

- Title: Virtual Prototyping in PSpice
- Product: PSpice

Summary: How to define C/C++, SystemC and Verilog-A components in PSpice. Extension to Hardware in the Loop using Arduino

Version up 17.2

Autor/Datum: Roberto Gandía / 22.07.2016

# Table of Contents

1	Ove	erview	2
	1.1	Contents of this document	2
	1.2	How to use this AN?	2
2	Dev	rice Modeling Interface Template Code Generator	4
	2.1	Required Software and Setup	4
	2.2	How to use Device Modeling Interface?	5
3	Exa	mples	10
	3.1	Digital Power Supply using a C/C++ defined PWM	10
	3.2	FIR Filter using SystemC	26
	3.3	Noise Filter using a MATLAB Block	32
	3.4	Capacitor behaviour analysis defined using VerilogA-ADMS	42
	3.5	Debugging	51
	3.6	Hardware in the Loop using Arduino	60



# 1 Overview

Virtual prototyping is a method in the process of product development, which allows to validate a design before making a physical prototype. Since V17.2, PSpice offers the opportunity to simulate System Designs using different kind of abstractions thanks to the Device Modeling Interface. With GUI, users can define C/C++, SystemC, and Verilog-A components and simulate them in simulator.

## **1.1 Contents of this document**

- How to use Device Modeling Interface.
- Setup for Visual Studio Community 2013.
- How to integrate C/C++, SystemC and Verilog-A models to be simulated in PSpice.
- Debug of C/C++, SystemC and VerilogA devices.
- Importation of MATLAB Blocks in PSpice.
- Hardware in the Loop using Arduino.

## 1.2 How to use this AN?

This document explains the steps for integrating C/C++, SystemC and Verilog-A models with PSpice Device Model Interface (DMI), so that they can be used for PSpice simulations.

This document is valid up Release 17.2. License required for:

- a. PSpice DMI Model development capability:
  - OrCAD PSpice Designer **OR**
  - OrCAD PSpice Designer Plus OR
  - Allegro PSpice Simulator
- b. PSpice DMI Model Simulation capability:
  - OrCAD PSpice Designer Plus **OR**
  - Allegro PSpice Simulator

Examples available:

- Digital Power Supply using C/C++ defined PWM.
- FIR Filter using SystemC.
- Capacitor behavior analysis defined with Verilog-A.
- Noise Filter using a MATLAB Block.
- Hardware in the Loop using Arduino.

The structure of the attached ZIP file is divided in 6 folders:



- Each folder at the same time is divided in two folders:



- You will work with the folder called "To\_be\_completed".
- In each example the reference <directory> is used. It means the directory where you place this ZIP File.

**NOTE**: You can find more specific information clicking on Start  $\rightarrow$  All Programs  $\rightarrow$  Cadence Release 17.2-2016  $\rightarrow$  Documentation  $\rightarrow$  Cadence Help. Search for the next documents:

- pspDMIRef
  - Information about the functions exposed by PSpice Engine and by the DLL files.
- pspcref
  - Information about the meaning of the different parameters you can define for DMI components.

Flow

Application Note



# 2 Device Modeling Interface Template Code Generator

## 2.1 Required Software and Setup

This module covers the information on the setup required for compilation of C/C++, SystemC and Verilog-A models in PSpice.

• SYSTEMC is an environment Variable you need to use if you want to define SystemC components. For that, setup SYSTEMC environment variable pointing to the SystemC installation path.

Systemvariable bearbeiten				
Name der Variablen:	SYSTEMC			
Wert der Variablen:	C:\Cadence\SPB_17.2\tools\pspice\tclscrip			
	OK Abbrechen			

%CDSROOT%\tools\pspice\tclscripts\pspModelCreate\SystemC

e.g.

C:\Cadence\SPB\_17.2\tools\pspice\tclscripts\pspModelCreate\SystemC

• Visual Studio Community 2013

It is a free software that you can download easily here:

https://www.visualstudio.com/en-us/news/vs2013-community-vs.aspx

You will be forwarded to Visual Studio Community 2013 Website. Click on Download and install it.

# Visual Studio Community 2013

November 12, 2014

Visual Studio Community 2013 is a new edition that enables you to unleash the full power of Visual Studio to develop cross-platform solutions. Create apps in one unified IDE. Get Visual Studio extensions that incorporate new languages, features, and development tools into this IDE. (These extensions are available from the Visual Studio Gallery.) Find out more details about Visual Studio Community 2013 here.

Download Visual Studio Community 2013.

After the installation, open the software and verify it works.

**NOTE:** It works with VS Express as well, but the project will not be generated automatically.



- Arduino IDE (Only in case you want to do Hardware in the Loop)
  - You can download the software from Arduino Website.
  - You also need the Hardware, specifically the Arduino Starter Kit.

## 2.2 How to use Device Modeling Interface?

Device Modeling Interface Template Code Generator can be launched from Model Editor in PSpice Accessories.





clicking on Model  $\rightarrow$  DMI Template Code Generator.



		Application I
DMI Templat	e Code Generator	
Use this dialog- SystemC. The d Recommended 1. Test the 2. Create th 3. Use the in the sc	box to auto-generate DMI template lialog-box also imports the Verilog- <b>i steps:</b> model code stand-alone by buildin he PSpice-DMI adapter code, and e generated PSpice library (.lib file) to hematic for PSpice simulation.	code for the following PSpice-DMI models: Analog, Digital, and A Compact Device models using ADMS. Ig an exe. edit it in Visual Studio to insert model code. to create a schematic symbol.The generated symbol can be placed
Part Details —		
	PartName	customPart
	Part Type	Digital C/C++
Ports		
	Interface Type	Combinatorial
	Port Entry	Ports O CSV File
Parameters -		
	Global Parameters	
	Device Parameters	
Output		
	DLL File Name	customPart.dll
	Log File Name	customPart.log
	DLL Location	C:\Users\rgandia.FC-EDA\DMIM Browse
		OK Cancel Help
PSpice DMI Te	emplate Generator	

This interface is divided into 4 sections:

- Part Details:
  - $\circ\,$  Part Name: Name for the function you want to define. For example, Pulse Width Modulation.
  - Part Type: Define the type of component you want to describe.

Digital C/C++	-
Digital C/C++	
SystemC	
Analog	
VerilogA-ADMS	

- Ports:
  - Interface Type: Classify your component as Clocked or as Combinatorial.

Flow(



• Port Entry: Define the inputs and the outputs of your device through an excel file or writing down them directly.

Port Entry				
<ul> <li>Specify f</li> <li>Specify f</li> <li>Initial val</li> <li>CLK port</li> </ul>	he port count for l Port size for vecto lues are used to in t is automatically	nput and IO Port r ports, default si nitialize ports in created for Clock	s. ze is 1. device constru ked interface t	uctor code. ype.
	Enter number	r of input ports	2	
	Enter number	r of IO ports	02	2
Port Name	Port Type	Port Size	Default Value	Port Description
Port0	Input	1	0	
Port1	Input	1	0	
Port2	IO	1	0	
Dort2	10	1	0	

- Parameters:
  - Global Parameters: Parameters that can be used by all the components which define them.

<ul> <li>Specify the g</li> </ul>	lobal parameters	which will be	used by the device logic.
These param	neters need to be	defined in the	top-level schematic - DMI model will
get their valu	les from PSpice.		
<ul> <li>Default value</li> </ul>	e will be used to i	nitialize the pa	rameters in device constructor code.
Enter number	of parameters	2	
Enternamber	orparameters	3	
Parameter Name	Parameter Type	Default Value	Parameter Description
	double	0	
Param0	uoubie		
Param0 Param1	double	0	

• Device Parameters: Parameters used just for a particular device.

				Application Note
Device Parameters	5			
Specify the E	)evice parameter	rs.	<u>∫</u> <sub>8</sub>	
Enter number	rofparameters	5		
Parameter Name	Parameter Type	Default Value	Parameter Description	
Param0	double	0		
Param1	double	0		
Param2	double	0		
Param3	double	0		
Param4	double	0		

**Flow**CA

• Output: Define the name for your DLL, the log file where information of the generation of the DLL is going to be written and the directory where it is going to be saved.

# 3 Examples

## 3.1 Digital Power Supply using a C/C++ defined PWM

This module shows a Digital Power Supply design, which uses models with different levels of abstraction.

This example demonstrates:

- The generation of template code for a DMI Model and the implementation of a Digital PWM Control block.
- Simulation of the DMI model in the context of a Digital Power Supply circuit.

Steps:

- 1. Launch Model Editor
- 2. Select Menu Item Model → DMI Template Code Generator
- 3. Enter the data as follows:

DMI Templa	ate Code Generator	× 1
Use this dialog SystemC. The <b>Recommende</b>	g-box to auto-generate DMI template dialog-box also imports the Verilog- id steps:	e code for the following PSpice-DMI models: Analog, Digital, and A Compact Device models using ADMS.
<ol> <li>Test the</li> <li>Create</li> <li>Use the in the s</li> </ol>	e model code stand-alone by buildin the PSpice-DMI adapter code, and a generated PSpice library (.lib file) t chematic for PSpice simulation.	ig an exe. edit it in Visual Studio to insert model code. to create a schematic symbol.The generated symbol can be placed
Part Details		
	PartName	PWMControl
	Part Type	Digital C/C++
Ports		
	Interface Type	Clocked
	Port Entry	Ports O CSV File
Parameters		
	Global Parameters	
	Device Parameters	
Output		
	DLL File Name	PWMControl.dll
	Log File Name	PWMControl.log
	DLL Location	D:\PSpice\Application Notes\Floi Browse
		OK Cancel Help
Specify device	e parameters; and global paramete	rs required by the model logic



4. Click on CSV File to enter ports using a csv file

Interface Type	Clocked	
Port Entry	O Ports O CSV File	

5. Browse to the csv file **portsv.csv** 

You will find this file in

<directory>\Circuits\Digital Power Supply\To\_be\_completed\Ports

8

1

Port Entry				2	
<ul> <li>The csv file needs to follow the following syntax:</li> <li><port name="">, <port input io="" type:="">, <port size="">, <initial value="">, <port description=""></port></initial></port></port></port></li> <li>For example,</li> </ul>					
		IN1, INPUT OUT, IO, 8,	, 1, X, Input Port 1 , 0, 10 Port 1		
	Select your C	SV file here	)MI\DigitalPowerSupp	Browse	
Port Name	Port Type	Port Size	Default Value	Port Description	
CLK	INPUT	1	0	Clock	
FB	INPUT	8	0	Feedback input	

The ports are automatically read from the csv file and populated in the form as above.

0

0

Reference input

Output Pulse Width

6. Review the port list and click OK.

REF

PW

7. Enter Global parameters as shown below:

INPUT

10

			Appli	cation Note
Global Parameters				
<ul> <li>Specify the global parameters</li> <li>These parameters need to be get their values from PSpice.</li> <li>Default value will be used to i Enter number of parameters</li> </ul>		which will be defined in the nitialize the pa	used by the device logic. e top-level schematic - DMI model will rameters in device constructor code.	
Parameter Name	Parameter Type	Default Value	Parameter Description	
PER	double	0	Period	
D	double	0	Duty Cycle	
			OK Cancel Apply	

FlowC

- 8. Click OK.
- 9. Before generating the whole files, browse to directory to locate them.

<directory>\Circuits\Digital Power Supply\To\_be\_completed\DMI\_Code

**NOTE:** This file does not generate any DLL. The DLL will be generate it in Visual Studio when the code is compiled.

10. Click OK and generate the files:

**NOTE:** A message might appear indicating that it could not be possible to find the lib. This message can be clicked away and ignored.

Flow<mark>CAD</mark>

Application Note

tran.out | capDMI.net | cap.lib | tran.out | psppspPWMControl.h | pspPWMControllog\* |

r Format Column Macro Advanced <u>W</u> indow <u>H</u> elp
Reading json
Json readLogging Model Creation
reading Model Name and Type
reading global and instance params for digital and analog
declaring the global variables
reading instance paramas.
declaring and defining instance paramas
Copying common files
Creating Digital Kodel
Getting port information
Initializing input port specific variables
Akaptura/DESIGN1-FSpiceFles/capDMIktran/tran.cir 10 variables writing port specific code in opp file
Changes in evaluate function
Digital/SystemC GUID creation
read input file CommonTemplate/pspEngFunc_model.cpp
opened output file D:/customers/Denmark2016/DNI/DigitalPowerSupply/Code//pspEngFung.cpp for writing
read input file CommonTemplate/pspEngFunc_model.h
opened output file D:/customers/Denmark2016/DMI/DigitalPowerSupply/Code//pspEngFung.h for writing
read input file DigitalTemplate/pspDigitalNodelName.cpp
opened output file D:/customers/Denmark2016/DNI/DigitalPowerSupply/Code//psppspPWMControl.cpp for writing
read input file DigitalTemplate/pspDigitalNodelName.h
opened output file D:/customers/Denmark2016/DNI/DigitalPowerSupply/Code//psppspPWMControl.h for writing
read input file DigitalTemplate/digitalmodel_user.cpp
opened output file D:/customers/Denmark2016/DNI/DigitalPowerSupply/Code//pspPUMControl_user.cpp for writing
Populating vcproj with the source and header files
read input file DigitalTemplate/DigitalModelName.vcxproj
Lib greated

11. The new created files are located in a new folder called PWMControl. You can find this folder in

<directory>\Circuits\Digital Power Supply\To\_be\_completed\DMI\_Code

In this folder, the created files are organized in two folders:



In code you will find the Visual Studio project files. With them you will generate the DLL:

pspEngFunc.cpp
 pspEngFunc.h
 PSpiceBase.h
 PSpiceCMIApiDefs.h
 PSpiceCommonAPIDefs.h
 PSpiceDigApiDefs.h
 pspPWMControl.cpp
 pspPWMControl.h
 PWMControl.log
 PWMControl.vcxproj
 PWMControl\_user.cpp
 StdAfx.cpp
 StdAfx.h

In lib you will find the PSpice Model (PWMControl.lib).



