OPAL-RT TECHNOLOGIES

FROM IMAGINATION TO REAL-TIME

Using IOs with OP4200: understanding the driver and model

Drivers Team

OPAL-RT Board /OP4200

- **Release versions:**
  - RT-LAB: 11.1.3
- **License keys:**
  - No license needed
- **Example project:**
  - RT-LAB Installation Folder\Examples\IO\Opal-RT\OP4200\OP4200_Generic_Integration
- **Example project Help:**
  - RT-LAB Installation Folder\Examples\IO\Opal-RT\OP4200\Generic_Serie\help\help.html
- **OPAL-RT Board documentation:**
  - Help -> Help Contents -> RT-LAB I/O Block Library Reference Guide -> Opal-RT -> OPAL-RT Board
- **OP4200 user manual:**
  - http://www.opal-rt.com/wp-content/themes/enfold-opal/pdf/L00161_0497.pdf

Using IOs with OP4200

At first glance..

Using IOs with OP4200

At first glance..

Using IOs with OP4200

At first glance..

The first official release of RT-LAB that contains support for OP4200 is 11.1.3 (released April 7th 2017 to the IVV team).

The first unofficial release of RT-LAB to contain support for OP4200 is 11.1.1 (released February 9th 2017); therefore it is also part of the unofficial release 11.1.2.

OP4200 does **not** work with the Simulink I/O blocks used for the other FPGA platforms (e.g. OP5142, ML605). Instead, in order to interface with the physical IOs, OP4200 makes use of the *OPAL-RT Board* driver.

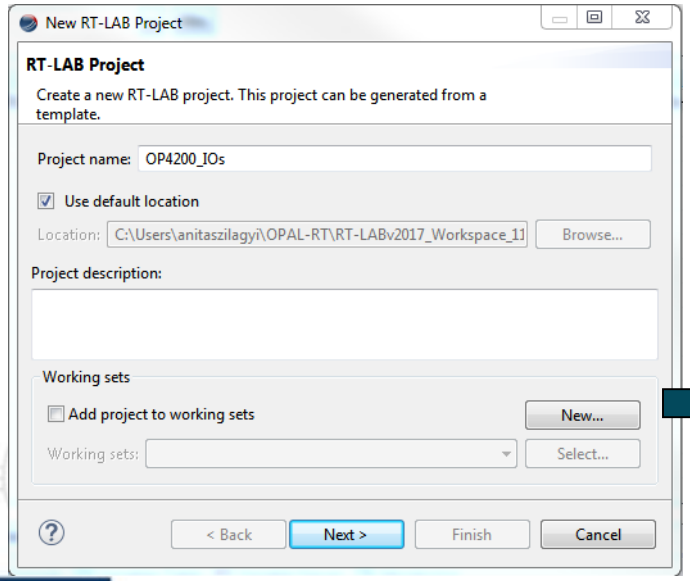
This driver is using the DiL architecture as opposed to being embedded in Simulink; this is why you will not find any mention of it in the Simulink Library Browser.

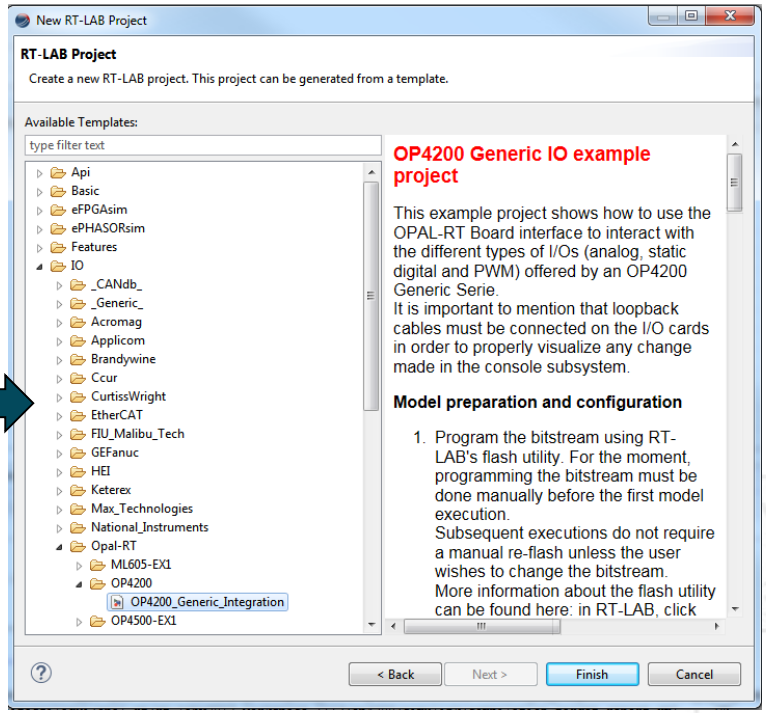
The driver and implicitly the IO communication are entirely configurable through a graphical user interface. This GUI is a new feature in RT-LAB and is not yet available to all supported drivers (work in progress).

Using IOs with OP4200

At first glance..

- Current support of OP4200 **does not** include asynchronous processes (such as TCP/IP, RS232..).
- Starting from version 11.1.3 of RT-LAB there is an example project included that demonstrates all IO capabilities currently supported on the OP4200 platform.
- The project contains the official bitstream delivered with all OP4200 systems: MEZX5-AX-0001-3_1_2_360-eHSgen3_with_IOs-21-17.bin. Currently, this is the only official bitstream for OP4200 released by OPAL-RT.
- You can find the example project at the following location:
  <YOUR_RTLAB_INSTALLATION>\Examples\IO\Opal-RT\OP4200\Generic_Serie\
- You can also add it to RT-LAB using the *New RT-LAB Project* wizard:

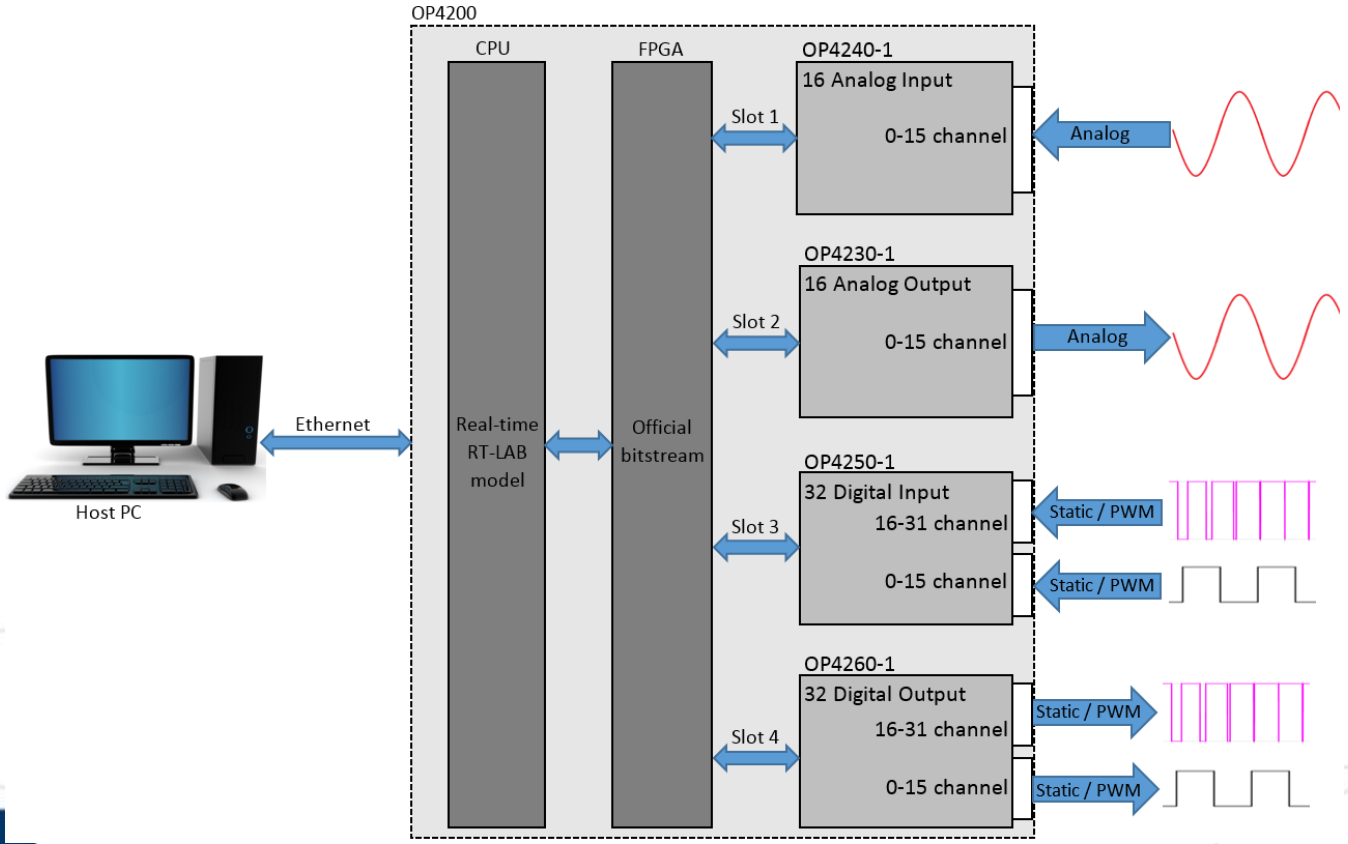




Using IOs with OP4200

At first glance..

MEZX5-AX-0001-3_1_2_360-eHSgen3_with_IOs-21-17.bin



Using IOs with OP4200: the driver

At first glance: new features found in OP4200 bitstreams

- The default bitstream contains Advanced Analog Out blocks as opposed to the standard blocks used thus far
  - They can perform the following extra processing of the analog signals inside the FPGA before outputting the signals through the IO module: apply a gain, an offset, a minimum cut-off and a maximum cut-off value
  - They allow for different signal sources; the sources can be either coming from the Simulink model or from logic internal to the bitstream (such as an eHS block)
  - The presence or absence of the Advanced Analog Out blocks in the bitstream is confirmed at once when navigating to the channels view of the analog out slots (if there are any); this is possible only after the bitstream configuration has been selected in the drop-down menu found in the *General* view of the driver configuration page
- The default bitstream contains Selectable Digital In/Out blocks as opposed to standard digital processing blocks used thus far
  - They permit the co-existence of the following functionalities: Static Digital, Time-stamped Digital and Pulse-width-modulated signals
  - The functionalities apply on groups of 8 channels
  - Between model loads, the user can select the type of functionality to apply to a group of channels
  - This eliminates the need to change the bitstream in case the functionality type needed has changed

For more information about these topics, please consult the help page of the driver (referenced at slide 11) or contact the eFPGAsim team or the Firmware team.

Using IOs with OP4200

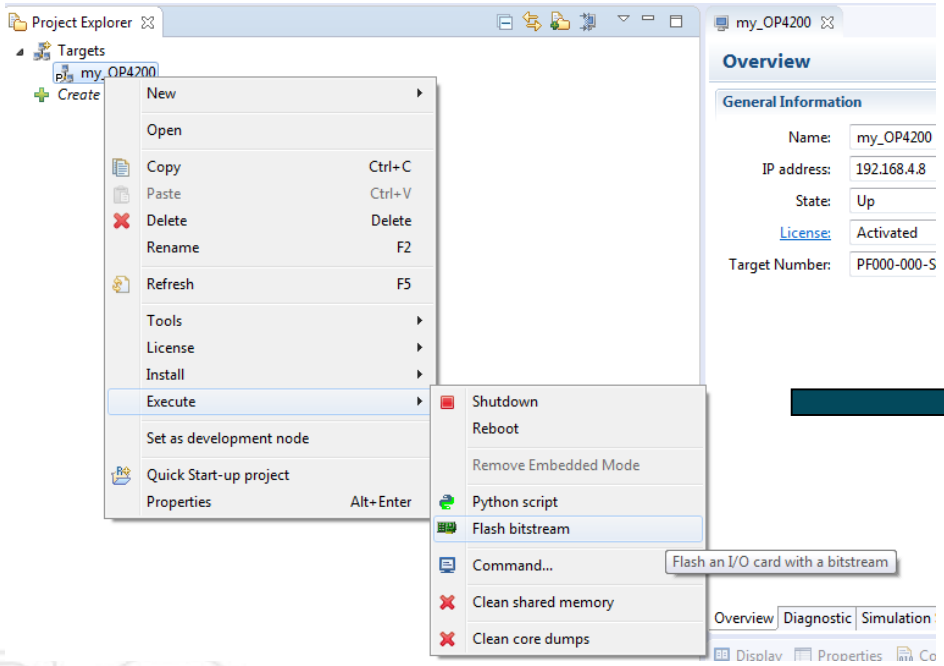
At first glance..

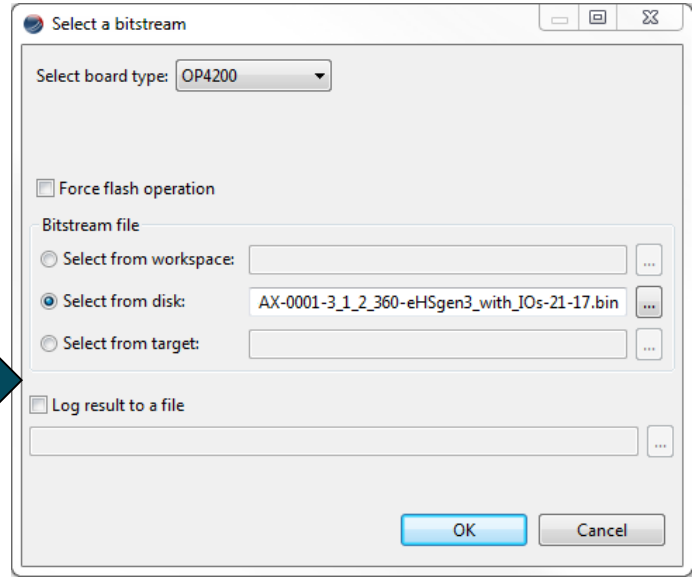
For other bitstream configurations, you need to send a request to the Product Office (Bureau des produits).

The purpose of this information session is to show how to add and configure IOs through the use of *OPAL-RT Board* and set up a model in order to run a simulation on the OP4200 simulator.

Using IOs with OP4200

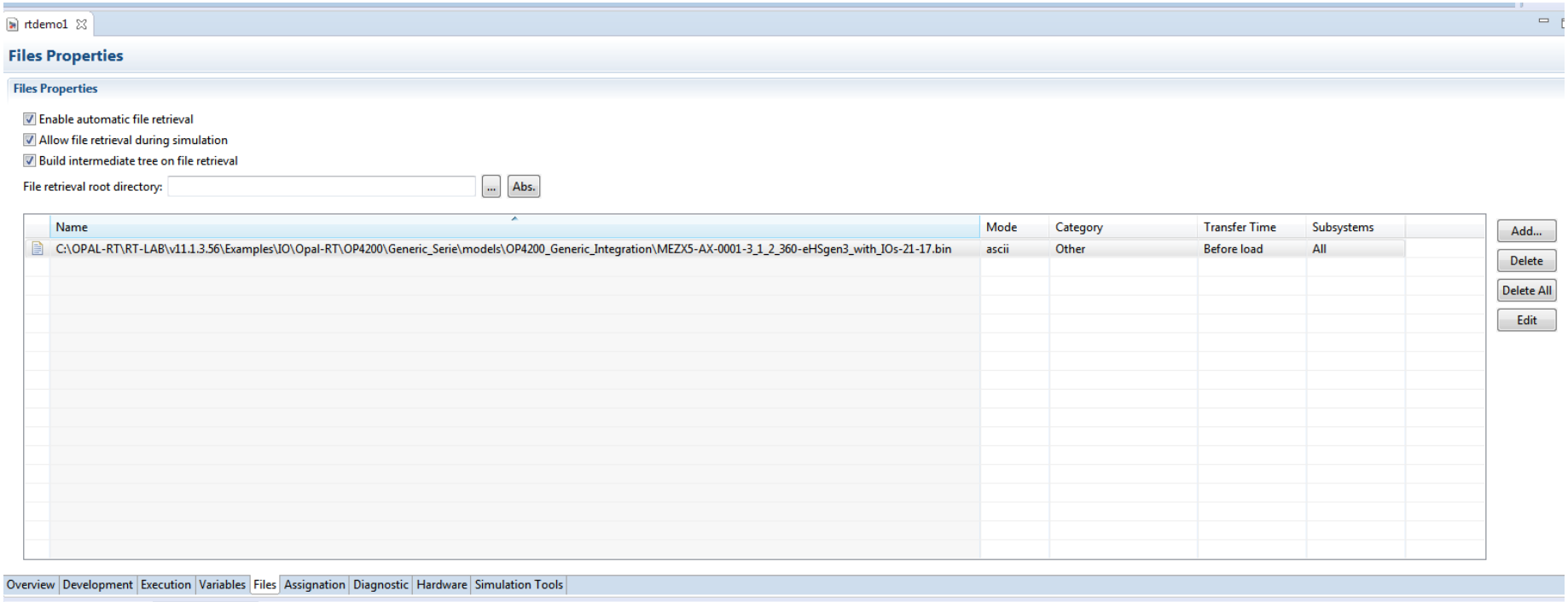
Programming the bitstream





Programming the bitstream

Alternatively, the bitstream to be flashed can be placed in the *Files* tab of the model:



If the bitstream has been flashed as depicted on slide 6 or the binary file is set to be transferred to the target as depicted above, the *OPAL-RT Board* driver will flash the same bitstream at every load of the model that it is associated to.

