The main Working Areas for designing in EICASLAB™

The Control Area

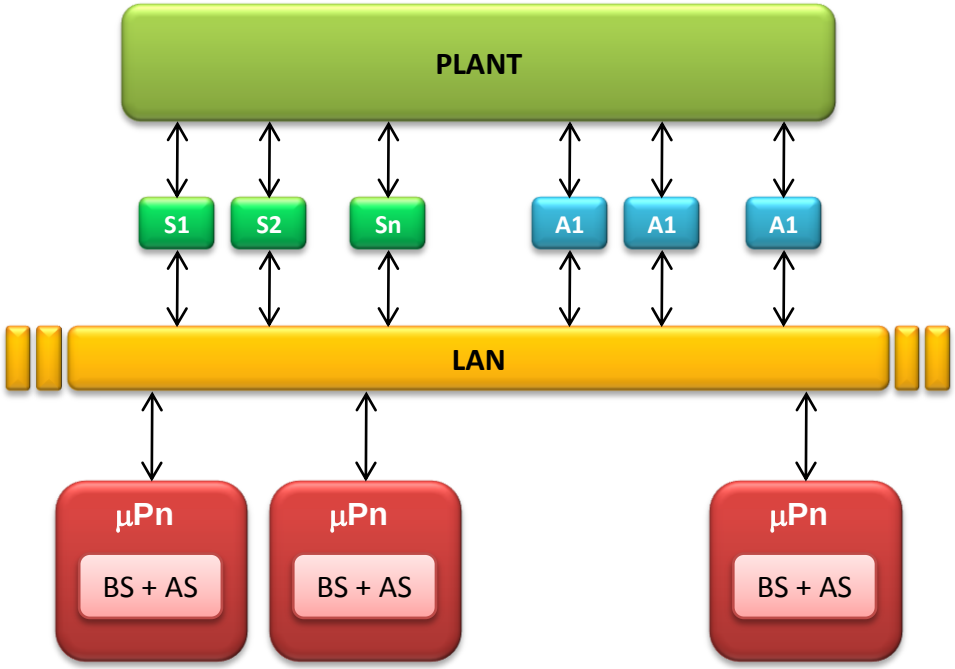
excellence and passion in automatic control design
EICAS Automazione S.p.A.
Via Vincenzo Vela, 27 10128 Torino - ITALY (IT)
Tel. +39 011 56 23 798 +39 011 56 23 088
Fax +39 011 43 60 679
www.eicas.it

TABLE OF CONTENT

- General description of the Control Area
- The Control block

Control software and hardware architecture

Concept



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

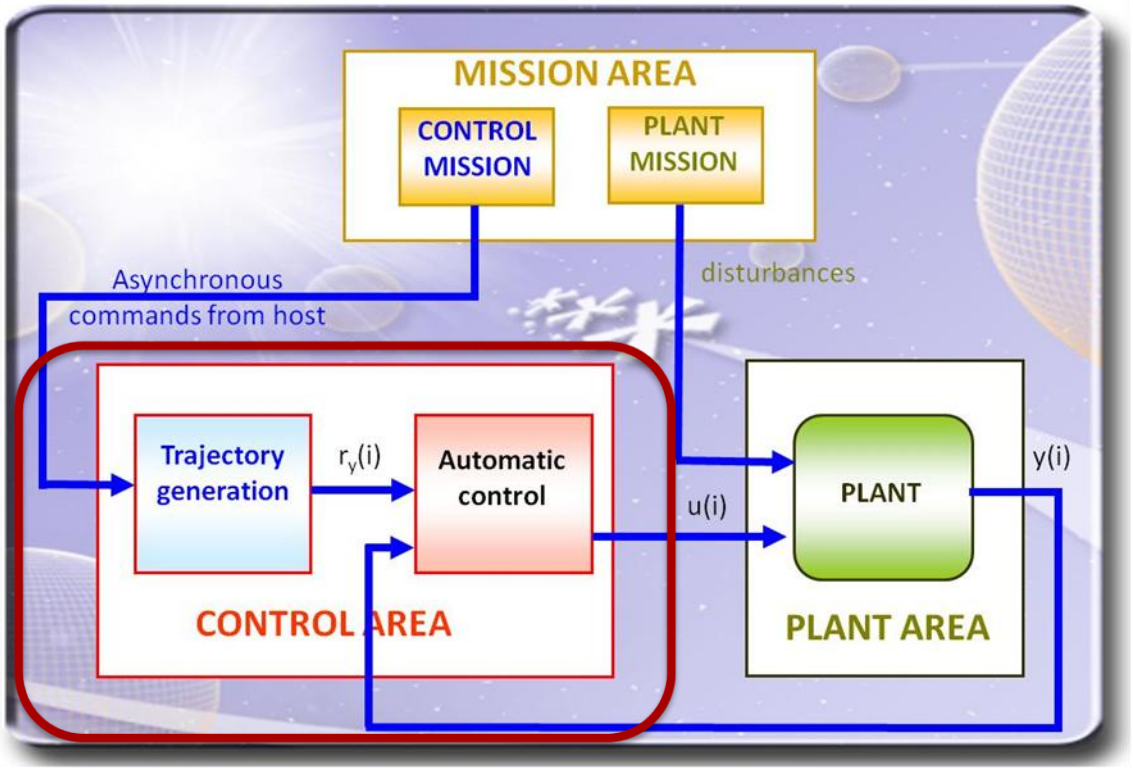
HW Components	
S1	Sensors
A1	Actuators
LAN	Local Area Network
uP	Microprocessor

SW Components	
BS	Basic Software
AS	Application Software

Control software and hardware architecture

The Control Area

EICASLAB provides a ***Working Area*** named ***Control Area*** which is **specifically devoted** to the development of the overall control system architecture.

