OPAL-RT TECHNOLOGIES

FROM IMAGINATION TO REAL-TIME

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

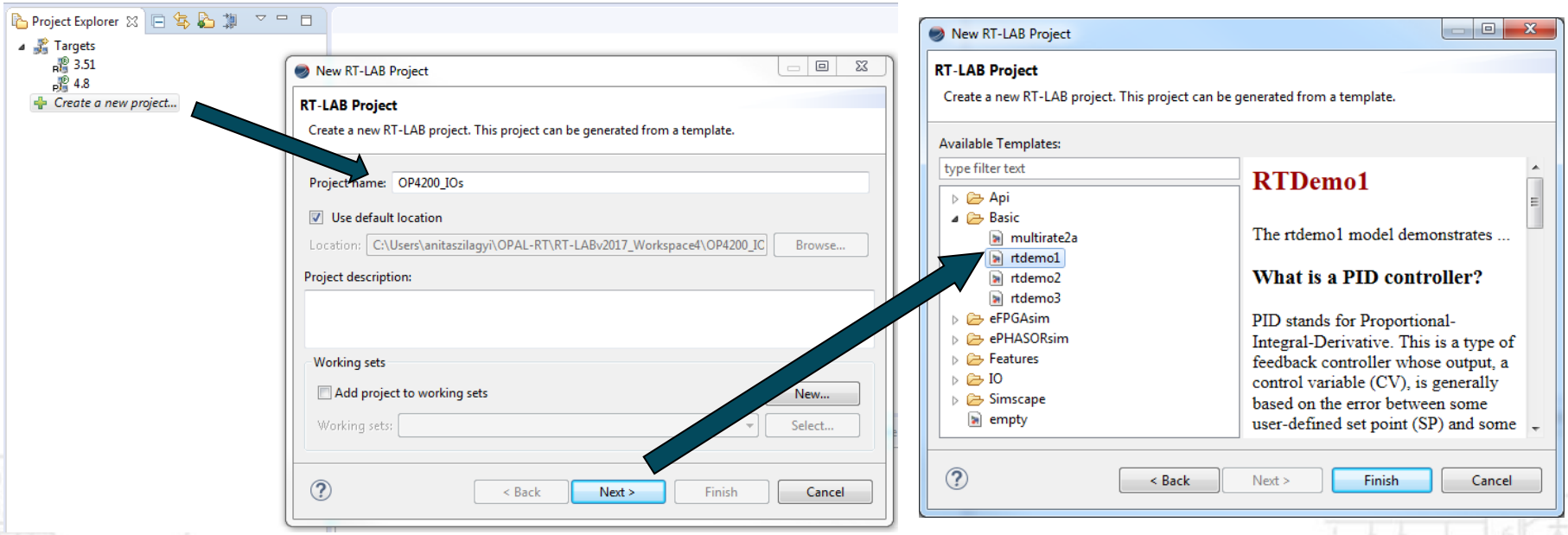
Drivers Team

Using IOs with OP4200: receiving values

Receiving 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

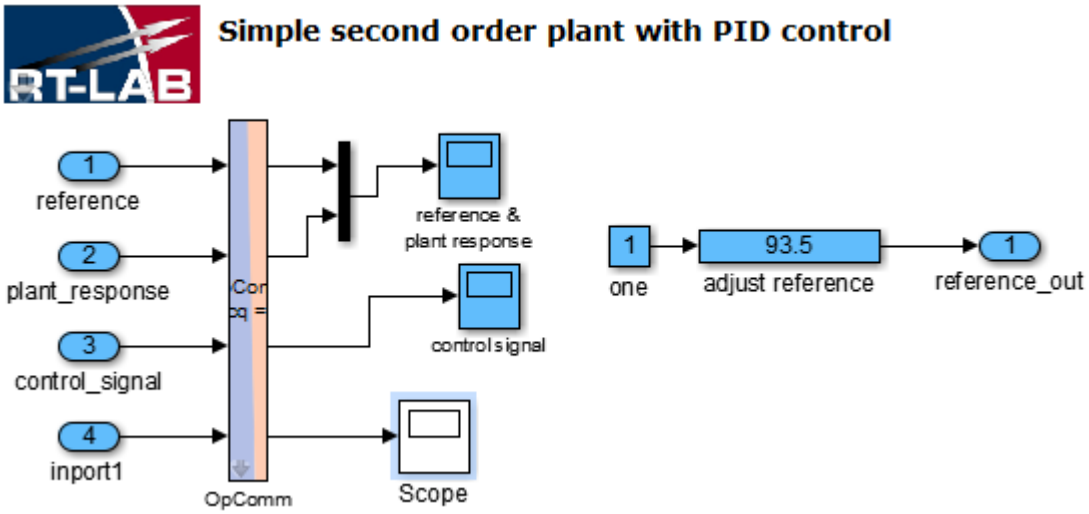


- Click on *Finish* once done

Using IOs with OP4200: receiving values

Receiving an analog value: model side

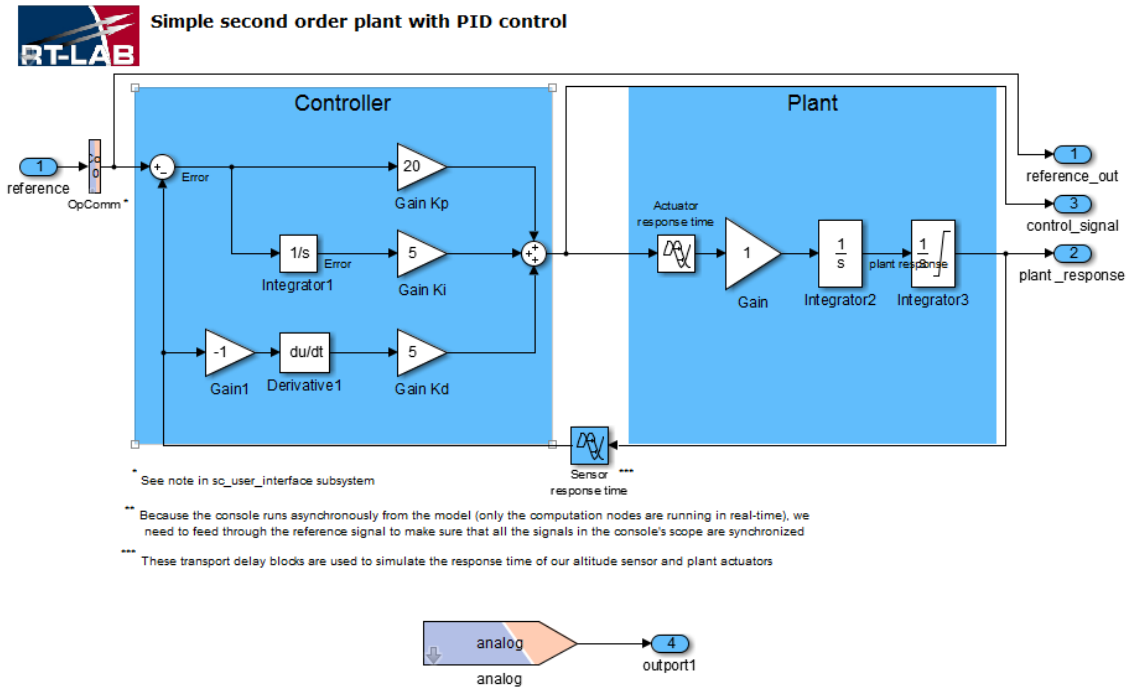
- Edit the model as such:
  - Go into the sc_user_interface subsystem
    - You can leave the current logic untouched, we will not be needing it
    - Double-click on the OpComm block to reveal its parameters
    - Change the number of inports to 4 then click OK
    - Add a new inport and connect it to the newly created input of the OpComm block
    - Add a scope block (found in Simulink's Library Browser -> Sinks) to the model
    - Connect the input of the scope block to the new output of the OpComm block
    - Your subsystem should look as such:



Using IOs with OP4200: receiving values

Receiving an analog value: model side

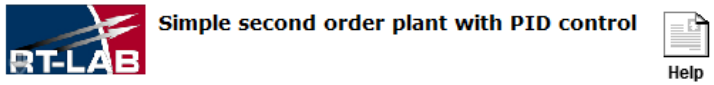
- Edit the model you just added as such:
  - Go into the sm_computation subsystem
    - You can leave the current logic untouched, we will not be needing it
    - Add an OpInput block from Library Browser -> RT-LAB
    - Name the OpInput block as you wish, but make sure to change its label to the same name
    - Connect the OpInput block to a new outport
    - Your subsystem should look as such:

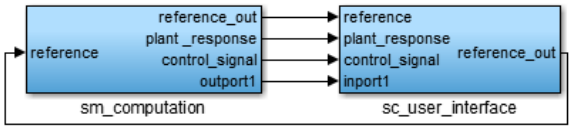


Using IOs with OP4200: receiving values

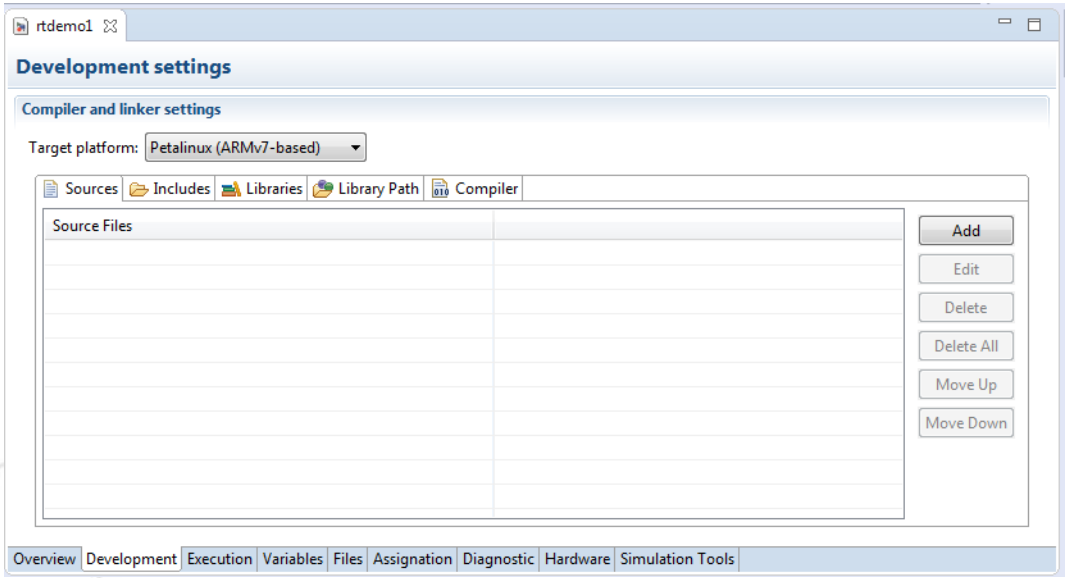
Receiving an analog value: model side

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





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



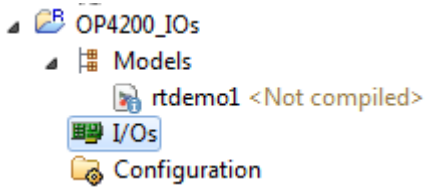
- Compile the model

Using IOs with OP4200: receiving values

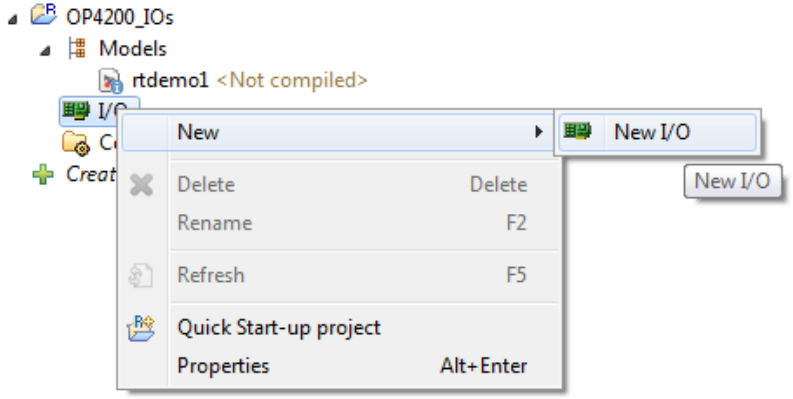
Receiving an analog value: driver side

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

1. Initial view of the project



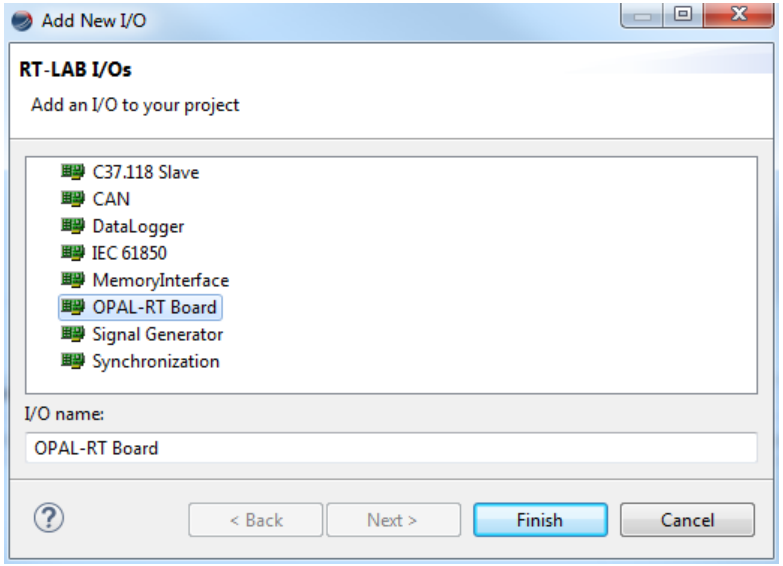
2. Adding a new I/O



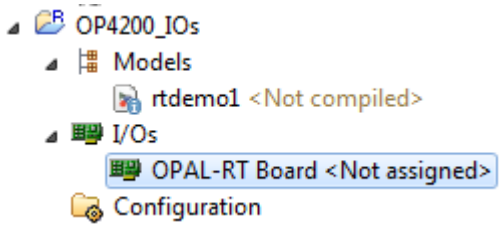
Using IOs with OP4200: receiving values

Receiving an analog value: driver side

3. Select OPAL-RT Board from the list



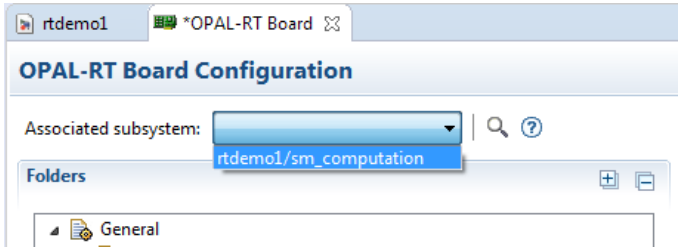
4. Clicking on “Finish” will add it to the project



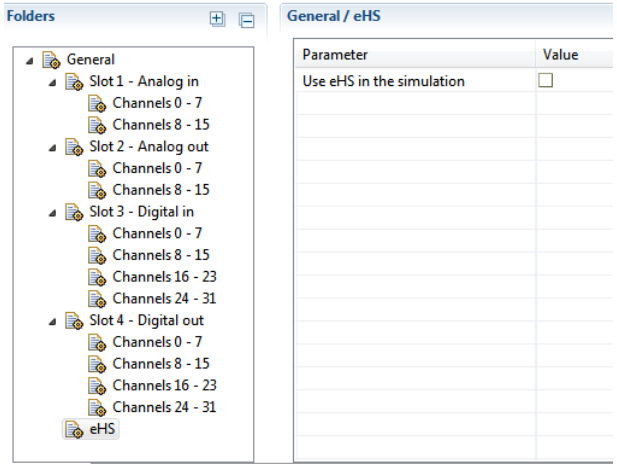
Using IOs with OP4200: receiving values

Receiving 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

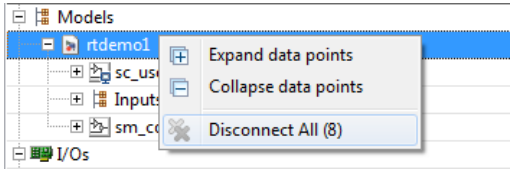


- Click on *Slot 1 – Analog In / Channels 0 – 7*
- Tick the *Enable* check-box; this will enable the use of the first 8 channels of the analog in module in the simulation
- Save the driver configuration by typing Control + S or by clicking on the save button under the menu bar

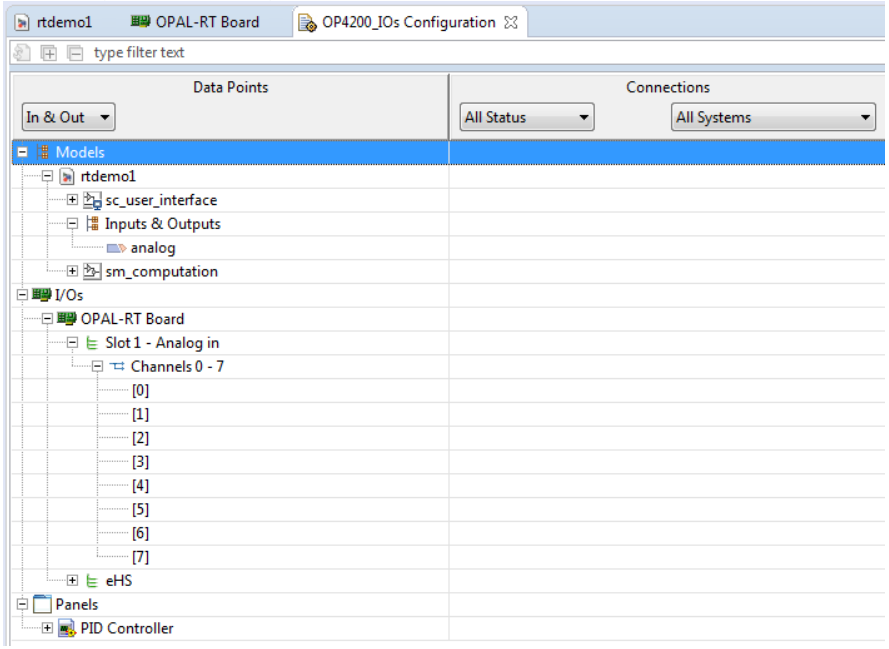
Using IOs with OP4200: receiving values

Receiving 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:



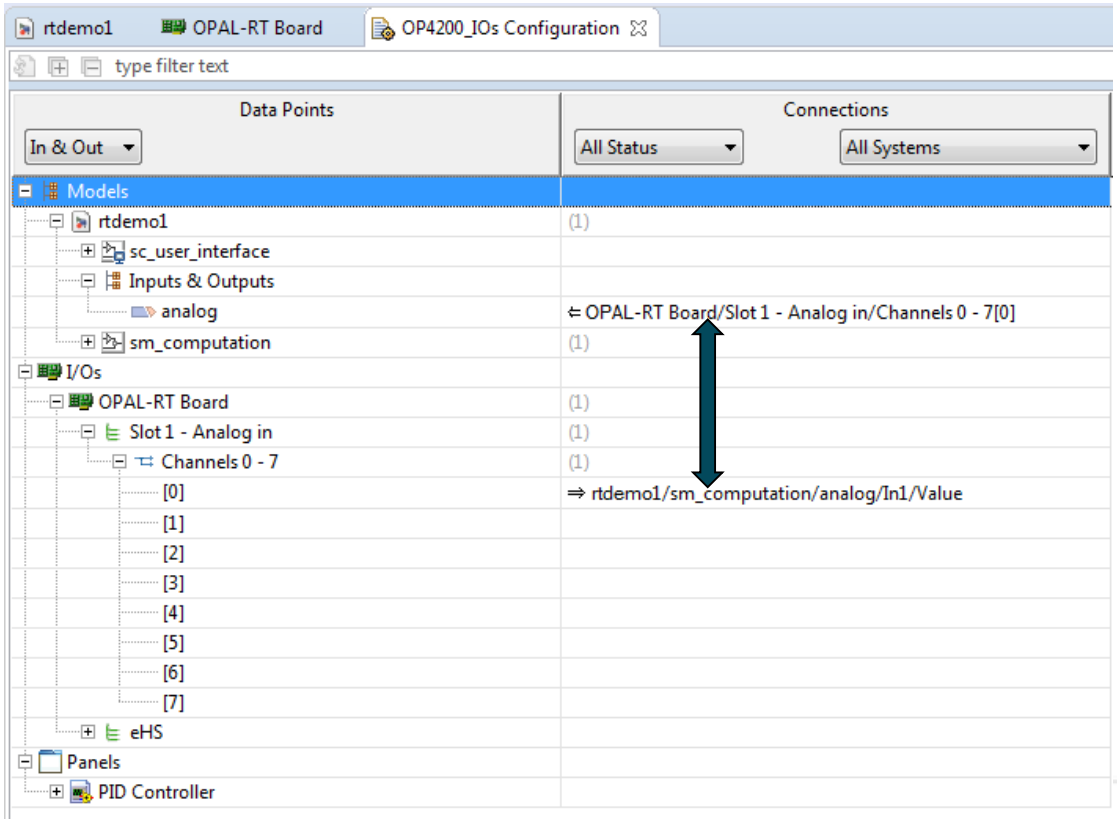
- Expand rtdemo1 -> Inputs & Outputs and I/Os -> OPAL-RT Board -> Slot 1 – Analog in -> Channels 0 – 7; final look:



Using IOs with OP4200: receiving values

Receiving an analog value: connecting the model to the driver

- In order for the value received on channel 0 of the digital input module to be displayed in the model's console, you need to drag and drop the OpInput block onto Channel 0
- If you prefer to use another physical channel for receiving analog values, drag and drop the OpInput onto the respective channel
- Final look:



Using IOs with OP4200: receiving values

Receiving an analog value: running the simulation

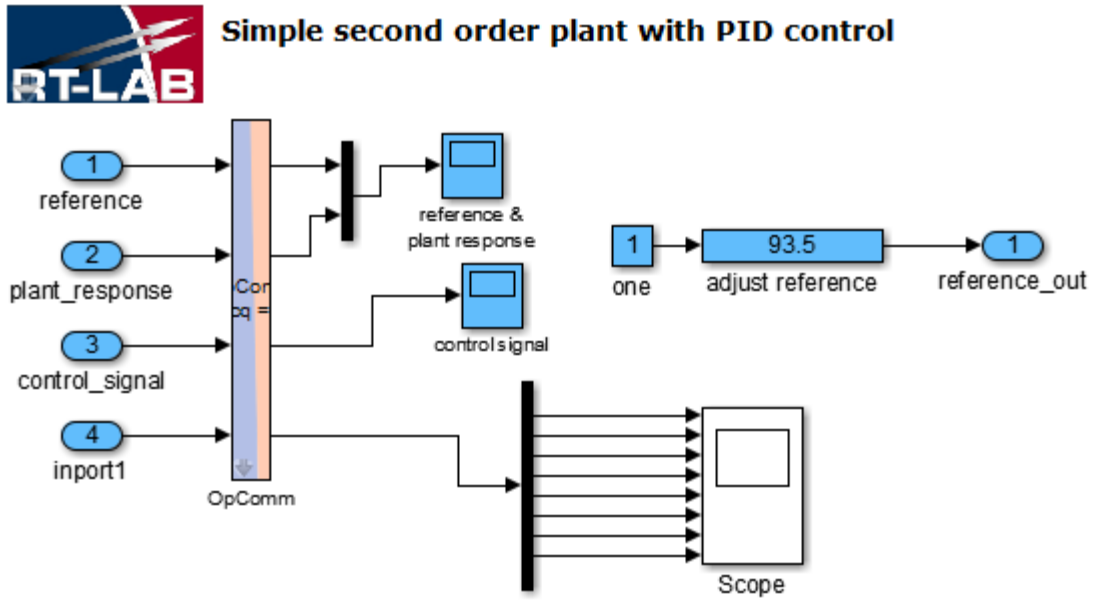
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- In the console, the value received on channel 0 of the analog input cassette found in slot 1 of the OP4200 should be displayed

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 input values from the physical channel of the analog input module.