Application Note

Models List 🛛 🗵	subckt X_PWMControl CLK FB_0 FB_1 FB_2 FB_3 FB_4 FB_5 FB_6 FB_7 REF_0
Model Name Type Mod	REF_1 REF_2 REF_3 REF_4 REF_5 REF_6 REF_7 PU
X_PWMControl SUBCKT	+ OPTIONAL: DPWR=\$G_DPWR DGND=\$G_DGND
	+ PARAMS:
	.model PUMControl_TIMING ugate (
I	+ tplhmn=6ns tplhty=9ns tplhmx=15ns
	+ tphlmn=6ns tphlty=10ns tphlmx=15ns
1	+)
I	U1 LOGICEXP( 17, 1 ) DPWR DGND
	+ CLK FB 0 FB 1 FB 2 FB 3 FB 4 FB 5 FB 6 FB 7 REF 0 REF 1 REF 2 REF 3
I	REF 4 REF 5 REF 6 REF 7 PW
I	+ PWNControl TIMING IO STD
I	+ C NODEL: PUMControl dll PUMControl
I	
I	+ PARAMS:
I	ends
1	1
1	1

**Note:** The generated model already points to the PSpice-DMI dll PWMControl.dll, although it still has to be generated – For that, the next step is to describe the model code and generate the dll.

- 12. Launch Visual Studio Community 2013
- 13. Click File  $\rightarrow$  Open  $\rightarrow$  Project/Solution

Search for PWMControl.vcxproj located in:

 $<\!\!directory>\!\!\backslash Circuits \\ Digital PowerSupply \\ To\_be\_completed \\ DMI\_Code \\ PWMControl \\ ode$ 

Community 2013	Disq	Open Project	lia Community 20	17			23
	Get the	Circuits > D	igital Porwer Supply 🕨 Completed 🕨	VS_Code	✓ 4 VS_Code dur	chsuchen	٩
Start	New to	Organisieren • Neuer Org	Iner				0
New Project	Create a	🕹 Musik 🔦	Name	Änderungsdatum	Тур	Größe	
Open Project Open from Source Control	See how Discover	OneDrive Suletzt besucht	PWMControl.vcxproj	13.06.2016 16:29	VC++ Project		8 KB
lution Explorer		Bibliotheken					
00G <b>x</b>		S Bilder					
		Dokumente =					
		Videos					
		Scomputer					
		😻 Windows7_OS (C:)					
		🥪 Data2 (D:)					
		🛫 support (\\dc01) ( + 🔞		III			•
		Dateiname			<ul> <li>All Project Files (*</li> </ul>	.sin;*.dsw;*	• •
					Öffnen	Abbrech	en

14. Change the default configuration on the top to Release x64





and Build the solution to verify that there are no build issues.



ıt		
output from:	Build	- 監 監 ■
Build s	started: Project: PWMControl, Configuration:	Release x64
pspEngFunc.	cpp	
pspPWMContro	pl.cpp	
PWMControl_u	user.cpp	
StdAfx.cpp		
Creating	library D:\PSpice\Application Notes\FlowCAD	_AN_Device_Modeling_Interface\Circuits\Digital Porwer Supply\Compl
Generating d	code	
Finished ger	nerating code	
PWMControl.	<pre>/cxproj -&gt; D:\PSpice\Application Notes\FlowC d: 1 succeeded, 0 failed, 0 up-to-date, 0 sk</pre>	AD_AN_Device_Modeling_Interface\Circuits\Digital Porwer Supply\Com ipped
	t voutput from: Build s pspEngFunc.o pspPWMControl_ StdAfx.cpp Creating Generating of Finished ger PWMControl.v Build	t voutput from: Build Build started: Project: PWMControl, Configuration: pspEngFunc.cpp pspPWMControl_cpp PWMControl_user.cpp StdAfx.cpp Creating library D:\PSpice\Application Notes\FlowCAD Generating code Finished generating code PWMControl.vcxproj -> D:\PSpice\Application Notes\FlowCA 

- 15. Edit the file PWMControl\_user.cpp
- 16. Go to evaluate function by searching for the text "pspPWMControl::evaluate". Now, search for "// LOGIC TO BE IMPLEMENTED BY USER"

This is the place where you will add the model evaluation code.

17. Open the file **COPY.TXT** located in

<directory>\Circuits\DigitalPowerSupply\To\_be\_completed\DMI\_Code



Application Note

copy the code that there is inside and paste it just after //LOGIC TO BE IMPLEMENTED BY USER.

```
// LOGIC TO BE IMPLEMENTED BY USER
          oldPW = PW;
          pspBits2Int(FB, FBInt, 8);
          pspBits2Int(REF, REFInt, 8);
          if (REFInt > FBInt && mD < 0.98) {
              mD += 0.01;
          3
          else if (REFInt < FBInt && mD > 0.02) {
              mD -= 0.01;
              //fprintf(stderr, "Reducing DutyCycle\n");
          }
          if (mCurrentCLKCount <= 0) {</pre>
              mCurrentCLKCount = mPER;
          }
          if (mCurrentCLKCount > mD * mPER)
              mPWStatus = false;
          else
              mPWStatus = true;
          if (mPWStatus == true && (int)PW != 1){
              PW = pspBit::HI;
          }
          else if (mPWStatus == false && (int)PW != 0){
              PW = pspBit::LO;
          }
          mCurrentCLKCount--;
```

fprintf(stderr, "FBInt=%d REFInt=%d mDutyCycle=%g PW=%d\n", FBInt, REFInt, mD, (int)PW);

- 18. Save the project.
- 19. This code uses some extra variables which need to be declared edit file pspPWMControl.h and add the following lines at the end before the last closing brace.



20. Build the solution again. The model dll is now built with the required model evaluation code.



Flow

Application Note

### **NOTE:** Do not consider the warning.

21. Go to

 $<\!\!directory>\Circuits\DigitalPowerSupply\To\_be\_completed\DMI\_Code\PWMControl\code\x64\Release$ 

and verify that the DLL and the PDB has been generated:

NWMControl.tlog
🕼 pspEngFunc.obj
🕏 pspPWMControl.obj
PWMControl.Build.CppClean.log
PWMControl.dll
PWMControl.exp
PWMControl.lib
PWMControl.log
PWMControl.pdb
🕏 PWMControl_user.obj
🕏 StdAfx.obj
vc120.pdb

- 22. Open OrCAD Capture and click File  $\rightarrow$  Open  $\rightarrow$  Project.
- 23. Open the Capture project located in <a href="https://circuits/Digital\_Power\_Suppy/To\_be\_completed/circuit/Power\_Supply\_inc">directory>Circuits/Digital\_Power\_Suppy/To\_be\_completed/circuit/Power\_Supply\_inc</a> omplete.dsn:



Click on the component Software Controlled Switch with RMB and click on Descend Hierarchy:



Now you have to create the Symbol for the PSpice component you have just described:  $\ensuremath{\mathsf{PWM Block}}$ 



24. Open Model Editor and click on Done.

**FlowCAD** 



25. Click on File  $\rightarrow$  Open and look for PWMControl.lib

 $<\!\!directory>\!\!\backslash Circuits \\ DigitalPowerSupply \\ To\_be\_completed \\ DMI\_Code \\ PWMControl \\ b$ 

26. Click on File  $\rightarrow$  Export to part library and click OK.

Enter In	out Model Library:
Porwer	Supply/Completed/Library_Rob/PWMControl.lib
	devid Devid Library a
Enter	inful Fan Linfan/
Enter O	aput Part Library.
Porwer	Supply\Completed\Library_Rob\PWMControl.olb Browse.

With this option you are creating the schematic part for this PSpice component automatically:

PWMControl.err	
PWMControl.lib	
PWMCONTROL.OLB	

- 27. Come back to the Schematic and place the symbol you have just created, modifying this symbol as seen below or using the one from
  - <directory>\Circuits\Digital Power Supply\To\_be\_completed\Library\Capture





28. Place the symbol and connect it in the circuit:



29. Define a Transient Analysis Simulation and call it Trans:

✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓      ✓
New Simulation
Name: Create
Cancel
none 👻 📖
Root Schematic: SCHEMATIC1

30. Define Run to Time = 5ms

Simulation Settings - Trans General Analysis Configuration	Files Options Data Collection Probe Window	
Analysis type: Time Domain (Transient) 💌	Run to time: 5ms seconds (T	"STOP)
Options: General Settings	Start saving data after: 0 seconds Transient options	
Monte Carlo/Worst Case Parametric Sweep Temperature (Sweep)	Maximum step size: seconds	SKIPBP)
Save Blas Point Load Blas Point Save Check Points	Run in resume mode Outp	ut File Options
	OK Abbrechen Übernehme	en Hilfe

**Flow**C

31. Click on Configuration Files. You have to import the PSpice model for the component you have just created and the already defined ones. Click on Browse, select one by one and click on Add to Design.

You will find your library PWMControl.lib in:

<directory>/Circuits/DigitalPowerSupply/To\_be\_completed/DMI\_Code/PWMControl/li b

You will find the other libraries in <directory>/Circuits/Digital Power Supply/To\_be\_completed/Library/Capture:

- a. Noisecomp.lib
- b. Matlab1.lib
- c. Matlab.lib

Category:	Details	
Stimulus		Browse
Include	Configured Files	
<ul> <li>Update Index</li> </ul>	\\ibrary\pwmcontrol.lib     \\ibrardAsiassamp.lib	Add as Global
	Nibrary/matlab1.lib	Add to Design
	Ilibrary\matlab.lib nom.lib	Add to Profile
		Edit
		Change
	< III ►	
	"C:\Cadence\SPB_17.2\tools\PSpice\Library"	Browse



32. Before running the simulation, do **NOT** forget to associate all the .DLL's, which describe the behaviour of your components. For that, place the .DLL and the .PDB in the Simulation Profile Folder that you have just created:

<directory>\Circuits\DigitalPowerSupply\To\_be\_Completed\circuit\Power\_Supply\_Un complete-PSpiceFiles\SCHEMATIC1\Trans

### NOTE:

Your created .DLL is in:

<directory>/Circuits/DigitalPowerSupply/To\_be\_completed/DMI\_Code/PWMControl/c ode/x64/Release

1	PWMControl.tlog
°Ę	pspEngFunc.obj
łę,	pspPWMControl.obj
	PWMControl.Build.CppClean.log
	PWMControl.dll
	PWMControl.exp
	PWMControl.lib
	PWMControl.log
	PWMControl.pdb
15	PWMControl_user.obj
°Ę	StdAfx.obj
Ð	vc120.pdb

The other one is in:

<directory>/Circuits/DigitalPowerSupply/To\_be\_completed/Library/DLL

**NOTE**: There is another option to store your created DLL's and PDB's using an Environment Variable. It allows you to use a central library for all your DLL's.

Systemvariable bear	beiten 🔀
Name der Variablen:	CDN_PSPICE_MODEL_PATH
Wert der Variablen:	<pre>:e_Modeling_Interface\Circuits\DLL_Library</pre>
	OK Abbrechen

In this Application Note however, where are placing them in the Simulation Profile Folder.

33. Click on Run:





34. Analyze the results. Click on PSpice → Markers → Voltage Level and place it in the node called VOUT.

PSpice Accessories Options	window Help
Mew Simulation Profile	
Edit Simulation Profile	
☑ <u>R</u> un F11	
View Simulation Results F12	
Vie <u>w</u> Output File	
Create Netlist	
View Netlist	
Advanced Analysis	•
Markers	Voltage Level
Bias Points	Voltage Differential
	Current Into Pin
NAME***	Bower Dissipation
PARAMETERS:	Advanced
PER = 10 D = 0.2	Plot Window <u>T</u> emplates
	Show All
50Vdc 🕂 Vs2	Hide All
	Delete All
	List
	<u>L</u> ist
L4	
· · · · · · · · · · · · · · · · · · ·	VOUT
47uH · · · · · · · · · · · ·	
	0 0000000000000000000000000000000000000
150	2 2022222222
	· · · · · · · · · · · · · · · · · · ·
	RLoad
	\$ 50
U2	1 11 11 11 11 11 18 11 19 11 11 1
Noisson	5 13 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
NOISECOMP	
	≑ C4

You will be able to see this results in the Probe Window:



**Flow**C

35. In Probe Window click on Plot  $\rightarrow$  Add plot to Window.



36. Go to OrCAD Capture again, and place another Voltage Marker in the wire called PW:



37. Visualize the results in the Probe Window:



**NOTE:** In this case everything has worked perfectly, but most times it is necessary to debug for detecting error and mistakes in the source code description. More information in section 3.3.