Using IOs with OP4200: the driver

OPAL-RT Board

- Developed in the DiL architecture, it does not need dedicated Simulink blocks in the model
- It covers all functionalities required by the FPGA: control, send, receive etc
- Configuration in Simulink blocks replaced by configuration in a graphical user interface that is part of RT-LAB
- Reference documents: in RT-LAB, click on Help -> Help Contents. In the window that appears, navigate to RT-LAB I/O Block Library Reference Guide -> Opal-RT -> OPAL-RT Board
  There you can find the documentation for the following:
  - Overview of the driver
  - Digital In
    - » Static
    - » Pulse-width-modulated
  - Digital Out
    - » Static
    - » Pulse-width-modulated
  - Analog In
  - Analog Out

Using IOs with OP4200: the driver

OPAL-RT Board

- Only OP4200 supported for the moment for the version of the driver available in RT-LAB

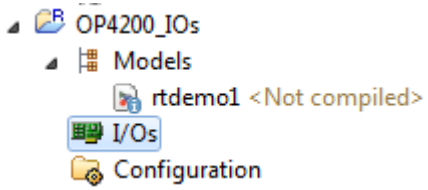
- Current functionalities supported:
  - Digital In/Out static
  - Digital In/Out PWM
  - Analog In/Out
  - Data In/Out (currently not part of the official OP4200 bitstream)

- Current limitations:
  - Does **not** support Load In/Load Out
  - Does **not** support Event Detector/Event Generator
  - Does **not** support Encoder In/Encoder Out
  - Does **not** support Resolver In/Resolver Out
  - Does **not** support TSB In

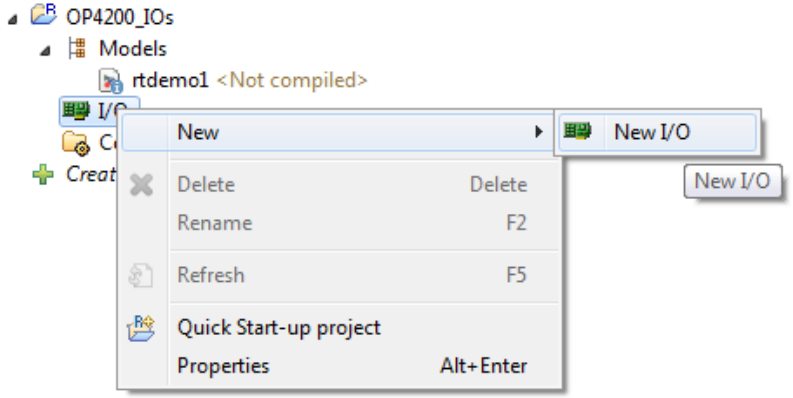
Using IOs with OP4200: the driver

Adding *OPAL-RT Board* to a project

1. Initial view of the project



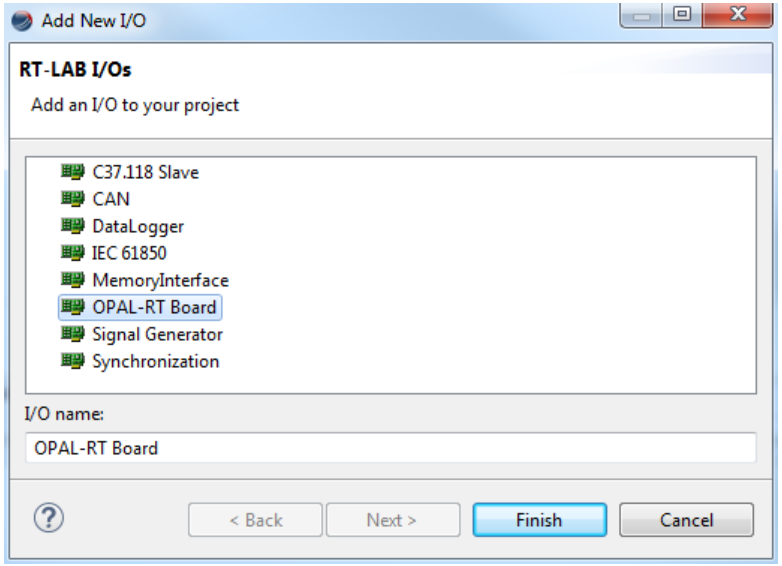
2. Adding a new I/O



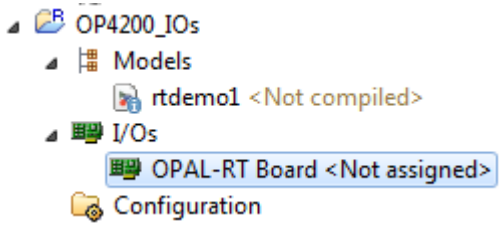
Using IOs with OP4200: the driver

Adding *OPAL-RT Board* to a project

3. Select OPAL-RT Board from the list



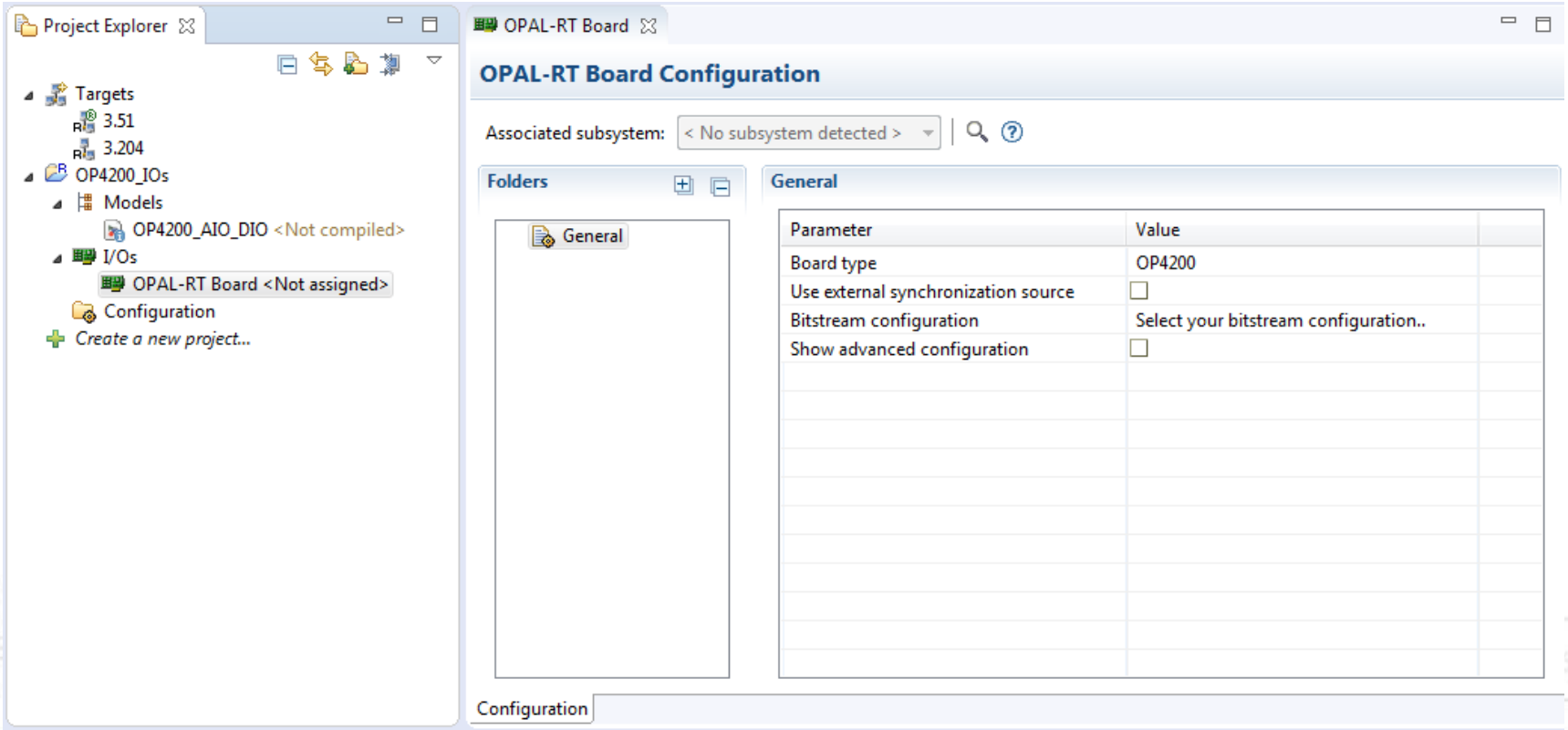
4. Clicking on “Finish” will add it to the project



Using IOs with OP4200: the driver

Understanding the driver: blank driver configuration page

Double clicking the driver in the I/Os section of the project explorer will reveal the driver configuration page.

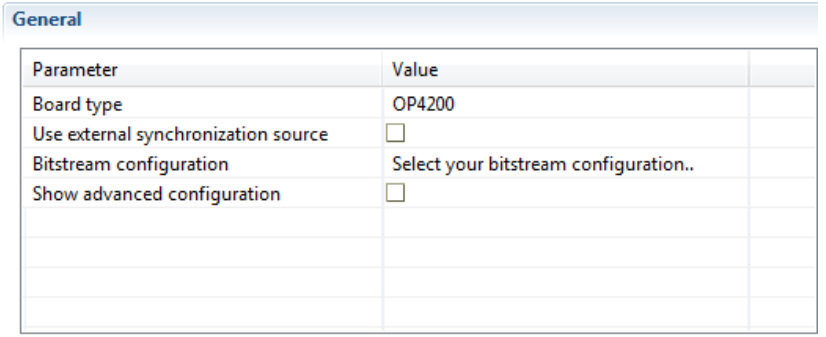


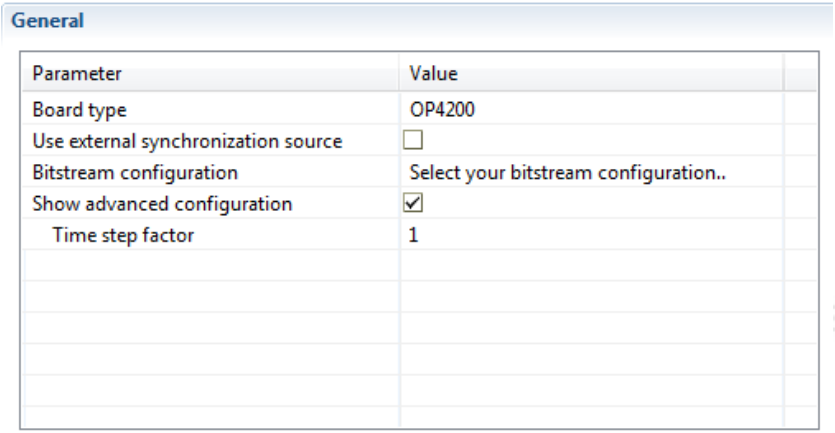
Using IOs with OP4200: the driver

Understanding the driver: blank driver configuration page

In the *General* view the user has the following options:

- Select the board type; currently only OP4200 is supported
- Use external synchronization source: select if the board is configured as a master or as a master with an external clock; configuration as slave is not supported since simulations on the OP4200 can only have 1 computation sub-system, hence only 1 FPGA in a simulation
- Selecting a bitstream configuration: this is detailed in the following slides
- Show advanced configuration: allows the user to set an FPGA time step factor i.e. the speed of the FPGA providing or reading data in relation with the timestep of the model





Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration

A bitstream configuration is (roughly) for the *OPAL-RT Board* driver what a .conf file is for the Simulink based FPGA drivers.

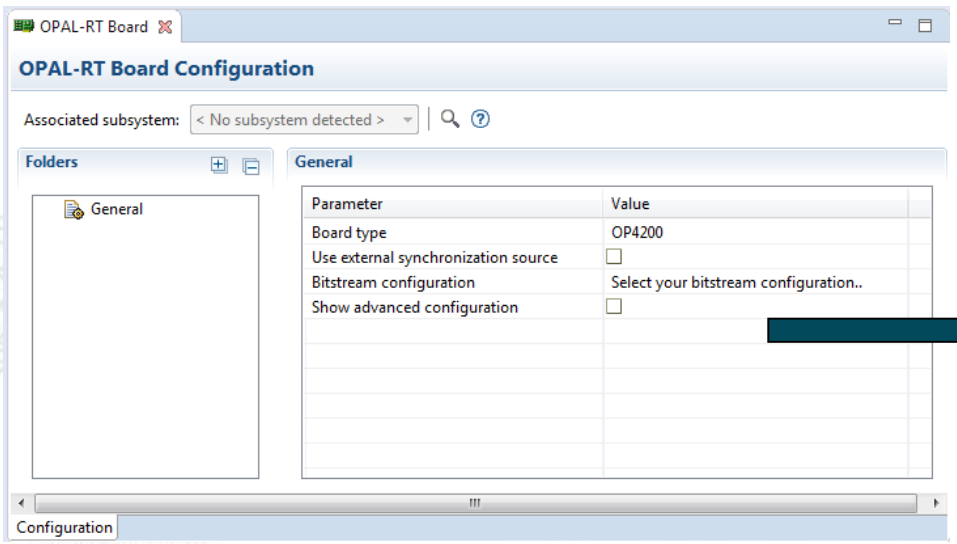
It describes the capabilities of the binary file programmed in the FPGA, i.e. the type of the IO module in each slot (or cassette for OP4200) and if there are any other proprietary tools in it (such as eHS).

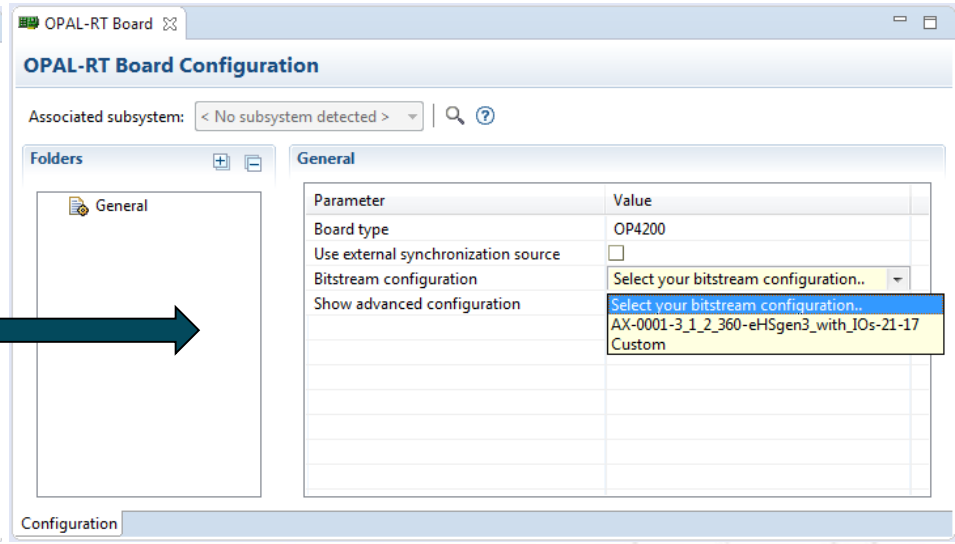
Very importantly, the binary file needs to be programmed in the FPGA prior to running a simulation using its respective configuration.

