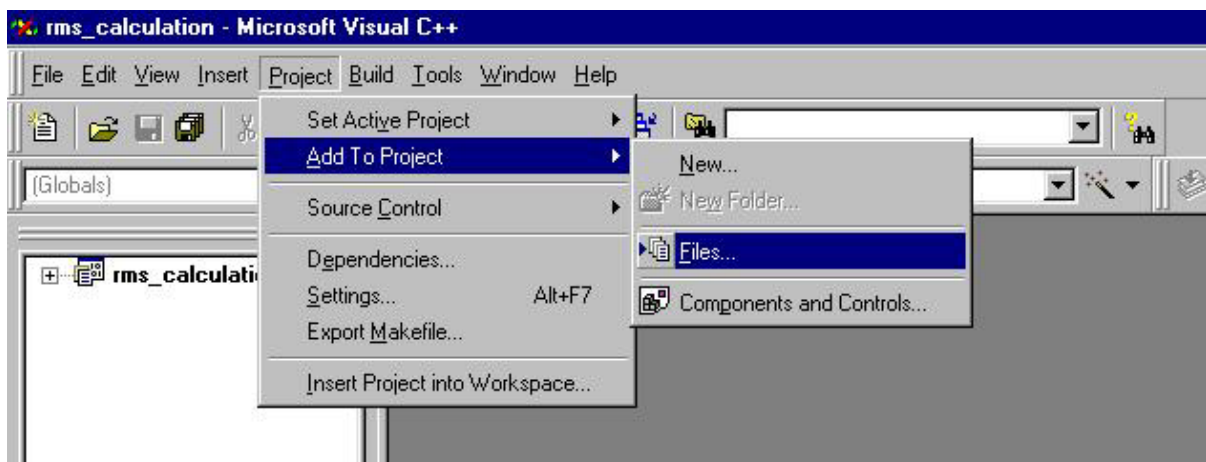
3. Save the file in another directory, for example : “E:\Test\dll-rms.sch”
4. copy the C code “msvc_dll.c” in the same directory (“E:\Test\msvc_dll.c”)
5. Start Visual C++. From the "File" menu, choose "New". In the "Projects" page, select "Win32 Dynamic-Link Library", and set "Project name" as "rms_calculation", and "Location" as “E:\Test\”. Make sure that "Create new workspace" is selected, and "Win32" is selected under "Platform",



6. [for Version 6.0] When asked "What kind of DLL would you like to create?", select "An empty DLL project"



7. From the "Project" menu, go to "Add to Project"/"Files...", and select "msvc_dll.c" which is in the directory "E:\Test\msvc_dll.c".



If you do not use the file "msvc_dll.c" but your own C code, please pay attention to the following :

- You need to have the following variables :
 - t: Time, passed from PSIM, by value
 - delt: Time step, passed from PSIM, by value
 - in: input array, passed from PSIM, by reference
 - out: output array, sent back to PSIM (Note: the values of out[*] can be modified in PSIM)
- The function name has to be "__declspec(dllexport) void simuser (t, delt, in, out)". You may change the variable names (say from "t" to "Time"). But DO NOT change the function name, number of variables, variable type, and sequence.
- The maximum length of the input and output array "in" and "out" is 30.
- Global variables above the function simuser (t,delt,in,out) are not allowed!!!

You then have the following code:

```
#include <math.h>

__declspec(dllexport) void simuser (t, delt, in, out)

// Note that all the variables must be defined as "double"
double t, delt;
double *in, *out;

{
// Place your code here.....begin

// Define "sum" as "static" in order to retain its value.
static double nsum=0., sum=0., rms;
double Tperiod;

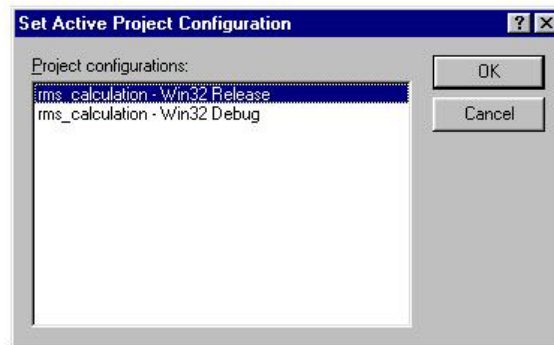
Tperiod=1./60.;

if (t >= nsum*Tperiod)
{
    nsum=nsum+1.;
    rms = sqrt(sum*delt/Tperiod);
    sum=0.;
}

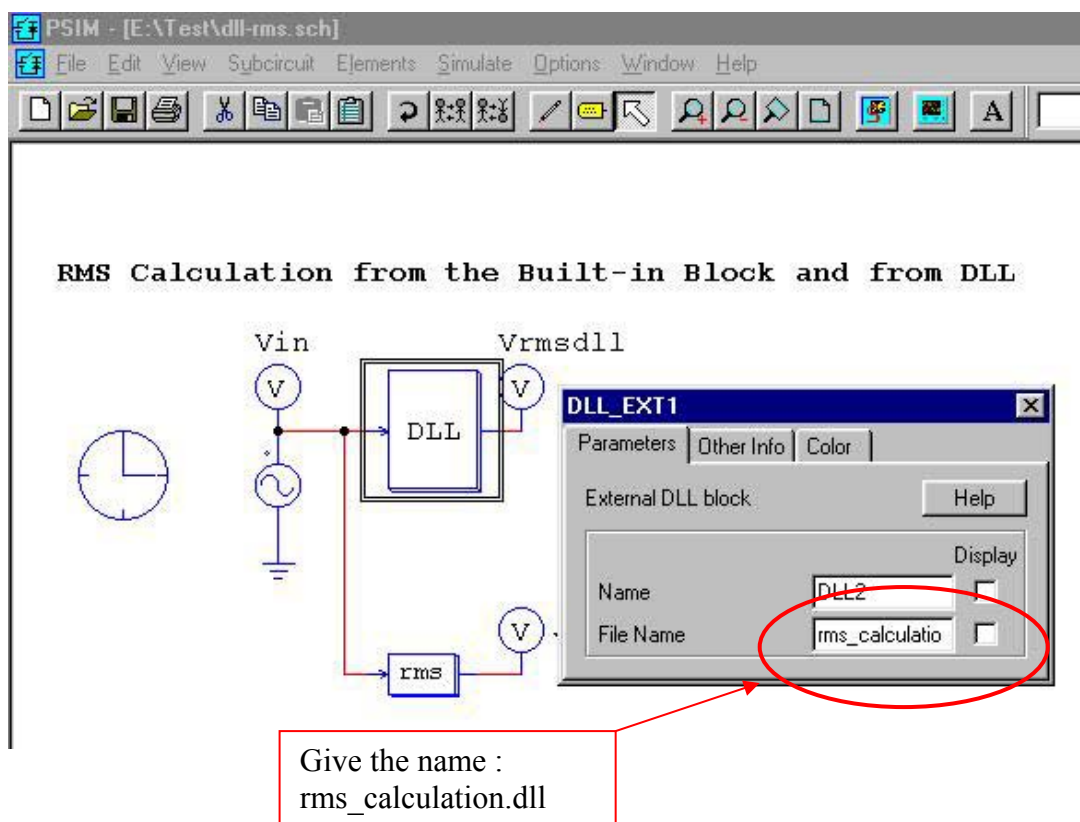
out[0] = rms;
sum=sum+in[0]*in[0];

// Place your code here.....end
}
```