# 3.2 FIR Filter using SystemC

This module explains definition and simulation of a SystemC model using PSpice-DMI.

It demonstrates:

- An example of a FIR model written in SystemC.
- Generation of DMI Template code for SystemC models using Model Editor.

Steps:

- 1. Launch Model Editor
- 2. Select Menu Item Tools/DMI Template Generator
- 3. Enter data as below:

	FlowCAD
	Application Note
DMI Template Code Generator	×
Use this dialog-box to auto-generate DMI template code for SystemC. The dialog-box also imports the Verilog-A Comp Recommended steps:	or the following PSpice-DMI models: Analog, Digital, and bact Device models using ADMS.
<ol> <li>Test the model code stand-alone by building an ex</li> <li>Create the PSpice-DMI adapter code, and edit it in</li> <li>Use the generated PSpice library (.lib file) to create in the schematic for PSpice simulation.</li> </ol>	e. Visual Studio to insert model code. a schematic symbol.The generated symbol can be placed
Part Details	
PartName	FIR
PartType	SystemC 💌
Ports-	
Interface Type	Clocked
Port Entry	Ports O CSV File
Parameters	
Global Parameters	
Device Parameters	
Output	
DLL File Name	FIR.dll
Log File Name	FIR.log
DLL Location	D:\PSpice\Application Notes\Flov
	OK Cancel Help
PSpice DMI Template Generator	

Do not click OK.

4. Click on Ports radio button to enter the following port data:

### Port Entry

- · Specify the port count for Input and IO Ports.
- · Specify Port size for vector ports, default size is 1.
- Initial values are used to initialize ports in device constructor code.
- · CLK port is automatically created for Clocked interface type.

	Enter number	2		
	Enter number	1		
Port Name	Port Type	Port Size	Default Value	Port Description
CLK	Input	1	0	Clock Port
input	Input	16	0	
output	IO	16	0	

- 5. Click OK to close the Port Entry and click OK to generate the code.
- 6. The PSpice library for the generated model is created automatically. The vector ports are expanded to PSpice-supported scalar ports in the lib file:

Models List	×	.subckt X_FIR CLK input_0 input_1 input_2 input_3 input_4 input_5 input_
Model Name	Type Modi	6 input_7 input_8 input_9 input_10 input_11 input_12 input_13 input_14
X_FIR	SUBCKT	input_15 output_0 output_1 output_2 output_3 output_4 output_5 output_6
		output_7 output_8 output_9 output_10 output_11 output_12 output_13
		output 14 output 15
		+ OPTIONAL: DPWR=\$G_DPWR_DGND=\$G_DGND
		+ PARAMS:
		.model FIR TIMING ugate (
		+ tplhmn=6ns tplhty=9ns tplhmx=15ns
		+ tphlmn=6ns tphlty=10ns tphlmx=15ns
		+ )
		U1 LOGICEXP( 17, 16 ) DPWR DGND
		+ CLK input_0 input_1 input_2 input_3 input_4 input_5 input_6 input_7
		input_8 input_9 input_10 input_11 input_12 input_13 input_14 input_15
		output 0 output 1 output 2 output 3 output 4 output 5 output 6 output 7
		output 8 output 9 output 10 output 11 output 12 output 13 output 14
		output 15
		+ FIR TIMING IO STD
		+ C MODEL: FIR.dll FIR
		+ PARAMS:
		.ends

This library points to the PSpice-DMI dll FIR.dll – The next step is to complete the model code and generate this dll.

7. Launch Visual Studio Community 2013 and click on Open Project.

#### Select

<directory>\Circuits\FIR\_Filter\To\_be\_completed\DMI\_Code\FIR\code\pspSysCFIR.v
cxproj

8. Change the default configuration on the top to Release x64, and Build the solution to verify that there are no build issues.

**NOTE:** There are many warnings, but you can omit them.

FlowC

Application Note

х



9. Edit SysCFIR.cpp

```
_void SysCFIR::entry() {

=//
      const sc_uint<8> coef[5] = { 18, 77, 107, 77, 18 };
     sc int<16> taps[5];
 11
 11
     output.write(0);
 11
      wait();
      while (true) {
 11
         for (int i = 4; i > 0; i - -) {
 11
 11
             taps[i] = taps[i - 1];
 11
         }
 11
         taps[0] = input.read();
 11
         sc_int<16> value;
 11
         for (int i = 0; i < 5; i++) {
 11
             value += coef[i] * taps[i];
 11
         }
 11
         output.write(value);
 11
         FILE* fp = fopen("out.vcd", "a");
 11
 11
         fprintf(fp, "\n%d %d %d", value);
 11
         fclose(fp);
 11
         cout << "Time[" << sc_time_stamp() << "] Value[0x"<< he</pre>
 11
 11
         wait();
 11
      }
 }
```

Search for SysCFIR::entry function in SysCFIR.cpp and uncomment the sample code inside the function clicking on Edit  $\rightarrow$  Advanced  $\rightarrow$  Uncomment. This code implements an FIR filter using SystemC.

10. Edit pspSysCFIR.cpp:

```
>pspSysCFIR::pspSysCFIR(const char* pInstName, void*pRef){
    mRef = pRef;
    mInstName = pInstName;
    mPortCount = 33;
    mInputPortCount = 17;

    m_SysCFIR = new SysCFIR(pInstName);
m_SysCFIR->CLK(sysCsig_CLK);
m_SysCFIR->reset(sysCsig_reset); <- Add it
m_SysCFIR->output(sysCsig_output);
}
```

- 11. Build the solution again to generate the PSpice-DMI dll.
- 12. Open Model Editor from PSpice Accessories, click File→ Open and load FIR.lib from <a href="https://www.click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.com/click.co
- 13. Click File  $\rightarrow$  Export to Part Library and click OK.

Cre	ate Parts for Library	X
	Enter Input Model Library:	
	:e\Circuits\FIR_Filter\Complete\DMI_Code\FIR\lib\FIR.lib	Browse
	Enter Output Part Library:	
	e\Circuits\FIR_Filter\Complete\DMI_Code\FIR\lib\FIR.olb	Browse
	OK Cancel Help	

**Flow**C

Application Note

- 14. Copy FIR.lib and FIR.olb files and paste them in <directory>\Circuits\FIR\_Filter\To\_be\_completed\Circuit\Library
- 15. Launch OrCAD Capture and open the project located in <a href="https://circuits/FIR\_Filter/To\_be\_Completed/Circuit"></a> <a href="https://circuits/FIR\_Filter/To\_be\_Completed/Circuit">https://circuits/FIR\_Filter/To\_be\_Completed/Circuit</a>

OPETIME = 1m	<ul> <li>U_CL</li> </ul>	K		COLK 1					100 1 1 1
		∽		OLK		INIO			<del>2</del>
DELAT -					· · · -	TNU .			<del>21</del>
STARTVAL = 0					· · · -	1012			82
OPPVAL = 1						111/2			<u>8</u>
						ING .			<u>2</u>
						IN4			200
						IND .			8
						100			8
· · · · · · · · · · · ·						101/			<u>28</u>
@FSpice:		and minute				INO .			03
_ TNO TN1 TN2 TN2 TN4	TNE TNE	5 TN7 TN8	TNO		· · · -	INITO .			810
+ IN10 IN11 IN12 IN3 IN1	TN14 TN	115	103		· · · <u></u>	IN10			812
+ IO STM IO LEVEL=0					· · · -	IN112			612
+ 0 0000						1012			813
+REPEAT FOREVER						INTS .			N14
+ +2ms ABCD						1014			015
+ +2ms DABC						INTO			
+ +2ms CDAB									
+ +2ms BCDA									
+ +2m5 ADCD									
T DRUNDFERI									