Once a configuration has been chosen, the communication parameters can be changed to fit the simulation's purposes.

Currently, there is only 1 officially distributed bitstream for OP4200. Its configuration is the only one that appears in the drop-down menu, as seen below.

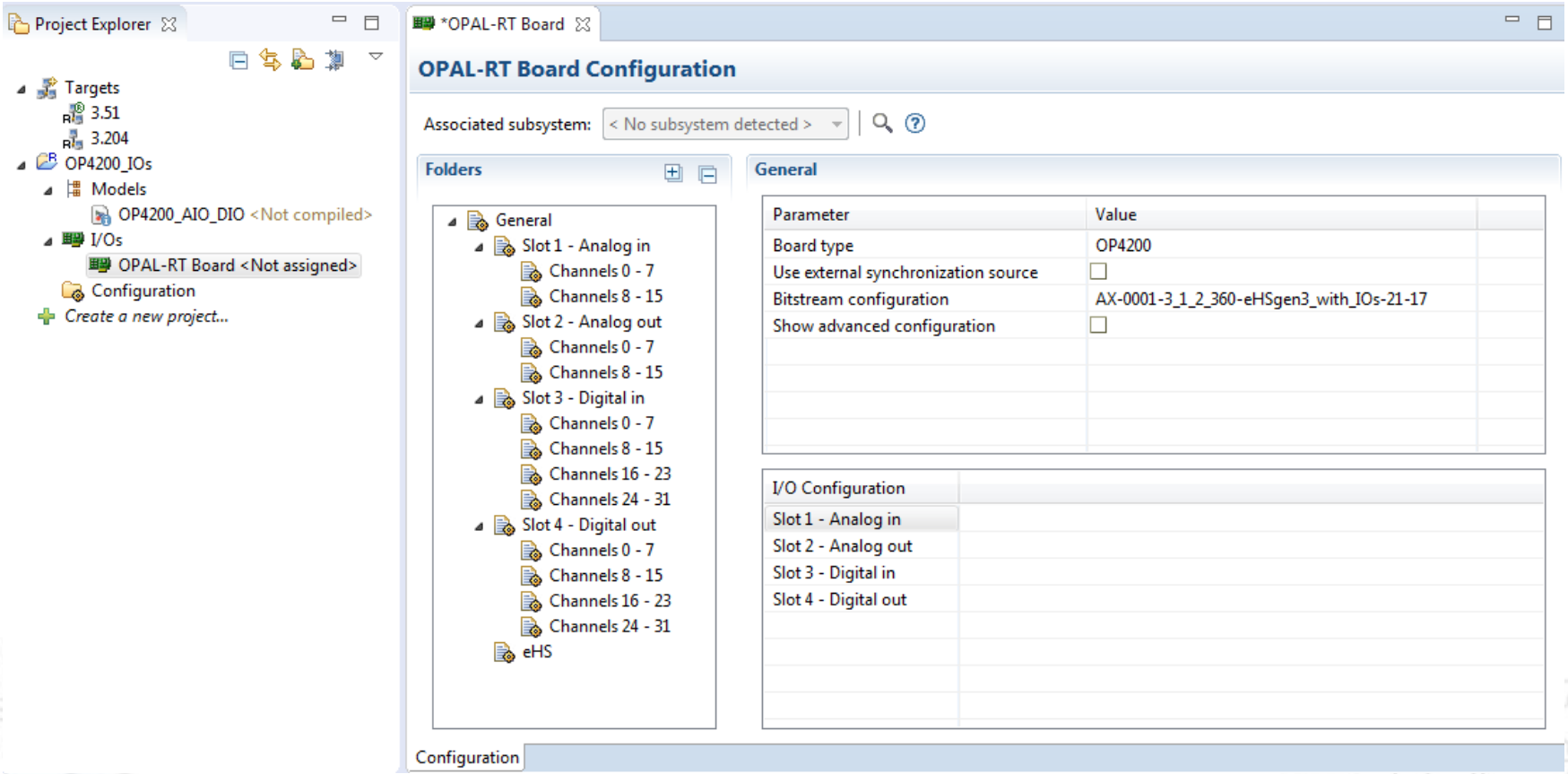
NOTE: the images below might be rendered out-of-date by a changes brought to the default bitstream (e.g. name, release version, improvements over time).





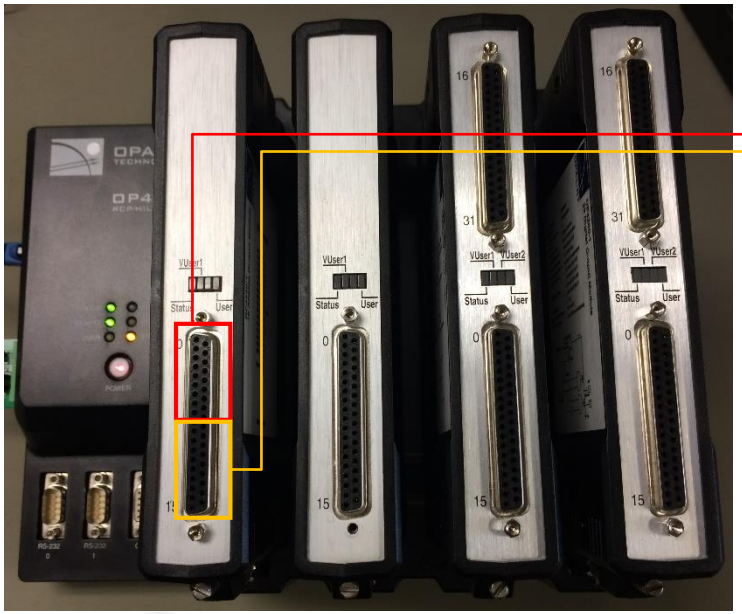
Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration

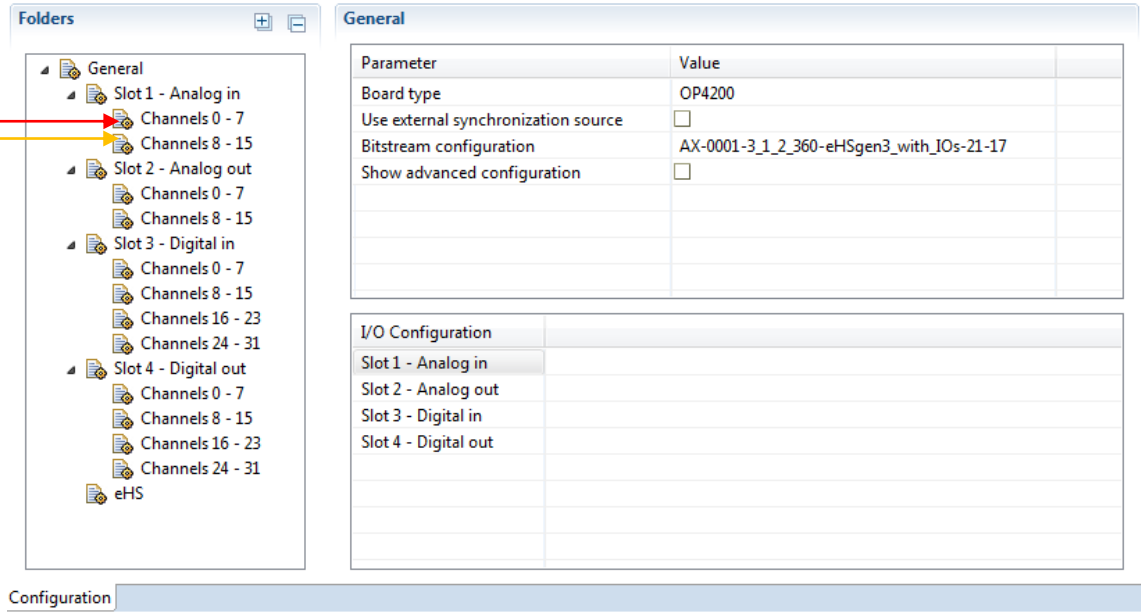


Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration

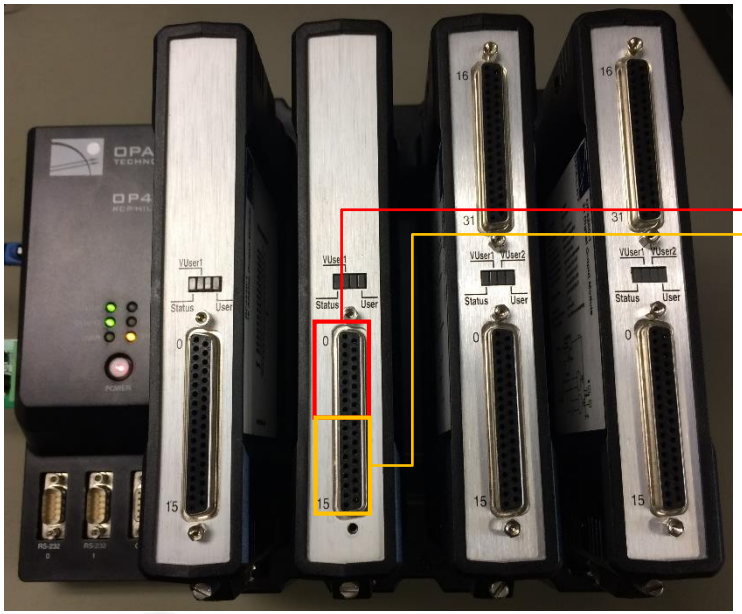


Slot 1
Analog in

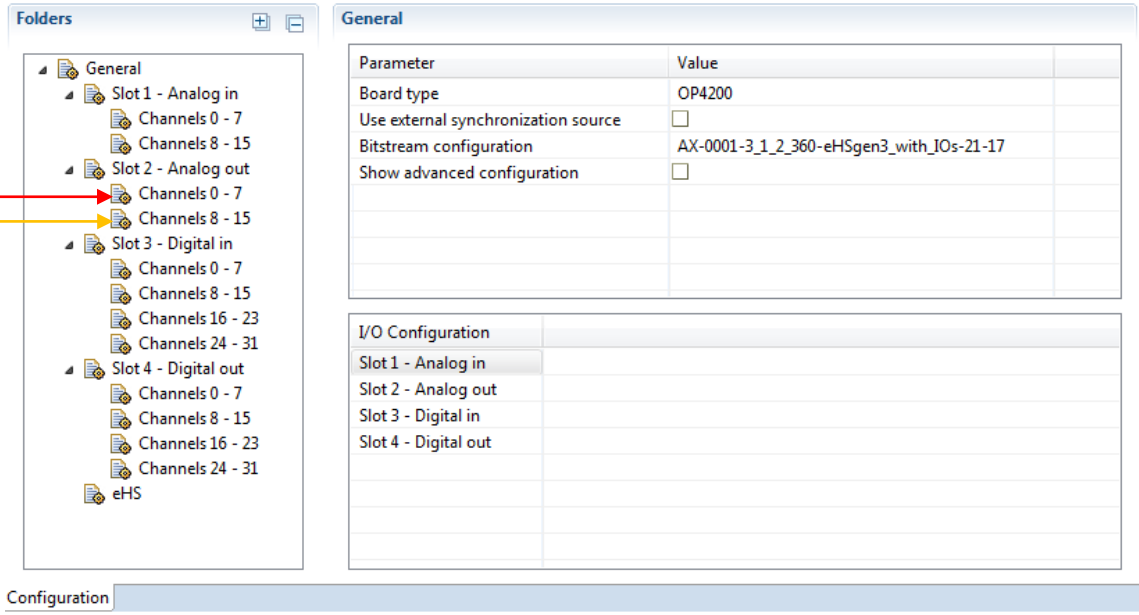


Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration

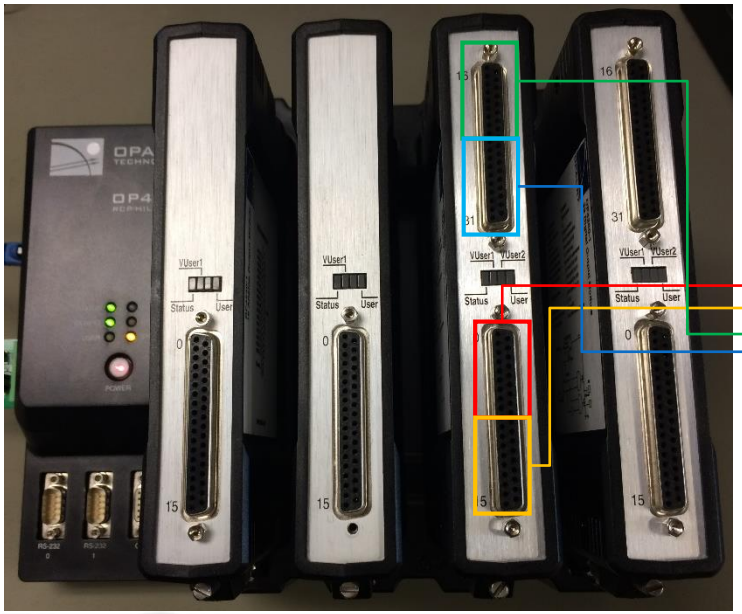


Slot 2
Analog out

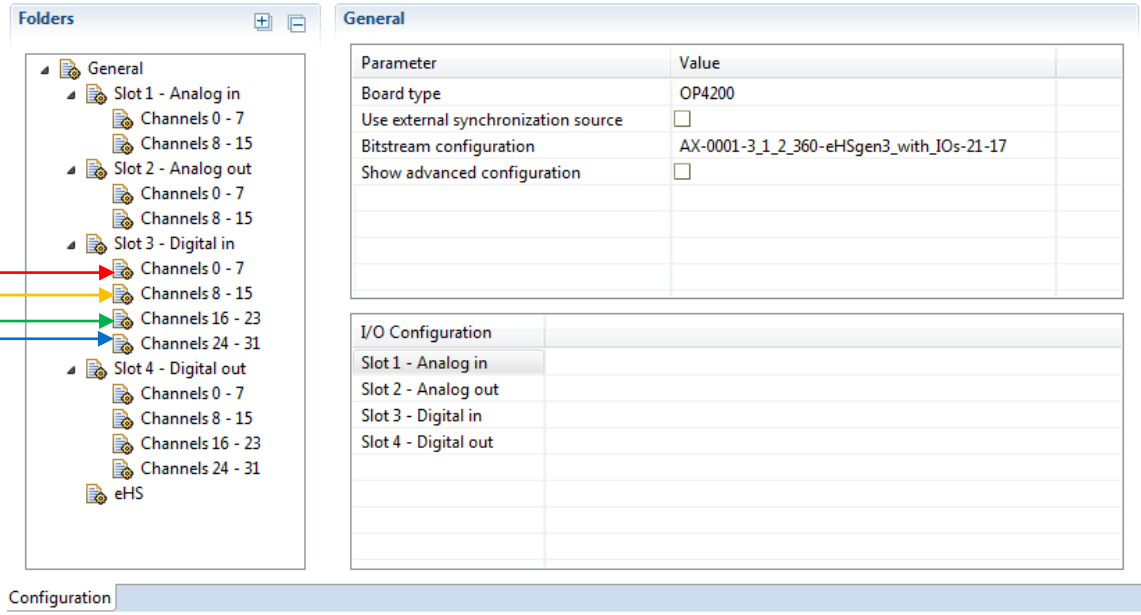


Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration

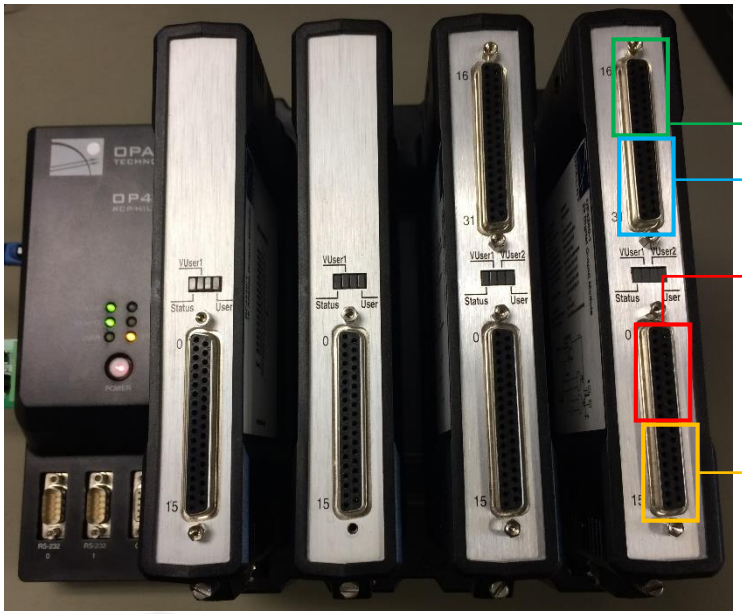


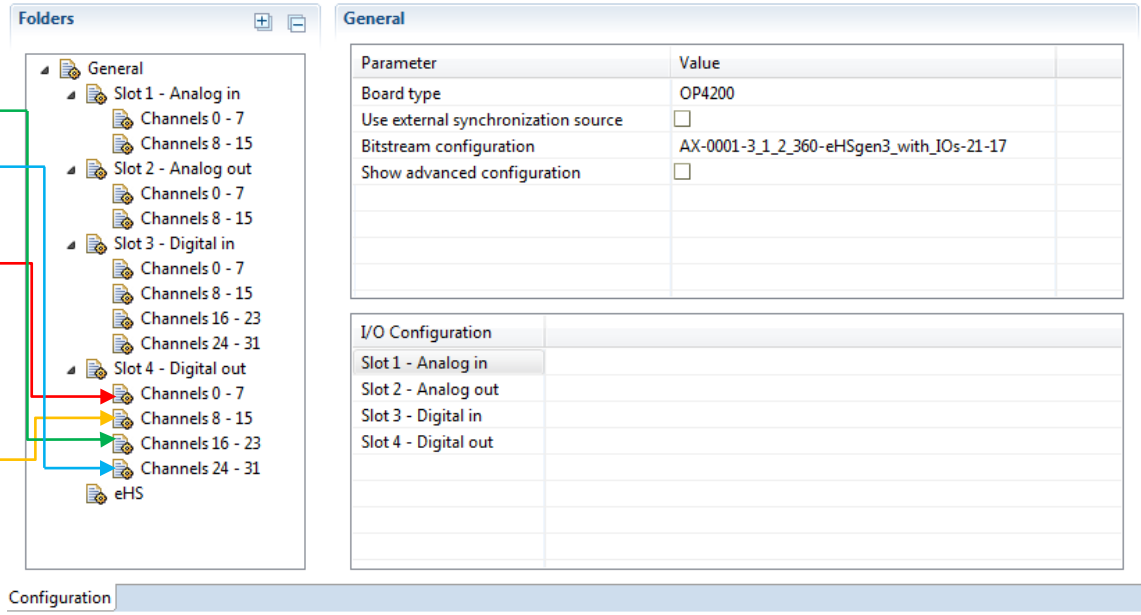
Slot 3
Digital in



Using IOs with OP4200: the driver

Understanding the driver: selecting a bitstream configuration





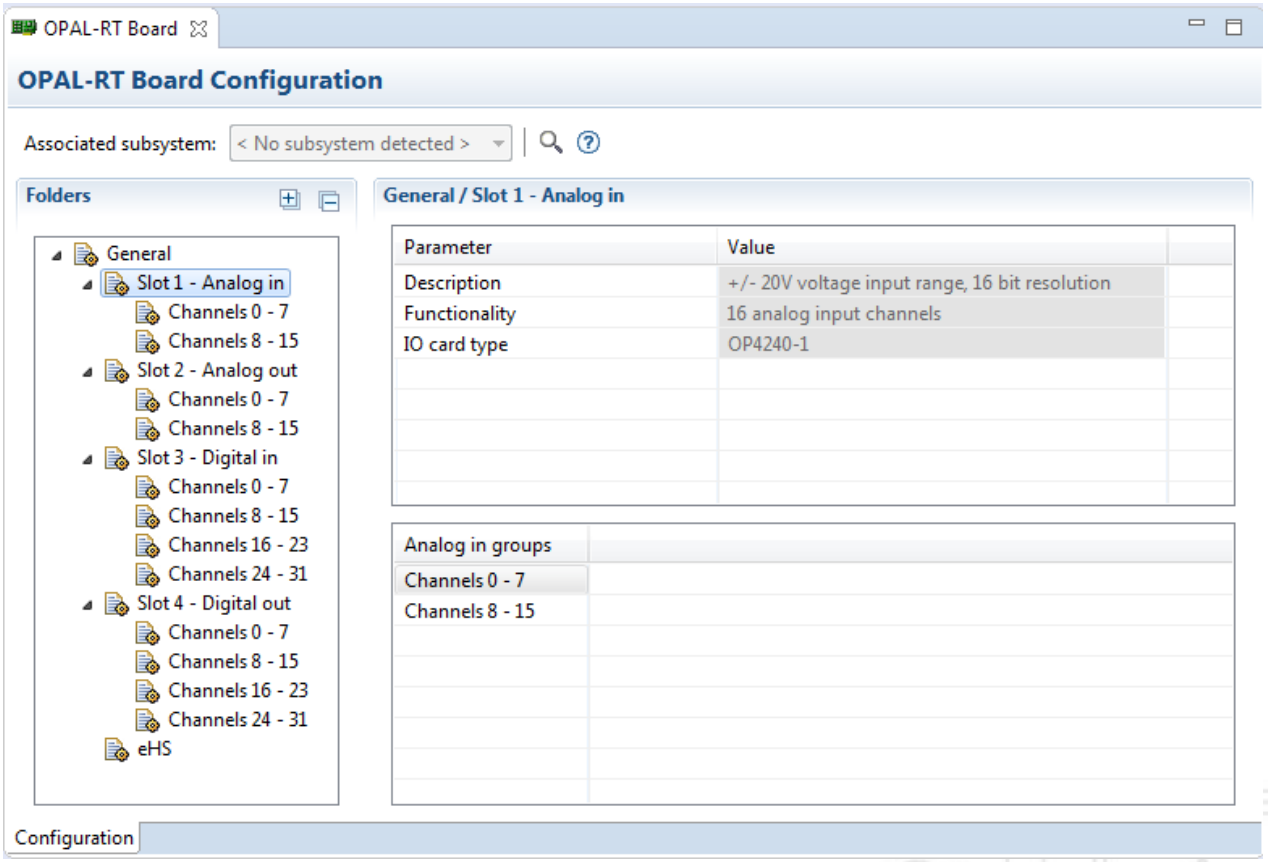
Slot 4
Digital out

Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog In

This view shows general information regarding the cassette found in slot 1.

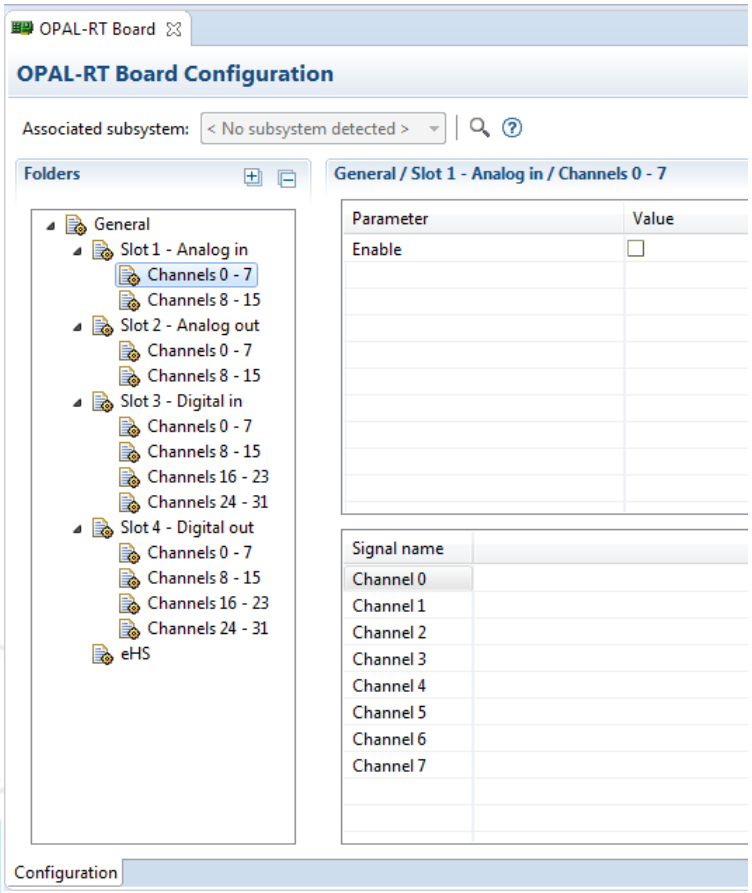
In this scenario, it is an Analog In module.

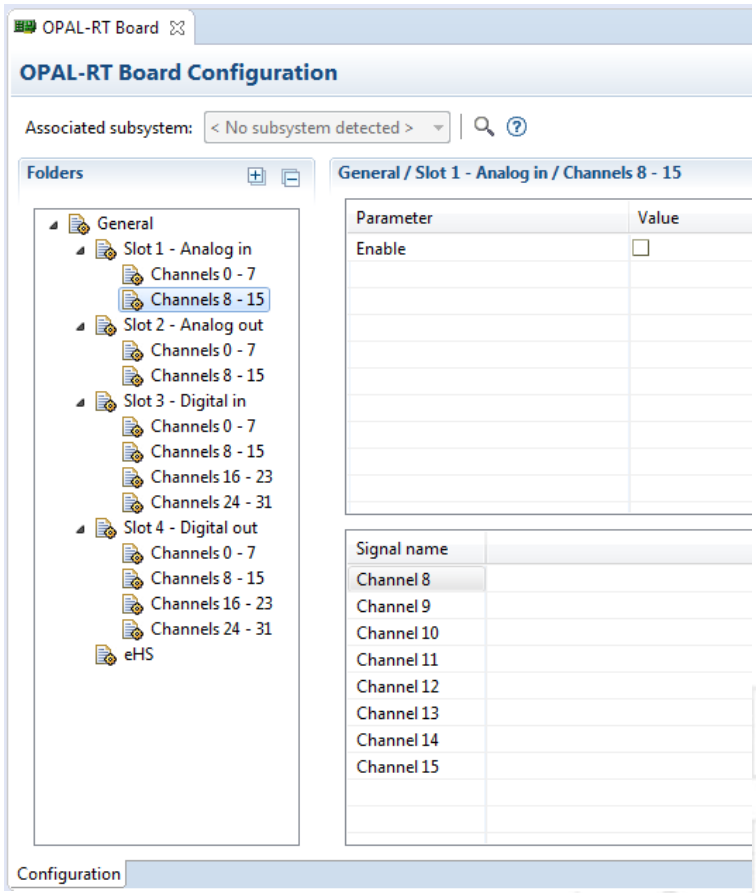


Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog In

Channels view: The user has the option of enabling or disabling the use of the group of 8 channels in the simulation.



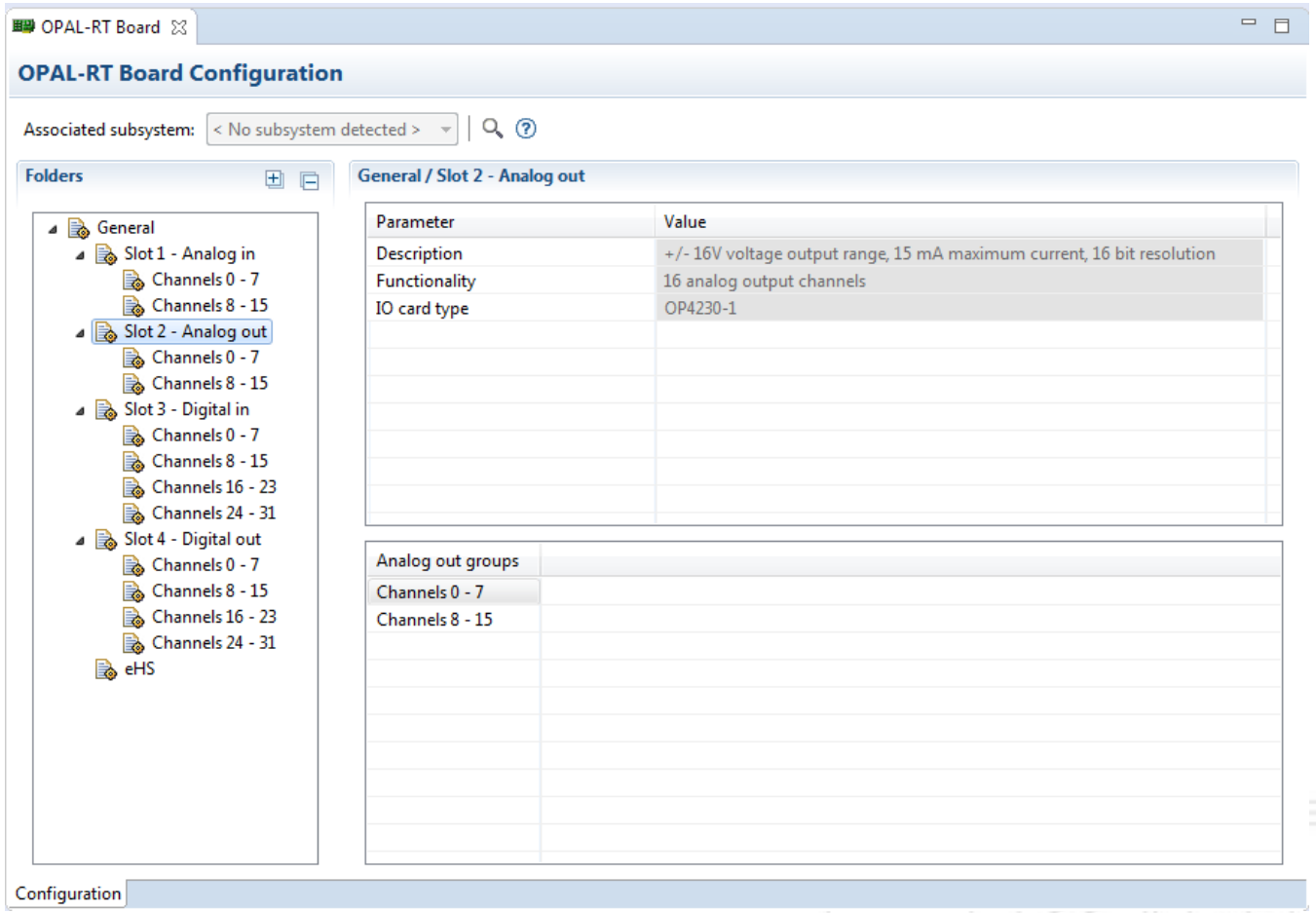


Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

This view shows general information regarding the cassette found in slot 2.

