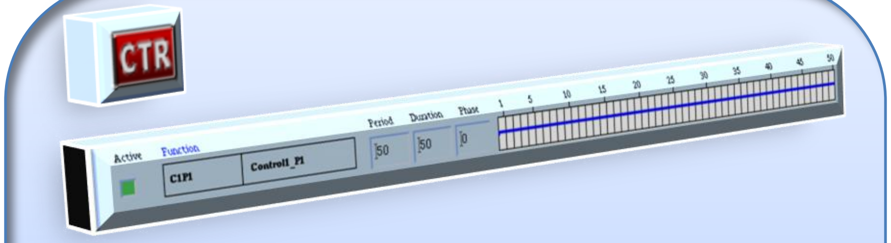


Active ■ | Function C1P1 | Control1_P1 | Period 50 | Duration 50 | Phase 0

The scheduling of the Control block

The scheduling of the Control block

The Control functions

The Control block may be programmed through a set of functions:

- **Graphical** Control:
  all the functions are entirely created and managed by EICASLAB and
  depend on the graphical scheme of the Control Layout and
  on the data (e.g. parameters, states) directly inserted by the user.

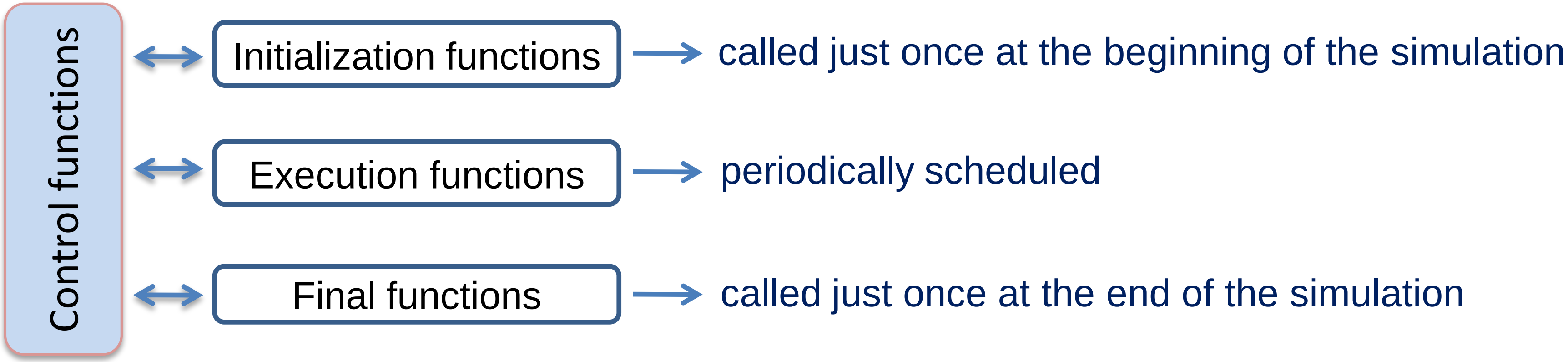
- Control programmed in **ANSI C**:
  all the functions have a template provided by EICASLAB and
  are managed by the user.

- Control programmed through the Automatic Algorithm Generation (**AAG**):
  all the functions are entirely created and managed by EICASLAB and
  depend on the data inserted by the user:
  - the control system requirements,
  - the simplified model of the Plant to be controlled.

The scheduling of the Control block

Functions categories

The functions belong to three main categories:



Control functions

Initialization functions → called just once at the beginning of the simulation

Execution functions → periodically scheduled

Final functions → called just once at the end of the simulation

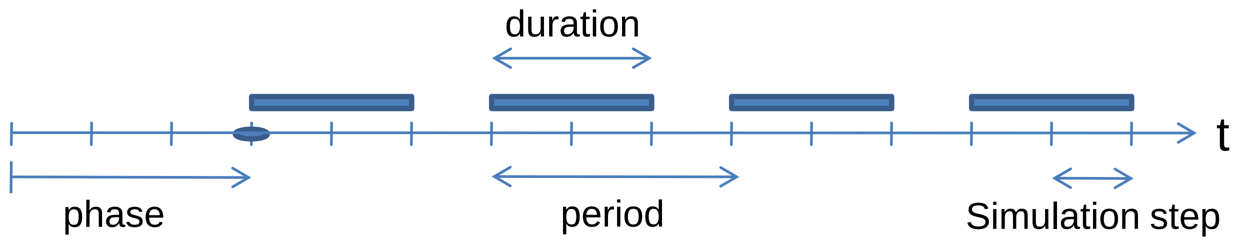
The scheduling of the Control block

Scheduling parameters

The user has to fix a **simulation step**,
which represents the time resolution applied in the simulation of the overall project.

