OPAL-RT TECHNOLOGIES

FROM IMAGINATION TO REAL-TIME

Using IOs with OP4200: Hands-on step-by-step walkthrough
Sending data

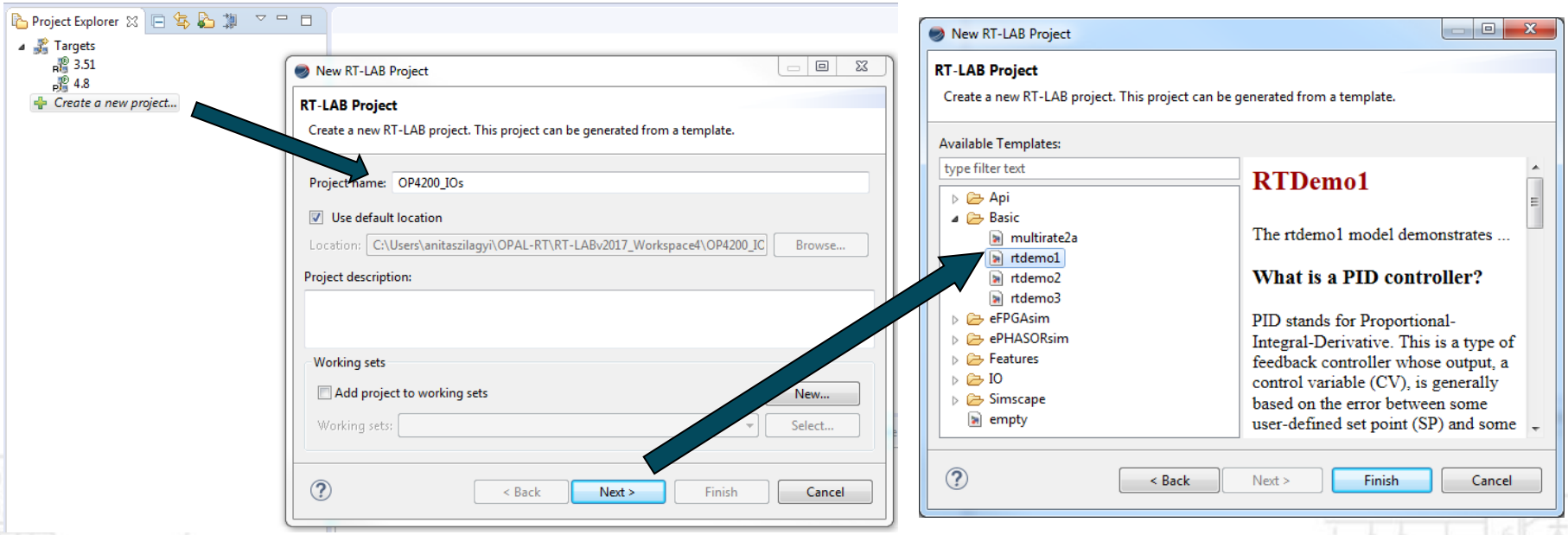
Drivers Team

Using IOs with OP4200: sending values

Sending an analog value: model side

We will use the RTDemo1 template to make the creation of the model easier. You can very easily apply the same steps in a blank model if needed. This tutorial assumes that you have the knowledge of how to create and edit RT-LAB models.

- Create a new RT-LAB project using the RTDemo1 template

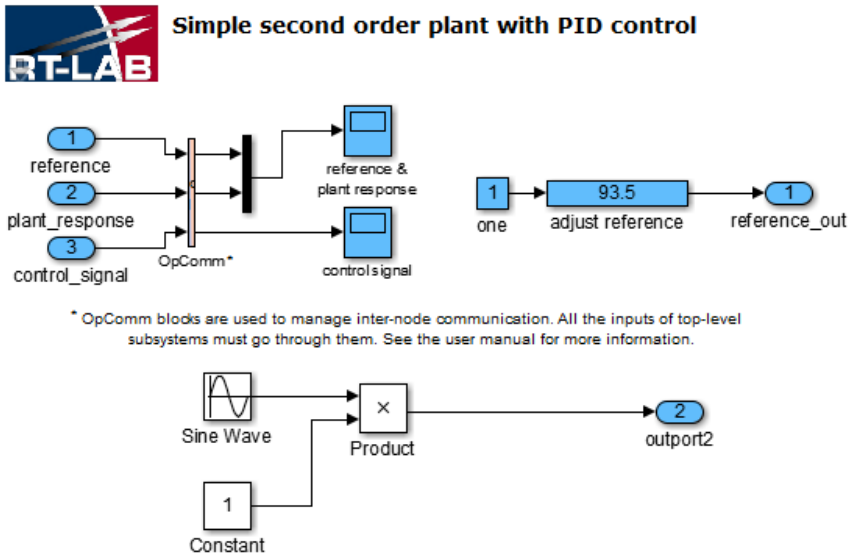


- Click on *Finish* once done

Using IOs with OP4200: sending values

Sending an analog value: model side

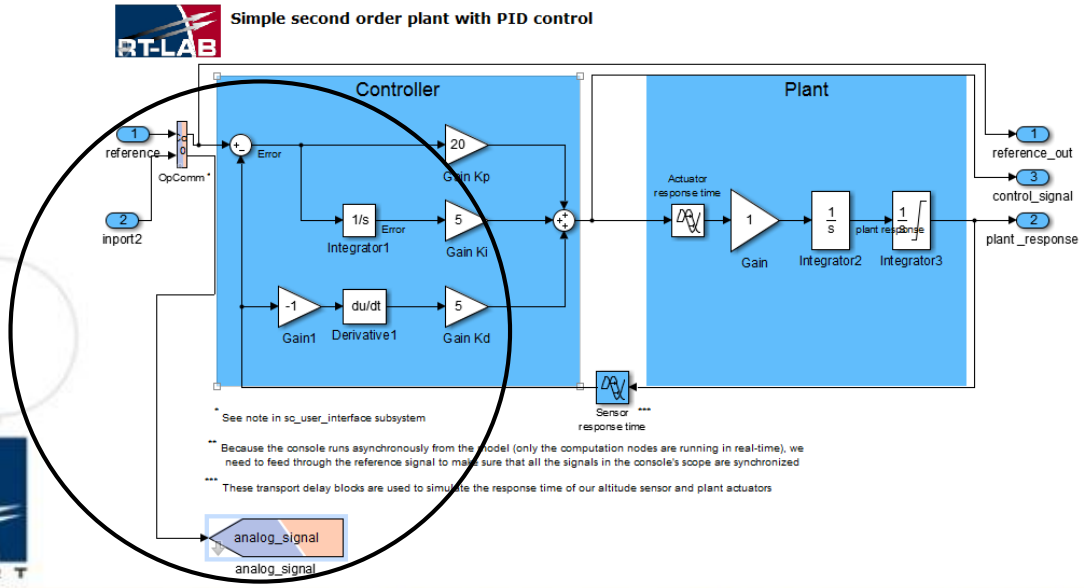
- Edit the model you just added as such:
  - Go into the sc_user_interface subsystem
    - You can leave the current logic untouched, we will not be needing it
    - Add a sine wave and a constant block (both found in Simulink's Library Browser -> Sources) to the model
    - Configure the frequency of the sine wave block to 2*pi
    - Add a product block (found in Library Browser -> Simulink -> Math Operations) and connect its inputs to the outputs of the sine wave block and the constant block
    - Connect the output of the product block to a new outport
    - Your subsystem should look as such:



Using IOs with OP4200: sending values

Sending an analog value: model side

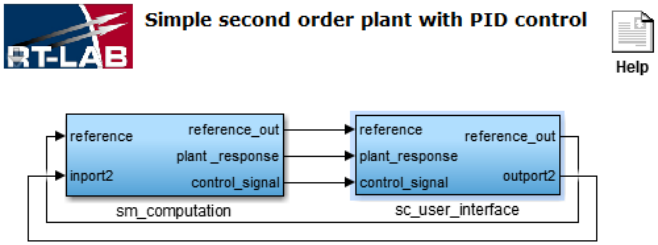
- Edit the model you just added as such:
  - Go into the sm_computation subsystem
    - You can leave the current logic untouched, we will not be needing it
    - Double-click on the OpComm block to reveal its parameters
    - Change the number of inports to 2 then click OK
    - Add a new inport and connect it to the newly created second input of the OpComm block
    - Add an OpOutput block from Library Browser -> RT-LAB
    - Name the OpOutput block as you wish, but make sure to change its label to the same name
    - Connect the OpOutput block to the newly created second output of the OpComm block
    - Your subsystem should look as such:



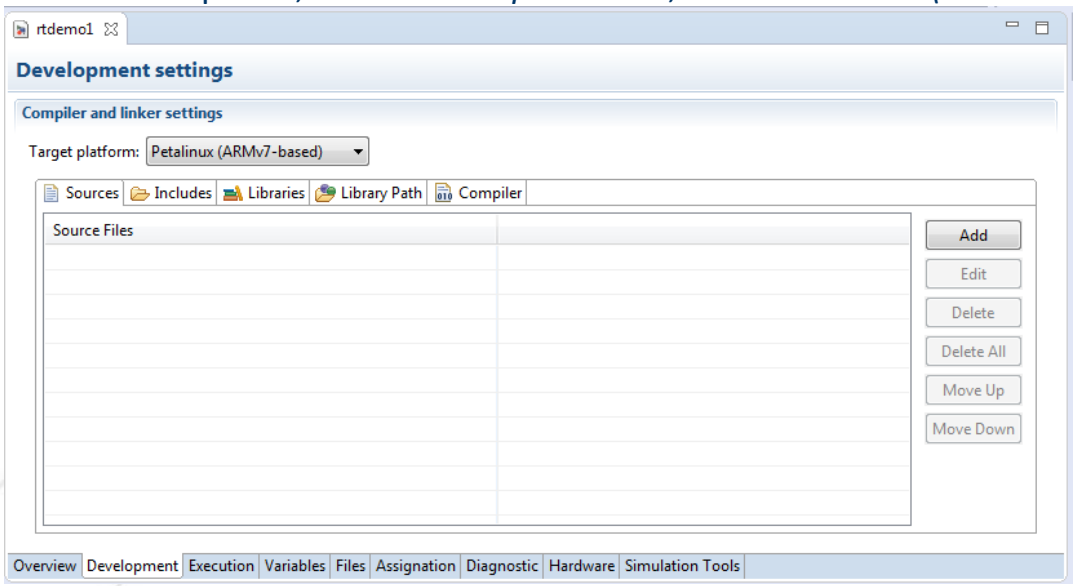
Using IOs with OP4200: sending values

Sending an analog value: model side

- Edit the model you just added as such:
  - Go into the top level view of the model
    - Connect the new output of the sc_user_interface to the new input of the sm_computation
- Save the model



- In the model options, in the *Development* tab, select *Petalinux (ARMv7-based)* as the target platform



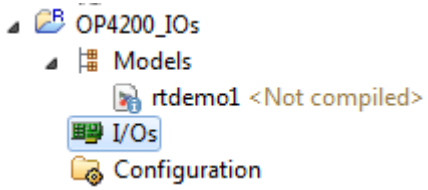
- Compile the model

Using IOs with OP4200: sending values

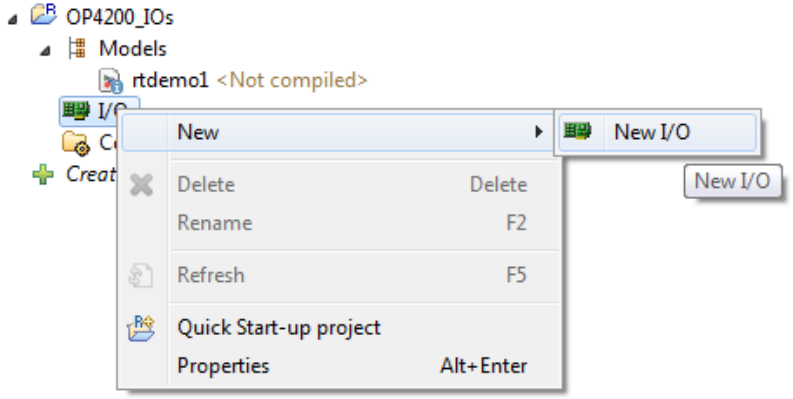
Sending an analog value: driver side

- In the same project where the model was created, add the *OPAL-RT Board* driver

1. Initial view of the project



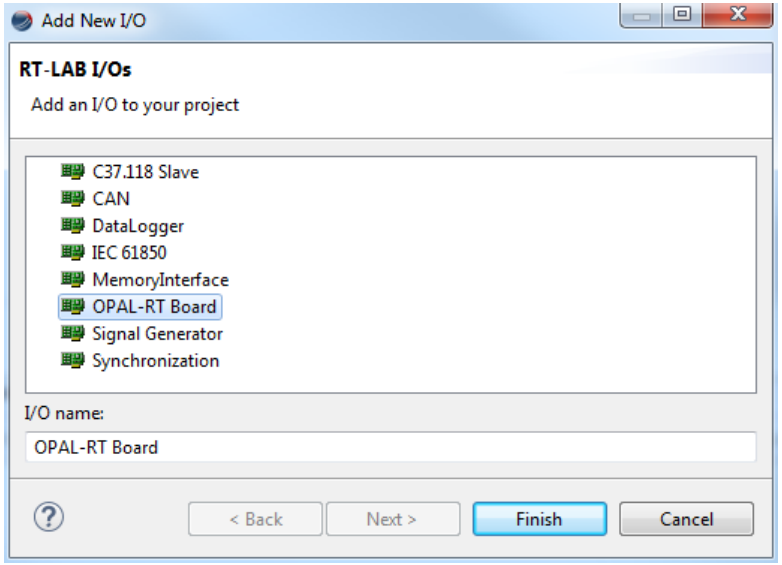
2. Adding a new I/O



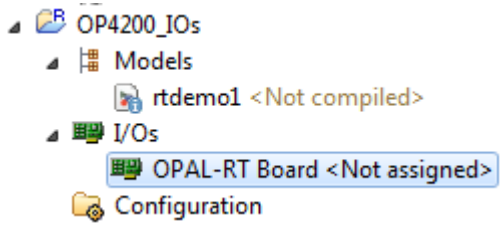
Using IOs with OP4200: sending values

Sending an analog value: driver side

3. Select OPAL-RT Board from the list



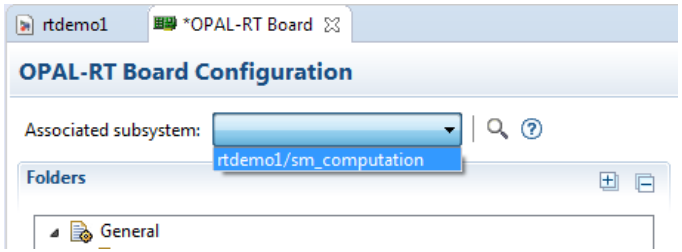
4. Clicking on “Finish” will add it to the project