16. Click on Add Library, select FIR.olb and place the FIR component:

OFFTIME = 1m         U_CLK         1           ONTIME = 1m         UZ_T																																	
ONTIME = 1m.         DX_m         CLK         I         CLK         OUTPUT_0         1s         OUT           STARTVAL = 0         IN1 3         INPUT_0         OUTPUT_1         20         02           OPPVAL = 1         IN2 4         INPUT_1         OUTPUT_2         21         03           IN3 5         INPUT_3         OUTPUT_1         22         04           IN3 5         INPUT_3         0UTPUT_1         22         04           IN4 6         INPUT_3         OUTPUT_1         22         05           IN5 7         INPUT_6         OUTPUT_7         22         06           IN5 7         INPUT_6         OUTPUT_7         23         05           IN5 7         INPUT_6         OUTPUT_7         26         08           U D\$TMI 1812 IN3 IN4 IN5 IN6 IN7 IN8 IN9         IN1 INPUT_7         OUTPUT_10         27         09           + 10 SIN1 IN12 IN13 IN14 IN15         IN11 IN12 IN13 IN14 IN15         IN11 IN12 IN1 IN12 IN13 IN14 IN15         IN12 IN1         INPUT_10         29 OT12           + 0 6000         EXEMPT FOREVER         IN12 IN1         INPUT_11         OUTPUT_113         32		OFF	TIM	E) =	1m	Ľ,	Ū	c	Lκ							0					٠.	Ľ,	Ú1							<u>.</u>			
STARTVAL = 0     INT 3     INPUT 0     OUTPUT 1     20 02       OPPVAL = 1     INT 3     INPUT 2     OUTPUT 2     21 03       N3 5     INPUT 2     OUTPUT 3     22 04       IN4 6     INPUT 3     OUTPUT 3     22 04       IN4 7     INPUT 4     OUTPUT 5     22 04       IN4 7     INPUT 4     OUTPUT 5     22 04       IN4 8     INPUT 4     OUTPUT 5     22 06       IN7 9     INPUT 6     OUTPUT 6     22 07       IN8 10     INPUT 6     OUTPUT 7     22 07       IN8 10     INPUT 7     OUTPUT 8     27 09       U DSTMI STIM(16,4444) \$6 DEWR \$6 DEWR     IN9 11     INPUT 7     OUTPUT 8       + IN10 IN1 IN12 IN3 IN4 HIS IN6 IN7 INB IN9     IN10 12     INPUT 9     OUTPUT 10       + IN10 IN1 IN12 IN3 IN14 HIS IN6 IN7 INB IN9     IN10 12     INPUT 10     OUTPUT 10       + IN10 IN1 IN12 IN3 IN14 HIS IN6 IN7 INB IN9     IN10 12     INPUT 10     OUTPUT 11     30 012       + IN10 IN1 IN12 IN3 IN14 IN15     IN11 13     IN11 11     IN12 IT1     OUTPUT 11     30 012       + AD SCD     IN12 IT4     INPUT 13     OUTPUT 13     32 014       + 42ms DABC     IN15 IT7     INPUT 14     OUTPUT 15       + 42ms DABC     IN15 IT7     INPUT 16 <th></th> <th>BEI</th> <th></th> <th>1</th> <th>lm.</th> <th></th> <th>α</th> <th>K</th> <th>U</th> <th>&gt;-</th> <th>_</th> <th></th> <th></th> <th></th> <th> G</th> <th>N.</th> <th> </th> <th></th> <th>IN</th> <th><u> </u></th> <th></th> <th>-</th> <th>CLK</th> <th></th> <th>- (</th> <th>DUT</th> <th>ΡU</th> <th>то</th> <th>H</th> <th>9</th> <th>- 64</th> <th></th> <th></th>		BEI		1	lm.		α	K	U	>-	_				 G	N.	 		IN	<u> </u>		-	CLK		- (	DUT	ΡU	то	H	9	- 64		
SIGKIVAL = 0 OPPVAL = 1     IN2 4 IN2 4     INPUT_1 INPUT_2     OUTPUT_2 OUTPUT_3     INPUT_2 22 03 INPUT_3       IN2 4     INPUT_3     OUTPUT_4     OUTPUT_4     INPUT_3     INPUT_3       IN5 7     INPUT_4     OUTPUT_4     INPUT_6     INPUT_6     INPUT_6       IN5 8     INPUT_6     OUTPUT_7     INPUT_6     INPUT_7     INPUT_6       U D37MI STIM(16,4444) \$6 DPWR \$6 DGND     INS 11     INPUT_7     OUTPUT_7     INPUT_7       U D37MI STIM(16,4444) \$6 DPWR \$6 DGND     INS 11     INPUT_7     OUTPUT_7     INPUT_7       + INO INI INZ IN3 IN4 IN5 ING INF INF INF     INT0     INT0 12     INPUT_9     OUTPUT_10       + SO 300     INT1 13     INIA 118     INIA 118     INPUT_11     OUTPUT_11       + 100 STM IO LEVEL=0     INI1 131     INIA 118     INPUT_12     OUTPUT_11       + 0 6000     IN12 14     INPUT_13     OUTPUT_12     INT0 112       + 20ms DASC     IN15 17     INPUT_14     OUTPUT_15       + 20ms CDAB     IN15 17     INPUT_16     INPUT_16		CTA	ρ.	wi -	- <i>6</i>													_	IN	Ť	- 3	Н	INPU'	0 7	÷ (	DUT	ΈU	T-1	1	ă –	- <del>M</del>		
OFFVALET         INS         INPUT         OUTPUT         INPUT         INPUT <thinput< th="">         INPUT         INPUT         <t< th=""><th></th><th>000</th><th></th><th><u>م</u></th><th>74</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>_</th><th>IN</th><th><del>;</del> -</th><th>- 4</th><th></th><th>INPU'</th><th>r 1 -</th><th>÷ (</th><th>DUT</th><th>ΈU</th><th>т_2</th><th>-5</th><th>Ť</th><th>62</th><th></th><th></th></t<></thinput<>		000		<u>م</u>	74													_	IN	<del>;</del> -	- 4		INPU'	r 1 -	÷ (	DUT	ΈU	т_2	-5	Ť	62		
IN4 6         INPUT 3         OUTPUT 4         23         05           IN5 7         INPUT 4         OUTPUT 5         24         06           IN5 7         INPUT 6         OUTPUT 6         24         06           IN7 9         INPUT 6         OUTPUT 7         25         07           IN7 9         INPUT 7         OUTPUT 7         27         09           IN8 10         INPUT 7         OUTPUT 7         27         09           IN9 11         IN12 IN3 IN4 IN5 INF INF INF INF         IN1 13         IN14 IN15         INPUT 7         00TPUT 10         28         010           + IN10 IN1 IN12 IN3 IN14 IN15 INF INF INF         IN1 13         IN14 IN15         IN11 13         IN14 IN15         011         00TPUT 10         28         010           + IN10 IN1 IN12 IN3 IN14 IN15 INF INF         IN11 13         IN14 IN15         IN11 13         IN14 IN15         0111         00TPUT 11         00012         0114           + 0 0000         IN12 I4         INPUT 11         OUTPUT 11         00TPUT 11         30         012           + 42ms DABC         IN15 17         INPUT 14         OUTPUT 14         33         015           + 42ms DABA         IN15 17         INPUT 16         INPUT 1		OFF		Ξ.														_	IN	2	5	-	INPU'	Ē2	1.	DUT	ΈU	т_3	2	<u>-</u>	- 64		•
																		_	TN.	ă –			INPU'	Γ3	1.	DUT	ΡU	T_4	2	3	05		
NR         NPUT         OUTPUT         6         25         67           WP3pice:         INPUT         OUTPUT         7         26         03           W DSTMI STIM(16,4444) \$G DFWR \$G DGND         INPUT         NIPUT         0UTPUT         27         09           + INO INI INIZ IN3 IN4 INS ING INT INB IN9         INI         INPUT         0UTPUT         28         011           + INO INI INIZ IN3 IN4 INS ING INT INB IN9         INI 1         INI INPUT         0UTPUT         10         29         011           + INO INI INIZ IN3 IN4 INS ING INT INB IN9         INI 1         INI INPUT         0UTPUT         10         001PUT         10         29         011           + O 0000         INI 1         INI 2         INI 1         10         OUTPUT         13         0012           + 22ms DABC         INI 1         INI PUT         13         0015         11         11         11         11         11         11         11         11         11         11         11         11         11         0017         11         32         014           + 22ms DABC         IN15         IN15         IN15         IN115         17         111         110         111         11																		_	ĪN	5	-7	-	INPU'	<u>-</u> 4	1.	DUT	ΡU	T_5	2	4	08		
INPUT_6         OUTPUT_7         28         68           UDSTMIL STIM(16,4444)         SG DEWR SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEWR         SG DEW         SG DEWE         SG																		_	ĪN	8	8	-	INPU'	r_5	1.	DUT	PU	T_6	2	5	07		
Bypice:         INPUT_7         OUTPUT_8         Z7         Og           U_DSTMLSTIM(16,4444)         SG_DEND         INPUT_8         OUTPUT_9         Z28         OTO           + IN0 IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8 IN9         IN10         IN1         INPUT_9         OUTPUT_10         228         OTO           + IN0 IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8         IN9         IN10         IN10         INPUT_10         OUTPUT_11         2011           + IN0 IN1 IN2 IN3 IN4 IN5         IN11         IN10         IN11         INPUT_11         300 OT2           + 0 STM 00_LEVEL0         IN12         IN1         INPUT_11         OUTPUT_11         300 OT2           + 0 6000         IN13         INPUT_10         OUTPUT_12         310 OT3         OT4           + 42ms ABCD         IN14         IN1         INPUT_14         OUTPUT_15         33         OT6           + 42ms CDAB         IN15         IN15         IN15         IN15         IN15         IN15																		_	ĪŇ	7	9		INPU'	<u>6</u>	÷ (	DUT	PU	T_7	2	8	08		
U DSTMI STIM(16,444) 66 DEWR 66 DEWD     INPUT 8     OUTPUT 9     28     O10       + INO INI INZ IN3 IN4 IN5 IN6 IN7 IN8 IN9     IN10 12     INPUT 9     OUTPUT 10     29     O11       + IN10 IN1 IN12 IN13 IN14 IN5 IN6 IN7 IN8 IN9     IN10 12     INPUT 10     OUTPUT 10     29     O11       + IN10 IN1 IN12 IN13 IN14 IN15     IN11 13     IN12 IN1     IN10 12     INPUT 10     OUTPUT 11     30     O12       + 0 0000     IN12 I4     INPUT 12     IN11 13     IN11 01     INPUT 13     OUTPUT 13     32     O14       + 20m SABCD     IN15 17     IN15 17     INPUT 14     OUTPUT 14     33     O15       + 42ms CDAB     X EIR     X EIR     X EIR     X EIR	<b>#PSpice</b>	- 1																_	IN	8	10		INPU'	_7	19	DUT	PU	T_8	2	7	09		
+ TNO INI INZ ING	U DSTM1	STI	6(16	,44	(44)	\$	GI	)PW	R	\$G	DGI	ŃD						_	IN	9	-11		INPU	_8	1	DUT	PU	T_9	2	8	01/	0	
+ IN10 IN11 IN12 IN13 IN14 IN15     IN11 13     INPUT_10     OUPUT_11     30     012       + IO_STM IO_LEVEL=0     IN12 14     INPUT_11     OUTPUT_12     31     013       + 0 0000     IN13 15     IN12 14     INPUT_12     017UT_12     31     013       + 82EPEAT FOREVER     IN14 16     INPUT_13     OUTPUT_14     33     015       + 42ms DABC     IN15 17     INPUT_15     INPUT_15       + 42ms CDAB     Y_EIR     Y_EIR	+ INO I	N1 11	12 I	N3	ING	Ì	N5	IN	6	IN7	п	NΒ	INS	11				_	IN	10	12		INPU	_9	0	UTF	UT	_10	2	9	01	1	
+ 10 STM 10_LEVEL=0 + 0 0000 +REPEAT FOREVER + 22ms DABC + 42ms CDAB + 42ms CDAB - + 2ms CDAB 	+ IN10	IN11	IN1	2 1	N13	1	N14	I	N1	5									IN	11	-13	5	INPU	_10	.0	UTH	10	-11	3	0	01	2	
+ 0000 0000 0000 0000 12 00000 12 00000 12 00000 12 0000 12 0000 12 00000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 0000 12 00000 12 0000 12 000000 12 00000 12 00000 12 00000 12 00000 12 00	+ 10_31	M 10	LEV	ET=	=0														IN	12	- 14	•	INPU	-11	20		201	-12	3	1	01	3	
INI4     16       INPUT_14     001P01_14       ABCD     INI4       + +2ms     DABC       + +2ms     CDAB       + +2ms     BCDA       X     FIR	+ 0 000	FOR	-																IN	13	15	)	INPU	-14			201	-13	- 3.	2	01	4	
+ + + 2ms DABC	+ +2ms	ABCD																_	IN	14	10	5	INPU	-13	22		101	-12	3	3	01	5	
+ +2ms CDAB + +2ms BCDA X. FIR	+ +2ms	DABC																_	IN	15	-17		INPU	-42			0	-19					
+ +2ms_BCDAX_FIR	+ +2ms	CDAB																				L	INFO	_10					<b>.</b> .				
	+ +2ms	BCDA																					X EIR										
+ +2ms ABCD ^	+ +2ms	ABCD																					÷										
+ LNURLPLAI	+ ENDRE	PLAT																															

- 17. Create a new Simulation Profile called Trans.
- 18. Complete the new Simulation Settings with this values:

neral Analysis Configuration nalysis type: "ime Domain (Transient) ptions: General Settings Monte Carlo/Worst Case Parametric Sweep	n Files Options Data Collection Prob Run to time: 100m Start saving data after: 0 Transient options Maximum step size:	e Window seconds (TSTOP) seconds
Temperature (Sweep) Save Bias Point Load Bias Point Save Check Points Restart Simulation	Skip the initial transient bias po Run in resume mode	int calculation (SKIPBP) Output File Options

Flow(

Application Note

19. Click on Configuration Files and add to Design the library you have just created:

Category:	Details Eilensame:	
Stimulus Library		Browse
Include	Configured Files	
<ul> <li>Update Index</li> </ul>	<ul> <li>\library\capture\fir.lib</li> <li>nom.lib</li> </ul>	Add as Global
		Add to Design
		Add to Profile
		Edit
		Change
	<ul> <li>✓ III ►</li> <li>Library Path</li> </ul>	
	"C:\Cadence\SPB_17.2\tools\PSpice\Library"	Browse

### 20. Copy the DLL from

<directory>/Circuits/FIR\_Filter/To\_be\_completed/DMI\_Code/FIR/code/x64/Release

And copy it in

<directory>/Circuits/FIR\_Filter/To\_be\_completed/Circuit/FIR\_Filter-PSpiceFiles/SCHEMATIC1/Trans

21. Simulate.

#### You will get this error:



To solve it, RMB on the FIR component and click on Edit PSpice Model.

22. Make next modifications and do not forget to write down + at the beginning of the new line:



- 23. Close and click OK to save.
- 24. Simulate again and analyze the results (plot each input and each output).

**NOTE:** Notice that the input signal is coming from text format placed on the Schematic.



## 3.3 Noise Filter using a MATLAB Block

This module explains a simple example of Analog Behavioural Model imported into PSpice as a DMI Model. This module takes the example of a MATLAB averaging filter to demonstrate this.

The details of generating the code may be found at



http://www.mathworks.com/help/coder/examples/averaging-filter.html?prodcode=ME&language=en

This Lab demonstrates:

- An example of MATLAB generated code imported to PSpice as a DMI model.
- Generation of a template code for an Analog behavioural model and its use in a PSpice simulation.

### Steps:

- 1. Launch Model Editor
- 2. Select Menu Item Model → DMI Template Code Generator
- 3. Enter the data as follows:

se this dialo	g-box to auto-generate DMI template	code for the following PSpice-DMI models: Analog, Digital, and
ystemC. The ecommend	e dialog-box also imports the Verilog-/	A Compact Device models using ADMS.
1. Testth	ne model code stand-alone by building	g an exe. di i i i a Marual Studio ta incast ma dal cada
3. Use th	e generated PSpice library (.lib file) to	o create a schematic symbol. The generated symbol can be place
in the	schematic for PSpice simulation.	
Part Details		
	Part Name	NoiseFilter
	PartType	Analog
	Fait Type	Analog
Ferminals –		
	Model Type	Function-Dependent Voltage Source
	Terminal Entry	
Parameters		
	Global Parameters	
	Device Parameters	
	Model Parameters	
Outout		
Julput		<b></b>
	DLL File Name	NoiseFilter.dll
	Log File Name	NoiseFilter.log
	DLL Location	D:\PSpice\Application Notes\Flov Browse
		OK Cancel Help

### Click on Browse to save the code in



<directory>\Circuits\MATLAB\_Block\_Simulation\To\_be\_combpleted\DMI\_Code

Do **NOT** click on OK.