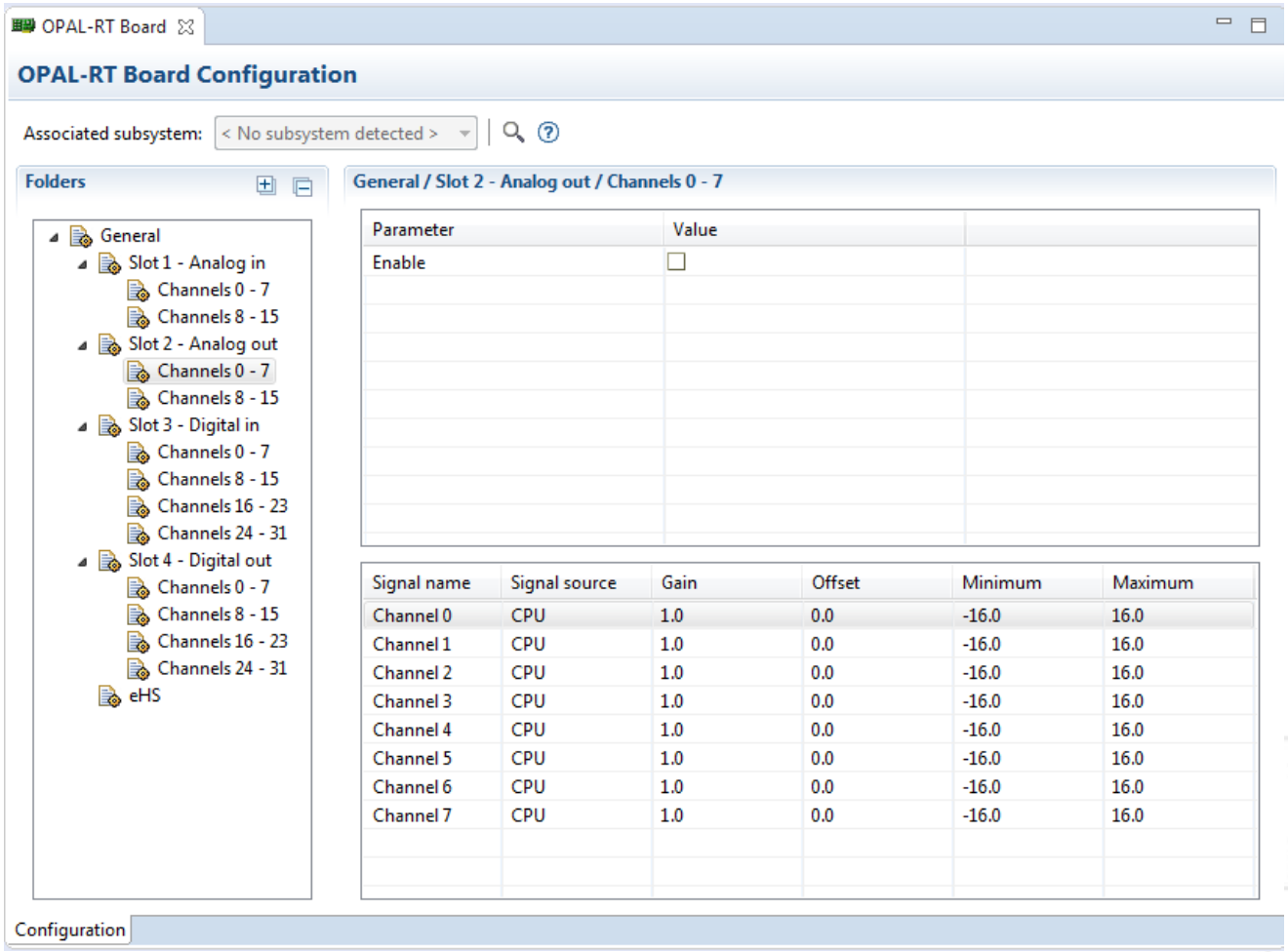
In this scenario, it is an Analog Out module.



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

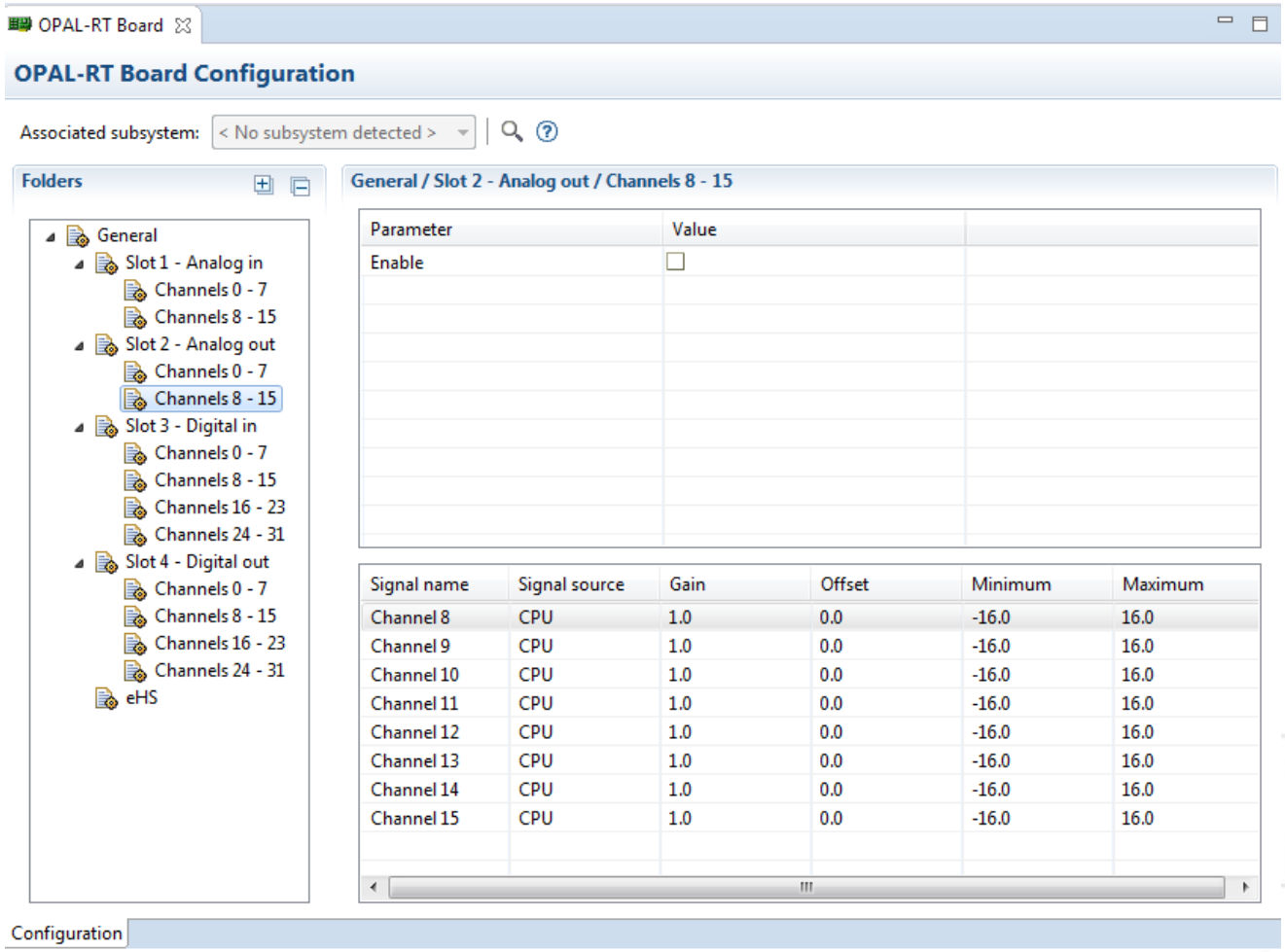
Channels 0 - 7 view



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

Channels 8 - 15 view



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

In the channels view the user has the option of:

- Enabling or disabling the use of the group of 8 channels in the simulation
- Setting parameters for each signal: source, gain, offset, minimum and maximum
- The signal source parameter for each channel is a particularity of the analog output block found in the bitstream:
  - The sources of the analog output channels of the module can be either coming from the Simulink model or from logic internal to the bitstream (such as an eHS block)

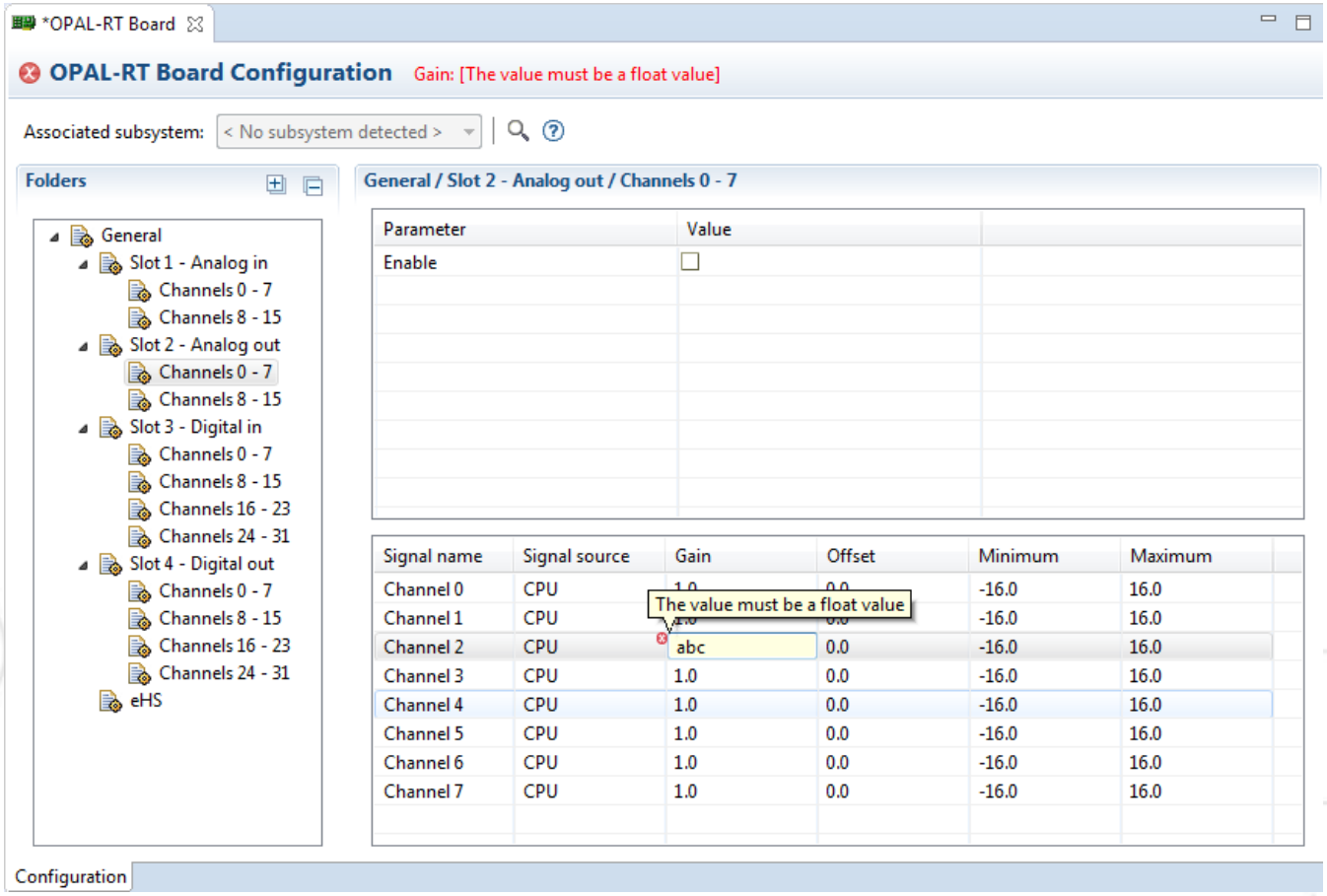
For more information on the Advanced Analog Out configuration, please check the help document, found here: in RT-LAB, click on Help -> Help Contents. In the window that appears, navigate to RT-LAB I/O Block Library Reference Guide -> Opal-RT -> OPAL-RT Board -> Analog Out

Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

Where appropriate, validation will be performed on a parameter to prevent the user from inputting erroneous values

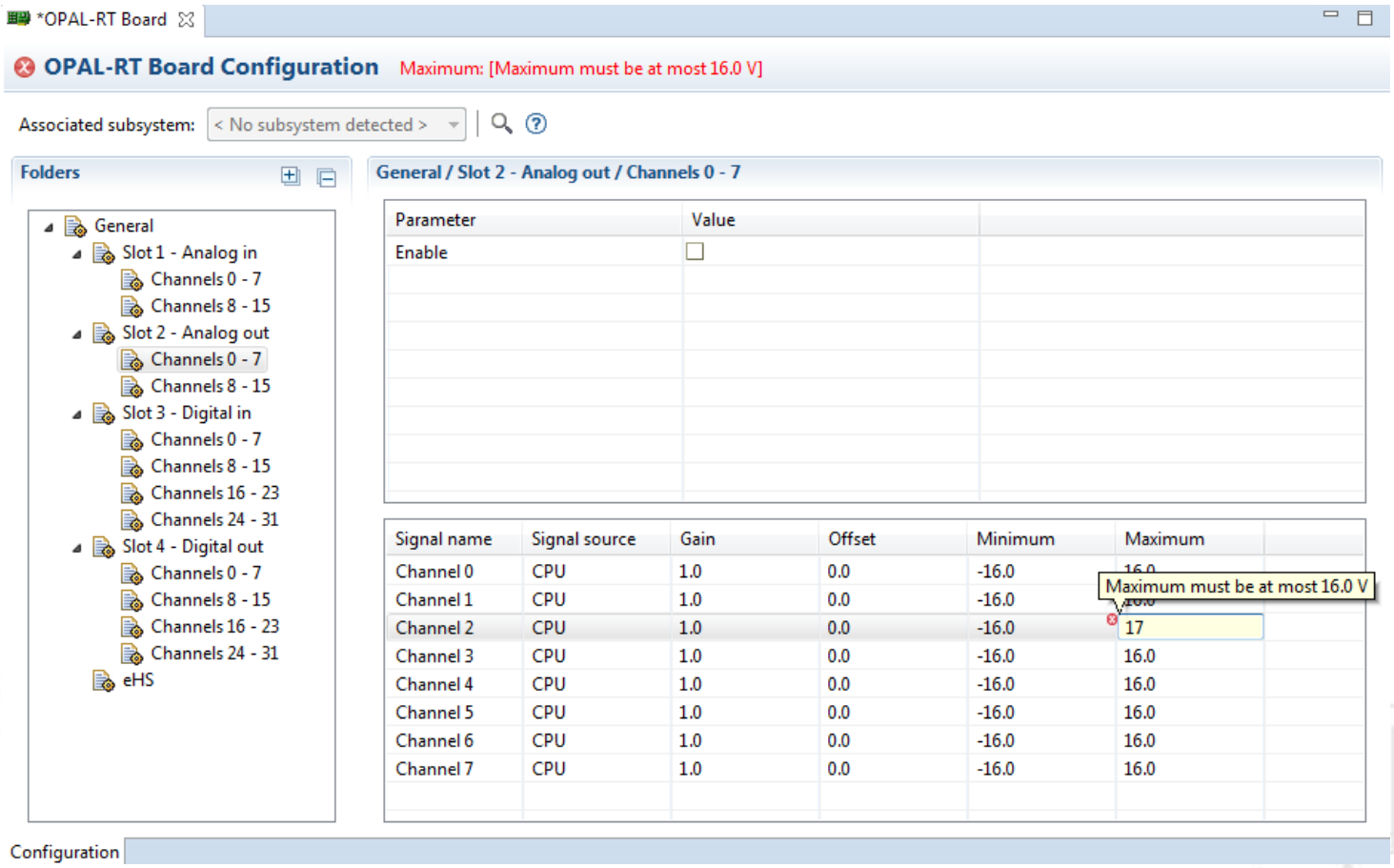
- Examples:



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Analog Out

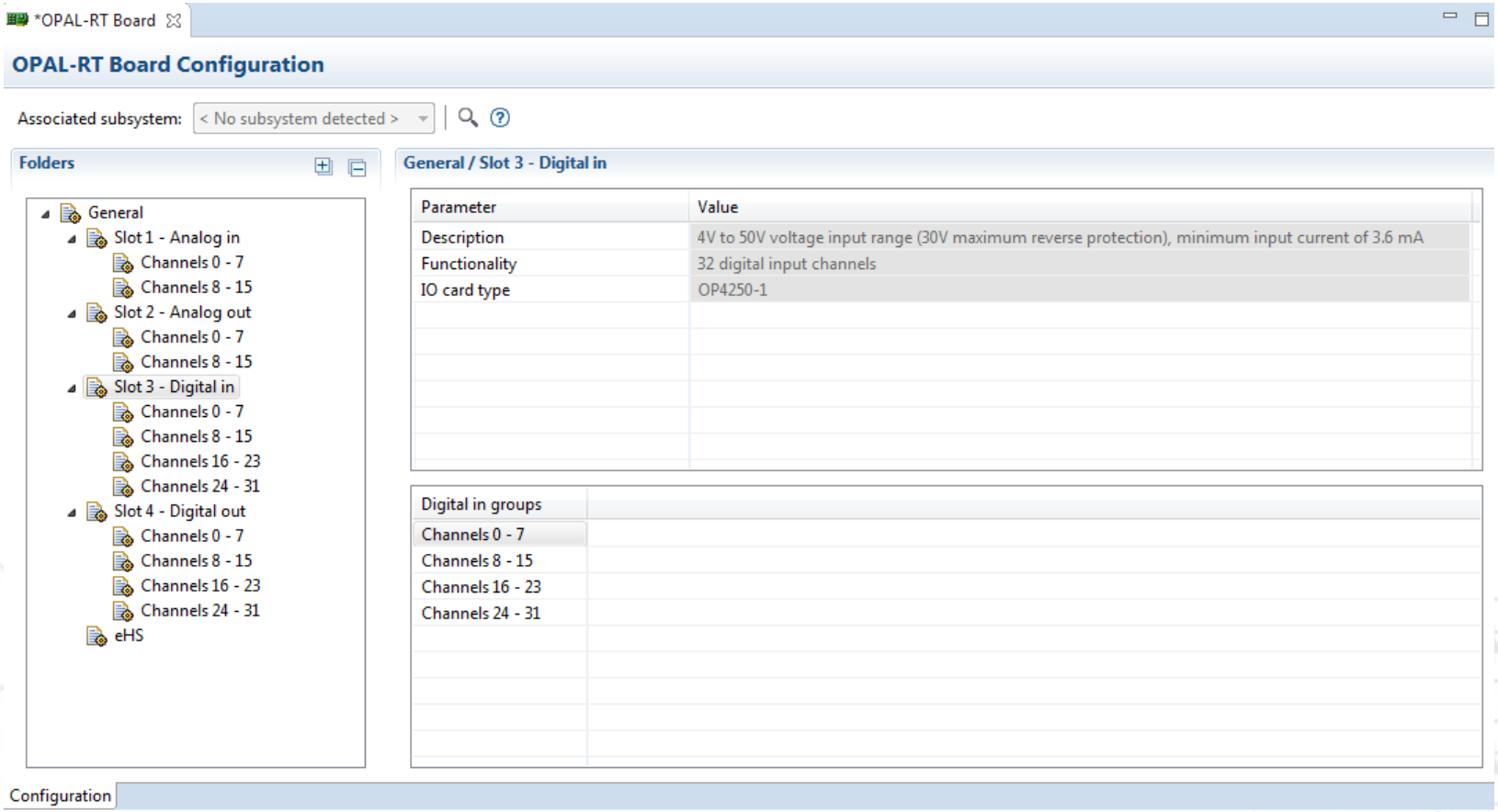
Examples (cont.)



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital In

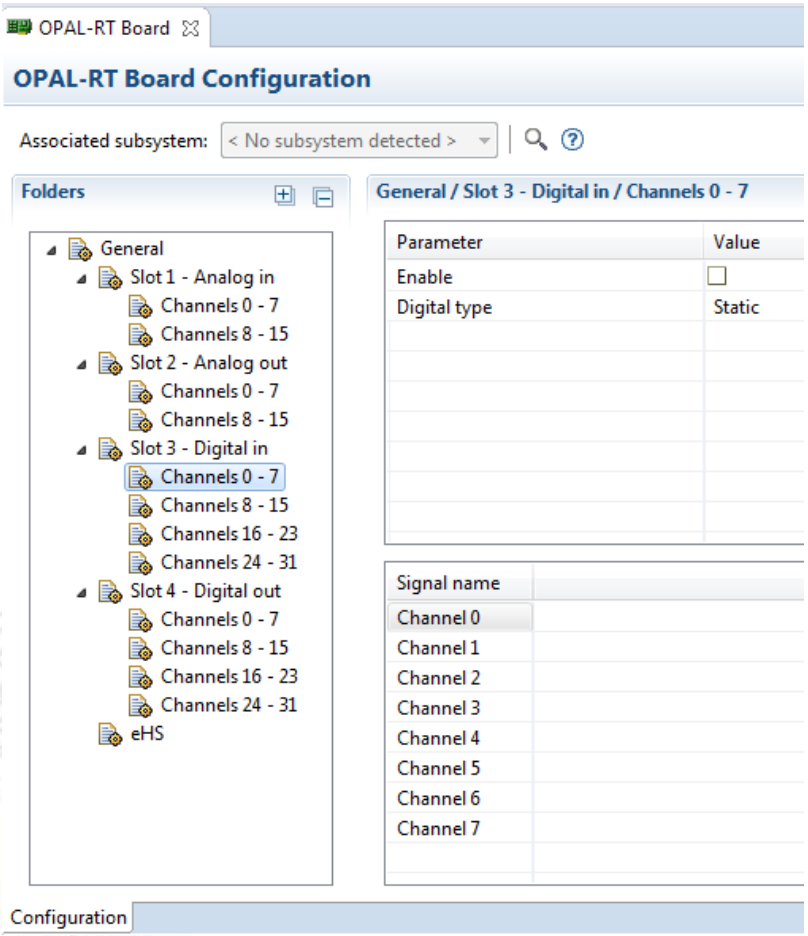
This view shows general information regarding the cassette found in slot 3.
In this scenario, it is a Digital In module.



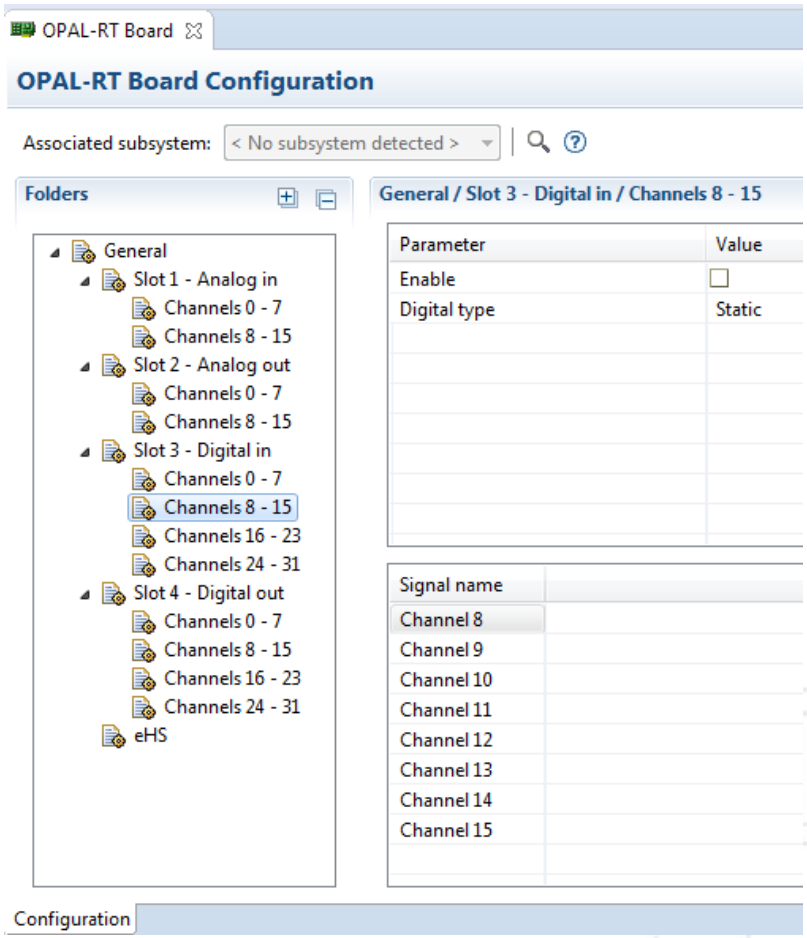
Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital In

Channels 0 – 7 view



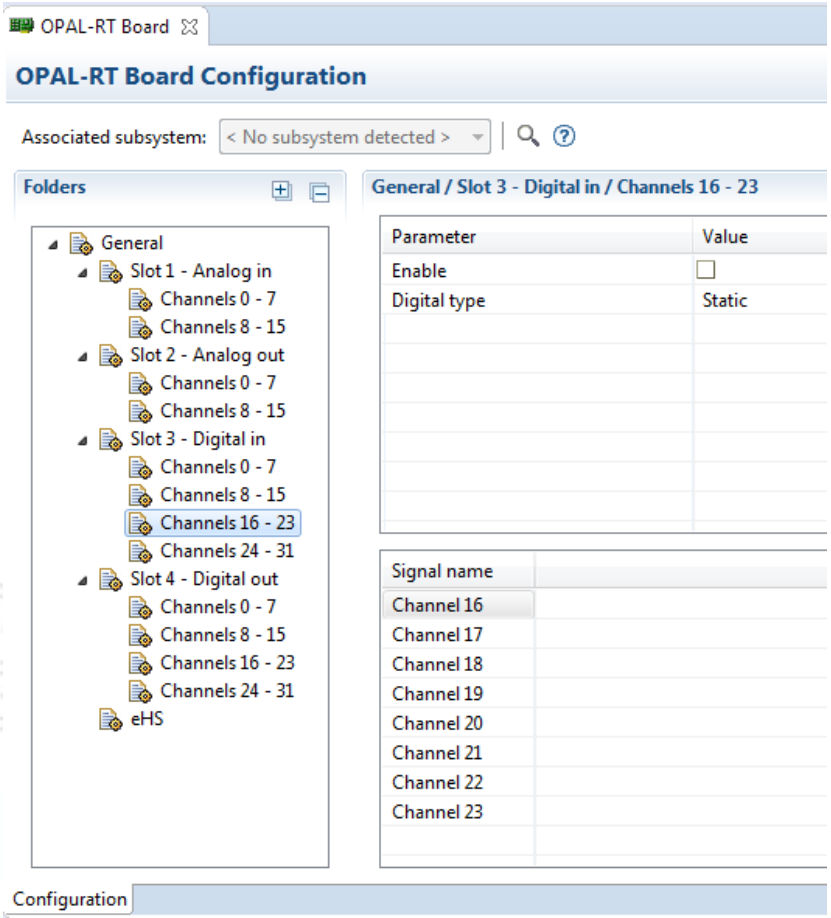
Channels 8 – 15 view



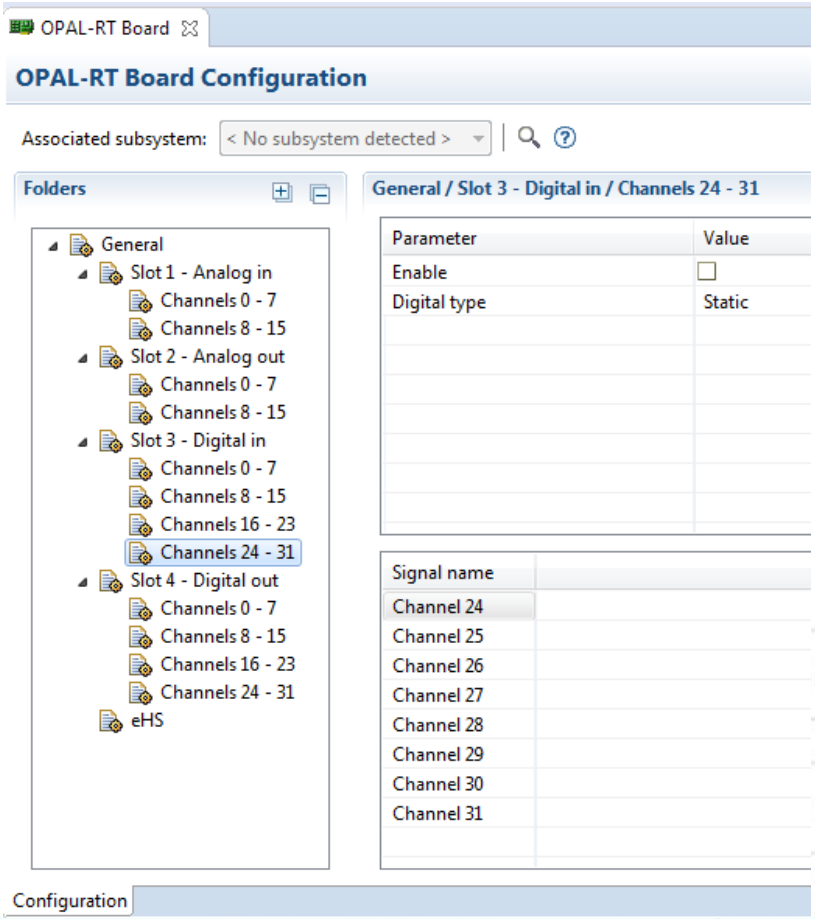
Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital In

Channels 16 – 23 view



Channels 24 – 31 view



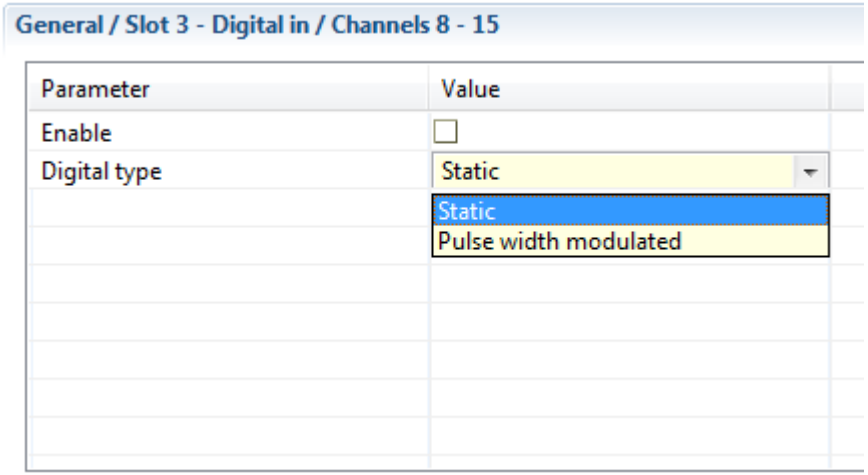
Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital In

In the channels view the user has the option of:

- Enabling or disabling the use of the group of 8 channels in the simulation
- Setting the type of digital signal expected for the 8 channels at hand; current options are:
  - Static: high-voltage values will be interpreted as the value ‘1’ or ‘true’ in the model whereas low-voltages values will be interpreted as ‘0’ or ‘false’
  - Pulse-width-modulated: the values reported in the CPU model will be the frequencies and duty cycles of the digital signals received on the channels of the digital in module