The execution functions implement periodic activities characterized by the following scheduling parameters (expressed as a multiple of the simulation step):

- **Phase** time at which they are called for the first time,
- **Period** their sample time interval,
- **Duration** their execution time.



The scheduling of the Control block

Initialization functions

Graphical or **AAG** Control:
→ functions entirely created and managed by EICASLAB,
Control programmed in **ANSI C:**
→ functions created by EICASLAB (template) and managed by the user.

The initial functions are called just once at the beginning of the simulation, in the following order:

1) Parameter file reading,

2) Control Design (for *ANSI C* and AAG Controls).,

3) Initial state file reading,

4) User initialisation function (Only when programmed in ANSI C language).

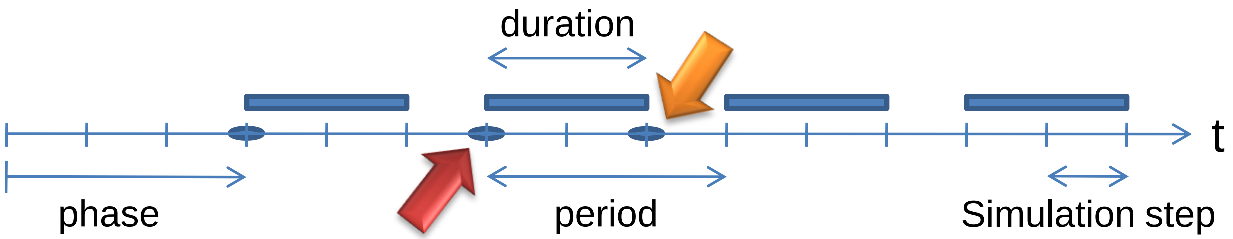
The scheduling of the Control block

Execution functions

Graphical or **AAG** Control:
→ functions entirely created and managed by EICASLAB,

Control programmed in **ANSI C:**
→ functions created by EICASLAB (template) and managed by the user.

‘Exe’ function	Implementing the Control algorithm, updating of the state of the Control, called when the Control is scheduled (considering its **phase** and **period**).
Output function	Computation of the outputs of the Control (as a function of its current state), called at the end of the ‘Exe’ function (depending on the **duration** of the ‘Exe’ function)



The scheduling of the Control block

Final functions

Graphical or **AAG** Control:

→ functions entirely created and managed by EICASLAB,

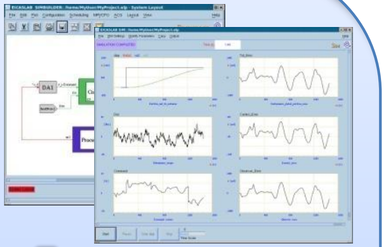
Control programmed in **ANSI C:**

→ functions created by EICASLAB (template) and managed by the user.

The final functions are called just once at the end of the simulation in the following order:

1) User final function (Only when programmed in ANSI C language),

2) Final state file writing.



Control scheduling in Modelling and Like Real-time Simulation phase

Control scheduling in Modelling and Like real-time Simulation phase

The SIM tool manages the Control scheduling by means of the **EICASLAB scheduler** that is **the core of the time scheduling algorithms and** allows to run like real-time simulations.

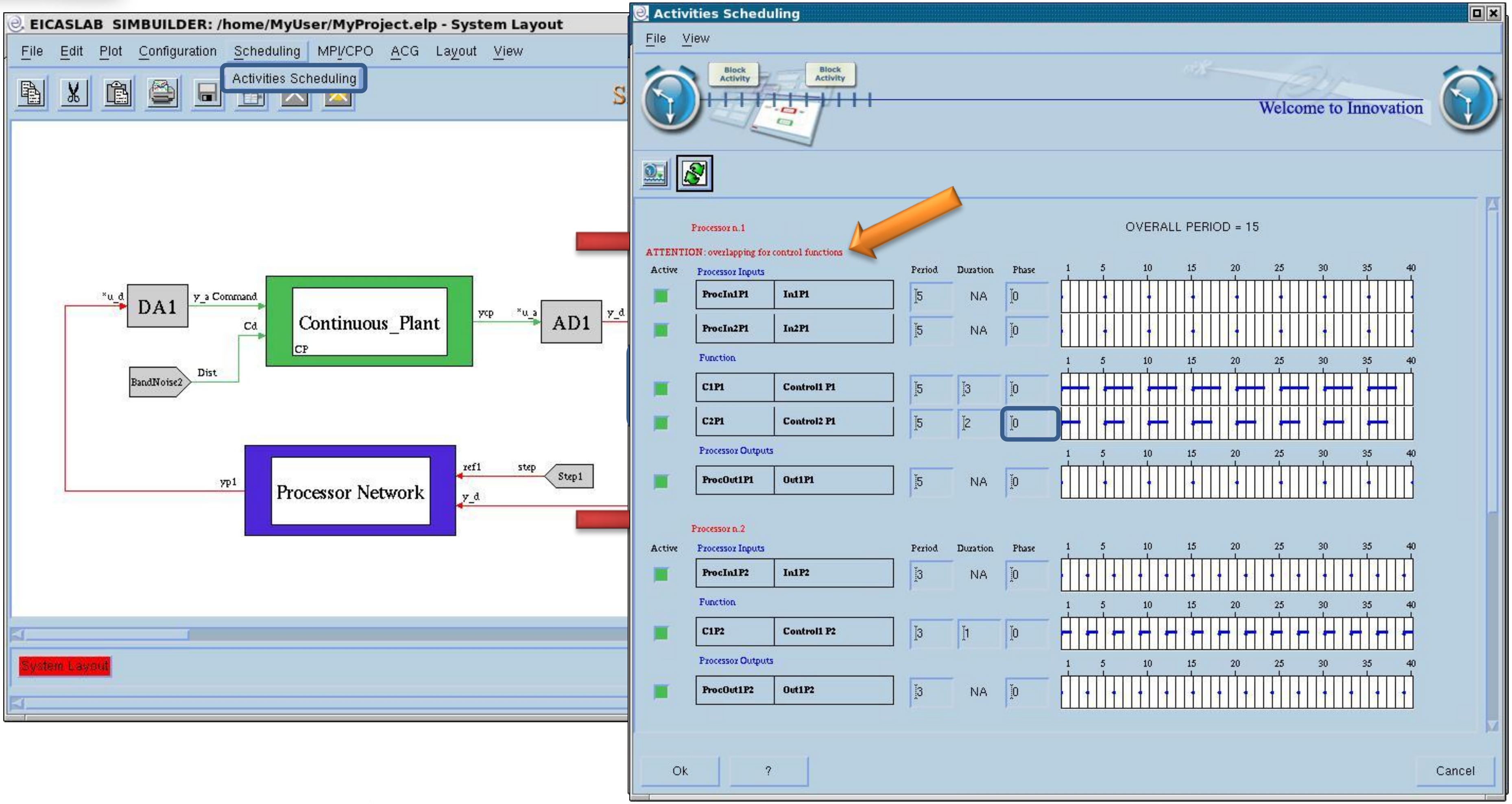
On the basis of the scheduling parameters

↓

It defines the order in which the Control functions have to be executed

Control scheduling in Modelling and Like real-time Simulation phase

How to set the scheduling



Processor n.1

ATTENTION: overlapping for control functions

OVERALL PERIOD = 15

Active	Processor Inputs		Period	Duration	Phase
🟩	ProcIn1P1	In1P1	5	NA	0
🟩	ProcIn2P1	In2P1	5	NA	0
	Function				
🟩	C1P1	Control1 P1	5	3	0
🟩	C2P1	Control2 P1	5	2	0
	Processor Outputs				
🟩	ProcOut1P1	Out1P1	5	NA	0

Processor n.2

Active	Processor Inputs		Period	Duration	Phase
🟩	ProcIn1P2	In1P2	3	NA	0
	Function				
🟩	C1P2	Control1 P2	3	1	0
	Processor Outputs				
🟩	ProcOut1P2	Out1P2	3	NA	0

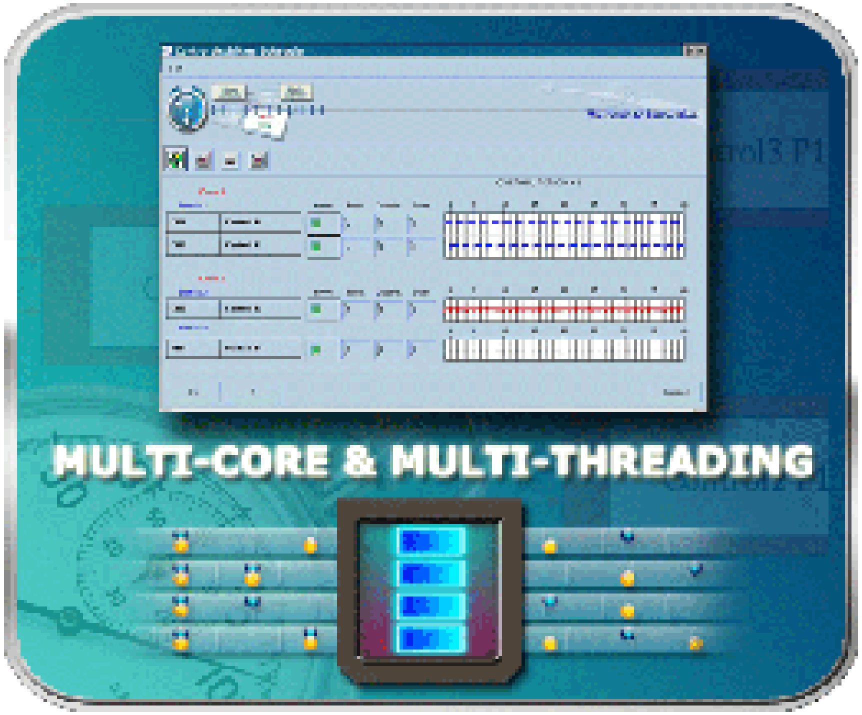
CTR

Control scheduling in Rapid Control Prototyping (RCP) phase

Control scheduling in RCP phase

RCP concept, scheduling review

During the RCP phase the Control system works directly with the real Plant. The control software is transferred in a smart PC Platform equipped with a suitable RTOS and by directly piloting your plant.



To guarantee the correct emulation of the Target control architecture, in the RCP phase, the user could have the need of reviewing the control scheduling.

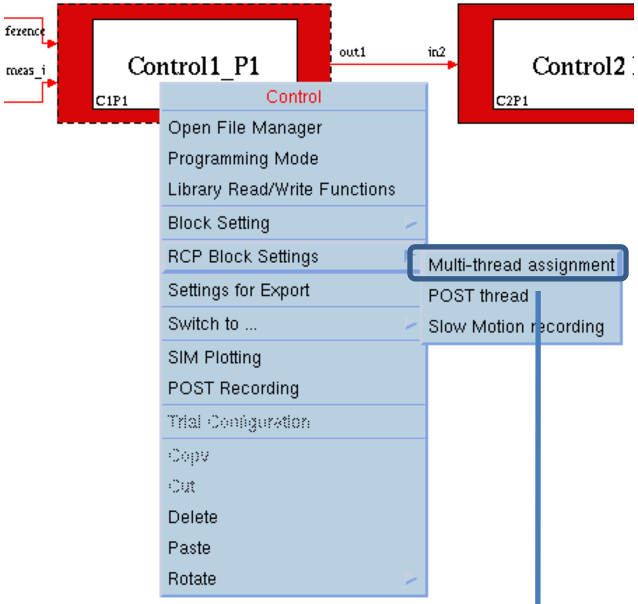
The new scheduling requires to organize the control functions in **threads**, that will be distributed on the **cores** of the multi-core CPU available in the PC platform, thanks to the application in EICASLAB of multi-threading and multi-core programming techniques.

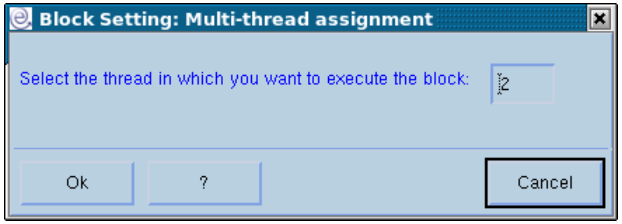
There is no more simulation step: instead of it the clock tick of the real-time operative system is considered.

Control scheduling in RCP phase

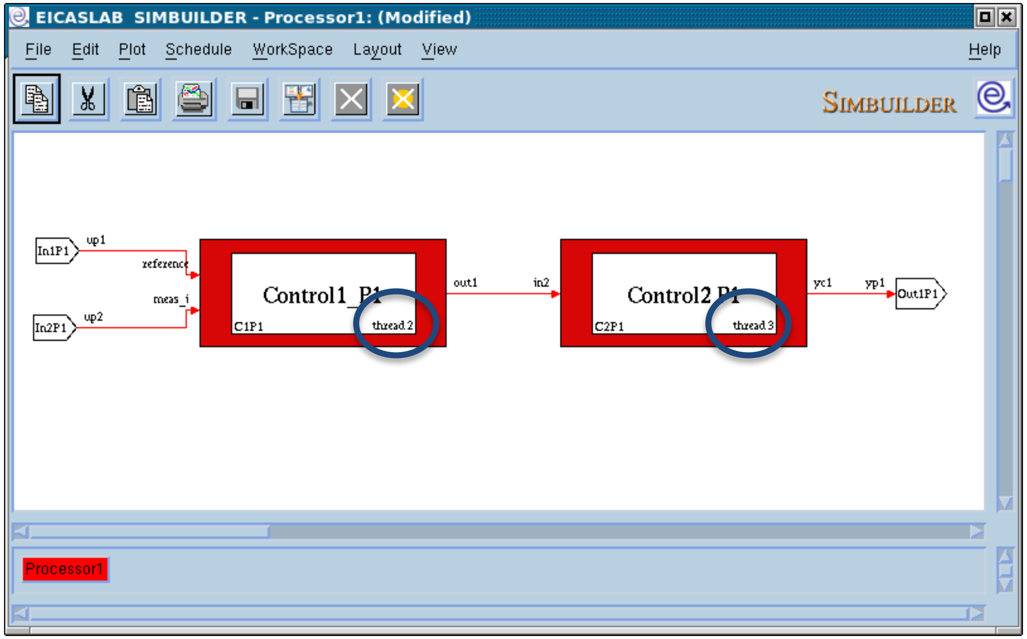
Thread assignment

Every Control block has to be associated to a thread.



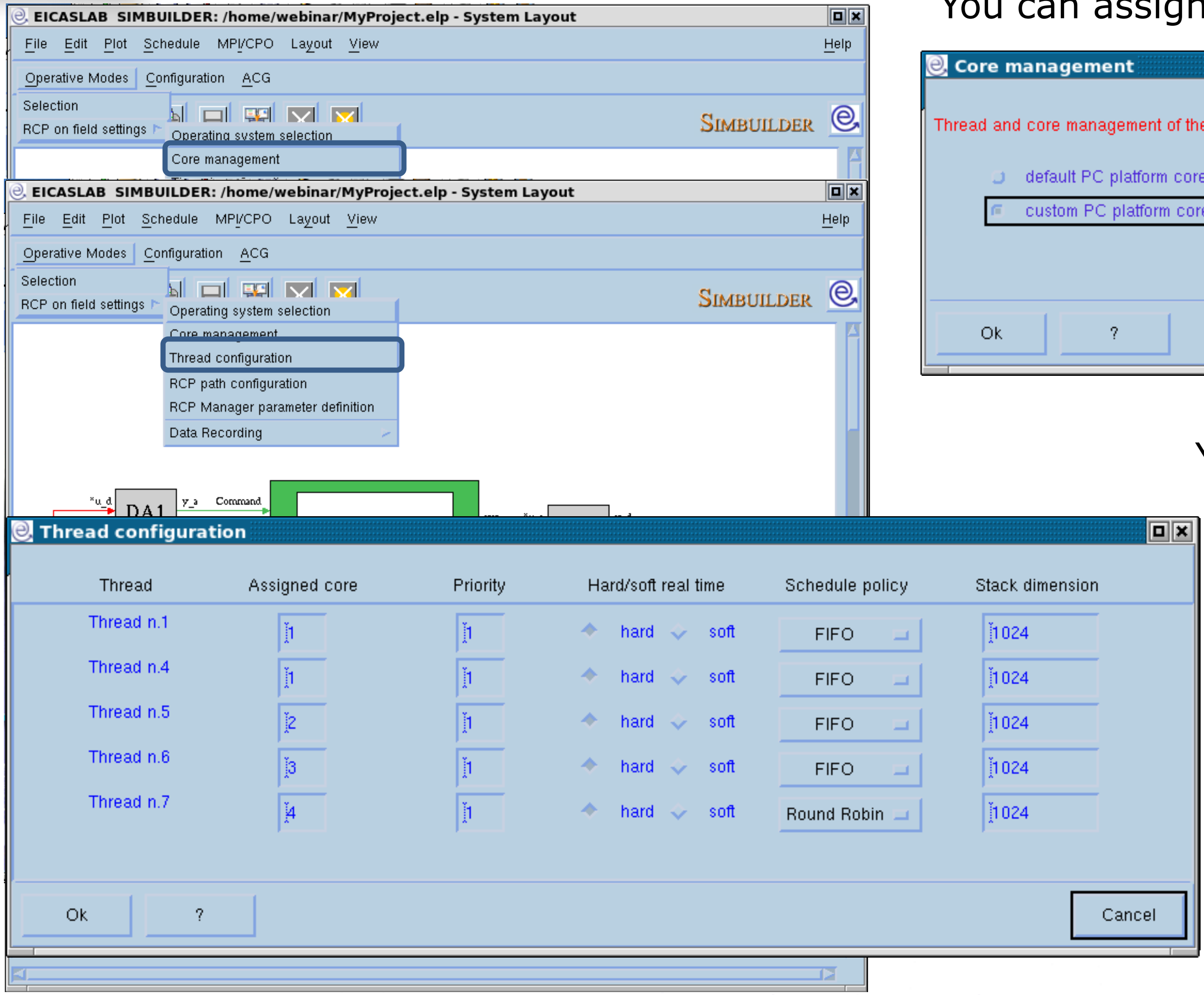


Select the thread in which you want to execute the block: 2



Control scheduling in RCP phase

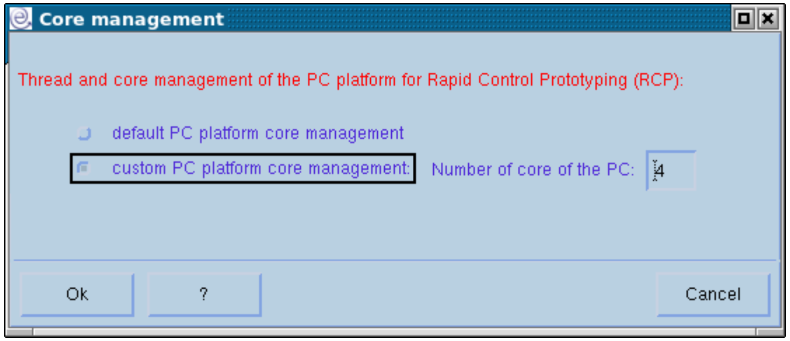
Core & thread setting



Thread configuration

Thread	Assigned core	Priority	Hard/soft real time	Schedule policy	Stack dimension
Thread n.1	1	1	● hard ○ soft	FIFO	1024
Thread n.4	1	1	● hard ○ soft	FIFO	1024
Thread n.5	2	1	● hard ○ soft	FIFO	1024
Thread n.6	3	1	● hard ○ soft	FIFO	1024
Thread n.7	4	1	● hard ○ soft	Round Robin	1024

You can assign to each thread a core.



Core management

Thread and core management of the PC platform for Rapid Control Prototyping (RCP):

○ default PC platform core management

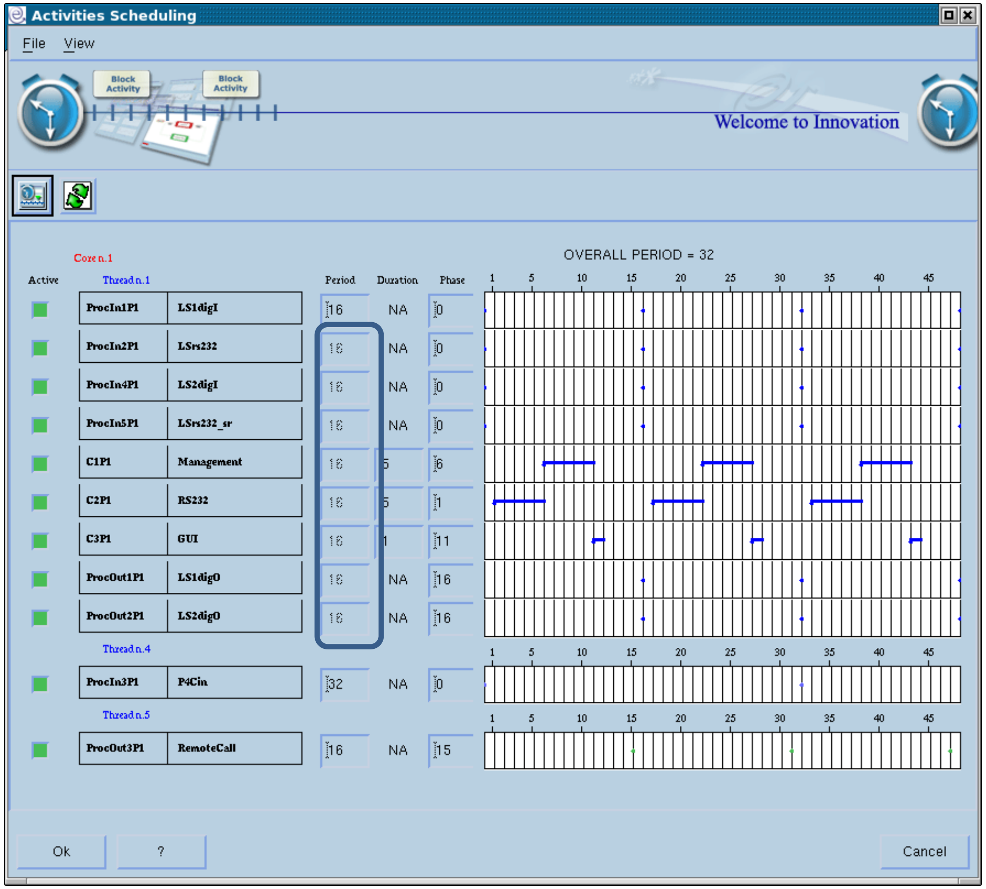
● custom PC platform core management: Number of core of the PC: 4

You can configure the threads:

- the assigned core,
  (in case of 'custom core management')
- the priority,
- the hard real time execution capability,
  (hard or soft),
- the schedule policy,
  (FIFO or Round Robin),
- the stack dimension.

Control scheduling in RCP phase

The scheduling window and the Scheduling constraints



Core n.1

Active	Thread n.1		Period	Duration	Phase
🟩	ProcIn1P1	LS1digI	16	NA	0
🟩	ProcIn2P1	LSrs232	16	NA	0
🟩	ProcIn4P1	LS2digI	16	NA	0
🟩	ProcIn5P1	LSrs232_sr	16	NA	0
🟩	C1P1	Management	16	5	6
🟩	C2P1	RS232	16	5	1
🟩	C3P1	GUI	16	1	11
🟩	ProcOut1P1	LS1digO	16	NA	16
🟩	ProcOut2P1	LS2digO	16	NA	16

Active	Thread n.4		Period	Duration	Phase
🟩	ProcIn3P1	P4Cin	32	NA	0

Active	Thread n.5		Period	Duration	Phase
🟩	ProcOut3P1	RemoteCall	16	NA	15

The control functions are listed core by core in case of custom core management, and always thread by thread.

- ○ All the Control functions belonging to a same thread must have the same period.
- ○ Two Control functions belonging to a same thread must not be overlapped.

EICASLAB supports you and prevents you to make scheduling errors.

All the scheduling problems occurred during the Experimental trial execution are reported to the user (e.g. Control function duration exceeding).