4. Click on Terminal Entry – for the selected predefined model type, the number of terminals is fixed at 4.

Terminal Name	Terminal Description
Input1	Noise Input 1
Input2	Noise Input 2
Output1	Output 1
Output2	Output 2

- 5. Click OK and generate the code.
- 6. Verify that the code and the lib folder are created correctly:

<directory>\Circuits\MATLAB\_Block\_Simulation\To\_be\_completed\DMI\_Code\NoiseF ilter\code

**NOTE**: the DLL will not be created until you compile the code in Visual Studio.

NoiseFilter.log
 NoiseFilter.vcxproj
 NoiseFilter\_user.cpp
 pspEngFunc.cpp
 pspEngFunc.h
 PSpiceBase.h
 PSpiceCoMIApiDefs.h
 PSpiceDigApiDefs.h
 pspNoiseFilter.cpp
 pspNoiseFilter.h
 StdAfx.cpp
 StdAfx.h

 $<\!\!directory>\!\!\backslash Circuits \\ MATLAB\_Block\_Simulation \\ To\_be\_completed \\ DMI\_Code \\ NoiseFilter \\ lib$ 



- J input.json
- NoiseFilter.lib
- 7. Launch Visual Studio Community 2013 and open the file NoiseFilter.vcxproj

FILE	EDIT VIEW DEBUG	TEAM TOOLS	TEST	ANALYZE WINDOW	HELP	
	New	•	ch	• Ů •   •   •		- 🏓 🚽
	Open	•	1	Project/Solution	Ν	Ctrl+Shift+O
	Close		٩	Web Site	63	Shift+Alt+O
	Close Solution			Open from Source Control		
	Save Selected Items	Ctrl+S	ta	Team Project		
	Save Selected Items As		2	File		Ctrl+O
ъ <sup>6</sup>	Save All	Ctrl+Shift+S		Convert		
	Export Template		1		_	
	Source Control	•				
	Page Setup					

The project is located in

 $<\!\!directory\!\!>\!\!Circuits\MATLAB\_Block\_Simulation\To\_be\_completed\DMI\_Code\NoiseFilter\code$ 



8. Change the configuration on the top as follows:





9. Select Menu Item Build/Build Solution and verify that the Build completes successfully.



- 10. The PSpice-DMI template code is ready. Now, the model behaviour for the exported averaging filter (using C-Coder in MATLAB) needs to be inserted in the code.
- 11. Open NoiseFilter\_user.cpp for editing:

Under the line: #include "pspNoiseFilter.h"

Add the text:

```
extern "C" {
#include "../../averaging_filter/averaging_filter.h"
}
```

This includes the MATLAB generated header file so that its Averaging filter function can be accessed.

12. Edit the load function as follows:

Under the line:

double gain = 0.0;

Add the text:

```
// User Code
    if (pMode != MDTRAN) {
        for (int i = 0; i < 16 + MSTVCT; i++) {
            sv.x[i] = xVal;
        }
     }
     sv.y[0] = yVal = averaging_filter(xVal, sv.x);
//</pre>
```

This code will update the state vector with the latest input value, and call the MATLAB averaging\_filter function to compute the gain.


13. Finally, add the MATLAB averaging\_filter file to the project. In the solution Explorer, RMB on NoiseFilter and select Menu Item Add – Existing Item

_								
4	] No	oiseFilte		0.114				
Þ	T 🗈	Externa		Build				
Þ	++	NoiseFi		Rebuild				
⊳	++	pspEng		Clean				
⊳	Þ	pspEng		View	,			
⊳	B	PSpicel		A				
⊳	B	PSpice		Analyze	,			
Þ	B	PSpice		Project Only	•			
⊳	Ь	PSpicel		Scope to This				
₽	++	pspNoi	Ē	New Solution Explorer View				
⊳	Б	pspNoi						
	++	StdAfx.		Profile Guided Optimization	•			
Þ	Þ	StdAfx.		Build Dependencies	•			
				Add	•	Ĝ	New Item	Ctrl+Shift+A
			₽.	Class Wizard	Ctrl+Shift+X	10	Existing Item	Shift+Alt+A
			苗	Manage NuGet Packages		-	New Filter	13
			Ф	Set as StartUp Project			References	
				Debug				

14. Browse to ../../averaging\_filter and select the averaging\_filter.c

rganisieren 🔹 Neuer C	Drdner		•	
🕹 Musik 🔦	Name	Änderungsdatum	Тур	Größe
ConeDrive	📙 interface	20.06.2016 09:49	Dateiordner	
S Zuletzt besucht	averaging_filter.c	21.03.2016 12:59	C Source	
Diblication	hi averaging_filter.h	21.03.2016 12:59	C/C++ Header	
	averaging_filter.lib	21.03.2016 12:59	LIB-Datei	
Bilder	averaging_filter_initialize.c	21.03.2016 12:59	C Source	
Jokumente =	h averaging_filter_initialize.h	21.03.2016 12:59	C/C++ Header	
	averaging_filter_ref.rsp	21.03.2016 12:59	RSP-Datei	1
S videos	averaging_filter_rtw.bat	21.03.2016 12:59	Windows-Batchda	
Computer	averaging_filter_rtw.mk	21.03.2016 12:59	Makefile	1
Windows7 OS (C)	averaging_filter_terminate.c	21.03.2016 12:59	C Source	
Data2 (D·)	前 averaging_filter_terminate.h	21.03.2016 12:59	C/C++ Header	
✓ Suture (0.) ✓ support (\\dc01) ( +	<ul> <li>bi averaging filter types.h</li> <li>✓</li> </ul>	21.03.2016 12:59 III	C/C++ Header	

- 15. Build Solution from the Menu Item Build/Build Solution and verify it works.
- 16. Open the location <a href="https://www.communicationscore-completed/DMI\_Code/NoiseFilter/code/x64/Release">https://www.communication/To\_be\_completed/DMI\_Code/NoiseFilter/code/x64/Release</a> and verify that the DLL and the PDB have been generated:



FlowC

Application Note

- 17. Launch Model Editor from PSpice Accessories.
- 18. Click Done
- 19. Click File→Open and select the library generated with DMI Template Code Generator. This is located in <directory>/circuits/MATLAB\_Block\_Simulation/To\_be\_completed/DMI\_Code/NoiseFi Iter/lib

📙 « Comple	te ► DMI_Code ► NoiseFilte	er ▶ lib 👻 🍫 lib
en 👻 Neu	r Ordner	
¢	Name	Änder
orive zt besucht	NoiseFilter.lib	20.06.
heken		

20. Select the component you have just loaded and click on File $\rightarrow$ Export to Part Library:



21. Click OK. A window indicating that everything worked properly should pop up:



Application Note



22. Copy NoiseFilter.lib and NoiseFilter.olb from

<directory>/circuits/MATLAB\_Block\_Simulation/To\_be\_completed/DMI\_Code/NoiseFi Iter/lib

and paste them in

<directory>/Circuits/MATLAB\_Block\_Simulation/To\_be\_completed/Circuit/Library/Cap ture

23. Launch OrCAD Capture and open the MatlabBlock design located in <directory>/Circuits/MATLAB\_Block\_Simulation/To\_be\_completed/Circuit



24. Click on Add Library, select the NoiseFilter.olb symbol and place it in the schematic:

<directory>/circuits/MATLAB\_Block\_Simulation/To\_be\_completed/Circuit/Library/Cap ture



25. Create a new Simulation Profile and name it Trans.

Name:	Creat
Trans	Creat
Inherit From:	Cance
none	▼

26. Complete the values like in the image:

eneral Analysis Configuration F	iles Options Data Collection Probe	Window		
Time Domain (Transient)	Run to time:     6       Start saving data after:     0       Transient options	seconds (TSTOP) seconds		
Parametric Sweep         Temperature (Sweep)         Save Bias Point         Load Bias Point         Save Check Points         Restart Simulation	Maximum step size: 1m seconds Skip the initial transient bias point calculation (SKIPBP) Run in resume mode Output File Op			

27. Without closing the Settings select Configurations Files and add to Design the NoiseFilter.lib:

Catego	Anaiysis	Details	
Stimulu	IS	Filename:	Browse
Include	•	Configured Files	
Vpdate Index		\/ibrary\capture\noisefilter.lib     nom.lib	Add as Global
			Add to Design
			Add to Profile
			Edit
			Change
		C:\Cadence\SPB_172\tools\PSpice\Library"	Browse

**FlowCAD** 

Application Note

- 28. Click OK.
- 29. Before simulating, place the created NoiseFilter.dll and its .pdb in the Simulation Profile Folder.

**DLL Location:** 

<directory>/Circuits/MATLAB\_Block\_Simulation/To\_be\_completed/DMI\_Code/NoiseF
ilter/code/x64/Release

FlowC

Application Note

Simulation Profile Location:

<directory>/Circuits/MATLAB\_Block\_Simulation/To\_be\_completed/Circuit/MatlabBloc k-PSpiceFiles/SCHEMATIC1/Trans

- 30. Simulate and verify it works:
- 31. Analyze the results. Click on PSpice  $\rightarrow$  Markers  $\rightarrow$  Voltage Level.

PSpice	Accessories	Options	W	indow	Help				
<mark>⊠ N</mark> ew <u>E</u> dit S	Simulation Pro Simulation Pro	ofile file				Ē.	JQ •	G. (	?)
© <u>R</u> un ⊻iew Vie <u>w</u>	Simulation Re Output File	F11 sults F12	2	1	Ĩ				
<u>C</u> reat V <u>i</u> ew	e Netlist Netlist				3				
A <u>d</u> va Mark	nced Analysis ers		•	<b>K</b> Vol	tage Level			N	
Bias	Points		•	∕& Vol	tage <u>D</u> iffer rent Into Pi	rentia in	I	6	

32. Place the markers like in the image:



33. Open Probe Window:



# **3.4 Capacitor behaviour analysis defined using VerilogA-ADMS**

This module explains the import of a VerilogA file and its translation to a DMI model. VerilogA import in PSpice is supported using ADMS parser – this is primarily useful for importing VerilogA compact models.

This Lab will demonstrate:

- An example of a capacitor model written in VerilogA, using 2 parameters to specify the capacitance value.
- Import of a VerilogA file using Model Editor and translation into PSpice-DMI model.
- Sample simulation and comparison of the results with a regular capacitor simulation.

#### Steps:

1. Launch Model Editor from PSpice Accessories:



- 2. Select Model  $\rightarrow$  DMI Template Code Generator.
- 3. Enter Part Name as cap

**NOTE:** Part Name should match the module name specified in the VerilogA file.

- 4. In the DMI Template code Generator UI, select Part Type as VerilogA-ADMS
- 5. In Verilog-A File field, enter the path to the cap.va file located in the VerilogA\_component folder:

<directory>/Circuits/VerilogA\_Capacitor/To\_be\_completed/VerilogA\_Component

**NOTE:** The XML Folder is automatically selected by the tool. You do not have to browse anything.

6. In the Output, select only the DLL Location to:

<directory>/Circuits/VerilogA\_Capacitor/To\_be\_completed/DMI\_Code

DMI Temp	late Code Generator	3
se this dialo	og-box to auto-generate DMI templa	ate code for the following PSpice-DMI models: Analog, Digital, and
ystemC. The	e dialog-box also imports the Verilo	g-A Compact Device models using ADMS.
ecommend	ded steps:	
1. Testt	he model code stand-alone by build	ding an exe.
2. Creat	e the PSpice-DMI adapter code, an	d edit it in Visual Studio to insert model code.
3. Use ti	he generated PSpice library (.lib file	e) to create a schematic symbol. The generated symbol can be place
in the	schematic for PSpice simulation.	
Part Details		
	PartName	cap
	PartTurne	
	Fait type	VeniogA-ADMS
nput —		
	Verilog-A File	D:\PSpice\Application Notes\Flov Browse
	XML Folder	C:\Cadence\SPB_17.2\tools\psp Browse
ouput	DI L El L Nerre	
	DLL File Name	cap.dll
	Log File Name	cap.log
	DLL Location	D:\PSpice\Application Notes\Flov Browse
		OK Cancel Help
PSpice DM	I Template Generator	
Spice DIM	i rempiate Generator	

The cap.va is a VerilogA model for a capacitor which uses 2 parameters to define the capacitor values: C1 and C2.

`include "discipline.h"
module cap(p,n);
inout p,n;
electrical p,n;
parameter real c1=0 from [0:inf);



parameter real c2=0 from [0:inf); analog I(p,n) <+ ddt**((c1+2\*c2**)\*V(p,n)); endmodule

7. Click OK. The DMI model is auto-generated from the VerilogA file and a log file is generated listing the translation steps.

ap.log - Editor	X
Datei Bearbeiten Format Ansicht ?	
Datei Bearbeiten Format Ansicht ? Datei Bearbeiten Format Ansicht ? Reading json Json readLogging Model Creation reading Model Name and Type reading global and instance params for digital and analog Copying common files Replacing the Model Name in C:\Cadence\SPB_17.2\tools\pspice\tclscripts\pspModelCree Verilog-A Path : D:/PSpice/Application Notes/FlowCAD_AN_Device_Modeling_Inter Name of Verilog-A file : cap.va Verilog-A Location : D:/PSpice/Application Notes/FlowCAD_AN_Device_Modeling_Inter XMLs Location : C:\Cadence(SPB_17.2/tools/pspice/api/adms/xmls/ Generating Files C:\Cadence(SPB_17.2/tools/pspice/api/adms/bin/admsXml.exe -ID:/PSpice/Application N C:\Cadence(SPB_17.2/tools/pspice/api/adms/bin/admsXml.exe -ID:/PSpice/Application	eate\Verili face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/Circi face/
D:\PSpice\Application Notes\FlowCAD_AN_Device_Modeling_Interface\Circuits\VerilogA	_Capacitor
cl /D "_WINDLL" /D "_MBCS" /D "_CRT_SECURE_NO_WARNINGS" /O2 /GL /FD /EHsc /	/MT /W3 /n +
	►

8. The generated code is automatically compiled using nmake. If there are no build errors, the dll and the lib files are generated, and can be directly used in a PSpice Simulation.