Using IOs with OP4200: receiving values

Receiving all analog values in a channel group: model side

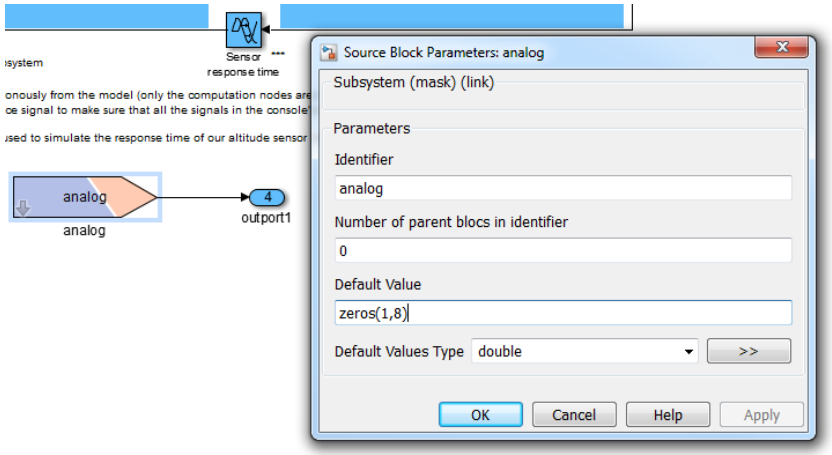
- Edit the model as such:
  - Go into the sc_user_interface subsystem
    - Edit the scope added earlier so it will have 8 plots instead of just 1
    - Add a demux block (Library Browser -> Simulink -> Signal Routing) and set it to have 8 outputs
    - Connect the demux outputs to the inputs of the scope
    - Your subsystem should look as such:



Using IOs with OP4200: receiving values

Receiving all analog values in a channel group: model side

- In the computation block
  - The OpInput block needs to be initialized to receive 8 values as opposed to just 1: double-click on it and set its initial value to zeros(1, 8)



- Save and compile the model

Using IOs with OP4200: receiving values

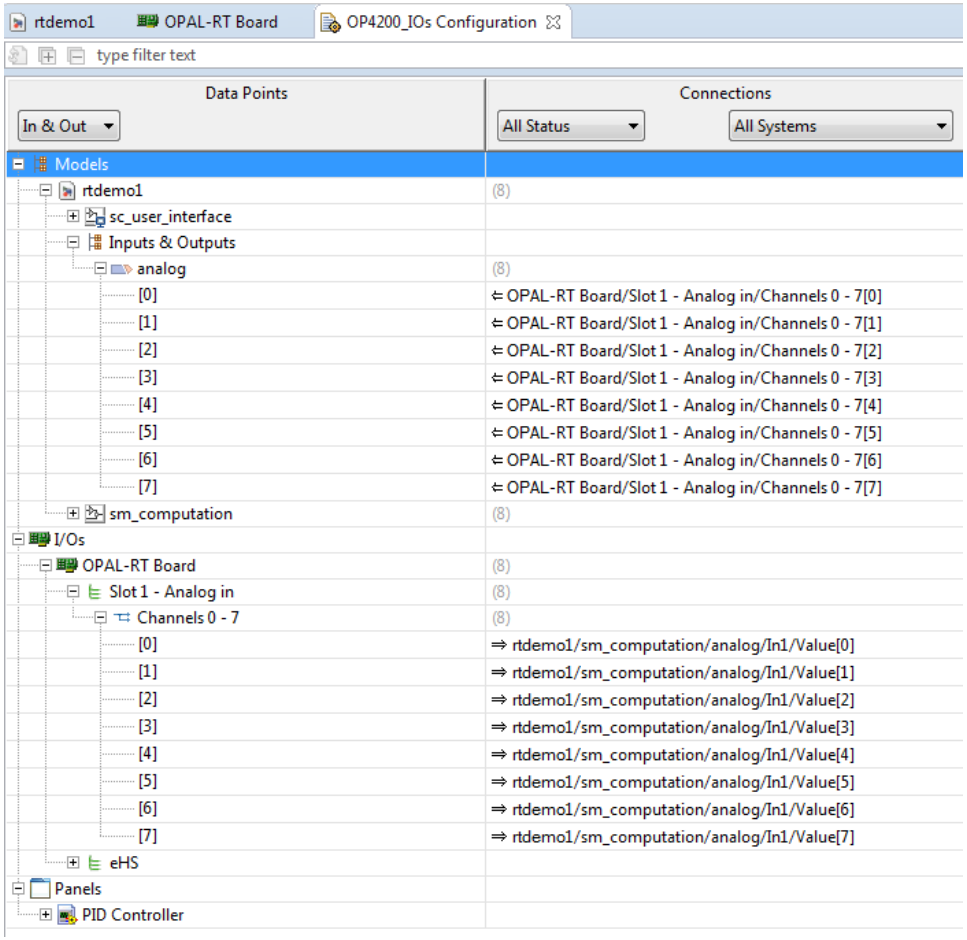
Receiving all analog values in a channel group : driver side

- 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: receiving values

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

- You can now connect all 8 signals of the compiled OpInput block to each physical analog input channel
- Alternatively, you can drag and drop the OpInput onto the Channels 0 – 7 vector
- Final look:



Using IOs with OP4200: receiving values

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

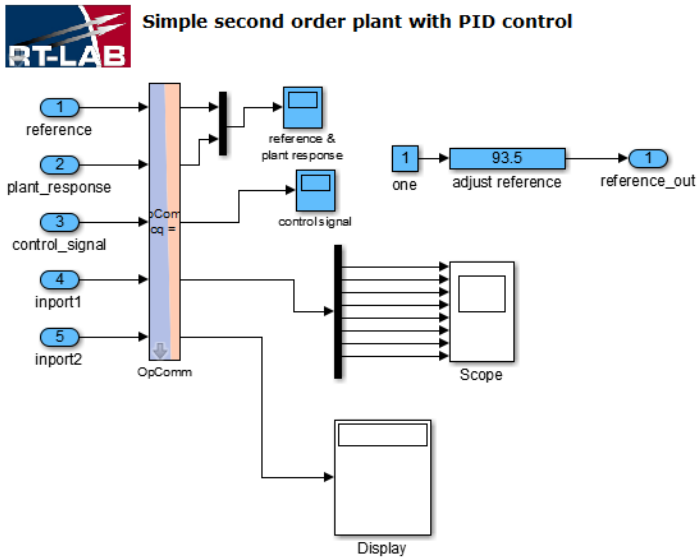
- Since the bitstream has not changed, we can disable the automatic flashing by setting the RT-LAB environment variable “DISABLE_FLASH_UPDATE” to “ON”
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- In the console, you can now monitor the values received on channels 0 to 7 of the analog input module in slot 1 of the OP4200

Using IOs with OP4200: receiving values

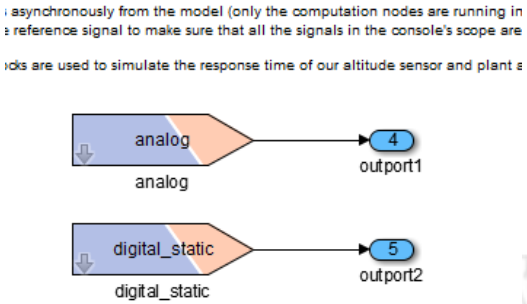
Receiving a static digital value

- For the model:
  - Go to the console subsystem; change the OpComm block to have 5 inputs and outputs
  - Connect a new display block to the newly added output of the OpComm block
  - Add a new inport and connect it to the 5th input of the OpComm block
  - Go to the computation subsystem; add a new OpInput block
  - Connect the OpInput to a new outport
  - Go to the top level model view; connect the new outport of the computation subsystem to the new inport of the console subsystem
  - Save and compile the model

Console subsystem



Computation subsystem



Using IOs with OP4200: receiving values

Receiving a static digital value

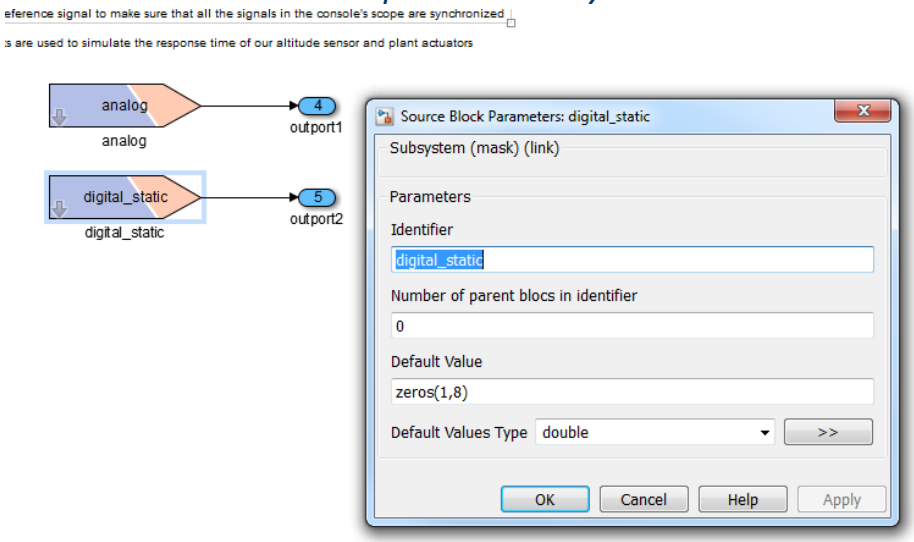
- For the driver:
  - Go to the Slot 3 – Digital in / Channels 0 – 7 view
  - Enable the group of channels
  - Save the configuration
- Connect the OpInput block to the digital input channel as you did for the analog input channel (seen on slide 10)
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- You can now monitor the static digital value received on channel 0 (or the one that you chose to make the connection on)

Using IOs with OP4200: receiving values

Receiving all static digital values in a channel group

- For the model:
  - No changes required for the console subsystem; the display block can adapt to receiving 8 inputs
  - Go to the computation subsystem; repeat the steps of initializing the new digital input OpInput block to 8 values
  - Save and compile the model

Computation subsystem



- For the driver:
  - No changes needed on the driver side
- Repeat the connection steps as was done on slide 15
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- You can now monitor the values received on channels 0 to 7 of the digital input module of the OP4200

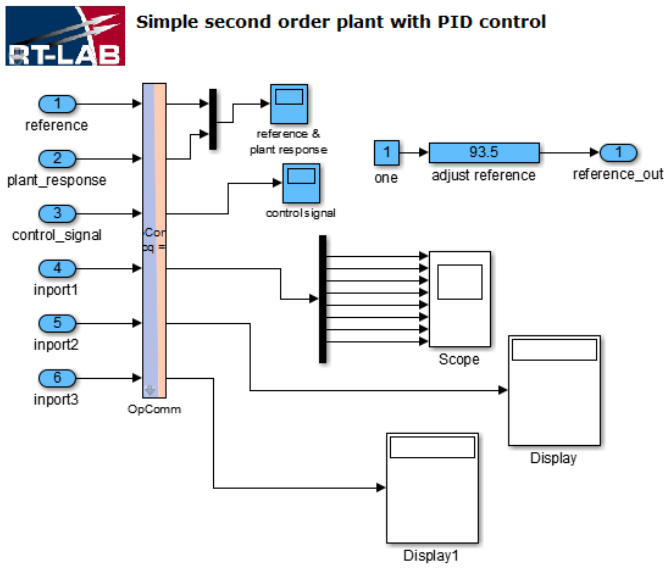
- **NOTE:** for a more complex example of how to receive static digital values, please consult the example model for OP4200 delivered with your RT-LAB installation

Using IOs with OP4200: receiving values

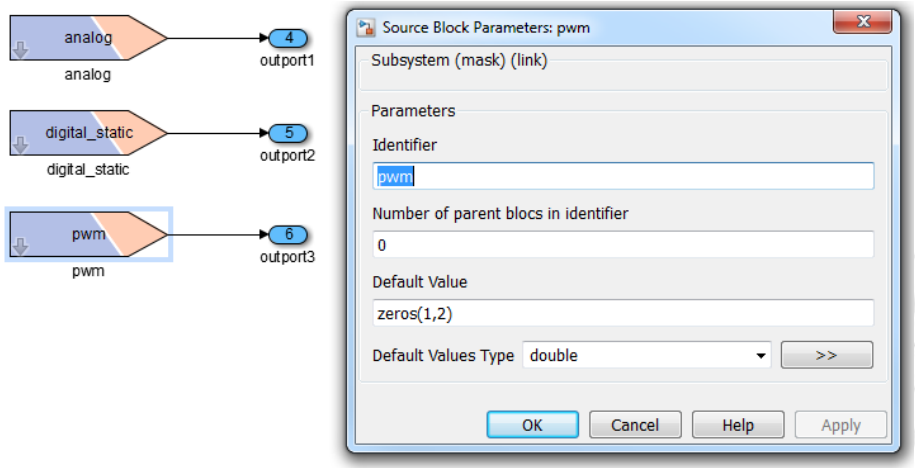
Receiving a PWM digital signal

- For the model:
  - Go to the console subsystem; change the OpComm block to have 6 inputs and outputs
  - Add a new inport and connect it to the new input of the OpComm block
  - Add a new display block and connect it to the new output of the OpComm block
  - Go to the computation subsystem; add a new OpInput block
  - Initialize the new OpInput block to receive 2 values (1 frequency and 1 duty cycle for 1 channel)
  - Connect the OpInput block to a new outport
  - Go to the top level model view; connect the new outport of the computation subsystem to the new inport of the console subsystem
  - Save and compile the model

Console subsystem



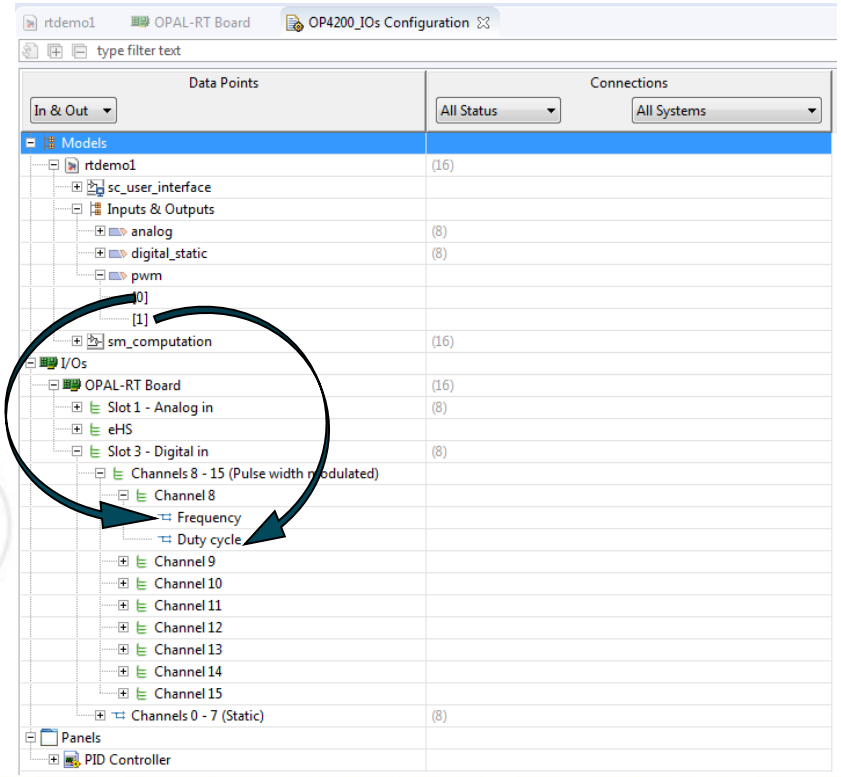
Computation subsystem



Using IOs with OP4200: receiving values

Receiving a PWM digital signal

- For the driver:
  - Go to the Slot 3 – Digital in / 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 OpInput signals to the frequency and duty cycle of channel 8 (or another channel of your choice)



Using IOs with OP4200: receiving values

Receiving 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	(8)	
digital_static	(8)	
pwm	(2)	
[0]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 8/Frequency	
[1]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 8/Duty cycle	
sm_computation	(18)	
I/Os		
OPAL-RT Board	(18)	
Slot 1 - Analog in	(8)	
eHS		
Slot 3 - Digital in	(10)	
Channels 8 - 15 (Pulse width modulated)	(2)	
Channel 8	(2)	
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[0]	
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[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: receiving values

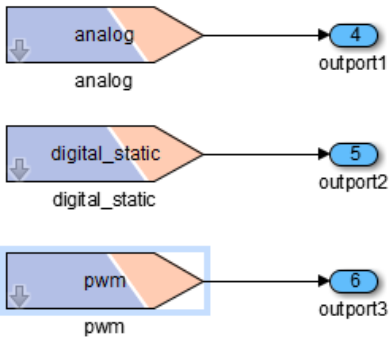
Receiving a PWM digital signal

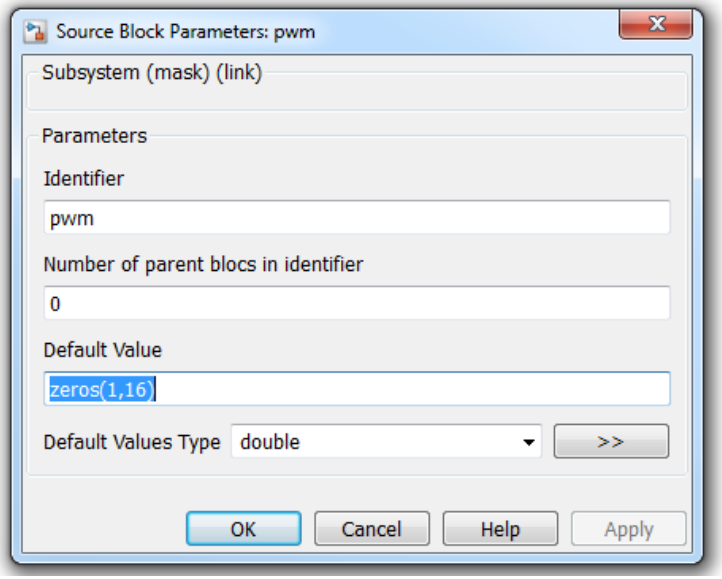
- Load the model
- Check the RT-LAB display to make sure that all connections were done successfully
- Run the model
- You should now be able to monitor the frequency and the duty cycle of the PWM signal received on channel 8 (or the one of your choice) of the digital input module of the OP4200

Using IOs with OP4200: receiving values

Receive all PWM digital values in a channel group

- For the model:
  - No changes are required for the console subsystem
  - Go to the computation subsystem; initialize the OpInput block dedicated to receiving PWM signals to 16 signals (1 frequency + 1 duty cycle for 8 channels)



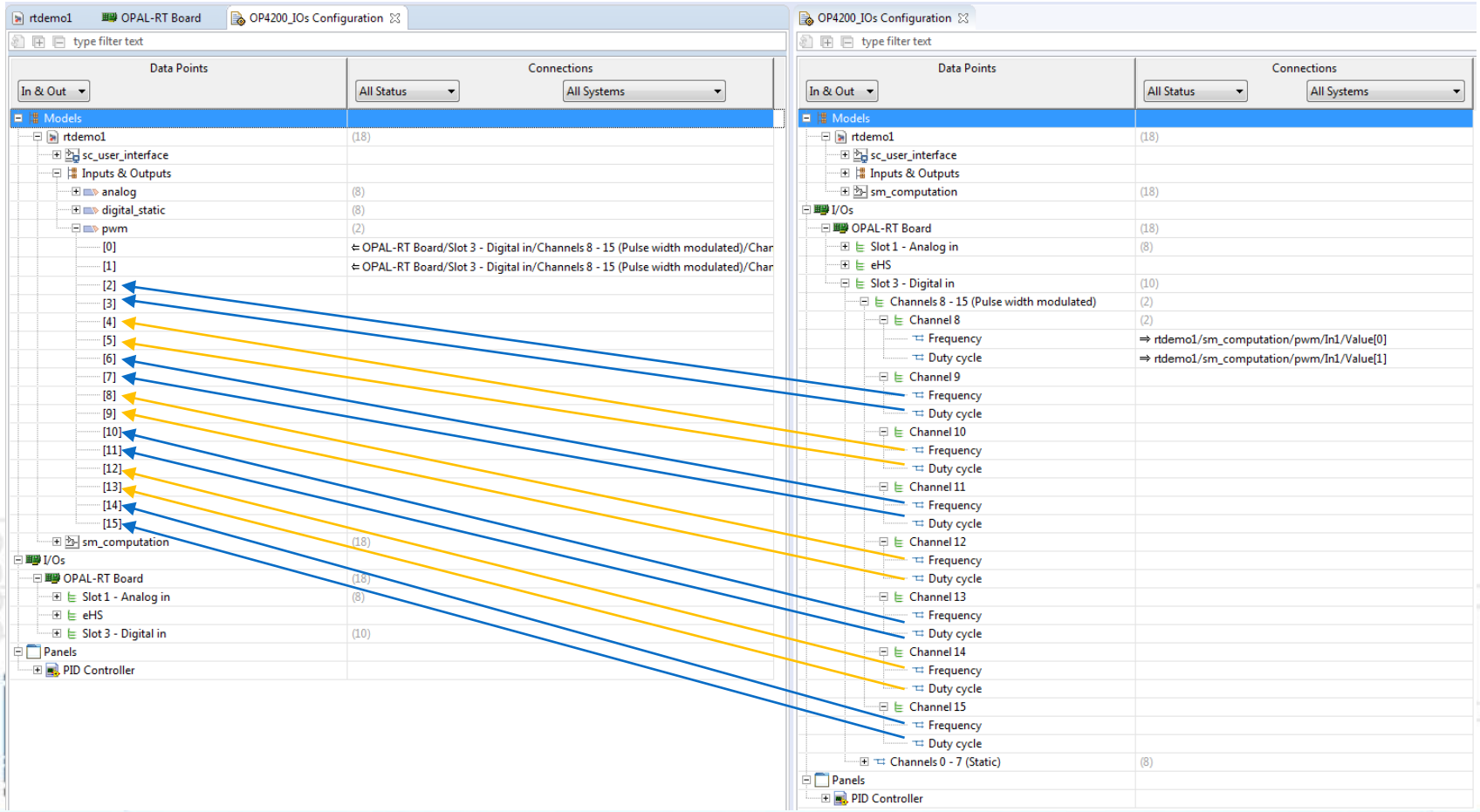


  - Save and compile the model

Using IOs with OP4200: receiving values

Receiving 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 OpInput signals to the frequencies and the duty cycles of the remaining channels



Using IOs with OP4200: receiving values

Receiving 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	(8)
digital_static	(8)
pwm	(16)
[0]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 8/Frequency
[1]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 8/Duty cycle
[2]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 9/Frequency
[3]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 9/Duty cycle
[4]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 10/Frequency
[5]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 10/Duty cycle
[6]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 11/Frequency
[7]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 11/Duty cycle
[8]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 12/Frequency
[9]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 12/Duty cycle
[10]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 13/Frequency
[11]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 13/Duty cycle
[12]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 14/Frequency
[13]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 14/Duty cycle
[14]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 15/Frequency
[15]	⇐ OPAL-RT Board/Slot 3 - Digital in/Channels 8 - 15 (Pulse width modulated)/Channel 15/Duty cycle
sm_computation	(32)
I/Os	
OPAL-RT Board	(32)
Slot 1 - Analog in	(8)
eHS	
Slot 3 - Digital in	(24)
Panels	
PID Controller	

OP4200_IOs Configuration

type filter text

Data Points (In & Out)	Connections (All Status / All Systems)
Models	
rtdemo1	(32)
sc_user_interface	
Inputs & Outputs	
sm_computation	(32)
I/Os	
OPAL-RT Board	(32)
Slot 1 - Analog in	(8)
eHS	
Slot 3 - Digital in	(24)
Channels 8 - 15 (Pulse width modulated)	(16)
Channel 8	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[0]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[1]
Channel 9	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[2]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[3]
Channel 10	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[4]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[5]
Channel 11	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[6]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[7]
Channel 12	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[8]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[9]
Channel 13	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[10]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[11]
Channel 14	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[12]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[13]
Channel 15	(2)
Frequency	⇒ rtdemo1/sm_computation/pwm/In1/Value[14]
Duty cycle	⇒ rtdemo1/sm_computation/pwm/In1/Value[15]
Channels 0 - 7 (Static)	(8)
Panels	
PID Controller	

Using IOs with OP4200: receiving values

Receiving 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
- You should now be able to monitor the values for the frequencies and duty cycles for channels 8 to 15 of the digital input module of the OP4200

IMPORTANT NOTES:

1. For a more complex example of how to receive 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 OpInput blocks necessary for your simulation. You can have 1 OpInput block for each PWM signal or 1 OpInput for each frequency and duty cycle of each signal or 1 OpInput 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.