Activity scheduling in EICASLAB™

Welcome to Innovation

Via Vincenzo Vela, 27 10128 Torino - ITALY (IT)
Tel. +39 011 56 23 798 +39 011 56 23 088
Fax +39 011 43 60 679
www.eicas.it

TABLE OF CONTENT

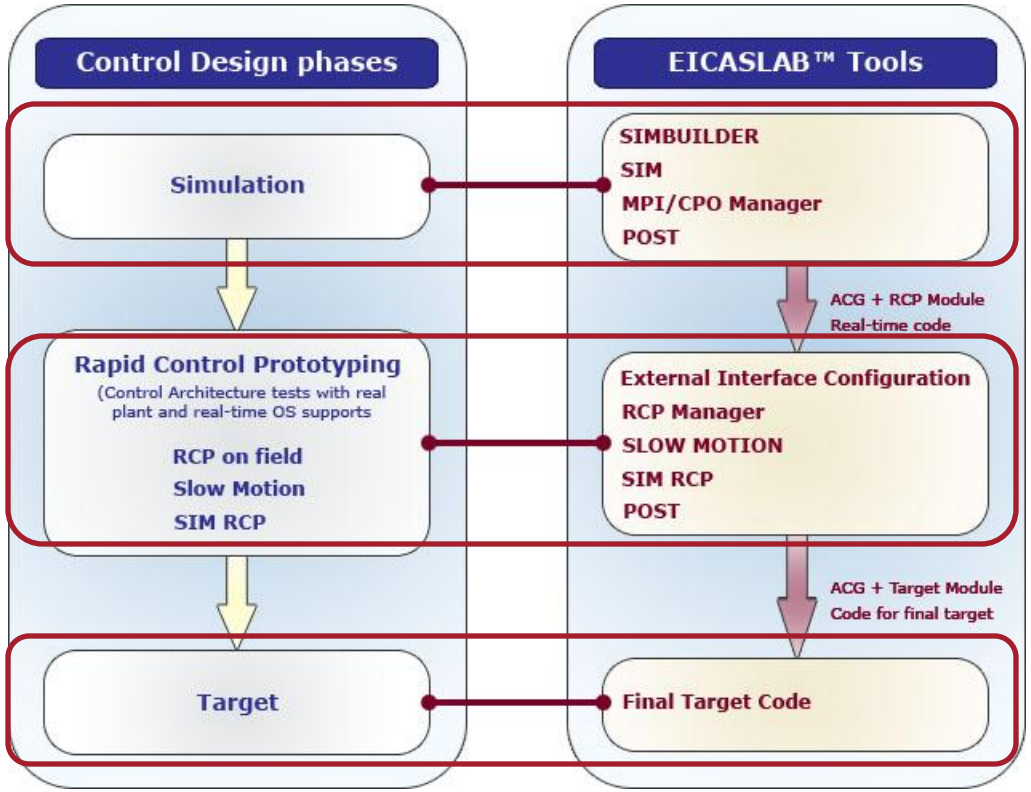
- Activity scheduling as professional support to design control hardware & software architecture
- Scheduling in the Simulation phase
- Scheduling in the Rapid Control Prototyping phase

EICASLAB™ assists you in all the control design phases

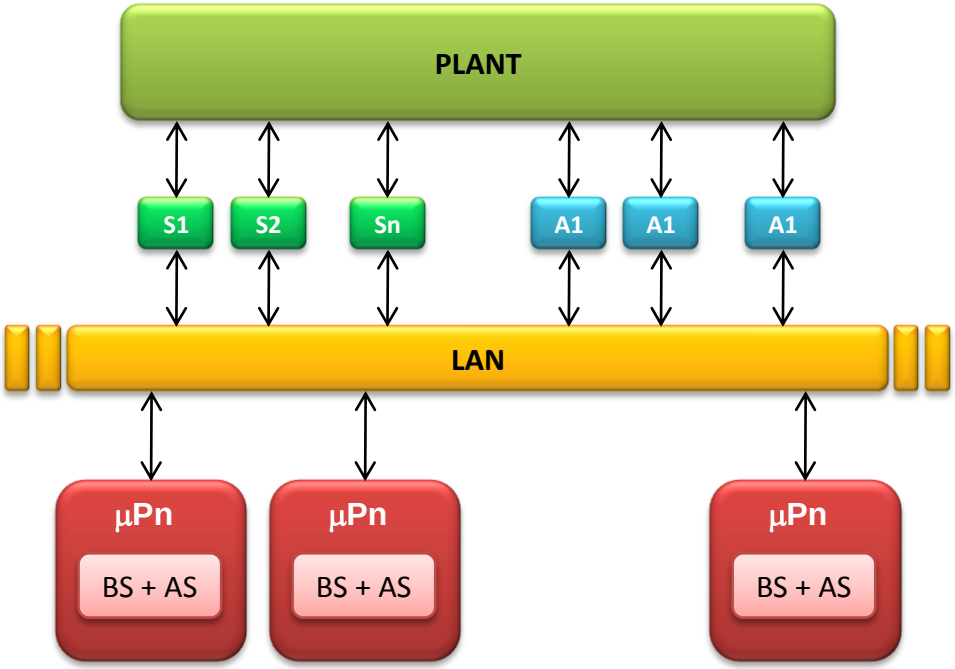
EICASLAB is the professional software suite that supports you in all the control design phases: from the system concept to the code generation for the final target.

Specifically, EICASLAB provides a professional support and specific tools devoted to the following control design phases:

- **Simulation**
- **Rapid Control Prototyping** (RCP)
- **Code generation for the final target**



EICASLAB™ takes particular care of the control software & hardware architecture



EICASLAB allows you to develop hardware architectures including multi-processors and software architectures including multi-level hierarchical control function.

A support is given for allocating each control activity to a given processor.

The scheduling of all the activities involved in a control system is a key task for the successful development of the control system itself.

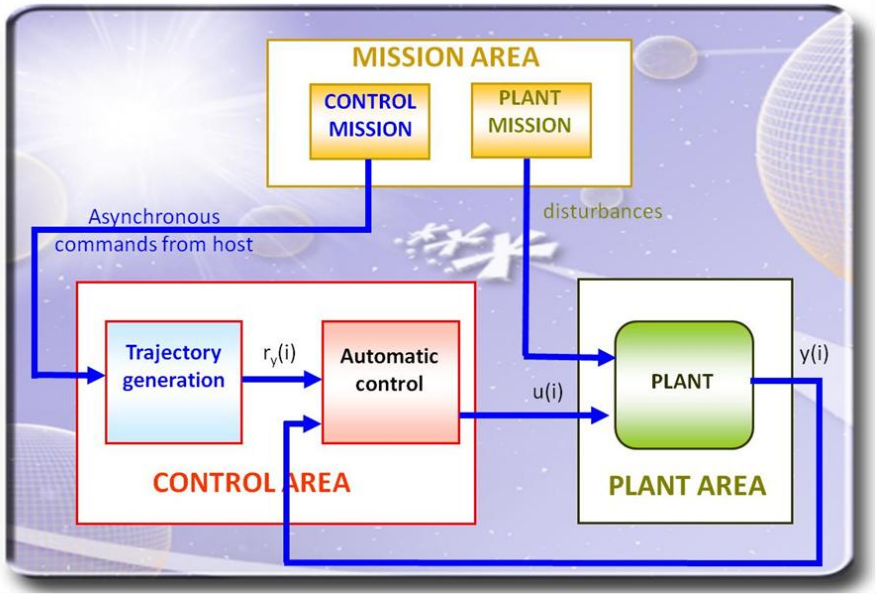
EICASLAB provides a fundamental and professional support for a correct scheduling of your system in all the control design phases.

Welcome to Innovation

The Scheduling in the Simulation phase

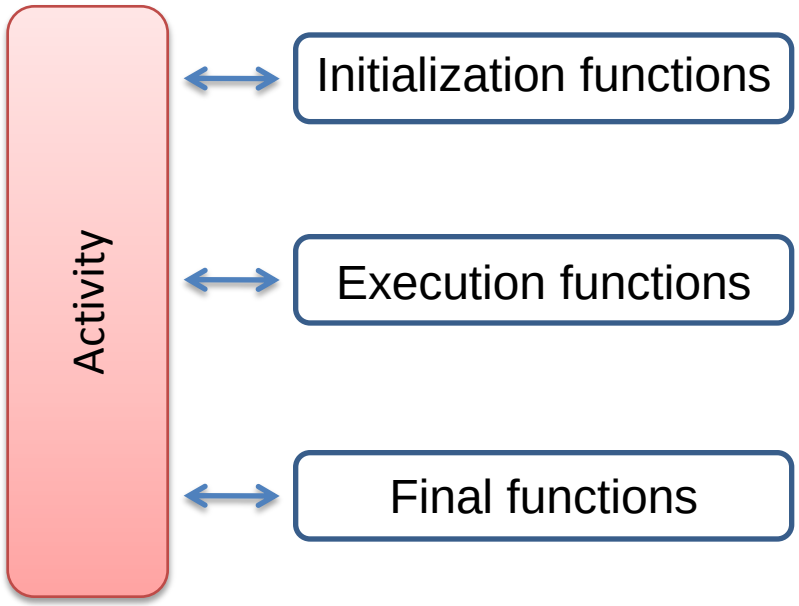
In the Simulation phase three different working areas are available:

- the **Control Area**,
- the **Plant Area**,
- the **Mission Area**.



The designer must schedule all the activities of these 3 areas.

Each activity is composed by:



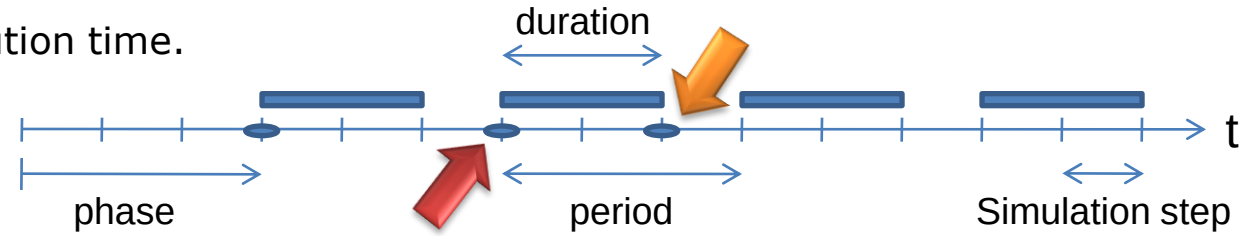
The Scheduling in the Simulation phase

The activities scheduling

The designer must fix a **simulation step**,
which represents the time resolution applied in the simulation of your overall project.

The periodic functions are 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.



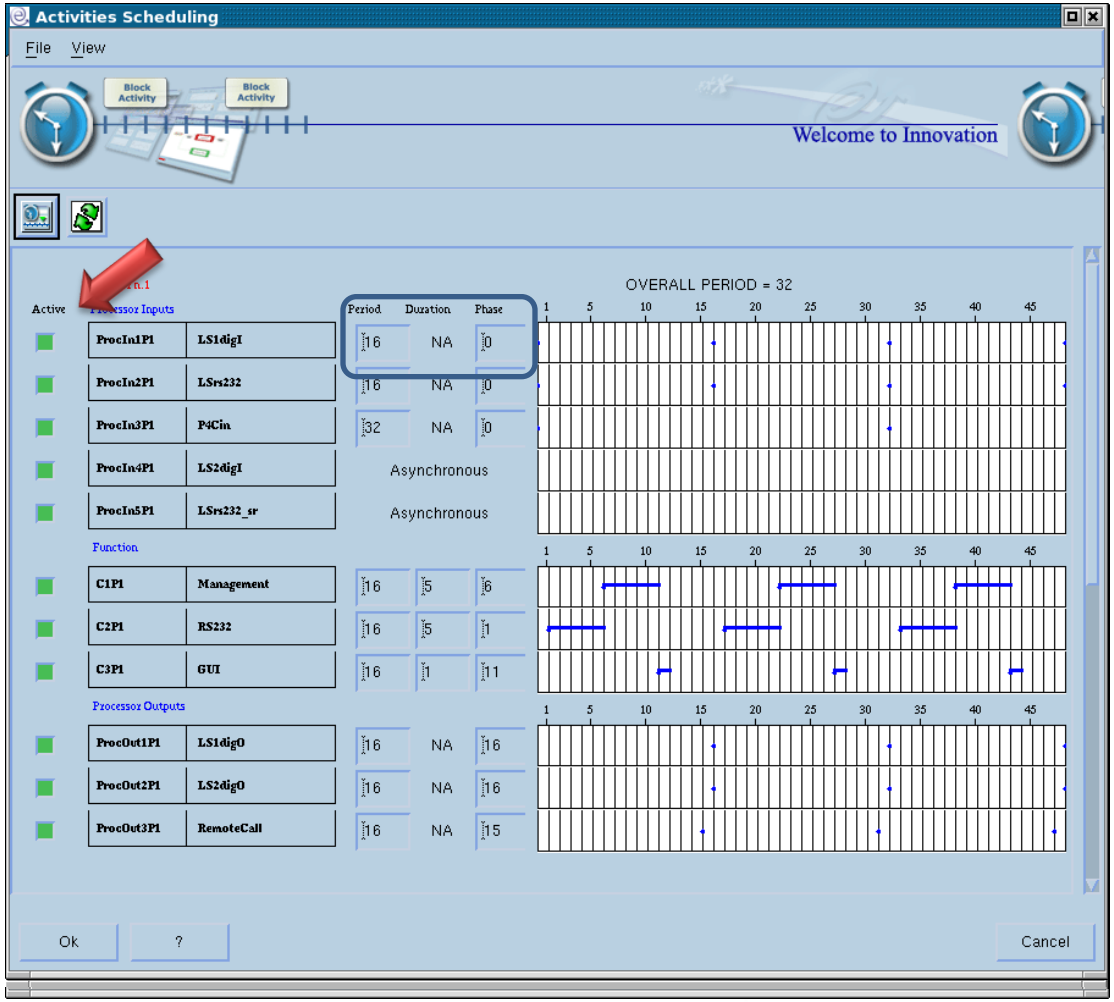
To guarantee the correct scheduling of the activity it is necessary to take into account its duration

Two periodic functions are available:

execution function	It executes all the operations that the activity must perform each time it is scheduled	It is called when the activity is scheduled (considering its phase and its period)
output function	It computes and updates the outputs of the activity as a function of its current state	It is called after the fixed duration

The Scheduling in the Simulation phase

The Activities schedule window



All the activities are listed in the following order:

- **Plant Area** activities,
- the **Converters** activities,
- **Mission Area** activities,
- **Control Area** activities.

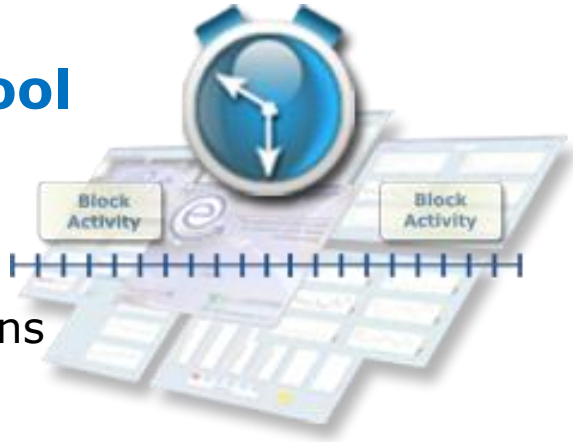
The 'Active' push enables or disables the corresponding activity.

The **Period**, **Duration** and **Phase** of each activity can be set.

EICASLAB supports you and prevents you to make scheduling errors.

The Scheduling in the Simulation phase

The EICASLAB scheduler of the SIM tool



The SIM tool manages the scheduling of activities
by means of the **EICASLAB scheduler**,
an advanced engine that allows to run like real-time simulations

The EICASLAB scheduler is **the core of the time scheduling algorithms.**

On the basis of the constraints fixed in the 'Activities Scheduling' window.

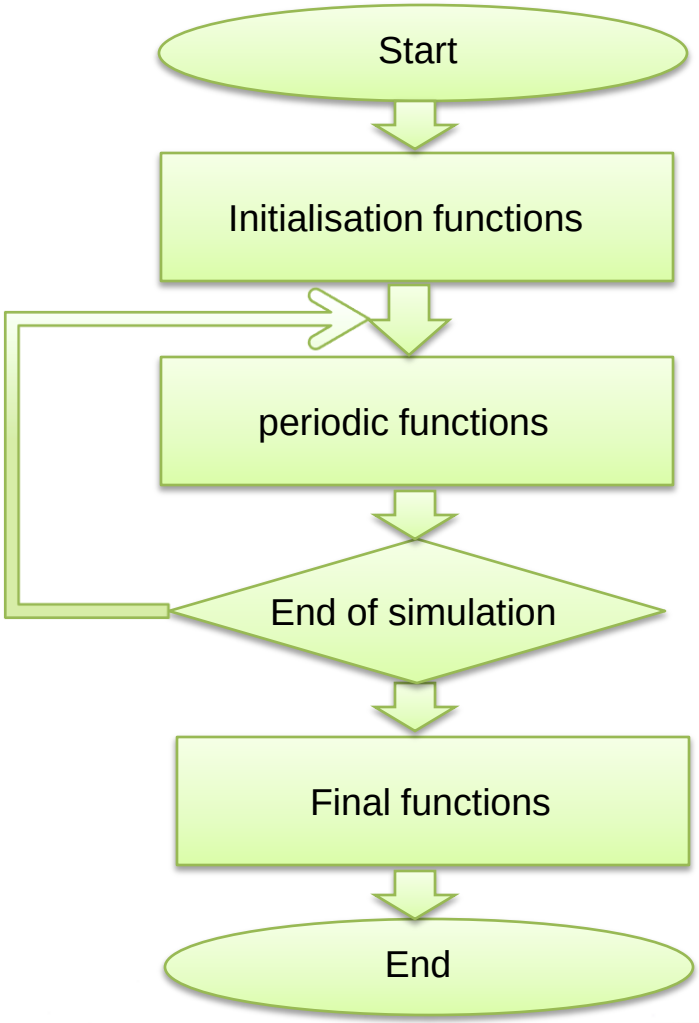
It defines the order in which the functions of a given activity have to be executed

It schedules the data trasmission among all the activities

The EICASLAB scheduler is directly linked to the clock of the CPU:
if the simulation is fast enough it may allow to run a real-time simulation.

The Scheduling in the Simulation phase

The EICASLAB scheduler simulations



The EICASLAB scheduler performs a simulation with period equal to the simulation step.

Intialization functions

- Project parameters reading
- Project initial state reading
- User initialisation functions

Periodic functions

- Output functions (Updating of the outputs)
- Signal Propagation
- Plot update (SIM plotting and POST file recording)
- Execution functions
- Time update

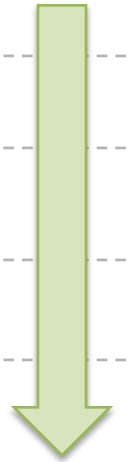
Final functions

- User final functions
- Project final state saving

The Scheduling in the Simulation phase

Initialisation functions

The schedulable activities have a set of 'initial functions' called just once at the beginning of the simulation.

Function description	Activities concerned	Order of scheduling
Parameter file reading	All activities that have parameters	
Resolution file reading	Only Continuous Plant	
Initial state file reading	All activities that have state variables	
Control design	Only the control functions	
User initialisation function	Only activities programmed in ANSI C language	

The Scheduling in the Simulation phase

Final functions

The schedulable activities have a set of 'final functions' called just once at the end of the simulation.

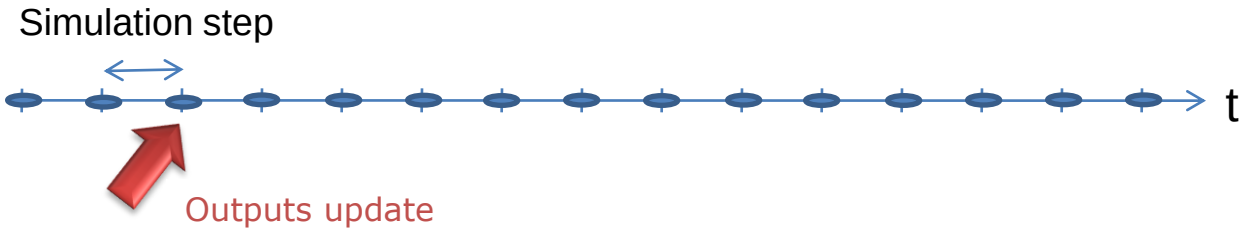
Function description	Activities concerned	Order of scheduling
User final function	Only activities programmed in ANSI C language	
Final state file writing	All activities that have state variables	

The Scheduling in the Simulation phase

Continuous Plant scheduling

The Continuous Plant is the mathematical fine model of the plant to be controlled.
It is a dynamic system - with state and outputs variables - that can be represented through a system of differential equations.
The integration of such a system is performed through a smart integration procedure embedded in EICASLAB.

The outputs of the Continuous Plant are updated at each simulation step.



State equation function	Computation of the State derivative	It is called by the EICASLAB routine that solves the system of differential equations
output function	It computes the outputs of the Continuous Plant as a function of its current state	It is called at each simulation step

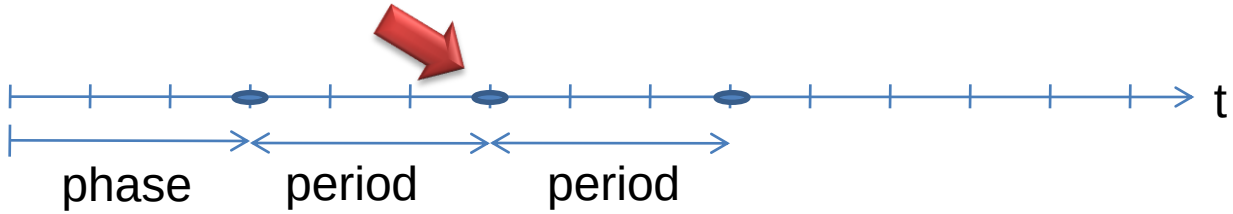
The Scheduling in the Simulation phase

Instantaneous activities scheduling

The duration of some activities is null or negligible with respect to the duration of the other ones: they are called instantaneous.

Their scheduling is then defined just by the **Phase** and **period**.

Instantaneous activities
Experimental Data
Elementary missions
A/D and D/A Converters
Processor Input/Output



Instantaneous activities periodic functions

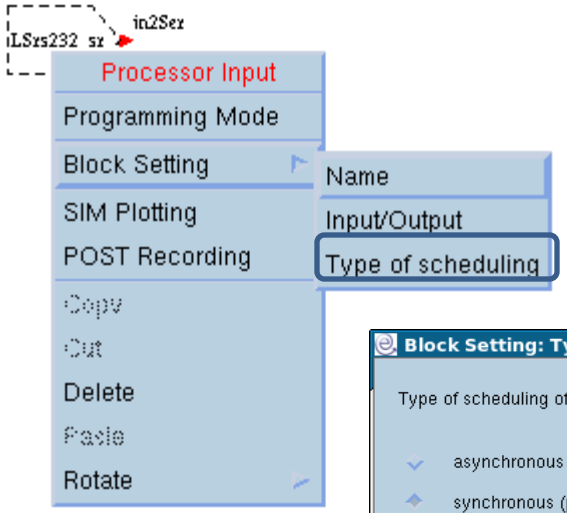
execution function	executes all the operations that the activity must perform each time it is scheduled and updates its outputs	called when the block is scheduled (considering its phase and its period)

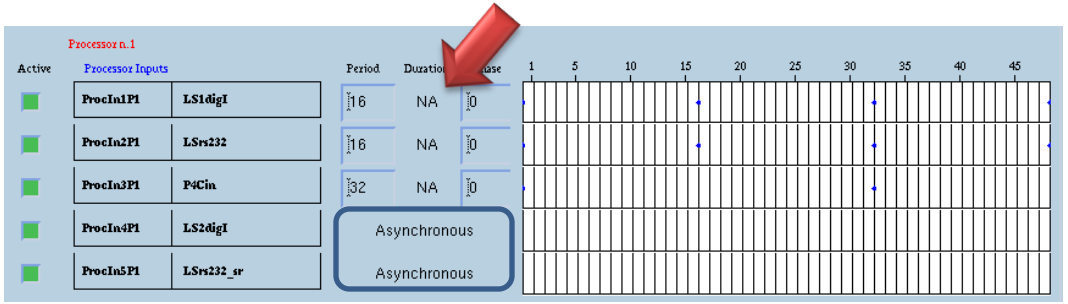
The Scheduling in the Simulation phase

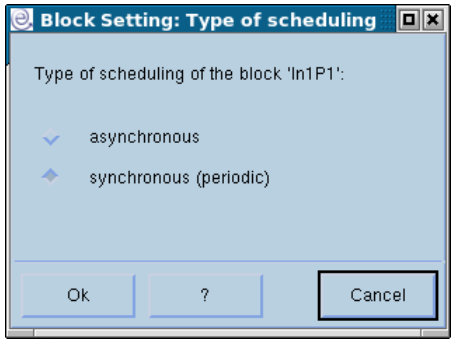
Processor Inputs: Synchronous and asynchronous activities

The processor inputs represent the interrupt activities that receive and process the inputs coming in your processor.
Such inputs can be received:

- with a given periodicity (**synchronous** processor inputs),
- through an asyncrounous communication (**asynchronous** processor inputs).







The scheduling in the Rapid Control Prototyping phase

The Rapid Control Prototyping phase allows you to test and validate your control software by transferring it in a smart PC Platform equipped with a suitable RTOS and by directly piloting your plant, before the transfer in the final target.

The Rapid Control Prototyping phase is fundamental for minimizing time and costs in the control tuning in field.

The EICASLAB Rapid Control Prototyping Multicore PC Platform

System Modelling

Like-Real time simulation

Model Parameter Identification

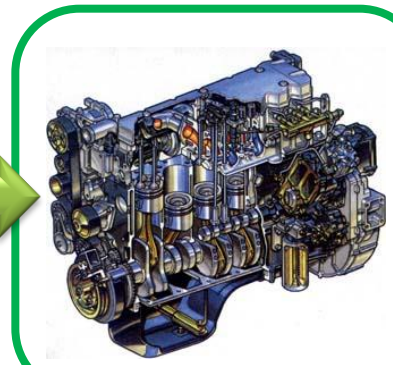
Control Parameter Optimisation

Data Post-processing

Simulation

EICASLAB Rapid Control Prototyping Multicore PC Platform

Real Time Control Algorithms



Real Plant

Real-Time OS

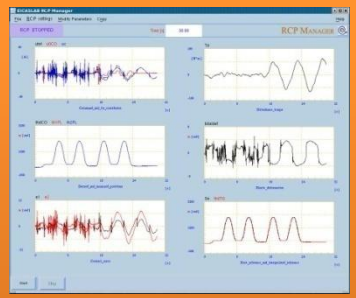
Multi-core and Multi-threading RT application

Interfaces and communications

RCP Manager

POST and Slow Motion data Recording

RCP Manager GUI



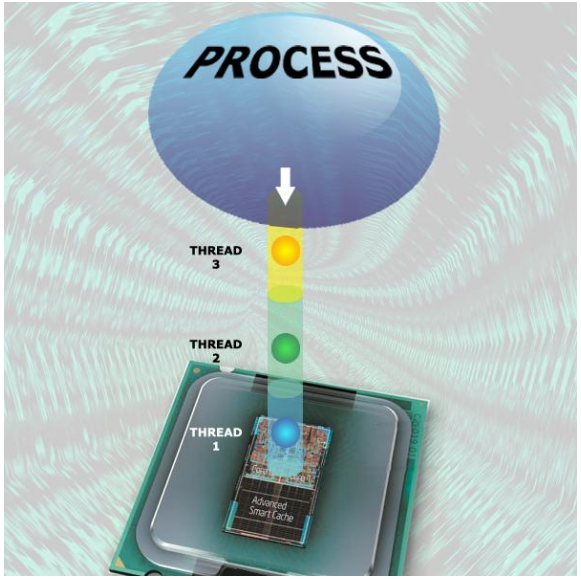
RCP

EICASLAB™ Advanced features

The scheduling in the Rapid Control Prototyping phase

Multi-core & multi-thread applications

A **multi-core processor** is a processing system composed by two or more independent cores.
While in single core CPU systems, it is not possible to execute more than one piece of code at a same time, in a multi-core processor each core can run a portion of code so that a real concurrent execution is achieved.



A **thread** is a unit of executable code. In a thread-based multitasking environment all processes have at least one thread of execution, which is called the main thread. Each multithreaded process starts with the main thread that creates one or more additional child threads.

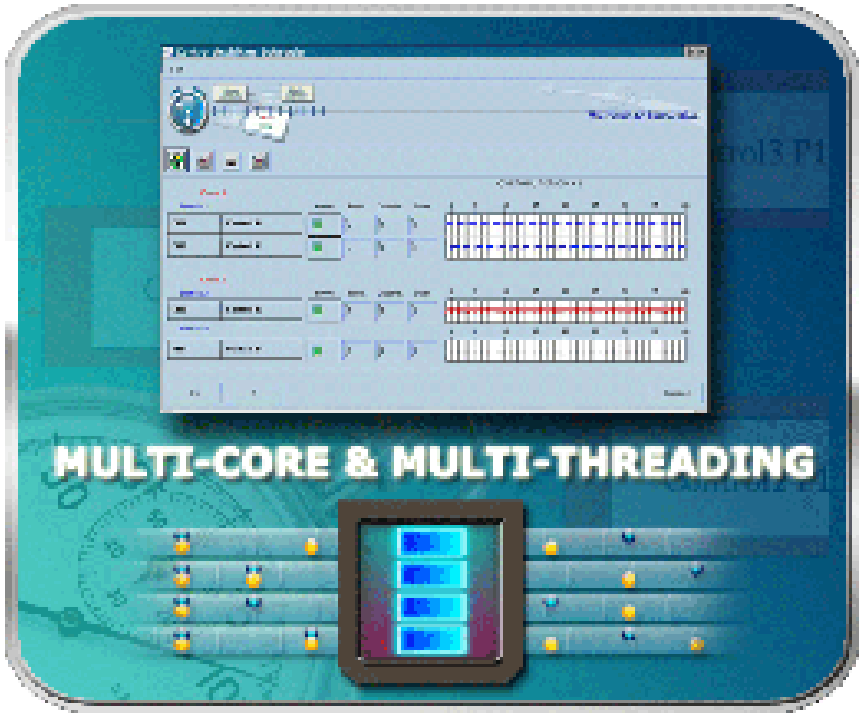
Multithreading changes the fundamental architecture of a program. Unlike a single-threaded program that executes in a strictly linear way, a multithreaded program executes portions of itself concurrently. Then a single program can perform two or more tasks concurrently. The true concurrent execution is possible only in a multiple-CPU or multi-core CPU systems.

The scheduling in the Rapid Control Prototyping phase

Scheduling, cores & threads

The Control system works directly with the real Plant, then just the Control Area must be scheduled.

To guarantee the correct emulation of the Target control architecture, in the **Rapid Control Prototyping** phase, the user could have the need of reviewing the control scheduling.

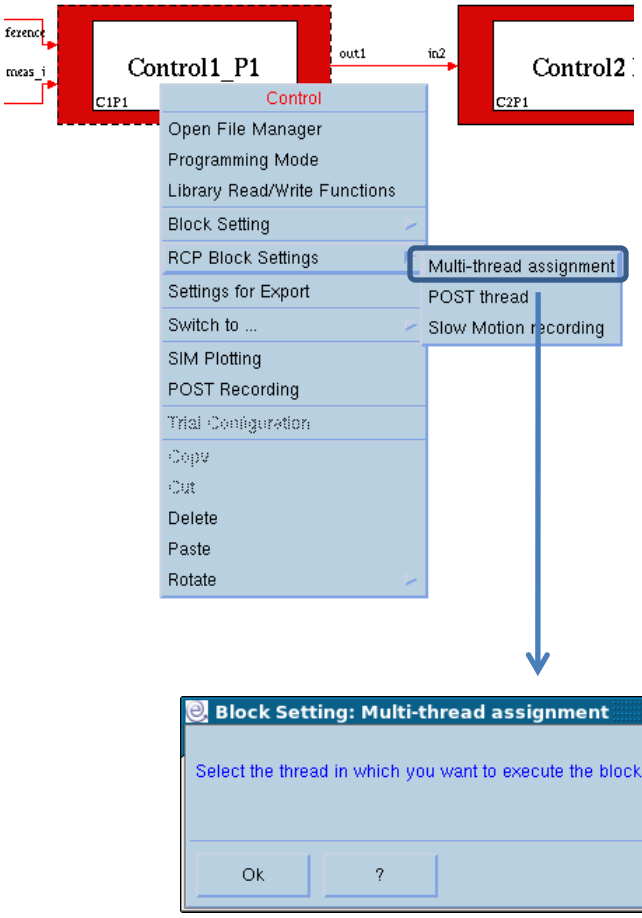


The new scheduling requires to organize the control functions in **threads**, that will be distributed on the **cores** of the multi-core CPU available in the PC platform, thanks to the application in EICASLAB of multi-threading and multi-core programming techniques.

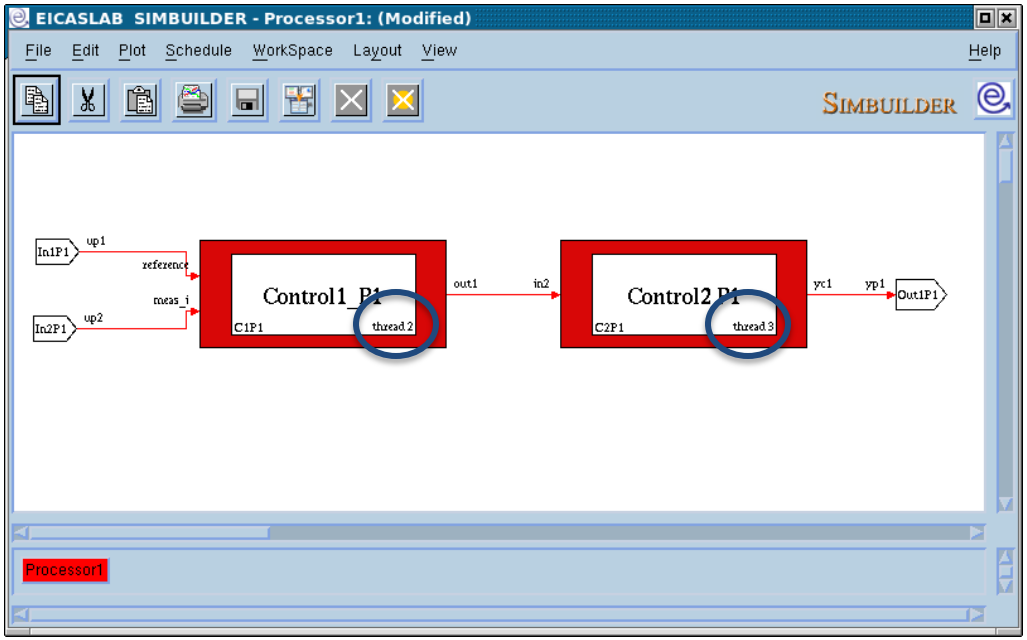
There is no more simulation step: instead of it the clock tick of the real-time operative system is considered.

The scheduling in the Rapid Control Prototyping phase

Thread of the activities

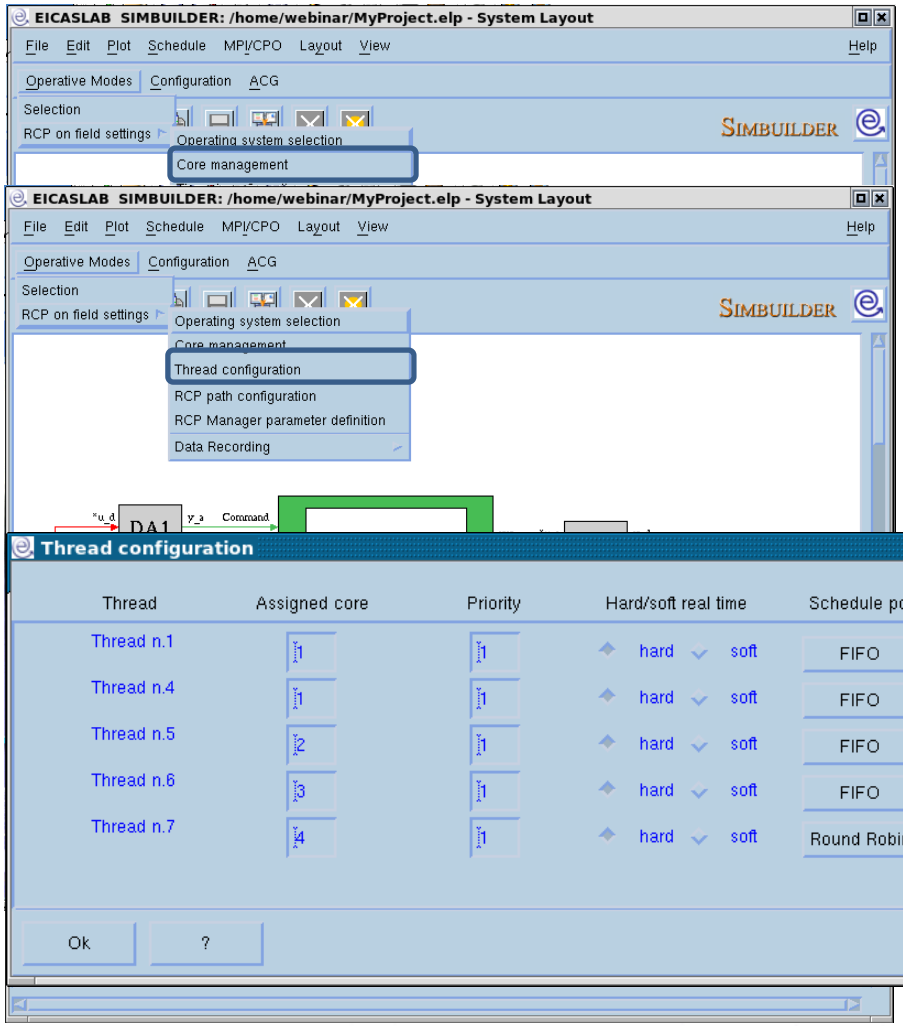


Every Control function and Processor Input/Output has to be associated to a thread.

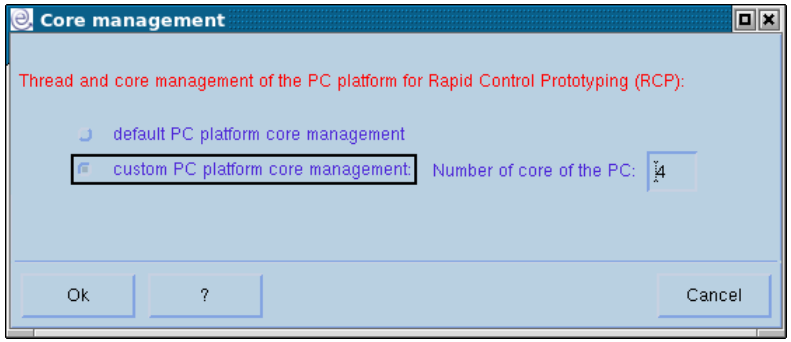


The scheduling in the Rapid Control Prototyping phase

Core & thread setting



You can assign to each thread a core.

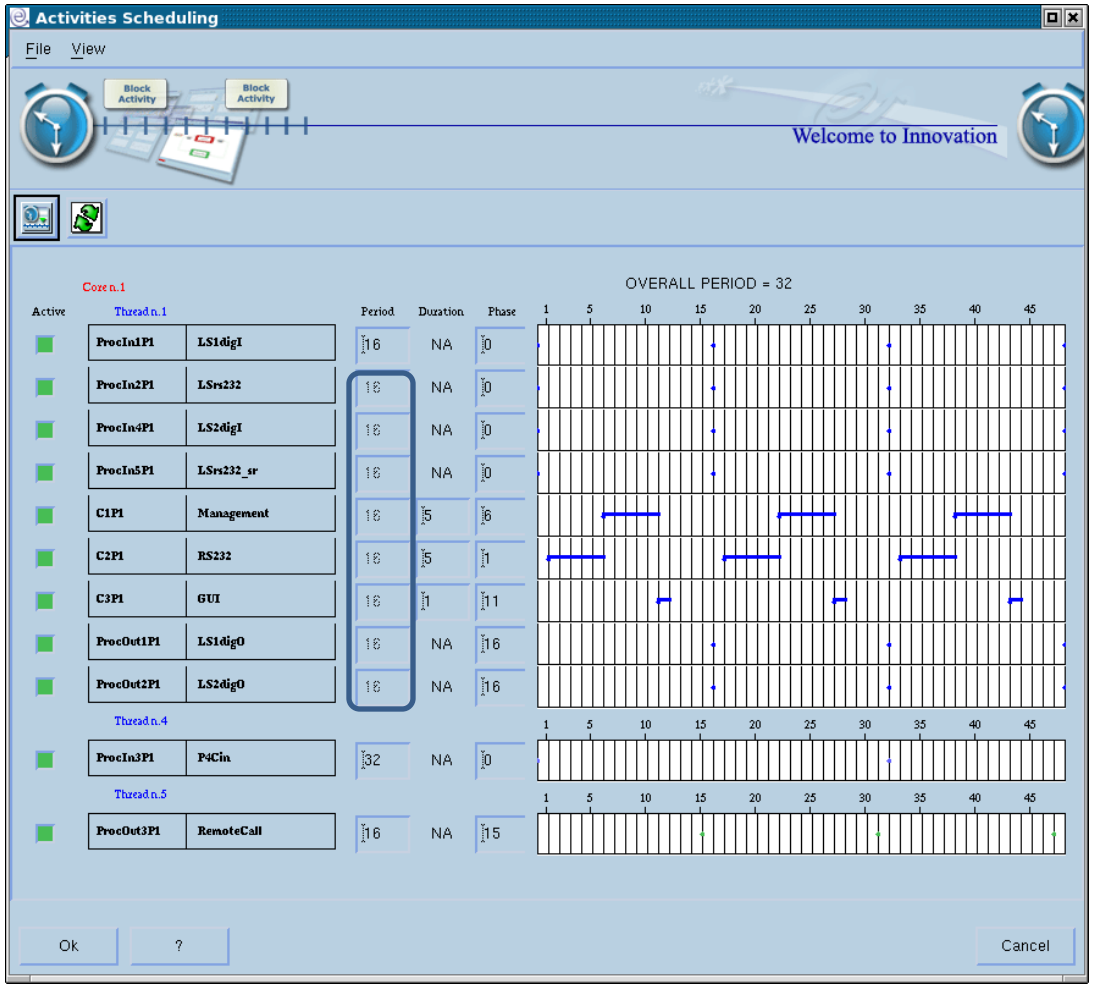


You can configure the threads:

- 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.

The scheduling in the Rapid Control Prototyping phase

The scheduling window and the Scheduling constraints



The control functions are listed 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,
- Each asynchronous Processor Input has its specific thread.

EICASLAB supports you and prevents you to make scheduling errors.

EICASLAB™

The Professional Software Suite
for Automatic Control Design
and Forecasting

for Linux & for Windows

www.eicaslab.com

Welcome to Innovation