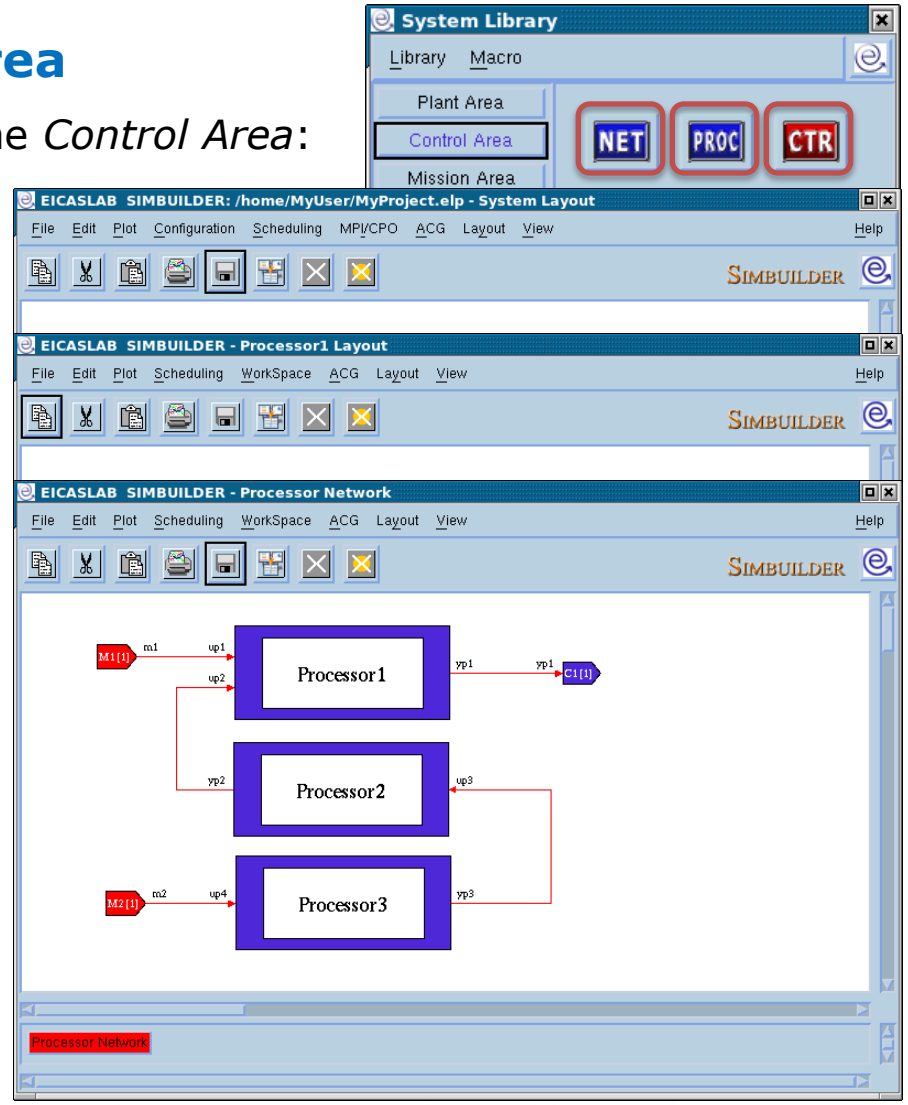


The Control Area

The blocks of the Control Area

The following blocks may be programmed in the *Control Area*:

Block name	Description	Block icon
Control	implement a single control function	CTR
Processor	Collect one or more schedulable control functions running on a single processor (mono or multicore)	PROC
Processor Network	Collect one or more Processors, allowing to simulate multi-processor hardware architectures	NET



The Control block in EICASLAB

The Control block

Concept

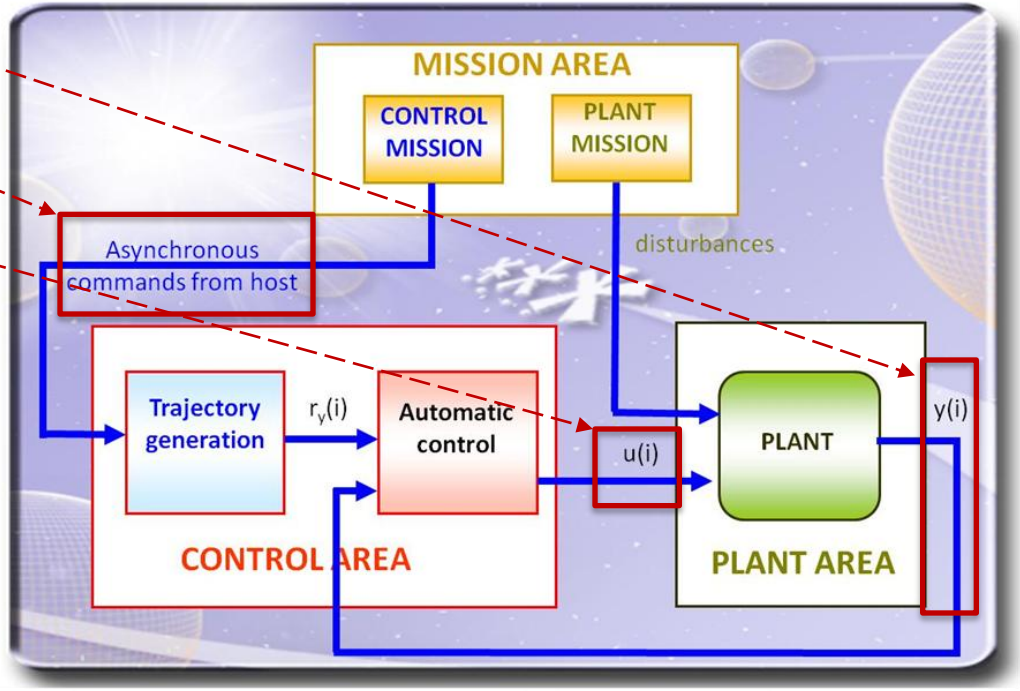
The Control block is devoted to implement a single control function.

The Control receives as inputs:

- measurements coming from the *Plant Area*,
- references coming from the *Mission Area,*

and provides as outputs:

- commands for piloting the Plant.



The Control block

Programming modes

You can develop your Control block:

- **graphically** programming:
  you work on **graphical layouts** equipped with specific and oriented **libraries** that contain a set of suitable pre-defined blocks,

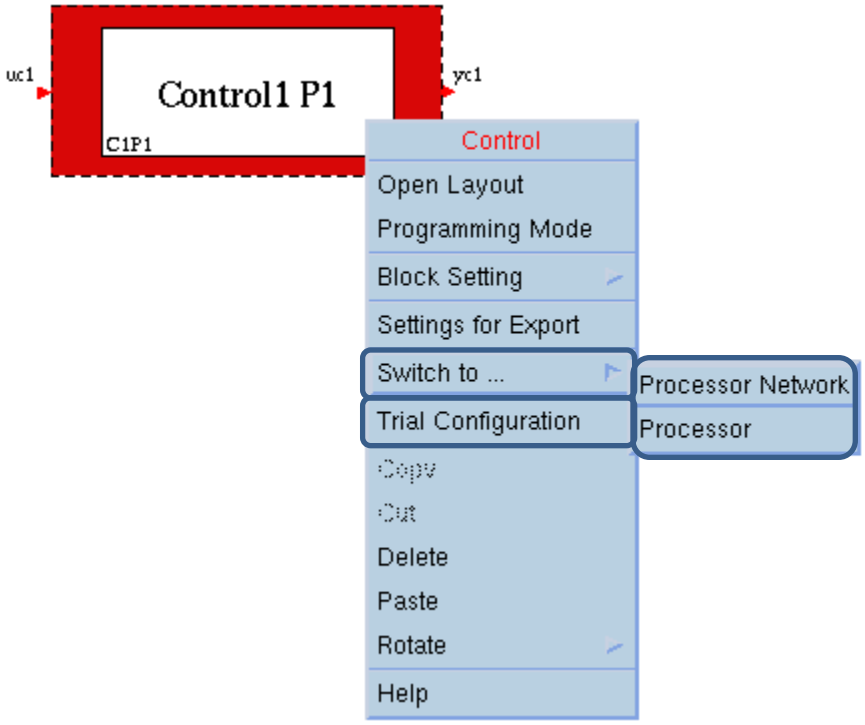
- programming with **ANSI C language**:
  EICASLAB allows an easy programming in ANSI C language
  by means of an open and customizable pre-organized structure
  that allows you to focus just on specific and crucial aspects of the control system to be programmed.
  You have at disposal a set of template files and libraries,

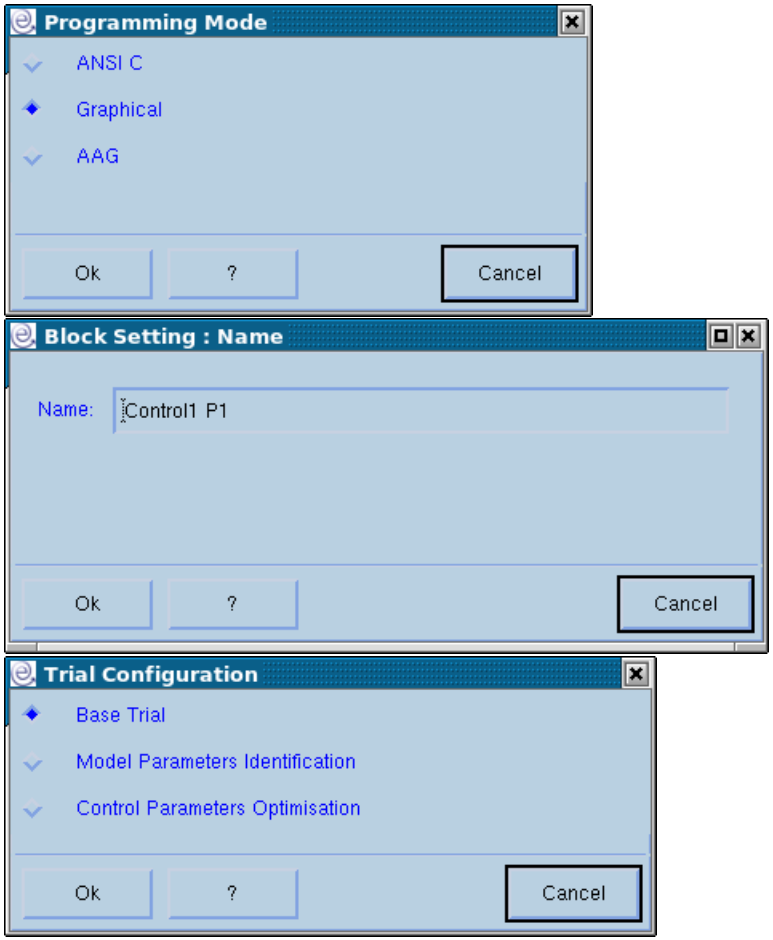
- Through the ***Automatic Algorithm Generation (AAG)*** advanced feature.

The Control block

Associated popup menu

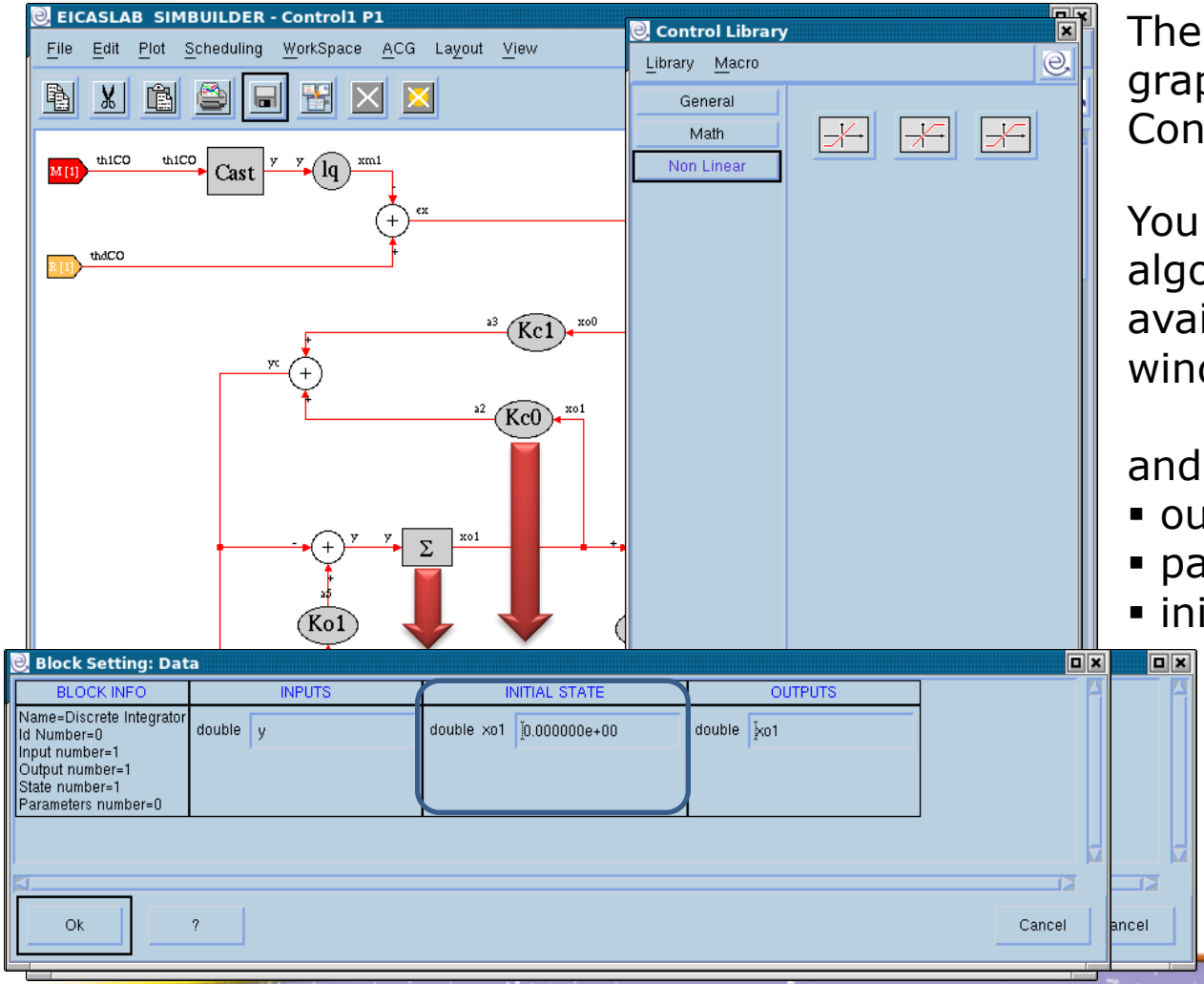
Popup menu of the Control block:





The Control block graphically programmed

The Control Layout



The Control Layout allows to graphically program the Control.

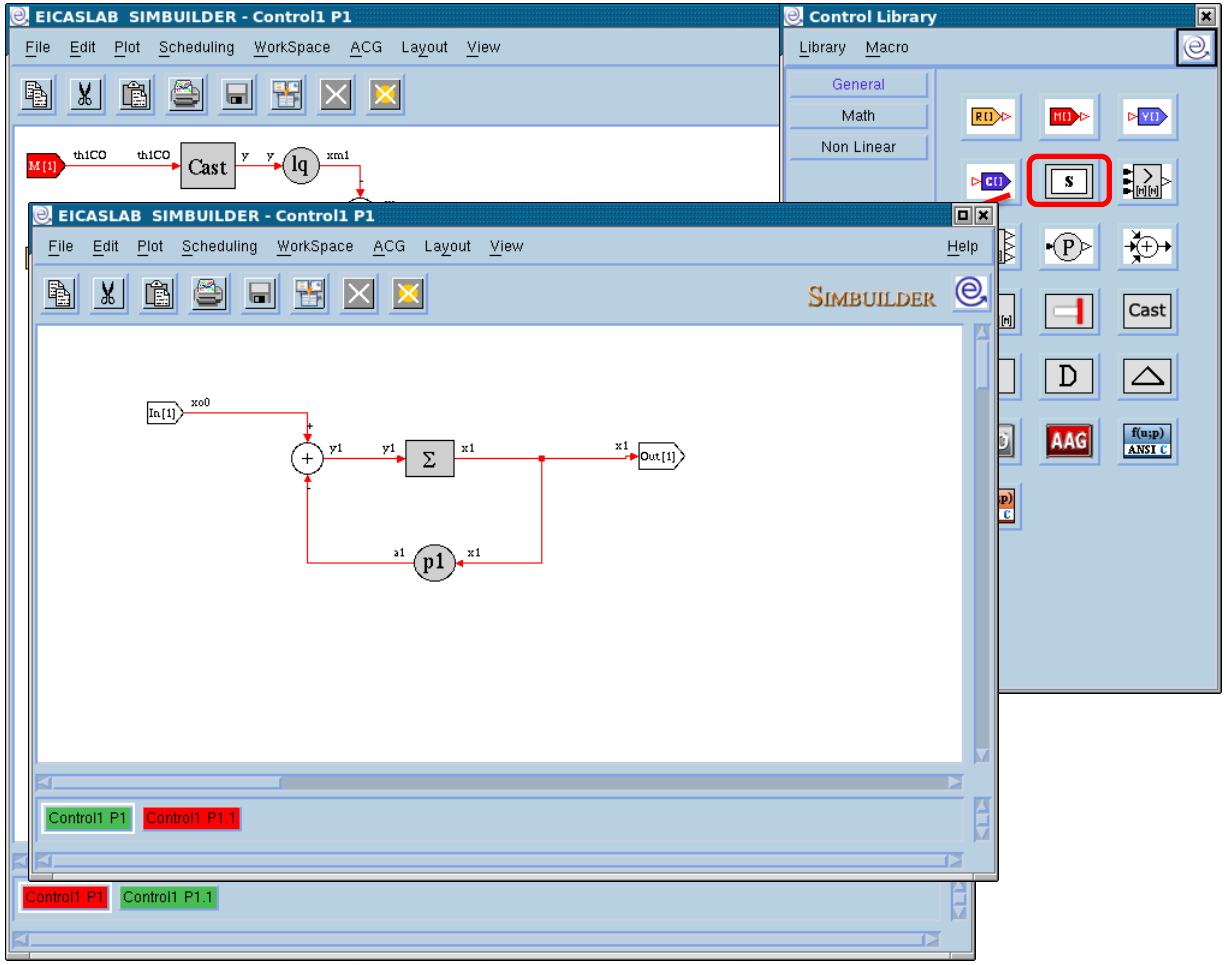
You can build your control algorithm by using the blocks available in the **Control Library** window,

and by setting their:

- outputs,
- parameters,
- initial states (dynamic blocks).

The Control block graphically programmed

The subsystems

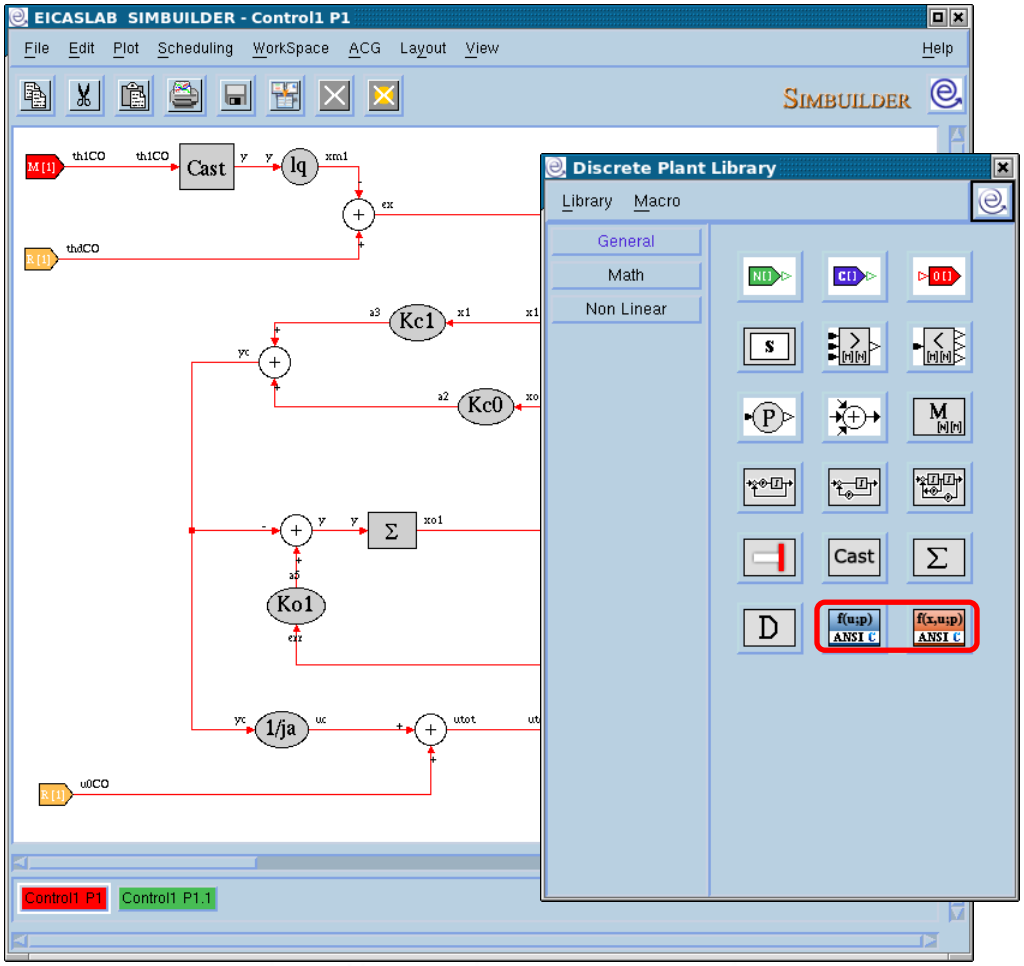


You can simplify the representation of your system by collecting parts of your block diagram in a block called **Subsystem**.

Double clicking on the subsystem opens the *Subsystem* layout, where you can use all the blocks available in the related library.
You can also create other subsystems in order to build a hierarchical block diagram.

The Control block graphically programmed

The ANSI C blocks



It is possible to use special blocks programmable in ANSI C language.

There are two types of blocks, allowing you to program in ANSI C language:

- static functions
  in this case the C block implements the function:
  y= f(u;par);

- dynamic functions
  in this case the C block implements the function:
  y= f(x,u;par);

(having indicated:
y: outputs, u inputs, x: states, par: parameters)

The Control block graphically programmed

The macros

The Control *library window* is **customizable** with user blocks called **'macros'**.

The macros are created by the user in order to complete the library according to the user needs.

The macros can be programmed:

- **graphically** (working on the Graphical Macro layout) or
- **in ANSI C language**.

They are then available in the library window of the layout, as all the other blocks and can be used in the current project.

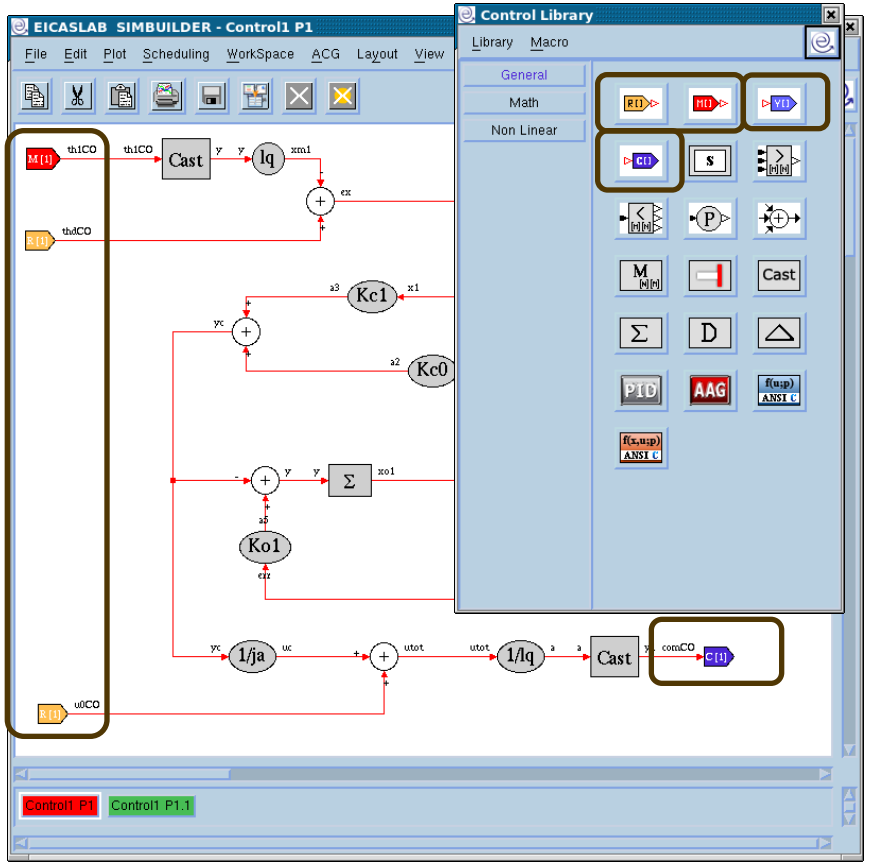
They can also be exported and then used in other projects.

The Control block graphically programmed

The Input/Output variables

In order to define the inputs and the outputs of a graphically programmed block:

insert inside the Graphical Layout the input / outputs blocks.



Control

- M[] Control Measure Input
- R[] Control Reference Input
- Y[] Control Data Output
- C[] Control Command Output

Welcome to Innovation

The Control programmed with ANSI C language

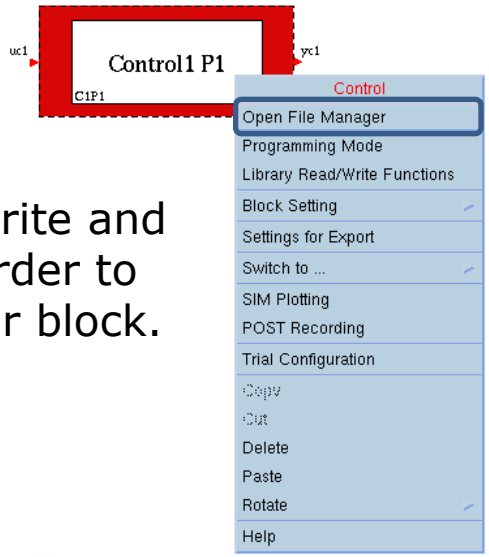
The file manager

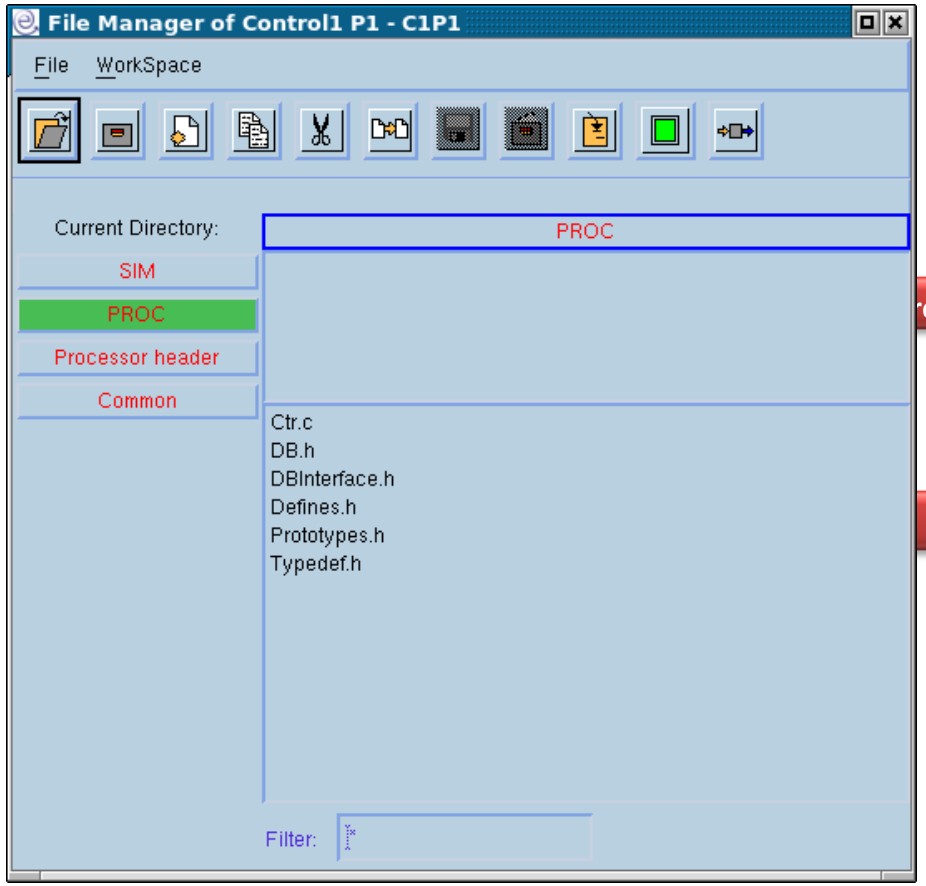
The Control programmed with ANSI C language has its own file manager through which it is possible to program the block.

EICASLAB provides a pre-organized structure: a set of template files subdivided in:

- data files,
- header files,
- ANSI C files,

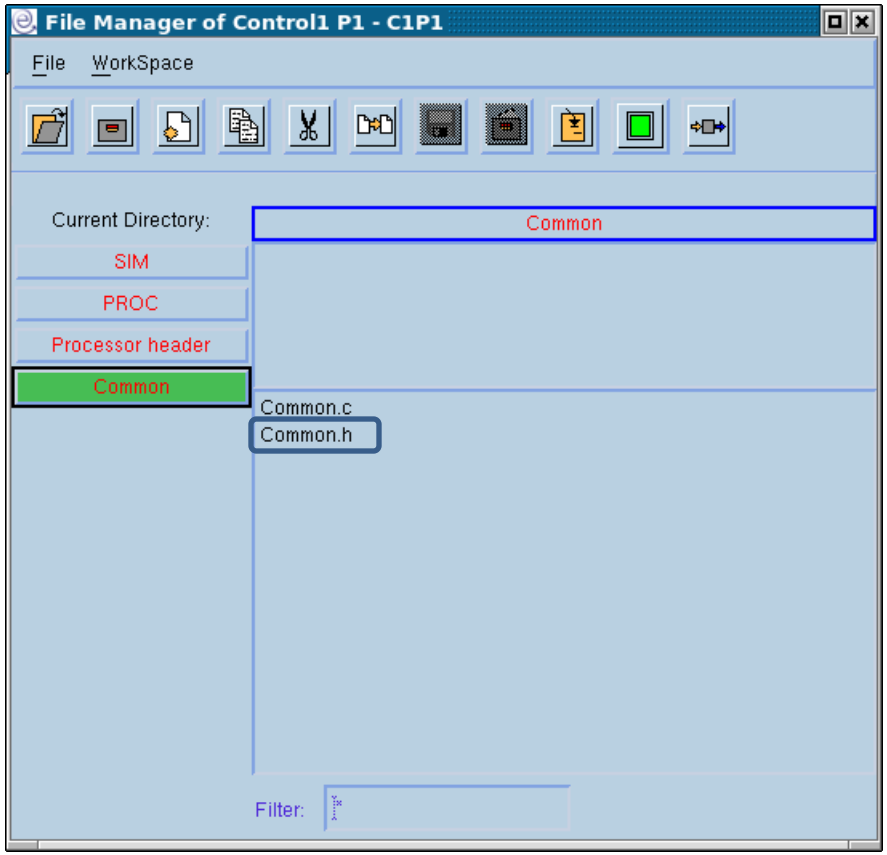
that you can write and customize in order to implement your block.





The Control programmed with ANSI C language

The header files



Header files of the pre-organized structure that are written by the user.

Defines.h	Definition of user constants
Typedef.h	Definition of user structures
DB.h	Definition / declaration of user variables
Prototypes.h	Declaration of the function prototypes
DBP.h	Available for all the Controls belonging to the same Processor and programmed in ANSI C
Common.h	Available for all the blocks programmed in ANSI C

The Control programmed with ANSI C language

Initialization functions

Name	Description	ANSI C File	Data File
C#P$_ReadPar	Parameter file reading	ReadPar.c	Ctr.par
C#P$_I es	User initial ation function	CtrD s.c	--
C#P$_ReadState	Initial state file reading	RWState.c	Ctr.inistate
C#P$_Ini	User initialisation function	Ctr.c	---

The Control programmed with ANSI C language

Execution functions

Name	Description	ANSI C File
Ctr#P$_Exe	Execution of the Control algorithm, updating of the Control states as a function of the current state and of the inputs	Ctr.c
Ctr#P$_Out	Computation of the outputs of the Control as a function of its current state	Ctr.c

The Control programmed with ANSI C language

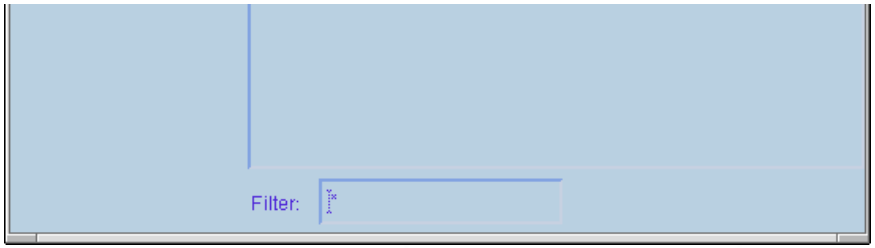
Final functions

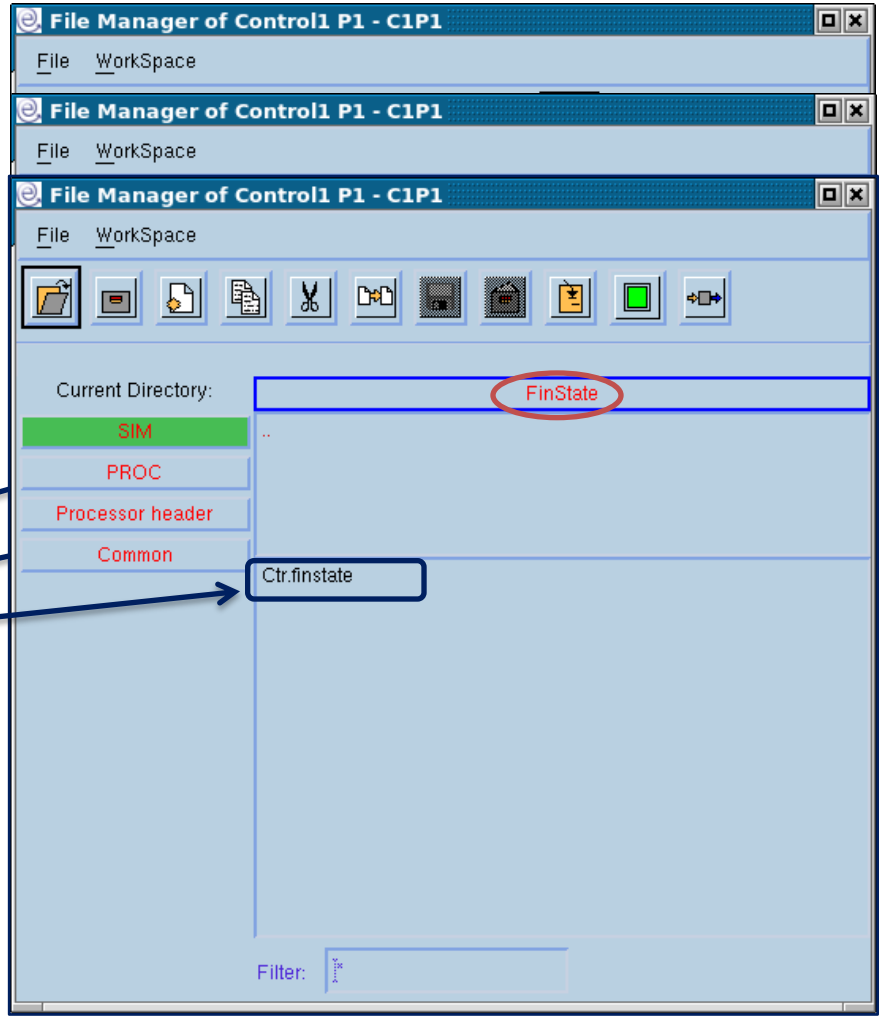
Name	Description	C File	Data File
Ctr#P$_Fin	User final function	Ctr.c	-
Ctr#P$_WriteState	Final state file writing	RWState.c	Ctr.finstate

The Control programmed with ANSI C language

Data file management

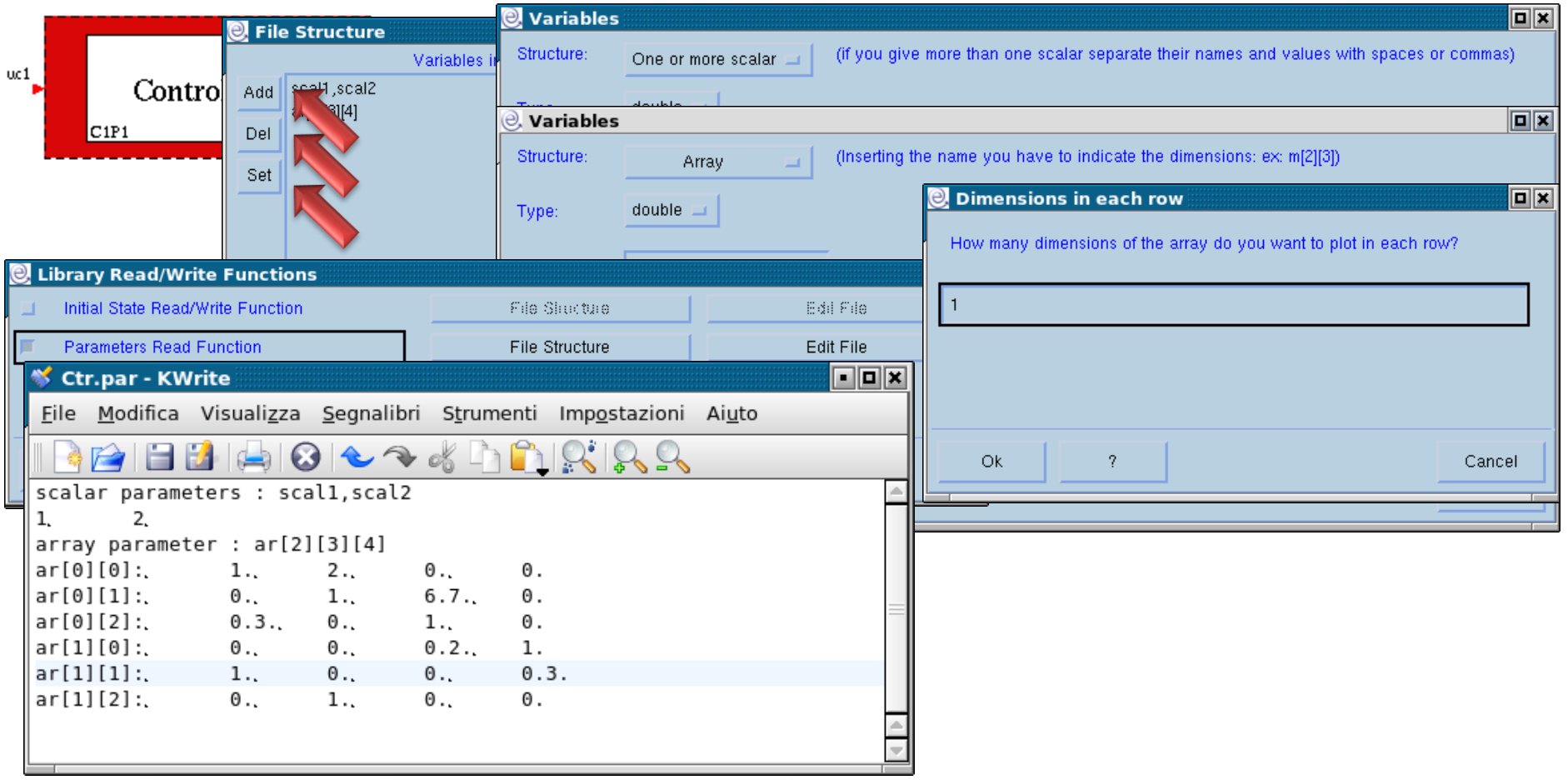
```
/****************************/
void    C1P1_ReadPar(FILE *fp)
/*
INPUTS:
fp      file pointer to the file Ctr.par

OUTPUTS:
value of the parameters of the Control1 P1

OBJECTIVES:
The function can read the parameter set of the control, from the file Ctr.par

All the parameters should be defined in:
        DBInterface.h   interface database of the Control1 P1 Module,
        DB.h            database of the Control1 P1 Module,
        DBP.h           database of the Processor 1,
        Common.h        file shared with the other C block Modules

SCHEDULE:
The function is called once at the beginning of the simulation,
before the functions C1P1_ReadState and C1P1_Ini.
*/
{


return;
}
/****************************/
```





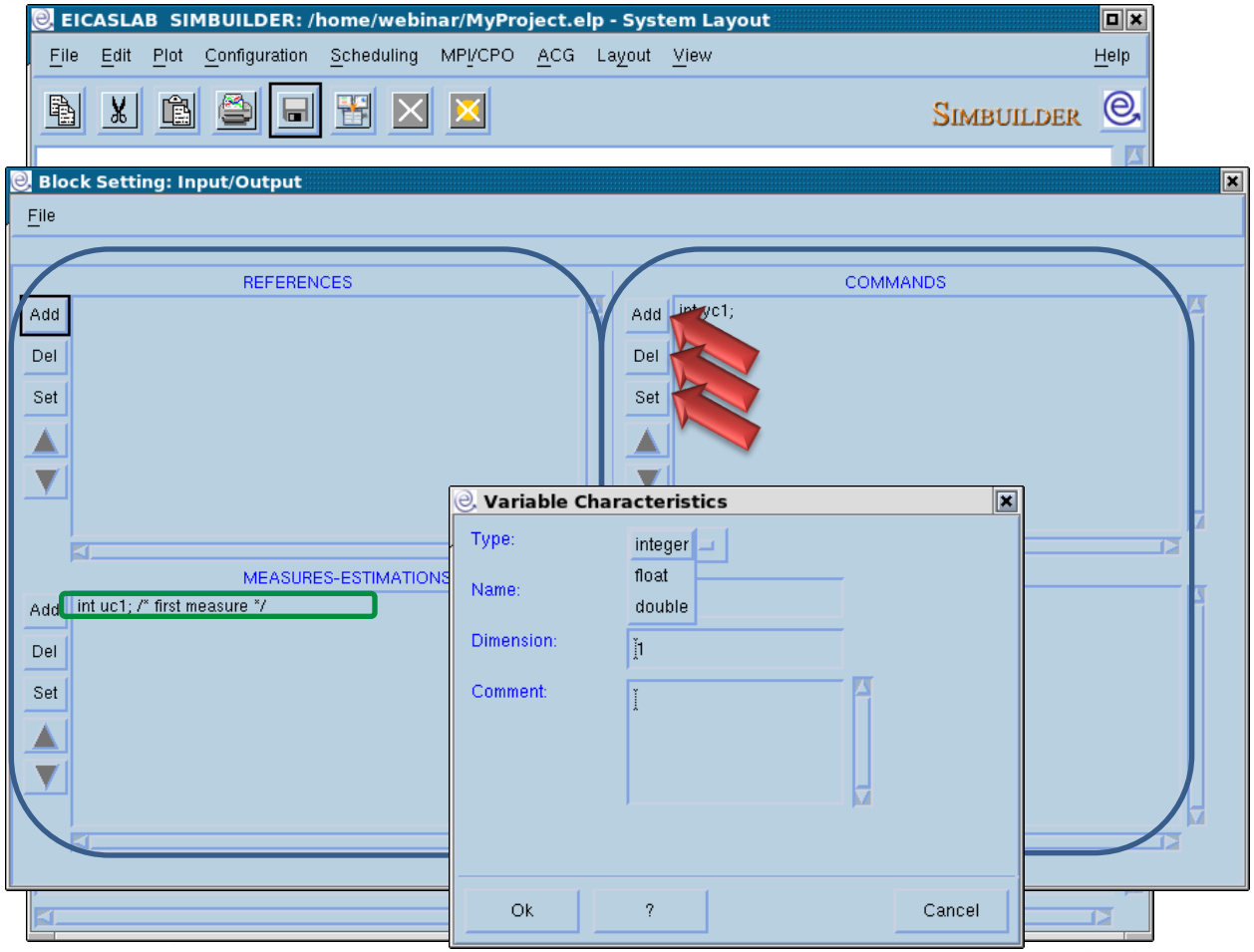
The Control programmed with ANSI C language

The Library Read/Write Functions



The Control programmed with ANSI C language

The Input/Output variables



The input/output variables of the block are defined by means of an appropriate window.

The input/output variables are ANSI C variables that can be used in any ANSI C function of the block.

EicasLab™

The Professional Software Suite
for Automatic Control Design
and Forecasting

for Linux & for Windows

www.eicaslab.com

Welcome to Innovation