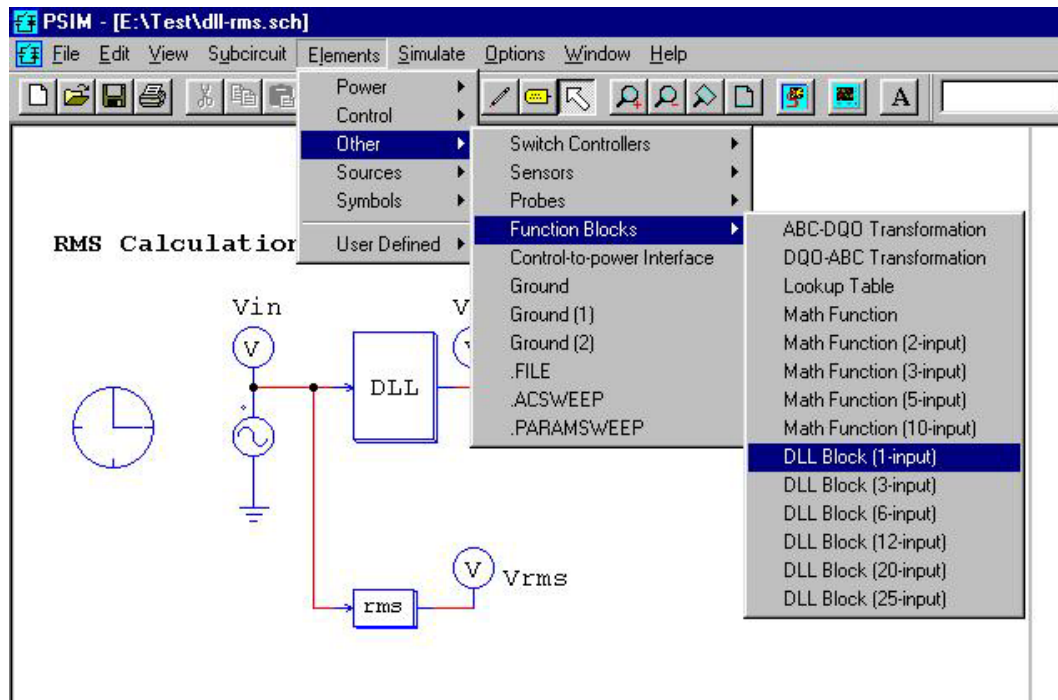
8. From the "Build" menu, go to "Set Active Configurations...", and select "Win32 Release". From the "Build" menu, choose "Rebuild All" to generate the DLL file "rms_calculation.dll". The DLL file will be stored under the directory "C:\Test\rms_calculation\release".



9. Copy the DLL file into the same directory as the schematic file. In this example : "E:\Test\"
10. In the circuit, specify the external DLL block file name



Note that DLL Blocks are in the menu “Elements\Other\Function Blocks\”



11. You are then ready to run PSIM with your own DLL

Additional remarks :

- The input pin with a dot is for the first input in[0]. Note that unused input nodes must be connected to ground.
- PSIM calls the DLL routine at each simulation time step. However, when the inputs of the DLL block are connected to the output of one of these discrete elements (zero-order hold ZOH, unit delay UDELAY, integrator I_D, differentiator D_D, z-domain transfer function block TFCTN_D, and digital filters), the DLL block is considered as a discrete element. In this case, the DLL block is called only at the discrete times.
- The DLL block receives the values from PSIM as the input, performs the calculation, and sends the output back to PSIM. The node assignments are: the input nodes are on the left, and the output nodes are on the right. The sequence is from the top to the bottom.
- The name of the DLL file can be arbitrary. The DLL file, however, must be in the same directory as the schematic file that uses the DLL file.