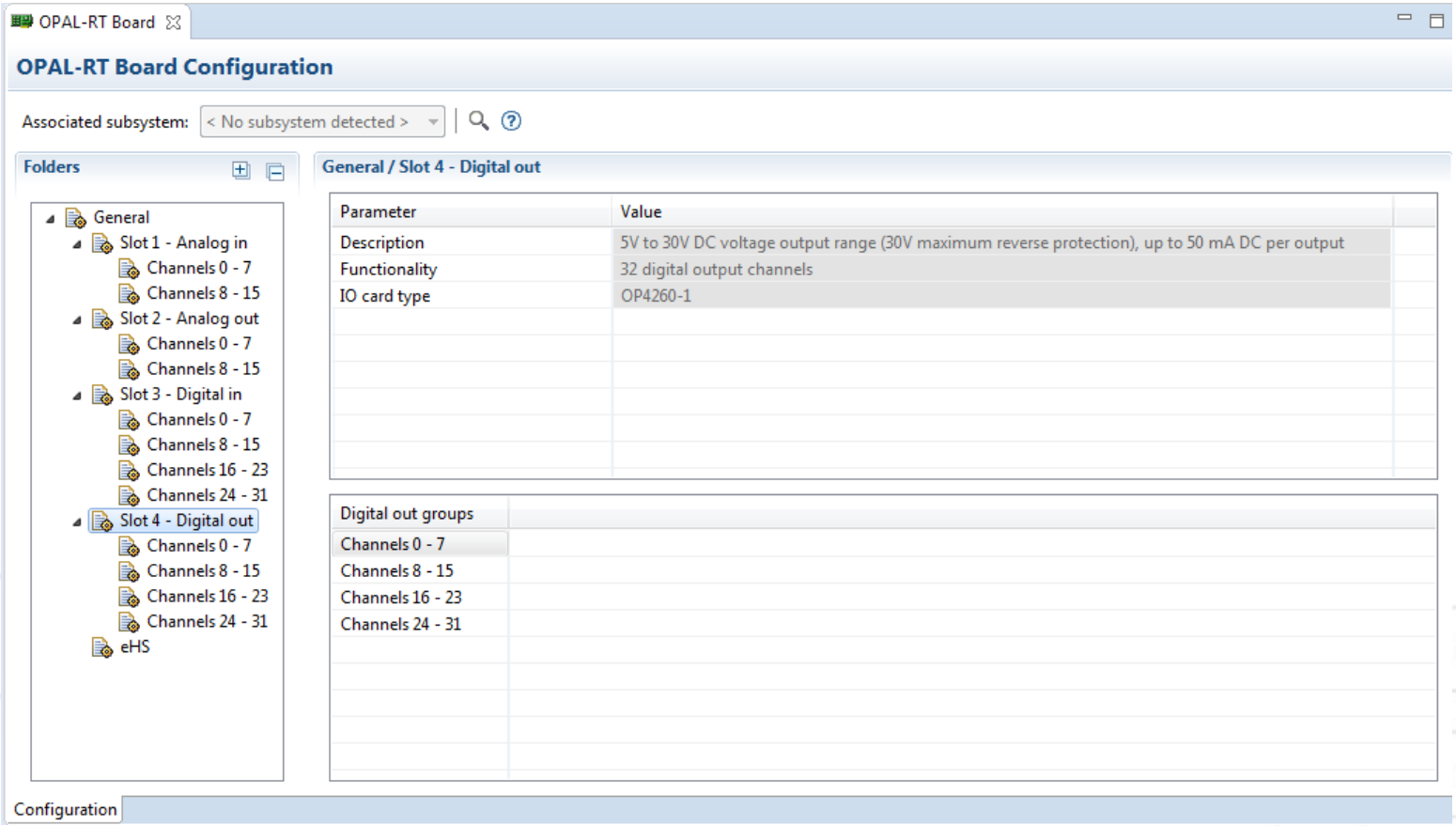
  NOTE: Time-stamped digital is not yet supported for OP4200
- Any combination of static-PWM type groups of channels is possible



Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital Out

This view shows general information regarding the cassette found in slot 4.
In this scenario, it is a Digital Out module.

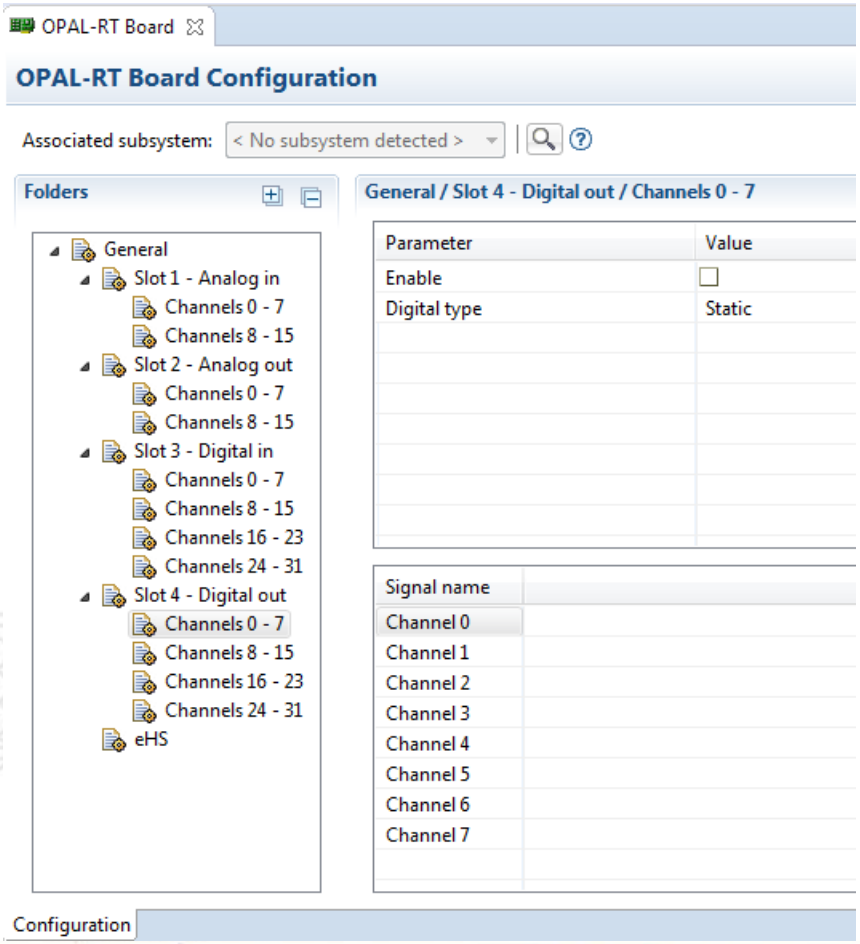


Using IOs with OP4200: the driver

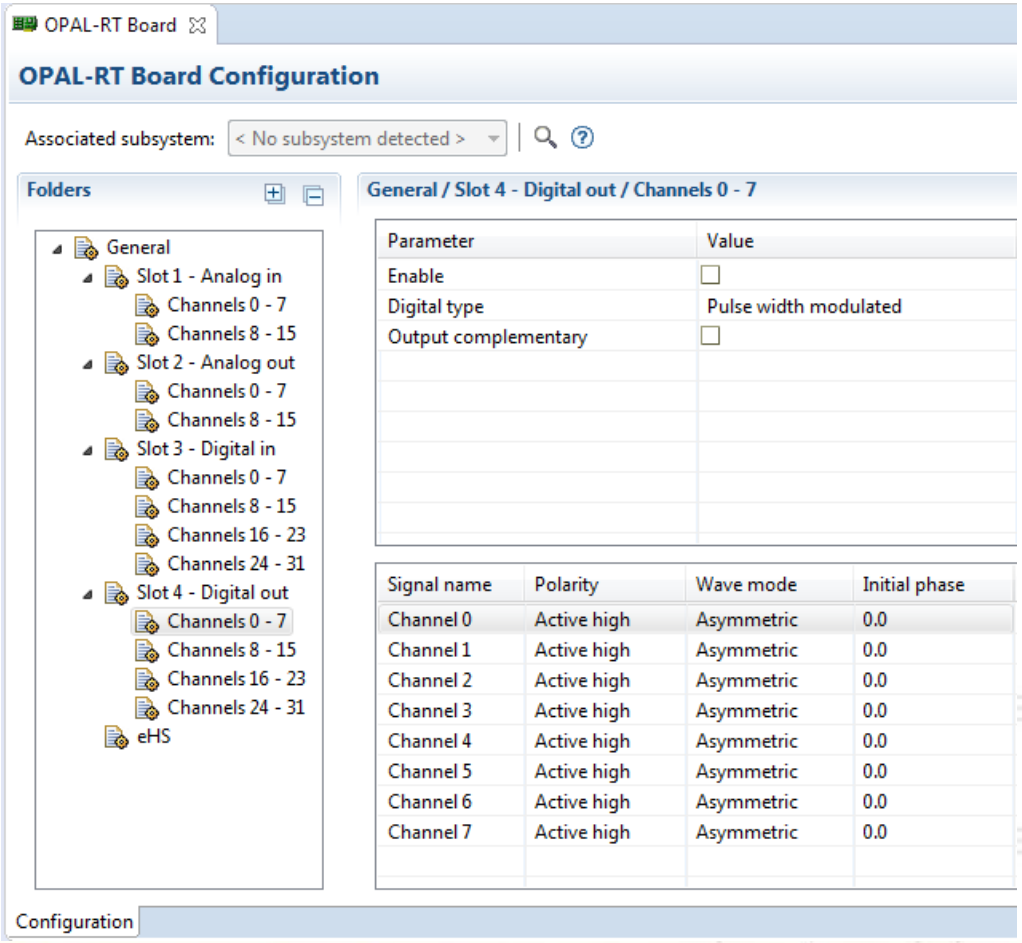
Understanding the driver: configuring the communication (IOs) – Digital Out

Channels 0 – 7 view

as Static digital channels



as Pulse-width-modulated channels

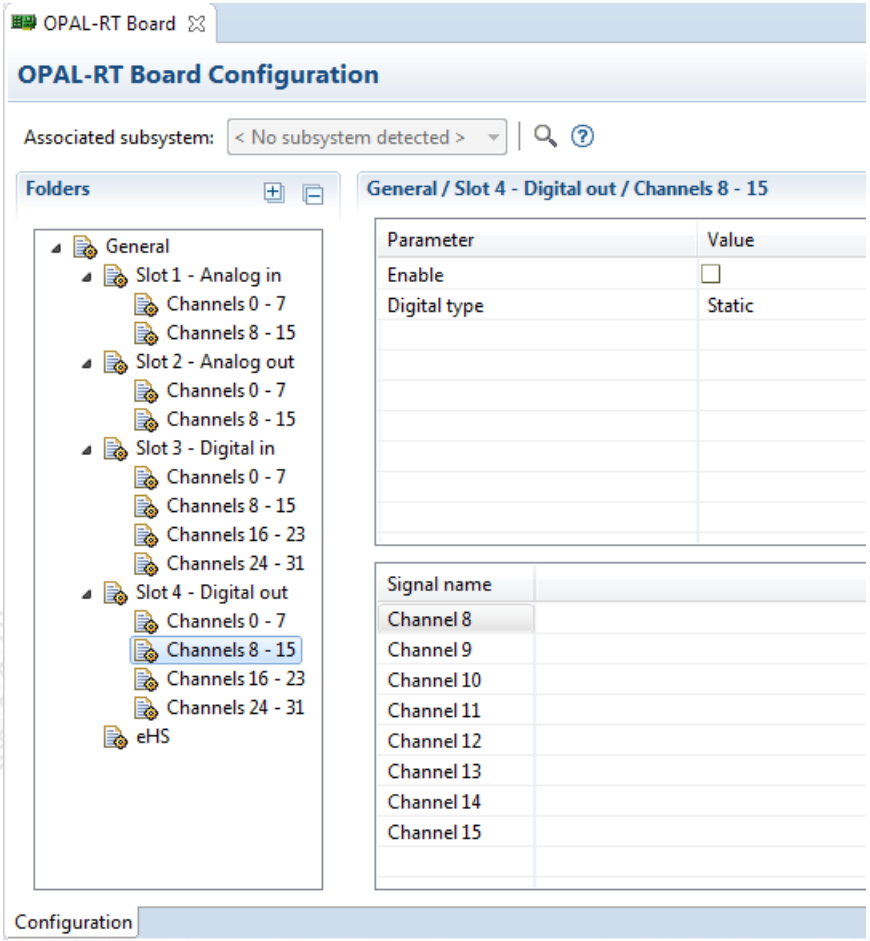


Using IOs with OP4200: the driver

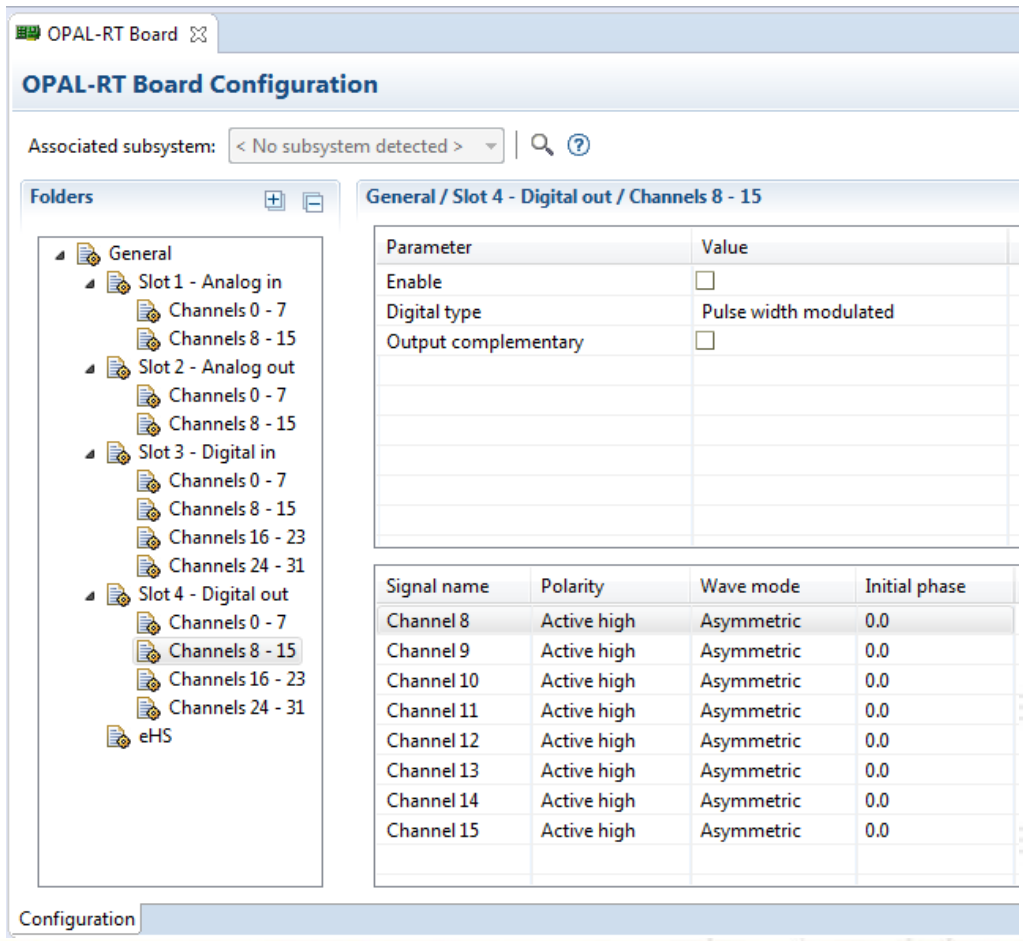
Understanding the driver: configuring the communication (IOs) – Digital Out