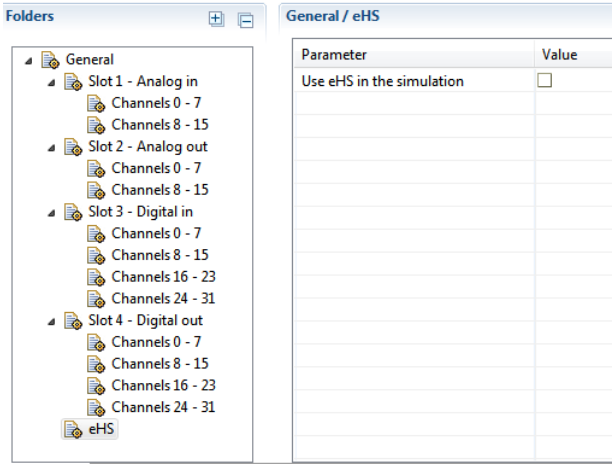
Using IOs with OP4200: sending values

Sending an analog value: driver side

- In the *Bitstream configuration* drop-down list select the AX-0001-3_1_2_360-eHSgen3_withIOs-21-17 configuration. This configuration relates to the MEZX5_AX-0001-3_1_2_360-eHSgen3_withIOs-21-17.bin bitstream
- Associate the driver to the master subsystem of the model that was just built:



- Click on *eHS* and untick the *Use eHS in the simulation checkbox*; testing eHS is beyond the scope of this tutorial

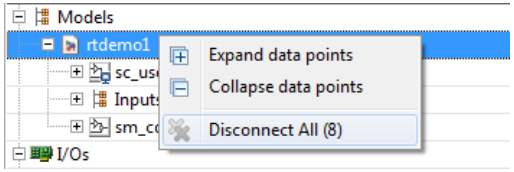


- Click on *Slot 2 – Analog Out / Channels 0 – 7*
- Tick the *Enable* check-box; this will enable the use of the first 8 channels of the analog out module in the simulation
- Save the driver configuration by typing Control + S or by clicking on the save button under the menu bar

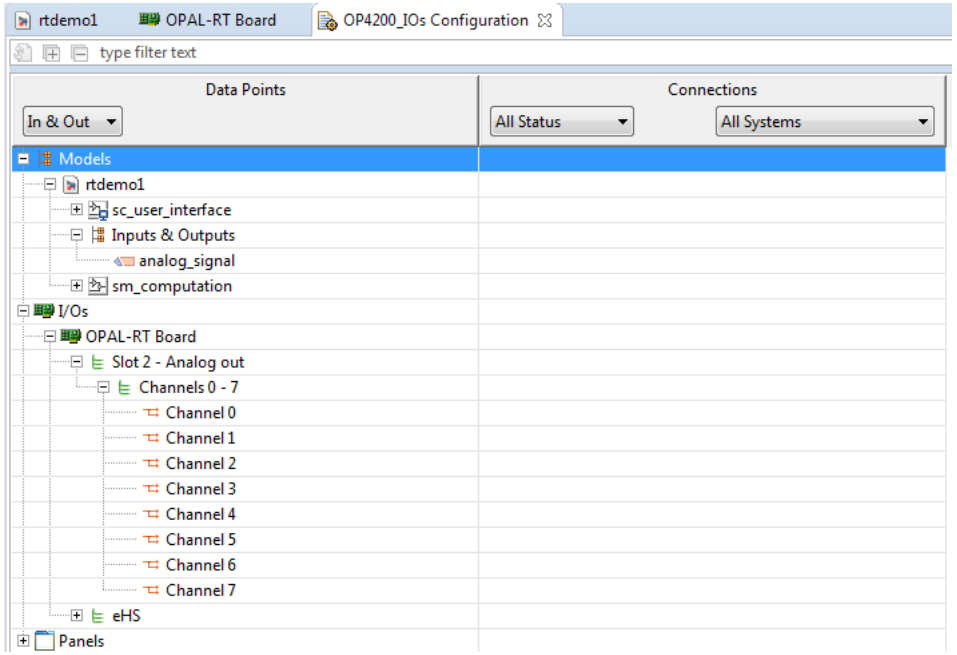
Using IOs with OP4200: sending values

Sending an analog value: connecting the model to the driver

- In the same project where the model was created, double click on *Configuration*
- Once opened, expand Models to reveal rtdemo1
- Right-click on rtdemo1 and select *Disconnect all;* this step is necessary to remove any example-type connections that come with the RTDemo1 project in order to allow us to focus entirely on the OP4200 connections:



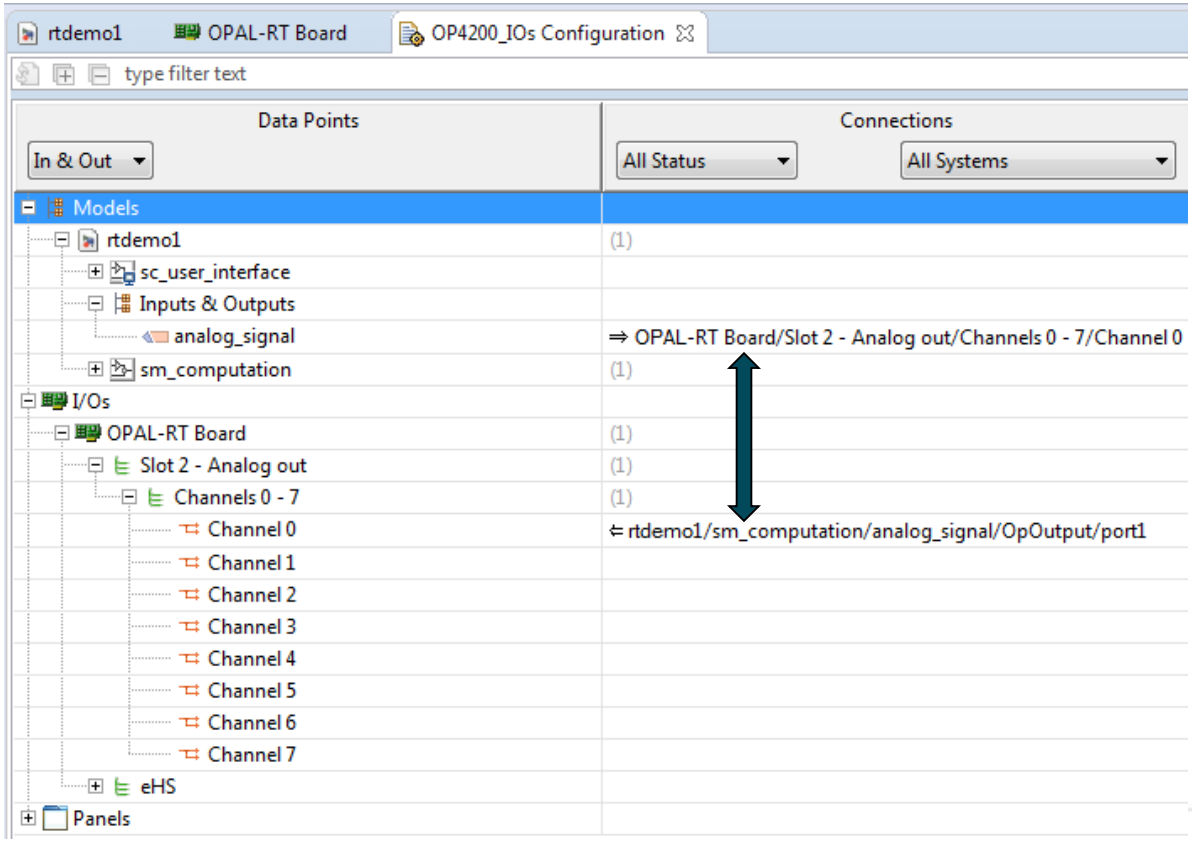
- Expand *rtdemo1 -> Inputs & Outputs* and *I/Os -> OPAL-RT Board -> Slot 2 – Analog out -> Channels 0 – 7*; final look:



Using IOs with OP4200: sending values

Sending an analog value: connecting the model to the driver

- In order for the values produced by the sine wave block in the model to be sent through the physical channel 0 of the analog out module, you need to drag and drop the OpOutput block onto Channel 0
- If you prefer to use another physical channel to output the sine wave, drag and drop the OpOutput onto the respective channel
- Final look:



Using IOs with OP4200: sending values

Sending an analog value: running the simulation

- Load the model
- Check the RT-LAB display to make sure that the connection was done successfully:

```
Connection successful: sm_computation/analog_out/OpOutput/port1 to OPAL-RT Board/Slot 2 - Analog out/Channels 0 - 7/Channel 0
```

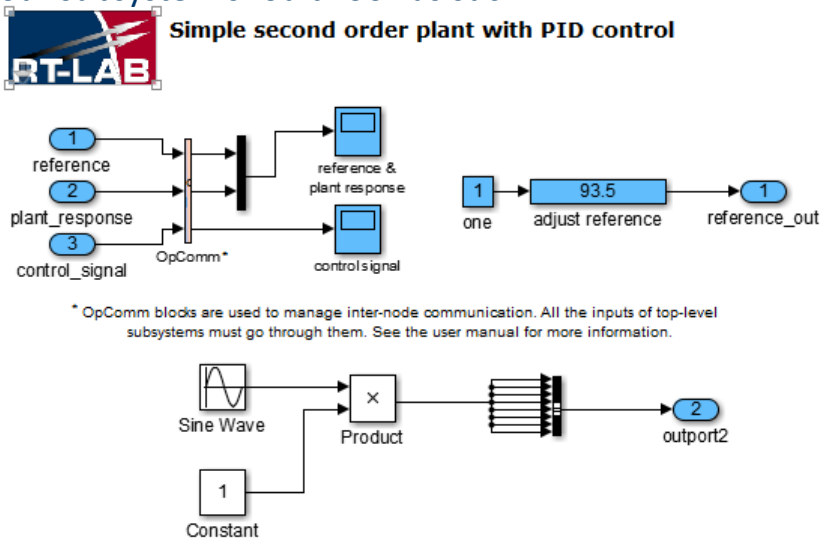
- Run the model
- In the console, change the value of the constant block to change the amplitude of the sine wave outputted on channel 0 of the analog output cassette found in slot 2 of the OP4200

NOTE: even though we have activated the entire group of 8 channels when we configured the driver (slide 8), only the channel that was connected as per the instructions in the previous slide will output values on the physical channel of the analog output module.

Using IOs with OP4200: sending values

Sending all analog values in a channel group: model side

- Edit the model as such:
  - Go into the sc_user_interface subsystem
    - Remove the signal connecting the product block and the outport associated to it
    - Add a bus creator block (found in Library Browser -> Simulink -> Signal Routing)
    - Set the bus creator to have 8 inputs and connect the output of the product block to all 8 inputs
    - Connect the outport to the output of the bus creator
    - Your subsystem should look as such:

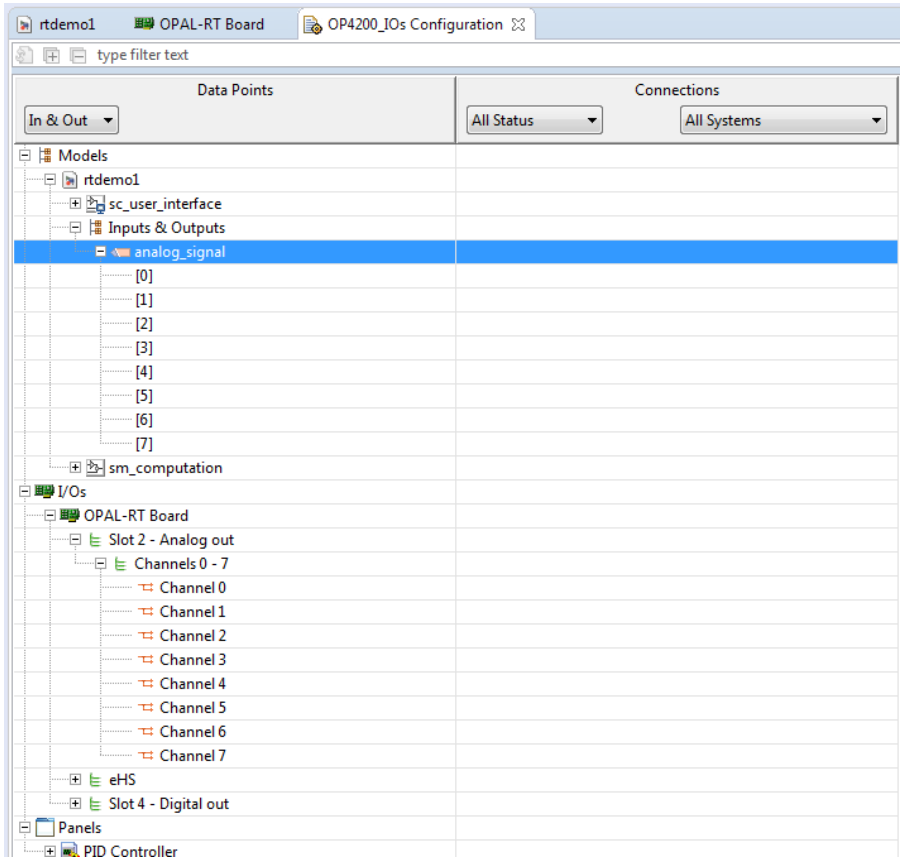


    - Note that this setup will send the same value on all 8 channels of the analog output group; simple adjustments can be made in order to have different values; for instance, see the example model delivered in RT-LAB for OP4200 to get an idea of the implementation

Using IOs with OP4200: sending values

Sending all analog values in a channel group: model side

- No changes need to be made to the computation block
- Save and compile the model
  - The OpOutput block does not need to know if its input is a vector with multiple items or with just one item
  - The vector size will be taken into account after the compilation of the model, in the *Configuration* view of the RT-LAB project:



Using IOs with OP4200: sending values

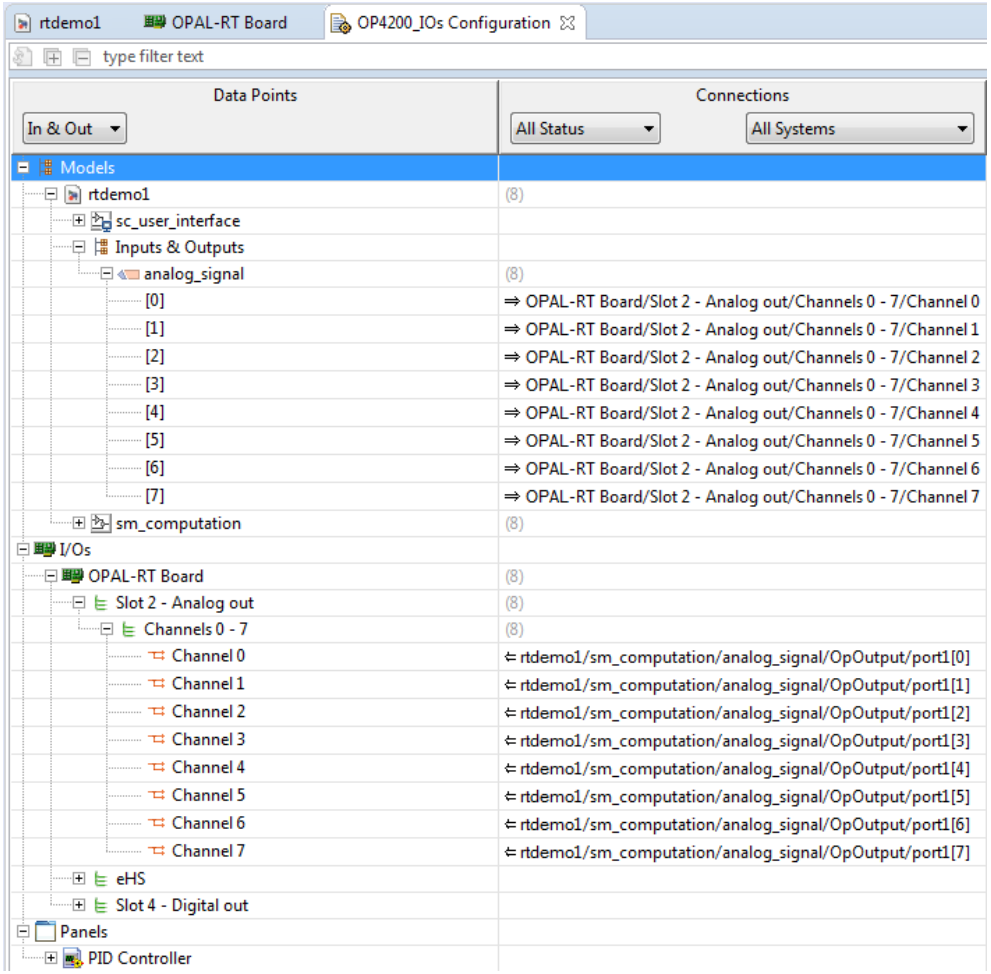
Sending all analog values in a channel group : driver side

- No changes need to be done on the driver side, since in the previous exercise we enabled the group of 8 channels

Using IOs with OP4200: sending values

Sending all analog values in a channel group : connecting the model to the driver

- You can now connect all 8 signals of the compiled OpOutput block to each physical analog output channel
- Alternatively, you can add an OpOutput block for each analog output channel, according to your needs
- Final look:



Using IOs with OP4200: sending values

Sending all analog values in a channel group : running the simulation

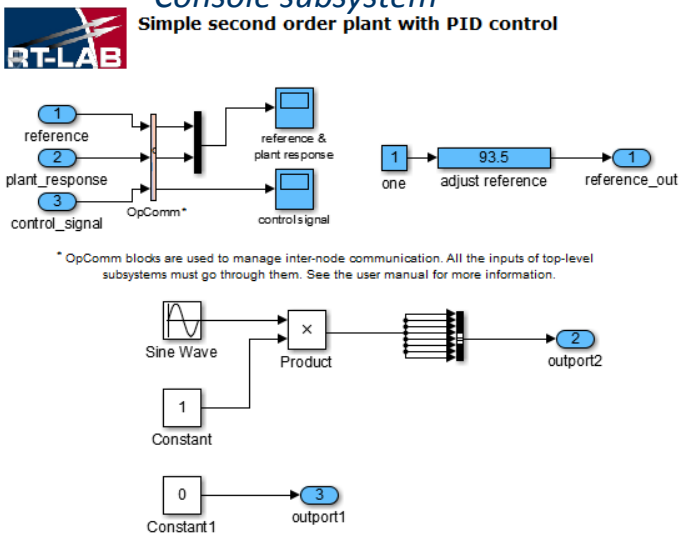
- Since the bitstream has not changed, we can disable the automatic flashing by setting the RT-LAB environment variable “DISABLE_FLASH_UPDATE” to “ON”
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- In the console, change the value of the constant block to change the amplitude of the sine waves outputted on channels 0 through 7 of the analog output cassette found in slot 2 of the OP4200

Using IOs with OP4200: sending values

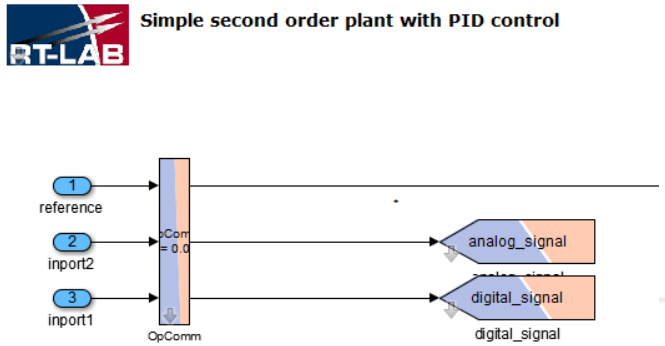
Sending a static digital value

- For the model:
  - Go to the console subsystem; add a constant block to the console subsystem – this will represent the digital value to be sent on a channel of the digital output module of the OP4200
  - Connect the constant block to a new outport
  - Go to the computation subsystem; change the OpComm block to have 3 inputs and outputs
  - Add another OpOutput block and connect it to the third output of the OpComm block
  - Add a new inport and connect it to the third input of the OpComm block
  - Go to the top level model view; connect the new outport of the console subsystem to the new inport of the computation subsystem
  - Save and compile the model

Console subsystem



Computation subsystem



Using IOs with OP4200: sending values

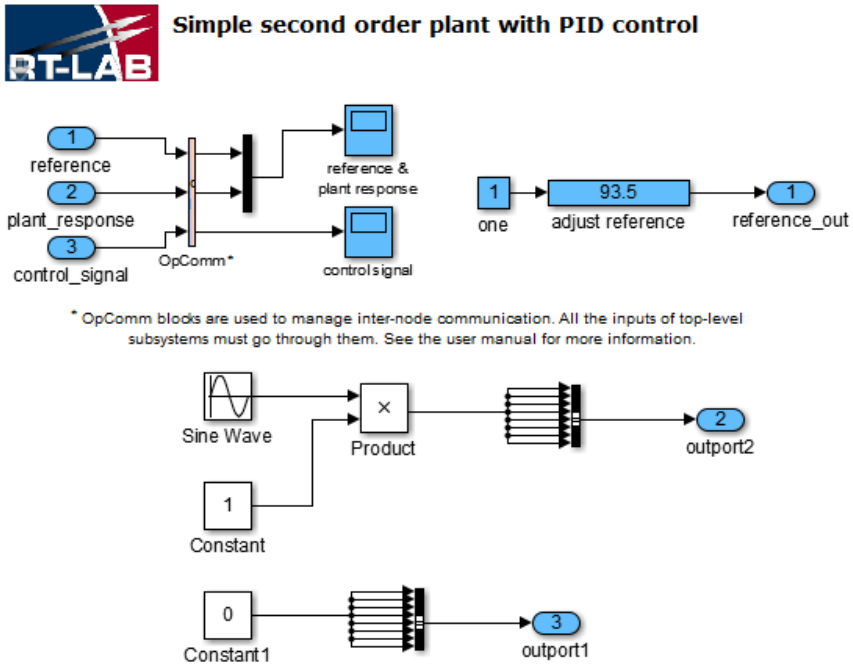
Sending a static digital value

- For the driver:
  - Go to the *Slot 4 – Digital out / Channels 0 – 7* view
  - Enable the group of channels
  - Save the configuration
- In the *Configuration* view, connect the OpOutput block to the digital output channel as you did for the analog output channel (seen on slide 10)
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- Change the value in the new constant block in the console subsystem to toggle the value on the digital output channel 0 (or the one that you chose to make the connection on)

Using IOs with OP4200: sending values

Sending all static digital values in a channel group

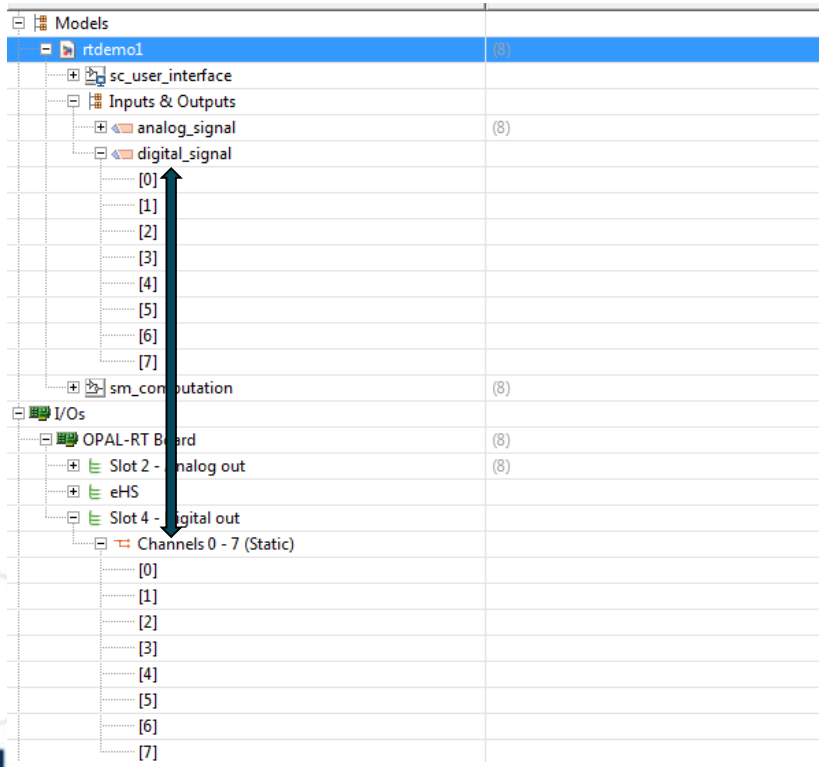
- For the model:
  - Go to the console subsystem; repeat the steps of adding the bus creator as was done for the analog outputs
  - No changes needed for the computation subsystem
  - Save and compile the model



Using IOs with OP4200: sending values

Sending all static digital values in a channel group

- For the driver:
  - No changes needed on the driver side
- Repeat the connection steps as seen on slide 13
  - Alternatively, you can drag-and-drop the entire group of channels onto the OpOutput:



- Doing so will connect signal 0 of the OpOutput to channel 0 of Slot 4 – Digital out / Channels 0 – 7 (Static), signal 1 of the OpOutput to channel 1 and so on.
- This is not possible for the analog out channels because they are of the Advanced Analog Out type.
- Please consult the documentation of the Analog Out functionality of the OPAL-RT Board driver to have a better understanding of why this is the case

Using IOs with OP4200: sending values

Sending all static digital values in a channel group

- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- Change the value in the constant block in the console subsystem to toggle the values on the digital output channels 0 through 7

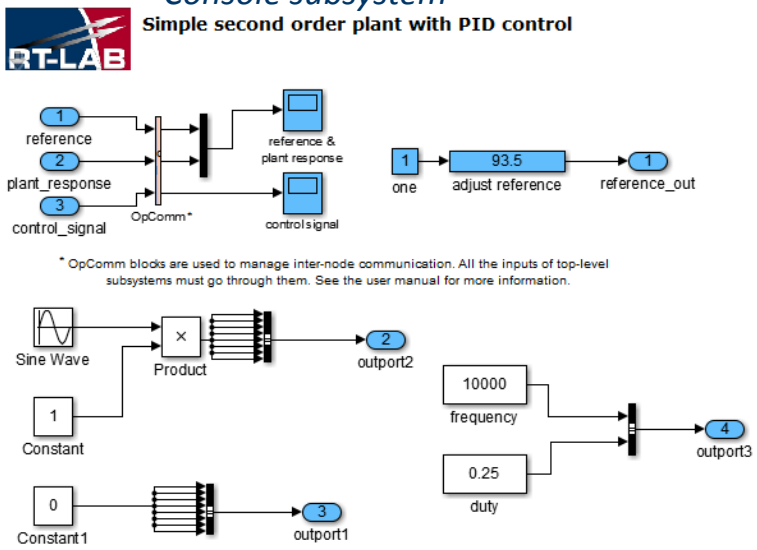
NOTE: for a more complex example of how to send static digital values, please consult the example model for OP4200 delivered with your RT-LAB installation

Using IOs with OP4200: sending values

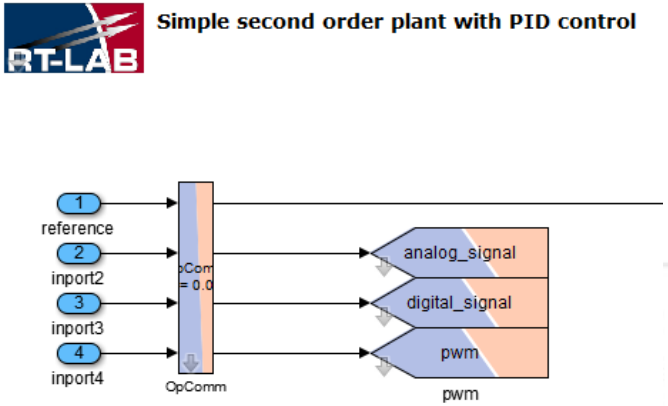
Sending a PWM digital signal

- For the model:
  - Go to the console subsystem; add 2 new constant blocks to the console subsystem – these will represent the frequency and duty cycle of the PWM signal to be sent on a channel of the digital output module of the OP4200
  - Use a bus creator to concatenate the outputs of the two constant blocks
  - Add a new outport and connect it to the output of the bus creator
  - Go to the computation subsystem; change the OpComm block to have 4 inputs and outputs
  - Add another OpOutput block and connect it to the 4th output of the OpComm block
  - Add a new inport and connect it to the 4th input of the OpComm block
  - Go to the top level model view; connect the new outport of the console subsystem to the new inport of the computation subsystem
  - Save and compile the model

Console subsystem



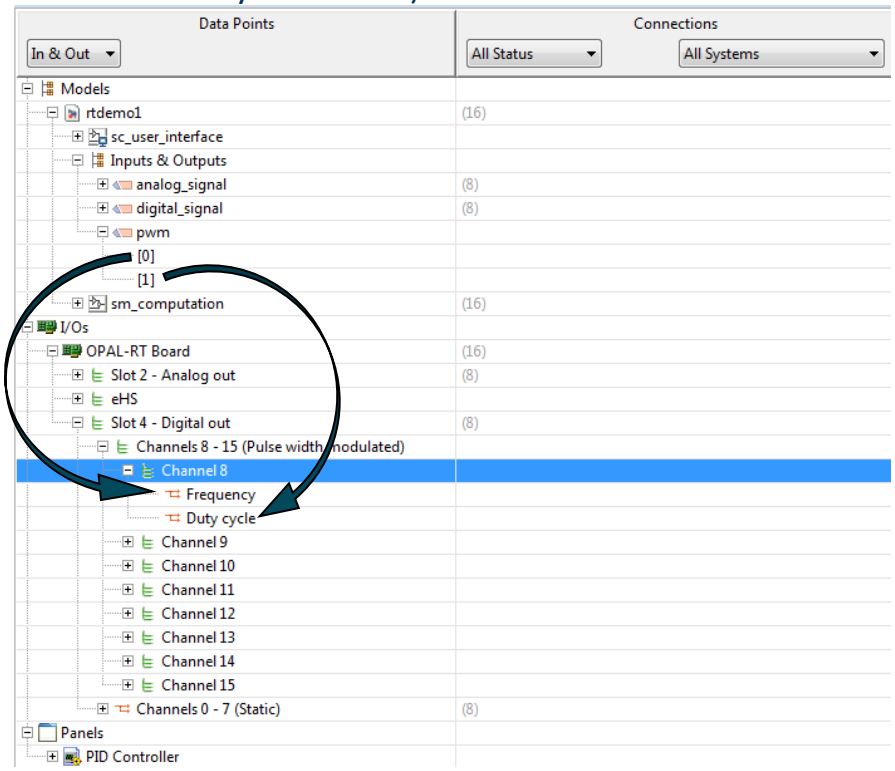
Computation subsystem



Using IOs with OP4200: sending values

Sending a PWM digital signal

- For the driver:
  - Go to the *Slot 4 – Digital out / Channels 8 – 15* view
  - Enable the group of channels
  - Set the *Digital type* to *Pulse width modulated*
  - Save the configuration
- In the *Configuration* view of the RT-LAB project, connect the new OpOutput signals to the frequency and duty cycle of channel 8 (or another channel of your choice)



Using IOs with OP4200: sending values

Sending a PWM digital signal

rtdemo1 | OPAL-RT Board | OP4200_IOs Configuration

type filter text

Data Points: In & Out	Connections: All Status	Connections: All Systems
Models		
rtdemo1	(18)	
sc_user_interface		
Inputs & Outputs		
analog_signal	(8)	
digital_signal	(8)	
pwm	(2)	
[0]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 8/Frequency	
[1]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 8/Duty cycle	
sm_computation	(18)	
I/Os		
OPAL-RT Board	(18)	
Slot 2 - Analog out	(8)	
eHS		
Slot 4 - Digital out	(10)	
Channels 8 - 15 (Pulse width modulated)	(2)	
Channel 8	(2)	
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[0]	
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[1]	
Channel 9		
Channel 10		
Channel 11		
Channel 12		
Channel 13		
Channel 14		
Channel 15		
Channels 0 - 7 (Static)	(8)	
Panels		
PID Controller		

Using IOs with OP4200: sending values

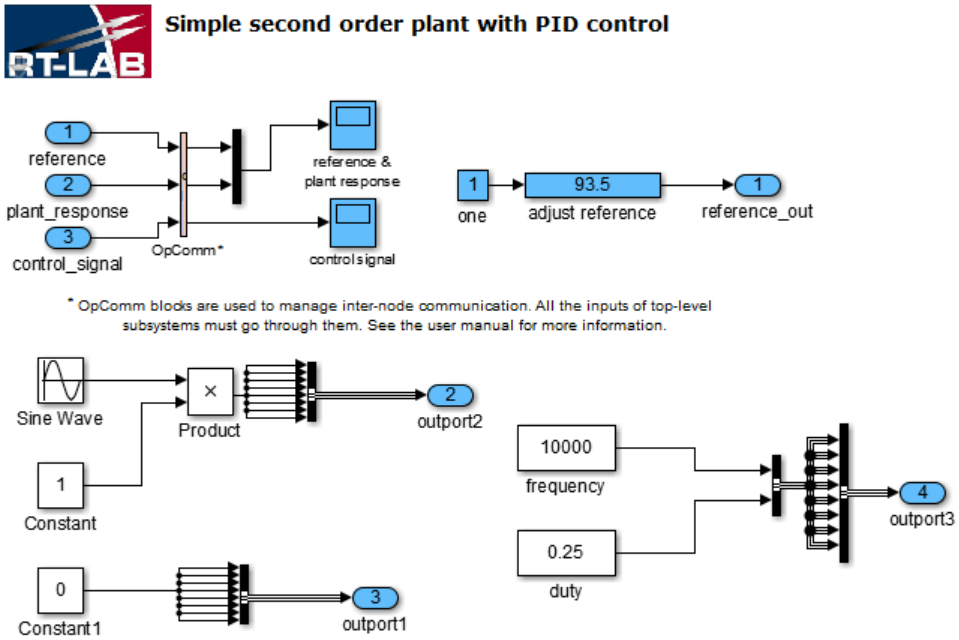
Sending a PWM digital signal

- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- Change the values in the new constant blocks for the frequency and duty cycle in the console subsystem to change the shape of the square wave outputted on channel 8 (or the channel that you chose to make the connection on) of the digital output module

Using IOs with OP4200: sending values

Sending all PWM digital values in a channel group

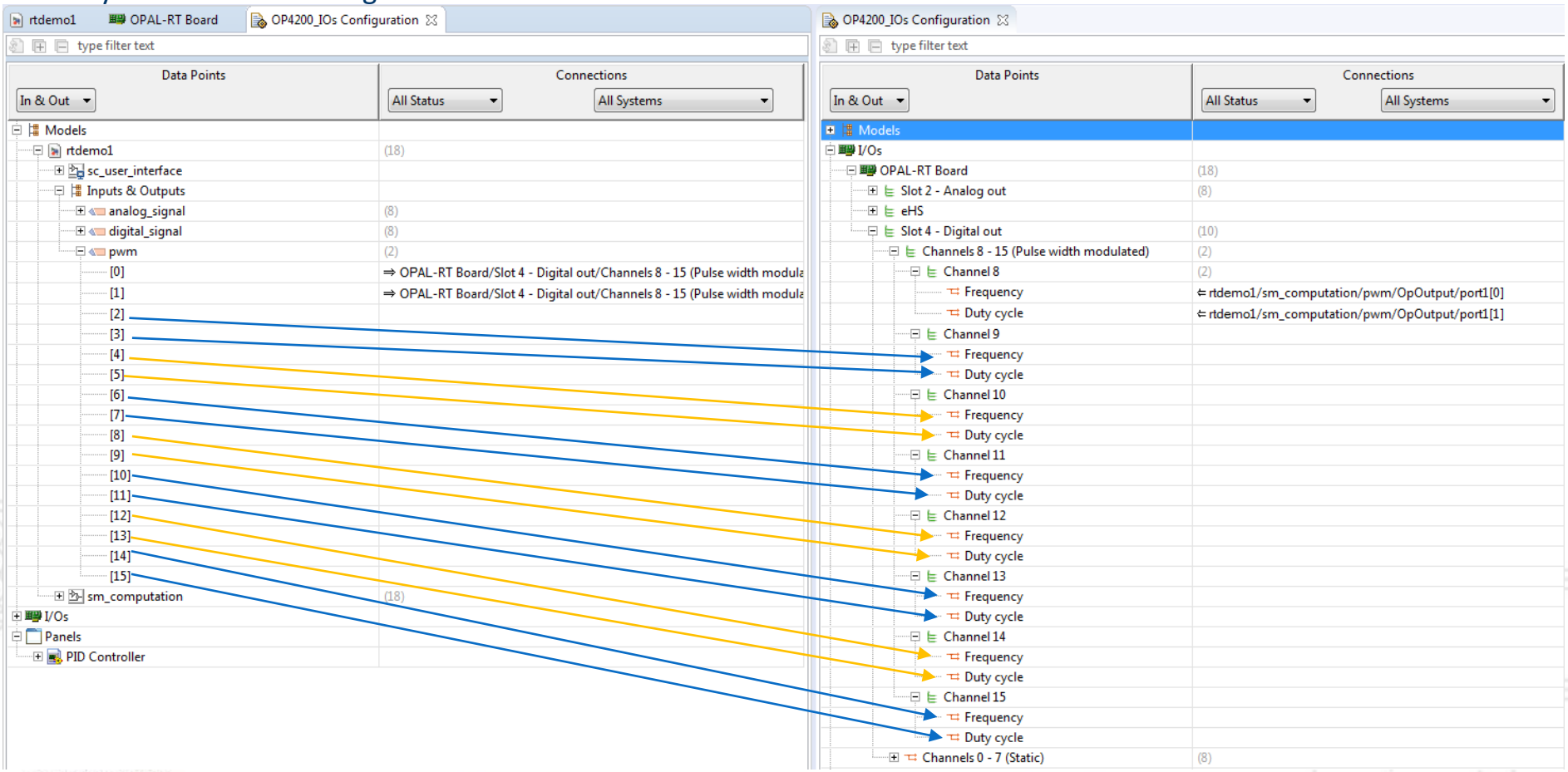
- For the model:
  - Go to the console subsystem; repeat the steps of adding the bus creator for the frequency and duty cycle for the remaining 7 channels in the group
  - You can regroup all frequencies and duty cycles with a bus creator before connecting them to the outport but make sure to keep track of the order of the signals in the bus – this will be important when connecting the signals to the channels in the driver
  - No changes required in the computation subsystem
  - Save and compile the model



Using IOs with OP4200: sending values

Sending all PWM digital values in a channel group

- For the driver:
  - No changes required
- In the *Configuration* view of the RT-LAB project, connect the new OpOutput signals to the frequencies and the duty cycles of the remaining channels



Using IOs with OP4200: sending values

Sending all PWM digital values in a channel group

rtdemo1 | OPAL-RT Board | OP4200_IOs Configuration

type filter text

Data Points (In & Out)	Connections (All Status / All Systems)
Models	
rtdemo1	(32)
sc_user_interface	
Inputs & Outputs	
analog_signal	(8)
digital_signal	(8)
pwm	(16)
[0]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 8/Frequency
[1]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 8/Duty cycle
[2]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 9/Frequency
[3]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 9/Duty cycle
[4]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 10/Frequency
[5]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 10/Duty cycle
[6]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 11/Frequency
[7]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 11/Duty cycle
[8]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 12/Frequency
[9]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 12/Duty cycle
[10]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 13/Frequency
[11]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 13/Duty cycle
[12]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 14/Frequency
[13]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 14/Duty cycle
[14]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 15/Frequency
[15]	⇒ OPAL-RT Board/Slot 4 - Digital out/Channels 8 - 15 (Pulse width modulated)/Channel 15/Duty cycle
sm_computation	(32)
I/Os	
Panels	
PID Controller	

OP4200_IOs Configuration

type filter text

Data Points (In & Out)	Connections (All Status / All Systems)
Models	
I/Os	
OPAL-RT Board	(32)
Slot 2 - Analog out	(8)
eHS	
Slot 4 - Digital out	(24)
Channels 8 - 15 (Pulse width modulated)	(16)
Channel 8	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[0]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[1]
Channel 9	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[2]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[3]
Channel 10	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[4]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[5]
Channel 11	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[6]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[7]
Channel 12	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[8]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[9]
Channel 13	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[10]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[11]
Channel 14	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[12]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[13]
Channel 15	(2)
Frequency	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[14]
Duty cycle	⇐ rtdemo1/sm_computation/pwm/OpOutput/port1[15]
Channels 0 - 7 (Static)	(8)
Panels	

Using IOs with OP4200: sending values

Sending all PWM digital values in a channel group

Hint: you can open 2 instances of the *Configuration* view to create the connections in case the number of the connectable items is greater than what the screen can fit

- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- Change the values in the constant blocks for the frequency and duty cycle in the console subsystem to change the shape of the square waves outputted on channels 8 through 15 of the digital output module of the OP4200

IMPORTANT NOTES:

1. For a more complex example of how to send PWM digital values, please consult the example model for OP4200 delivered with your RT-LAB installation
2. The steps presented in this guide are mostly for example purposes. There are many other ways to create the OpOutput blocks necessary for your simulation. You can have 1 OpOutput block for each PWM signal or 1 OpOutput for each frequency and duty cycle of each signal or 1 OpOutput for all signals (as in our example) and more. Be creative and try to find the most efficient way to regroup the signals when constructing your model.