<directory>/Circuits/VerilogA\_Capacitor/To\_be\_completed/DMI\_Code/cap/lib



• Dll file created in the folder code:



- .adms.implicit.xml .cap.va.adms .interface.xml analogfunction.cpp i analogfunction.h 📋 сар.срр cap.dll cap.dll.manifest cap.exp 🛅 cap.h 🔳 cap.lib cap.log 🗂 cap.vcxproj CapacId.cpp 🛗 capdefs.h Capguesstopology.cpp C capload.cpp Capnoise.cpp
- Lib file created in the folder lib:

.subckt	t cap_su	bpnF	ARAMS:	c1=0	c2=0
Y1 p n	CMI cap	.dll ca	p_mod		
.model	cap_mod	CMI ca	p c1={c	:1} c2	2={c2}
.ends					

- 9. Copy the cap.lib and paste it in <directory>\circuits\VerilogA\_Capacitor\To\_be\_completed\Circuit\Library\Capture in order to generate the symbol (olb) to be used in Capture.
- 10. Copy the DLL and paste it in <a href="https://circuits/VerilogA\_Capacitor/To\_be\_completed/Circuit/Library/DLL">https://circuits/VerilogA\_Capacitor/To\_be\_completed/Circuit/Library/DLL</a>
- 11. Open Model Editor from PSpice Accessories
- 12. Click on File  $\rightarrow$  Open and look for cap.lib in

<directory>\circuits\VerilogA\_Capacitor\To\_be\_completed\Circuit\Library\Capture

13. Click on File  $\rightarrow$  Export to part library

,	
Enter Input Model Library:	
√erilogA_Capacitor\Complete\DMI_Code\cap\lib\cap.lib	Browse
Enter Output Part Library:	<b>_</b>
Enter Output Part Library: 'erilogA_Capacitor\Complete\DMI_Code\cap\ib\cap.olb	Browse
Enter Output Part Library: 'erilogA_Capacitor\Complete\DMI_Code\cap\lib\cap.olb	Browse

14. Now open in OrCAD Capture the example design where you are going to evaluate a capacitor from the default Cadence library and the capacitor you have just defined. Click on File→Open→Project and select the project located in

<directory>\Circuits\VerilogA\_Capacitor\To\_be\_completed\Circuit\Example.opj



16. Make double click on Page1 from the schematic Cap:



17. Click on Place Part and select C from the library analog.olb:



FlowC/

Application Note



18. Place it in the schematic and change the value to 5n



19. Create a new Simulation Profile and call it Trans:



25. Complete the Simulation profile with these values:

analysis type: Time Domain (Transient) 💌 Options:	Run to time: 60u seconds (TSTOP) Start saving data after: 0 seconds
General Settings Monte Carlo/Worst Case Parametric Sweep Temperature (Sweep) Save Bias Point Load Bias Point Save Check Points Restart Simulation	I rensient options         Maximum step size:         Skip the initial transient bias point calculation (SKIPBP)         Run in resume mode         Output File Options

- 26. Click OK.
- 27. Run the simulation clicking on Play.
- 28. In OrCAD Capture click on PSpice → Markers → Voltage Level and place the markers as in the image:







22. Come back to OrCAD Capture, select the DMICap Schematic and with RMB make it Root. Now you are going to simulate your Verilog-A Cap.



NOTE: If you have not saved, it will ask you automatically. Click on Save.

23. Click on Add Library and select the .OLB you have created

<directory>/Circuits/VerilogA\_Capacitor/To\_be\_completed/Circuit/Library/Capture





and place the capacitor changing the values of C1 and C2 to 1n. You will find these properties making double click on the DMICAP.

	1		
· · · · · · · · · · · · · · · · · · ·	N · · · · P		
C	AP_SUB		
V1 = 0 V1 V2 = 5 V1			
TD = 1n V.V TR = 1u			
TF = 1u PW = 10u			
PER = 20u			
	Α		
	DMICap : PAGE1 : U1		
BiasValue Power	0W		
C1	1n		
C2	1n		
Color	Default		
Designator			
Graphic	CAP_SUB.Normal		
Designator			
Implementation	CAP_SUB		
Implementation Path			
Implementation Type	PSpice Model		
Location X-Coordinate	410		
Location Y-Coordinate	140		
Name	INS707		
Part Reference	U1		
PCB Footprint			
Power Pins Visible			
Primitive	DEFAULT		
PSpiceTemplate	X^@REFDES %P %N @MO		
Reference	U1		
Source Library	C:\TEMPORAL\TEST\C		
Source Package	CAP_SUB		
Source Part	CAP_SUB.Normal		
Value	CAP SUB		

24. Create a new Simulation profile called Trans, with the same values than before and click Configuration Files and Libraries:

Analysia	Configuration Files Options Data Collection Broke Window	13
al Allalysis	Details	
egory:	Filename:	
nulus rarv		Browse
lude	Configured Files	
✓ Update Index	<ul> <li>\library\capture\cap.lib</li> <li>nom.lib</li> </ul>	Add as Global
		Add to Design
		Add to Profile
		Edit
		Change
	۰ III ۲	
	Library Path	
	"C:\Cadence\SPB_17.2\tools\PSpice\Library"	Browse

25. Add to Design the .lib generated you located in

<directory>/Circuits/VerilogA\_Capacitor/To\_be\_completed/Circuit/Library/Capture

26. Before simulating, you have to place the .DLL in the Simulation Profile Folder you have just created. Place it in:

<directory>\VerilogA\_Capacitor\To\_be\_completed\Circuit\ExamplePSpiceFiles\DMICap\Trans



27. Simulate and analyse the results placing the markers as done before:

Flow(



## 3.5 Debugging

This section explains how the behaviour of DMI Models can be debugged on the circuit where it has been designed. There are two options to debug DMI models. To understand both of them, you will use the project that is located in

<directory>\Circuits\Debugging\Circuit\MatlabBlock.opj

#### Option 1: Setup path to psp\_cmd.exe in Visual Studio

- 1. Open Visual Studio Community 2013
- 2. Click File $\rightarrow$ Open $\rightarrow$ Project/Solution and search for:

<directory>\Circuits\Debugging\DMI\_Code\NoiseFilter\code\noiseFilter.vcxproj

3. Ensure DEBUG profile is selected on the top:



4. Select the project with RMB and select properties:



Solution '	Noise	Filter' (1 project)		
🔺 💽 Noise	4			
🕨 🎵 Ex	÷	Build		
▷ ++ av		Rebuild		
▷ ++ N¢		Clean		
▶ ++ ps		View	•	
P ⊡ ps		Analyze	•	
D D PS		Project Only	•	
Þ 🖻 PS		Scope to This		
D 🖻 PS	Ð	New Solution Explorer View		
P ⁺+ ps ▷ ⊡ ps		Profile Guided Optimization	•	
*+ St		Build Dependencies	•	
St		Add	•	
	₽.*	Class Wizard	Ctrl+Shift+X	
	ă	Manage NuGet Packages		
	Ф	Set as StartUp Project		
		Debug	•	
		Source Control	•	
	ж	Cut	Ctrl+X	
	பி	Paste	Ctrl+V	
Frror List	×	Remove	Del	
<b>T</b> - 🛛 0	X	Rename		
Descript		Unload Project		
Courp		Rescan Solution		
	¢	Open Folder in File Explorer		
	۶	Properties	Alt+Enter	

5. A window pops up. Click on Debugging and setup three property values:

Command	Enter absolute path to psp_cmd.exe
Command Arguments	Enter absolute path to cir file which contains the C Model instance
Working Directory	Enter the absolute path to the directory which contains the cir file

For this example:

Command: %CDSROOT%\tools\bin\psp\_cmd.exe

Command Arguments: <directory>\Circuits\Debugging\Circuit\MatlabBlock-PSpiceFiles\SCHEMATIC1\Trans\Trans.cir

Working Directory: <directory>\Circuits\Debugging\Circuit\MatlabBlock-PSpiceFiles\SCHEMATIC1\Trans

Flow<mark>CAD</mark>

Application Note

nfiguration: Active(Debug)		Platform: Active(x64)     Configuration Manager		
Common Properties     Configuration Properties     General     Configuration Properties     General				
Debugging VC++ Directories	Command	C:\Cadence\SPB_17.2\tools\bin\psp_cmd.exe		
▷ C/C++	Command Arguments	D:\PSpice\Application Notes\FlowCAD_AN_Device_Modeling_Interface\Circuits\Debugging\Cir		
Þ Linker	Working Directory	D:\PSpice\Application Notes\FlowCAD_AN_Device_Modeling_Interface\Circuits\Debugging\Ci		
Manifest Tool	Attach	No		
XML Document Generator	Debugger Type	Auto		
Browse Information	Environment			
Build Events	Merge Environment	Yes		
Custom Build Step	SQL Debugging	No		
Code Analysis	Amp Default Accelerator	WARP software accelerator		

- 6. Verify that you have the same values in the next options:
  - Click on Optimization inside of C/C++

onfiguration: All Configura	tions	Platform:	Active(x64	4) 🗸	Configuration Manager.
Common Properties		Optimization		Disabled (/Od)	<u>^</u>
<ul> <li>Configuration Properties</li> </ul>		Inline Function Ex	xpansion	Default	
General	=	Enable Intrinsic F	unctions		
Debugging	-	Favor Size Or Spe	eed	Neither	
VC++ Directories		Omit Frame Poin	ters		
▲ C/C++		Enable Fiber-Safe	e Optimiz	a No	
General		Whole Program (	Optimizat	i.	
Optimization					
Preprocessor					
Code Generation					
Language					
Precompiled Header Output Files	s +	Optimization Select option for coo	de optimiz	zation; choose Custo	om to use specific optimiza
( III I					

FlowCAD Application Note

• Click on Debugging inside of Linker:

onfiguration: All Configurat	ions	Platform: Active(x6	54)  Configuration Manager.
All Options		Generate Debug Info	Yes (/DEBUG)
Command Line		Generate Program Databa	as \$(OutDir)\$(TargetName).pdb
Linker		Strip Private Symbols	
General		Generate Map File	No
Input		Map File Name	
Manifest File		Map Exports	No
Debugging		Debuggable Assembly	
System Optimization	Ш		
Windows Metadata			
Advanced All Options	Ŧ	Generate Debug Info The /DEBUG option creates d	lebugging information for the .exe file or DLL.

- 7. Click Apply and OK.
- 8. Open file NoiseFilter\_user.cpp and add breakpoints clicking on the desired line:

```
Output
            Solution Explorer
                                 NoiseFilter_user.cpp 👎 😕
NoiseFilter
         //double xVal = fp_getV(OutPort1,OutPort2);
         double xVal = fp_getV( Input1 , Input2 );
         double delta = fp_getDelta();
         double yVal = 0.0;
         double gain = 0.0;
     // User Code
         if (pMode != MDTRAN) {
             for (int i = 0; i < 16 + MSTVCT; i++) {</pre>
                 sv.x[i] = xVal;
             3
         }
         sv.y[0] = yVal = averaging_filter(xVal, sv.
      11
         fp_applyValueItem(ribr,yVal );
         fp_applyValueItem(Output1_cibr,1);
         fp_applyValueItem (Output2_cibr, -1);
         fp_applyValueItem (cibr_Output1, 1 );
         fp_applyValueItem (cibr_Output2, -1 );
         return returnVal:
```

9. Run the simulation in debug mode

DE	BUG	TEAM	TOOLS	TEST	ANALYZE	WIND
	Win	dows				•
	Grap	phics				- •
►	Star	t Debuggi	ng		F5	
⊳	Star	t Without	Debuggin	g	Ctrl+F5	~
o <sup>©</sup>	Atta	ich to Proc	ess			
	Othe	er Debug	Targets			
	Exce	eptions			Ctrl+Alt+	E
	Perf	ormance a	and Diagno	ostics	Alt+F2	
ς.	Step	o Into			F11	
¢,	Step	o Over			F10	
	Tog	gle Break	point		F9	
	New	v Breakpoi	nt			
5	Dele	ete All Bre	akpoints		Ctrl+Shift	t+F9
	Disa	able All Br	eakpoints			
	Clea	ar All Data	Tips			
	Expo	ort DataTi	ps			
	Imp	ort DataTi	ps			
	Opti	ions and S	ettings			
ş	Nois	seFilter Pro	operties			

10. Choose Yes on the following message:

No Debug	gging Information
<u> </u>	Debugging information for 'pspice.exe' cannot be found or does not match. Cannot find or open the PDB file.
	Do you want to continue debugging?
🔲 Don't	show this dialog again
	Yes No

11. Observe that simulation stops at breakpoint:

Application Note
double yVal = 0.0;
double gain = 0.0;
// User Code
if (pMode != MDTRAN) {
 for (int i = 0; i < 16 + MSTVCT; i++) {
 sv.x[i] = xVal;
 }
 sv.y[0] = yVal = averaging\_filter(xVal, sv.x);
//</pre>

Flow(

12. Click on Continue to jump from one breakpoint to another one.

D	EBUG	TEAM	TOO	LS TES
-	► Co	ntinue 🝷	Ċ -	Debug
•	E Life	ecycle Eve	ents 🝷	Thread:
			•	→ pspN

13. Use Visual Studio Watch function to see any variable value

Code	11	Insert Snippet	Ctrl+K, Ctrl+X	
(pMode != MDTR	ta	Surround With	Ctrl+K, Ctrl+S	Ŀ
<pre>sv.x[i] =</pre>	Ξ	Peek Definition	Alt+F12	
}	•	Go To Definition	F12	
v[0] = vVa] =	*	Go To Declaration	Ctrl+F12	
)[0] ).01		Find All References	Shift+F12	
	$\mathbb{Z}$	View Call Hierarchy	Ctrl+K, Ctrl+T	
annlvValueItem		Toggle Header / Code File	Ctrl+K, Ctrl+O	
applyValueItem		Breakpoint	•	
annlvValueItem	⇔	Add Watch	N	h
	⇔	Add Parallel Watch	13	- 1
Va	⇔	QuickWatch	Shift+F9	e
M N		Pin To Source		del

**NOTE:** If it is said that the project is out of date or you make many changes in the code, build the project again, copy .dll and .pdb and past them in the Simulation Profile Trans, so that you can debug properly.

FlowCAD

Application Note

Aut	os			<b>-</b> ₽	×
N	ame	e	Value	Туре	-
	0	MDTRAN	MDTRAN (3)	modeFla	a
	0	gain	0.000000000000000	double	
	0	pMode	1	int	
⊳	0	this	0x000000002860e80 {Input1=0x000000002860ea0 "N01546"	pspNois	Æ
					-

#### **Option 2: Attach Visual Studio to PSpice**

1. Open Visual Studio and load the previous project.

<directory>\Circuits\Debugging\DMI\_Code\NoiseFilter\code\NoiseFilter.vcxproj

2. Open the project in OrCAD Capture

<directory>\Circuits\Debugging\Circuit\MatlabBlock.opj

3. Open NoiseFilter\_user.cpp and place your breakpoints in the code



4. In Visual Studio, click on Tools→Attached to process:

Flow<mark>CAD</mark>

Application Note

FC-RGAN	IDIA				
	NDIA		•	Find.	
ts you sel SVSMON	ect processes on this compu EXE).	ter or a remote computer runni	ng the Microsoft Visual Stud	dio Remote	3
Automat	iic: Native code			Select	
ID	Title	Туре	User Name	Session	
9028		x86	FC-EDA\rgandia	1	
10416		x64	FC-EDA\rgandia	1	
1244		x64	FC-EDA\rgandia	1	
7148		x64	FC-EDA\rgandia	1	
7224		x86	FC-EDA\rgandia	1	
7064		x64	FC-EDA\rgandia	1	
10016		x64	FC-EDA\rgandia [a	1	
6644		x64	FC-EDA\rgandia	1	
4472		x64	FC-EDA\rgandia	1	=
5848		x64	FC-EDA\rgandia [a	1	
3716		x64	FC-EDA\rgandia	1	-
					1000
all users				Refresh	
	s you sel VSMON Automat D 2028 10416 1244 7148 7224 7064 10016 5644 4472 5848 8716 5716 5716 5716 5716 5716 5716 5716 5	s you select processes on this compu VSMON.EXE). Automatic: Native code D Title 2028 10416 1244 7/148 7/224 7/064 10016 5/644 1472 5/848 8/716	s you select processes on this computer or a remote computer runnin VSMON.EXE).	s you select processes on this computer or a remote computer running the Microsoft Visual Stur VSMON.EXE). Automatic: Native code D Title Type User Name 2028 x86 FC-EDA\rgandia 10416 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 1244 x64 FC-EDA\rgandia 10016 x64 FC-EDA\rgandia	s you select processes on this computer or a remote computer running the Microsoft Visual Studio Remote VSMON.EXE). Automatic: Native code Select D Title Type User Name Session 2028 x86 FC-EDA\rgandia 1 20416 x64 FC-EDA\rgandia 1 2244 x64 FC-EDA\rgandia 1 244 x64 FC-EDA\rgandia 1 255 x64 FC-EDA\rgandia 1 26644 x64 FC-EDA\rgandia 1 26644 FC-EDA\rgandia 1 26644 FC-EDA\rgandia 1 26644 FC-EDA\rgandia 1 267 FC-EDA\rgandia 1 26848 X64 FC-EDA\rgandia 1 26848 FC-EDA\rgandia 1 26848 FC-EDA\rgandia 1 2716 FC-EDA\rgandia 1 2716 FC-EDA\rgandia 1 2717 FC-EDA\rgandia 1 2718 FC-EDA\rgandia 1 2718 FC-EDA\rgandia 1 2719 FC-EDA\rgandia 1 2710 FC-EDA

- 5. Select all the SimSrvr.exe available (clicking on Ctrl+LMB) and click on Attach.
- 6. Click on Run Simulation in PSpice
- 7. If everything works fine, PSpice keeps running and the pointer remains in VS:

/pspCoreLos /pspinit.tc	———— INFO(ORPROBE-3209): Simulation Profile: SCHEMATIC1-Tr INFO(ORPROBE-3183): Simulation running	ans ———



Flow(

Application Note

8. Click on Continue to jump from breakpoint to breakpoint.



9. Use Visual Studio Watch function to see any variable value



Watch 1		<b>▼</b> □ ×
Name	Value	Туре 🗅
🕝 mDigSimTimeSt	1.0000000000000000e-010	double



### 3.6 Hardware in the Loop using Arduino

This module shows Hardware in the Loop using an Arduino Board where the communication between the physical board and PSpice is done using Serial USB protocol.

This board has been chosen because it is worldwide used for multiple applications, it is cheap and it allows to demonstrate this new PSpice feature. Of course another boards could be used.

This example demonstrates:

- The advantages of using Virtual Prototyping in PSpice, focusing it to Hardware in the Loop, where data flows from PSpice to the board and vice versa.
- How to define a component using DMI Template Code Generator in the context of Hardware in the Loop.

The steps to be followed to design such example are more or less the same than for Digital Power Supply, but some extra steps have to be considered. Let start with the next schema:



Steps:

- 1. Launch Model Editor
- 2. Select Menu Item Model → DMI Template Code Generator

Enter the data as follows:

			Application Note
DMI Template	Code Generator	8	
Use this dialog-b SystemC. The dia Recommended 1. Test the n 2. Create th 3. Use the g in the sch	box to auto-generate DMI template alog-box also imports the Verilog steps: model code stand-alone by buildi e PSpice-DMI adapter code, and renerated PSpice library (lib file) rematic for PSpice simulation.	e code for the following PSpice-DMI models: Analog, Digital, and p-A Compact Device models using ADMS. ing an exe. ledit it in Visual Studio to insert model code. to create a schematic symbol.The generated symbol can be placed	
Part Details —			1
	PartName	ArduinoHILDemo1	
	Part Type	Analog	
Terminals			, 1
	Model Type	Voltage-Controlled Voltage Source 💌	
	Terminal Entry		
Parameters -			
	Global Parameters		
	Device Parameters		
	Model Parameters		
Output			1
	DLL File Name	ArduinoHILDemo1.dll	
	Log File Name	ArduinoHILDemo1.log	
	DLL Location	D:\PSpice\Application Notes\Flov Browse	
Consili dest		OK Cancel Help	
Specity device	parameters; and global paramete	ers requirea by the model logic	

On Terminal Entry, this is what you have to see:

Terminal Name	Terminal Description
Input1	Control Input 1
Input2	Control Input 2
Output1	Output 1
Output2	Output 2

For the DLL Location select please the directory of this example:

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed/DMI\_Arduino

3. Click on Device Parameters and define two new parameters as in the image:

Flow

Device Parameters	;		
Specify the D	evice paramete	rs.	
Enter number	ofparameters	2	
Parameter Name	Parameter Type	Default Value	Parameter Description
BAUDRATE	double	9600	
COMPORT	string	COM4	
			OK Cancel Apply

- 4. Click Apply and OK.
- 5. Click OK and generate all the files. Automatically a new folder called ArduinoHILDemo1 (Part Name) is included in

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed/DMI\_Arduino
with another two folders: code and lib.



In the code folder you will find the files to be used in Visual Studio to generate the DLL:

ArduinoHILDemo1.log
🛱 ArduinoHILDemo1.vcxproj
TrduinoHILDemo1_user.cpp
bspArduinoHILDemo1.cpp
bi pspArduinoHILDemo1.h
bspEngFunc.cpp
bi pspEngFunc.h
b PSpiceBase.h
B PSpiceCMIApiDefs.h
B PSpiceCommonAPIDefs.h
B PSpiceDigApiDefs.h
StdAfx.cpp
前 StdAfx h

In Lib folder you will find the PSpice Model:

**Flow**C

Application Note



ArduinoHILDemo1.lib
input.json

NOTE: The DLL will be generated when all the files are compiled in VS.

6. Include inside of the folder

 $<\!\!directory>Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\DMI\_Arduino\ArduinoHILDemo1\code$ 

the files, which allow the USB Serial transmission of data. They are located in

<directory>Circuits/Hardware\_in\_the\_Loop/To\_be\_completed/USB\_Serial\_Protocol



7. Open Visual Studio Community 2013 and click on File→Open→Project/Solution to load the project:

Open Project					×
	_completed	rduinoHILDemo1 🕨	code	✓ ← code durchs	uchen 🔎
Organisieren • Neu	uer Ordner				•
la OneDrive	Name	-	Änderungsdatum	Тур	Größe
lass Zuletzt besucht	T ArduinoHILDemo1.vcxp	proj	21.06.2016 15:07	VC++ Project	8 KB
<ul> <li>Bibliotheken</li> <li>Bilder</li> <li>Dokumente</li> <li>Musik</li> <li>Videos</li> <li>Computer</li> <li>Windows7_OS (C:)</li> <li>Data2 (D:)</li> </ul>	II				
support (\\dc01) (			111		- F
	Options: C Add to Solution Close Solution				
Datei	iname: ArduinoHILDemo1.vcxproj	j		✓ All Project Files (	*.sln;*.dsw;* 🔻
				Öffnen	Abbrechen

8. On the solution Explorer Tab select ArduinoHILDemo1, click RMB → Add → Existing Item and look for the USB Serial Protocol Files.



- 9. Select CSerial.cpp and CSerial.h and click Add.
- 10. Open next files and compare them with the completed ones that are located in <directory>/Completed/DMI\_Arduino in order to analyse the code that was added. These files define the working of the model and internal configuration of the code so that the dll works properly using the interface. (Optional).
  - ArduinoHILDemo1\_user.cpp
  - pspArduinoHILDemo1.cpp
  - pspArduinoHILDemo1.h
- 11. Now that you know the differences in terms of code among these files, go to the directory

<directory>\Circuits\Hardware\_in\_the\_Loop\Completed\DMI\_Arduino\ArduinoHILDem o1\code

copy the files that you can see in point 10 and add them in

 $<\!\!directory>\!\!\backslash Circuits \\ Hardware_in\_the\_Loop \\ To\_be\_completed \\ DMI\_Arduino \\ HILDemo1 \\ code$ 

replacing those you have created previously.

FlowC



12. Open Visual Studio again and select Release and x64 from the top of the window:



#### 13. Click on Build $\rightarrow$ Build Solution and generate DLL:

1>-	Build started: Project: ArduinoHILDemo1, Configuration: Release x64
1>	ArduinoHILDemo1_user.cpp
1>A	rduinoHILDemo1_user.cpp(52): warning C4267: 'argument' : conversion from 'size_t' to 'int', possible loss of data
1>	CSerial.cpp
1>	pspArduinoHILDemo1.cpp
1>	pspEngFunc.cpp
1>	StdAfx.cpp
1>	Creating library D:\PSpice\Application Notes\FlowCAD_AN_Device_Modeling_Interface\Circuits\Hardware_in_the_Loc
1>	Generating code
1>	Finished generating code
1>	ArduinoHILDemo1.vcxproj -> D:\PSpice\Application Notes\FlowCAD_AN_Device_Modeling_Interface\Circuits\Hardware_in_
	Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped

14. Install Arduino IDE in your computer downloading the software from the official Homepage:

https://www.arduino.cc/en/Main/Software

- 15. Open the Arduino Code located in
  - <directory>Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\Arduino\_Code\Example



- 16. Connect the Arduino Board with the USB Cable.
- 17. Click on Tools and make sure that the Port, where the Arduino Board is connected is selected to COM4.

1 1 1	Auto Format	Strg+T		
	Archive Sketch			
	Fix Encoding & Reload			
	Serial Monitor	Strg+Umschalt+M		
	Serial Plotter	Strg+Umschalt+L		
Stri	Board: "Arduino/Genuino Un	o"	•	
nacc	Port: "COM4 (Arduino/Genuir	no Uno)"		Serial ports
DPir				
EDPir	Drogrommer "AV/DICD mkII"			COM3



18. Upload the code in the Arduino Board clicking on Sketch  $\rightarrow$  Upload:



- 19. If everything works well, you will see the LED, the pin L and the pin ON turned on.
  - In Arduino software something like that is shown:



20. Design the Hardware to be simulated:







Hardware connection schematic:



- 21. Open OrCAD Capture
- 22. Select Allegro Design Entry CIS or OrCAD PSpice Designer Plus:

**Cadence Product Choices** Please select the suite from which to check out the OrCAD Capture feature: OrCAD Capture . OrCAD PSpice Designer Ξ OrCAD PCB Designer Professional OK OrCAD PCB Designer Professional w/PSpice Cancel OrCAD\_Capture\_CIS\_option with Capture OrCAD\_Capture\_CIS\_option with OrCAD PCB Designer Profession: OrCAD\_Capture\_CIS\_option with OrCAD PCB Designer Profession: -III \*D\_0------00 ----4 b 📃 Use as default

FlowC

Application Note

23. Open the project located in <a href="https://circuits/Hardware\_in\_the\_Loop/To\_be\_completed/Circuit/ArduinoHiL.op"></a>, Circuits/Hardware\_in\_the\_Loop/To\_be\_completed/Circuit/ArduinoHiL.op</a>



As you can see, everything is defined except the Arduino component.

24. Open Model Editor from PSpice Utilities:



Flow

Application Note

25. Click on File → Open and load the library that was created automatically when you defined the Arduino component using DMI Template Code Generator. It should be placed in

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\DMI\_Arduino\Arduino
HILDemo1\Lib



26. Modify the description introducing the information that is available in the file "Change\_Me" located in

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\DMI\_Arduino

FlowCAD Application Note

#### Copy and paste the whole data in your PSpice Model

ArduinohilDemo1.lib:Arduir	noHILDemo	1 - PSpice Model Editor - [Model Text] ols Window Help	
	% D 🖻		
Models List Model Na Type I ArduinoHIL SUBCKT	(a) Modified	<pre>.subckt ArduinoHILDemol D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 A2 A1 A0 + PARAMS: inst BAUDRATE=9600 inst COMPORT="COM4" mod_MaxStep: Y1 D10 0 A1 0 A0 D8 0 CMI ArduinoHILDemol.dll ArduinoHILDemol BAUDRATE={inst_BAUDRATE} COMPORT={inst_COMPORT} *Y1 D10 0 A1 X CMI ArduinoHILDemol.dll ArduinoHILDemol_model {inst_BAUDRATE} COMPORT={inst_COMPORT} .model ArduinoHILDemol_model CMI ArduinoHILDemol MaxStepSize= RD13 D13 0 1G RD12 D12 0 1G RD10 D10 0 1G RD00 D9 0 1G RD00 D9 0 1G RD00 D5 0 1G RD00 D5 0 1G RD04 D4 0 1G RD04 D4 0 1G RD03 D3 0 1G RD04 D4 0 1G RD01 D1 0 1G RA5 A5 0 1G RA4 A4 0 1G RA3 A3 0 1G RA4 A4 0 1G RA1 A1 0 1G RA0 A0 0 1G *RX X 0 1G .ends</pre>	D2 D1 D0 A5 A4 A3 Size=1e+30 1_model PARAMS: PARAMS: BAUDRATE= ={mod_MaxStepSize}

- 27. Save.
- 28. With Model Editor opened, click on File → Export to Part Library to generate the symbol (.olb), which is being placed in the schematic:

File	Edit	View	Model	Plot	Tools
<u>N</u>	ew			Ct	trl+N
0	pen			Ct	trl+O
<u>C</u> I	ose				
🖬 <u>S</u> a	ive			C	trl+S
Sa	ave <u>A</u> s	•			
Pr	int			С	trl+P
🗅 Pr	int Pre	view			
Pa	ige Set	t <u>u</u> p			
Ex	port T	o Part L	ibrary		Ν
М	odel Ir	mport V	Vizard		63
尾 Er	ncrypt l	Library			
1	arduin	ohilden	no1.lib		
2	Arduin	oHILDe	emo1.lib		
<u>3</u>	arduin	ohilden	no1.lib		
<u>4</u>	N:\tmp	_rober	to\\MUF	RS320.	lib
<u>5</u>	arduin	ohilden	no1.lib		
<u>6</u>	arduin	ohilden	no1.lib		
Ex	it				

29. Click on OK

Create Parts for Library	×
Enter Input Model Library:	
p\Completed\Circuit\Library\Capture\arduinohildemo1.lib	Browse
Enter Output Part Library:	
\Completed\Circuit\Library\Capture\arduinohildemo1.olb	Browse
OK Cancel Help	1

FlowC

Application Note

30. Now that your Symbol has been created, open OrCAD Capture and place it. But do **NOT** take the symbol you have just created, but the symbol located in

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\Circuit\Library\Capture

as I have reorganized the pins position.

a. Click on Place  $\rightarrow$  Part



b. Click on Add Library



Place Part		t . x
Part		2
		_
PartList		Y
0402CS-10N/COILCRAFT 0402CS-11N/COILCRAFT 0402CS-12N/COILCRAFT 0402CS-13N/COILCRAFT 0402CS-15N/COILCRAFT 0402CS-15N/COILCRAFT 0402CS-18N/COILCRAFT 0402CS-19N/COILCRAFT		•
Libraries:	~~~	~
1_SHOT 7400 74AC 74ACT 74ALS 74AS	Ad	ld Library (Al1
Packaging		
Parts per P	kg:	1
Part.		Ŧ
Type:		
Search for Part		

c. Select Arduino.OLB and place it making double click on the component. When it is placed, click on H to mirror horizontally and connect it with the pins:



d. As you have placed an Arduino Symbol without an associated PSpice Model, select the Symbol you have just placed and click RMB:


0.000		In the second
	Associate PSpice Model	
	Edit PSpice Model	MOI
UNC	Edit PSpice Stimulus	
011	Export Parameters to Optimizer	1111111
9-11	Import Model Parameters	
Arduino UNC	Descend Hierarchy	1 T
	Synchronize Up	<b>F</b>
12	Synchronize Down	1111111
10	Synchronize Across	
7	Connect to Bus	
4	User Assigned Reference	1111111
3		

e. Click Yes and a windows pops up:

-	D:\temp\Circuits\Hardware_in_the_Loop\Completed	\Circuit\Li	brary\Capture\arduinoh	ildemo1.lib 🔻
	Model: ARDUINOHILDEM01		Part: U12	
	Show All		Arduir	IO UNO
4	Matching Models		A0 A1 A2 A3	D13 D12 D11 D10 D9 D8 D7
97			A4 A5	D6 D5 D4 D3 D2 D1 D0

f. Click Next:

A CARGONIA DA	Use "View Model Text" but	on to view the model defi	nition.	,	
and a second	Model: ARDUINOHILDE	MO1		Part: U12	
	Show Invisible Symbol I	Power Pins		Ardu	ino UNO
Concession of the local division of the loca	Model Terminal	Symbol Pin	•		D13
	D13				D12
ALC: NO.	D12		=	A0	D10
	D11			A1 A2	D8
and interactions	D10			A3	D7
	D9			A4	D6
547	D8			~~~	D4
N NYA	D7				D3
	D6				D1
	D5				D0
	D4		*		

g. Associate each Model Terminal yith the corresponding Symbol PIN:

FlowCAD

Application Note

Constrain de las	Use "View Model Tex	t" button to view the model d	efinition.	inodel terminals in	lay be leit unassociate
	Model: ARDUINO	HILDEMO1		Part: U12	
	Show Invisible Syn	mbol Power Pins		A	rduino UNO
Name of Street	Model Terminal	Symbol Pin	•		D13
	D8	D8			D12
D.C.	D7	D7		A0	D10
	D6	D6		A1	D9
and some states	D5	D5	Ξ	A3	D7
and the second	D4	D4		A4	D6
07	D3	D3		Ab	D3
	D2	D2			D3
	D1	D1			D2
	D0	D0	-		D0
	< III		•		

h. Click on Fertig and next Windows pops up:

Associate	PSpice Model			×
?	Would you like to ass same source packag	ociate PSpice model on e in the design?	y to the selected part insta	ince, or all part instances of the
	Update Current	Update All	Cancel	Help

i. Click on Update All and OK



31. Now you are ready to define the Simulation Profile. Click on New Simulation with the name Trans and click on Create:

- <u>R</u>		R	12	R.	18 (
🖛 🔤 New	Simulati	on Pr	ofile	<b> </b>	È
AGE1	U1	٦			
5					

	FlowCAD
-	Application Note
New Simulation	
Name:	
Trans	
Inherit From:	
none 🗸	
Root Schematic: SCHEMATIC1	

32. Define a Time Domain Simulation and fill in the options with the next values:

Simulation Settings - Trans General Analysis Configuration	Files Options Data Collection Probe V	Vindow
Analysis type: Time Domain (Transient) Options: General Settings Monte Carlo/Worst Case Parametric Sweep Temperature (Sweep) Save Bias Point Load Bias Point Save Check Points Restart Simulation	Run to time:       10         Start saving data after:       0         Transient options          Maximum step size:       1m         Skip the initial transient bias point         Run in resume mode	seconds (TSTOP) seconds calculation (SKIPBP) Output File Options
	OK Abbrechen	Übernehmen Hilfe

33. Click on Configuration Files, select Library and add the PSpice Model for the filter and the Arduino Board:

Category:	Details	
Stimulus		Browse
Include	Configured Files	
✓ Update Index	<ul> <li>\library\capture\arduinohildemo1.lib</li> <li>\library\capture\matlab.lib</li> </ul>	Add as Global
	• nom.lib	Add to Design
		Add to Profile
		Edit
		Change
	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )	
	Library Path	
	"C:\Cadence\SPB_17.2\tools\PSpice\Library"	Browse



Your library for the Arduino Board is in <directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\DMI\_Arduino\A rduinoHILDemo1\lib

The library for the Matlab Filter is in <directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\Circuit\Library\ Capture

34. Click on Probe Window and select the next configuration:

🛃 Simulation Settings - Trans	×
General Analysis Configuration Files Options Data Collection Probe Window	
Display Probe window when profile is opened.	
✓ Display Probe window:	
Ø during simulation.	
C after simulation has completed.	
Show     All markers on open schematics.     C Last plot.     Nothing.	
OK Abbrechen Übernehmen Hi	fe

35. Click on OK.

## 36. In the Schematic click on PSpice $\rightarrow$ Markers $\rightarrow$ Voltage Level

PSpice	Accessories	Options W	indow Help	
<mark>⊠ N</mark> ew <u>E</u> dit S	Simulation Pro Simulation Pro	ofile ifile		Q & Q
Run View View Creat View	Simulation Re Output File e Netlist	F11 esults F12	<u>-100b</u> ,	
A <u>d</u> va	nced Analysis	•	-	
Mark	ers		R Voltage Level	
<u>B</u> ias I	Points	•	Voltage Differenti	al

and place this marker as in the image:



Flow(

Application Note

37. Before you can simulate Hardware in the Loop, you have to place the DLL's in the PSpice Simulation Settings Folder you have just created. Copy your DLL located in

 $\label{eq:linear} $$ $$ circuits\Bardware_in_the\_Loop\To\_be\_completed\DMI\_Arduino\Arduino\HILDemo1\code\x64\Release\ArduinoHILDemo1.dll $$$ 

and past it in

<directory>Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\Circuit\ArduinoHiLPSpiceFiles\SCHEMATIC1\Trans

38. Do the same with the DLL for the Matlab Filter located in

<directory>\Circuits\Hardware\_in\_the\_Loop\To\_be\_completed\Circuit\Library\DLL

39. Now connect the hardware to the computer (if it is not already connected) and click on Run Simulation:



NOTE: If you get a Netlisting Error, you have to connect all the unconnected pins together.

			Application Note
	Arduino UNO		
	AIGUINO		
	· ·		
	D13		
	D12		
	D11		
	D10 A A A A AO		
	D9 * * * * * * A1		
	D8 * * * * * * A2	· · · ·	
	D7 * * * * * * A3	<b>•</b>	
	D6 A A A A A A	<b>f</b> (a	
	D5 A A A A A A A A A A A A A A A A A A A		
Move	D4 * * * * * * * * * *		
	D3 and a second second second		
· · · · · · · · ·	D2 * * * * * * * * * *		
· · · · · · · ·	D1 · · · · · · · · · · · · ·		
	D0 · · · · · · · · · · · ·		
< R2			
≥ 1G			
	Arduino		

Flow

40. When the Probe Window opens, click on Tools  $\rightarrow$  Options and select Auto-Update Intervals as in the image.

Probe Settings				8
General Large Data File	Cursor Settings	Color Settings	Font Settings	
Use Symbols Properties Never Always	Trace Color Normal Match A Sequer Unique	Scheme xxis ntial Per Axis By File	Mark Data Points Display Evaluation Display Statistics Highlight Error States	
Use ScrollBars Auto Never Always	Auto-Update Auto Every Every	e Intervals		
10     Number of Histogram Divisions       1     Default Trace Width				
OK Abbrechen Reset				

41. Test the simulation HiL moving the board, incrementing the sensor temperature or varying the amount of received light:



**NOTE:** Move the voltage Marker you placed to the different nodes to visualize the different results.

- Temperature Sensor:

- Light Sensor:



- Tilt Sensor:



