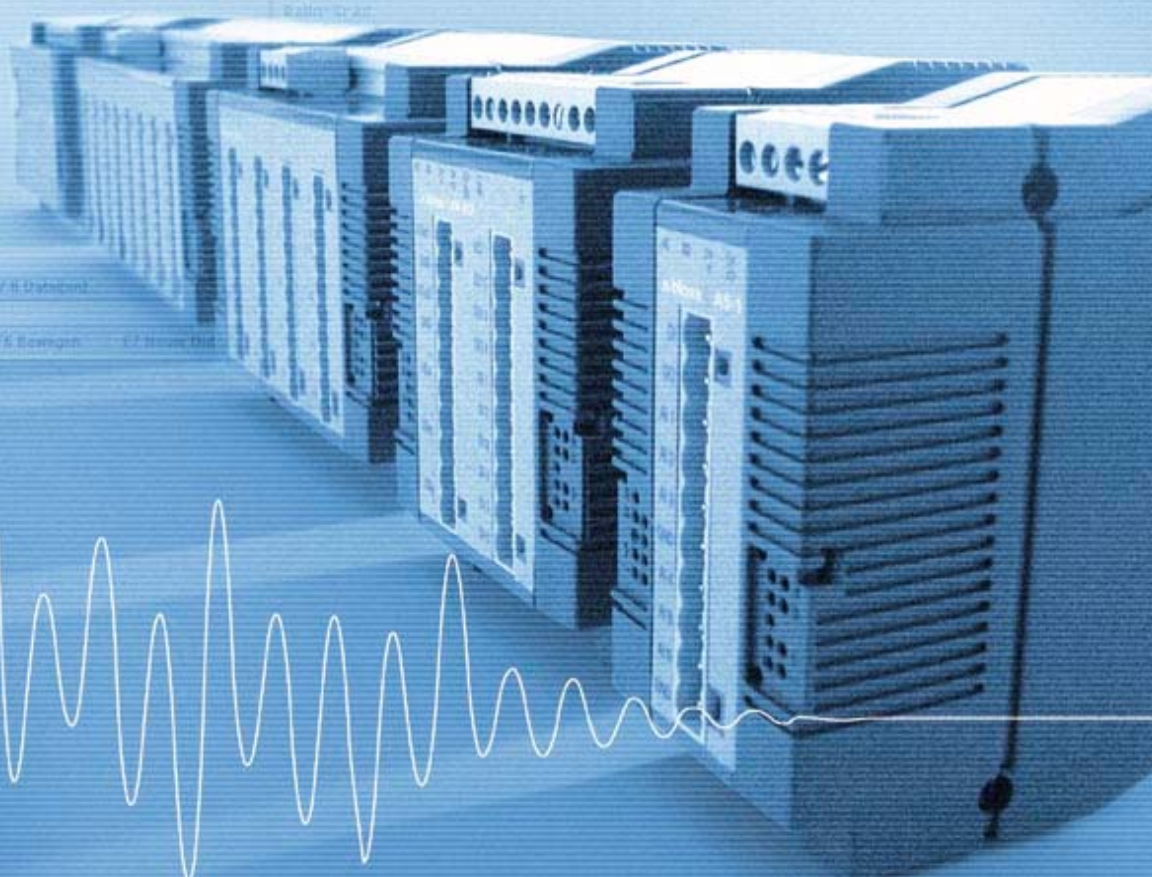


# e.con Instruction Manual





© Copyright 2004 by GANTNER INSTRUMENTS Test & Measurement GMBH, Schruns (Austria).

**Copyrights:** Operating instructions, manuals and software are protected by copyright ©. All rights are reserved. Copying, duplication, translation, installation in any electronic medium or machine-readable form in whole or in part is prohibited. The sole exception is represented by creation of a back-up copy of software for own use as a safeguard, so far as this is technically possible and recommended by us. Any infringement will render the party committing such infringement liable to compensation payment.

**Liability:** Any claims against the manufacturer based on the hardware or software products described in this manual shall depend exclusively on the conditions of the guarantee. Any further-reaching claims are excluded, and in particular the manufacturer accepts no liability for the completeness or accuracy of the contents of this manual. The right is reserved to make alterations, and alterations may be made at any time without prior notice being given.

**Trade marks:** Attention is drawn at this point to markings and registered trade marks used in this manual, in particular to those of Microsoft Corporation, International Business Machines Corporation and Intel Corporation.



**Important:** Before commencing installation, commissioning, putting into service and before any maintenance work is carried out, it is essential that the relevant warning and safety instructions in this manual are read!



## TABLE OF CONTENTS

<b>1.</b>	<b>ABOUT THIS MANUAL</b> .....	<b>6</b>
<b>2.</b>	<b>SOFTWARE DESCRIPTION</b> .....	<b>7</b>
2.1.	Overview .....	7
<b>3.</b>	<b>INSTALLATION OF THE SOFTWARE</b> .....	<b>8</b>
3.1.	System Requirements.....	8
3.2.	Installation and File Structure.....	8
<b>4.</b>	<b>STRUCTURING A PROJECT</b> .....	<b>9</b>
4.1.	General.....	9
4.2.	Project Levels.....	9
4.3.	Classes and Instances.....	10
4.4.	Practical Use of the Structuring.....	11
<b>5.</b>	<b>MENUS</b> .....	<b>14</b>
<b>6.</b>	<b>THE 4 STATES OF E.CON AND THE SUITABLE TOOLBARS</b> .....	<b>15</b>
6.1.	EDIT State.....	15
6.2.	DESIGN State .....	16
6.3.	RUN State .....	17
6.4.	ONLINE OBSERVATION State .....	18
<b>7.</b>	<b>OPERATION FUNCTIONS</b> .....	<b>19</b>
7.1.	General Items.....	19
7.2.	Interacting between e.pac and e.con - summary.sta .....	19
7.3.	Starting a Project.....	20
7.4.	Connecting e.pac and e.con .....	21
7.5.	Save and Close Projects.....	22
<b>8.</b>	<b>WORKSHEET</b> .....	<b>23</b>
8.1.	General.....	23
8.1.	Configuration of the Worksheet.....	23
<b>9.</b>	<b>PROGRAMMING AN APPLICATION</b> .....	<b>25</b>
9.1.	Work with Libraries.....	25
9.2.	Extract of some Structure Function Blocks .....	26
<b>10.</b>	<b>LOADING AND RUNNING A PROJECT</b> .....	<b>32</b>
10.1.	Cycle Time .....	32
10.2.	Download into the Target System e.pac .....	32
10.3.	System Diagnostic.....	34
10.4.	Tips.....	34
<b>11.</b>	<b>SUPPORT</b> .....	<b>35</b>

## 1. ABOUT THIS MANUAL

This manual describes the handling of the e.con Software for programming of an e.bloxx system with an e.pac. This software package is a comfortable tool for graphical programming of such a system. Several features are available to build up simple and even very complex systems.

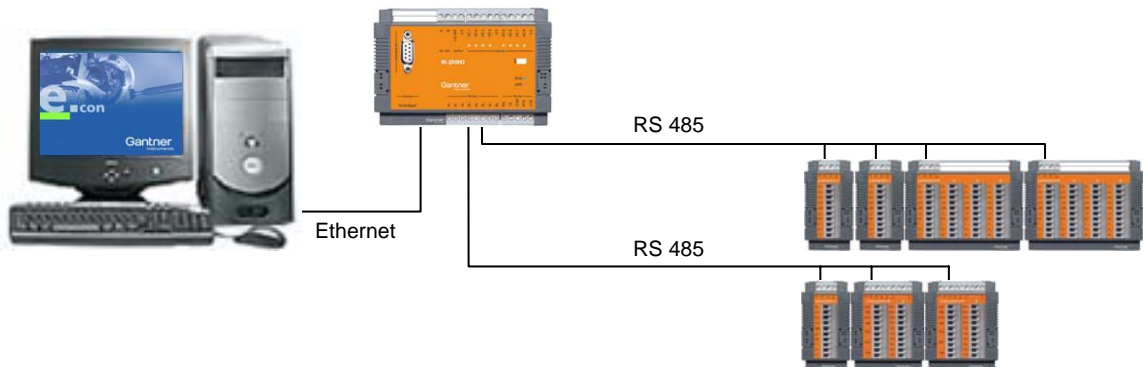
The following information can be found in this manual:

- Description of the e.con
- Installation description
- Description of the different types of functionality

## 2. SOFTWARE DESCRIPTION

### 2.1. Overview

The e.bloxx modules have been developed for the industrial and experimental testing technology, especially for the multi-channel measurement of electrical, thermal or mechanical quantities at test and monitoring application.



**Picture 2.1** – e.bloxx system with several modules and e.pac

For a test application several numbers of analog and digital inputs and outputs are required. For all these signals we provide different kind of measurement and I/O modules which will be connected to an e.pac which is a “programmable automation controller”. In combination with the e.bloxx modules the e.pac permits complex control functionality such as PID controller, state machine, arithmetic, numeric and logical operations, transmission terms, function generators, time functions as well as visualization and additional functionality.

The software e.con is an easy-to-handle graphical programming system to define the e.pac functionality. As only graphical blocks are used, time to get familiarized with the system is minimized. The presentation of graphical blocks follows existing specification and technical languages. Programming can be done on a data flow basis (function block language) or by following the control flow (sequences, flow charts). Structure blocks permit a hierarchic structure of projects. Linked projects and export mechanisms facilitate the reuse of structure blocks that have been created earlier.

e.con combines programming, simulation, testing and running in one tool. Special blocks and additional tools permit online observation of signals and signal tracing as well as run-time measurements.

The user interface of e.con is similar to that of many other advanced development systems.

Commands for working with the system can be selected from the Menu, the Toolbar, the Quickstart Window, and Context Menus as well as by using Keyboard Shortcuts. The structure of the project is presented in the Project Tree. A number of operations can also be carried out from the tree. The working area contains the Worksheet Windows with the blocks and their connections. Instructions, status information and messages are displayed in the Status Line and the Messages Window.

### 3. INSTALLATION OF THE SOFTWARE

#### 3.1. System Requirements

Windows 98 or higher  
 Min. 80 MB of free disc space for both languages English and German  
 Connected Ethernet Interface  
 e.con dongle

#### 3.2. Installation and File Structure

The installation files of the software e.con are available from our Company CD or can be downloaded from our homepage [www.gantner-instruments.com](http://www.gantner-instruments.com)

After selecting the preferred language the installation process will start.

To use the licensed version a dongle is necessary; a license code is not required. The dongle has to be installed separately, please use the software available after installation of e.con. Without this dongle a test version will be available with full functionality for 2 hours for each project. The advantage of the dongle is the possibility to use the software on different PCs with one license only (not in parallel).



Picture 3.1 - Installation of e.con

The following file structure will be set during the installation on the PC:

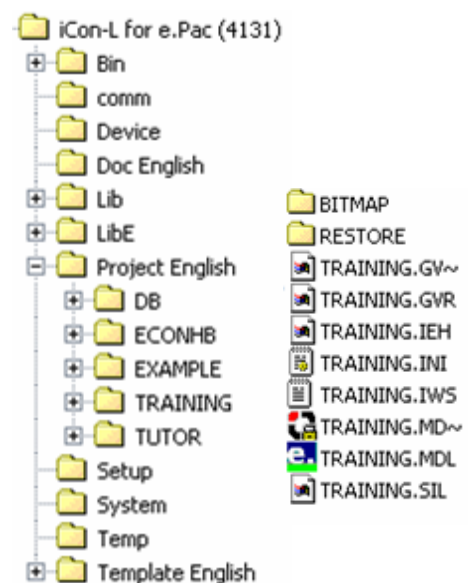
The directory *Project English* contains all the projects that will be defined. After the installation the projects *EXAMPLE* and *TUTOR* are available.

The structure of each project contains the directories *BITMAP* and *RESTORE*.

*BITMAP* contains BMP-files which can be used as graphical elements e.g. for macros or as a background of the working sheet.

In *RESTORE* all stored versions of a project are available in a ZIP-file, which helps in case of unintentionally destroying of a project.

The MDL file is the starting file of the project. By clicking the file the project will start.



Picture 3.2 - File structure of e.con

## 4. STRUCTURING A PROJECT

### 4.1. General

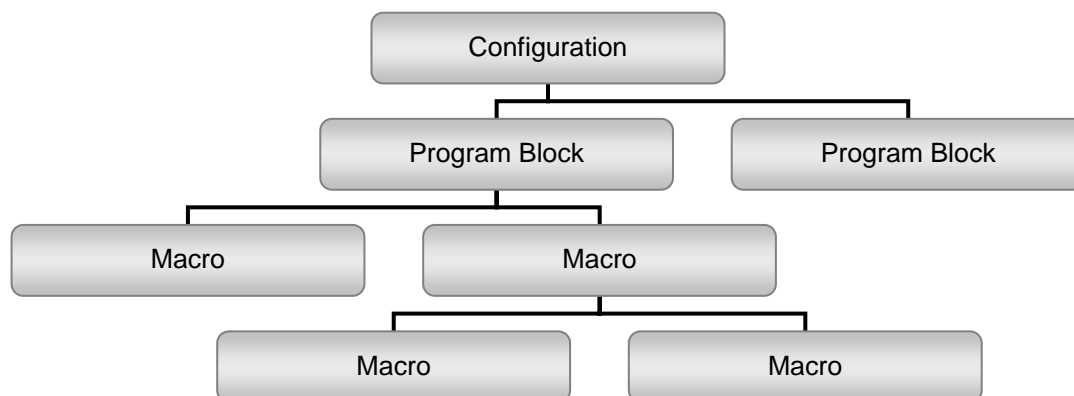
The whole functionality of the function blocks of the system is defined in the e.pac. The software e.con is just a tool for combining and “wiring” the single blocks.

The project is hierarchically structured into several levels by means of structure blocks. The structure blocks can be Programs or Macros.

When a project is processed, there is a difference between classes and instances. Classes are generated during editing operations and contain preset values. Instances are formed when changing to the *Run* mode and can be parameterized there. So it is possible to define a special functionality in the *Edit* mode which can be used in different Programs/Macros with different parameters.

### 4.2. Project Levels

A project can be subdivided into any number of hierarchic structures; however, it always comprises not less than two levels, i.e. its configuration and program blocks.



**Picture 4.1** - Structure of e.con

#### Configuration:

The top level of a project is the configuration (&Main). It allows the graphic configuration of the target system tasks. Program blocks are added to the configuration. By determining their priority and cycle time the task management parameters are defined. Connections between the program blocks facilitate the asynchronous data exchange via the e.pac. The number of program blocks is limited by the number of the parallel tasks in the target system (usually 15). A configuration without any program block would generate a program without any tasks and, hence, it is of no use. Apart from program blocks, also blocks without any target function, such as parameter and visualization blocks, can be included in the configuration level.

#### Program blocks:

Program blocks form the second level of the project where the functions are called by tasks are defined by inserting the relevant blocks and their sequence is established. Programs are structured by using macro blocks. They make it easier to call functions which are used repeatedly. Program blocks are flagged with a \$ symbol preceding its designation.

#### Macro blocks:

Macro blocks can be inserted into program blocks and other macro blocks. However, recursive calling is not permitted. Macro blocks contain sub functions of the programs and relating local parameters. These are no subprograms. During code generation they generate the complete target code for each call.

All project levels may contain visualization blocks (without target function). For exchanging data between project levels, the Input and Output blocks from the Standard Library are used. The Enable block from the same library is used for conditional processing operations of program and macro blocks.

### 4.3. Classes and Instances

Program and macro blocks can be used several times in one project. To parameterize these calls in different ways, e.con uses the concept of instantiation. When changing to the *Run* mode, a new instance of the structure block is formed for each call. Although each instance possesses the same structure (blocks and connections) and can be assigned to an individual set of local parameters. These local instance parameters are defined in the blocks of the structural block and subordinate levels.

During editing operations only preset values are defined in the parameter dialogs of the blocks. During the first change to the *Run* mode they are copied into instance parameters. Thereafter, the instance parameters can be modified in the *Run* mode. However, these modifications have no effect on the preset values in the structure block classes.

When returning to *Edit* and modifying preset values, the instance parameters remain unaffected. The instances of newly inserted structure blocks contain the preset values as instance parameters.

When structure blocks are extended by inserting new blocks, the instance parameters of the existing blocks remain unchanged. The newly inserted blocks are initialized based on the preset values.

**Note:** When changing a parameter in the *Run* mode (instance parameter) it is necessary to download them into the e.pac to ensure that the new parameter is still valid in the e.pac after a power off/on, otherwise the old parameter are loaded after a new start.

To facilitate easier differentiation between structure block instances, they can be assigned to different names. The context menus for structure blocks in the *Run* mode contain the *Instance Name* command. Upon opening the dialog a designation may be entered which afterwards in the symbol of the structure block and in the project tree.

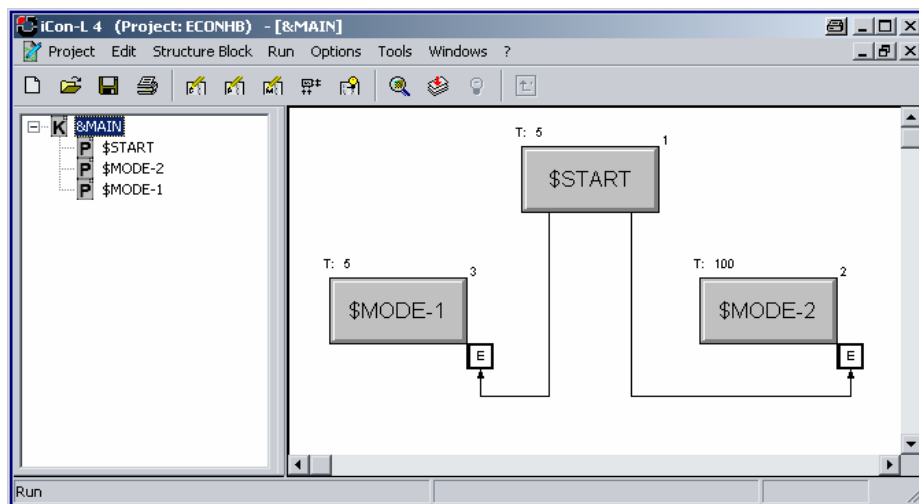
Instance names are only indicated in the *Run* mode. An exception is the configuration where the instance name of the programs remains visible even during editing operations.

#### 4.4. Practical Use of the Structuring

In the practical use the combined measurement, automation and control tasks consist of many sub tasks, which themselves consist of different sequences, flow parameter and evaluations resp. decisions.

In a first step we recommend the definition of the process inputs and the process outputs. This could be done in a macro *Inputs* and in a macro *Outputs*. Due to the use of global parameter these signals are available in any further program and macro block.

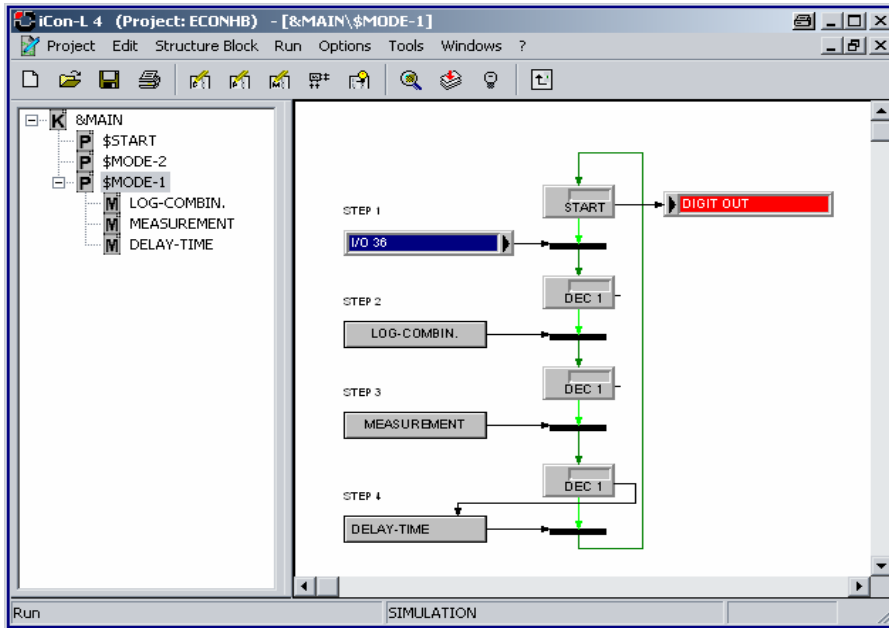
In the next step the total function of the project should be partitioned in logical function such as Test Mode 1 and Test Mode 2. The block *START* will enable/disable the different Test Modes.



**Picture 4.1** - Example: Project with Test Mode 1 and Test Mode 2

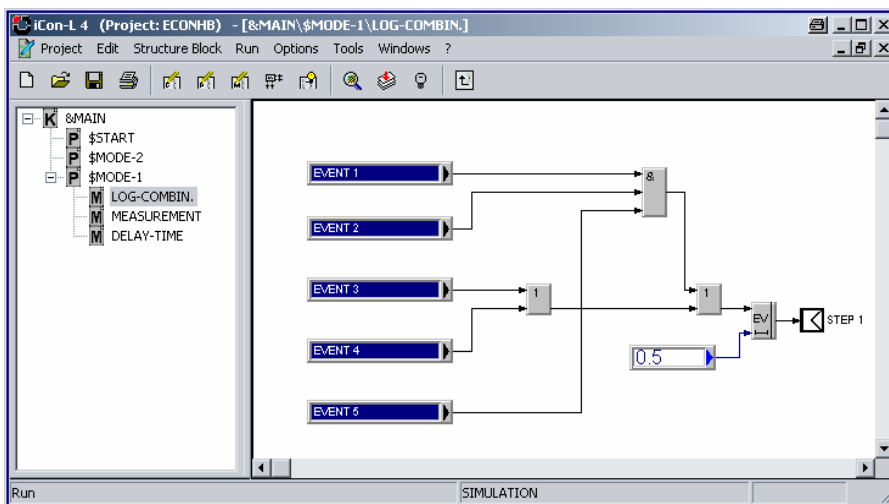
A Test Mode consists of different steps; each step can be controlled by a process I/O (host or process controlled) or by a decision made in the e.pac. The program MODE-1 could contain e.g.:

- Step 1: Proceed at an external I/O signal
- Step 2: Proceed by a combination of events which have happened
- Step 3: Proceed when a measurement signal exceeds a set point limit
- Step 4: Proceed after a special time (delay)



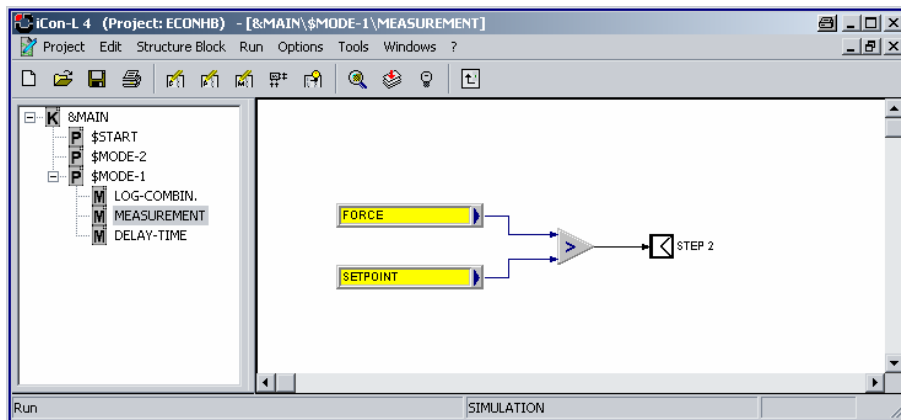
Picture 4.1 - Example: Function of Test Mode 1

In the example the macros LOG-COMBIN, MEASUREMENT and DELAY TIME have the following function:



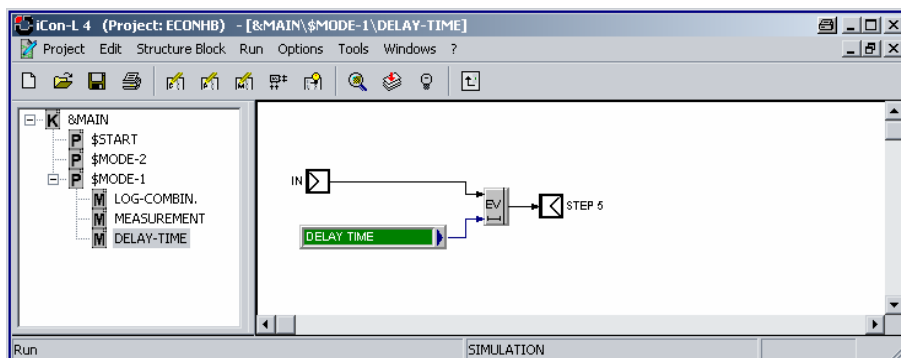
Picture 4.2 - Example: Function of macro LOG-COMBIN

Providing the events according to the logical combinations as a “true” signal at the output STEP 1, the condition is fulfilled and the next condition is required.



Picture 4.2 - Example: Function of macro MAESUREMENT

If *FORCE* exceeds the *SETPOINT* the output of *STEP 2* will be “true” and the next macro is in charge.



Picture 4.2 - Example: Function of macro DELAY-TIME

After a selectable *DELAY-TIME* the input will be switched to the output *STEP 3* and all three conditions are fulfilled.

This example shows how a structure can be used to setup a clear functionality by using different program and macro levels. For a better understanding and documentation text comments can be added to the worksheet.

## 5. MENUS

Menus consist of several submenus. The submenus contain the commands required for creating projects and handling the system. If commands are not available in a current system state, they are deactivated. A detailed description of the sub menus is available in the HELP-function of e.con.

### Submenu Content

Project	Load, save and print projects Work with libraries
Edit	Select the project level for editing Undo last Edit operation Set the Edit mode Work with the clipboard
Structure Block	Define the display of structure blocks Copy, delete and rename structure blocks Define access rights for structure blocks Import and export structure blocks and instance parameters
Run	Switch to <i>Run</i> mode Establish the connection with simulation or e.pac Create the target code and download Switch to Online Observation Work with parameter values of the project
Options	Activate and deactivate elements of the user interface Define the display of the worksheets Set system parameters
Extras	Additional tools
Windows	Activate, arrange and close worksheet windows Export window contents as bitmaps into the clipboard
?	Access online help Display program version and contact address Show the used time running the non-licensed demo version (max. 2 h)

## 6. THE 4 STATES OF E.CON AND THE SUITABLE TOOLBARS

e.con facilitates creating, changing, testing and running projects by using one tool only. The program distinguishes between the following 4 system states:

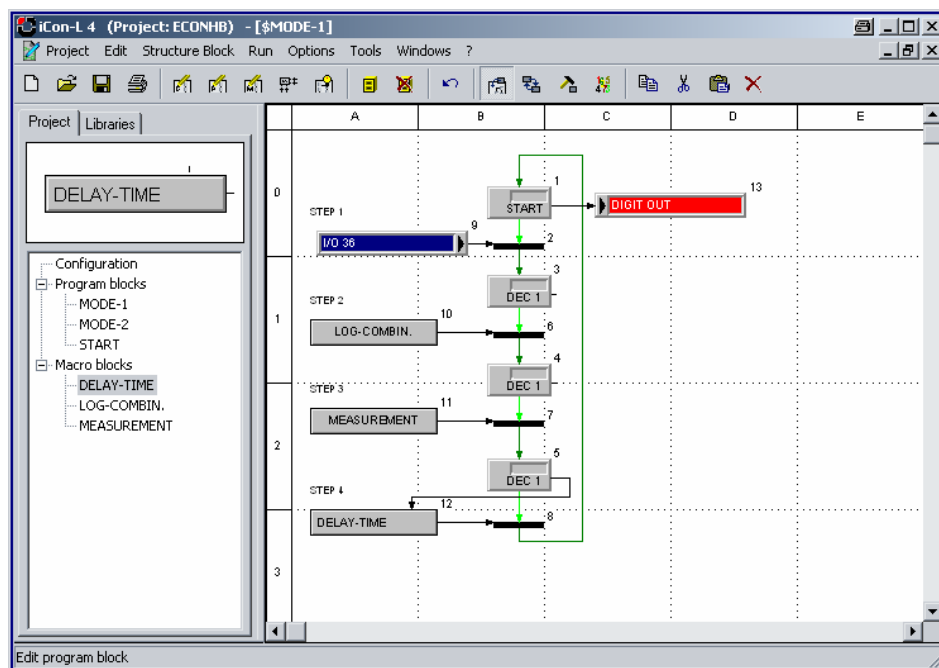
- EDIT
- DESIGN
- RUN
- ONLINE OBSERVATION

Each state can be obtained by using the menu, the toolbar or partly the context menu.

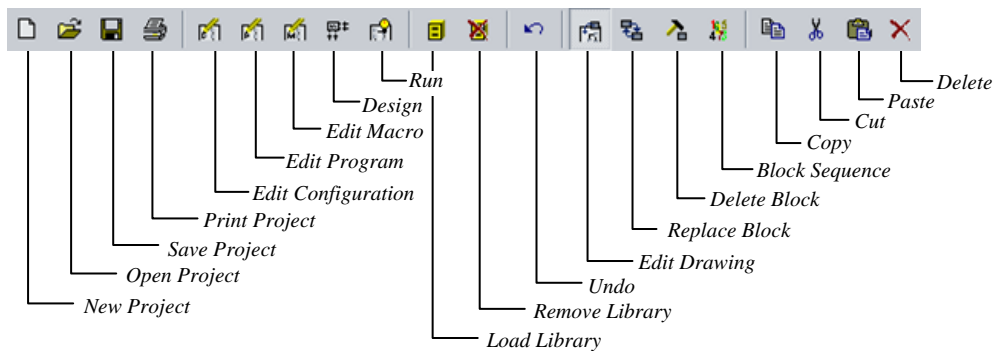
### 6.1. EDIT State

"Edit" is the basic state of the system. It is the state where a project is mainly processed. The structure of a project is determined by loading and removing libraries, defining and deleting structure blocks, and inserting, removing and connecting structure blocks. In the Edit mode the classes of structure blocks are processed. The values entered in the parameter dialogs of the blocks only serve as presetting for the instances visible in the *Run* mode.

Upon entering one of the first three commands of the *Edit* menu or closing all work sheet windows, the system changes to the "Edit" state. This state can be easily identified through the appearance of the project tree. There, the defined structure blocks and loaded libraries are listed in two views. In addition, the grid for block positions and connections are indicated by means of dotted lines in the work sheet windows.



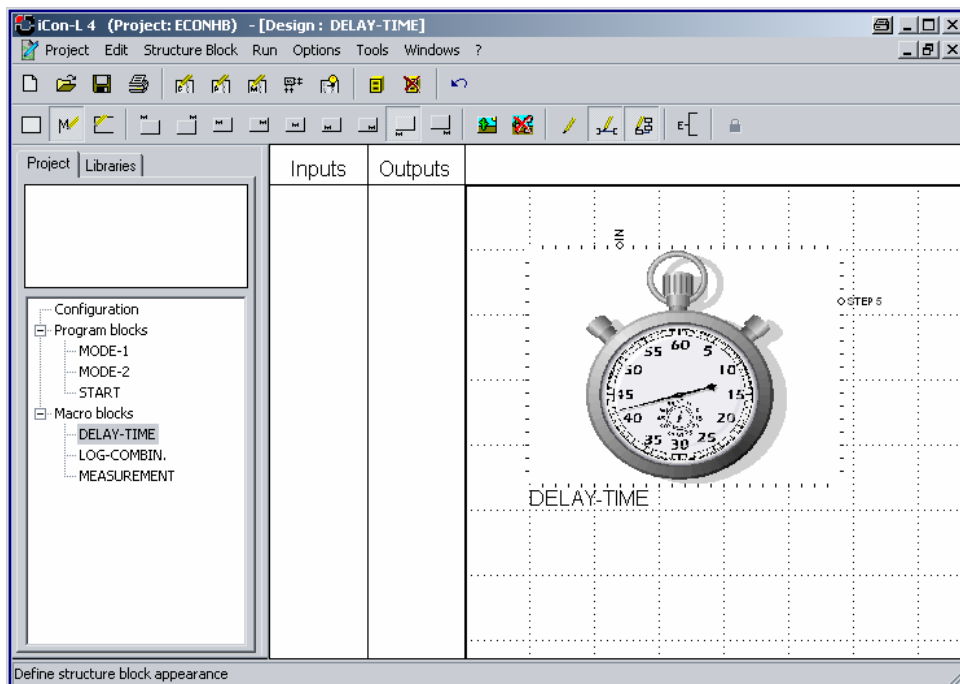
Picture 5.1 - Worksheet EDIT state



Picture 5.2 - Toolbar EDIT state

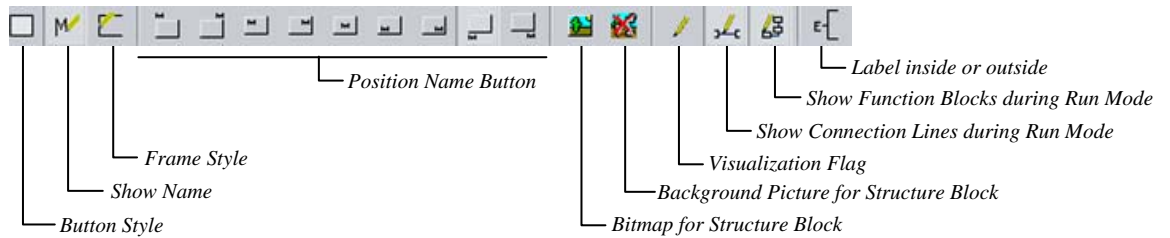
## 6.2. DESIGN State

Apart from the operations described above, i. e. inserting and deleting objects, also the structure block design (Design Structure Block) can be used. From a specific work sheet window and by means of an additional toolbar, label positions, the appearance of symbols and the presentation in the work sheet window can be changed.



Picture 5.3 - Worksheet DESIGN state

Additionally to the already mentioned toolbar there are some more functions available to design the appearance of the project, the programs and the macros.

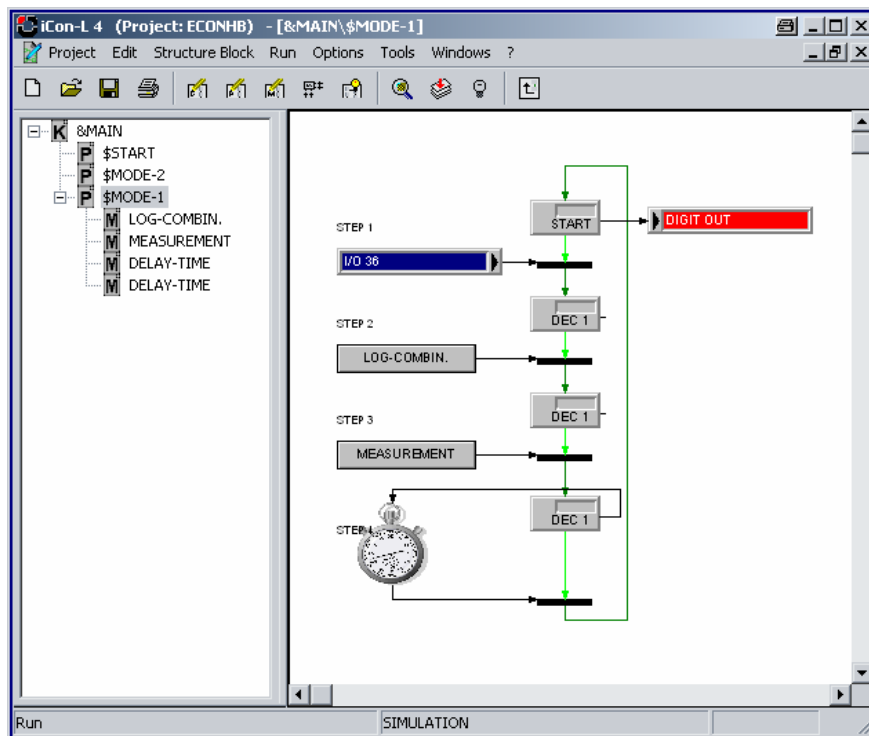


Picture 5.4 - Toolbar DESIGN state

### 6.3. RUN State

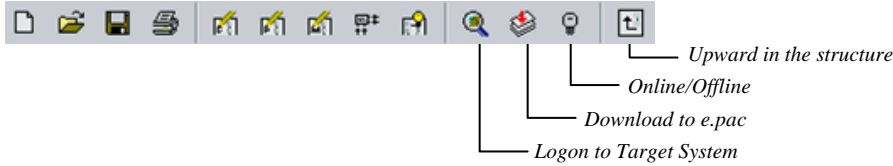
When using the Enter command from the *Run* submenu the program turns from the *Edit* to the *Run* state. During this changeover the program performs an instantiation. Instances are formed from the classes established during editing operations (refer to chapter 4.3 Classes and Instances). When this option is chosen, a new work sheet window is opened, indicating the configuration. In contrast to the work sheet windows in the Edit mode, the work sheet windows in the Run state do not contain a grid. The project tree offers one view only, presenting the hierarchic structure of the project.

The Run mode serves to assign instance parameters used by the blocks later on in the e.pac. Commands for importing and exporting parameter files (refer to the submenus Structure Block and Run) support this work. In addition, cycle times of tasks can be defined. In the Run state structural changes of the project are no longer possible. The other commands in the Run submenu serve to start simulation, establish target system connections as well as to create, load and start the program and change to online observation.



Picture 5.5 - Worksheet RUN state

The toolbar of the Run state is similar to that one of the edit state. The difference is shown in the next picture.



Picture 5.6 - Toolbar RUN state

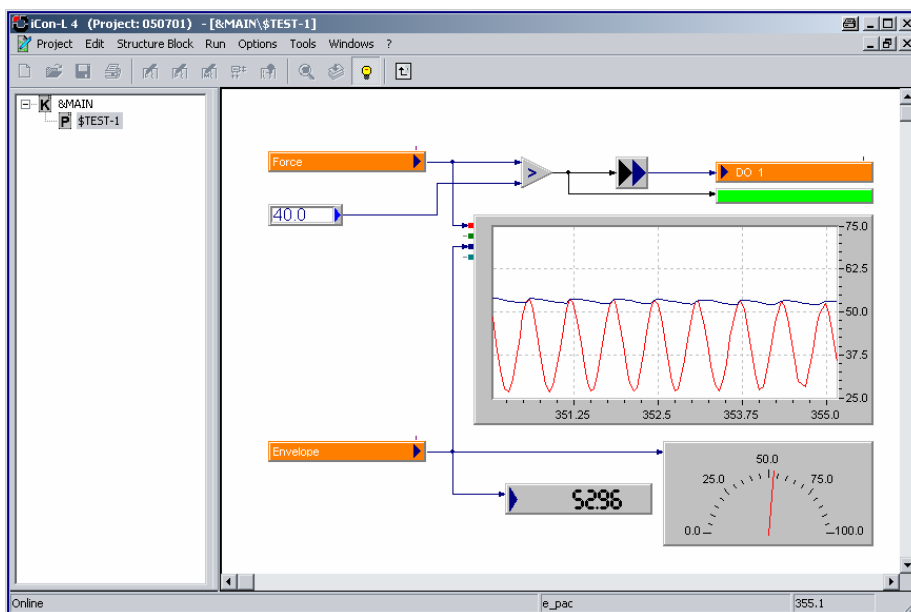
### 6.4. ONLINE OBSERVATION State

When a program has been successfully loaded into simulation or e.pac, online observation starts automatically. For switching from the Run mode without performing a download before directly, the Online command in the Run submenu is being used.

For online observation the visualization blocks indicate the current values of connected signals. Clicking the connection lines by means of the left mouse button opens a window displaying the current signal value. Via parameter blocks and signal editors values are being changed online. However, they only influence the volatile memory (RAM). For transferring the values into the nonvolatile memory of the e.pac, the *Parameter* command in the *Logon Target System* dialog is used.

If the project uses archive blocks, they record signal patterns in the e.pac. Signal patterns can be read out and displayed by means of an additional tool iTrend.

The refresh rate of value representation can be defined in the Work Sheet Settings dialog. The online observation can be stopped by using the Offline command/button or by exiting e.con.



Picture 5.7 - Worksheet ONLINE OBSERVATION state

## 7. OPERATION FUNCTIONS

### 7.1. General Items

Projects in e.con are designed by means of graphical objects. The main tool to work on these objects is the mouse. By pressing its left button main functions are called, depending on the system's state and the object. Other functions are available in the Menu and in the Context Menus of the objects.

This chapter describes main operating functions of the system itself and in combination with the e.pac. Necessary procedures of how to handle these functions are explained to enable the user to get familiarized with the system and to provide an overall survey.

In many cases there are several options to carry out operations, e.g. commands can be called from the relevant menu, the toolbar, context menus or by using keyboard shortcuts. Various context menus may contain identical commands. Moreover, there is no strict procedure as to the calling sequence of commands.

Redundancy and freedom of operation are intentional features of the program. They support the individual operating style of the user and may expedite work. Of course, this chapter cannot give an exhaustive description of all combinations inherent in the program, which usually emerge when users develop their own style of work.

For detailed Information the excessive HELP function of e.con is available

### 7.2. Interacting between e.pac and e.con - #summary.sta

The Programmable Automation Controller e.pac and the graphical configuration tool e.con are two components to create application oriented solution on the base of the modular measurement and I/O system e.bloxx.

The whole functionality of the structure blocks is defined and stored in the e.pac. Each function is available there. The e.con is a tool to program the functionality - the combination, the parameterization and the "wiring" of the function blocks in the e.pac. To do this the e.con is reading all the actual parameter and variables out of the e.pac by an initial introduction.

As soon as the introduction is done the software e.con is able to use the measurement and I/O signals as well as the set point values defined by the e.commander e.g. for writing controller parameter, reading calculated results etc.

After opening a new project the first step should be the "introduction" of the e.pac and to read the #summary.sta file from the e.pac into the e.con. In this file all relevant parameters are included. The #summary.sta file will be stored into the project folder.

If it is required to operate without a connected e.pac a valid #summary.sta file can be imported into e.con.

The #summary.sta file will be generated by the e.pac. After an e.pac system is configured by the e.commander the project #summary.sta file will be available in the e.commander folder too. This file can be used in any e.con project.



Picture 7.1 - summary.sta file

If the configuration had been changed and for e.con relevant parameters are modified, a new #summary.sta file will be created and has to be re-loaded into the e.con. Relevant parameters are

- Name of the variable
- Format of the variable (data type and direction)
- Sequence of the variable within a module

For the e.con software the names of the measured and I/O variables are relevant. Therefore it is mandatory to use unique names for each variable. The configuration software e.commander supports the assigning of the variables.

E.g. the variable 1 of module 3 is named as *Temperature 5*, e.con is reading the name *Temperature 5* only. In case the name has to be modified to *Temperature 05* a new #summary.sta file will be created and has to be loaded into the e.con. The symbol for *Temperature 5* is shown in red color, has to be renamed to *Temperature 05* and the project has to be loaded into the e.pac once again.

### 7.3. Starting a Project

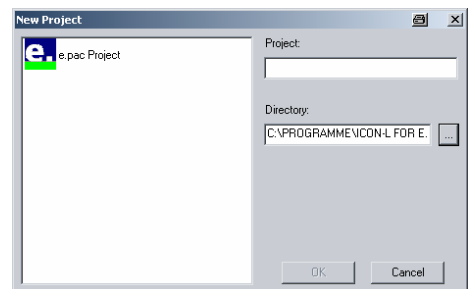
After the program e.con had been started it is possible

- to start a new project,
- to open an existing project,
- to upload the project from a connected e.pac or
- to load a project from the restore directory.

Start a new project:

Click on the button *Create a New Project* or select the menu *File - New Project* and the following window will appear:

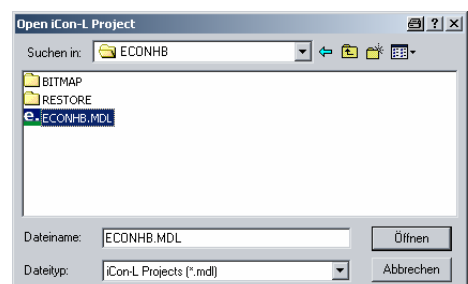
Enter a project name and confirm/modify the directory where the project should be stored.



Picture 7.2 - Window New Project

Open an existing project:

Select the project and open the xxx.MDL file

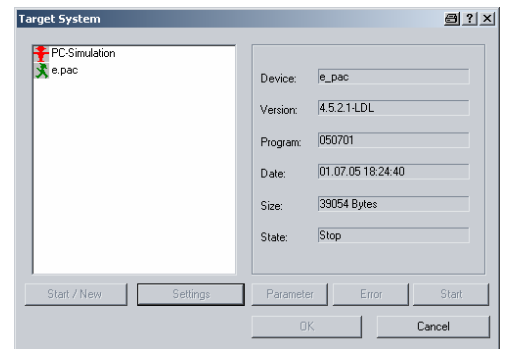


Picture 7.3 - Window Open Project

Upload a project from e.pac:

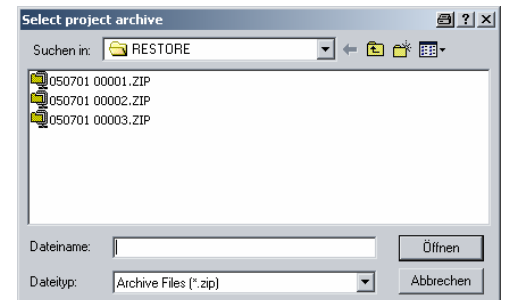
The window for the selection of the so called *target system* will open. There all selected e.pacs and the PC-simulation are offered. By clicking the relevant e.pac the color will change from red to green.

In case the icon color is yellow the settings (IP address) have to be checked.



Picture 7.4 - Window Upload from e.pac

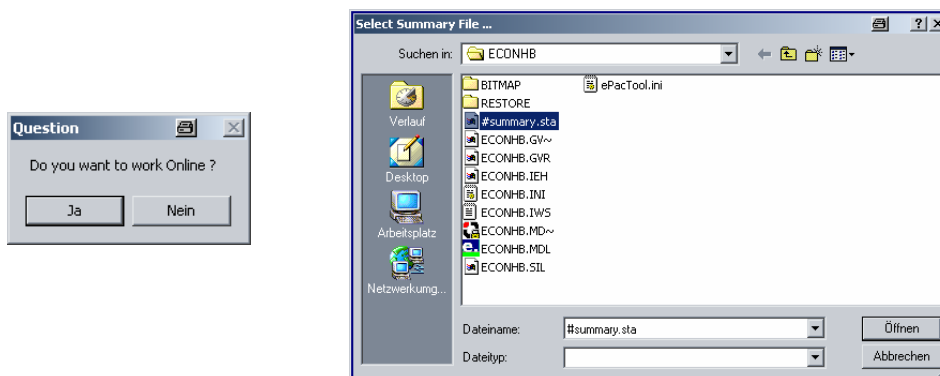
Load a project from restore directory (zipped file):  
Selection of the file to be opened.



Picture 7.5 - Window Load from Restore file

## 7.4. Connecting e.pac and e.con

Is a new or an already existing project open the e.con has to be selected first by using the menu *Tool - Select Device*: Now it is possible to select *Online* or *Offline*. In case *Offline* is selected a browser will open to select a #summary.sta file.



Picture 7.6 - Selection Online/Offline and Browser to choose a #summary.sta file

Selecting *Online* a network scan will be done. After selecting the e.pac the successful selection has to be confirmed.



*Picture 7.7 - Successful selection of an e.pac*

Now the software e.con will work with all the variables defined in the measuring and I/O system.

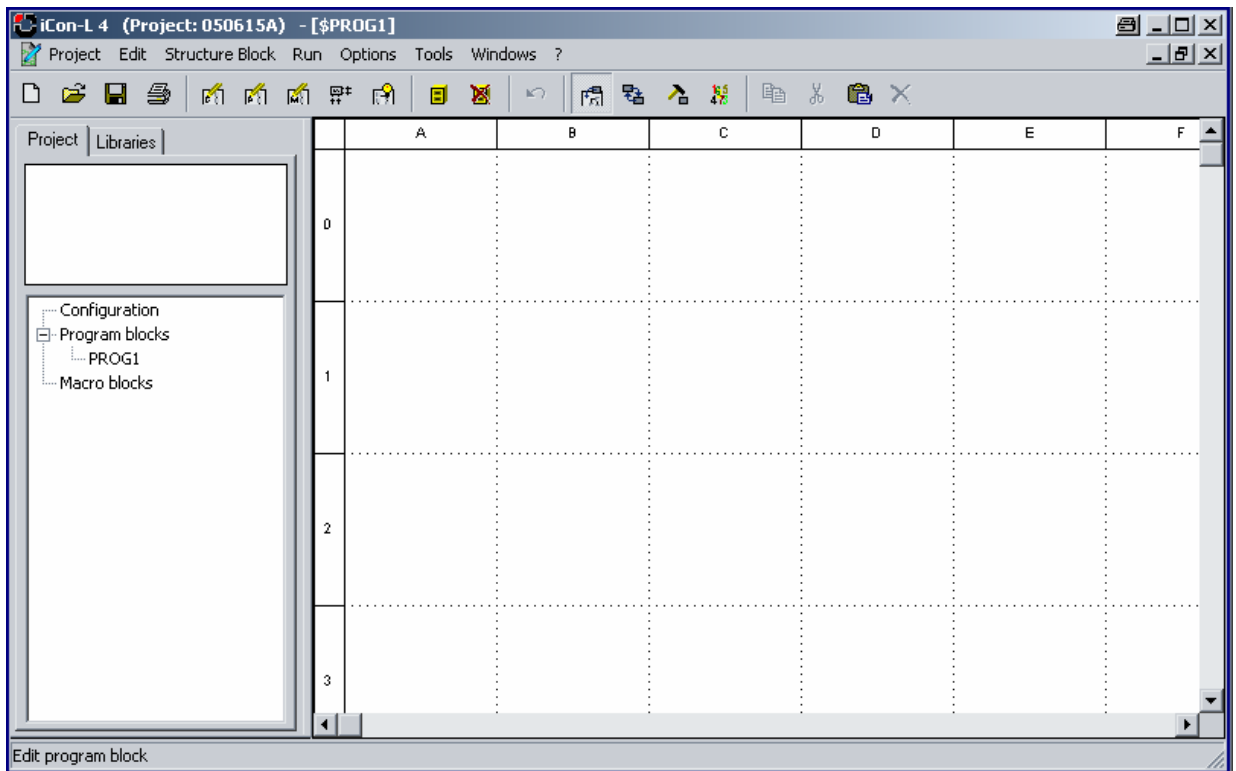
## 7.5. Save and Close Projects

These functions work in the typical and well known way.

## 8. WORKSHEET

### 8.1. General

After a new project is started a worksheet is available in the edit state for the programming of the required functionality.



**Picture 8.1** - Empty worksheet after starting a new project

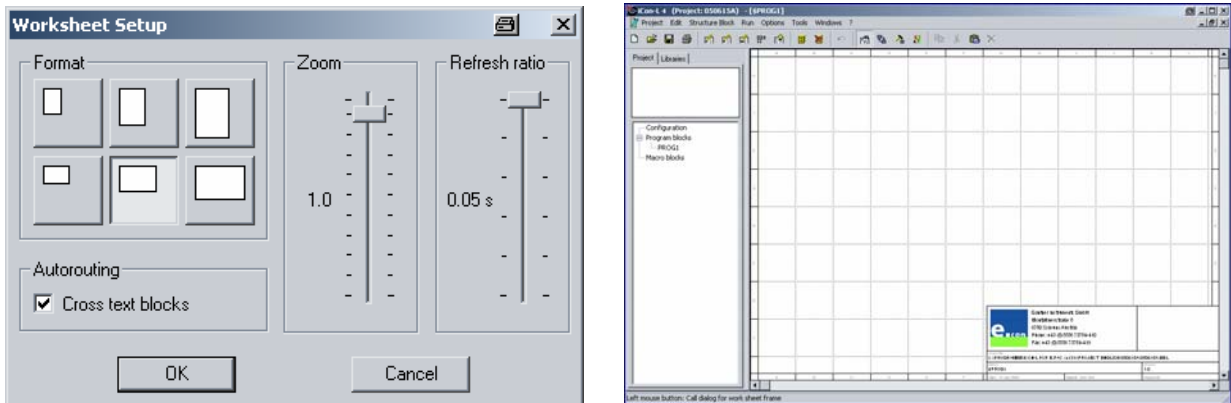
This new project exists of the following parts

- Menu and icon bar
- Worksheet
- Project tree/Libraries

Using the menu *Options* it is possible to select single parts of the window such as Status Line on/off, Project Tree on/off etc.

### 8.1. Configuration of the Worksheet

The menu Option - Worksheet permits an adaptation of the view, the zoom and the refresh rate for online visualization of the worksheet to the application.



Picture 8.2 - Configuration of the worksheet setup and the resulting effect

The label of the worksheet can be modified by clicking into the label. Logo, company, address and more can be edited.



Picture 8.3 - Editing of the worksheet label

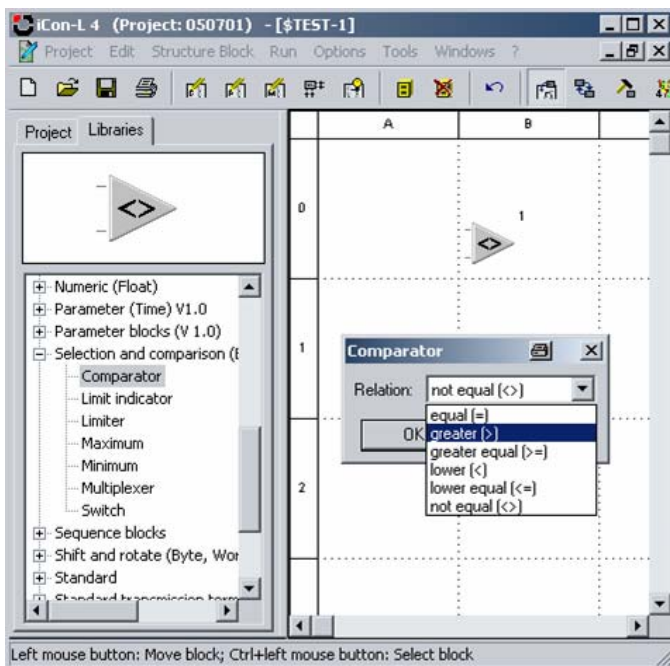
## 9. PROGRAMMING AN APPLICATION

### 9.1. Work with Libraries

For the programming of the application a lot of libraries are available. In the section Project Tree the libraries can be selected. They are available in an alphabetic order. In e.con the parameter and operation can be chosen in different formats. Each format is shown with its own color. The formats are:

Format	Color
Bit/Byte	black
Word	violet
Long	turquoise
Float	dark blue
Text	light blue
Time	green

To use a **Structure Function Block (SFB)** the library has to be selected and the target SFB has to be marked. The icon of the SFB appears in the upper window and by drag and drop it can be placed into the worksheet.



For SFB with a possibility of parameterization an additional window will open.

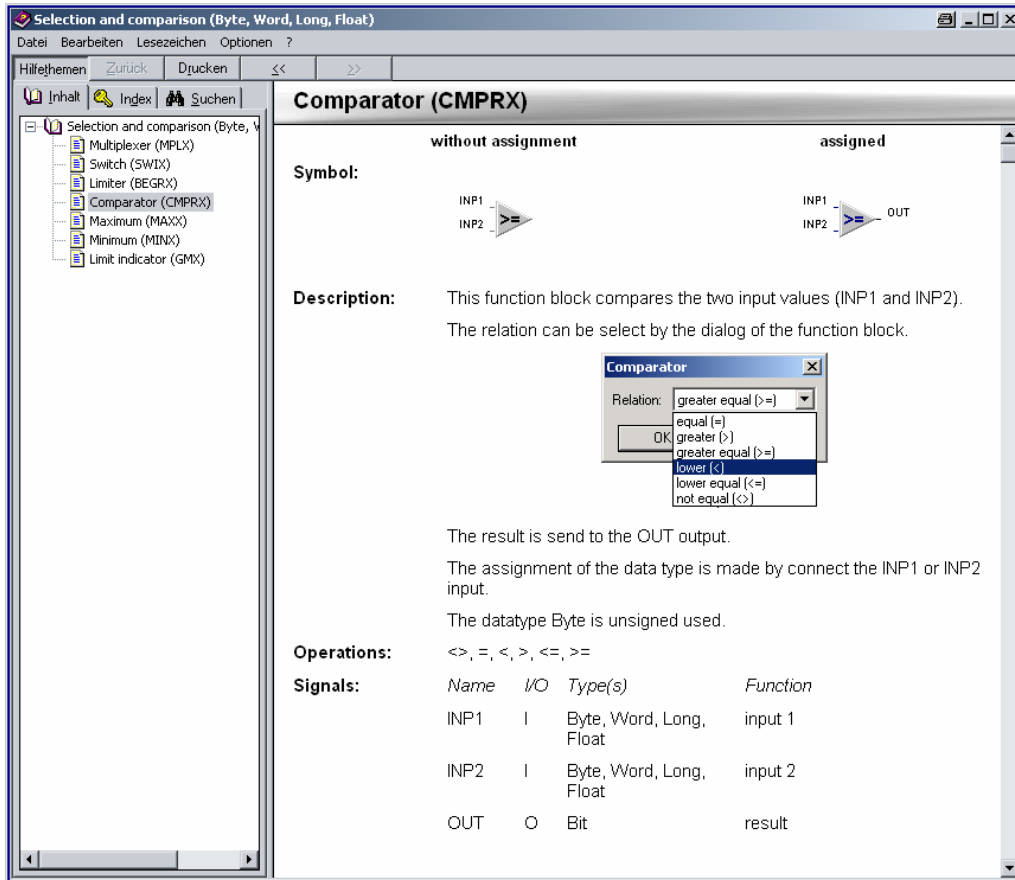
As soon as the parameters are selected, the SFB will be shown in the worksheet with a sequence number. With the menu *Options - Connector Labels* the connectors can be selected "visible".

The context menu offers the function:



Picture 9.1 - Selection and parameterization of a SFB

For all libraries a very detailed help support is available. Clicking the right mouse button onto the library or the function block provides the help information.



Picture 9.2 - Help information for the function block "Comparator"

It is possible to add or delete libraries to adapt the project to the application. The menu *File* offers the functions *Load Libraries*, *Remove Libraries* and *Replace Libraries*.

This functionality is strongly recommended to be used by advanced users only.

## 9.2. Extract of some Structure Function Blocks

### 9.2.1 SFB Archive

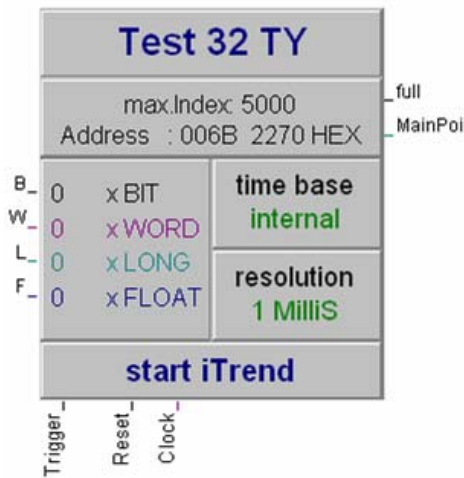
The SFB *Archive* - data storage - can be used to store measured data, I/Os and parameter. The conditions (Trigger, Reset, channels...) can be set with the context menu *Parameter Dialog...* There are 5 archives available with 1 MByte each, these are e.g. 200.000 measurements in float format inclusive time stamp per archive.

The data of SFB archive are stored in binary format into the e.pac and can be read out e.g. via Ethernet. The file names in the e.pac are

~archiev0.dat ... ~archive5.dat

The file ~arcind.dat indicates the name of the archive:





The archive icon shows some information on the parameter:

- Name of the archive (e.g. Test 32 TY),
- Number of stored data sets (e.g. 5000)
- Time Base (e.g. internal)
- Time resolution (e.g. 1 ms)
- Number and kind of connected variables to be stored

iTrend is a tool to show the contents of the archive

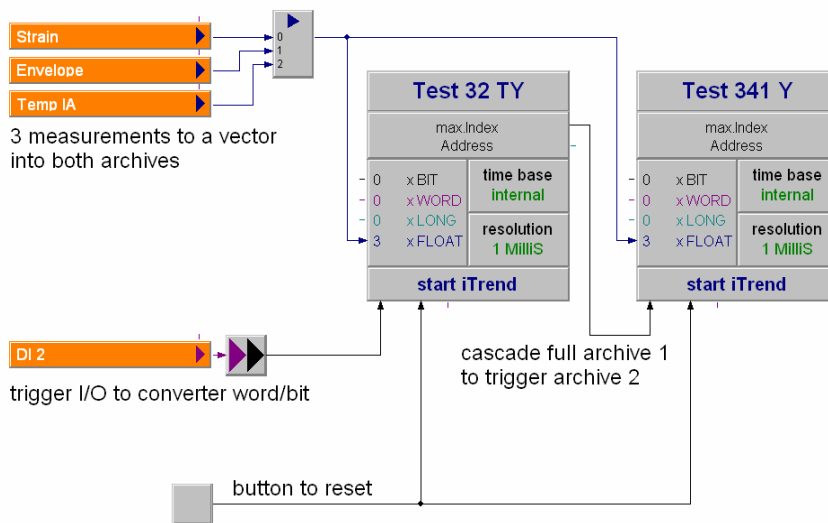
The signals to be stored can be connected to terminal B, W, L and F. Further connectors are Trigger, Reset, Clock, full and Main Pointer.

The connector full can be used to cascade archives (full to Trigger).

Picture 9.3 - icon of the SFB Archive

The parameter of the archive can be set only in edit or offline state. The iTrend is available in the online state only.

The following example shows a 2 MByte (2 x 1 MByte archive in cascade) data storage, triggered by a digital I/O and reset by a button:

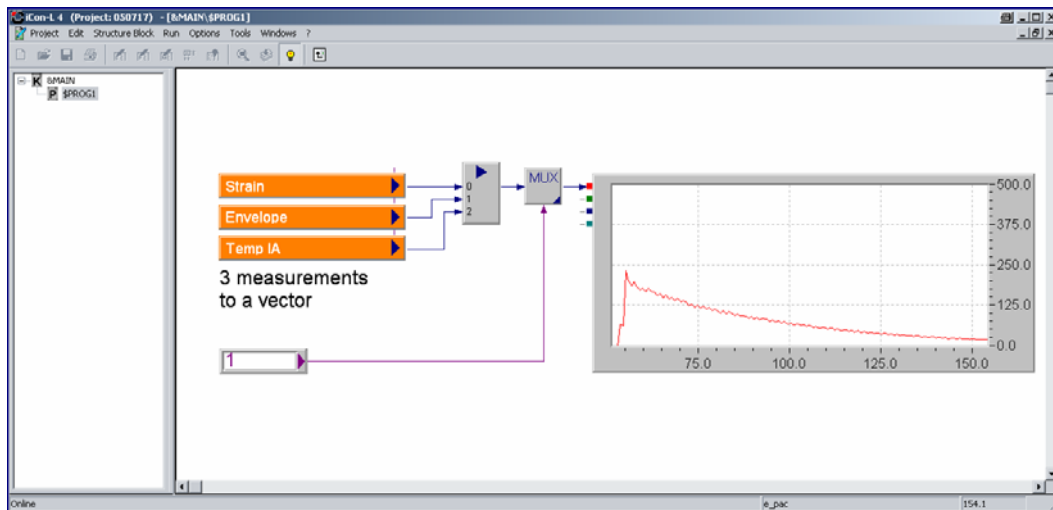


Picture 9.4 - Example of the use of the SFB archive

### 9.2.2 SFB Convert - Split - Vector

As mentioned in chapter 9.1 the variables can have different formats. The SFB Convert converts the format to allow different formats in an operation. In the example in picture 9.4 a digital input with the format WORD is used to trigger an input requiring a BIT format.

The vector and splitter have the task to put discrete values together into one string and vice versa, e.g. by selecting one of 3 signals by using a selector 0 - 1 - 2.

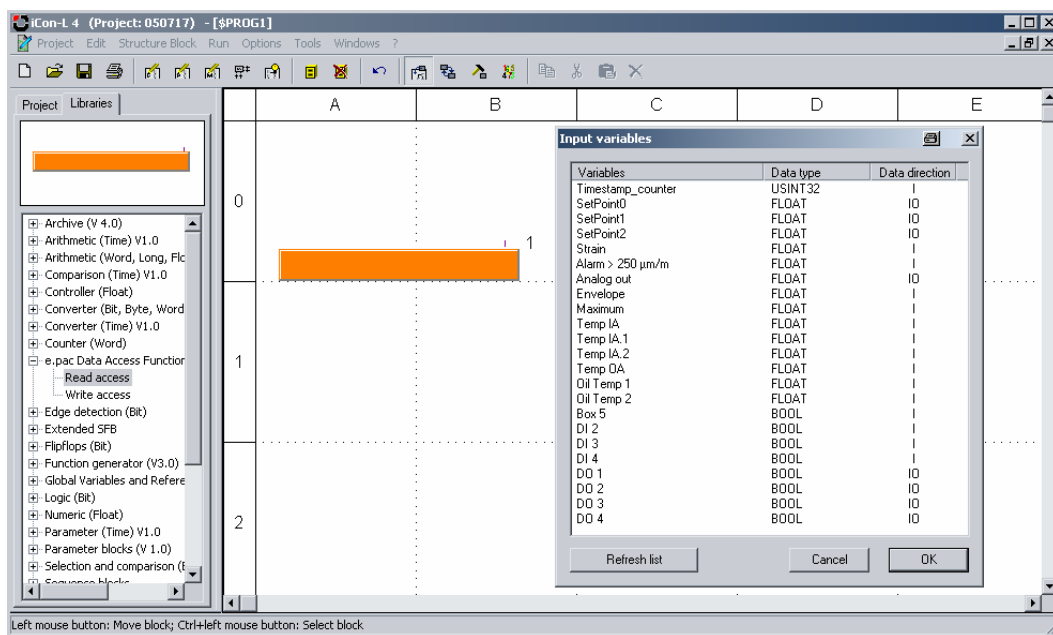


Picture 9.5 - Example of the use of the vector: The signal Envelope (sel 1) should be selected by the MUX.

### 9.2.3 SFB e.pac Data Access Function

With this function it is possible to read (Read Access) or to write (Write Access) measurements, digital I/Os, analog outputs, set points time stamps etc. All the variables defined in the e.pac are available in these SFBs.

Select *Read Access* or *Write Access* an e.bloxx colored bar will appear in the drag and drop field. Place the bar into the worksheet, the selection window *Input variables* will appear to select the relevant variable.



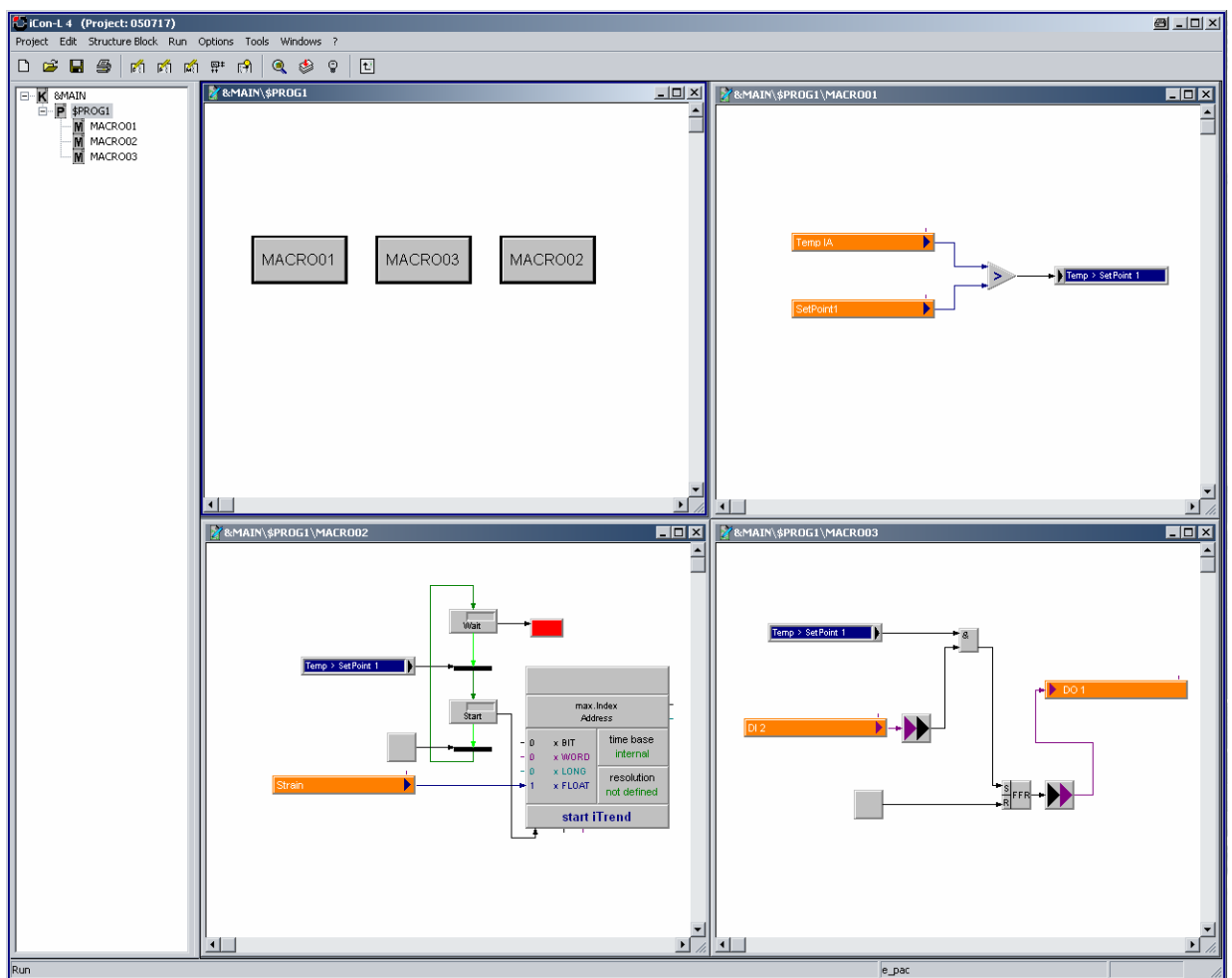
Picture 9.6 - Selection of an e.pac data access (read or write)

The information on the available variables is based of the *#summary.sta* file. For that reason it is necessary to re-download the application in case of modifying relevant parameters (see 7.2).

The e.pac provides 100 read/write and 100 read set points, e.g. to parameterize PID controller or to read calculations. How to set such a set point is described in the e.commander manual. The defined set points are available in the list Input variables / Output variable to be used in the e.con application.

### 9.2.4 SFB Global Variables and references / Read - Write

The function of the *Global Variables* is to keep a clear overview of the worksheet, especially when using some program blocks and some macros. If in macro01 a supervising of a measured signal is done and the result should be available in macro02 and macro03, than it is recommended to use global variables. A small example is given in the next picture. In macro01 global variable will be active when  $Temp > SetPoint1$ . This variable will be used in macro02 to trigger a data store and in macro03 to set a digital output when DI 2 is active. A flip-flop will keep the state until a button is pushed.



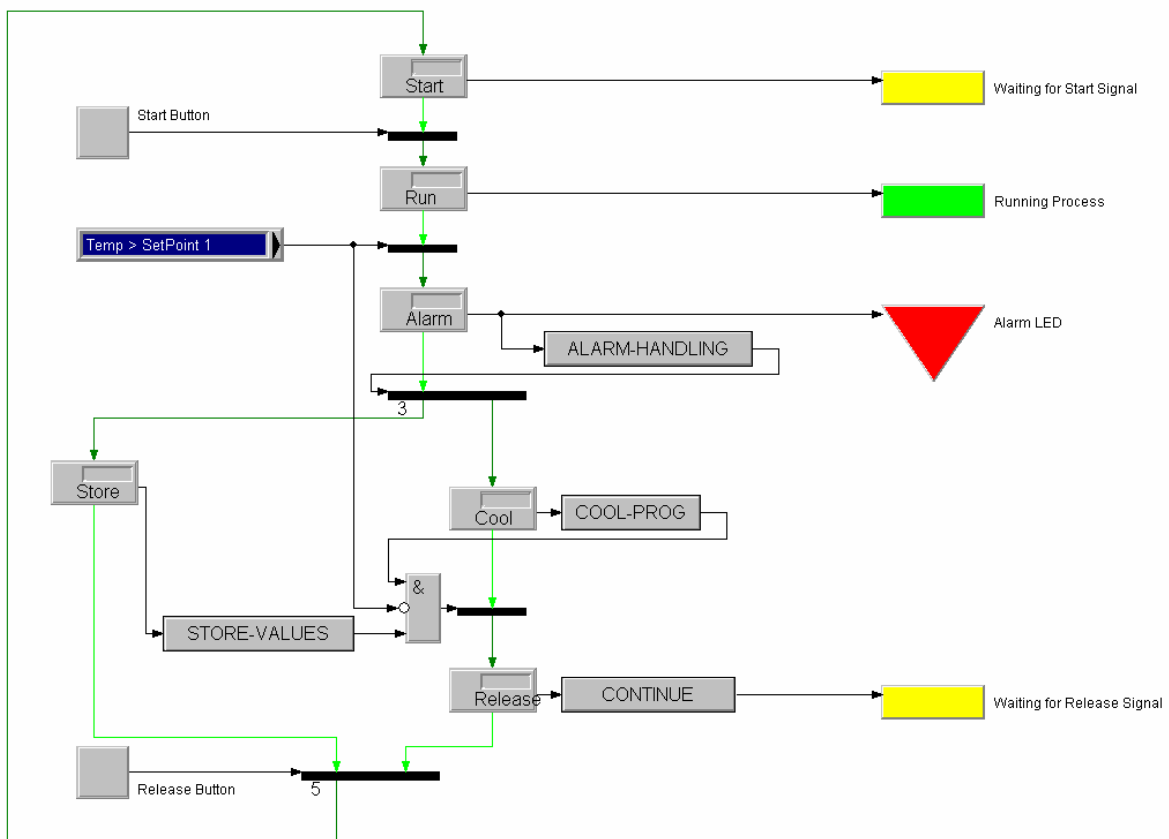
Picture 9.7 - Example of the use of the global variables

### 9.2.5 SFB Sequence blocks

Explaining e.con up to now was more or less signal flow based. A further possibility is offering the control flow based operation by sequences and flow charts. These functions are helpful to control a process by different conditions such as events, I/Os, calculated results, Boolean combinations or times. The set up of such a sequence can be done by the basic SFB *Step* and *Transition*.

The *Step* is like a register with an output and a *Transition* is like a switch. Further SFB in this library are joining and splitting transitions as well as preset.

A simple example is shown in the next picture:



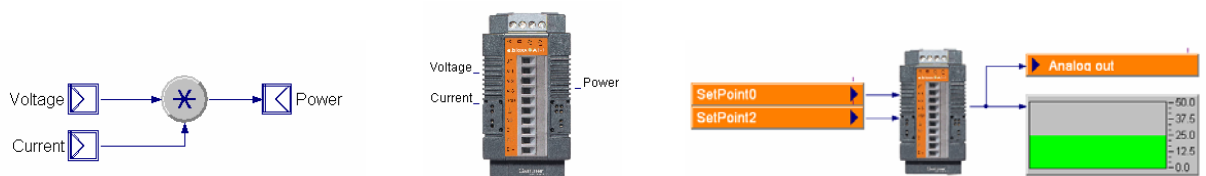
Picture 9.8 - Example of the use of sequence block.

The splitting and joining functions are used in case of an event two separate processes are running (example COOL-PROG and CONTINUE parallel to STORE VALUES). The joining SFB will synchronize both branches.

### 9.2.6 SFB Standard

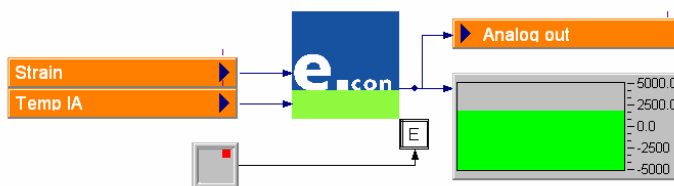
This library provides the possibility to set inputs and outputs for a macro or a program, functions can be enabled/disabled and text comments can be set.

A macro is a separated block where frequently used functionalities are processed. If there are one or more in- or outputs are required, the SFB *Input* or *Output* has to be used. For example a macro should calculate the electrical power by voltage and current. Therefore two inputs and one output are required. The design (Design State) of the macro shows an e.bloxx picture which can be scaled freely and the location of the in- and outputs can be defined as well. In the next step the macro can be used to connect the measurements and e.g. an analog output and a bar graph indicator to the macro output.



Picture 9.9 - Definition of the in- and outputs of a macro

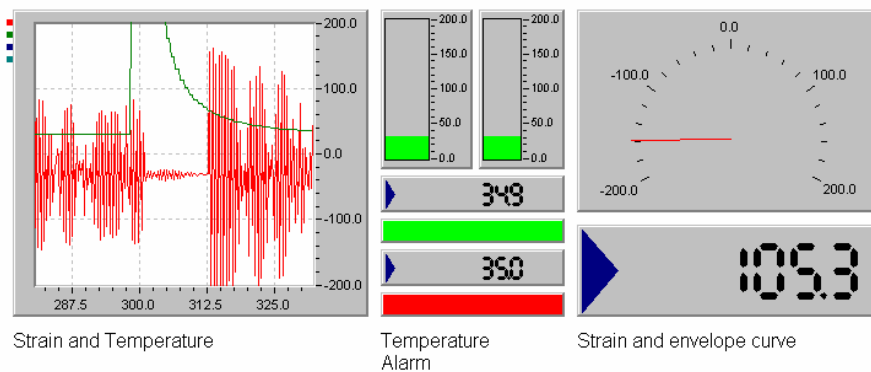
The enable SFB can be used to enable or disable macros or programs. The enable signals require bit format 1/0.



Picture 9.10 - Enabling a macro by a digital signal

### 9.2.7 SFB Visualization Blocks

e.con is not just a programming tool, it is also a visualization tool. For that reasons there are some SFB for the visualization of states, measurements and I/Os in an analog as well as a digital way. In the *State Design* it is possible to hide all the blocks and lines just to show the visualization elements.

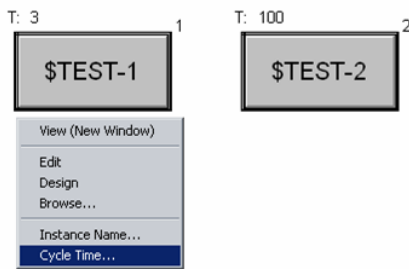


Picture 9.11 - Visualization with e.con

## 10. LOADING AND RUNNING A PROJECT

### 10.1. Cycle Time

With e.con it is possible to run different tasks with a different cycle time. So a not time relevant temperature controlling can be done with a cycle time of 100 ms and a measurement evaluation can run with e cycle time of 3 ms. Each single program can be set to an individual cycle time by clicking the right mouse button onto a program block the run state and afterwards a context menu will offer the *Cycle Time...* set up.



The macros and SFB within a program are running according to sequences with a fixed cycle time. The order of the sequences can be set with the context menu *Block Number...* or with the command *Block Sequence*.



Picture 10.1 - Cycle time of a program block

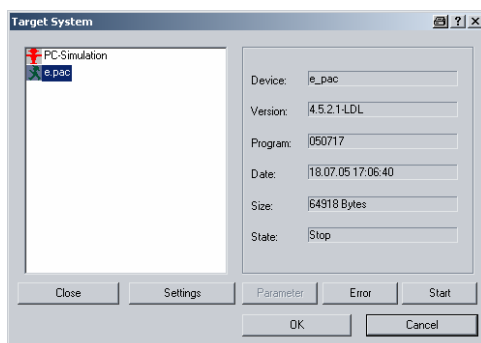
To check whether the target system can run the selected cycle time the function *System Diagnostic* can be used.

**Notice:** The minimum cycle time of an e.pac is 3 ms.

### 10.2. Download into the Target System e.pac

The offline developed configuration has to load into the target system e.pac where it is running autonomously. The e.pac should be selected already by the menu *Select Device* to work with the *summary.sta* file. Following the steps to download an e.con application into the e.pac:

- After finishing the programming in the *Edit State* push the *Run* button  or select the menu *Run*. The dotted lines will disappear and the program block will be shown. The window *Messages* will report the process. In the sense of classes and instances (see 4.3.) the system is changing into a new instance. This means, the parameters now can be set for this application.
- Click to the button *Logon to target system*  or select the menu *Run*. This logon has to be done only at the first download. The next window offers the option PC-Simulation and e.pac. Select the target system, in case the e.pac does not start, the settings (IP address) have to be checked. To see the proper function of the project in principle or to test the project without an e.pac being connected the PC-Simulation can be selected.



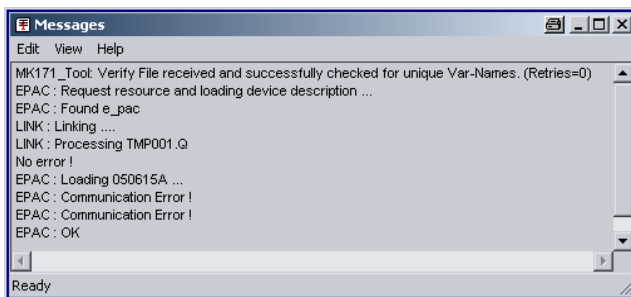
**Attention:** This simulation doesn't recognize errors regarding the time relevant functions.

Picture 10.2 - Select target system

Symbol	Target System State
	No connection
	Active connection
	Connection not used

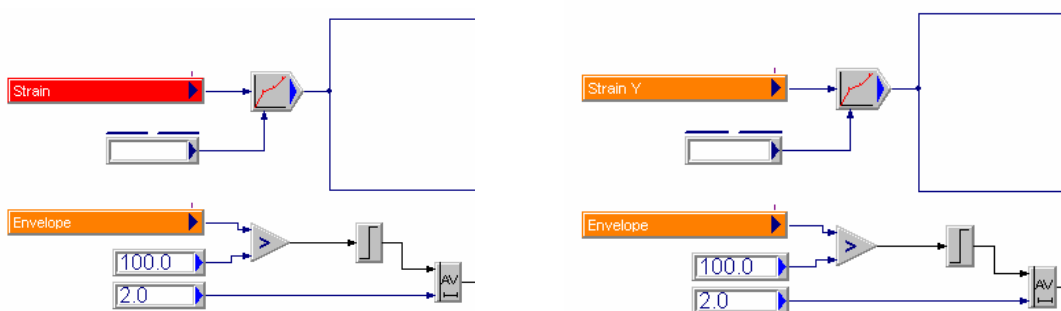
Picture 10.3 - Meaning of the symbols

- After the target system is selected the download can be started with the function *Download* or with the menu *Run*. After confirming the saving of the application the download will start. Afterwards the Online Observation State will start automatically.
- The Messages window follows up the process and confirms the download with an ok. The note "Communication Error!" is not relevant and it is caused by the FTP data transfer.



Picture 10.4 - Messages window after downloading an application

If it is necessary to change a name, a format or the order of the module variables or to add a new variable the *#summary.sta* file will be changed by the *e.pac* (see 7.2). After starting the *e.con* project again and doing another download an error message will appear. In the *Run* state the modified variables will change color from orange to red. Return to the *Edit* State and with the menu *Select device* the modified *#summary.sta* file will be read. Use the context menu *Parameter Dialog...* to refresh the actual setup.

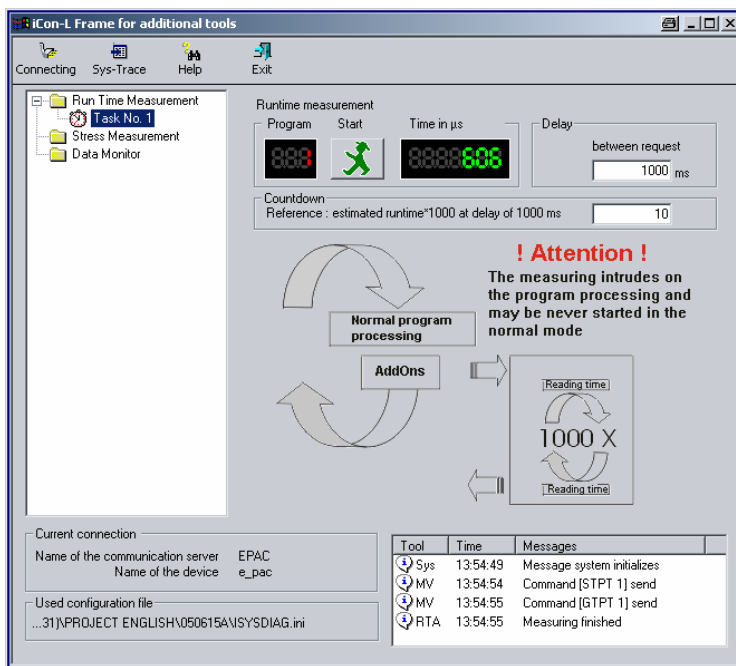


Picture 10.5 - Red variable after changing the name, orange variable after re-introduction using the Parameter Dialog

### 10.3. System Diagnostic

The e.pac is a powerful system to run quite complex applications. To run the system fast in a real time relevant application the effective time relations in the e.pac has to be well known. Therefore e.con offers a diagnostic tool. To use this tool it is once necessary to enter the run state and to logon to the target system. Later the diagnostic is available in the *edit state* too.

With the menu Tool - System diagnostic... the diagnostic window will open. Start a new item in the tree *Run Time Measurement* and start the *Runtime Measurement* with the button :



Picture 10.6 - Runtime measurement of an application in the target system (e.pac)

### 10.4. Tips

Some functions and features are supporting the work with e.con. As those not specifically belong to the described items in the chapters above they will be listed in the following. The description of the shortcuts and commands is available in the menu ? - *Shortcuts* .

Function	Where to find	Explanation
<b>Edit State</b>		
Move Worksheet	context menu worksheet	move all elements of a worksheet e.g. to add elements
Insert column/row	context menu worksheet	get more space into the application
Copy/paste	context menu worksheet	copy and paste a SFB in the worksheet
Left...Align	select blocks - context menu	align blocks left/right/top/bottom
Equal ...Separation	select blocks - context menu	place blocks horizontal/vertical equally spaced
Negate Connector	context menu input logic SFB	negate an input of a logical function block (OR, AND)

Mark Connection	context menu on a line	color the connections red to see where connected
De-Mark Connection	click on the marked line	change the color into black again
Orthogonal Connection	context menu connector	modify a "cross-country" line into an orthogonal line
Undo	menu Edit	undo the latest operations
Demo Time	Menu ? - About	available time of the demo mode for this project

**Run State**

View New Window	context menu program/macro	open the program/macro in a new window
Upwards	context menu worksheet	move one step upwards in the hierarchy

**Online Observation State**

What's in the line	Double click onto a line	show the actual signal/state in a connection line
Upwards	context menu worksheet	move one step upwards in the hierarchy



## 11. SUPPORT

It is our goal to provide our users an excellent service and support - that's our image. To support our user regarding e.pac and e.con application we are offering different possibilities.

**Help system**

The e.con software contains an online help with very detailed information and some examples.

**Hotline**

For a telephone support we offer the line +49 6151 951360 without extra costs.

**Seminar**

To our customers we offer a free 1 day seminar to learn about e.pac and e.con.

**Project management**

To provide "turn-key" solutions to our customers we work together with exclusive partner to fulfill your requirements on the base of e.bloxx series.



---

**Notice:**

Information in this manual is valid from July 17<sup>th</sup> 2005 until revocation.  
Further changes and completion of this manual are reserved and possible without notice.

— [www.gantner-instruments.com](http://www.gantner-instruments.com) —

GANTNER INSTRUMENTS TEST & MEASUREMENT GMBH  
Montafonerstraße 8 • A-6780 Schruns/Austria  
Tel.: +43 (0)5556-73784-410 • Fax: +43 (0)5556-73784-419  
E-Mail: [office@gantner-instruments.com](mailto:office@gantner-instruments.com)

**Gantner**  
instruments