Channels 8 – 15 view

as Static digital channels



as Pulse-width-modulated channels

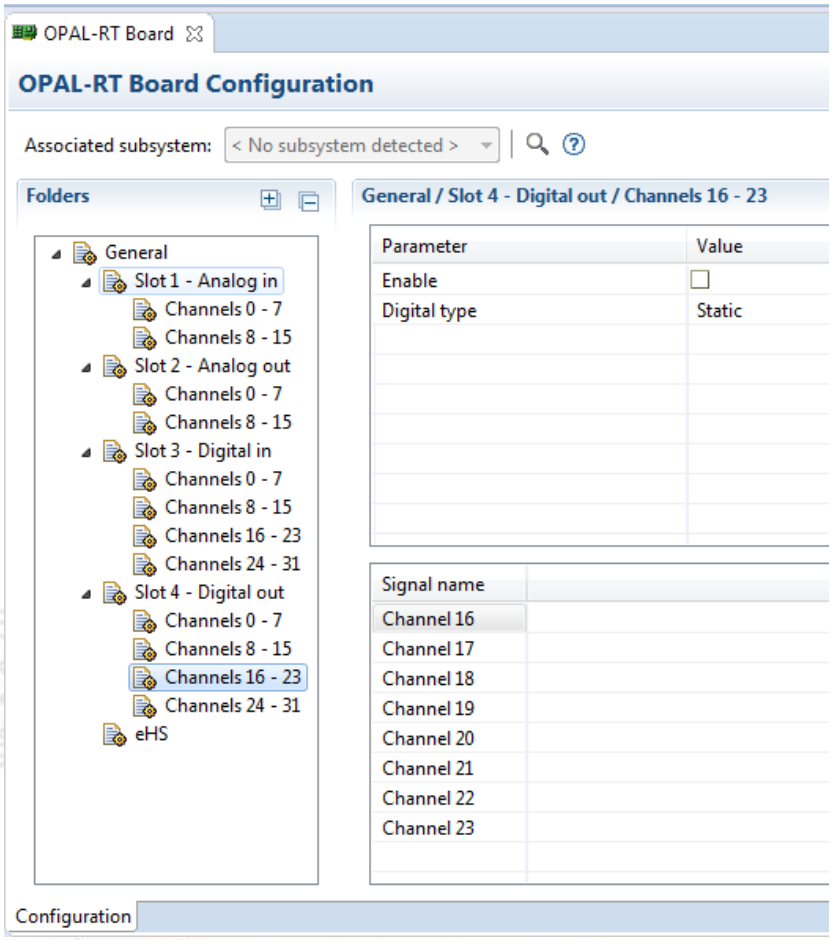


Using IOs with OP4200: the driver

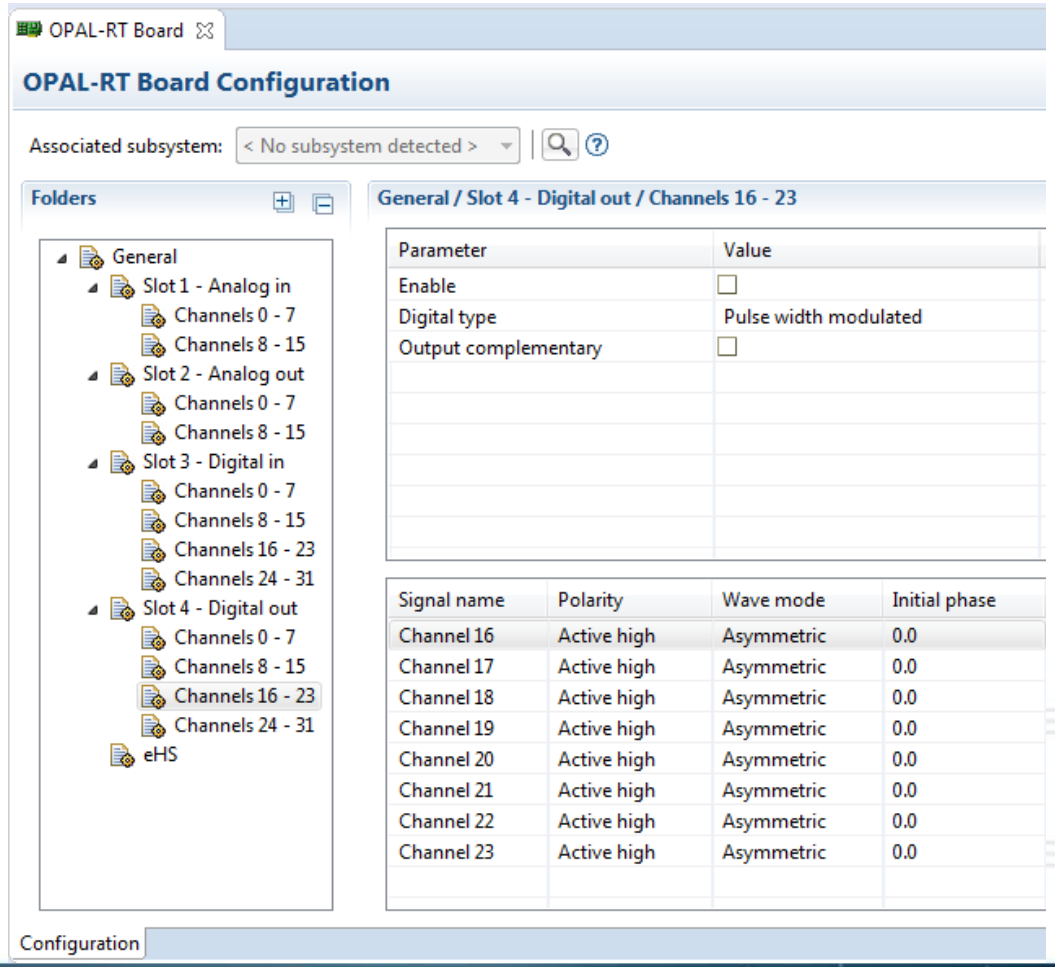
Understanding the driver: configuring the communication (IOs) – Digital Out

Channels 16 – 23 view

as Static digital channels



as Pulse-width-modulated channels

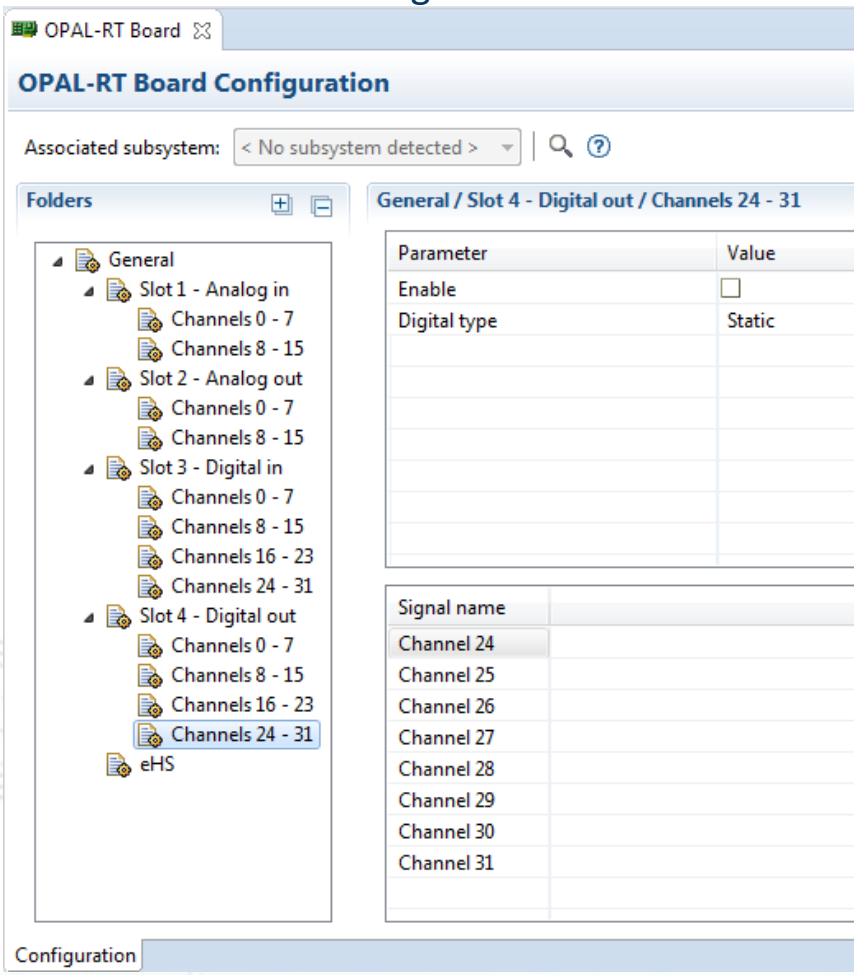


Using IOs with OP4200: the driver

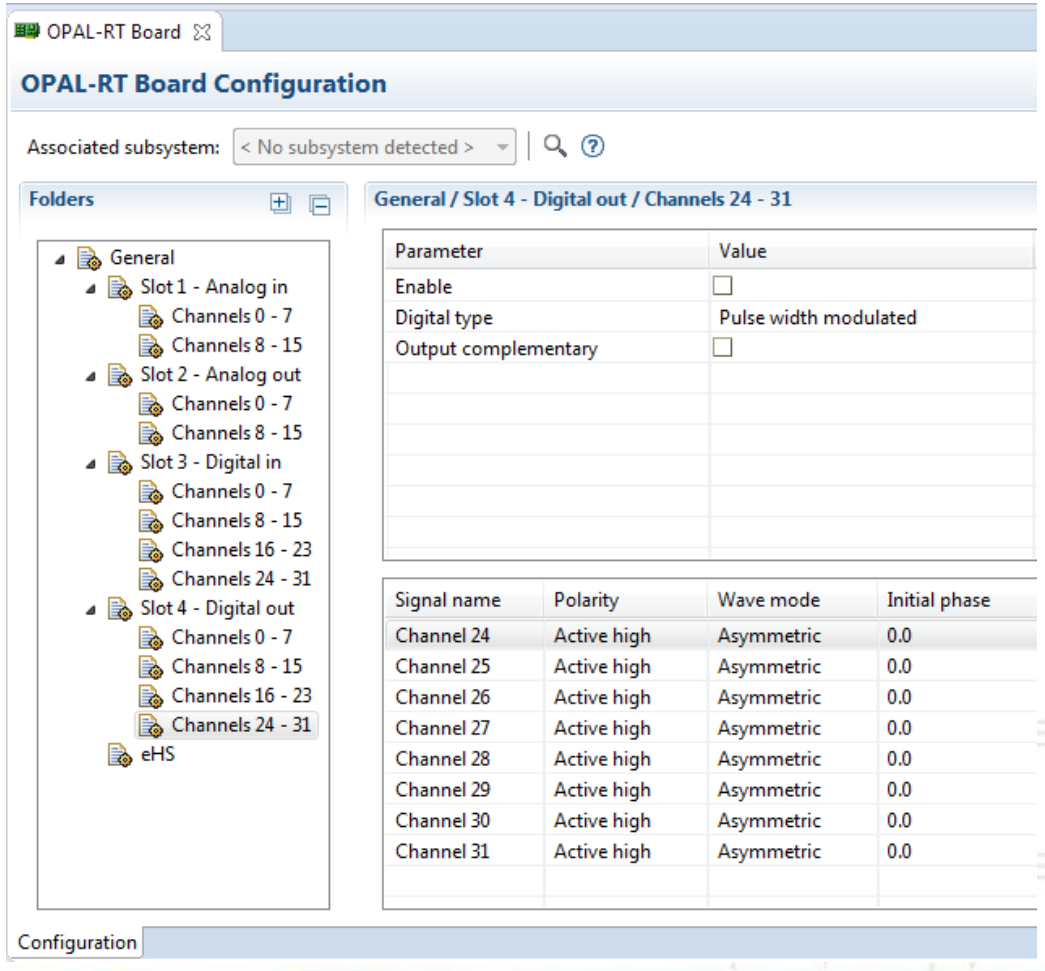
Understanding the driver: configuring the communication (IOs) – Digital Out

Channels 24 – 31 view

as Static digital channels



as Pulse-width-modulated channels



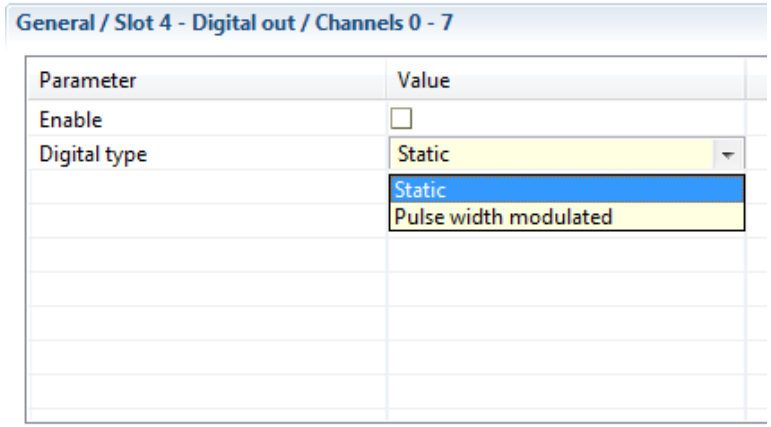
Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital Out

In the channels view the user has the option of:

- Enabling or disabling the use of the group of 8 channels in the simulation
- Setting the type of digital signal to be sent for the 8 channels at hand; current options are:
  - Static: the value ‘1’ sent from the CPU model will be outputted as high-voltage whereas the value ‘0’ will be outputted as low-voltage
  - Pulse-width-modulated: the values expected by the FPGA from the CPU model will be the frequencies and duty cycles of the digital signals to be outputted on the channels of the digital out module

  NOTE: Time-stamped digital is not yet supported for OP4200
- Any combination of static-PWM type groups of channels is possible



Understanding the driver: configuring the communication (IOs) – Digital Out

Particularities of the Pulse-width-modulated output type

When the Pulse width modulated option is selected from the drop-down menu seen in the previous slide new options appear.

- Output complementary: enabling this option will cause every channel's complementary value to be outputted on the adjacent physical channel

- The user will also have the option of setting each channel's polarity (from a drop-down menu), wave mode (from a drop-down menu) and initial phase

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
Channel 0	Active high	Asymmetric	0.0
Channel 1	Active high	Asymmetric	0.0
Channel 2	Active high	Asymmetric	0.0
Channel 3	Active high	Asymmetric	0.0
Channel 4	Active high	Asymmetric	0.0
Channel 5	Active high	Asymmetric	0.0
Channel 6	Active high	Asymmetric	0.0
Channel 7	Active high	Asymmetric	0.0

Using IOs with OP4200: the driver

Understanding the driver: configuring the communication (IOs) – Digital Out

Particularities of the Pulse-width-modulated output type (cont.)

- By enabling the *Output complementary* option every channel's complementary value will be outputted on the adjacent physical channel
- This is shown to the user by graying out the channels that will be used for complementary values.
- Furthermore, a new column is added to the signals list, the *Dead time*

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

For more information on the PWM digital output configuration, please check the help document, found here: in RT-LAB, click on Help -> Help Contents. In the window that appears, navigate to RT-LAB I/O Block Library Reference Guide -> Opal-RT -> OPAL-RT Board -> PWM Out

Using IOs with OP4200: the model

Understanding the model

Given that the driver controlling the IOs of the OP4200 is no longer part of the Simulink context, it employs a different way to extract values to be sent and/or to report values received back to the model.

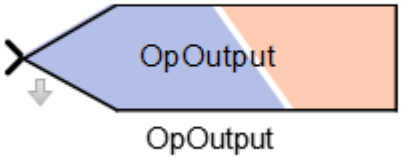
OPAL-RT Board is using the OpInput and OpOutput blocks. However, these blocks are driver independent, meaning any DiL driver needs to be used with them if there are values to be exchanged between the Simulink model and the hardware controlled by the driver in question.

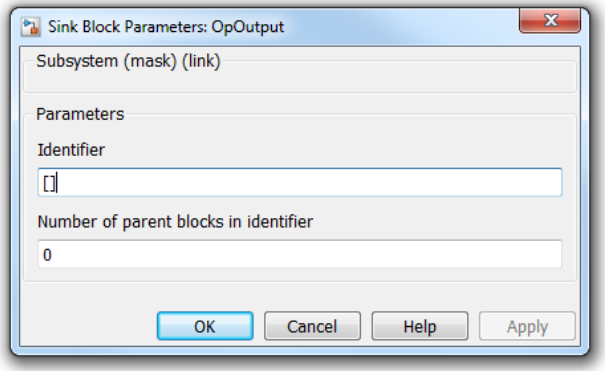
Modules 2 and 3 of this training will give more detail on how to use the OpInput and OpOutput blocks with the *OPAL-RT Board* driver.

Using IOs with OP4200: the model

Understanding the model

- Every signal that is meant to be transmitted to the FPGA (which in turn will output it through the analog or digital output modules) needs to be routed through an OpOutput block:





- Similarly, every signal that is meant to be received from the FPGA (which in turn will receive it through the analog or digital input modules) needs to be routed through an OpInput block: