OPAL·RT

OP4200 EHS QUICKSTART GUIDE

eFPGAsim Toolbox

RTLAB / CPU Section

www.opal-rt.com

1751 Richardson, suite 2525
Montréal (Québec) Canada H3K 1G6

www.opal-rt.com

Contents

INTRODUCTION

The eHS solver is an FPGA-based technology developed by OPAL-RT for real-time power electronics simulation. Thanks to a convenient circuit schematic graphical user interface, the FPGA code is automatically generated, making FPGA-based simulation accessible to a large number of users.

This document provides basic information on how to create a new project with a template that includes the eHS solver and how to run a real-time simulation with an example model included in the template. The guide is designed to be used by the OP4200 platform.

REQUIREMENTS

OPAL-RT HARDWARE

- OP4200

RECOMMENDED OPERATING SYSTEM

Microsoft Windows 7 64bits Version 6.1 (Build 7601: Service Pack 1)

PRE-INSTALLED SOFTWARE

The following software must be installed and functioning to perform the tests described in this document. Confirm that the required software is installed and test to ensure they are functioning properly.

Type >> *ver* on the MATLAB prompt and verify that you have the following toolboxes (or later versions of them) installed.

Recommended configurations for basic use (only circuit simulation):

- MATLAB 2011b / 2013a / 2015aSP1, 32-bit version
- RT-LAB v11.1.1 or later[1]
- At least one of this tool:
  - SimPowerSystem “Specialized Technology” Simulink library
  - Powersim PSIM v9.3.4 or v10.0.6 (v10.0.5 is unsupported)
  - Plexim PLECS Simulink Library v3.7.4
  - NI Multisim 13

The RT-XSG toolbox requires the following software to generate programming files for reconfigurable devices and to program the platform:

Recommended configuration for firmware generation, for Xilinx Zynq 7 series:

- MATLAB 2015b, 64-bit version
- Xilinx Vivado suite 2015.3 & Xilinx Vivado System Generator for DSP 2015.3
- RT-XSG v3.1.2 or later

LICENSED RT-LAB COMPONENTS

You must have licenses for the following RT-LAB components to run the examples provided in this document. Verify with your sales representative that your system includes these licenses.

RTLAB_RT	RTLAB_XHP	RTLAB_DEV	RTLAB_NUM_CORES >= 1
RTE_RT	RTE_NUM_CORES >= 1	XSG_EHS	

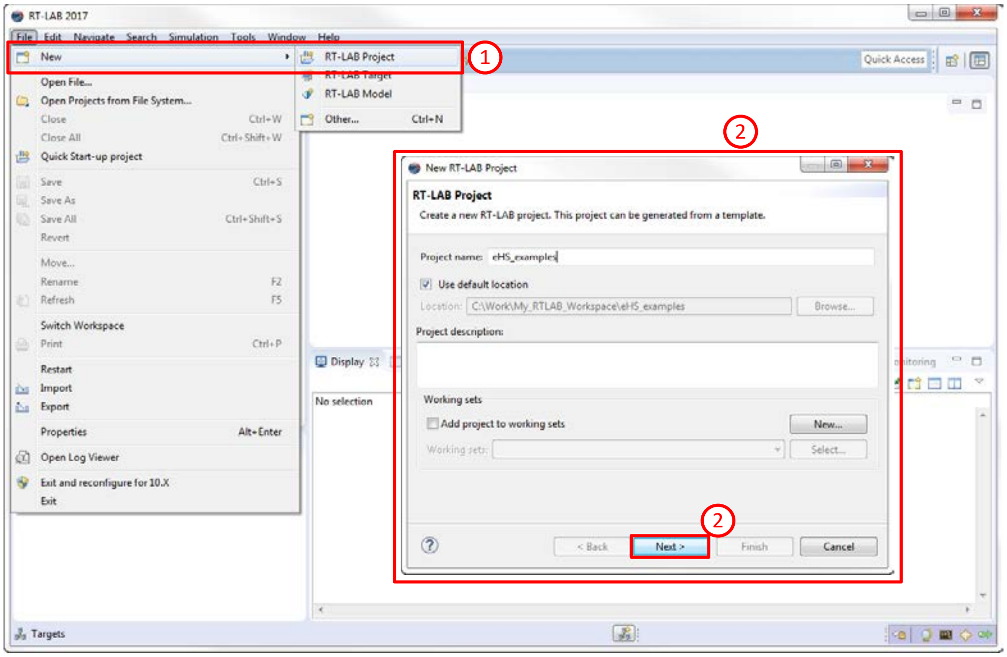
[1] The eHSx32 for OP4200 block is currently only supported as of RT-LAB 11.1.1.

RUNNING A MODEL ON OP4200 TARGETS

CREATING A NEW RT-LAB PROJECT

Open RT-LAB and follow these steps to create a new RT-LAB project.

1. In the main RT-LAB UI, navigate to **File > New > RT-LAB Project** to open a new project wizard.
2. Enter a project name (e.g. *eHS_examples*), then click **Next**.
3. Browse the template directory to select the model and add it to the project. Open the *eFPGAsim* folder and select **eHS with IOs > OP4200 > OP4200_eHSx32_Boost_and_TwoLevel_Inverter.**
4. Click **Finish** to create the new RT-LAB project in the Project Explorer.



New RT-LAB Project

RT-LAB Project

Create a new RT-LAB project. This project can be generated from a template.

Project name: eHS_examples

☑ Use default location

Location: C:\Work\My_RTLAB_Workspace\eHS_examples

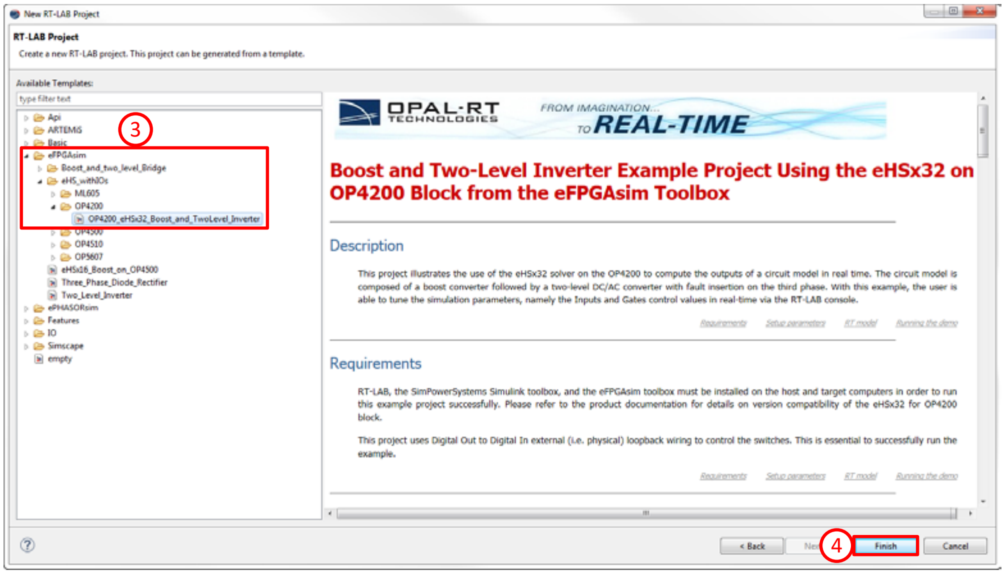
Project description:

Working sets

☐ Add project to working sets

Working sets:

Figure 1: Creating a new project



New RT-LAB Project

RT-LAB Project

Create a new RT-LAB project. This project can be generated from a template.

Available Templates:

type filter text

Boost and Two-Level Inverter Example Project Using the eHSx32 on OP4200 Block from the eFPGAsim Toolbox

Description

This project illustrates the use of the eHSx32 solver on the OP4200 to compute the outputs of a circuit model in real time. The circuit model is composed of a boost converter followed by a two-level DC/AC converter with fault insertion on the third phase. With this example, the user is able to tune the simulation parameters, namely the Inputs and Gates control values in real-time via the RT-LAB console.

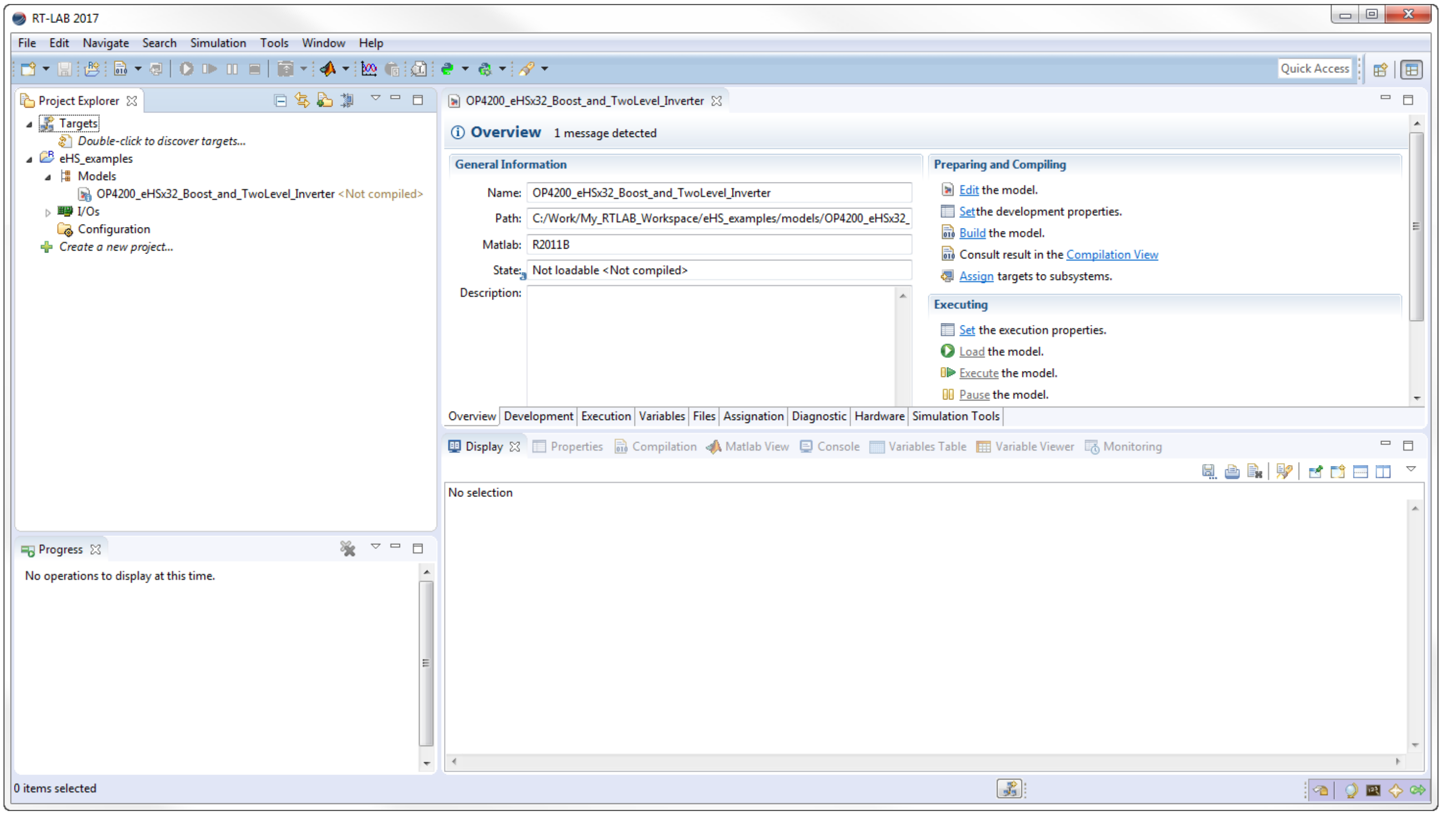
Requirements

RT-LAB, the SimPowerSystems Simulink toolbox, and the eFPGAsim toolbox must be installed on the host and target computers in order to run this example project successfully. Please refer to the product documentation for details on version compatibility of the eHSx32 for OP4200 block.

This project uses Digital Out to Digital In external (i.e. physical) loopback wiring to control the switches. This is essential to successfully run the example.

Figure 2: Browsing model templates

The *eHS_examples* project will be created and the corresponding RT-LAB model will be imported into the project workspace (in this example, the model is named *OP4200_eHSx32_Boost_and_TwoLevel_Inverter*).



Double-click to discover targets...

OP4200_eHSx32_Boost_and_TwoLevel_Inverter <Not compiled>

Create a new project...

No operations to display at this time.

1 message detected

Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter

Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_

Matlab: R2011B

State: Not loadable <Not compiled>

Description:

Edit the model.

Setthe development properties.

Build the model.

Consult result in the Compilation View

Assign targets to subsystems.

Set the execution properties.

Load the model.

Execute the model.

Pause the model.

No selection

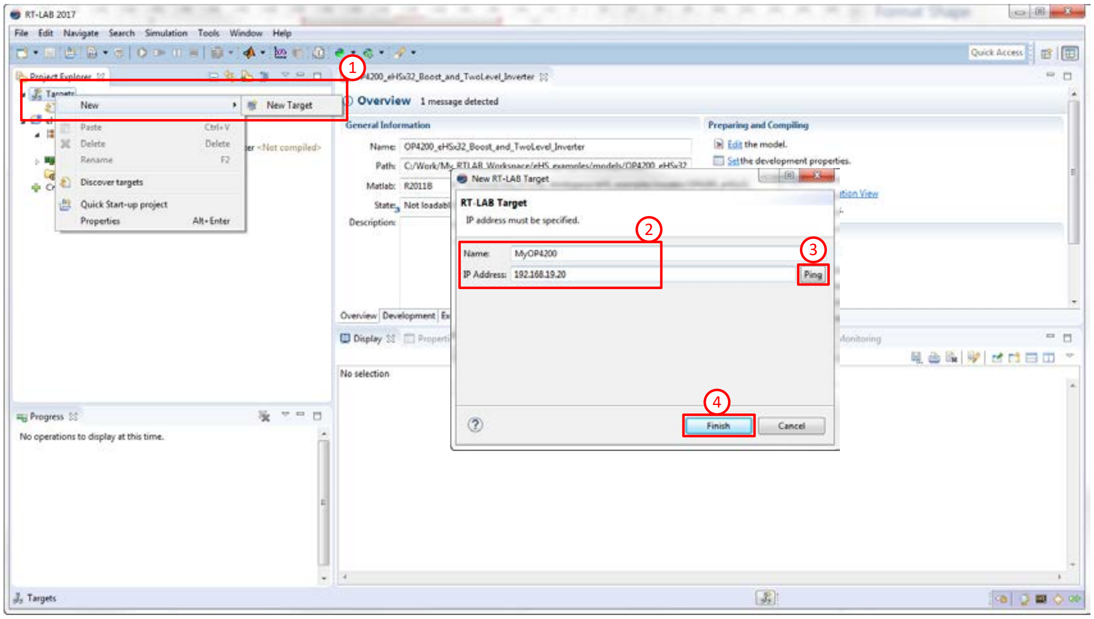
0 items selected

Figure 3: Example workspace

ADDING A TARGET TO THE WORKSPACE

To simulate a model in real-time, a real-time target must be added to the workspace. Follow the steps below to add a target to the RT-LAB workspace.

1. Right-click on the **Targets** heading and select **New>New Target.**
2. Enter a name for the target (could be anything) and enter the IP address of the target.
3. (Optional) Click on the **Ping** button to ping the target and validate that it can be discovered on the network.
4. Click **Finish.** The target is now added to the RT-LAB workspace.



Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32
Matlab: R2011B
State: Not loadable

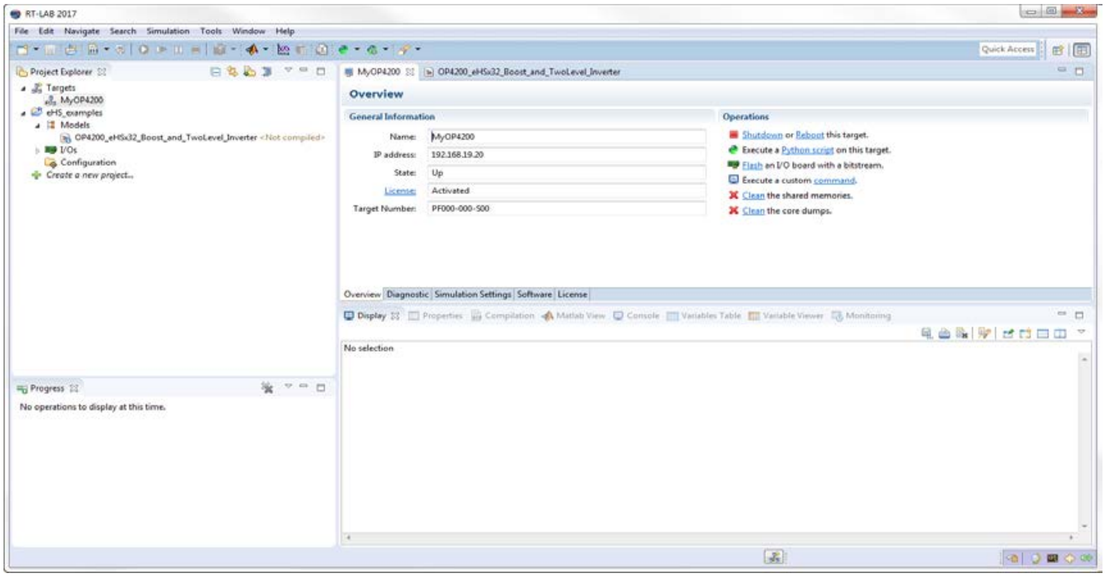
New RT-LAB Target

RT-LAB Target

IP address must be specified.

Name: MyOP4200
IP Address: 192.168.19.20

Figure 4: Adding a target to the workspace


RT-LAB 2017
File Edit Navigate Search Simulation Tools Window Help
Project Explorer
Targets
MyOP4200
eHS_examples
Models
OP4200_eHSx32_Boost_and_TwoLevel_Inverter <Not compiled>
I/Os
Configuration
Create a new project...
MyOP4200
OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Overview
General Information
Operations
Overview Diagnostic Simulation Settings Software License
Display Properties Compilation Matlab View Console Variables Table Variable Viewer Monitoring
Progress
No operations to display at this time.
No selection

Name: MyOP4200
IP address: 192.168.19.20
State: Up
License: Activated
Target Number: PF000-000-500

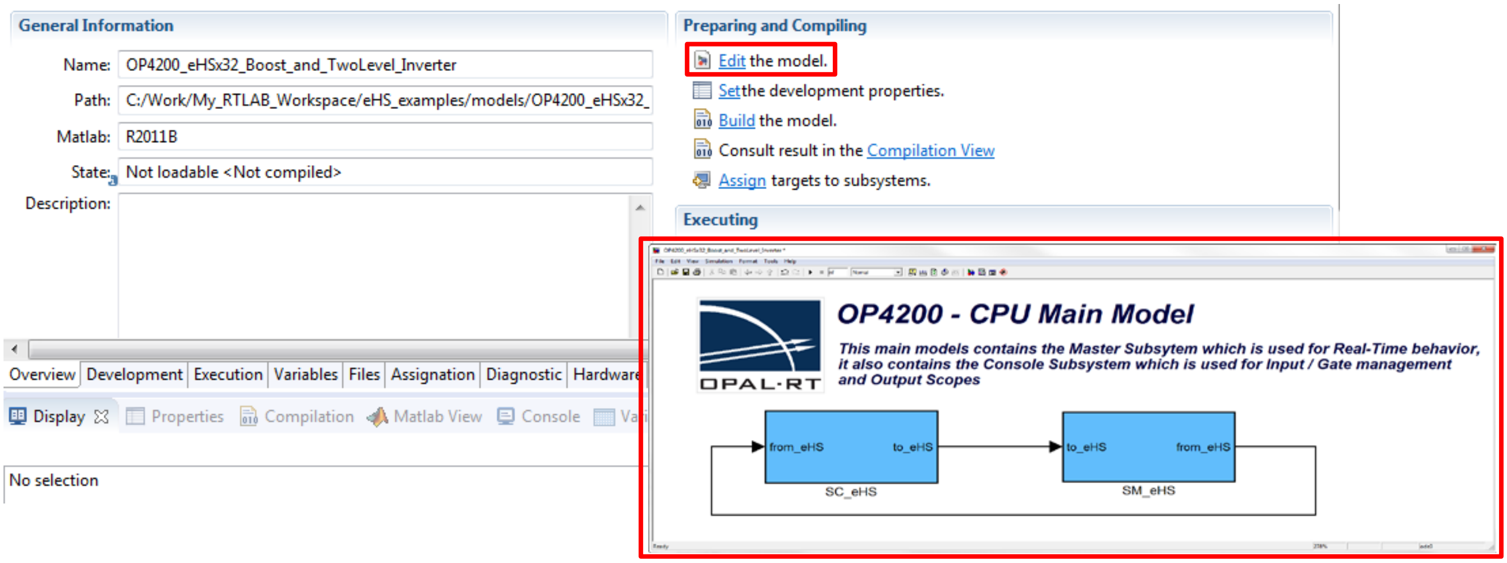
Shutdown or Reboot this target.
Execute a Python script on this target.
Flash an I/O board with a bitstream.
Execute a custom command.
Clean the shared memories.
Clean the core dumps.

Figure 5: Target added to workspace

EDITING THE MODEL

Before building the model, verify that you can access and edit the model.

Click on **Edit the model** in the *Preparing and Compiling* pane. The model file will be opened in a new MATLAB/Simulink window.



General Information

Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter

Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_

Matlab: R2011B

State: Not loadable <Not compiled>

Description:

Preparing and Compiling

- Edit the model.
- Set the development properties.
- Build the model.
- Consult result in the Compilation View
- Assign targets to subsystems.

Executing

No selection

OP4200 - CPU Main Model

This main models contains the Master Subsytem which is used for Real-Time behavior, it also contains the Console Subsystem which is used for Input / Gate management and Output Scopes

Figure 6: Editing the model

MODEL DESCRIPTION

The model is composed of two main subsystems:

1. *SC_eHS*: Subsystem executed by the host computer during the simulation to monitor and control the simulation.
2. *SM_eHS*: Subsystem executed by the target simulator, in real-time, on the system CPU that communicates with the FPGA board and the physical system I/Os.

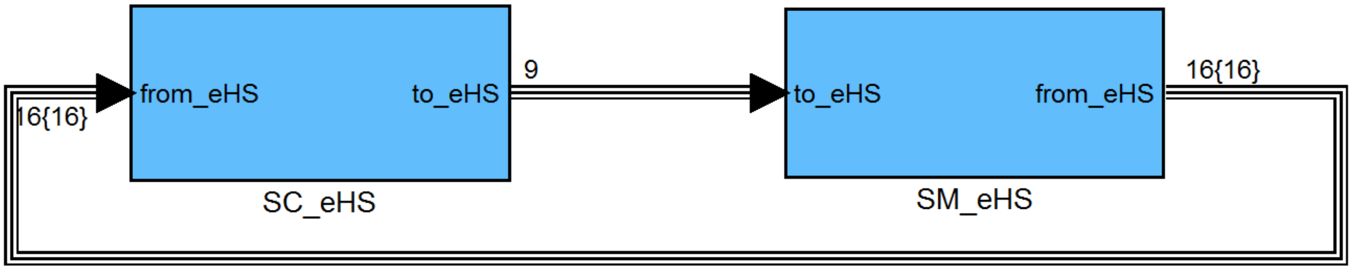
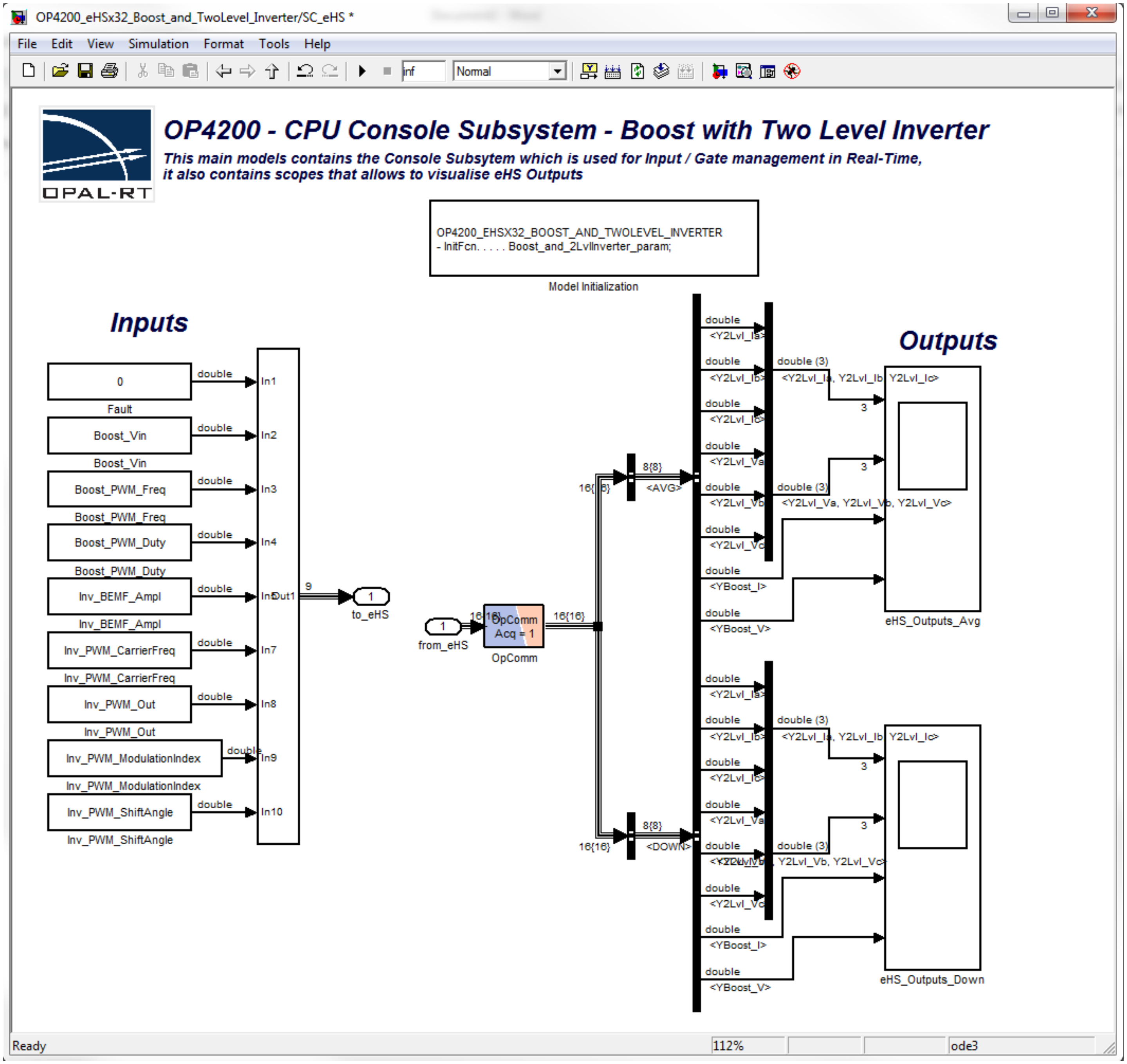


Figure 7: Model diagram

SC_EHS

Double click on the **SC_eHS** subsystem to open the *SC_eHS* details window.



OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SC_eHS *

OP4200 - CPU Console Subsystem - Boost with Two Level Inverter

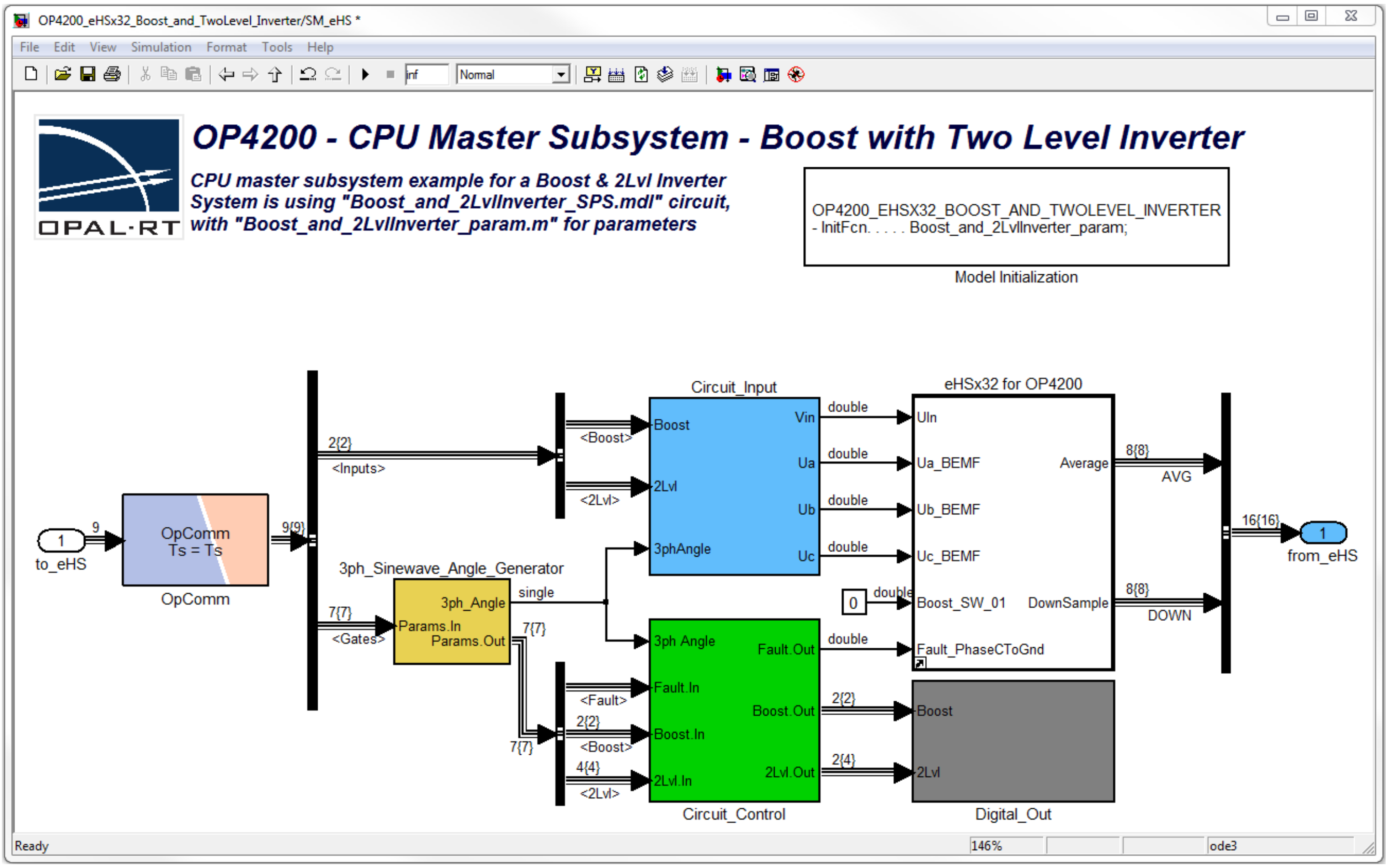
This main models contains the Console Subsytem which is used for Input / Gate management in Real-Time, it also contains scopes that allows to visualise eHS Outputs

OP4200_EHSX32_BOOST_AND_TWOLEVEL_INVERTER
- InitFcn. . . . . Boost_and_2LvlInverter_param;

Figure 8: Model SC_eHS window

SM_eHS

Double-click on the **SM_eHS** to open the *SM_eHS* details window. This window provides a detailed diagram of the Master subsystem.



OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS *

OP4200 - CPU Master Subsystem - Boost with Two Level Inverter

CPU master subsystem example for a Boost & 2Lvl Inverter System is using "Boost_and_2LvlInverter_SPS.mdl" circuit, with "Boost_and_2LvlInverter_param.m" for parameters

OP4200_EHSX32_BOOST_AND_TWOLEVEL_INVERTER
- InitFcn. . . . . Boost_and_2LvlInverter_param;

Model Initialization

Figure 9: Model SM_eHS window

eHS Block for OP4200 (eHSx32)

The *eHS block for OP4200* communicates with the FPGA board to initialize the eHS core. During real-time operation, it provides eHS inputs (the circuit's Sources and Gates control signals) and reads the eHS outputs (the circuit's current and voltage measurements) at the RT-LAB model rate.

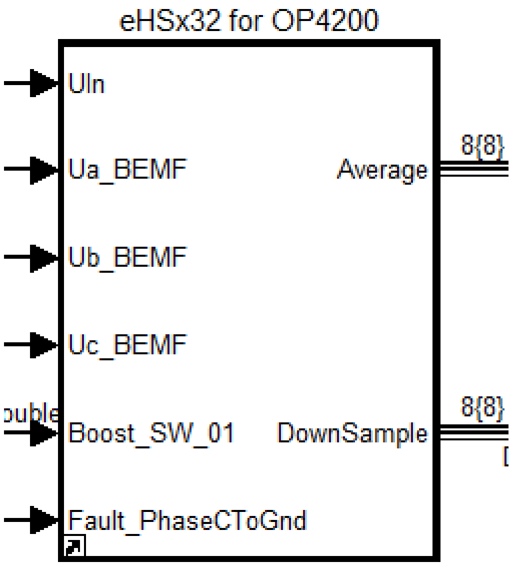
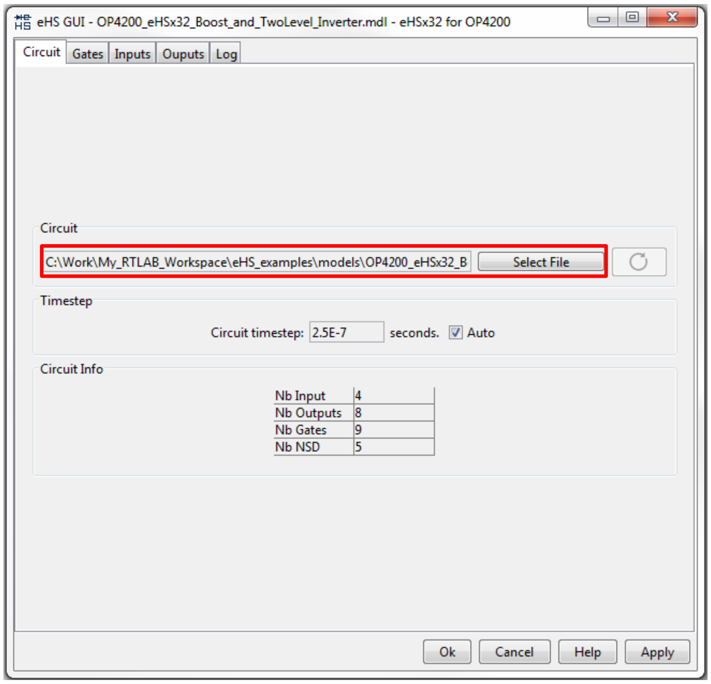


Figure 10: eHS CPU block diagram for OP4200

The user must provide a circuit file (SimPowerSystems Simulink model, PSIM file, PLECS Simulink model, or NI Multisim 13 XML netlist) to declare the circuit that will be simulated inside the eHS core. To declare a specific circuit file, double-click on the **eHSx32 for OP4200** block to open the graphical user interface. In the **Circuit** tab, a circuit file can be selected and parsed. In this example, the circuit file is a SimPowerSystems Simulink model entitled *Boost_and_2LvInverter_SPS.mdl.*



eHS GUI - OP4200_eHSx32_Boost_and_TwoLevel_Inverter.mdl - eHSx32 for OP4200

Circuit

C:\Work\My_RTLAB_Workspace\eHS_examples\models\OP4200_eHSx32_B

Timestep

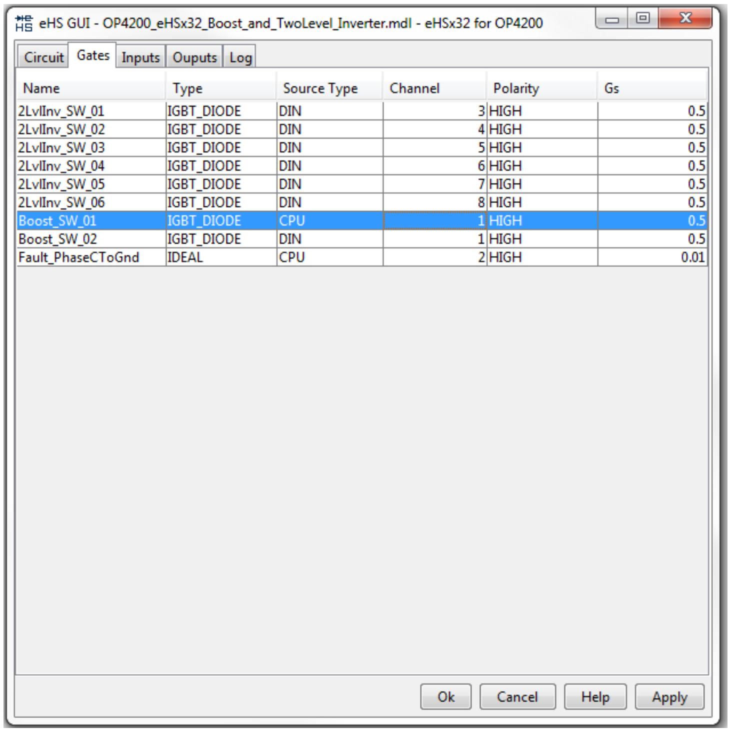
Circuit timestep: 2.5E-7 seconds. ☑ Auto

Circuit Info

Nb Input	4
Nb Outputs	8
Nb Gates	9
Nb NSD	5

Figure 11: Selecting a model in the *Circuit* tab

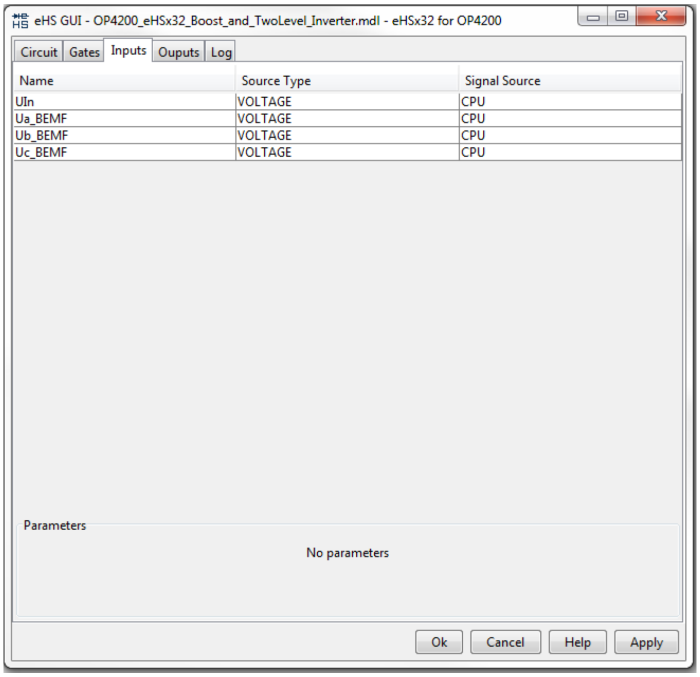
Navigating to the **Gates** tab will display and list the switches that are present in the declared circuit file. The **Gates** tab allows the user to configure the Switch Source Type & Channel for each switch (either from **Digital In** channel or from the **CPU**[9]), the polarity of each switch (**High** or **Low**), and the Switch Conductance (Gs) value of each switch.


eHS GUI - OP4200_eHSx32_Boost_and_TwoLevel_Inverter.mdl - eHSx32 for OP4200
Circuit
Gates
Inputs
Ouputs
Log
Ok
Cancel
Help
Apply

Name	Type	Source Type	Channel	Polarity	Gs
2LvlInv_SW_01	IGBT_DIODE	DIN	3	HIGH	0.5
2LvlInv_SW_02	IGBT_DIODE	DIN	4	HIGH	0.5
2LvlInv_SW_03	IGBT_DIODE	DIN	5	HIGH	0.5
2LvlInv_SW_04	IGBT_DIODE	DIN	6	HIGH	0.5
2LvlInv_SW_05	IGBT_DIODE	DIN	7	HIGH	0.5
2LvlInv_SW_06	IGBT_DIODE	DIN	8	HIGH	0.5
Boost_SW_01	IGBT_DIODE	CPU	1	HIGH	0.5
Boost_SW_02	IGBT_DIODE	DIN	1	HIGH	0.5
Fault_PhaseCToGnd	IDEAL	CPU	2	HIGH	0.01

Figure 12: *Gates* tab

Navigating to the **Inputs** tab will display and list the circuit inputs that are present in the declared circuit file. The **Inputs** tab allows the user to configure the different Current and Voltage Sources used in the circuit model as well as the parameters (used to configure the value of the **Constant**) of each input. By selecting **CPU**, the block regeneration will expose this input as an input port to the block.


eHS GUI - OP4200_eHSx32_Boost_and_TwoLevel_Inverter.mdl - eHSx32 for OP4200
Circuit
Gates
Inputs
Ouputs
Log
Ok
Cancel
Help
Apply

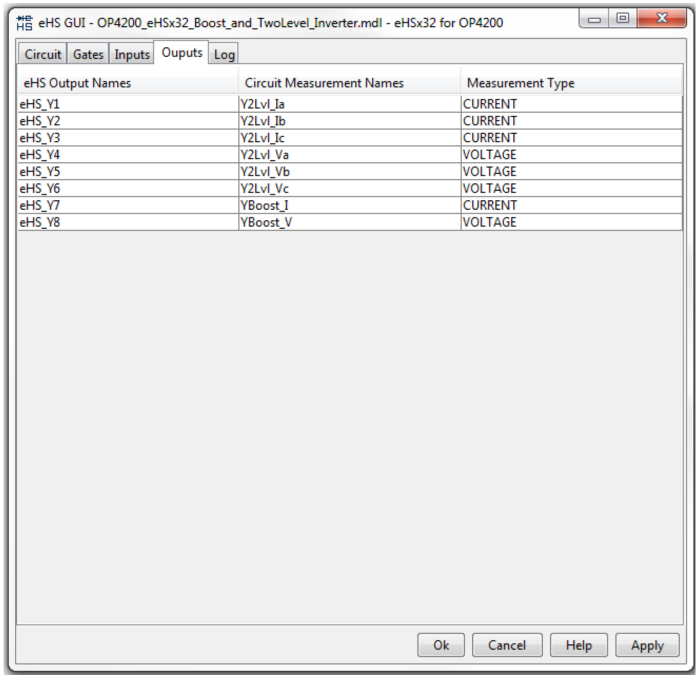
Name	Source Type	Signal Source
UIn	VOLTAGE	CPU
Ua_BEMF	VOLTAGE	CPU
Ub_BEMF	VOLTAGE	CPU
Uc_BEMF	VOLTAGE	CPU

Parameters

No parameters

Figure 13: *Inputs* tab

Navigating to the **Outputs** tab will display and list the circuit outputs that are present in the declared circuit file. The **Outputs** tab allows the user to view the Measurement points configured within the circuit model.


eHS GUI - OP4200_eHSx32_Boost_and_TwoLevel_Inverter.mdl - eHSx32 for OP4200
Circuit
Gates
Inputs
Ouputs
Log
Ok
Cancel
Help
Apply

eHS Output Names	Circuit Measurement Names	Measurement Type
eHS_Y1	Y2Lvl_Ia	CURRENT
eHS_Y2	Y2Lvl_Ib	CURRENT
eHS_Y3	Y2Lvl_Ic	CURRENT
eHS_Y4	Y2Lvl_Va	VOLTAGE
eHS_Y5	Y2Lvl_Vb	VOLTAGE
eHS_Y6	Y2Lvl_Vc	VOLTAGE
eHS_Y7	YBoost_I	CURRENT
eHS_Y8	YBoost_V	VOLTAGE

Figure 14: *Outputs* tab

CIRCUIT FILE

The circuit file models a boost and a two-level inverter. The eHS solver will extract the components netlist from this file and calculate the system equations before running them on the FPGA board.

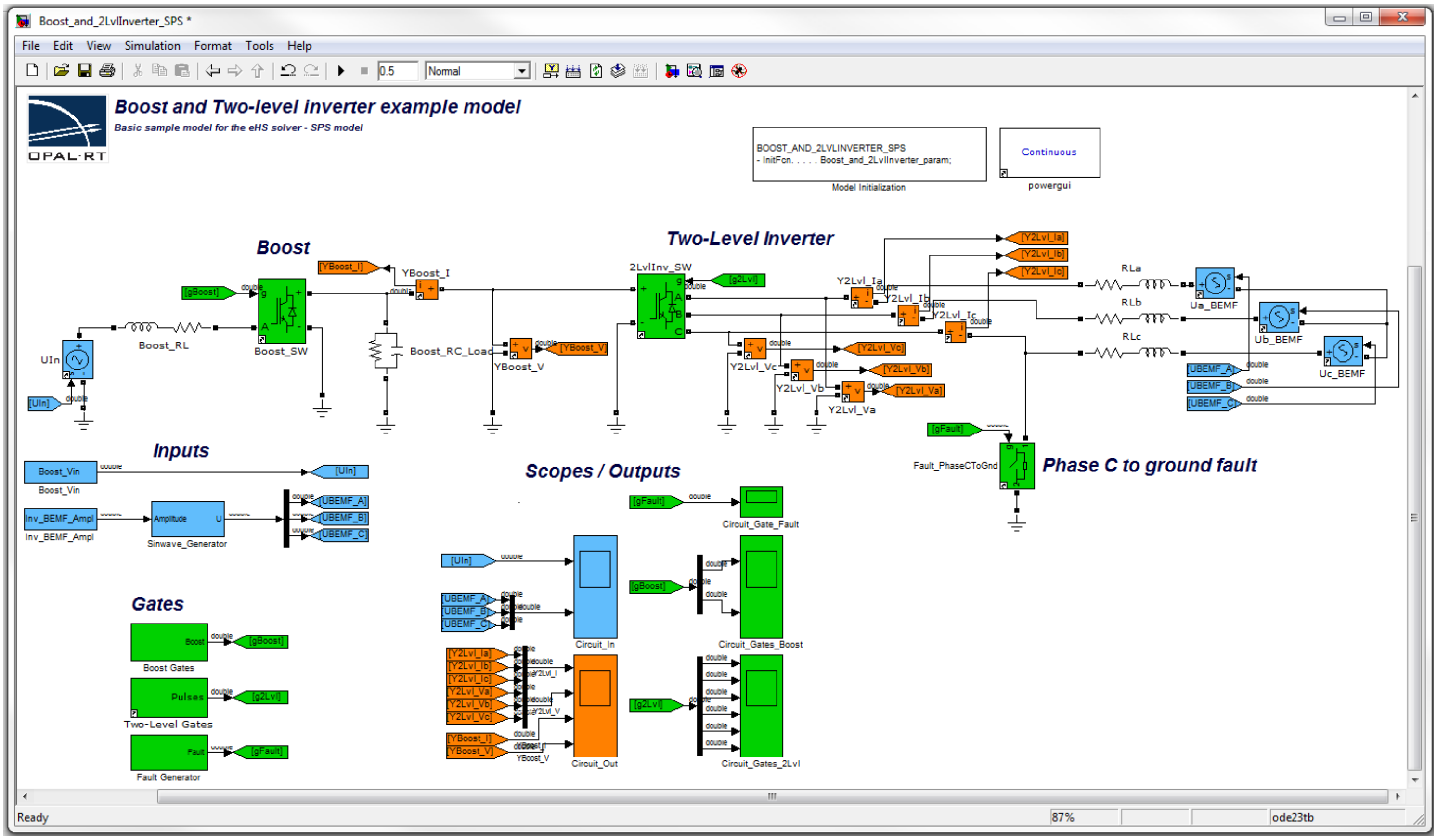


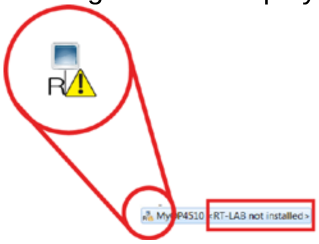
Figure 15: Circuit model diagram

VALIDATING THE MODEL

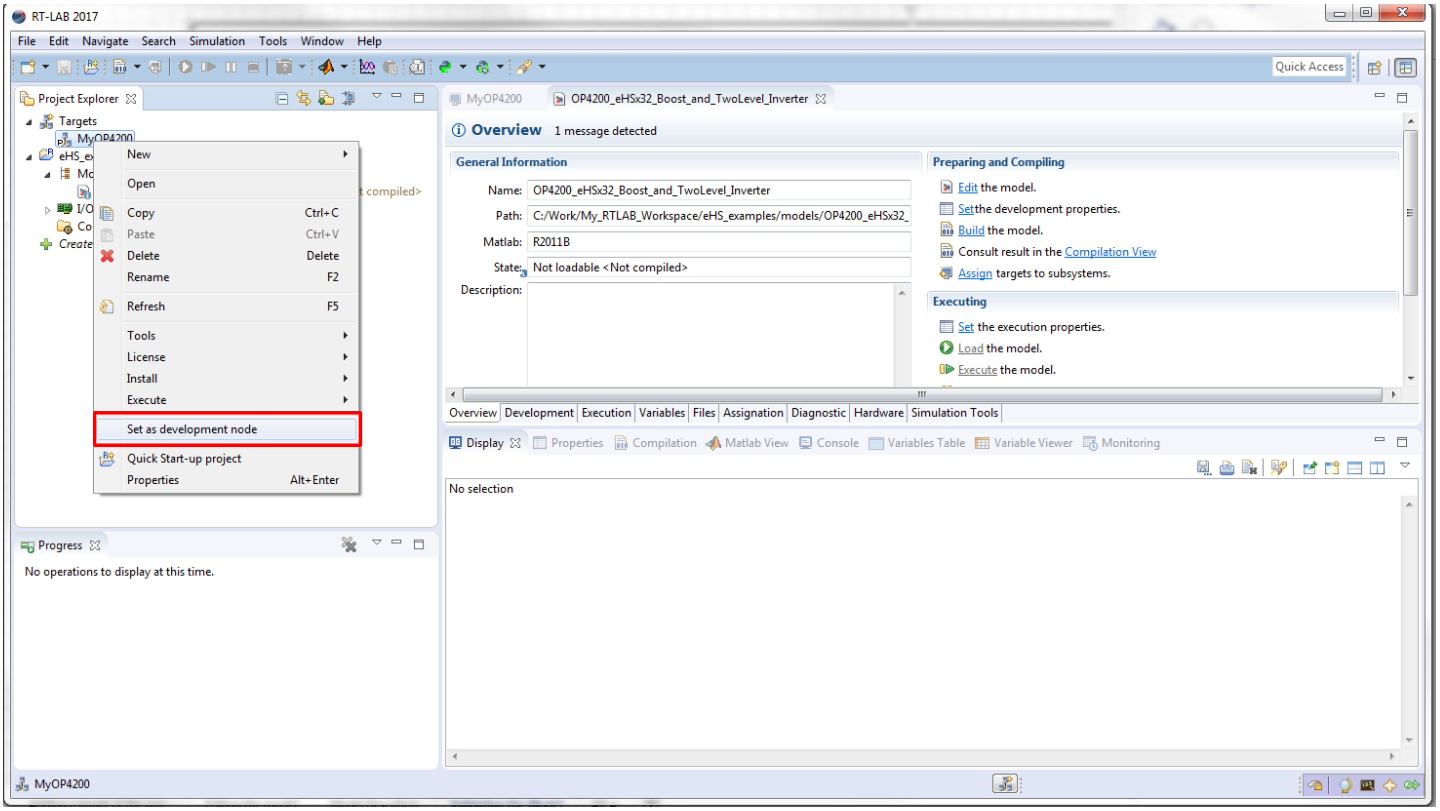
Run an offline simulation from the RT-LAB model to ensure that all the library links are resolved using your MATLAB instance.

Verify that the model runs properly.

- In RT-LAB, check that your target is available in the Targets list, and is up and running (any problems with the target will be displayed by RT-LAB as an icon change and added text after the target name).



- Right-click your target and select **Set as development node**.


RT-LAB 2017
File Edit Navigate Search Simulation Tools Window Help
Quick Access
Project Explorer
Targets
MyOP4200
eHS_e
Mc
I/O
Co
Create
compiled>
New
Open
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Rename F2
Refresh F5
Tools
License
Install
Execute
Set as development node
Quick Start-up project
Properties Alt+Enter
Progress
MyOP4200
OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Overview
Overview Development Execution Variables Files Assignation Diagnostic Hardware Simulation Tools
Display Properties Compilation Matlab View Console Variables Table Variable Viewer Monitoring
MyOP4200

Overview 1 message detected

General Information

Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_
Matlab: R2011B
State: Not loadable <Not compiled>
Description:

Preparing and Compiling

Edit the model.
Setthe development properties.
Build the model.
Consult result in the Compilation View
Assign targets to subsystems.

Executing

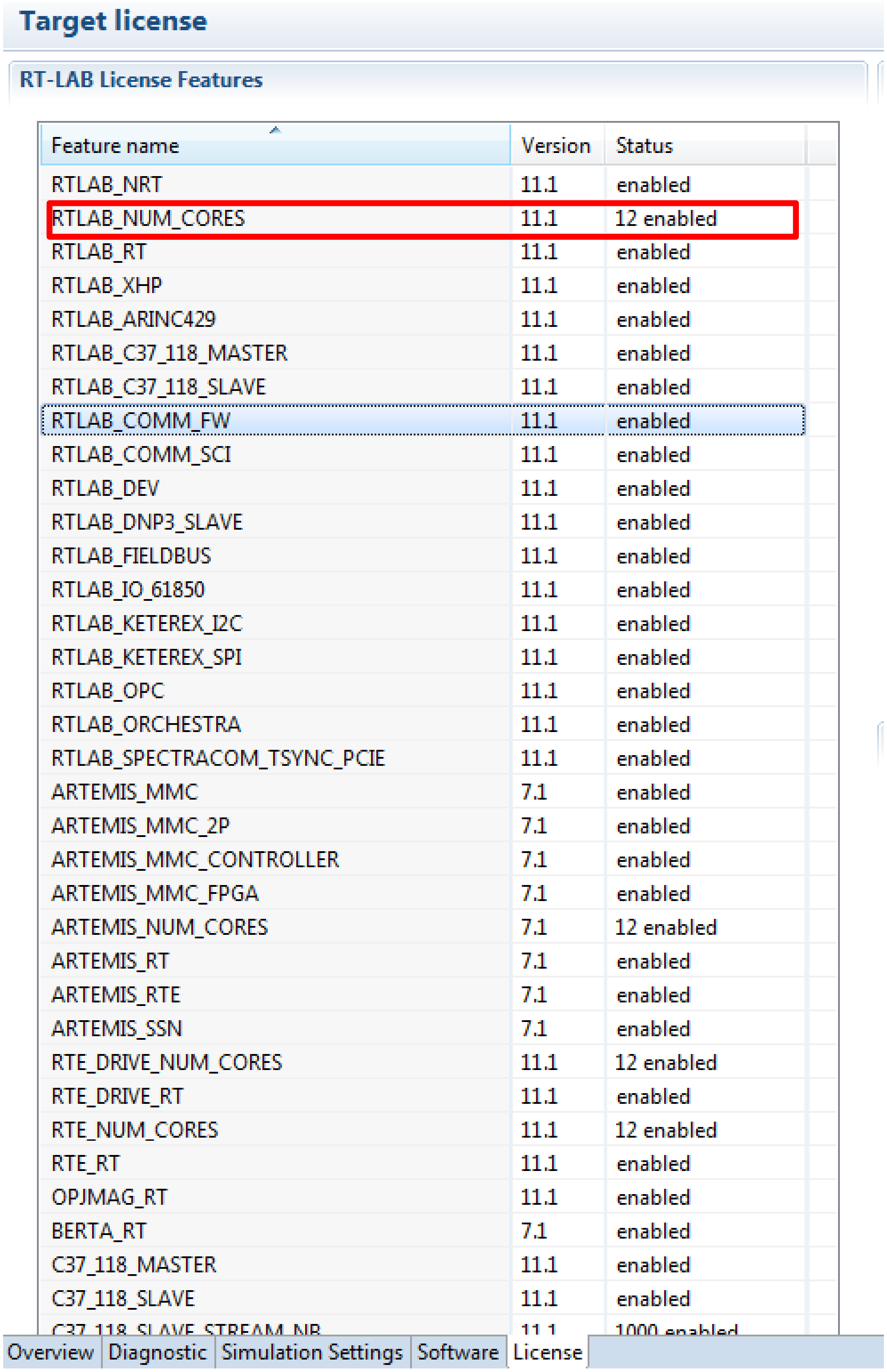
Set the execution properties.
Load the model.
Execute the model.

No operations to display at this time.

No selection

Figure 16: Setting the target as development node

- Click on the **License** tab of the target to display the list of software licenses and make sure that XSG_EHS, RTLAB_RT, RTLAB_DEV, RTLAB_XHP licenses are enabled. You must also have at least RTLAB_NUM_CORES >= 1.



Target license

RT-LAB License Features

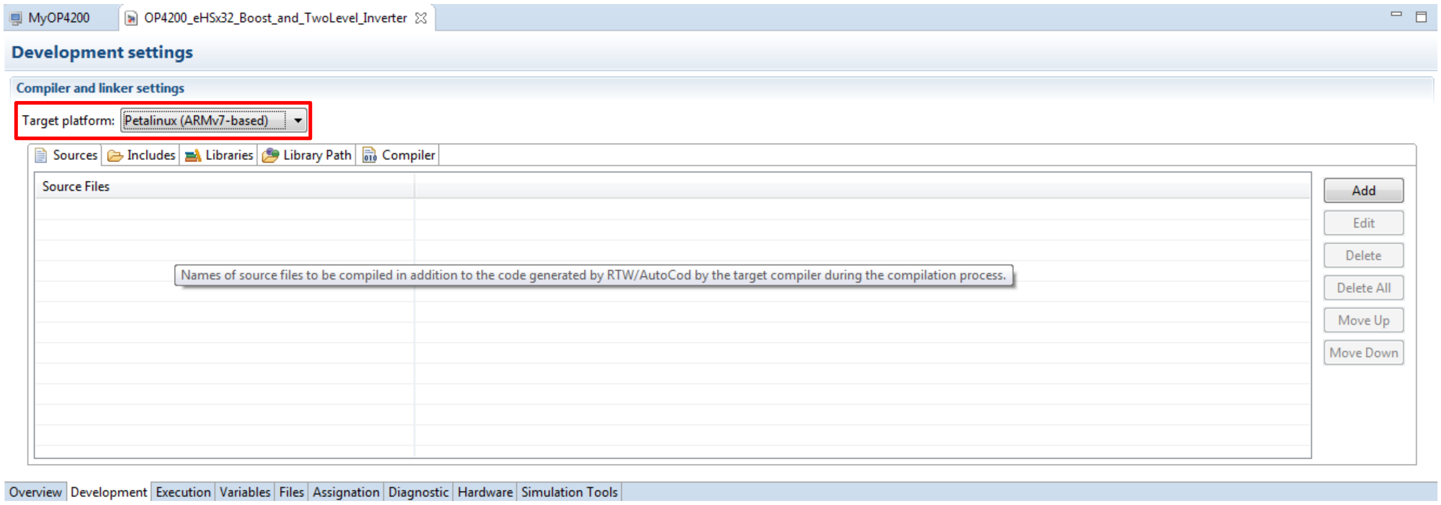
Feature name	Version	Status
RTLAB_NRT	11.1	enabled
RTLAB_NUM_CORES	11.1	12 enabled
RTLAB_RT	11.1	enabled
RTLAB_XHP	11.1	enabled
RTLAB_ARINC429	11.1	enabled
RTLAB_C37_118_MASTER	11.1	enabled
RTLAB_C37_118_SLAVE	11.1	enabled
RTLAB_COMM_FW	11.1	enabled
RTLAB_COMM_SCI	11.1	enabled
RTLAB_DEV	11.1	enabled
RTLAB_DNP3_SLAVE	11.1	enabled
RTLAB_FIELDBUS	11.1	enabled
RTLAB_IO_61850	11.1	enabled
RTLAB_KETEREX_I2C	11.1	enabled
RTLAB_KETEREX_SPI	11.1	enabled
RTLAB_OPC	11.1	enabled
RTLAB_ORCHESTRA	11.1	enabled
RTLAB_SPECTRACOM_TSYNC_PCIE	11.1	enabled
ARTEMIS_MMC	7.1	enabled
ARTEMIS_MMC_2P	7.1	enabled
ARTEMIS_MMC_CONTROLLER	7.1	enabled
ARTEMIS_MMC_FPGA	7.1	enabled
ARTEMIS_NUM_CORES	7.1	12 enabled
ARTEMIS_RT	7.1	enabled
ARTEMIS_RTE	7.1	enabled
ARTEMIS_SSN	7.1	enabled
RTE_DRIVE_NUM_CORES	11.1	12 enabled
RTE_DRIVE_RT	11.1	enabled
RTE_NUM_CORES	11.1	12 enabled
RTE_RT	11.1	enabled
OPJMAG_RT	11.1	enabled
BERTA_RT	7.1	enabled
C37_118_MASTER	11.1	enabled
C37_118_SLAVE	11.1	enabled
C37_118_SLAVE_STREAM_NB	11.1	1000 enabled

Figure 17: Example for how to verify that all licenses are present

BUILDING THE MODEL

Follow the steps detailed below to build a model.

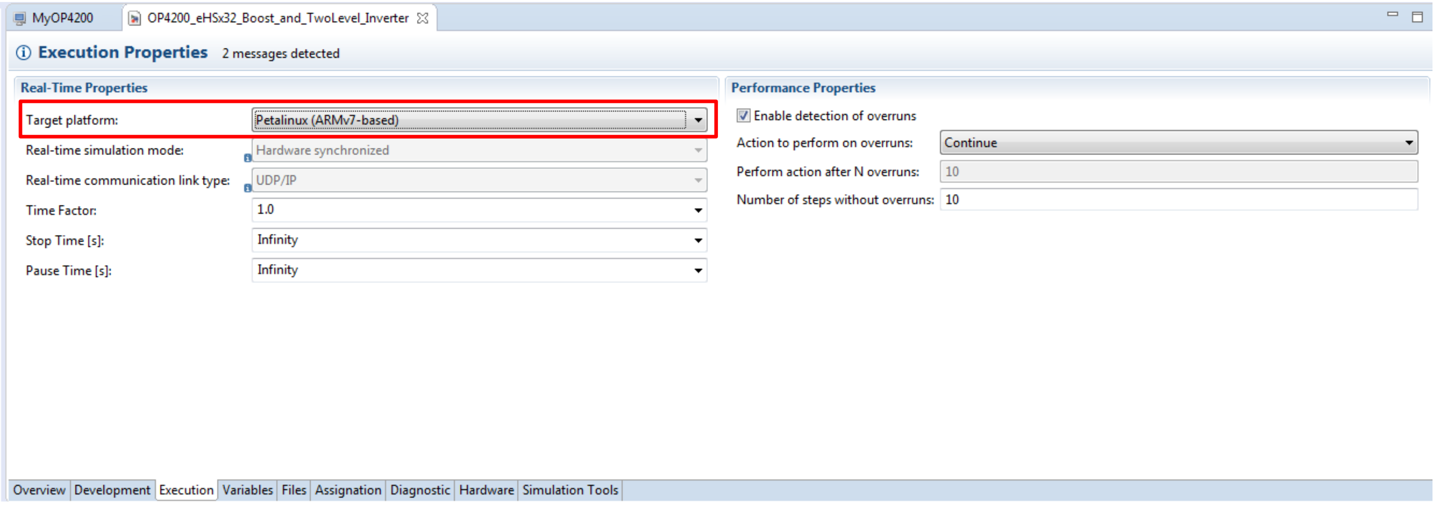
1. In the **Development** and **Execution** tabs, make sure that the target platform is set correctly. For this example, the target platform must be set to **Petalinux (ARMv7-based)**.



Target platform: Petalinux (ARMv7-based)

Names of source files to be compiled in addition to the code generated by RTW/AutoCod by the target compiler during the compilation process.

Figure 18: Setting the Target platform



2 messages detected

Target platform: Petalinux (ARMv7-based)

Real-time simulation mode: Hardware synchronized

Real-time communication link type: UDP/IP

Time Factor: 1.0

Stop Time [s]: Infinity

Pause Time [s]: Infinity

☑ Enable detection of overruns

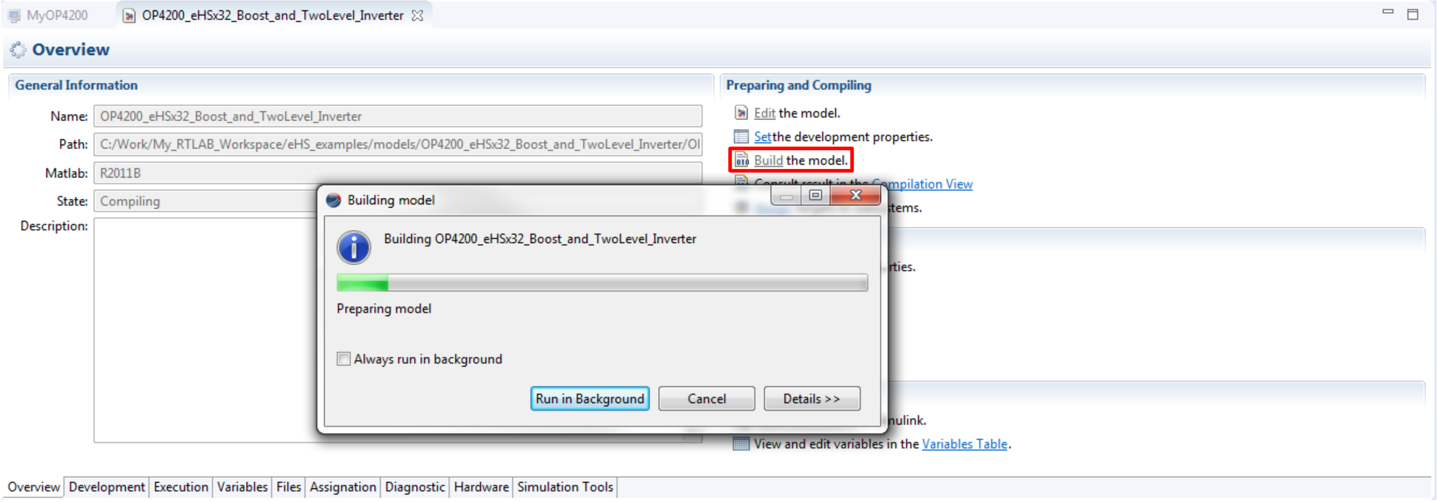
Action to perform on overruns: Continue

Perform action after N overruns: 10

Number of steps without overruns: 10

Figure 19: Setting the Target platform

2. In the *Preparing and Compiling* window, click **Build the model**. The *Building Model* window appears.
3. Verify that the model was successfully built by clicking **Consult result in Compilation View**.



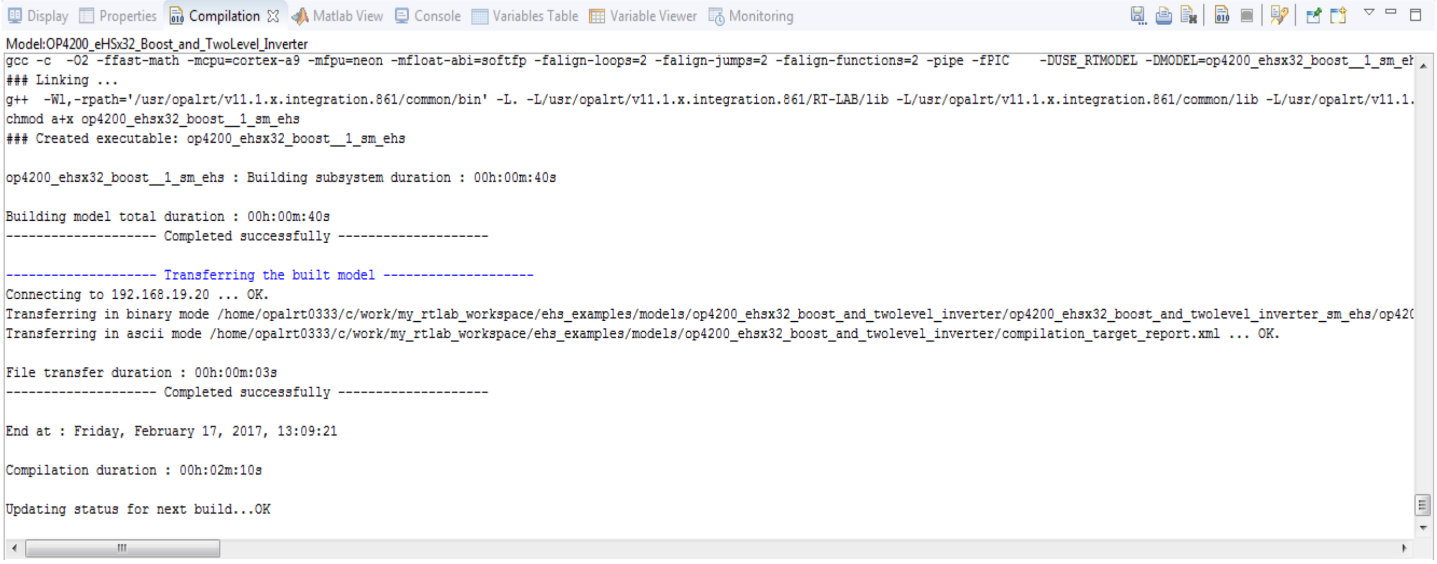
Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_Boost_and_TwoLevel_Inverter/OI
Matlab: R2011B
State: Compiling
Description:

Building OP4200_eHSx32_Boost_and_TwoLevel_Inverter

Preparing model

☐ Always run in background

Figure 20: Building the model



Model:OP4200_eHSx32_Boost_and_TwoLevel_Inverter

```
gcc -c  -O2 -ffast-math -mcpu=cortex-a9 -mfpu=neon -mfloat-abi=softfp -falign-loops=2 -falign-jumps=2 -falign-functions=2 -pipe -fPIC    -DUSE_RTMODEL -DMODEL=op4200_ehsx32_boost__1_sm_eh
### Linking ...
g++  -Wl,-rpath='/usr/opalrt/v11.1.x.integration.861/common/bin' -L. -L/usr/opalrt/v11.1.x.integration.861/RT-LAB/lib -L/usr/opalrt/v11.1.x.integration.861/common/lib -L/usr/opalrt/v11.1.
chmod a+x op4200_ehsx32_boost__1_sm_ehs
### Created executable: op4200_ehsx32_boost__1_sm_ehs

op4200_ehsx32_boost__1_sm_ehs : Building subsystem duration : 00h:00m:40s

Building model total duration : 00h:00m:40s
-------------------- Completed successfully --------------------

-------------------- Transferring the built model --------------------
Connecting to 192.168.19.20 ... OK.
Transferring in binary mode /home/opalrt0333/c/work/my_rtlab_workspace/ehs_examples/models/op4200_ehsx32_boost_and_twolevel_inverter/op4200_ehsx32_boost_and_twolevel_inverter_sm_ehs/op420
Transferring in ascii mode /home/opalrt0333/c/work/my_rtlab_workspace/ehs_examples/models/op4200_ehsx32_boost_and_twolevel_inverter/compilation_target_report.xml ... OK.

File transfer duration : 00h:00m:03s
-------------------- Completed successfully --------------------

End at : Friday, February 17, 2017, 13:09:21

Compilation duration : 00h:02m:10s

Updating status for next build...OK
```

Figure 21: Compilation view

CREATING AND CONFIGURING AN OPAL-RT BOARD

To route physical IO (AI, AO, DI) to the eHS core during real-time simulation, an *OPAL-RT Board* must be created, configured, and assigned to an associated subsystem (i.e RT-LAB Model). If an OPAL-RT Board is not created and configured, the eHS running on the OP4200 will not be able to interact with physical IO. Follow the steps below to configure an *OPAL-RT Board* interface.

1. To create a new *OPAL-RT Board*, right-click on the **I/Os** item and select **New>New I/O**. In the *Add New I/O* window, select **OPAL-RT Board** and provide an *I/O name*. Click **Finish**.

> **Note:** The Boost and Two-Level example already contains a pre-configured OPAL-RT board so there is no need to create a new one.

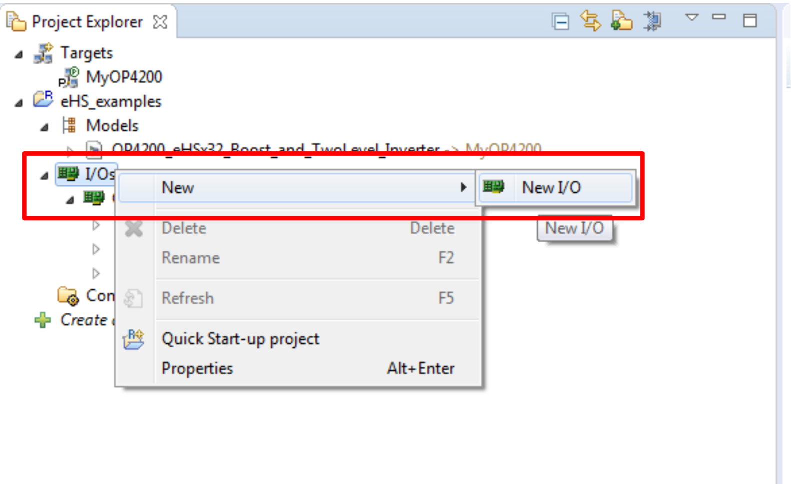
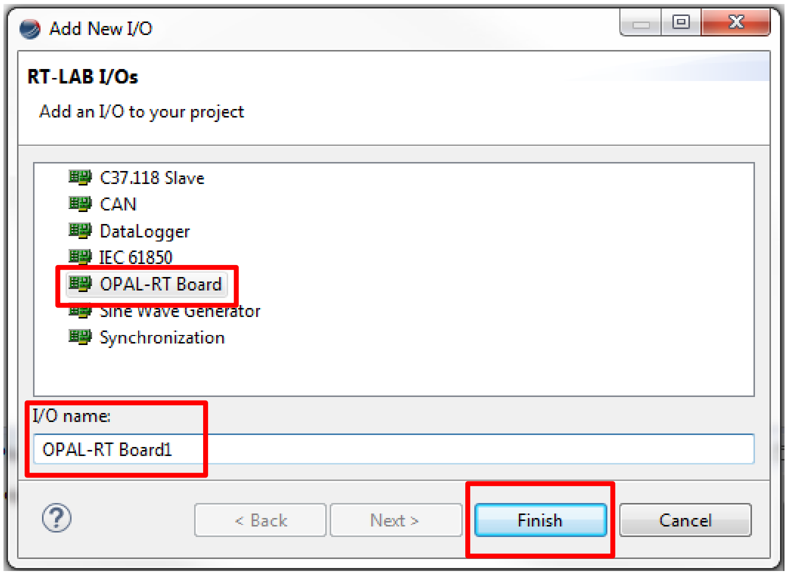


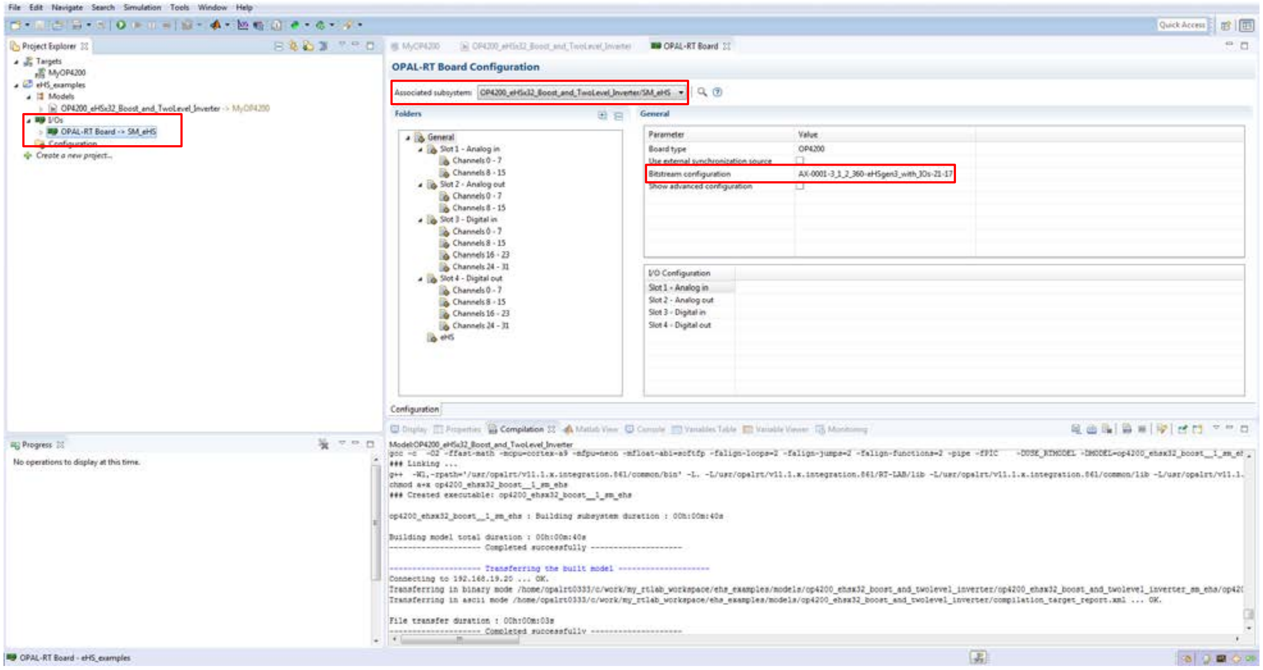
Figure 22: New I/O



I/O name: OPAL-RT Board1

Figure 23: New OPAL-RT Board

2. Double-click on the **OPAL-RT Board** to open the *OPAL-RT Board Configuration* window. Set the *Associated subsystem* to the **OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS.** In addition, select the appropriate *Bitstream configuration* from the drop-down menu.



OPAL-RT Board Configuration

Associated subsystem: OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS

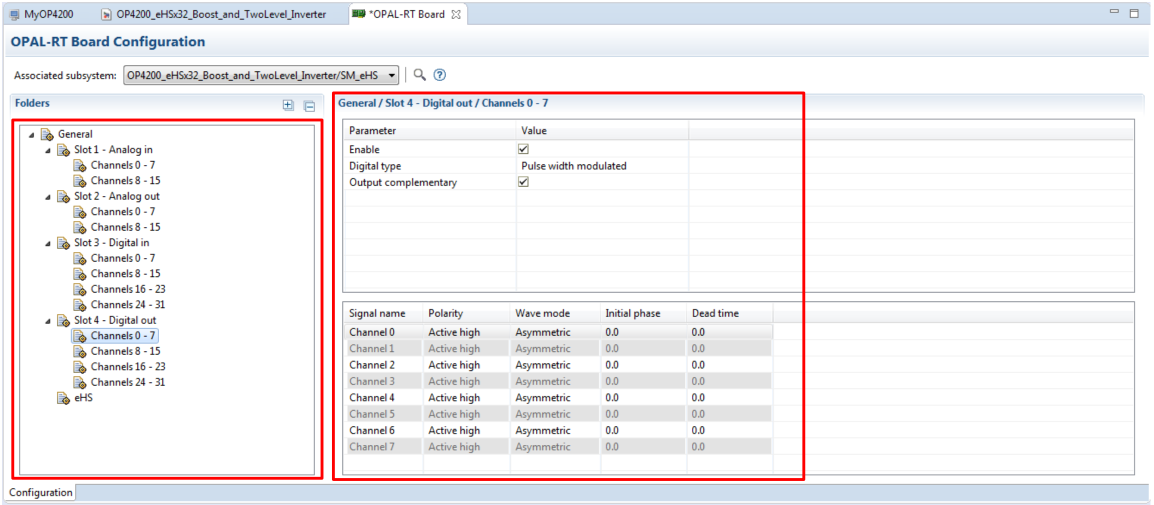
General

Parameter	Value
Board type	OP4200
Use external synchronization source	□
Bitstream configuration	AX-0001-3_1_2_360-eHSgen3_with_IOs-21-17
Show advanced configuration	□

```
Model:OP4200_eHSx32_Boost_and_TwoLevel_Inverter
gcc -c  -O2 -ffast-math -mcpu=cortex-a9 -mfpu=neon -mfloat-abi=softfp -falign-loops=2 -falign-jumps=2 -falign-functions=2 -pipe -fPIC    -DUSE_RTMODEL -DMODEL=op4200_ehsx32_boost__1_sm_eh
### Linking ...
g++  -Wl,-rpath='/usr/opalrt/v11.1.x.integration.861/common/bin' -L. -L/usr/opalrt/v11.1.x.integration.861/RT-LAB/lib -L/usr/opalrt/v11.1.x.integration.861/common/lib -L/usr/opalrt/v11.1.
chmod a+x op4200_ehsx32_boost__1_sm_ehs
### Created executable: op4200_ehsx32_boost__1_sm_ehs

op4200_ehsx32_boost__1_sm_ehs : Building subsystem duration : 00h:00m:40s

Building model total duration : 00h:00m:40s
-------------------- Completed successfully --------------------

-------------------- Transferring the built model --------------------
Connecting to 192.168.19.20 ... OK.
Transferring in binary mode /home/opalrt0333/c/work/my_rtlab_workspace/ehs_examples/models/op4200_ehsx32_boost_and_twolevel_inverter/op4200_ehsx32_boost_and_twolevel_inverter_sm_ehs/op42
Transferring in ascii mode /home/opalrt0333/c/work/my_rtlab_workspace/ehs_examples/models/op4200_ehsx32_boost_and_twolevel_inverter/compilation_target_report.xml ... OK.

File transfer duration : 00h:00m:03s
-------------------- Completed successfully --------------------
```

Figure 24: Configure the OPAL-RT Board

3. With the above step complete, all the I/O supported by the firmware (defined by the *Bitstream configuration*) will be displayed to the left of the *General* tab. Each channel in each type of I/O board can be configured to meet the simulation requirements. For this example, the I/O has been pre-configured to meet the simulation requirements of the declared circuit model (*Boost_and_2LvInverter_SPS.mdl*).



OPAL-RT Board Configuration

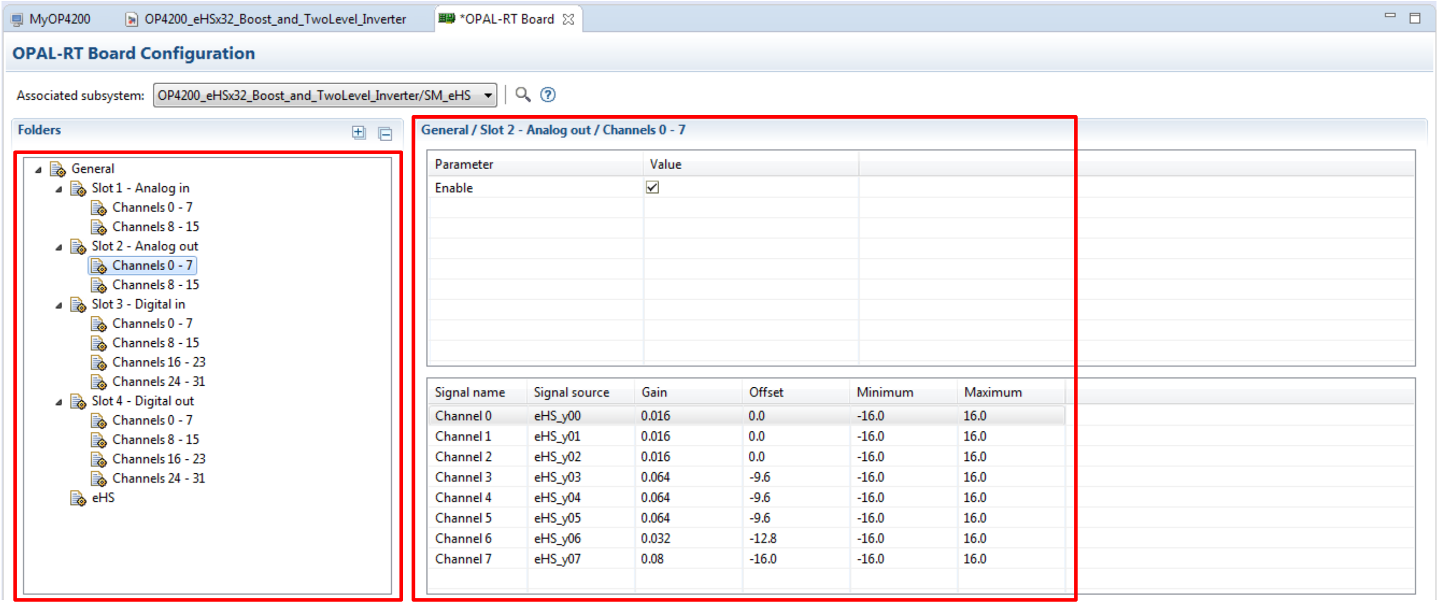
Associated subsystem: OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS

General / Slot 4 - Digital out / Channels 0 - 7

Parameter	Value
Enable	☑
Digital type	Pulse width modulated
Output complementary	☑

Signal name	Polarity	Wave mode	Initial phase	Dead time
Channel 0	Active high	Asymmetric	0.0	0.0
Channel 1	Active high	Asymmetric	0.0	0.0
Channel 2	Active high	Asymmetric	0.0	0.0
Channel 3	Active high	Asymmetric	0.0	0.0
Channel 4	Active high	Asymmetric	0.0	0.0
Channel 5	Active high	Asymmetric	0.0	0.0
Channel 6	Active high	Asymmetric	0.0	0.0
Channel 7	Active high	Asymmetric	0.0	0.0

Figure 25: Configure DO as PWM using the OPAL-RT Board



OPAL-RT Board Configuration

Associated subsystem: OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS

General / Slot 2 - Analog out / Channels 0 - 7

Parameter	Value
Enable	☑

Signal name	Signal source	Gain	Offset	Minimum	Maximum
Channel 0	eHS_y00	0.016	0.0	-16.0	16.0
Channel 1	eHS_y01	0.016	0.0	-16.0	16.0
Channel 2	eHS_y02	0.016	0.0	-16.0	16.0
Channel 3	eHS_y03	0.064	-9.6	-16.0	16.0
Channel 4	eHS_y04	0.064	-9.6	-16.0	16.0
Channel 5	eHS_y05	0.064	-9.6	-16.0	16.0
Channel 6	eHS_y06	0.032	-12.8	-16.0	16.0
Channel 7	eHS_y07	0.08	-16.0	-16.0	16.0

Figure 26: Configure advanced analog output using the OPAL-RT Board

4. When all the I/O parameters of the simulation have been configured, save the configuration using the save button.

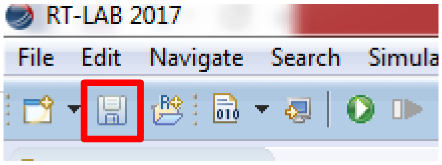
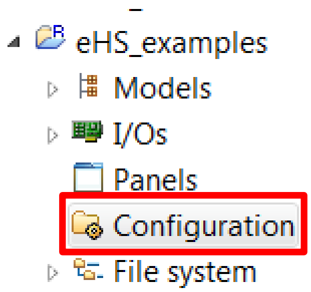


Figure 27: Save OPAL-RT Board configuration

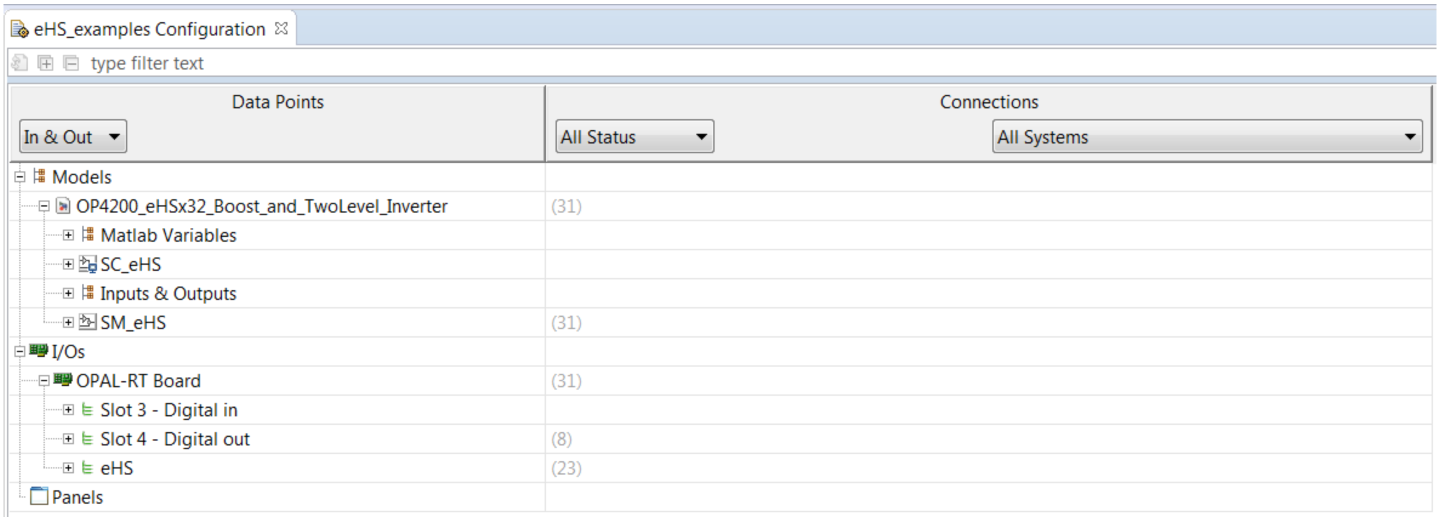
5. After changing and saving the OPAL-RT Board configuration, it is necessary to recompile the CPU model by selecting **Build the model** from the *Preparing and Compiling* window. This allows RT-LAB to automatically detect the connections made between the model and the OPAL-RT Board.

6. Configure the links between OPAL-RT Board and the CPU Model. To achieve this task, double-click on **Configuration** from the Project Explorer tree.



- eHS_examples
  - Models
  - I/Os
  - Panels
  - Configuration
  - File system

Figure 28: Open OPAL-RT Board connections



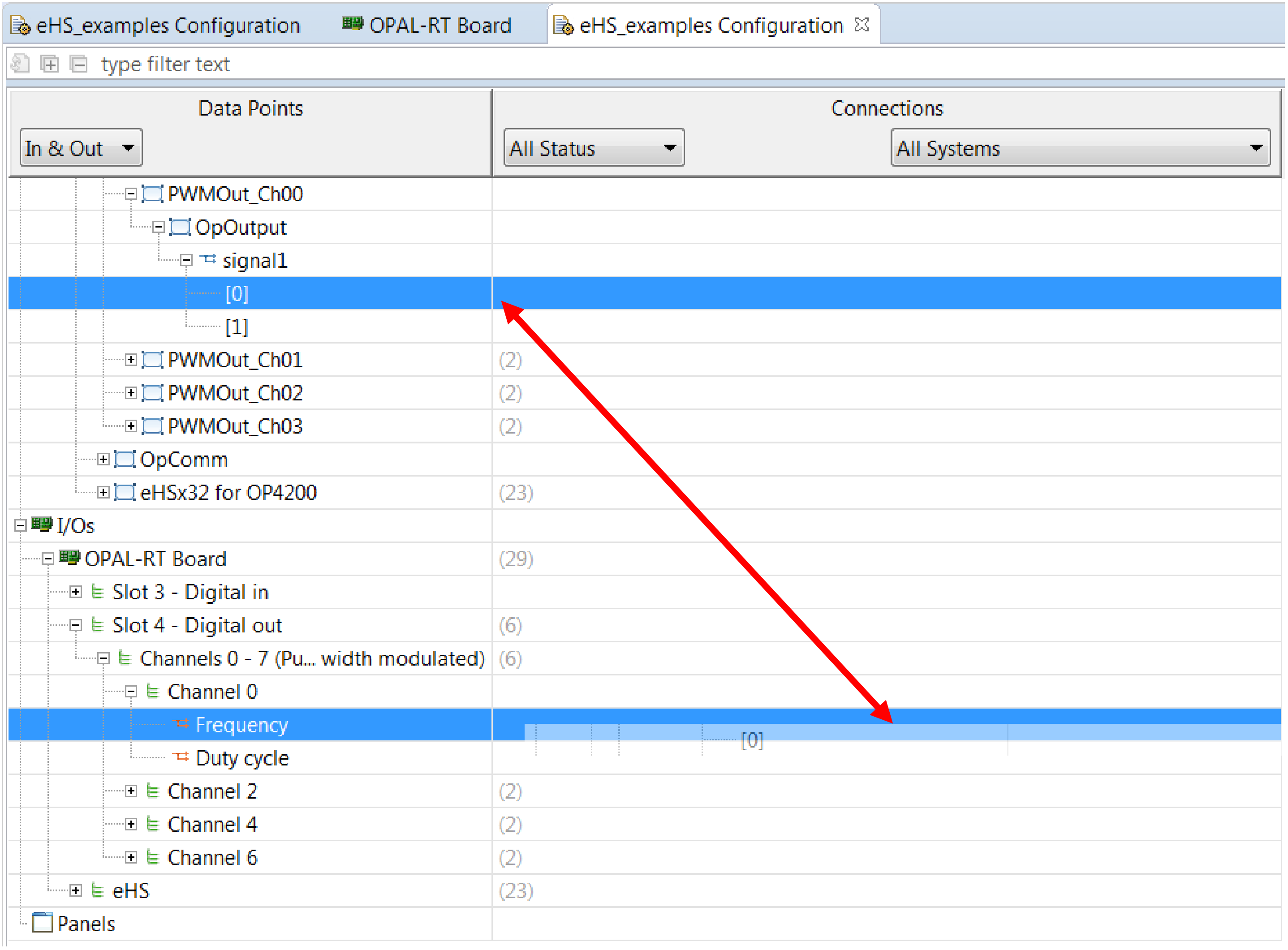
eHS_examples Configuration

type filter text

Data Points	Connections
In & Out	All Status · All Systems
Models	
OP4200_eHSx32_Boost_and_TwoLevel_Inverter	(31)
Matlab Variables	
SC_eHS	
Inputs & Outputs	
SM_eHS	(31)
I/Os	
OPAL-RT Board	(31)
Slot 3 - Digital in	
Slot 4 - Digital out	(8)
eHS	(23)
Panels	

Figure 29: OPAL RT Board connections panel

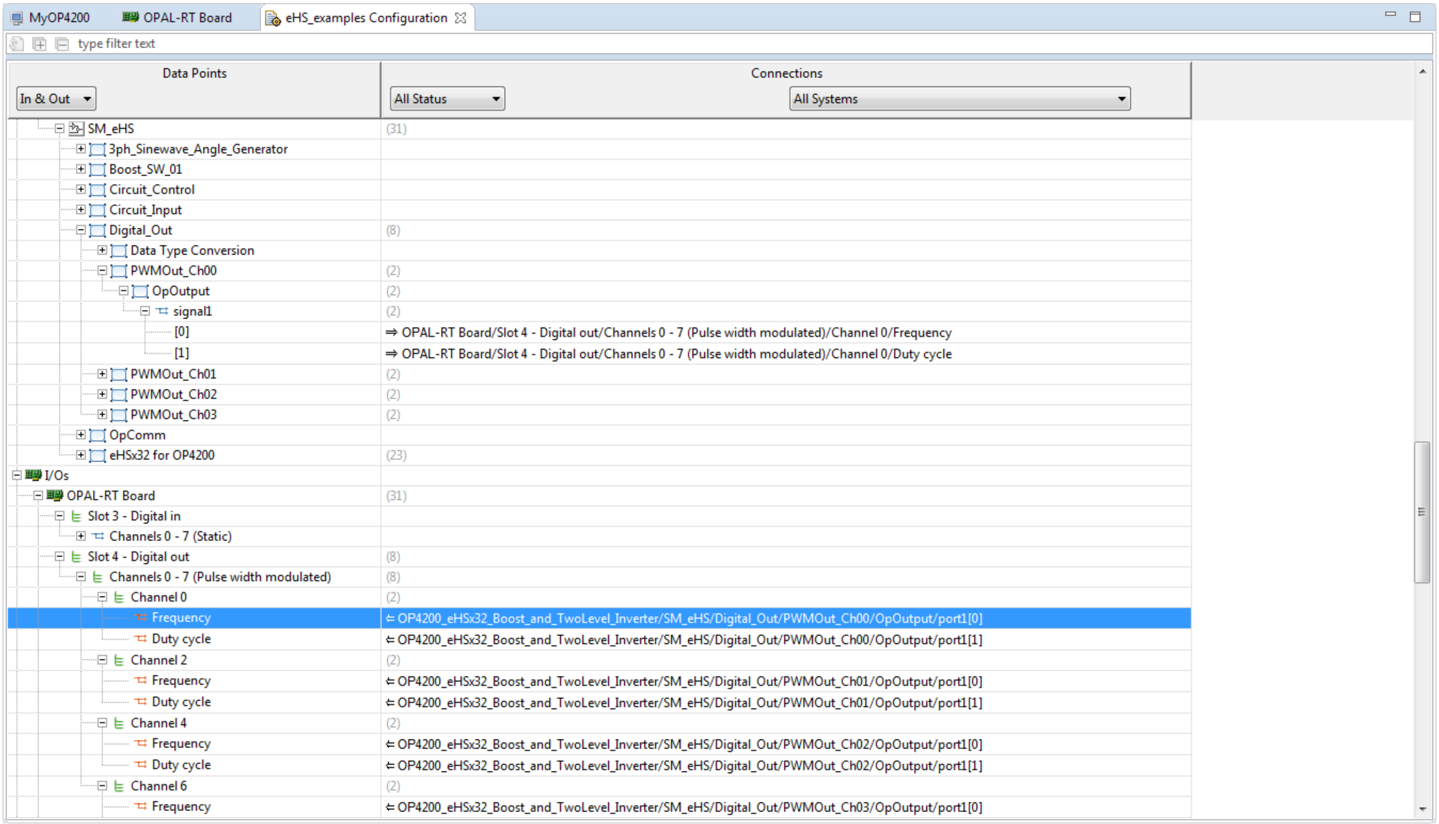
7. The displayed panel allows for connections to be made between the CPU Model and the IO data points (i.e. it provides links between the Real-Time controller and the FPGA). For instance, to create a connection between the PWM Out defined in the OPAL-RT Board and the Model, simply drag and drop the **Frequency** and **Duty Cycle** from Channel 0 of the OPAL-RT Board tree to the desired signal inputs of the eHS tree.



Data Points	Connections
In & Out	All Status All Systems
PWMOut_Ch00	
OpOutput	
signal1	
[0]	
[1]	
PWMOut_Ch01	(2)
PWMOut_Ch02	(2)
PWMOut_Ch03	(2)
OpComm	
eHSx32 for OP4200	(23)
I/Os	
OPAL-RT Board	(29)
Slot 3 - Digital in	
Slot 4 - Digital out	(6)
Channels 0 - 7 (Pu... width modulated)	(6)
Channel 0	
Frequency	[0]
Duty cycle	
Channel 2	(2)
Channel 4	(2)
Channel 6	(2)
eHS	(23)
Panels	

Figure 30: Connection example for PWM within OPAL RT Board and Model

Note: In this particular example, the switches in the simulated circuit model are driven by the internal PWM generators available in the FPGA firmware. To route these PWM signals to the inputs of the eHS solver, a DIO loopback connection should be made between the DO and DI cassettes. Next, the *OPAL-Board Configuration* tab can be used to easily route the internal PWM signals to the digital outputs as demonstrated in the figure below.



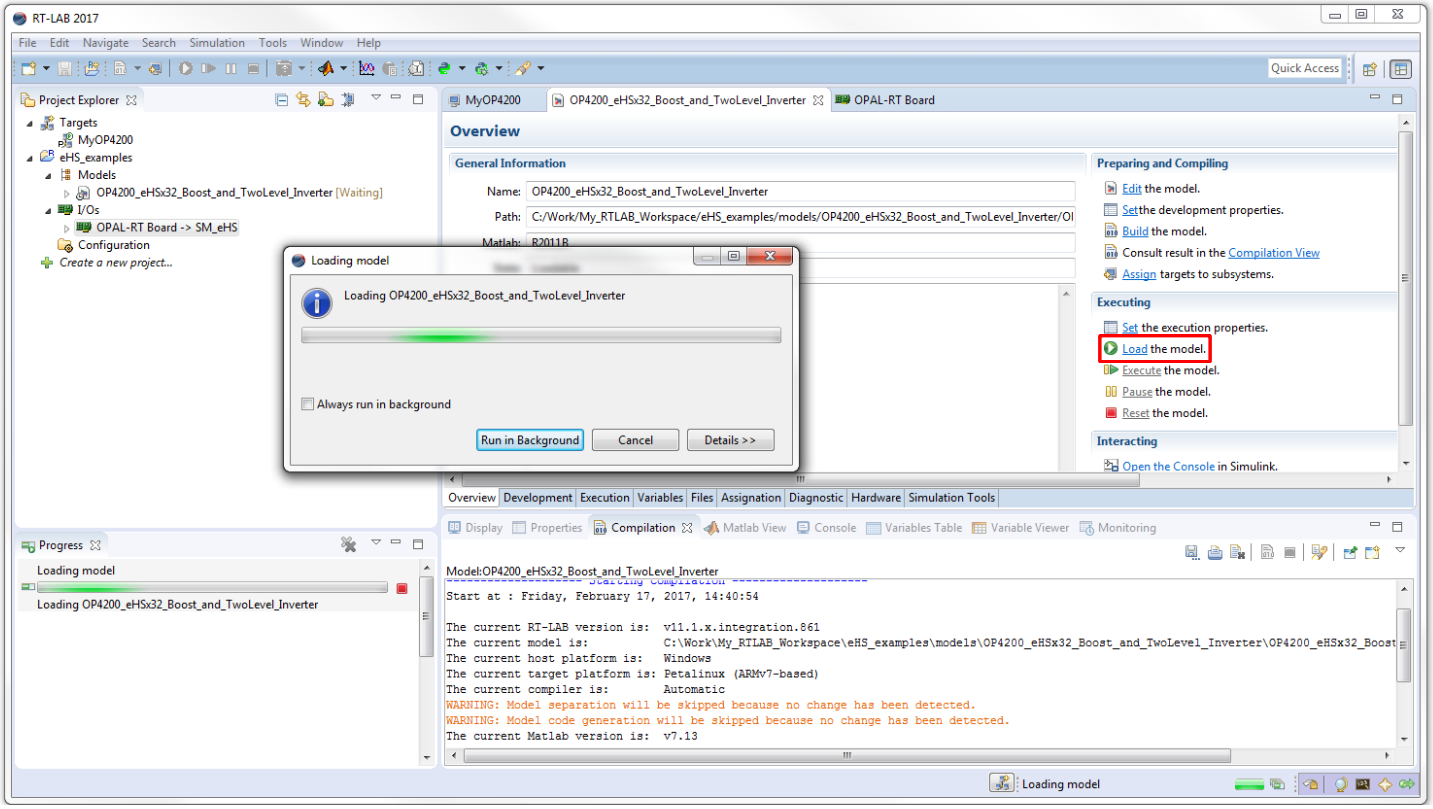
Data Points	Connections
SM_eHS	(31)
3ph_Sinewave_Angle_Generator	
Boost_SW_01	
Circuit_Control	
Circuit_Input	
Digital_Out	(8)
Data Type Conversion	
PWMOut_Ch00	(2)
OpOutput	(2)
signal1	(2)
[0]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 0 - 7 (Pulse width modulated)/Channel 0/Frequency
[1]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 0 - 7 (Pulse width modulated)/Channel 0/Duty cycle
PWMOut_Ch01	(2)
PWMOut_Ch02	(2)
PWMOut_Ch03	(2)
OpComm	
eHSx32 for OP4200	(23)
I/Os	
OPAL-RT Board	(31)
Slot 3 - Digital in	
Channels 0 - 7 (Static)	
Slot 4 - Digital out	(8)
Channels 0 - 7 (Pulse width modulated)	(8)
Channel 0	(2)
Frequency	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch00/OpOutput/port1[0]
Duty cycle	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch00/OpOutput/port1[1]
Channel 2	(2)
Frequency	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch01/OpOutput/port1[0]
Duty cycle	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch01/OpOutput/port1[1]
Channel 4	(2)
Frequency	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch02/OpOutput/port1[0]
Duty cycle	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch02/OpOutput/port1[1]
Channel 6	(2)
Frequency	⇐ OP4200_eHSx32_Boost_and_TwoLevel_Inverter/SM_eHS/Digital_Out/PWMOut_Ch03/OpOutput/port1[0]

Figure 31: Connections used by the example project

LOADING THE MODEL

In the Overview tab, click on Load **the model**. The console will open, and the real-time code will be uploaded to the simulator. The **Loading model** window appears briefly during the loading process.

The **Display** tab in the lower portion of the window will show load progress and details.


RT-LAB 2017
File Edit Navigate Search Simulation Tools Window Help
Quick Access
Project Explorer
Targets
MyOP4200
eHS_examples
Models
OP4200_eHSx32_Boost_and_TwoLevel_Inverter [Waiting]
I/Os
OPAL-RT Board -> SM_eHS
Configuration
Create a new project...
MyOP4200
OP4200_eHSx32_Boost_and_TwoLevel_Inverter
OPAL-RT Board
Overview
General Information
Preparing and Compiling
Edit the model.
Set the development properties.
Build the model.
Consult result in the Compilation View
Assign targets to subsystems.
Executing
Set the execution properties.
Load the model.
Execute the model.
Pause the model.
Reset the model.
Interacting
Open the Console in Simulink.
Overview Development Execution Variables Files Assignation Diagnostic Hardware Simulation Tools
Progress
Loading model
Loading OP4200_eHSx32_Boost_and_TwoLevel_Inverter
Display Properties Compilation Matlab View Console Variables Table Variable Viewer Monitoring
Loading model

Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter

Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_Boost_and_TwoLevel_Inverter/OI

Matlab: R2011B

Loading model

Loading OP4200_eHSx32_Boost_and_TwoLevel_Inverter

☐ Always run in background

Run in Background | Cancel | Details >>

Model:OP4200_eHSx32_Boost_and_TwoLevel_Inverter

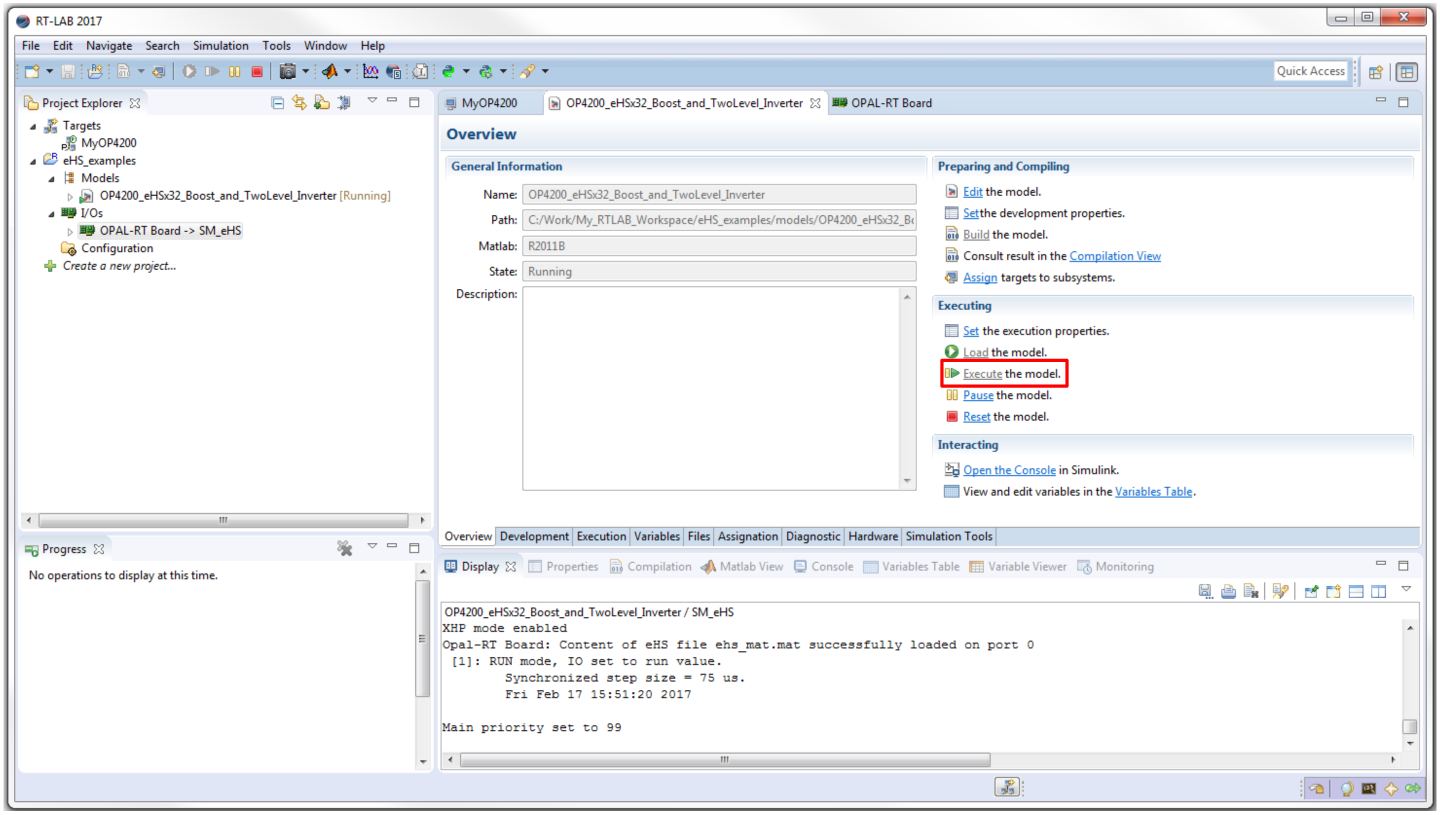
```
------------------- Starting compilation --------------------
Start at : Friday, February 17, 2017, 14:40:54

The current RT-LAB version is:  v11.1.x.integration.861
The current model is:           C:\Work\My_RTLAB_Workspace\eHS_examples\models\OP4200_eHSx32_Boost_and_TwoLevel_Inverter\OP4200_eHSx32_Boost
The current host platform is:   Windows
The current target platform is: Petalinux (ARMv7-based)
The current compiler is:        Automatic
WARNING: Model separation will be skipped because no change has been detected.
WARNING: Model code generation will be skipped because no change has been detected.
The current Matlab version is:  v7.13
```

Figure 32: Loading model window

EXECUTING THE SIMULATION

Click on Execute **the model** to start the simulation. At the beginning of the simulation, eHS will initialize (it takes about 10,000 simulation steps). During this time, the eHS outputs will remain at 0.

Clicking on the **Display** tab shows execution details.


RT-LAB 2017
File Edit Navigate Search Simulation Tools Window Help
Quick Access
Project Explorer
Targets
MyOP4200
eHS_examples
Models
OP4200_eHSx32_Boost_and_TwoLevel_Inverter [Running]
I/Os
OPAL-RT Board -> SM_eHS
Configuration
Create a new project...
Progress
MyOP4200
OP4200_eHSx32_Boost_and_TwoLevel_Inverter
OPAL-RT Board
Overview
General Information
Preparing and Compiling
Executing
Interacting
Overview Development Execution Variables Files Assignation Diagnostic Hardware Simulation Tools
Display Properties Compilation Matlab View Console Variables Table Variable Viewer Monitoring

No operations to display at this time.

Name: OP4200_eHSx32_Boost_and_TwoLevel_Inverter

Path: C:/Work/My_RTLAB_Workspace/eHS_examples/models/OP4200_eHSx32_B(

Matlab: R2011B

State: Running

Description:

Edit the model.

Setthe development properties.

Build the model.

Consult result in the Compilation View

Assign targets to subsystems.

Set the execution properties.

Load the model.

Execute the model.

Pause the model.

Reset the model.

Open the Console in Simulink.

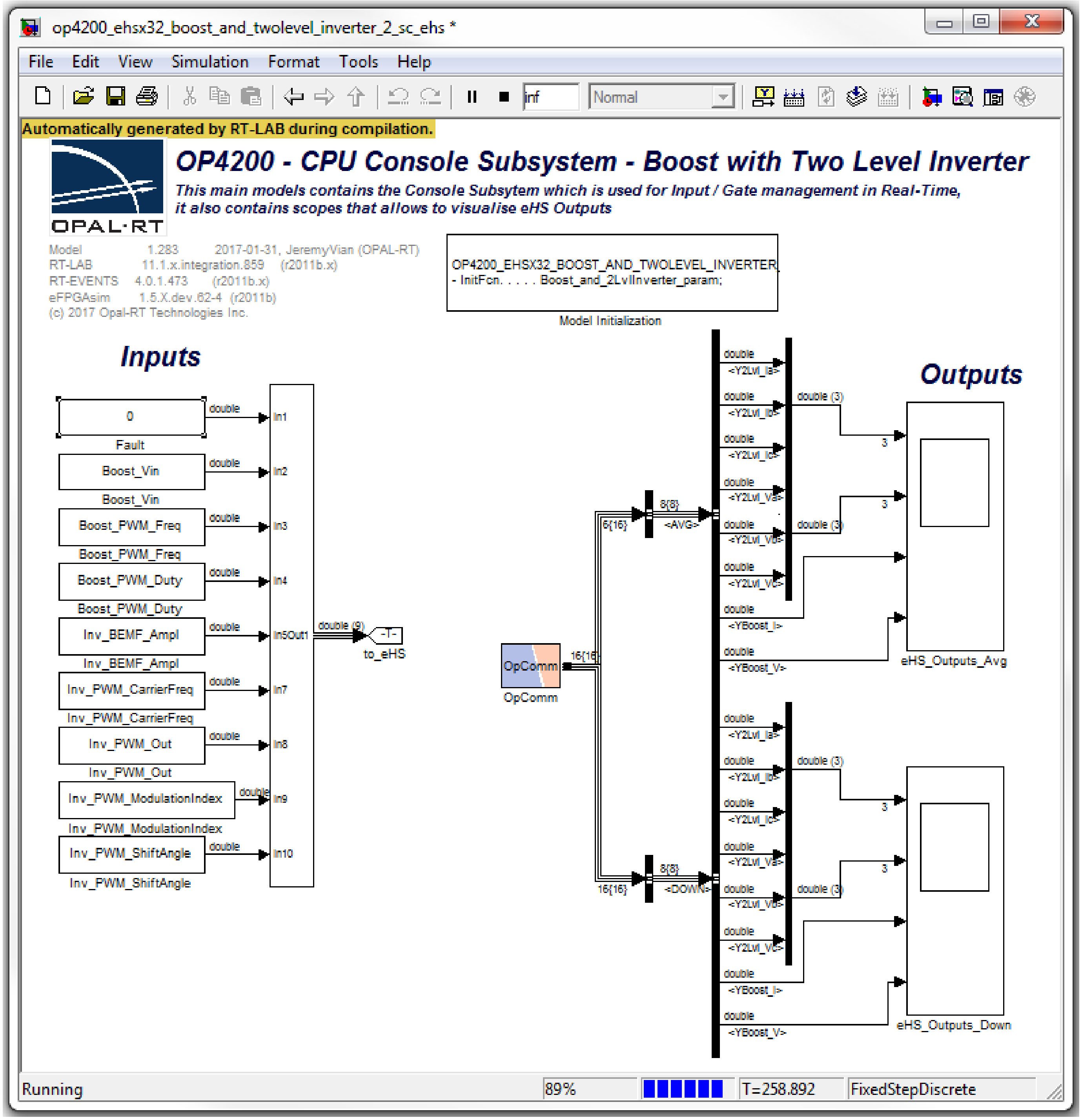
View and edit variables in the Variables Table.

```
OP4200_eHSx32_Boost_and_TwoLevel_Inverter / SM_eHS
XHP mode enabled
Opal-RT Board: Content of eHS file ehs_mat.mat successfully loaded on port 0
 [1]: RUN mode, IO set to run value.
        Synchronized step size = 75 us.
        Fri Feb 17 15:51:20 2017

Main priority set to 99
```

Figure 33: Displaying execution details

MONITORING THE SIMULATION

As soon as you load and execute the model, the Simulink Console opens (behind RT-LAB) and you should be able to see the simulation running in the Console window. Double-clicking on the **eHS_Outputs_Avg** scope will display the results of the eHS simulation.



op4200_ehsx32_boost_and_twolevel_inverter_2_sc_ehs *

Automatically generated by RT-LAB during compilation.

OP4200 - CPU Console Subsystem - Boost with Two Level Inverter

This main models contains the Console Subsytem which is used for Input / Gate management in Real-Time, it also contains scopes that allows to visualise eHS Outputs

```
Model        1.283          2017-01-31, JeremyVian (OPAL-RT)
RT-LAB       11.1.x.integration.859    (r2011b.x)
RT-EVENTS    4.0.1.473      (r2011b.x)
eFPGAsim     1.5.X.dev.62-4  (r2011b)
(c) 2017 Opal-RT Technologies Inc.
```

OP4200_EHSX32_BOOST_AND_TWOLEVEL_INVERTER
- InitFcn. . . . . Boost_and_2LvlInverter_param;

Model Initialization

Running | 89% | T=258.892 | FixedStepDiscrete

Figure 34: Console window of running model

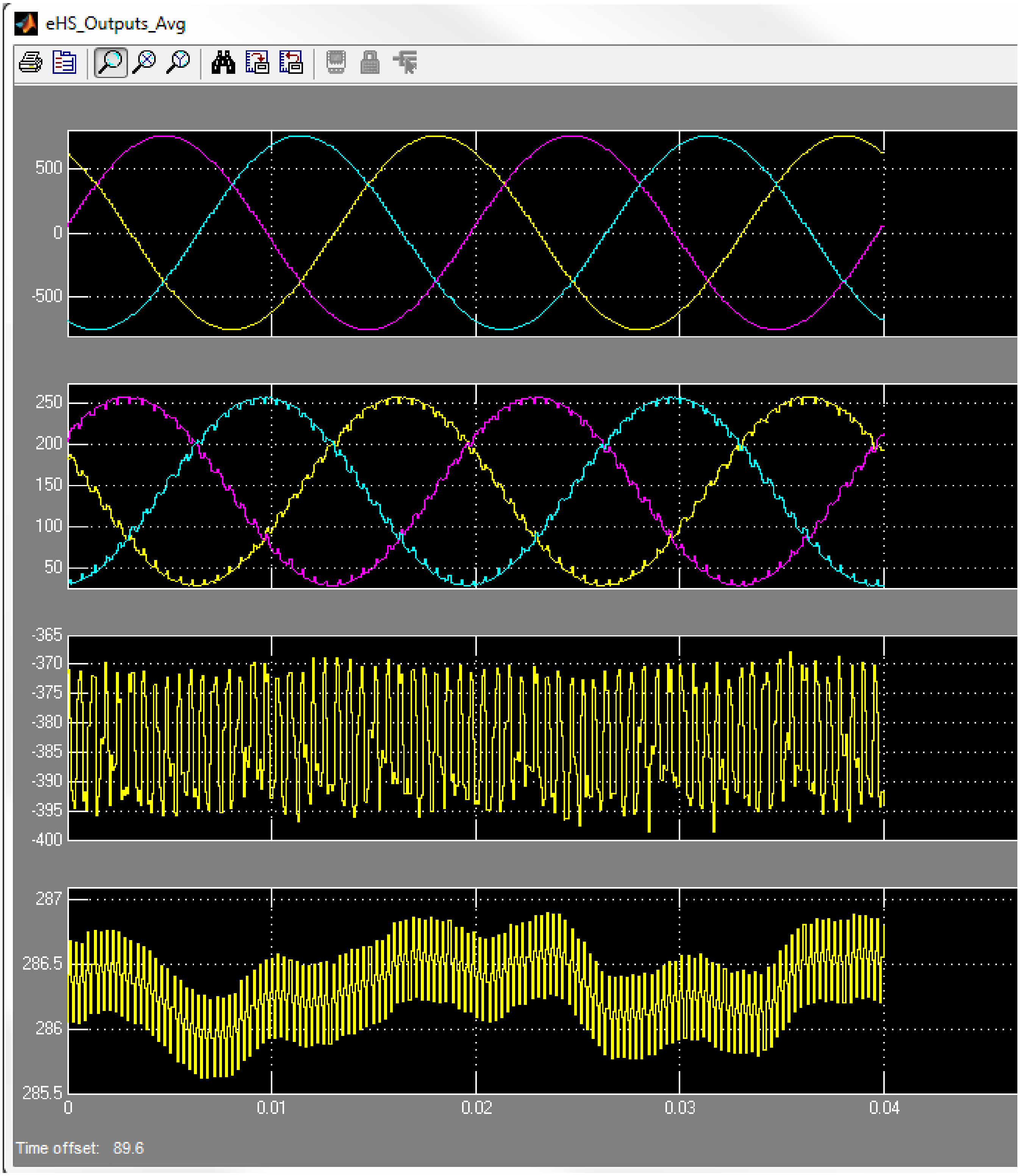
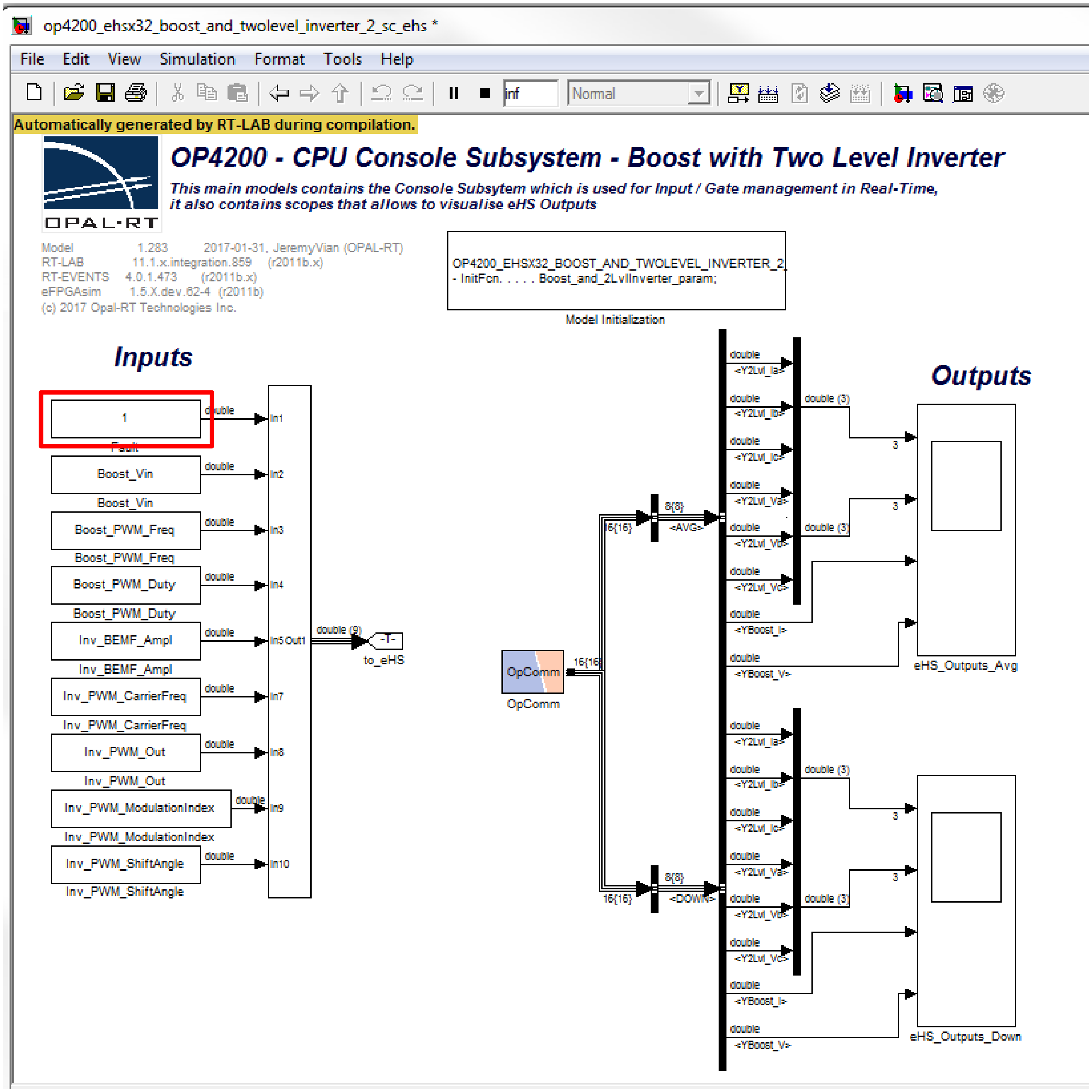


Figure 35: Displaying signals in the Simulink console

CONTROLLING THE SIMULATION

Simulation operating conditions can now be modified directly from the Console. Simply click on the desired signal in the Console to open its *Block Parameters* window. In this particular example, most of the signals that are sent to the eHS are configured using a file (*Boost_and_2LvInverter_param.m*). Any of these signals can be changed during execution if desired. For example, change the *Fault* value to **1** and observe the effects in the *eHS_Outputs_Avg* scope.



op4200_ehsx32_boost_and_twolevel_inverter_2_sc_ehs *

Automatically generated by RT-LAB during compilation.

OP4200 - CPU Console Subsystem - Boost with Two Level Inverter

This main models contains the Console Subsytem which is used for Input / Gate management in Real-Time, it also contains scopes that allows to visualise eHS Outputs

```
Model        1.283        2017-01-31, JeremyVian (OPAL-RT)
RT-LAB       11.1.x.integration.859    (r2011b.x)
RT-EVENTS    4.0.1.473       (r2011b.x)
eFPGAsim     1.5.X.dev.62-4  (r2011b)
(c) 2017 Opal-RT Technologies Inc.
```

OP4200_EHSX32_BOOST_AND_TWOLEVEL_INVERTER_2
- InitFcn. . . . . Boost_and_2LvlInverter_param;

Model Initialization

1

Figure 36: Changing parameter values in the console

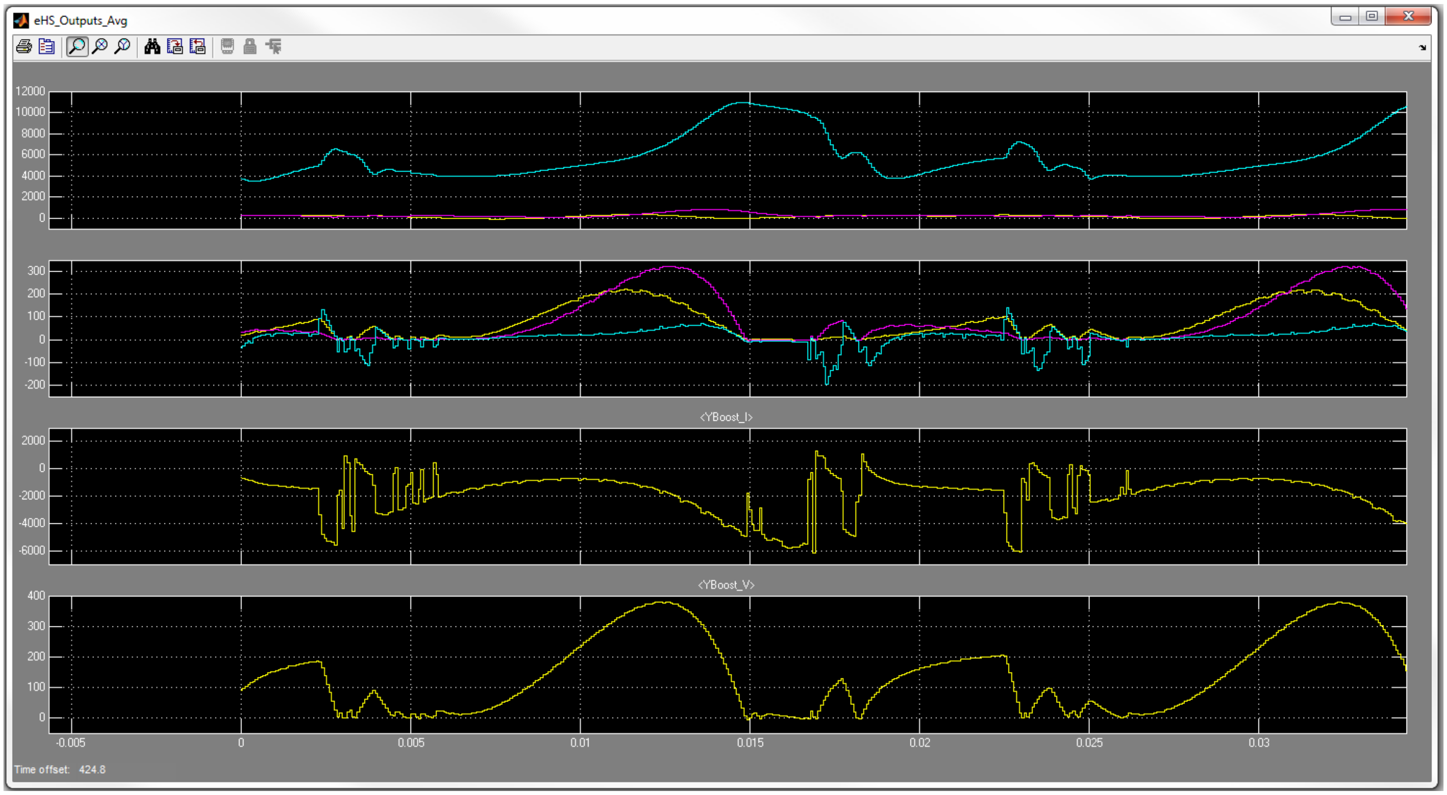


Figure 37: Impact of changing parameter values

CONTACT

OPAL-RT Corporate Headquarters

1751 Richardson, Suite 2525
Montréal, Québec, Canada
H3K 1G6
Tel.: 514-935-2323
Toll free: 1-877-935-2323

Technical Services
www.opal-rt.com/support

Note:
While every effort has been made to ensure accuracy in this publication, no responsibility can be accepted for errors or omissions. Data may change, as well as legislation, and you are strongly advised to obtain copies of the most recently issued regulations, standards, and guidelines.
This publication is not intended to form the basis of a contract.

14/06/17