ETAS INTECRIO V5.0

Getting Started

www.etas.com

INTECRIO V5.0 I Getting Started R05 EN - 06.2023

Contents

1 Safety and Privacy Information

In this chapter, you can find information about the intended use, the addressed target group, and information about safety and privacy related topics.

Please adhere to the ETAS Safety Advice (**Help > Safety Advice**) and to the safety information given in the user documentation.

ETAS GmbH cannot be made liable for damage which is caused by incorrect use and not adhering to the safety information.

1.1 Intended Use

INTECRIO is an integration platform for prototyping of automotive embedded control systems. It allows for the integration of application software from a wide variety of sources (e.g., ASCET, MATLAB® and Simulink®, C code) on embedded control units.

Virtual prototyping minimizes development times and is therefore becoming increasingly important. With the virtual prototyping capabilities of INTECRIO, system models can be analyzed without the need for complex prototyping hardware.

With the rapid prototyping capabilities of INTECRIO, control and diagnostic functions can be validated and verified under real conditions – also in the vehicle. Prototypes can be integrated into existing ECU vehicle networks via the ETAS rapid prototyping hardware. In a bypass application (via ETK, XETK, FETK, and XCP), INTECRIO enables the rapid prototyping hardware to be used as a simulation controller and calculates the parameters for new ECU functions.

1.2 Target Group

This manual is intended for trained personnel specializing in the area of function and software development for embedded electronic systems.

INTECRIO users should be familiar with the operating system Microsoft Windows® 10 or Windows® 11.

Knowledge of a programming language, preferably ANSI-C, can be helpful to advanced users.

Knowledge of the behavioral modeling tools ASCET and MATLAB and Simulink supported by INTECRIO are assumed.

1.3 Classification of Safety Messages

Safety messages warn of dangers that can lead to personal injury or damage to property:

> ⚠ DANGER
>
> **DANGER** indicates a hazardous situation with a high risk of death or serious injury if not avoided.

WARNING

WARNING indicates a hazardous situation of medium risk, which could result in death or serious injury if not avoided.

CAUTION

CAUTION indicates a hazardous situation of low risk, which may result in minor or moderate injury if not avoided.

NOTICE

NOTICE indicates a situation, which may result in damage to property if not avoided.

1.4 Safety Information

Please adhere to the ETAS Safety Advice and to the following safety information to avoid injury to yourself and others as well as damage to property.

WARNING

Harm or property damage due to unpredictable behavior of vehicle or test bench

Wrongly initialized NVRAM variables can lead to unpredictable behavior of a vehicle or a test bench. This behavior can cause harm or property damage.

INTECRIO systems that use the NVRAM possibilities of the experimental targets expect a *user-defined* initialization that checks whether all NV variables are valid for the current project, both individually and in combination with other NV variables. If this is not the case, all NV variables have to be initialized with their (reasonable) default values.

Due to the NVRAM saving concept, this is *absolutely necessary* when projects are used in environments where any harm to people and equipment can happen when unsuitable initialization values are used (e.g. in-vehicle-use or at test benches).

In addition, take all information on environmental conditions into consideration before setup and operation (see the documentation of your computer, hardware, etc.).

Further safety advice for this ETAS product is available in the following formats:

- In electronic form on the installation medium: `Documentation\General\ETAS Safety Advice.pdf`
- The "ETAS Safety Advice" window that opens when you start the program, or when you select **Help > Safety Advice**.

1.5 Privacy

Your privacy is important to ETAS so we have created the following Privacy Statement that informs you which data are processed in INTECRIO, which data categories INTECRIO uses, and which technical measure you have to take to ensure the users' privacy. Additionally, we provide further instructions where this product stores and where you can delete personal data.

1.5.1 Data Processing

Note that personal data respectively data categories are processed when using this product. The purchaser of this product is responsible for the legal conformity of processing the data in accordance with Article 4 No. 7 of the General Data Protection Regulation (GDPR). As the manufacturer, ETAS GmbH is not liable for any mishandling of this data.

1.5.2 Data and Data Categories

Please note that this product creates files containing file names and file paths, e.g. for purposes of error analysis, referencing source libraries, or for communicating with third-party programs.

The same file names and file paths may contain personal data, if they refer to the current user's personal directory or subdirectories (e.g., `C:\Users\`*<UserId>*`\Documents\...`).

Furthermore, using ETAS Rapid Prototyping solutions in test vehicles connected to real sensors, buses or ECUs, the ETAS tools may get access to personal data of the driver.

This data can also be stored using dataloggers as provided by INCA-EIP or the ETAS Experiment Environment.

When using the ETAS License Manager in combination with user-based licenses, particularly the following personal data respectively data categories can be recorded for the purposes of license management:

- Communication data: IP address
- User data: UserID, WindowsUserID

1.5.3 Technical and Organizational Measures

This product does not itself encrypt the personal data respectively data categories that it records. Ensure that the data recorded are secured by means of suitable technical or organizational measures in your IT system.

Personal data in log files can be deleted by tools in the operating system.

2 About INTECRIO

Nowadays developers of ECU software often have the problem that control algorithms are developed for an Embedded Control software without any target hardware available for execution. The algorithms are created with behavioral modeling tools such as ASCET or MATLAB® and Simulink® – i.e. with tools that enable the generation of code using models. To bridge the gap left by the missing target hardware, a Rapid Prototyping hardware system, such as the ES800 system by ETAS, is used.

INTECRIO is an ETAS product family which supports users in their daily work developing Embedded Control software by providing a platform for Rapid Prototyping.

INTECRIO integrates code from various behavioral modeling tools, makes it possible to configure the prototype as well as a hardware system for Rapid Prototyping, and allows the generation of executable code.

This manual supports the user when getting to know INTECRIO to ensure fast results. It provides a step-by-step introduction to the system with all information easy to look up.

2.1 System Information

The INTECRIO product family consists of several components:

- **INTECRIO-IP** – the integration platform of INTECRIO. It contains everything you need to integrate models from different behavioral modeling tools and to generate an executable prototype.

  Several sample files used in the tutorial and an example for the configuration and control of a third-party hardware I/O module are provided; you find them in the `<sample files>` directory specified during installation (cf. Page 19).

  The package contains the following components:

  - *Project Configurator* – used to specify software systems and system projects
  - *Hardware Configurator* – used, in combination with INTECRIO-RP or INTECRIO-VP, to configure the prototyping hardware and to connect hardware and software.
  - *OS Configurator* – used to configure the operating system.
  - *Project Integrator* – used to combine all components of a system project into an executable file.
  - *Documentor* – used to generate documentation for the components of a system project.
  - *MATLAB and Simulink Connectivity* – required to integrate Simulink models into INTECRIO for integration and Rapid Prototyping.
- **INTECRIO-RP** – the rapid prototyping package of INTECRIO. The package contains the following features:
  - *ES900 Connectivity* – used to configure the experimental target ES910 and the connection of hard- and software.
  - *ES800 Connectivity* – used to configure the experimental target ES830 with ES891, ES892, ES882, or ES886 and the connection of hard- and software.

- **INTECRIO-VP** – the virtual prototyping package of INTECRIO. The package contains the following features:
  - *VP-PC Connectivity* – used to configure and use the computer as virtual prototyping target.
- **ETAS Experiment Environment**

The following INTECRIO-related add-ons to ETAS products must be installed separately:

- *ASCET Connectivity* – makes it possible to integrate ASCET models into INTECRIO for integration and Rapid Prototyping.

  In ASCET V6.3 and higher, ASCET connectivity is integrated in ASCET-MD.

  In ASCET V6.2 and lower, ASCET connectivity is available as a separate add-on named INTECRIO-ASC, which was shipped with ASCET-MD, or as a part of ASCET-RP.

2.2 Documentation Structure

The *INTECRIO documentation* consists of an online help which describes the operation, a Getting Started manual which describes installation and also contains an overview of the most important features of operation, and a User Guide that provides background information.

The *INTECRIO Getting Started* manual consists of the following chapters:

- **"Safety and Privacy Information"**

  In this chapter, you can find information about the intended use, the addressed target group, and information about safety and privacy related topics.
- **"About INTECRIO" (this chapter)**

  This chapter gives an initial overview of the possible field of application of INTECRIO.
- **"Installation"**

  This chapter is intended for all users who install, maintain and uninstall INTECRIO. It provides important information on the delivery scope, hardware and software requirements. The sequence of both the installation and uninstallation of INTECRIO is described.
- **"INTECRIO Quick Guide"**

  This chapter is a quick introduction to the program concept of INTECRIO. The realistic working examples, displayed in the form of flowcharts, provide you with an overview of the program functionality and operating mode.

- **"INTECRIO Tutorial"**

  The tutorial is intended for new users of INTECRIO. You will learn how to work with INTECRIO using examples. The information is divided into short individual sections. The first seven lessons each lead on to the next, whereas lesson 8 is an isolated example. Before you work with the tutorial, you should have worked through chapter 3 "Understanding INTECRIO" in the user guide.

  > i NOTE
  >
  > ETAS provides training sessions for a more thorough introduction to INTECRIO which are particularly useful if you want to learn about working with INTECRIO in-depth in a relatively short period of time.

- **"ETAS Network Manager"**

  This chapter contains information on the ETAS Network manager.

- **"Troubleshooting General Problems"**

  This troubleshooting chapter gives some information of what you can do when problems arise that are not specific to an individual software or hardware product.

- **"Contact Information"**

  This chapter contains ETAS contact addresses.

- **"Glossary"**

  Abbreviations and specialist terms which occur in the Getting Started manual or the user guide are described in the glossary. The terms are listed in alphabetical order.

The *INTECRIO user guide* consists of the following chapters:

- **"Safety and Privacy Information"**

  In this chapter, you can find information about the intended use, the addressed target group, and information about safety and privacy related topics.

- **"About INTECRIO"**

  This chapter gives a very brief overview of INTECRIO, and explains the purpose of the user guide..

- **"Understanding INTECRIO"**

  This chapter provides an overview of the INTECRIO system and the development process supported by the system. This chapter should be read first by all new INTECRIO users.

- **"INTECRIO and AUTOSAR"**

  This chapter describes how INTECRIO supports AUTOSAR.

- **"INTECRIO Components"**

  This chapter describes the individual components of INTECRIO and their tasks. It is not intended as operating instructions; for this purpose, please use the online help.

- **"SCOOP and SCOOP-IX"**

  This chapter describes the concept for the description of interfaces, SCOOP, and the relevant interface description language, SCOOP-IX.

- **"Modeling Hints"**

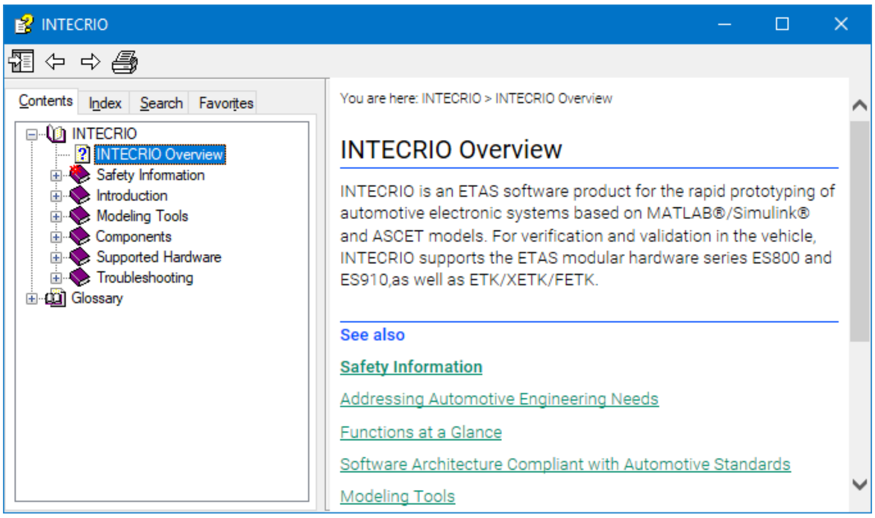
  This chapter provides an overview of the modeling philosophy of INTECRIO and describes how the behavioral modeling tools are used in conjunction with INTECRIO.
- **"Contact Information"**

  This chapter contains ETAS contact addresses.
- **"Glossary"**

  Abbreviations and specialist terms which occur in the Getting Started manual or the user guide are described in the Glossary. The terms are listed in alphabetical order.

2.3 INTECRIO Online Help – Quick Guide

Use the **Help > Help** menu option or the button to invoke the general help function. Press the <F1> function key to call context-sensitive help.



The tabs of the help window provide you with the following options:

- The "Contents" tab allows you to browse the help topics by categories.
- The "Index" tab lists all index entries. Browse the entire list, or enter a search term to limit the scope of listing.
- The "Search" tab allows you to search for individual words or terms included in a help topic. Type a search string and let the help function list the entries it has found related to this term.
- The "Favorites" tab allows you to bookmark topics.

3 Installation

This chapter is intended, on the one hand, for all users who install, maintain and uninstall INTECRIO or the ETAS Experiment Environment on a computer or in a network, and, on the other, for system administrators who make INTECRIO or the ETAS Experiment Environment available on a file server for installation via the network. It provides important information on the delivery scope, hardware and software requirements for single station and network installation as well as on how to prepare the installation. The sequence of both the installation and uninstallation of INTECRIO and the ETAS Experiment Environment is also described.

> **NOTE**
>
> INTECRIO V5.0.4 can be installed on Windows 11 computers. The ETAS Experiment Environment V3.8.4, which is supported by INTECRIO, *cannot* be installed on Windows 11.
>
> When you install INTECRIO on a Windows 11 computer, you have to use INCA/INCA-EIP V7.4 for experiments.

3.1 Preparation

Check the delivery package to make sure it is complete and make sure your system corresponds to the system requirements. Depending on the operating system and network connection used, you must ensure that you have the necessary user privilege.

3.1.1 Delivery Scope

The delivery scope of INTECRIO includes:

- INTECRIO installation disk
  - INTECRIO program files
  - ETAS Virtual OS Execution Platform / RTA tools required for virtual prototyping
  - INTECRIO Getting Started and User Guide as well as ETAS hardware documentation in PDF format[1)]
  - Safety hints in PDF format[1)]
  - ETAS Experiment Environment program files
  - Manual "Licensing End User Guide" in PDF format[1)]

    a link to the HSP web site, which provides information on HSP and the most recent program files

1) PDF reader required

- MDA program files
- information on open-source components used in INTECRIO

Furthermore, you require valid licenses for the use of INTECRIO or the ETAS Experiment Environment. You can obtain the license files required for licensing either from your system administrator or through a self service portal on the ETAS Internet Site; the URL is given in the entitlement letter. To request the license file you have to enter the activation number which you received from ETAS during the ordering process.

3.1.2 System Prerequisites

The INTECRIO system prerequisites are listed in the release notes of INTECRIO.

The system prerequisites for the ETAS Experiment Environment are given in section 3.3.1 on page 27.

3.1.3 Required User Privileges for Installation and Operation

In order to install INTECRIO or the ETAS Experiment Environment on a computer, you need the user privileges of an administrator. Please contact your system administrator, if necessary.

In order to operate INTECRIO, you do not need special privileges.

The privileges required to operate the ETAS Experiment Environment depend on the target server version you are using.

3.2 Installing INTECRIO

To work with INTECRIO, you must install at least the INTECRIO integration platform. In addition, you can choose individually to install the add-on for rapid prototyping and the add-on for virtual prototyping. The ETAS Experiment Environment is independent from the INTECRIO integration platform and its add-ons. It can be installed and used separately and individually. Without the ETAS Experiment Environment, you can use INTECRIO to configure projects and generate code for them. However, no experimenting is possible, and the **Open Experiment Environment** toolbar button and menu option in INTECRIO will be disabled.

The installation is independent of whether you install INTECRIO or the ETAS Experiment Environment from data carrier or a network drive.

Section 3.2.1 describes the initial installation of INTECRIO. Special installation features of the Virtual Prototyping package are described in section 3.2.2.
Section 3.2.4 lists the options for command line installation.

3.2.1 Initial Installation

<u>To start the INTECRIO installation</u>

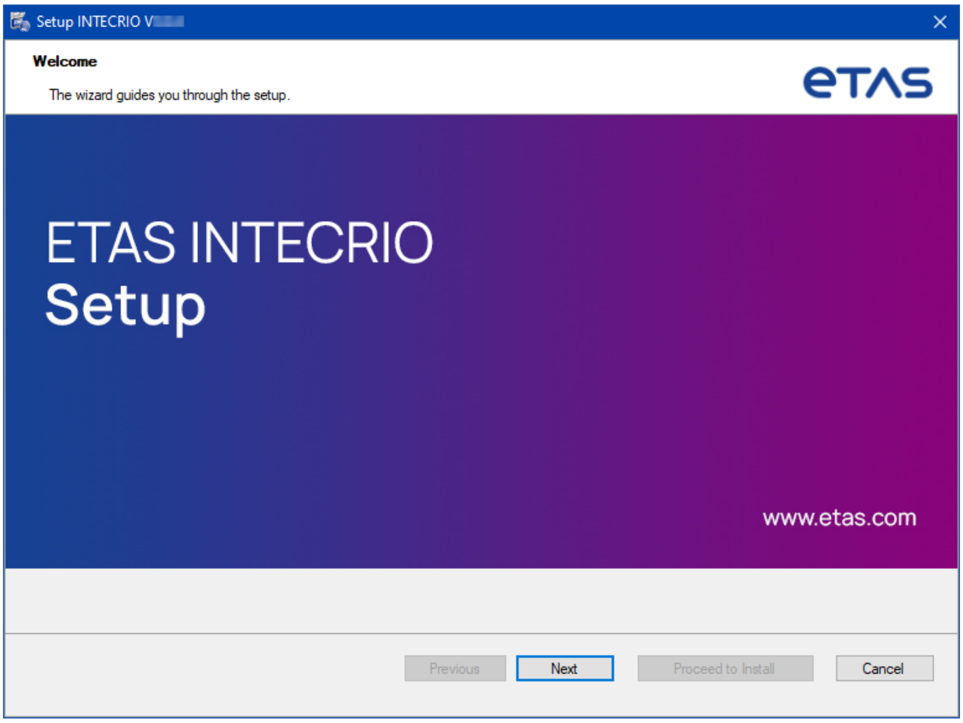
For a list of supported operating systems, see the INTECRIO release notes. If you try to install INTECRIO on a computer with an unsupported OS, an error message opens and the installation is aborted.

1. Insert the data carrier in the respective drive on your computer.

   An installation dialog window opens.

2. In that window, follow the "Installation" link, then follow the "INTECRIO V5.0.*" link.
3. Alternatively, select the drive in the Windows Explorer and run the `setup.exe` file from the `Installation` folder.

   The ETAS Installer is launched.



4. Click **Next** to get to the next installation window.

   Use **Back** to get to the previous window and **Cancel** to cancel installation.

   The installer checks if your computer meets the system requirements for INTECRIO installation. The result is displayed in the "System Check" window.

   If your system meets the installation requirements, the installer proceeds automatically to the next window.

Uninstalling a previously installed version

> **NOTE**
>
> If no incompatible version of INTECRIO is installed on your computer, continue with “License Agreement and Safety Hints” on page 17.

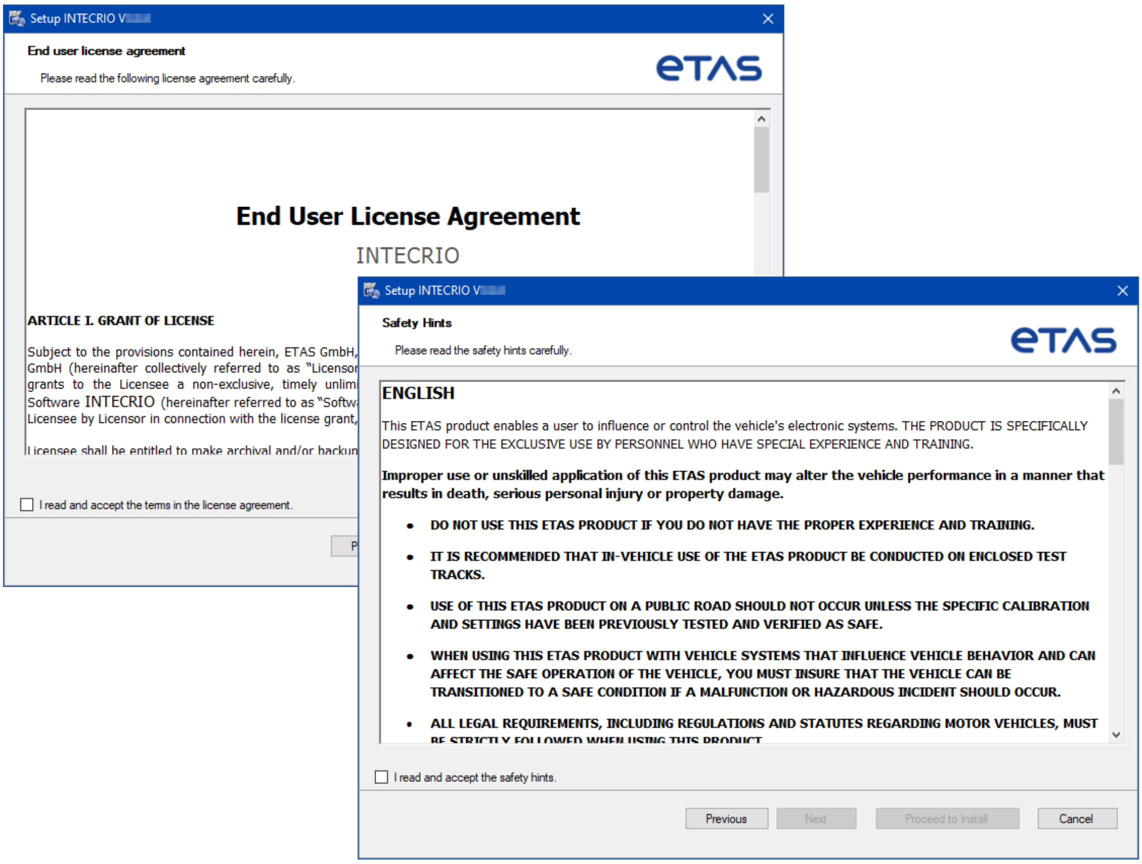
If an INTECRIO version incompatible with INTECRIO V5.0.4 is present on your computer, that version is listed in the "Uninstall previous products" window.

1. Click **Uninstall now**.

   The previous version of INTECRIO is uninstalled. You can now continue the installation of INTECRIO V5.0.4.
2. When the uninstallation is complete, click **Next** to continue.

License Agreement and Safety Hints

The following two windows show license agreement and safety hints in several languages.



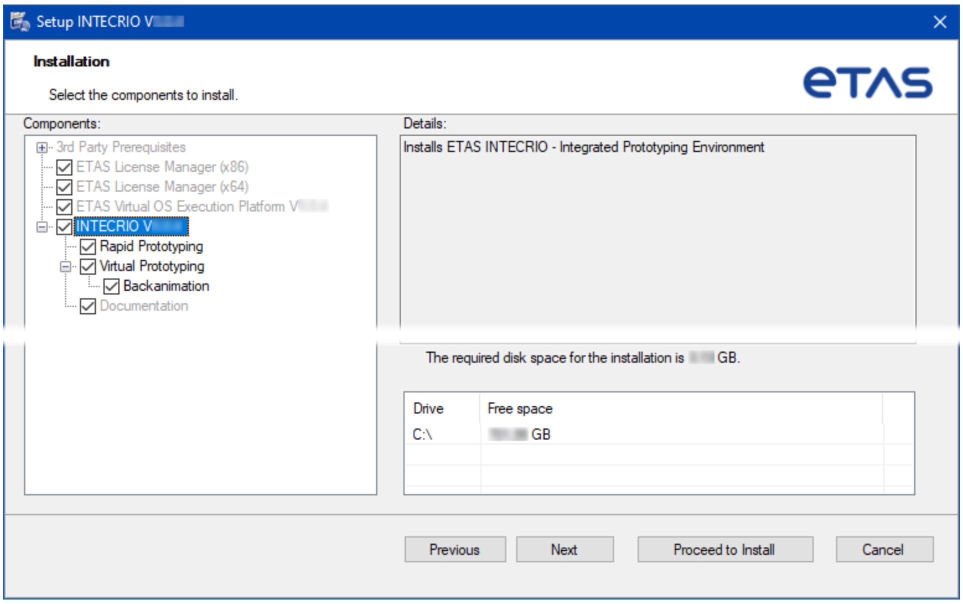
1. Read the license agreement, then activate the **I read and accept the terms in the license agreement** option.

   If you do not accept the license agreement, you cannot continue the installation.
2. Click **Next**.
3. Read the safety hints carefully, then activate the **I read and accept the safety hints** option.

   If you do not accept the safety hints, you cannot continue the installation.
4. Do one of the following:
   - Click **Next**.
   - Click **Proceed to Install** to go to the "Ready to install" window immediately, with default settings for components to be installed, installation paths and loopback network address.
     Continue reading in section "To install INTECRIO" on page 21.

To select components

The component selection window lists the necessary 3rd-party prerequisites, ETAS tools (i.e. the license manager and the ETAS Virtual OS Execution Platform), and the components of INTECRIO. Prerequisites and components already present on your computer are marked accordingly.



Components whose names appear in black font can be selected or deselected for installation. Thus, you specify the functional scope of the installed software.

The text field on the right displays information on the highlighted component.

1. To select/deselect a component for installation, activate/deactivate the respective option.

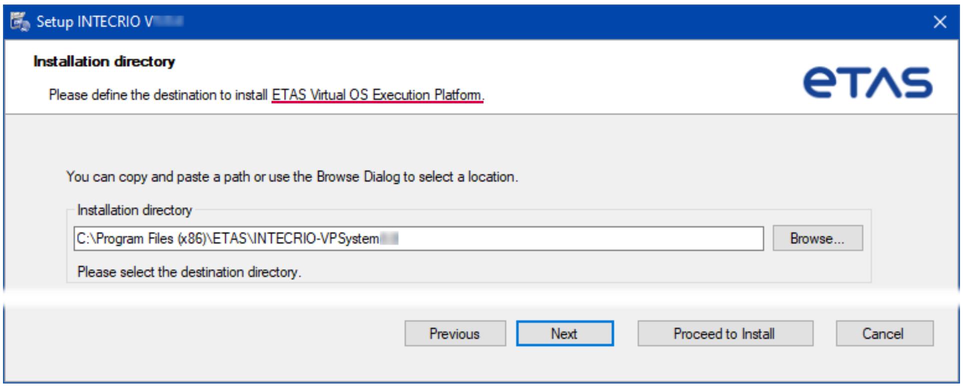
   > **NOTE**
   >
   > You need at least one of the Rapid Prototyping and Virtual Prototyping packages to create and configure hardware systems.

   Installing INTECRIO without Rapid Prototyping and Virtual Prototyping packages may be useful if you want to create modules for usage with INTECRIO out of Simulink.

2. Do one of the following:
   - Click **Next**.
   - Click **Proceed to Install** to go to the "Ready to install" window immediately, with default settings for installation paths and loopback network address.
     Continue reading in section “To install INTECRIO” on page 21.

To define path settings

In the first "Installation directory" window, you are prompted to enter a destination directory for the ETAS Virtual OS Execution Platform.



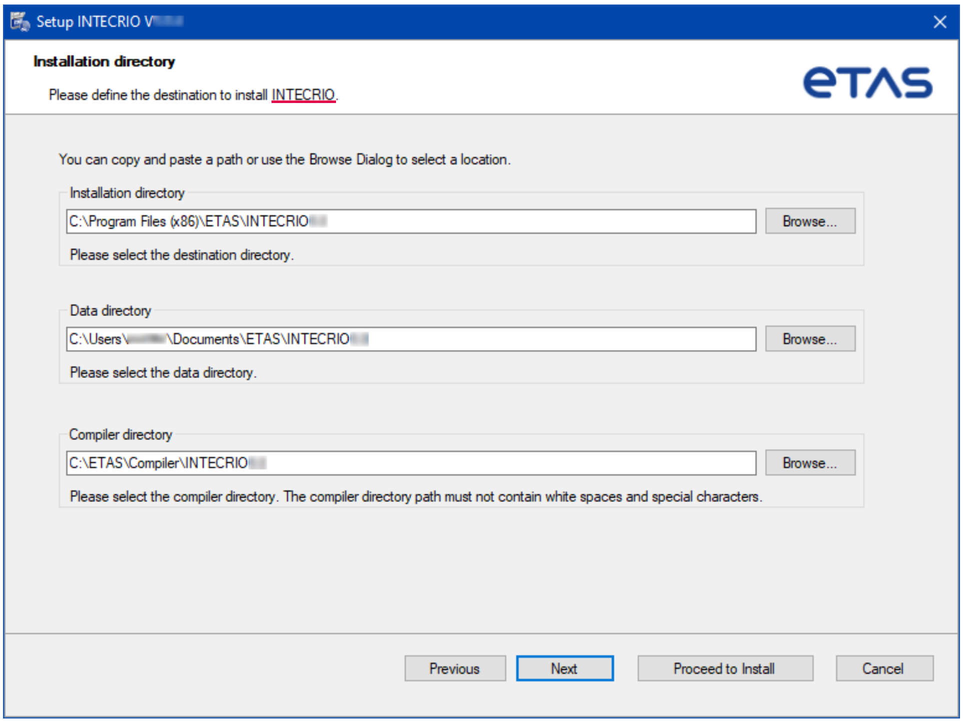
NOTE

This window is omitted if the ETAS Virtual OS Execution Platform is already installed on your computer.

1. Enter or select (via the **Browse** button) a valid path.

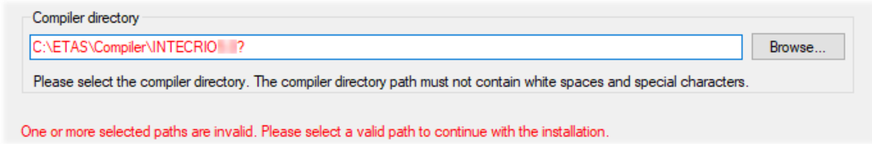
   An invalid path is displayed in red, and a warning appears.
2. Click **Next**.

In the second "Installation directory" window, you are prompted to enter destination directories for the INTECRIO installation (referred to as *`<installation>`*), for sample files (referred to as *`<sample files>`*) and for the compilers.



3. To change a preset directory, enter or select (via the **Browse** button) the new path.

   If you entered an invalid path, that path is displayed in red, and a warning appears.



NOTE

Paths with blanks are not possible for the compilers used in INTECRIO. If you entered a compiler path with blanks, that path is marked as invalid.

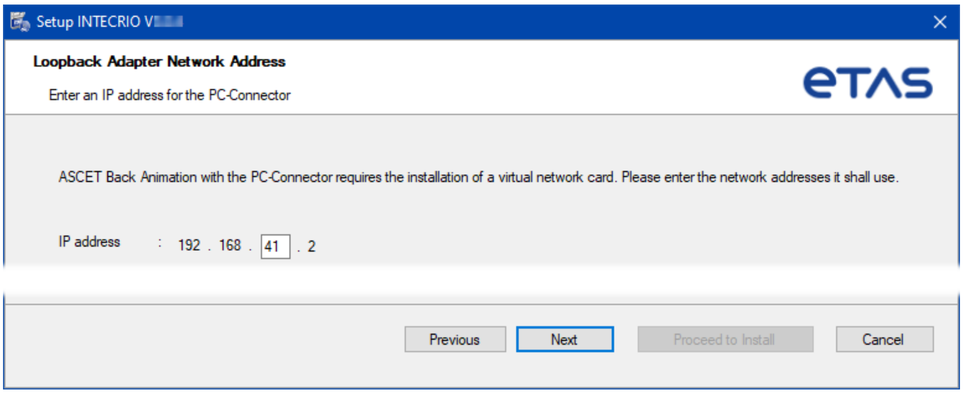
4. Do one of the following:
   - Click **Next**.
   - Click **Proceed to Install** to go to the "Ready to install" window immediately, with default settings for installation paths and loopback network address.

     Continue reading in section "To install INTECRIO" on page 21.

To enter the loopback network address (Virtual Prototyping only)

If you are installing the Back Animation part of the Virtual Prototyping package, a virtual network board is installed automatically. You have to enter an IP address for this board.

1. In the "Loopback Adapter Network Address" window, complete the address.



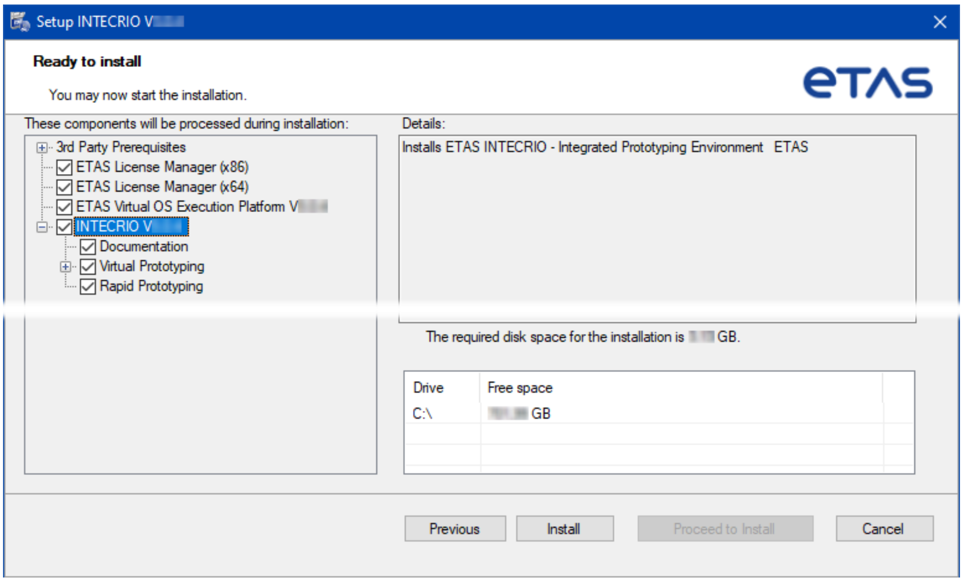
   The address must be free; if in doubt, contact your system administrator.

   If you enter a wrong address, a warning is issued.

2. Click **Next** or **Proceed to Install** to continue.

To install INTECRIO

The "Ready to install" window lists the components selected for installation. You cannot change the selection here; to do so, you must go back to the "Installation" window (cf. Page 18).



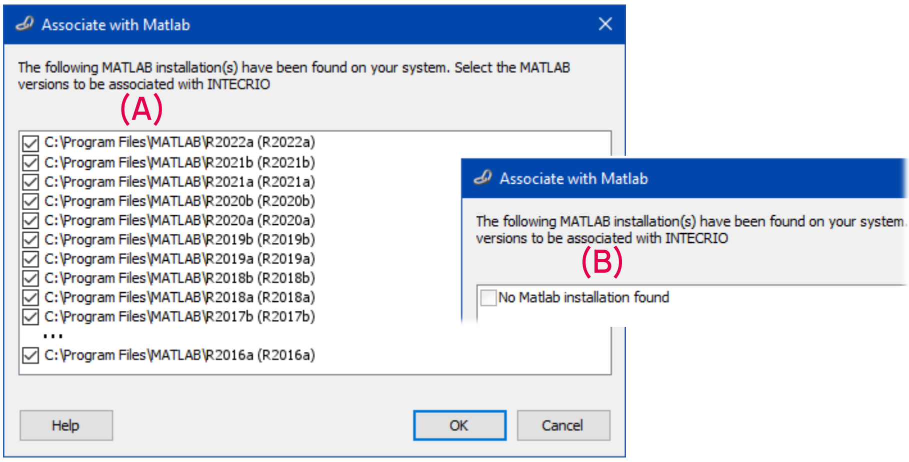
> **NOTE**
>
> The next step will start the installation. You **cannot** cancel the installation once it is running; the **Cancel** button is deactivated.

1. In the "Ready to Install" window, click **Install**.

   The selected components (cf. Page 18) are installed. Components already present on your computer are skipped. A progress indicator shows how the installation is progressing.

To connect INTECRIO and MATLAB and Simulink

After the installation is completed, the "Association with MATLAB" window opens. It offers all supported MATLAB and Simulink installations (R2016a – R2022a and their related service packs known at the time of the INTECRIO V5.0.4 release) available on your system for selection (A in the screenshot below). Even if no MATLAB and Simulink installation is found, the "Association with MATLAB" window opens (B in the screenshot).



> **NOTE**
>
> If you want to use a network installation of MATLAB and Simulink, click the **Help** button and follow the instructions given in the message window.

1. In the "Association with Matlab" window, select the MATLAB and Simulink installation(s) you want to associate with INTECRIO.

   You can select no, one or several installations.

   Cancel closes the window without establishing an association to MATLAB and Simulink. MATLAB and Simulink connectivity is installed nevertheless, but you must associate INTECRIO to MATLAB and Simulink manually before you can use it.

2. Click **OK**.

   INTECRIO is associated with the selected MATLAB and Simulink installation(s).

   You can change the MATLAB and Simulink version associated with INTECRIO later; the procedure is described in section 5.1 "MATLAB® and Simulink® Connectivity" in the INTECRIO user guide.

To complete the installation

Once all components have been installed successfully, you are prompted to end the installation.

1. It is recommended that you activate the **Show readme when setup is closed** option
2. Click **Finish** to end the installation.

In the Start menu, the **ETAS INTECRIO 5.0** folder with the following entries is created:

- **INTECRIO V5.0**

  The INTECRIO program is started.

- **INTECRIO V5.0 – Tools** folder

  Opens a directory with the following tools:

  - **Associate with ASCET**

    Determines the INTECRIO version(s) that can be used during ASCET code generation.
  - **Associate with MATLAB**

    Connects a selectable MATLAB and Simulink installation with INTECRIO (see also section "MATLAB® and Simulink® Connectivity" in the INTECRIO user guide).
    This is necessary if MATLAB and Simulink is being newly installed without INTECRIO being newly installed at the same time.
- **Manuals and Tutorials**

  Opens the ETAS documentation directory, which contains a link to the INTECRIO documentation.

The ETAS License Manager can be found in the app list of the Windows Start menu at **E > ETAS > ETAS License Manager**.

The ETAS Virtual OS Execution Platform has another separate folder **ETAS Virtual OS Execution Platform *<x>.<y>*** in the Windows Start menu, with the following content.

- **Signal Configuration Editor V *<x>.<y>***

  Irrelevant for INTECRIO.

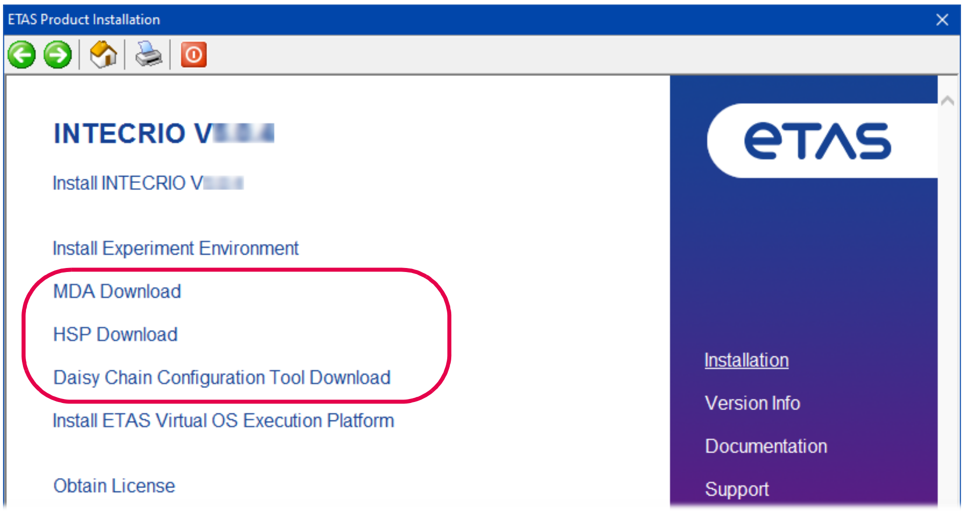
The Folder `LicenseTerms` in the `<installation>` directory contains license terms for open-source software delivered with INTECRIO.

The following icon is stored on the desktop of your workstation:

3.2.2 Special Installation Steps: Virtual Prototyping Package

Virtual prototyping requires *RTA-OSEK Tools* and *RTA-OSEK for PC*. These products are installed with INTECRIO as parts of the *Virtual Prototyping* installation component (see “To select components” on page 18).

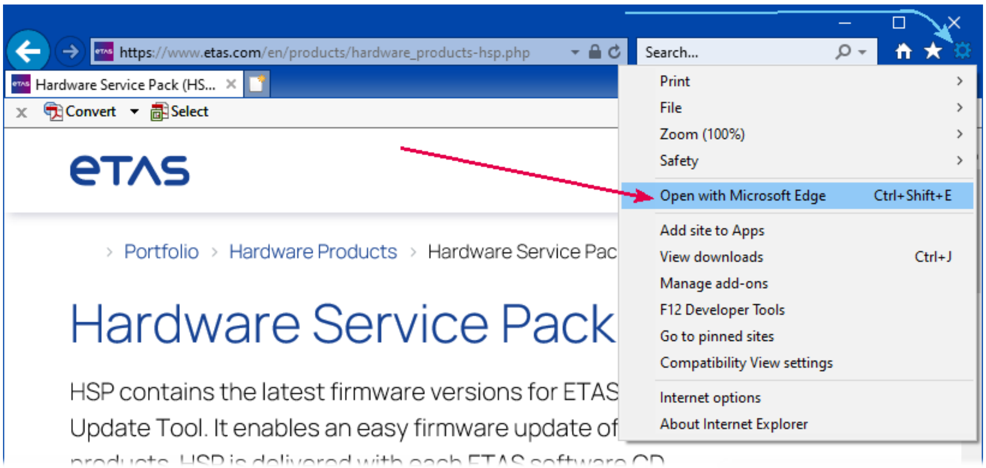
3.2.3 Special Installation Steps: Auxiliary Tools



The "ETAS Product Installation" window offers download links for the MDA, HSP, and Daisy Chain Configurator.

In Windows 10, these links are opened in Internet Explorer, however, the ETAS download center does not allow download via Internet Explorer.

You have to open the download link in another browser, e.g., via the **Open with Microsoft Edge** menu function (red arrow) in the drop-down menu (light-blue arrow), or via Copy&Paste.



3.2.4 Command-Line Installation

When you start the INTECRIO installation from a command line, you can use several command-line parameters to customize the installation.

Each execution of `Setup.exe` writes a log file *`<date_time>`* `Setup.log`. These log files are stored in the `C:\ProgramData\ETAS\SETUP\Logs` folder.

/silent

Silent installation mode. With this installation mode, no dialog windows requiring user information open.

> NOTE
>
> To accept EULA and safety hints (cf. Page 17) during silent installation, you must use the command-line parameters **/EULAAccepted** and **/SafetyHintsAccepted**.
>
> To deal with a possible request for a computer restart, you must use either the **/NoRestart** or the **/AllowRestart** command-line parameter.
>
> To automatically uninstall an existing older INTECRIO version, you must use the **/UninstallPreviousVersion** command-line parameter.

Default values are used for all information normally requested in installation dialog windows. Error messages are hidden, too.

/EULAAccepted

Accepts the license agreement.

The text of the license agreement is provided on the installation disk, in the `EULA` subfolders.

/SafetyHintsAccepted

Accepts the safety hints.

The text of the safety hints is provided on the installation disk, in the `SafetyHints` subfolders.

/NoRestart

Suppresses a computer restart that may be required at the end of the installation. If a reboot is suppressed, a log message is issued.

/AllowRestart

Allows a computer reboot restart that may be required at the end of the installation. A restart is performed without further notice.

/UninstallPreviousVersion

Uninstalls an existing older INTECRIO version installed on your computer.

> NOTE
>
> If you do not use this command-line parameter, `setup.exe` will abort with an error If a previous INTECRIO version is found.

`/UninstallPreviousVersion` is not used in combination with `/Uninstall`.

/Debug

Writes additional log files for `*.msi` packages.

These files are stored in the `C:\ProgramData\ETAS\SETUP\Logs` folder.

/DefaultSettings

Allows to specify an own XML file with default settings (instead of using InstallationDefaultSettings.xml), e.g., the following:

Variable	Meaning	See also
PRODINSTDIR	installation directory	
PRODUSERDOCUMENTSDIR	data directory	Page 19
PRODCOMPILERDIR	compiler directory	
LOOPBACK_IP_ADDRESS	IP address for loopback adapter	Page 20

You can specify a relative path if the file is relative to Setup.exe, otherwise you have to specify an absolute path.

Syntax: /DefaultSettings:"*<path>**<filename>*.xml"

/Uninstall

Uninstalls INTECRIO. Can be combined with /silent for uninstallation without user interaction.

/Repair

In combination with /Silent, the repair process is triggered. Otherwise, the maintenance mode is triggered.

> NOTE
>
> /Uninstall and /Repair cannot be used with setup.exe in the installation location. You ***must*** use the setup.exe file provided in the C:*<programs>*\ETAS\GENericSetup\IPE INTECRIO-IP\ 5.0.*<x>*.*<y>* folder.
>
> *<programs>* = Program Files (x86) (64 bit OS) or Program Files (32 bit OS)

Examples:

```
Setup.exe /silent /EULAAccepted /SafetyHintsAccepted
```

Triggers a silent installation.

```
"C:\Program Files (x86)\ETAS\GENericSetup\IPE INTECRIO-
IP\5.0.<x>.<y>\setup.exe" /uninstall /Debug
```

Triggers a non-silent uninstallation and writes additional logs.

3.3 Installing ETAS Experiment Environment

> **NOTE**
>
> Beginning with V5.0.4, INTECRIO supports ETAS Experiment Environment V3.8.4, but not V3.9.
>
> ETAS Experiment Environment V3.7 is *no longer supported.*

3.3.1 System Prerequisites for the Experiment Environment

	Required	**Recommended**
hardware	- 2 GHz Pentium Dual Core computer - 2 GB RAM - Ethernet Adapter 10/100BaseT - DVD-ROM drive - Graphic adapter with DirectX 9 or higher - Resolution 1280x1024, 16 bit color - Hard disc ≥ 20 GB	- 2 GHz Pentium Dual Core computer - 4 GB RAM - 2nd Ethernet Adapter 10/100BaseT - DVD-ROM drive - Graphic adapter with DirectX 9 or higher and HW acceleration - Resolution 1600x1200, 512 MB, 32 bit color
operating system	- Windows® 10 (x64)	- Windows 10® (x64)
free disk space	- > 1.8 GB	- > 2 GB

3.3.2 Initial Installation

Without INTECRIO, you can use the ETAS Experiment Environment for experiments with existing projects. However, you cannot create projects.

If you try to install the ETAS Experiment Environment on a computer with an unsupported OS, an error message opens and the installation is aborted.

You cannot choose the installation path of the ETAS Experiment Environment. The ETAS Experiment Environment is installed below the `Program Files (x86)\Common Files` folder on your system drive.

<u>To start the ETAS Experiment Environment installation</u>

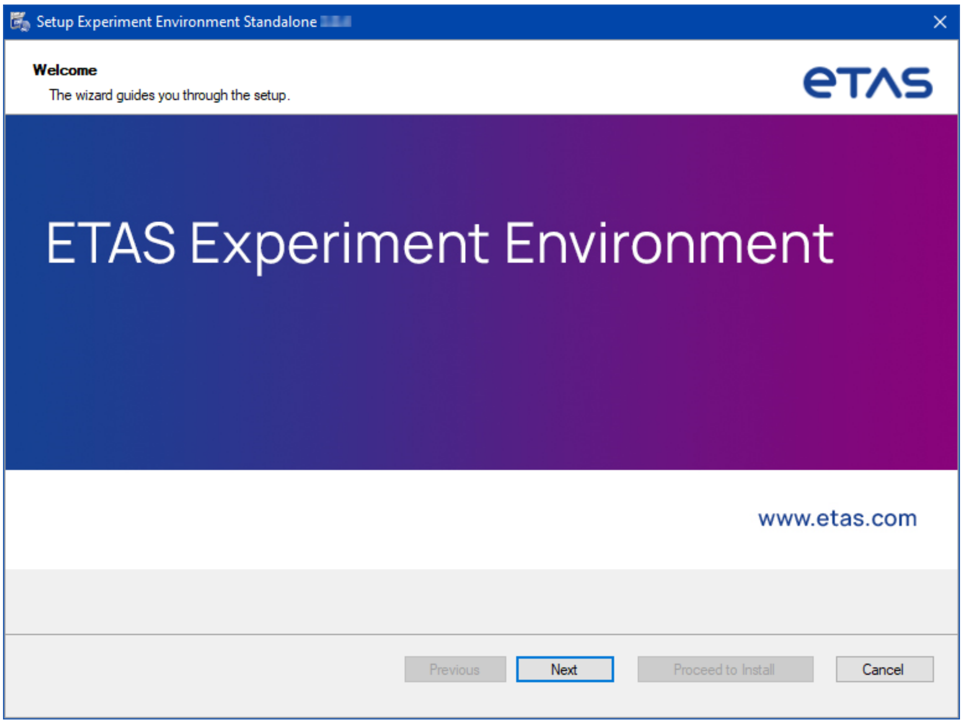
1. Insert the data carrier in the respective drive on your computer.

   An installation dialog window opens.

2. In that window, follow the "Main" link, then follow the "Experiment Environment V *<x>.<y>.<z>* Installation" link.

3. Alternatively, select the drive in the Windows Explorer, go to the `Experiment Environment` folder and run the `setup.exe` file.

   The ETAS Installer is launched.



4. Click **Next** to continue.

   The installer checks if your computer meets the system requirements for installation of the ETAS Experiment Environment. The result is displayed in the "System Check" window.

   If your system meets the installation requirements, the installer proceeds automatically to the next window.

License Agreement:

The next window shows the license agreement.

1. Read the license agreement, then activate the **I read and accept the terms in the license agreement** option.

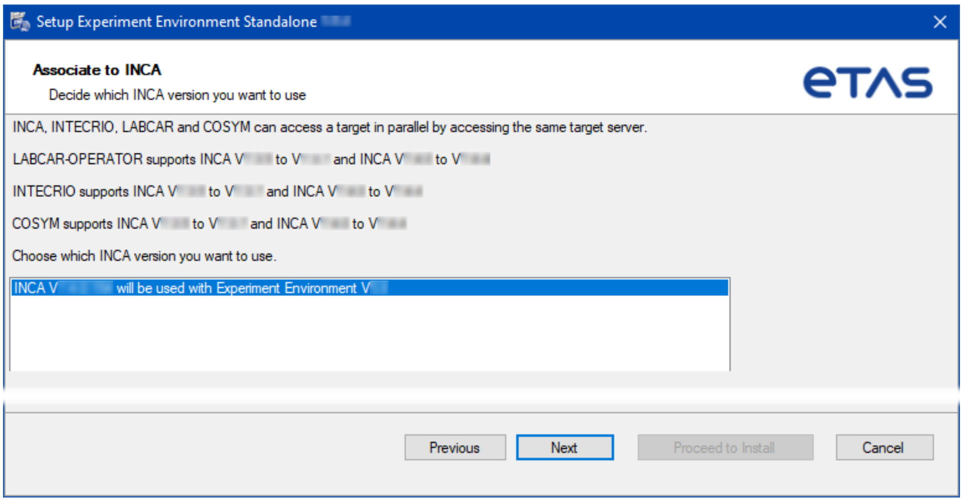
   If you do not accept the license agreement, you cannot continue the installation.

2. Do one of the following:
   - Click **Next** to continue.
   - Click **Proceed to Install** to go to the "Ready to Install" window immediately, with default settings for association with INCA and components to be installed.

     Continue reading in section “To install the ETAS Experiment Environment” on page 30.

To associate with INCA

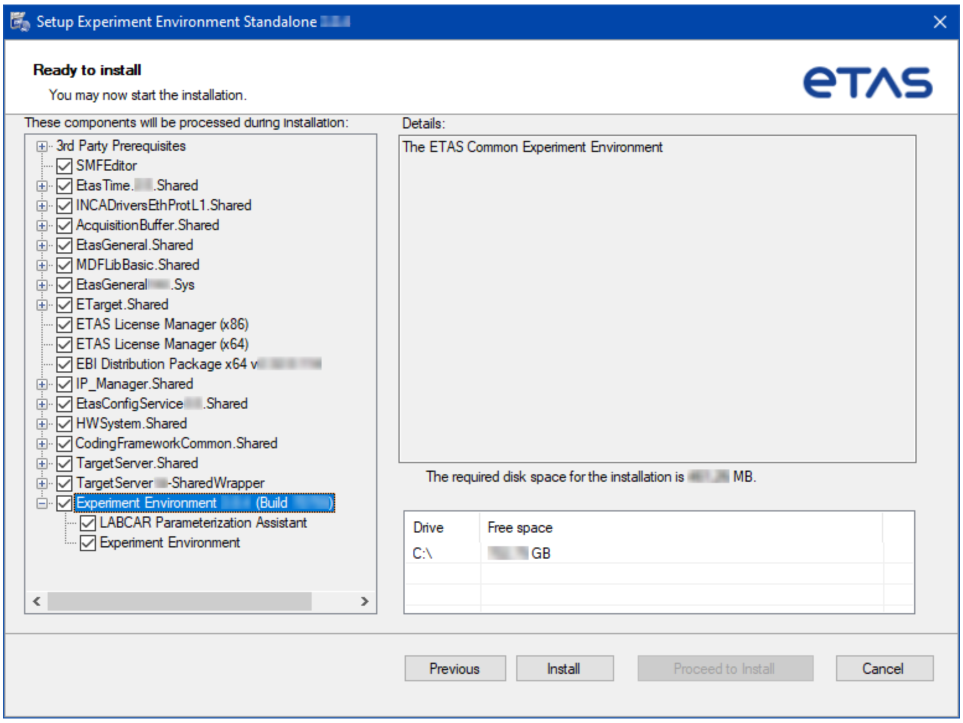
The "Associate to INCA" window offers all INCA versions on your computer that can access the same target server as the Experiment Environment, as well as the selection `Default (No INCA connection)`.



1. Select the INCA version you want to associate with the ETAS Experiment Environment.
2. Click **Next** or **Proceed to Install** to continue.

To install the ETAS Experiment Environment

The "Ready to install" window lists the components selected for installation.



> **NOTE**
>
> The next step will start the installation. You **cannot** cancel the installation once it is running; the **Cancel** button is deactivated.

1. In the "Ready to Install" window, click **Install**.

   The components are installed. A progress indicator shows how the installation is progressing.

To complete the installation

1. Once all components have been installed successfully, you are prompted to end the installation.
2. Click **Finish** to end the installation.

A folder **Experiment Environment *<x>.<y>*** is created in the ETAS program group of the Windows Start menu; *<x>.<y>* being the version number. This folder contains the following items:

- **Experiment Environment V *<x>.<y>*** Tools

  Opens a window that contains the following shortcuts:

  - **Associate to INCA**

    Determines the INCA version that can be used with the ETAS Experiment Environment, and the version of the shared modules.
  - **Associate to RTA-TRACE**

    Checks the installation of RTA-TRACE. If the check fails, hints for troubleshooting are given.
- **Experiment Environment V *<x>.<y>***

  Opens the ETAS Experiment Environment.
- **INCA V7.4 TargetServer – Tools**

  Opens a window that contains the following shortcuts.

  - **ETAS Network settings**

    Starts the assistant for the configuration of the ETAS network.
  - **Search for connected hardware**

    The Target server is started. A search for connected hardware is conducted.
- **Manuals and Tutorials**

  Opens the ETAS documentation directory, which contains a link to the documentation for the ETAS Experiment Environment.

  - **Instrument Programming**

    Opens a help file which offers assistance for programming and integrating user-defined instruments to the ETAS Experiment Environment.
  - **Online Help**

    Opens the ETAS Experiment Environment online help.
  - **Scripting API**

    Opens a help file that describes the scripting API of the ETAS Experiment Environment.

The following icon is stored on the desktop of your workstation:

The SMF editor is used in combination with LABCAR. For INTECRIO users, it does not provide useful functionality. It has a separate folder, **SMFEditor *‹x›.‹y›***, in the ETAS program group of the Start menu; this folder contains the following items:

- **SuT Mapping File Editor**

  Opens the SMF editor.
- **SMFE Online Help**

  Opens the online help for the SMF editor.

3.3.3 Command Line Installation

When you start the installation of the ETAS Experiment Environment from a command line, you can use several command-line parameters to customize the installation.

Each execution of `Setup.exe` writes a log file `%appdata%`[1]`\ETAS\SETUP\` *`<date_time>`*` Setup.log`.

The following command line options exist:

`/silent`

Silent installation mode. With this installation mode, no dialog windows requiring user information open.

> NOTE
>
> To accept EULA and safety hints (cf. Page 29) during silent installation, you must use the command-line parameters **`/EULAAccepted`** and **`/SafetyHintsAccepted`**.
>
> To deal with a possible request for a computer restart, you must use either the **`/NoRestart`** or the **`/AllowRestart`** command-line parameter.
>
> To automatically uninstall an older version of the ETAS Experiment Environment, you must use the **`/UninstallPreviousVersion`** command-line parameter.

Default values are used for all information normally requested in installation dialog windows. Error messages are hidden, too.

`/EULAAccepted`

Accepts the license agreement.

The text of the license agreement is provided on the installation disk, in the `ExperimentEnvironment\EULA` subfolders.

---

1) `%appdata% = C:\Users\`*`<username>`*`\AppData\Roaming`

/SafetyHintsAccepted

Accepts the safety hints.

The text of the safety hints is provided on the installation disk, in the `ExperimentEnvironment\SafetyHints` subfolder.

/NoRestart

Suppresses a computer restart that may be required at the end of the installation. If a reboot is suppressed, a log message is issued.

/AllowRestart

Allows a computer reboot restart that may be required at the end of the installation. A restart is performed without further notice.

/UninstallPreviousVersion

Uninstalls an existing older INTECRIO version installed on your computer.

> **NOTE**
>
> If you do not use this command-line parameter, `setup.exe` will abort with an error If a previous version of the Experiment Environment is found.

`/UninstallPreviousVersion` is not used in combination with `/Uninstall`.

/Debug

Writes additional log files for `*.msi` packages.

These files are stored in the `%appdata%`[1] `\ETAS\SETUP` folder.

/DefaultSettings

Allows to specify an own XML file with default settings (instead of using `InstallationDefaultSettings.xml`).

You can specify a relative path if the file is relative to `setup.exe`, otherwise you have to specify an absolute path.

Syntax: `/DefaultSettings:"<path>\<filename>.xml"`

/uninstall

Uninstalls the ETAS Experiment Environment. Can be combined with `/silent` for uninstallation without user interaction.

Use the `setup.exe` file provided in the `C:\<programs>`[1]`\ETAS\GENericSetup\EE EE Standalone\3.8.<x>.<y>` folder.

Examples

`Setup.exe /silent /EULAAccepted /SafetyHintsAccepted`

Triggers a silent installation.

```
"C:\Program Files (x86)\ETAS\GENericSetup\EE EE Standalone\
 3.8.<x>.<y>\setup.exe" /uninstall /Debug
```

Triggers a non-silent uninstallation and writes additional logs.

`setup.exe /DefaultSettings:"D:\myOwnSettings.xml"`

Triggers a non-silent installation that uses your own default settings for the installation.

1) `<programs>` = `Program Files (x86)` (64 bit OS) or `Program Files` (32 bit OS)

3.4 Setting the Licensing Behavior

The INTECRIO installation includes an installation of the ETAS license manager. You can define the way in which INTECRIO (and other ETAS software programs) access the required licenses in the `[Licensing]` section of an `*.ini` file.

Your installation disk contains two such `*.ini` files, one for INTECRIO, the other for the ETAS Experiment Environment. Both are named `Licensing.ini`.

The location of *Licensing.ini for INTECRIO* is given in the file `InstallationDefaultSettings.xml`, package `INTECRIO License`, variable `LIMA_INIFILE`.

Licensing.ini for the ETAS Experiment Environment can be found in the `ExperimentEnvironment\Packages\EE License_3.8.4.<x>.<y>` folder.

General Procedure

The procedure to edit a `Licensing.ini` file and start the installation depends on the original location of the installation files.

The installation files are placed in a directory where you cannot edit them: Use one of the following procedures.

- Copy *all* installation files to a directory where you can edit the files.

  There, edit `Licensing.ini` (cf. Page 34), then start the installation via double-click on `Autostart.exe` or `setup.exe` (cf. sections 3.2.1 and 3.3.2) or via command line (cf. sections 3.2.4 and 3.3.3).
- Copy only a `Licensing.ini` file to a directory where you can edit the file.

  There, edit `Licensing.ini` (cf. Page 34). Create your own XML file with default settings (cf. Page 33) and enter the location of your edited `Licensing.ini` file in the `LIMA_INIFILE` variable:

  ```
  <Variable
       name="LIMA_INIFILE"
       defaultValue="<path>\Licensing.ini"
       validation="fileExists" />
  ```

  Start the installation via command line (cf. sections 3.2.4 and 3.3.3) and use `/DefaultSettings:"<path>\<filename>.xml"` to add your own XML file and with that your `License.ini` file.

The installation files are placed in a directory where you can edit them: Edit the desired `Licensing.ini` file (cf. Page 34) and start the installation.

Editing the `*.ini` File

To define the access to the required licenses

1. Open the `*.ini` file with a text editor.
2. Go to the `[Licensing]` section and modify the settings as desired.

   The parameters that can be included in this section, and their settings are described below.
3. Save your changes.

The following parameters may be used:

- `LicenseFileName`

  Defines the absolute path and file name of the license file which is to be added.

- `LicensesToBorrow`

  You can use this setting if licenses can be borrowed from a license server. To enable the borrow mechanism, you must enter the name of the product or features license (e.g., `INTECRIO`). If you enter more than one license, the license names must be separated by blanks.

  INTECRIO and the ETAS Experiment Environment use the following licenses:

License name	Functionality
`INTECRIO`	INTECRIO prototyping environment[a)]
`INT_ASC`	ASCET connectivity[b)]
`INT_ECC`	MATLAB and Simulink Embedded Coder connectivity[a]
`INT_RTA-RTE2.0`	AUTOSAR-RTE integration for prototyping[c)]
`INT_SLC`	MATLAB and Simulink connectivity[a]
`INT_TC_RTT`	RTA-Trace Connectivity[a,d)]
`INT_UCC_ES900`	ES900 connectivity[e)]
`INT_UCC_PC`	VP-PC connectivity[f)]
`INCA-VIP`	ETAS virtual OS execution platform[e]
`LD_RTA-OS_VRTA_SC34`	RTA-OSEK variant used for virtual prototyping[e]
`EE`	ETAS Experiment Environment
`INT_RP_EE`	Rapid prototyping with INTECRIO and ETAS Experiment Environment
`INT_VP_EE`	Virtual prototyping with INTECRIO and ETAS Experiment Environment
`INT_UCC_LCO_FIL`, `LCS_LCO_CCI`, `LCS_LCO_LCE`, `LCS_LCO_LCX`	Irrelevant for the combination of ETAS Experiment Environment and INTECRIO

  a) shipped with INTECRIO integration platform
  b) shipped with ASCET
  c) RTA-RTE (shipped with INTECRIO integration platform)
  d) RTA-TRACE is discontinued. This license is relevant *only* for existing installations.
  e) part of INTECRIO-RP (rapid prototyping)
  f) part of INTECRIO-VP (virtual prototyping)

- `BorrowExpiryMode`

  Defines the way in which the expiration of the borrow status is given. Possible values are:

  - `Date`

    If `BorrowExpiryMode` is set to `Date`, the borrow period will expire at a certain date which is specified under `BorrowExpiryDate`.

- `Interval`

  If `BorrowExpiryMode` is set to `Interval`, the borrow period will expire after a certain number of days which is specified under `BorrowExpiryInterval`.

– `BorrowExpiryDate`

  If `BorrowExpiryMode` is set to `Date`, the parameter **`BorrowExpiryDate`** specifies the date when the borrow period expires. The format is `yyyy-mm-dd`.

– `BorrowExpiryInterval`

  If `BorrowExpiryMode` is set to `Interval`, the parameter **`BorrowExpiryInterval`** specifies the length of the borrow period in days.

– `ExecuteBorrowAutomaticExtensionInterval`

  Defines at what point of time the borrow period will be automatically extended. This parameter specifies the number of days before the expiration of the current borrow period. When this time is reached, the borrow period is automatically extended to the interval specified under `BorrowAutomaticExtensionInterval`.

– `BorrowAutomaticExtensionInterval`

  This parameter specifies the borrow interval in days that is applied when an automatic extension of the borrow period is executed (as defined under `ExecuteBorrowAutomaticExtensionInterval`).

– `ImmediateBorrow`

  You can define that a license is automatically borrowed. Possible values are:

  - `True`

    The license is borrowed automatically at installation time.

    > NOTE
    >
    > `ImmediateBorrow='True'` works *only* for the user who performs the installation. Other users who work on the same computer do not own the borrowed license.

  - `False`

    The license will be borrowed at the first time when the program connects to the license server.

– `CustomLicenseFolder`

  Due to the fact that the default location for added license files (e.g., `C:\ProgramData\ETAS\FlexNet`) is only writable for users with admin rights, a different path for the license file folder may be specified with this parameter.

The following example defines that borrowing is enabled for INTECRIO. The license will be borrowed when INTECRIO is started for the first time; by default the license expires after 100 days.

```
[Licensing]
LicenseFileName = 'd:\licenses\MyLicense.lic'
LicensesToBorrow = 'INTECRIO'
BorrowExpiryMode = 'Interval'
```

```
BorrowExpiryInterval = '100'
ImmediateBorrow = 'false'
```

3.5 Licensing the Software

A valid license is required for using INTECRIO. You can obtain the license file required for licensing either from your tool coordinator or through a self service portal on the ETAS Internet Site under https://www.etas.com/support/licensing. To request the license file you have to enter the activation number which you received from ETAS during the ordering process.

In the Windows Start menu, go to the app list and select **E > ETAS > ETAS License Manager**.

Follow the instructions given in the dialog. For further information about, for example, the ETAS license models and borrowing a license, press <F1> in the ETAS License Manager.

3.6 Uninstalling

If you uninstall INTECRIO, all installed components and add-ons are uninstalled automatically, except for the target server and the ETAS Experiment Environment.

Use one of the following ways to start the INTECRIO or ETAS Experiment Environment uninstall process:

A **Programs and Features** in the Windows Control Panel

B the `/Uninstall` command-line parameter (Page 26 and Page 33)

To uninstall INTECRIO or the ETAS Experiment Environment

1. Start the uninstall procedure.

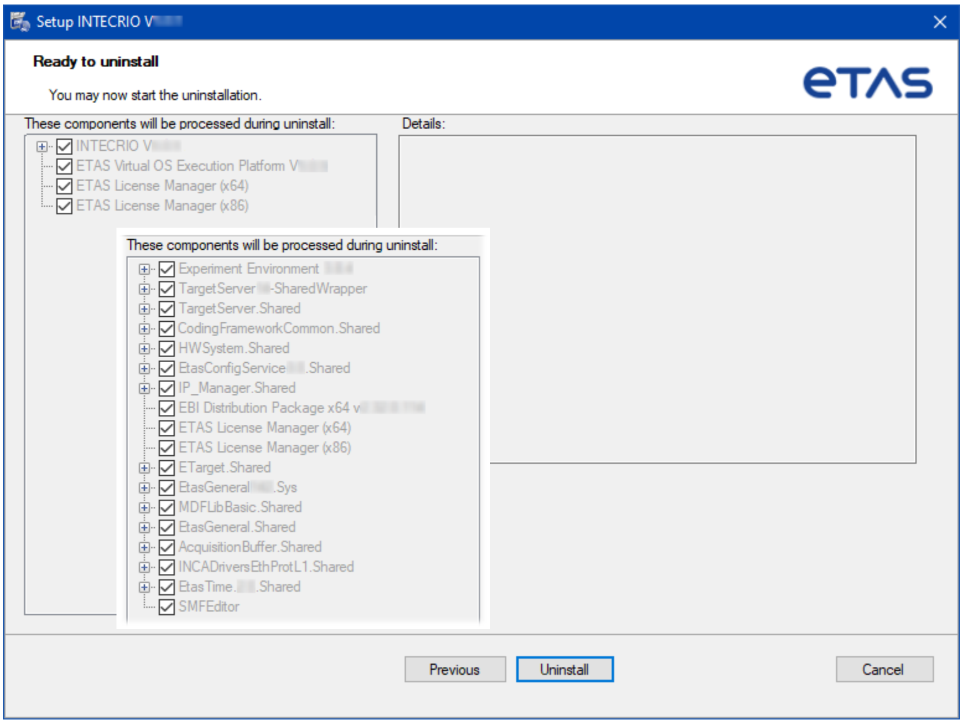
   The ETAS installer is launched.

   You can use the **Next** button to get to the next uninstallation window; **Back** to get to the previous window, and **Cancel** to cancel uninstallation.

2. Click **Next** to check if your computer meets the system requirements for uninstallation of the ETAS Experiment Environment.

   The result is displayed in the "System Check" window.

   If the system requirements for uninstallation are met, the "Ready to uninstall" window opens. It lists the components that will be uninstalled. You cannot change the selection.



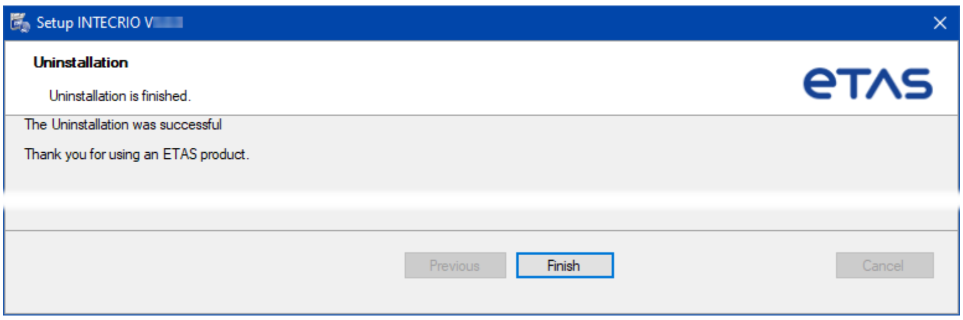
   The ETAS Virtual OS Execution Platform and the License Manager are only uninstalled if no other program that uses them is available on your computer.

> **NOTE**
>
> The next step will start the uninstallation. You **cannot** cancel uninstallation once it is running; the **Cancel** button is deactivated.

3. Click **Uninstall** to actually uninstall the ETAS Experiment Environment.

   A progress indicator shows how the uninstallation is progressing.

   Once all components have been uninstalled, a success window opens.



4. Click **Finish** to end the uninstallation.

To uninstall the target server

NOTICE
If you uninstall only the target server, downloads to the target are no longer possible. INTECRIO and the ETAS Experiment Environment can be started, but they produce error messages.

1. Use **Programs and Features** icon in the Windows Control Panel to start the uninstallation of `ETAS Target Access - `*<x>*`.`*<y>*.
2. If a safety inquiry opens, confirm with **Yes**.

   The ETAS Installer is launched.

   You can use the **Next** button to get to the next uninstallation window; **Back** to get to the previous window, and **Cancel** to cancel uninstallation.
3. Select your preferred setup language, then click **Next**.

   The installer checks if your computer meets the system requirements for uninstallation of the ETAS Target Server.

   The result is displayed in the "System Check" window.

   If the system requirements for uninstallation are met, the "Ready to uninstall" window opens. It lists the components that will be uninstalled. You cannot change the selection.

NOTE
The next step will start the uninstallation. You **cannot** cancel uninstallation once it is running; the **Cancel** button is deactivated.

4. Click **Uninstall** to actually uninstall the ETAS Target Server.

   A progress indicator shows how the uninstallation is progressing.

   Once all components have been uninstalled, a success window opens.
5. Click **Finish** to end the uninstallation.

4 INTECRIO Quick Guide

4.1 Introduction

This chapter is intended to provide a quick guide to the most important features of INTECRIO V5.0. The examples, displayed in the form of flowcharts, provide you with an overview of the program functionality and operating mode. In these flowcharts, dashed black arrows lead from an action to its result. Solid red arrows mark locations where actions shall take place.

Below, you find a short explanation of the most important abbreviations used here.

- **BMT** – Behavioral Modeling Tool
- **EE** – ETAS Experiment Environment
- **ES** – Environment system
- **ISR** – Interrupt service routine
- **HBB** – Hook-based bypass
- **HC** – Hardware Configurator
- **HW** – Hardware
- **HWS** – Hardware System
- **PC** – Project Configurator
- **PI** – Project Integrator
- **OSC** – OS Configurator
- **RE** – Runnable entity
- **RP** – Rapid prototyping
- **SBB** – Service-based bypass
- **SP** – System Project
- **SWC** – legacy AUTOSAR module (i.e. an *AUTOSAR **s**oftware **c**omponent* that has been imported into an INTECRIO workspace)
- **SWS** – Software System
- **VP** – Virtual prototyping
- **WS** – Workspace

4.2 Creating a New Workspace

The main task of the workspace (WS) is to systematically store the data arising from your work with INTECRIO (modules, functions, hardware and software systems, system projects and experiments) in a database and manage this data in a clear interface. As in Windows Explorer, you can create, move, copy, import and export directories and objects but also create completely new workspaces.

The following data is managed in the workspace:

- functions
- hardware and software systems
- system projects

The following data is **not** managed in the workspace:

- `*.c`, `*.h` and `*.a2l` files

- `*.a2l.cod` files
- measure files (`*.dat`)

INTECRIO offers the possibility to work with several workspaces. This makes it possible to keep the quantity of data small and clear. The performance of the system can deteriorate with extremely large quantities of data.

When you start work, it is best to create a new workspace. This makes it easier to assign data to specific projects later. The advantage of a new workspace is that you only create the entries you really need.

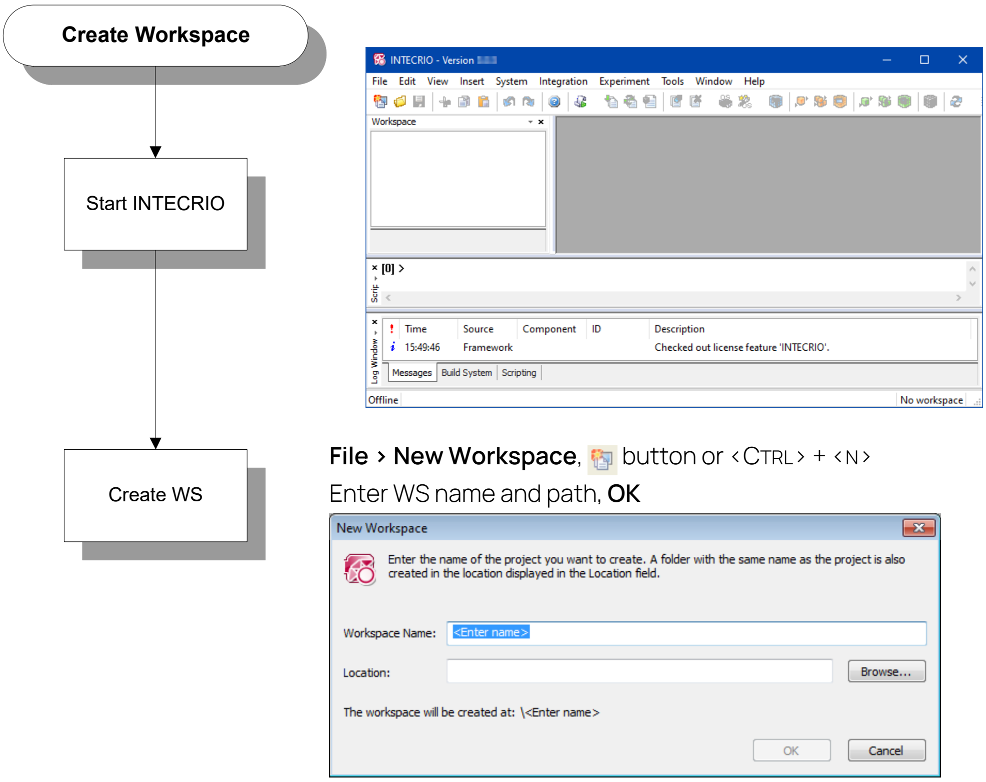


Fig. 4-1 Creating a Workspace

4.3 Preparing and Importing Modules

When working with INTECRIO you need functioning software modules, which can have been created using different BMTs. To be used by INTECRIO, they have to be available as suitable description files (`*.c`, `*.a2l`, `*.six`).

The `*\Modules` folders are the folders for modules in the workspace.

4.3.1 Preparing Modules

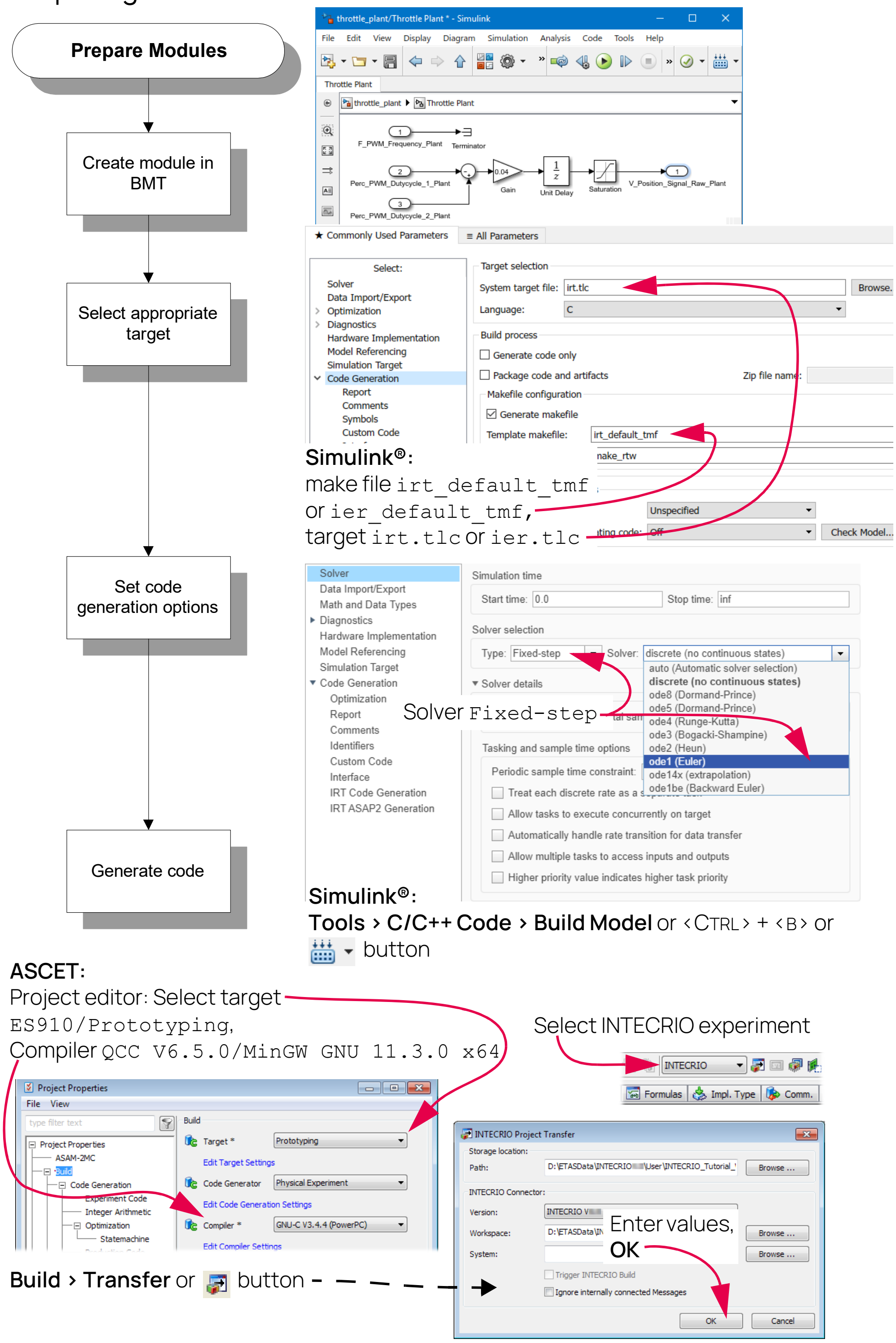


Fig. 4-2 Preparing a Module

4.3.2 Importing Modules

Only the interface description file is read in when you import modules (`*.six`; see chapter "SCOOP and SCOOP-IX" in the user guide).

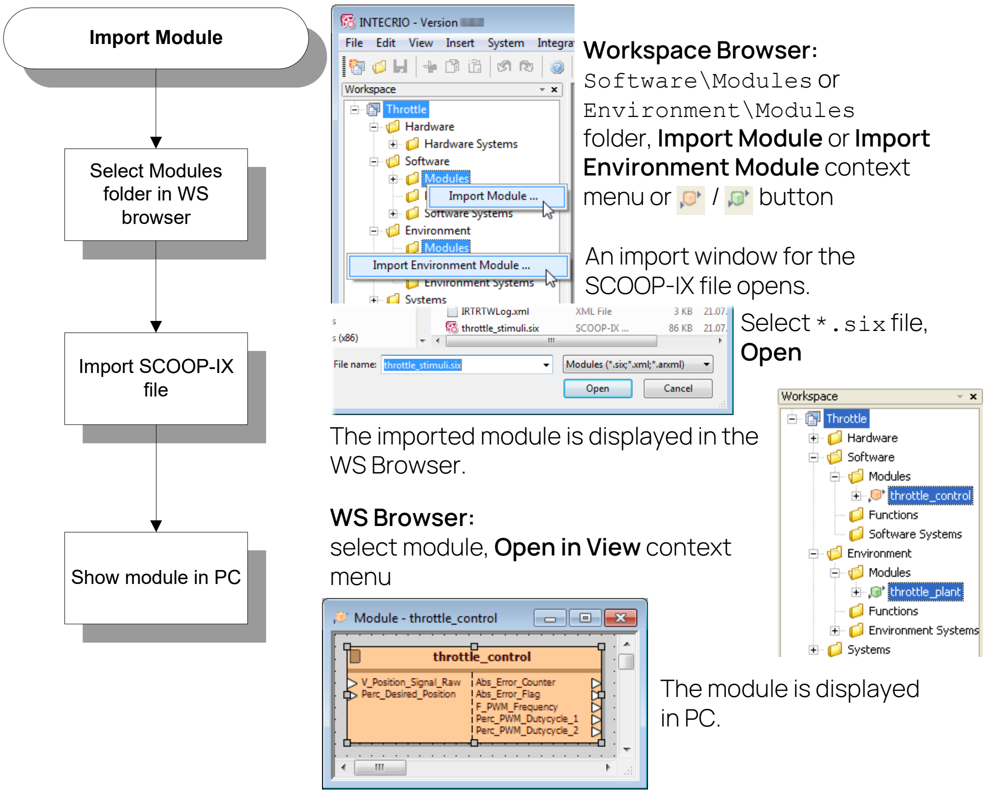


Fig. 4-3 Importing Modules

> **NOTE**
>
> In INTECRIO V5.0.4, AUTOSAR support is limited to using existing workspaces that contain legacy AUTOSAR modules. It is not possible to import new AUTOSAR software components.

4.4 Creating a Function

Modules and legacy SWC can be grouped as functions to improve clarity or to make reuse simple. These functions are structure objects without their own functionality; they can contain only modules, only SWC, or both modules and SWC. The `Software\Functions` folder is the folder for functions in the workspace.

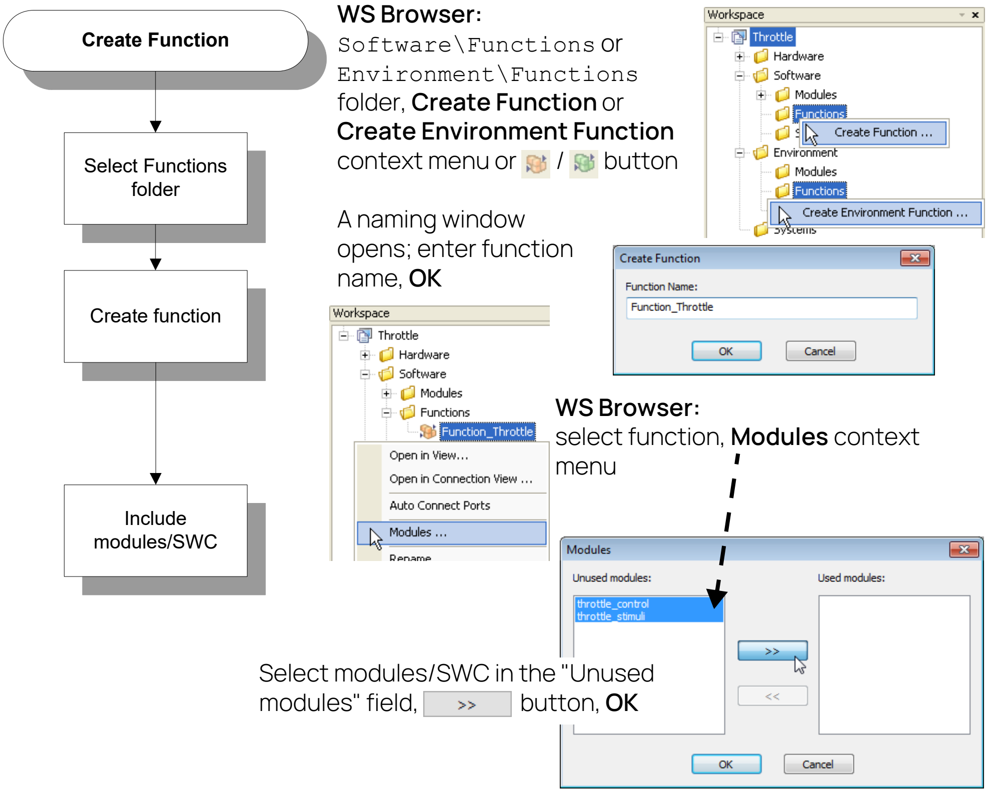


Fig. 4-4 Creating a Function

The connections between the modules or SWC in a function, as well as the ports for the connection with the outside world, have to be specified manually.

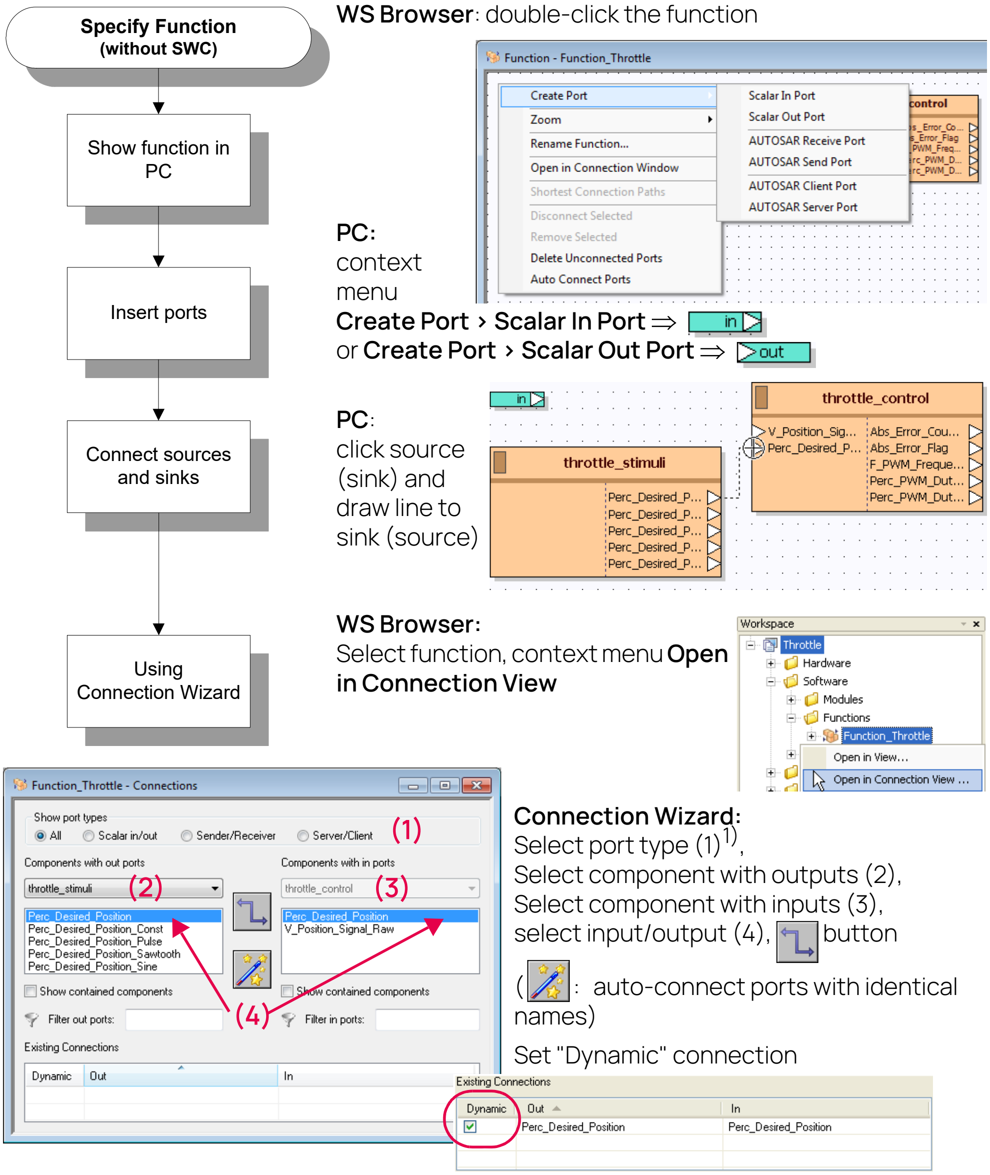


Fig. 4-5 Specifying a Function without SWC

1) **Scalar in/out**: modules; **Sender/Receiver** / **Server/Client**: AUTOSAR SWC

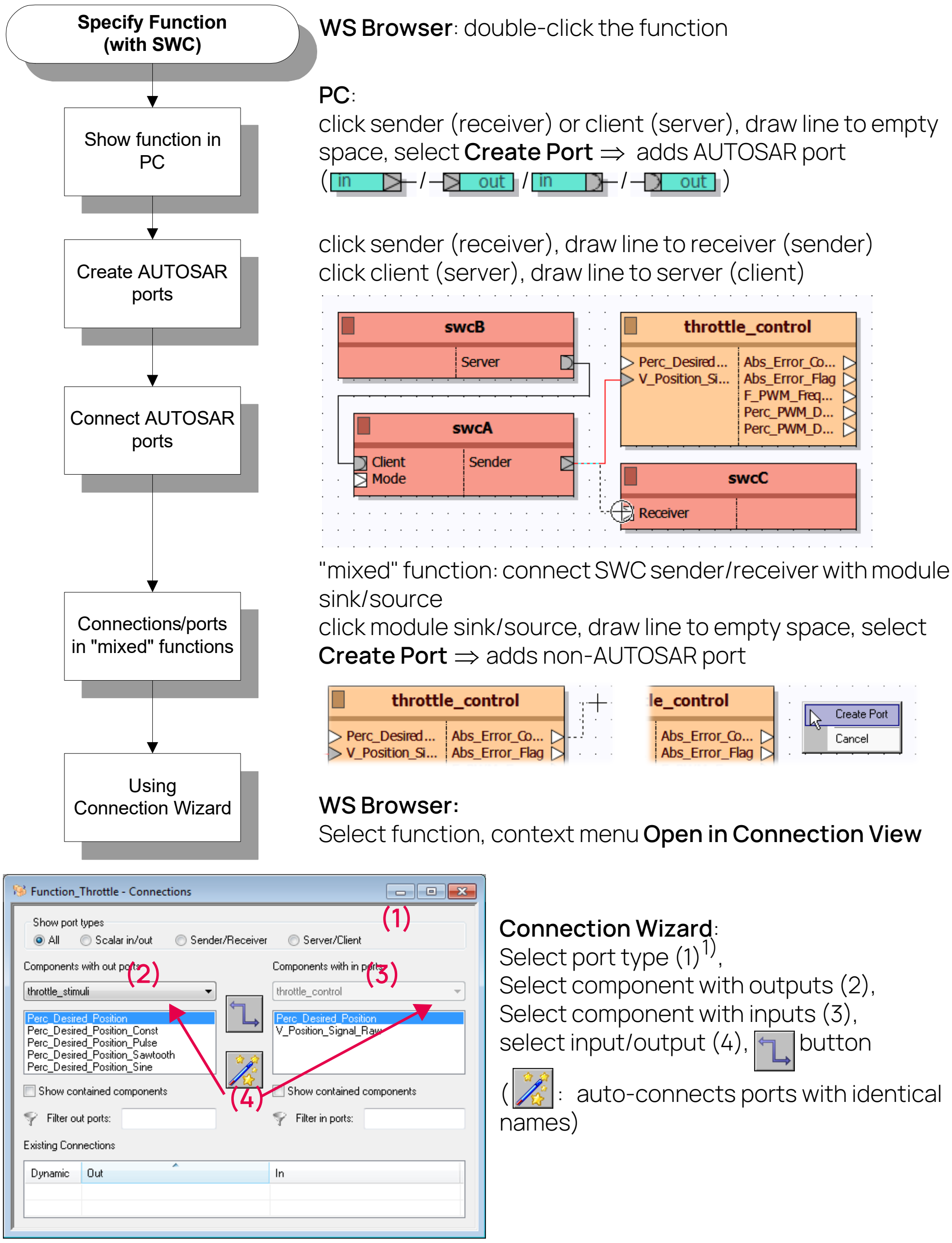


Fig. 4-6 Specifying a Function with SWC

4.5 Creating a Software System or Environment System

Modules, legacy SWC, and functions are inserted into the *software system* (SWS) or *environment system* (ES) as references. Each module or SWC can be incorporated exactly once into a software system, regardless of whether directly or as part of a function. The signal sources and sinks of the functions, modules and SWC are connected to each other; interfaces are created for the connection to the hardware.

---

1) **Scalar in/out**: modules; **Sender/Receiver** / **Server/Client**: AUTOSAR SWC

The `Software\Software Systems` folder is the folder for software systems in the workspace, and `Environment\Environment Systems` is the folder for environment systems.

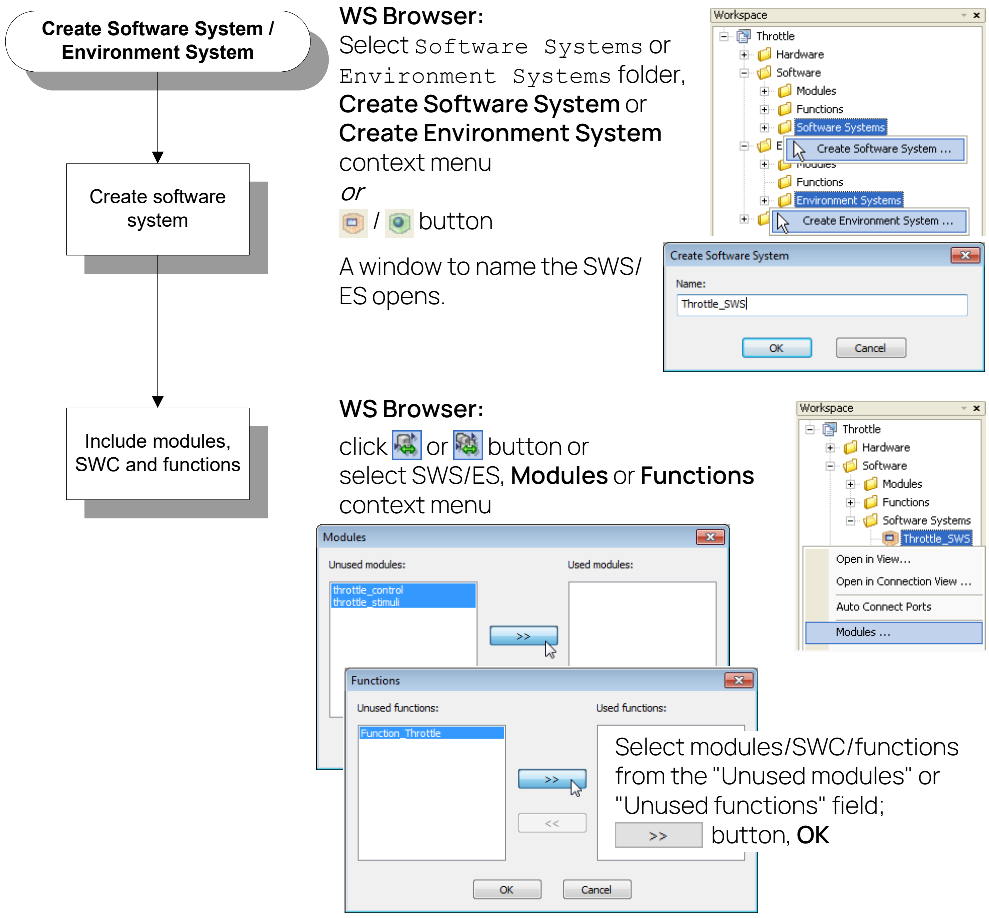


Fig. 4-7 Creating a Software System or Environment System

The connections between the modules, SWC and functions in an SWS, as well as the ports for the connection with the hardware, have to be specified manually in the project configurator (PC).

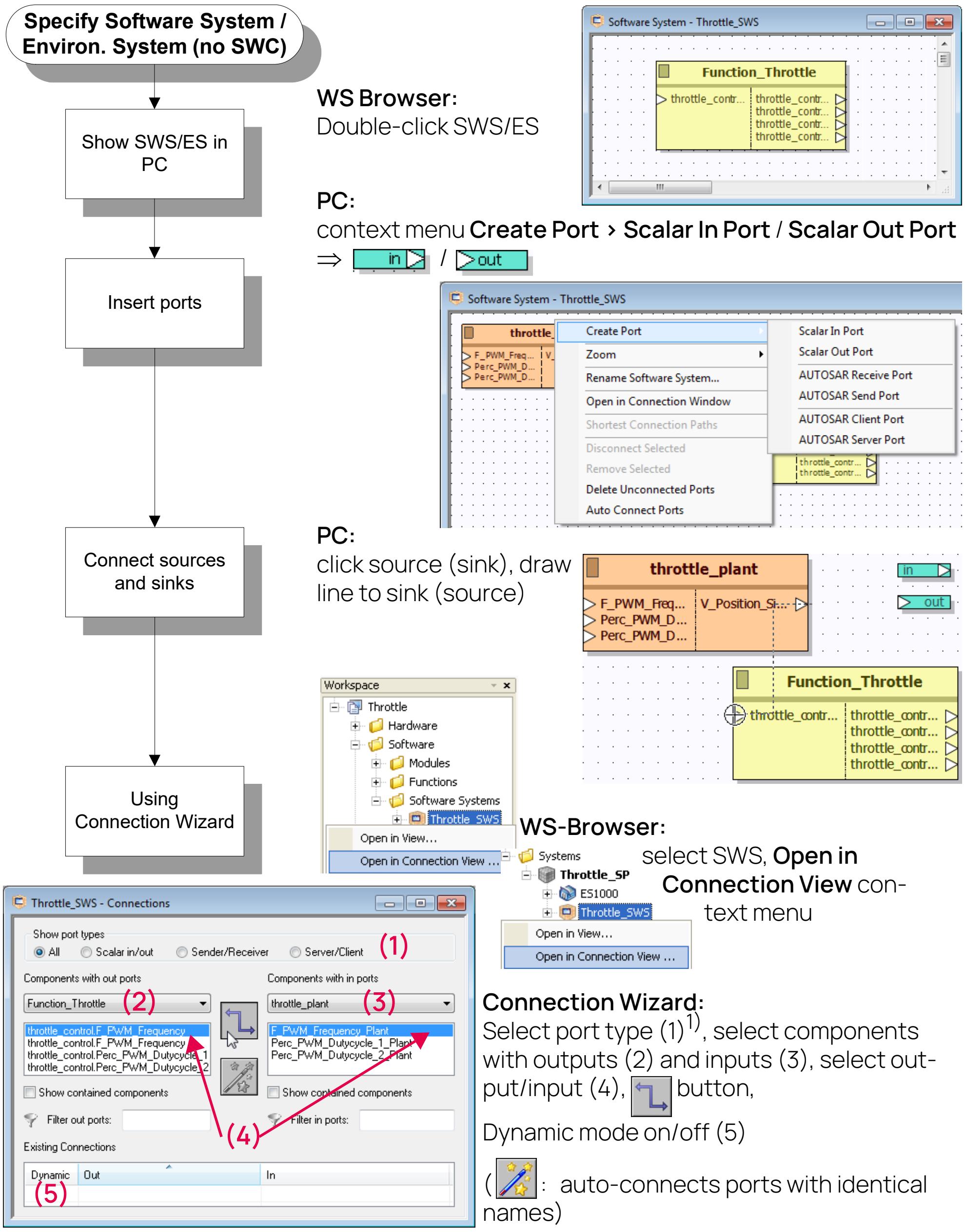
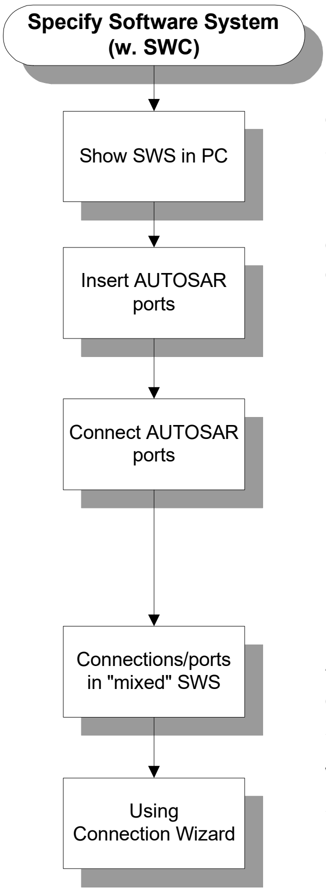


Fig. 4-8 Specifying a Software System/Environment System without SWC

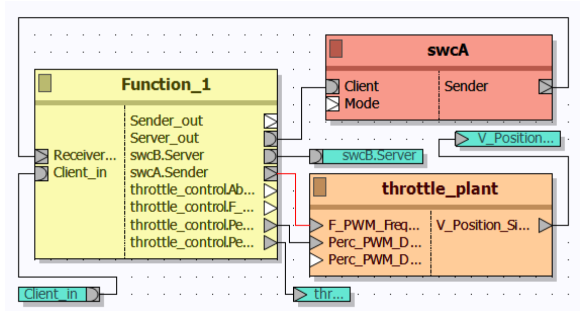
1) **Scalar in/out**: modules; **Sender/Receiver** / **Server/Client**: AUTOSAR SWC



WS Browser: Double-click SWS

PC:
click sender (receiver) or client (server), draw line to empty space, select **Create Port** ⇒ adds AUTOSAR port (in / out / in / out)

click sender (receiver), draw line to receiver (sender)
click client (server), draw line to server (client)



"mixed" SWS: connect SWC sender/receiver with non-AUTOSAR sink/source; connect non-AUTOSAR ports

click non-AUTOSAR source/sink, draw line to empty space, select **Create Port** ⇒ adds non-AUTOSAR port

WS-Browser:
select SWS, **Open in Connection View** context menu

Connection Wizard:
Select port type (1)[1], select components with outputs (2) and inputs (3), select output/input (4), button

(: auto-connects ports with identical names)

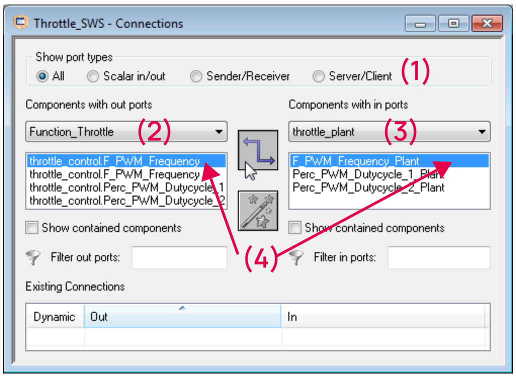


Fig. 4-9 Specifying a Software System with SWC

4.6 Creating and Configuring a Hardware System

Use the Hardware Configurator to create and configure the hardware system you want to use in the experiment. The "Insert Hardware System" or "Insert New Item" window contains the elements available on every hierarchical level.

You can make various settings (e.g. module parameters, channel settings for measure hardware etc.) for every hardware component using the relevant dialogs.

The `Hardware` folder is the folder for hardware systems (HWS) in the workspace.

---

1) **Scalar in/out**: modules; **Sender/Receiver** / **Server/Client**: AUTOSAR SWC

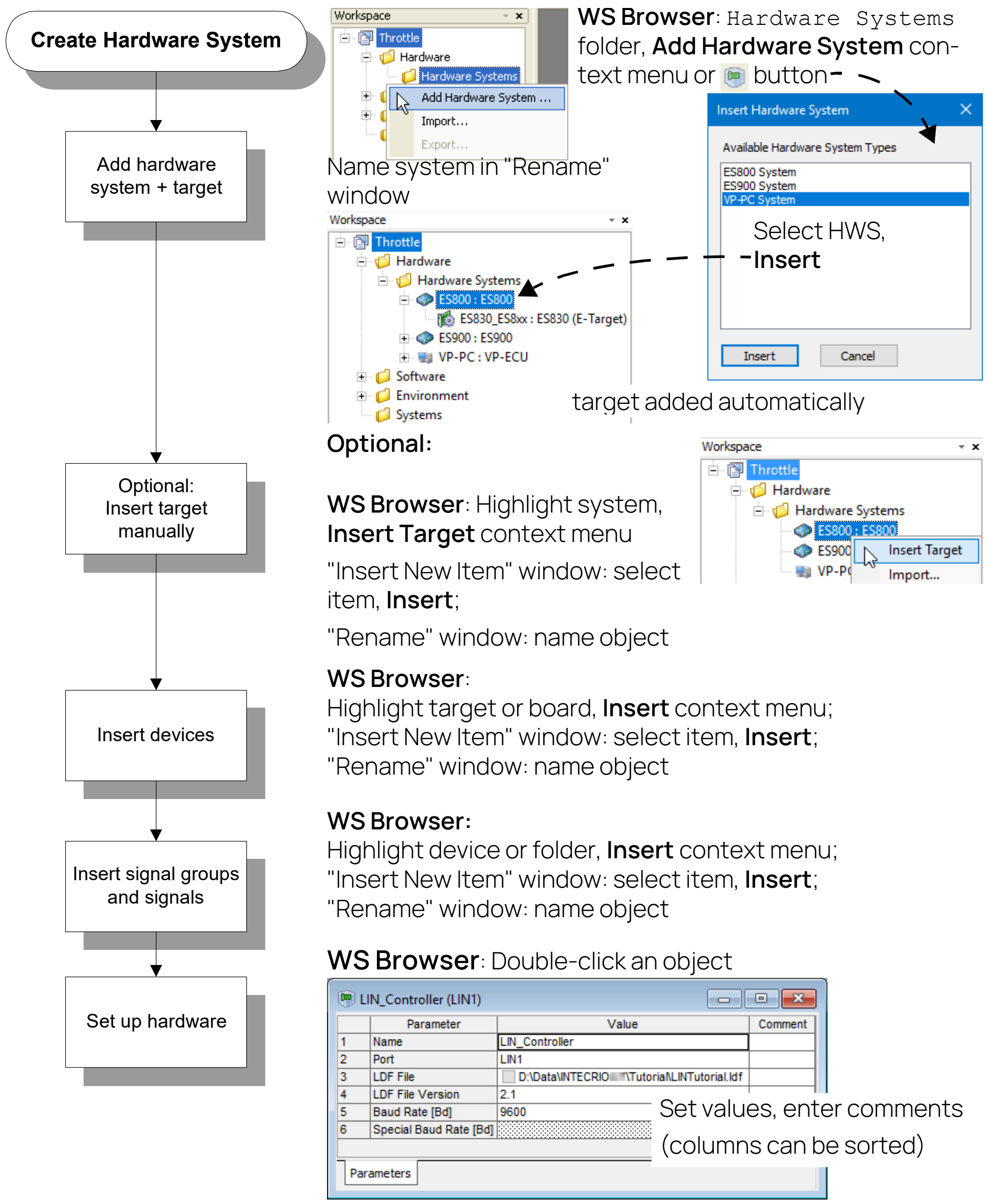


Fig. 4-10 Creating and Configuring a Hardware System

4.6.1 Importing a Hardware System

As an alternative to manual creation and setup of a HWS, you can import a hardware description, `*.hwx`, created with INTECRIO, INTECRIO-RLINK or ASCET-RP into a new, or an existing, hardware system.

The file extension `*.hwx` is used for two description formats (HWX1 and HWX2).

> **NOTE**
>
> Since INTECRIOV4.7.3, the HWX1 format can no longer be imported.

After import file selection, HWX2 import is aided by a wizard (see Fig. 4-11).

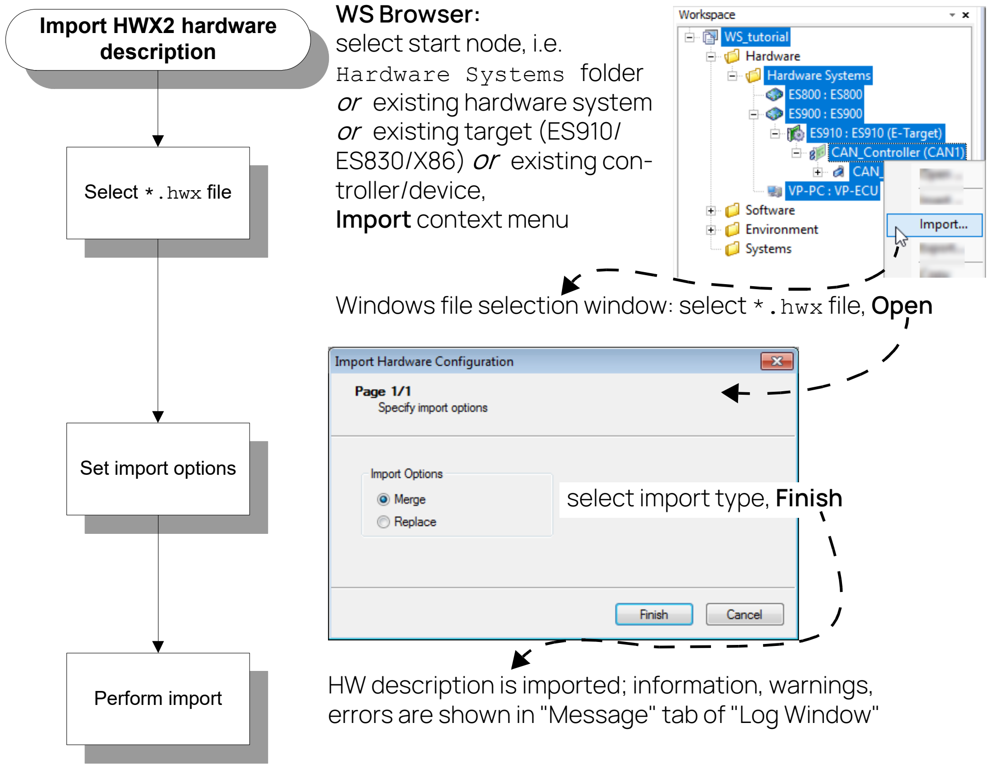


Fig. 4-11 Importing a HWX2 hardware description (*.hwx file)

4.6.2 Configuring a Daisychain

The configuration of a daisychain follows a different procedure.

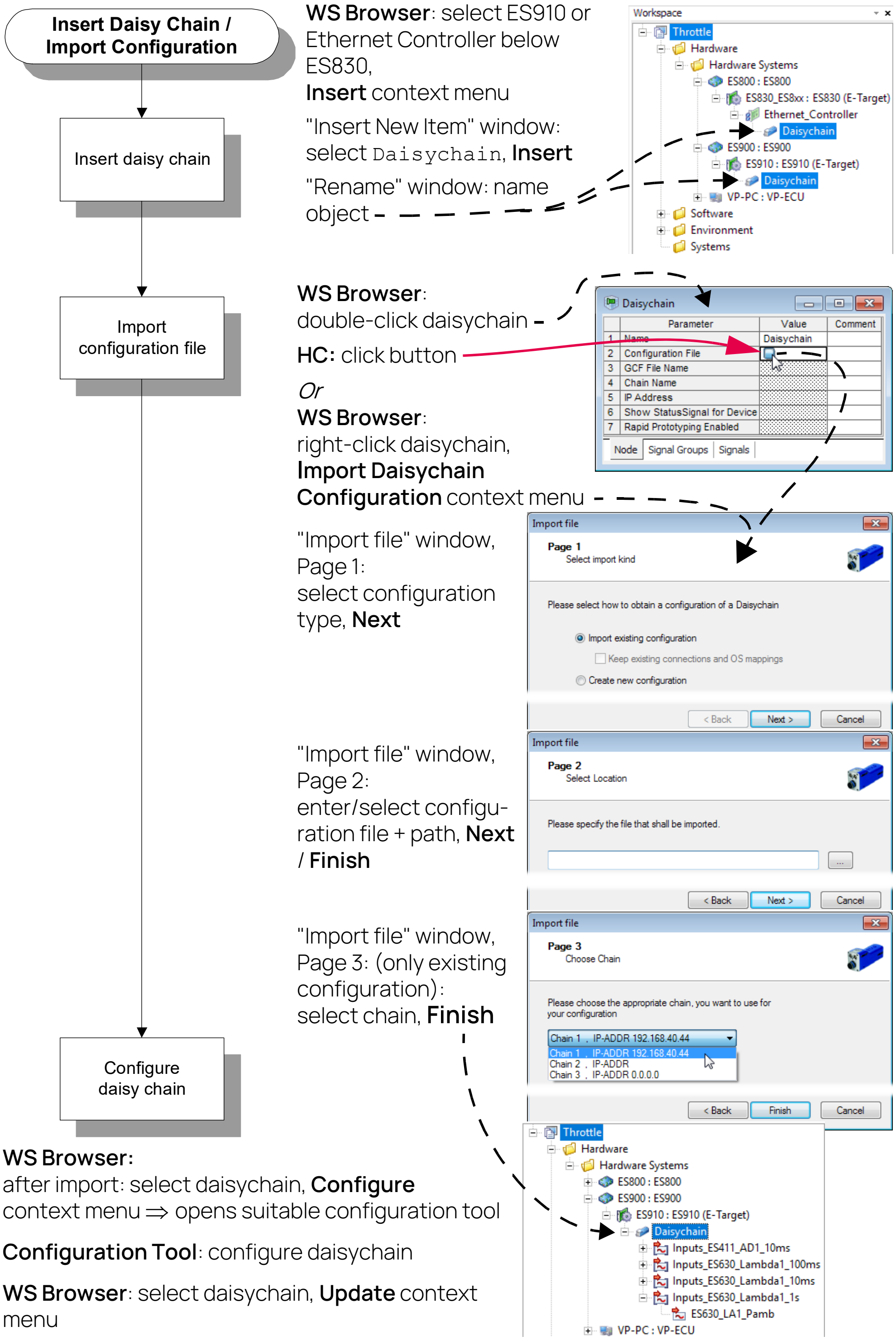


Fig. 4-12 Creating and Configuring a Daisychain

4.6.3 Configuring a LIN Controller

The configuration of a LIN controller can be imported.

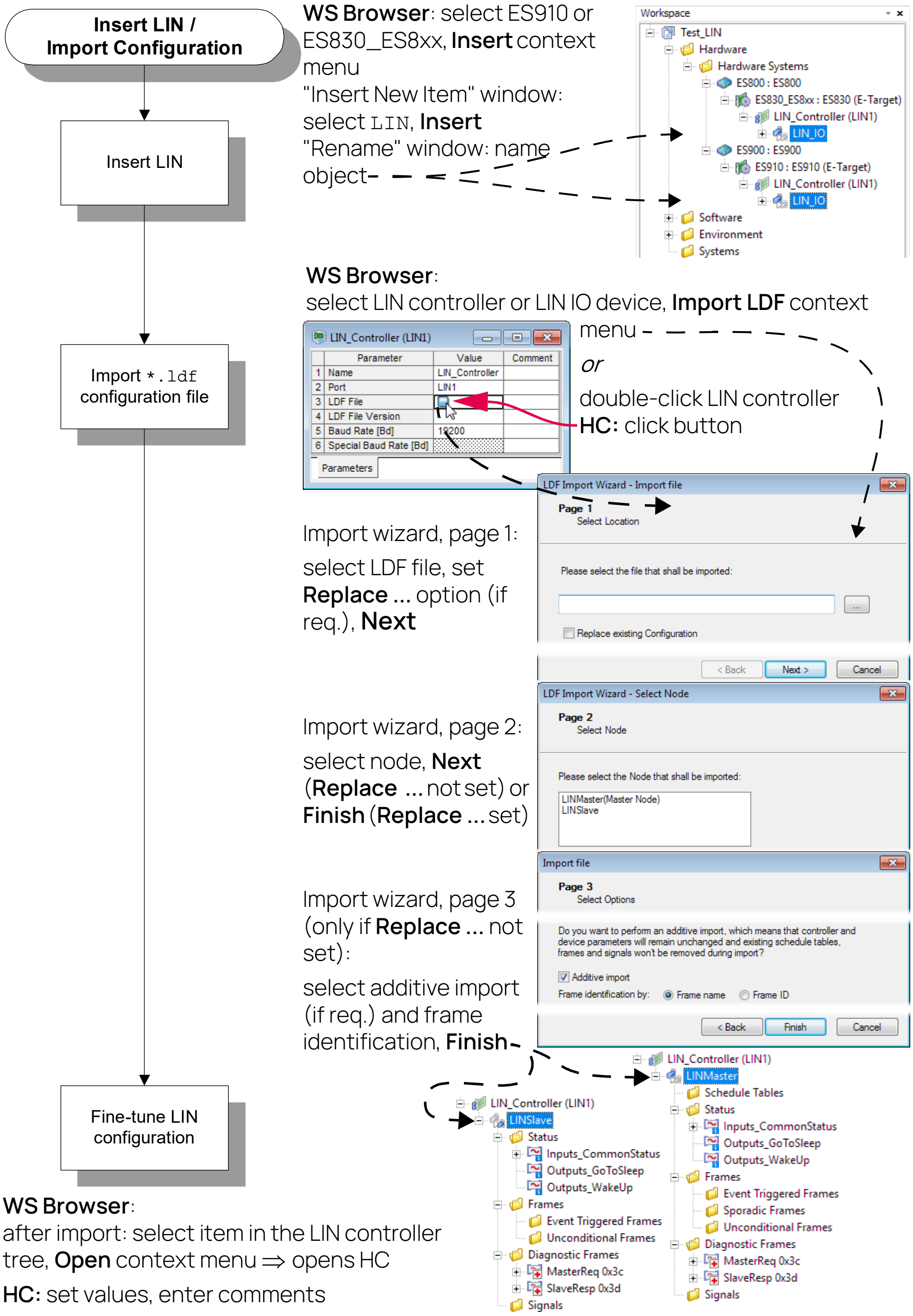


Fig. 4-13 Creating and Configuring a LIN Controller

LIN nodes (devices) can be of type Master or Slave. You can switch the node type; type-dependent settings are adjusted automatically.

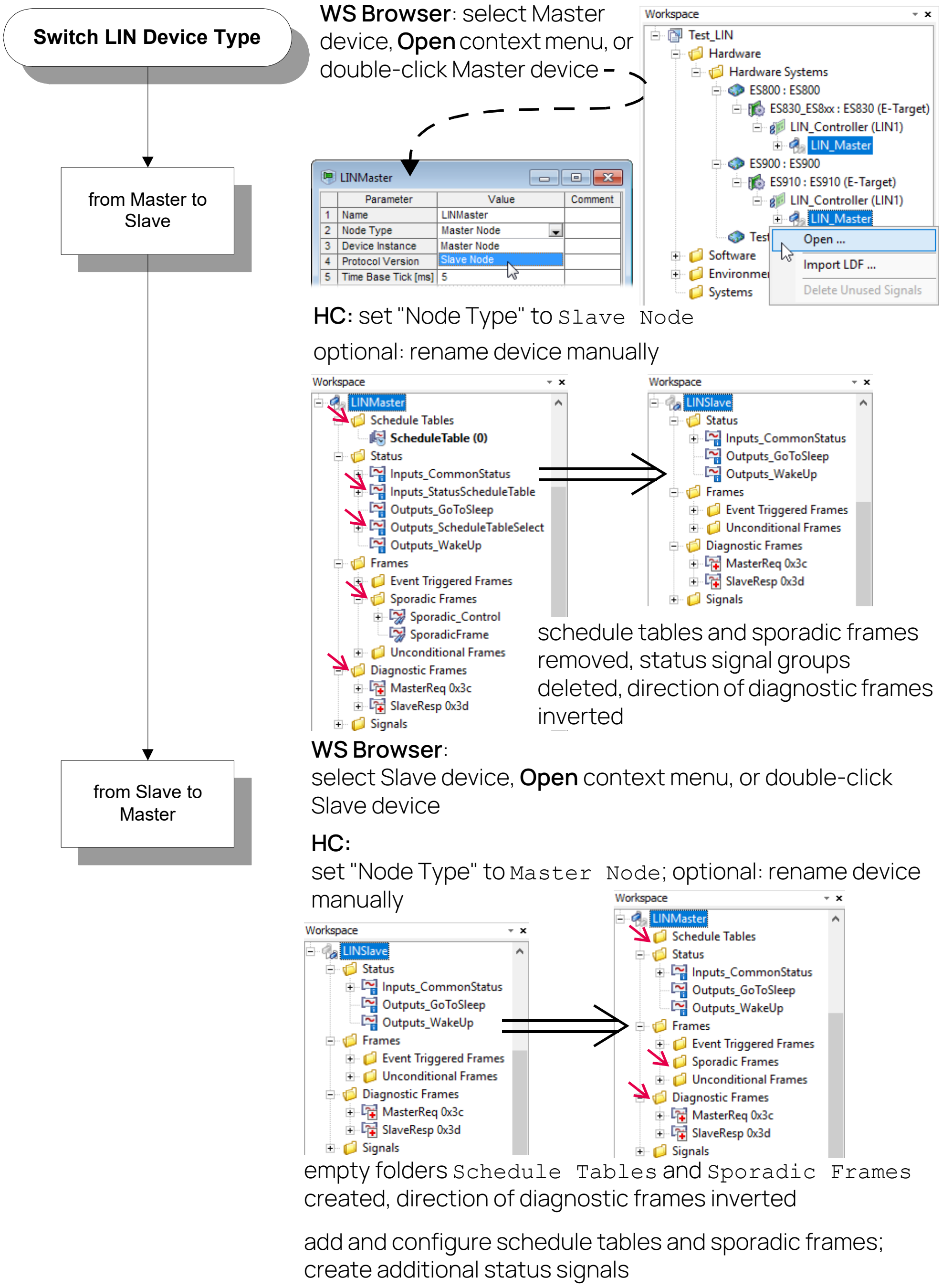


Fig. 4-14 Switching the LIN Node Type

4.6.4 Configuring a Bypass

> **NOTE**
>
> Service-based bypass on an ETK with 8 Mbit/s is not supported.
>
> Starting with INTECRIO V5.0.1, FETK bypass on ES910 is deprecated.

After the bypass device is added (see Fig. 4-10), a bypass is configured as follows.

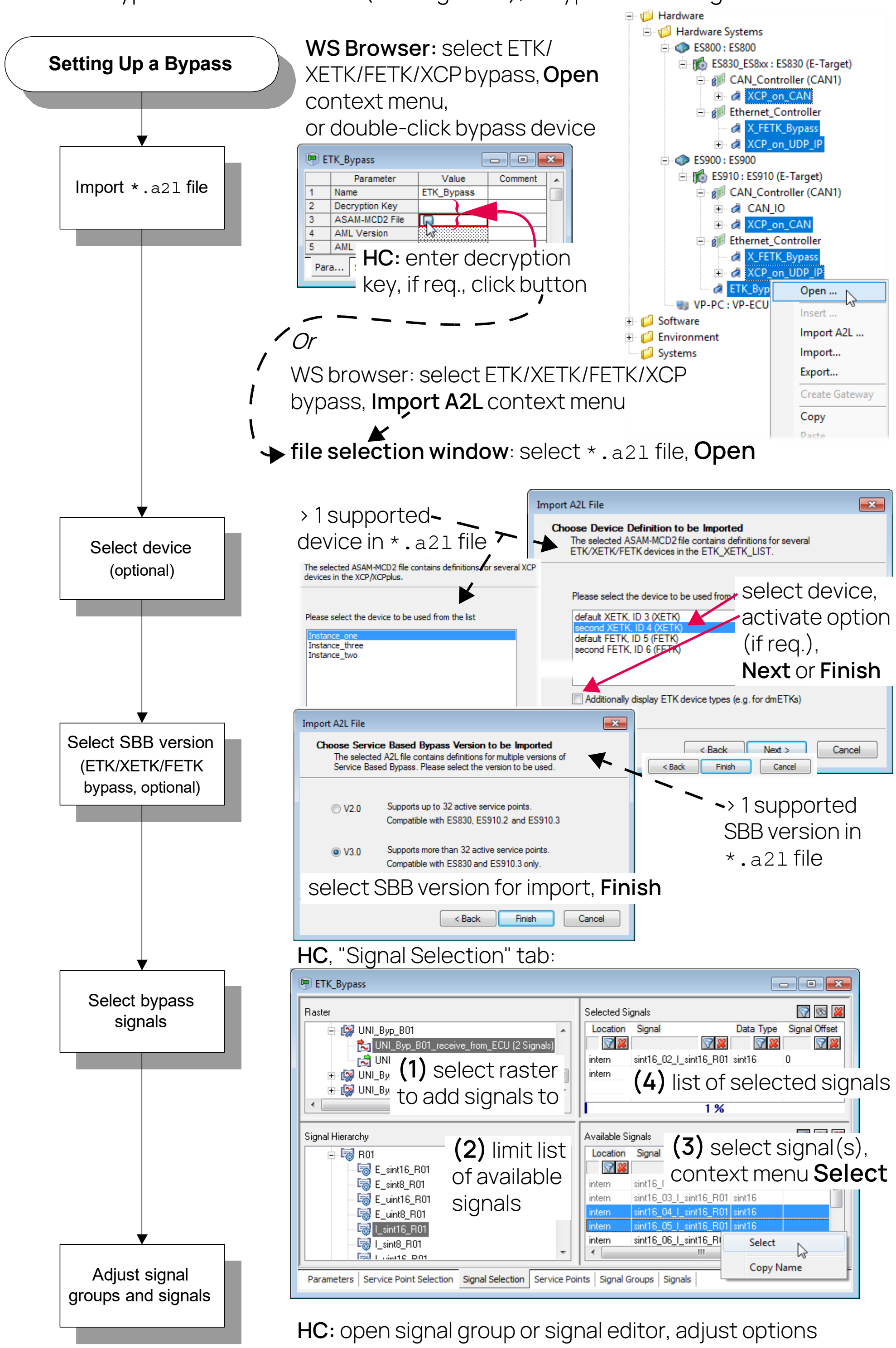


Fig. 4-15 Setting up a Bypass Device

The service point descriptions for a service-based bypass are provided in the `*.a2l` file. The user selects the actually needed service points and signals after importing the `*.a2l` file.

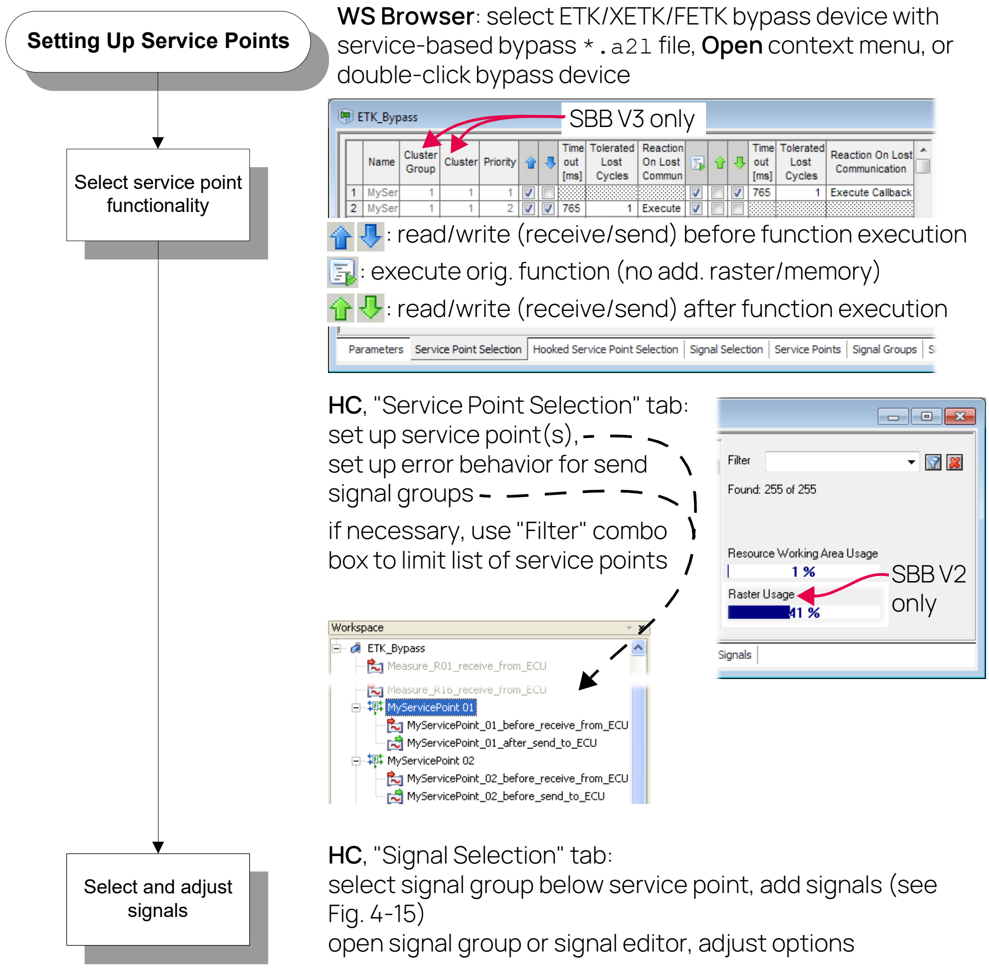


Fig. 4-16 Setting up Service Points (Service-Based Bypass)

> **NOTE**
>
> Service-based bypass on an ETK with 8 Mbit/s is not supported.

In connection with Distab17, the concept of service point configuration extends to the hook-based bypass. You have to set up hooked service points before you can select the signals.

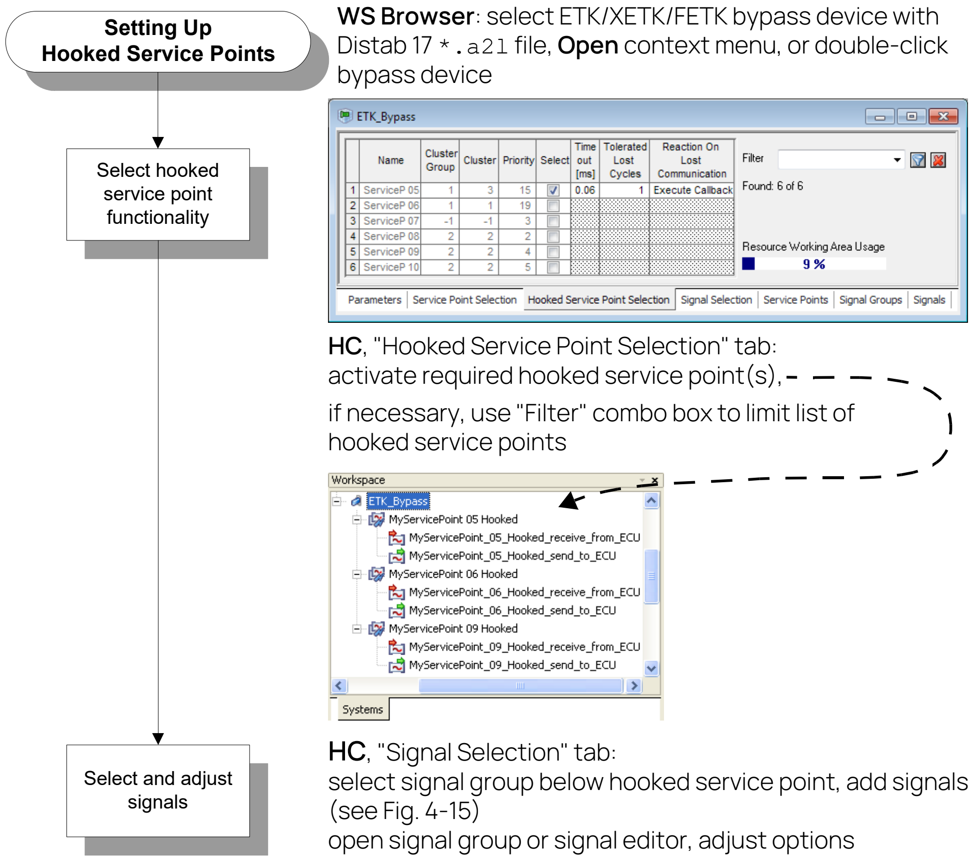


Fig. 4-17 Setting up Hooked Service Points (Hook-Based Bypass + Distab 17)

4.6.5 Importing a CAN Configuration File

After a CAN device is added (see Fig. 4-10), you can create signals and frames manually, or you can import a CAN configuration file, i.e. either a CAN database (`*.dbc`) or an AUTOSAR-based file (`*.arxml`). Both classic CAN and CAN FD (ES910+ES922 and ES830) are supported.

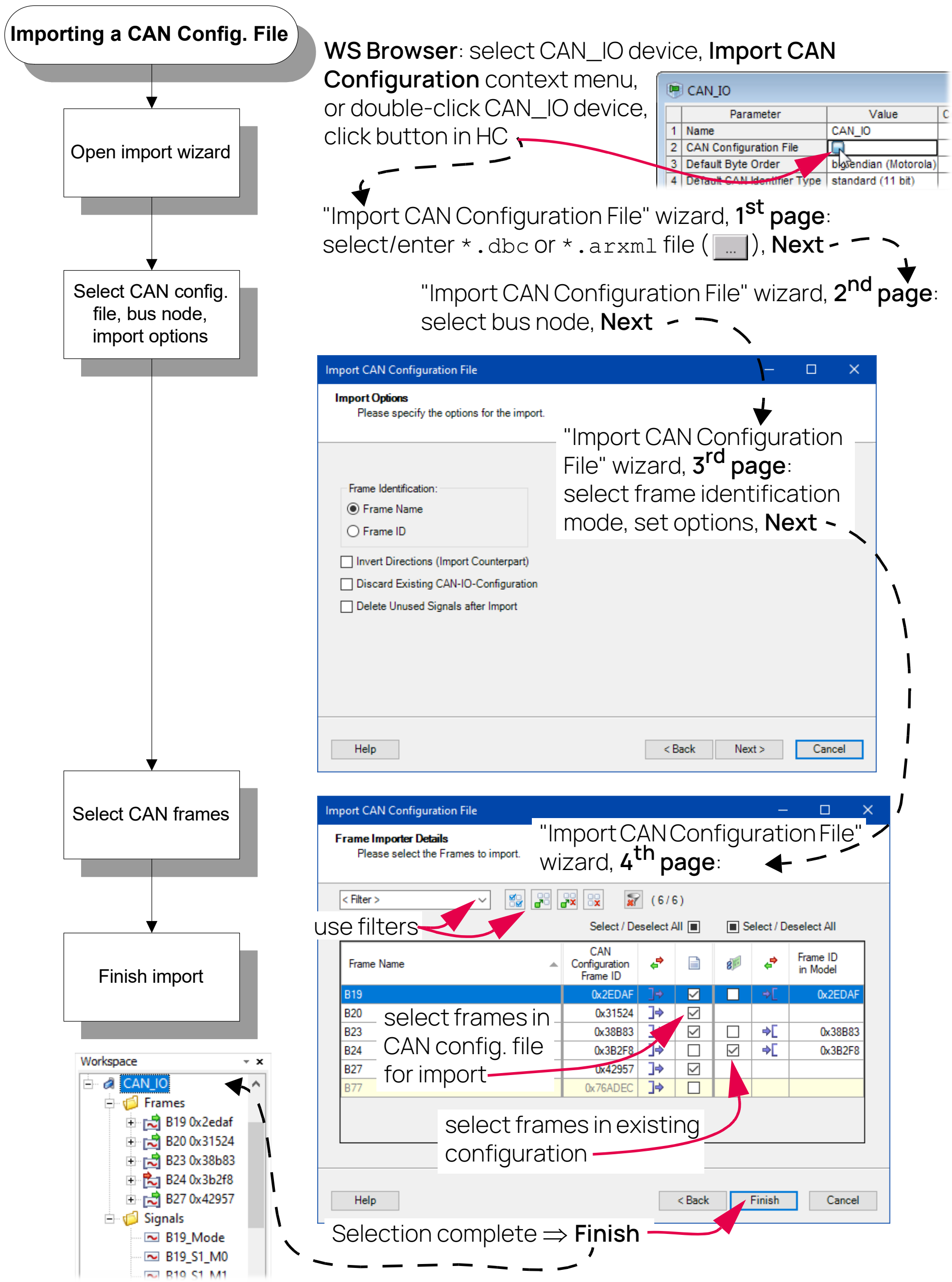


Fig. 4-18 Importing a CAN Configuration File

4.6.6 Exporting a CAN Configuration

> **NOTE**
>
> Exporting a CAN configuration as `*.arxml` file is currently not supported.

After you have configured a CAN device, you can export the configuration as CAN database file (`*.dbc`). Both classic CAN and CAN FD (ES910+ES922 and ES830) are supported.

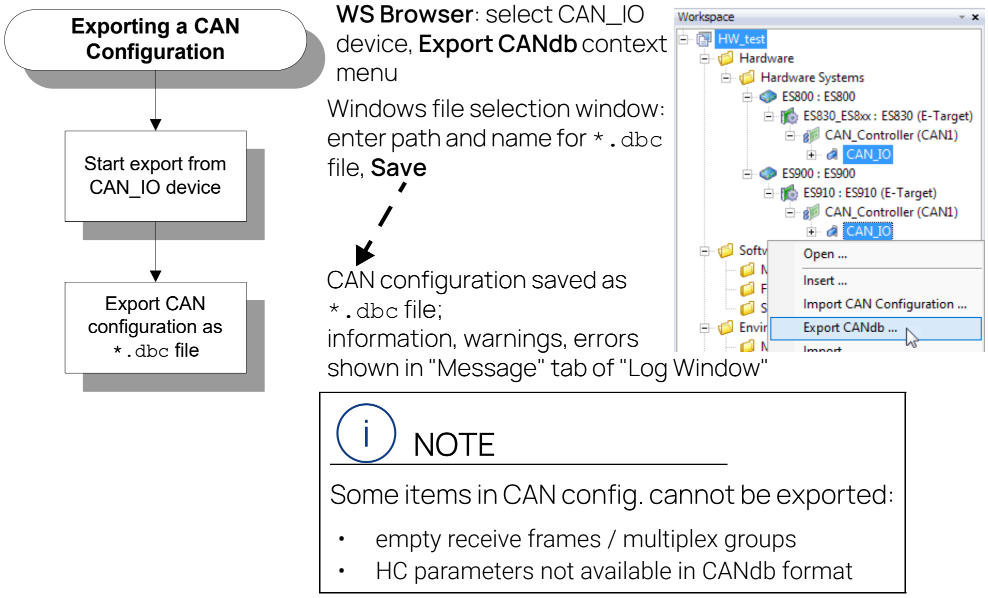


Fig. 4-19 Exporting a CAN Database

4.7 Setting Up a System Project

The system project (SP) combines one hardware system, one software system, and – optionally – one environment system to form an overall project.

In the system project, the signals provided by the hardware are mapped to the signals available in the software and the sequence order of the processes determined within the scope of the operating system configuration.

4.7.1 Creating a System Project

The `Systems` folder is the folder for system projects in the workspace. It can contain several system projects, although only one project is active at a time.

This section describes the creation of a system project for rapid prototyping. Creating a system project for virtual prototyping is described in section 4.11 "Virtual Prototyping".

First of all, create the system project (SP) under `Systems`, integrate the hardware system (HWS), software system (SWS) and environment system (ES) in the system project.

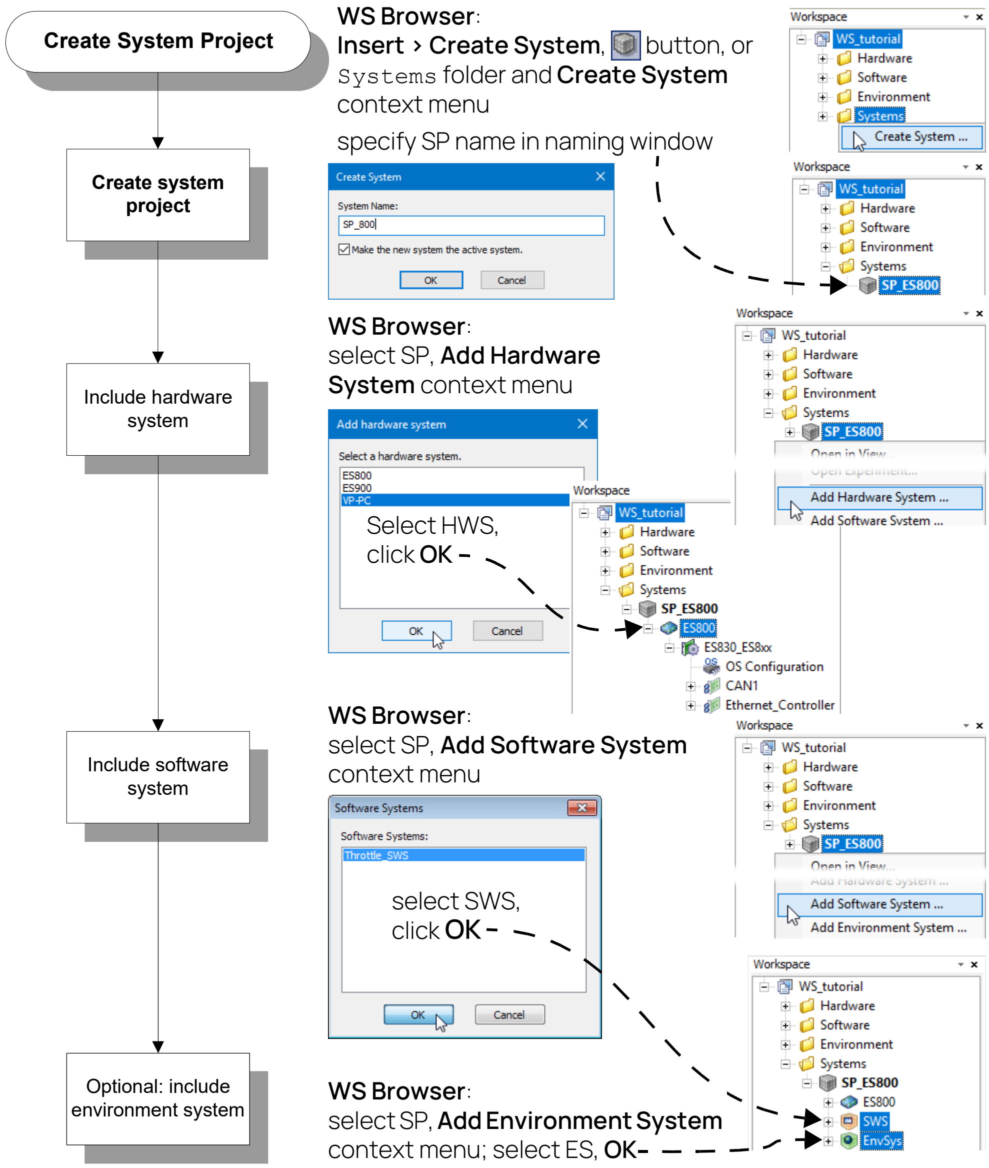


Fig. 4-20 Creating a System Project

4.7.2 Connecting Hardware and Software

You must connect hardware and software before you can generate an executable prototype.

By default, each hardware device is displayed as a set of two blue blocks. Hardware channels that can be connected to the signal sinks of the software system are arranged at the right side of one block. Hardware channels that can be connected to the signal sources of the software system are arranged at the left side of the second block. You can add/remove hardware blocks manually.

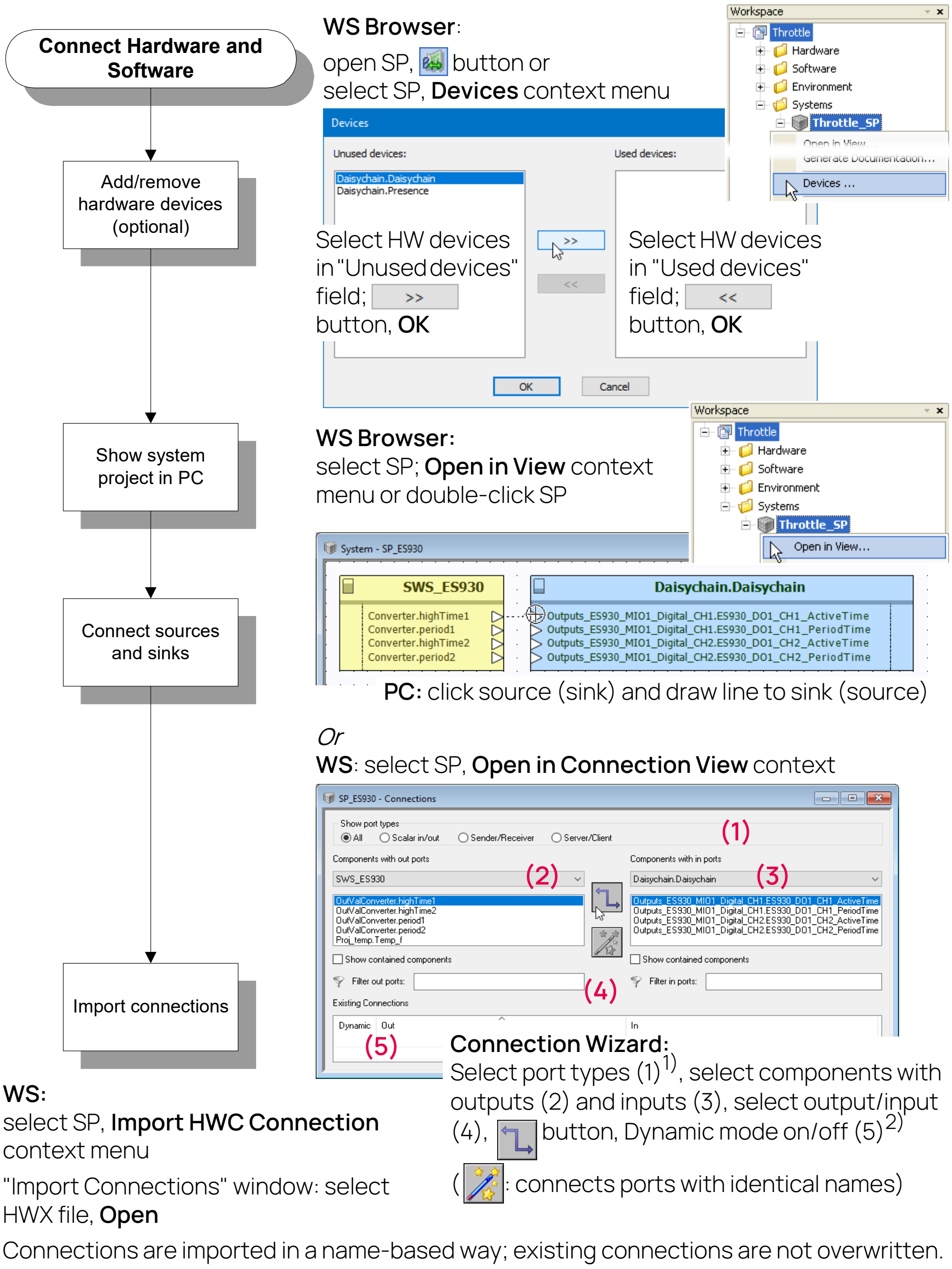


Fig. 4-21 Connecting Hardware and Software

1) **Scalar in/out**: modules; **Sender/Receiver**: AUTOSAR SWC; **Server/Client** not supported for hardware connections
2) no "Dynamic" connection mode for SWC connections

4.8 Configuring the Operating System

The configuration of the operating system is a very important task when creating a real-time prototype.

The operating system (OS) determines the processing sequence of the tasks and processes or runnable entities (RE) which compete for the processor and, if necessary, toggles between the execution of different tasks. All the necessary settings are made when the operating system is configured.

4.8.1 Automatic OS Configuration

INTECRIO offers several possibilities for automatic OS configuration.

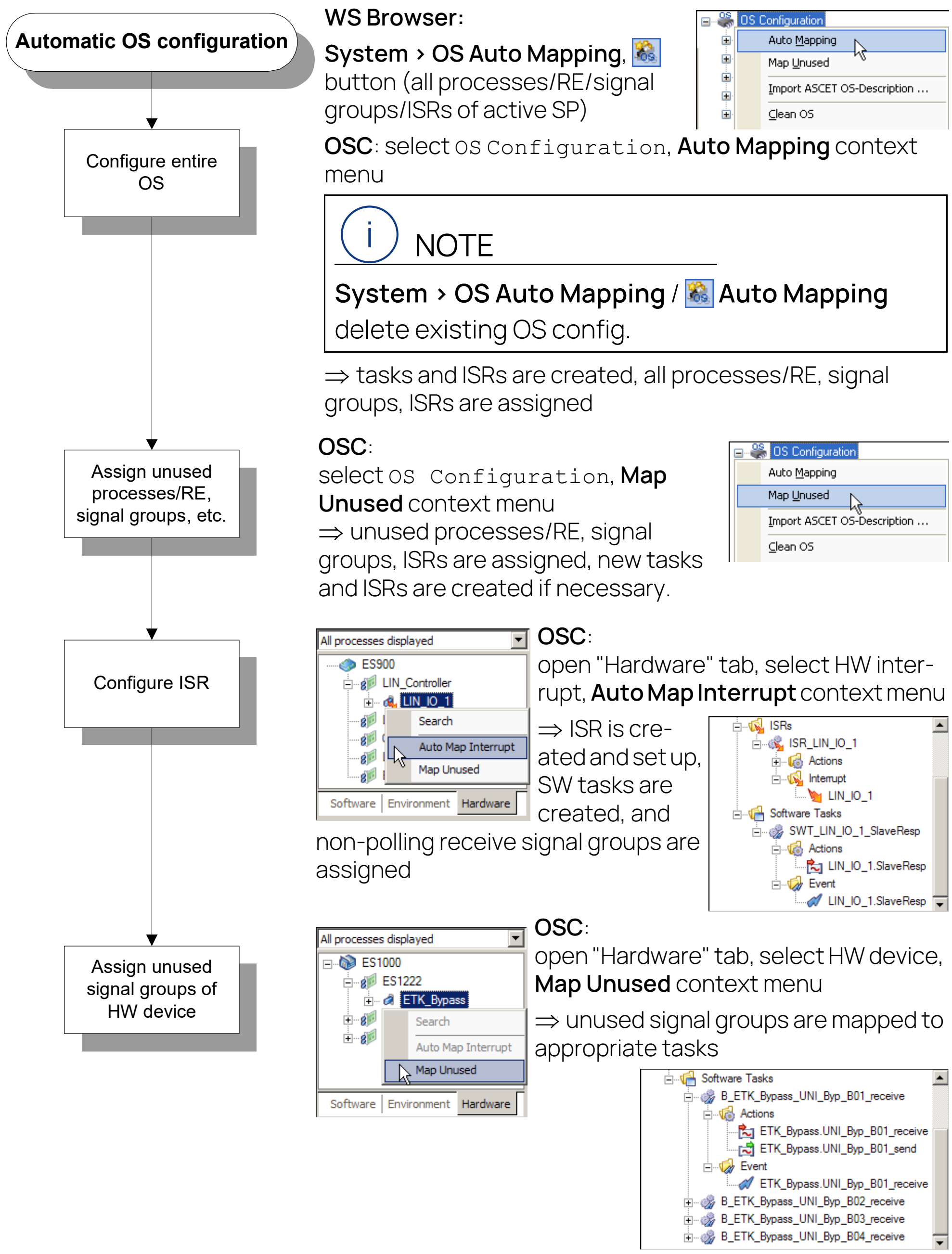


Fig. 4-22 Automatic Operating System Configuration

4.8.2 Manual OS Configuration

Manual OS configuration is also possible in INTECRIO. You can configure the OS from scratch, or you can use manual configuration to adjust the results of automatic configuration (see Fig. 4-22).

First, you create the necessary timer and software tasks and ISRs. The `Init` and `Exit` tasks have been created automatically.

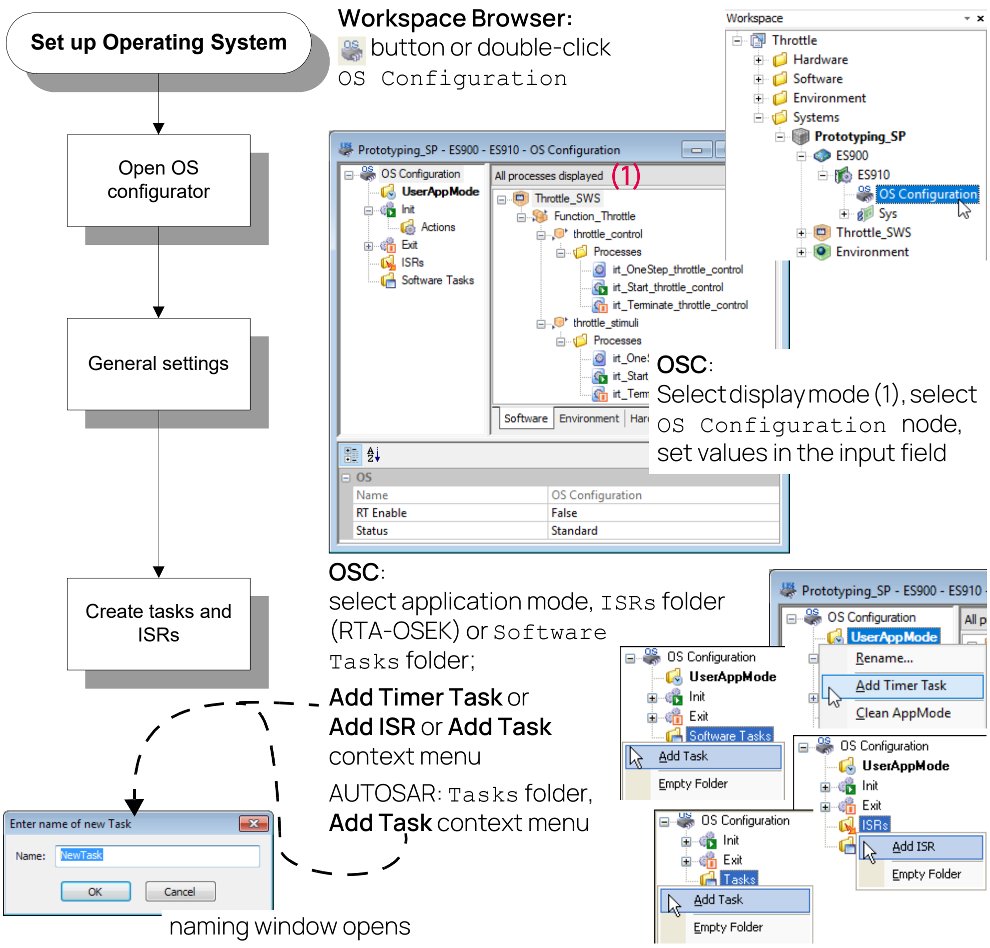


Fig. 4-23 Setting up the Operating System

Next, you set the properties of the tasks you have created. The properties of the `Init` and `Exit` tasks are defined automatically by INTECRIO. Then you assign processes or RE and signal groups to the tasks.

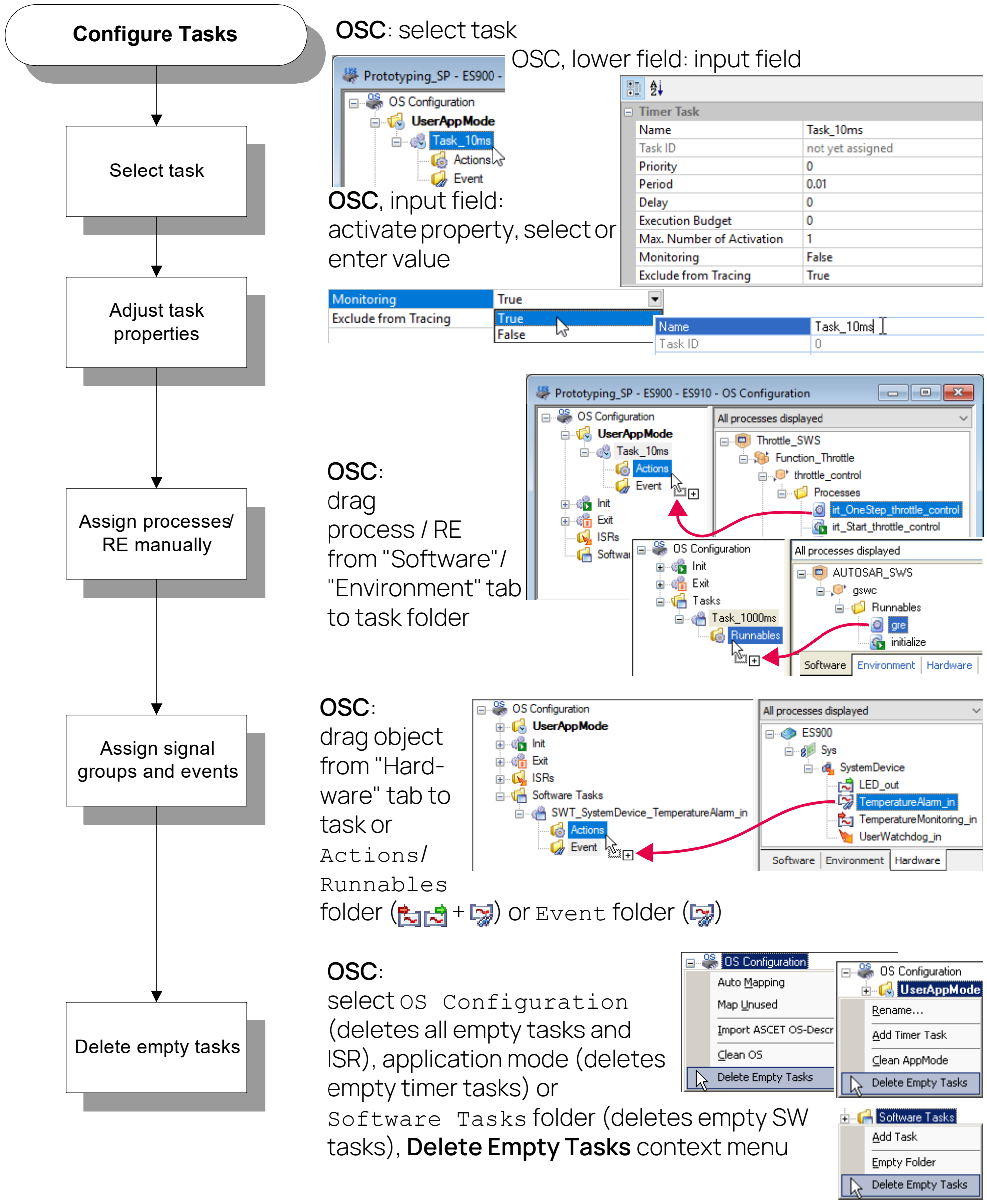


Fig. 4-24 Configuring Tasks Manually

Interrupt service routines (ISR) are configured similar to tasks. Each ISR is assigned one HW interrupt and the necessary processes and signal groups. For some HW interrupts, the trigger event can have sub-events with individual processes/signal groups.

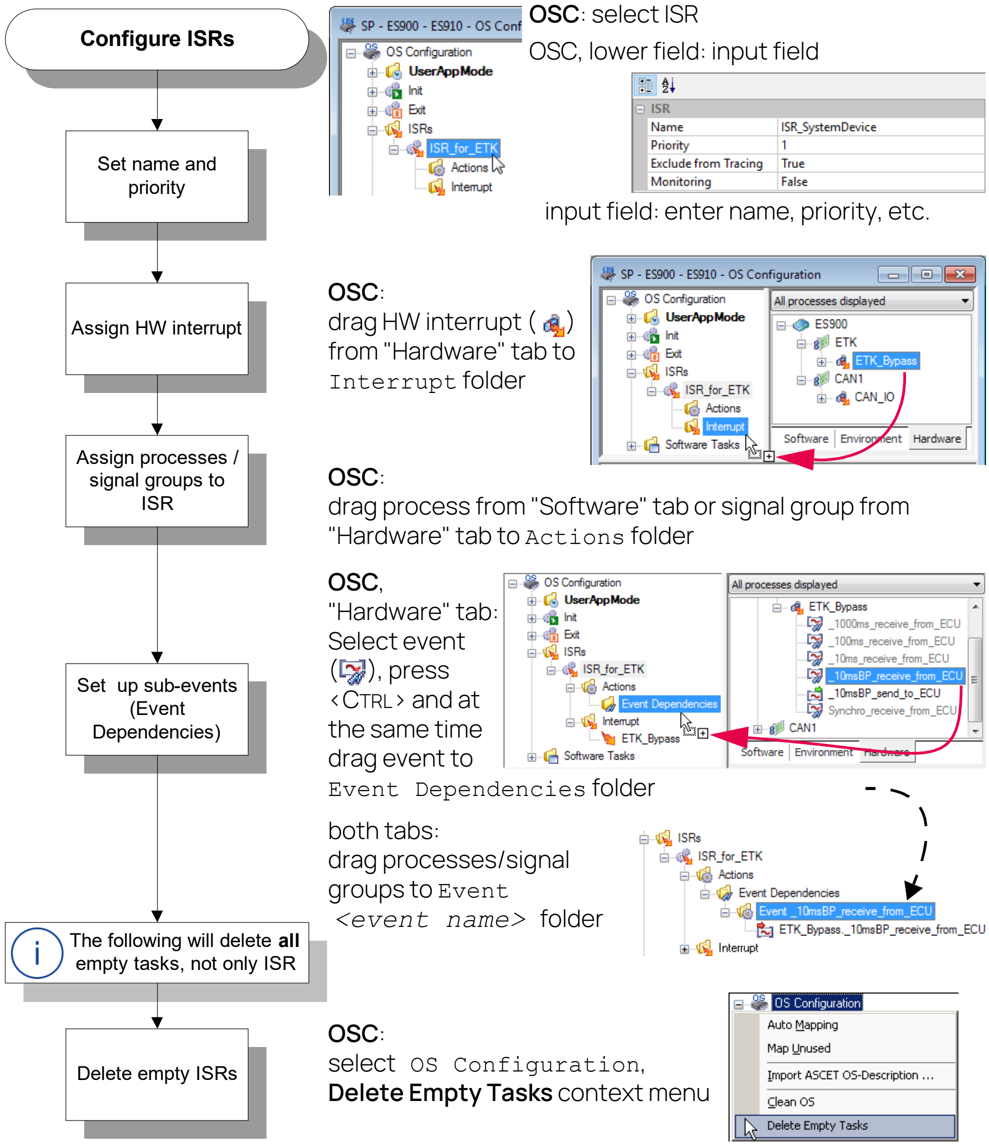
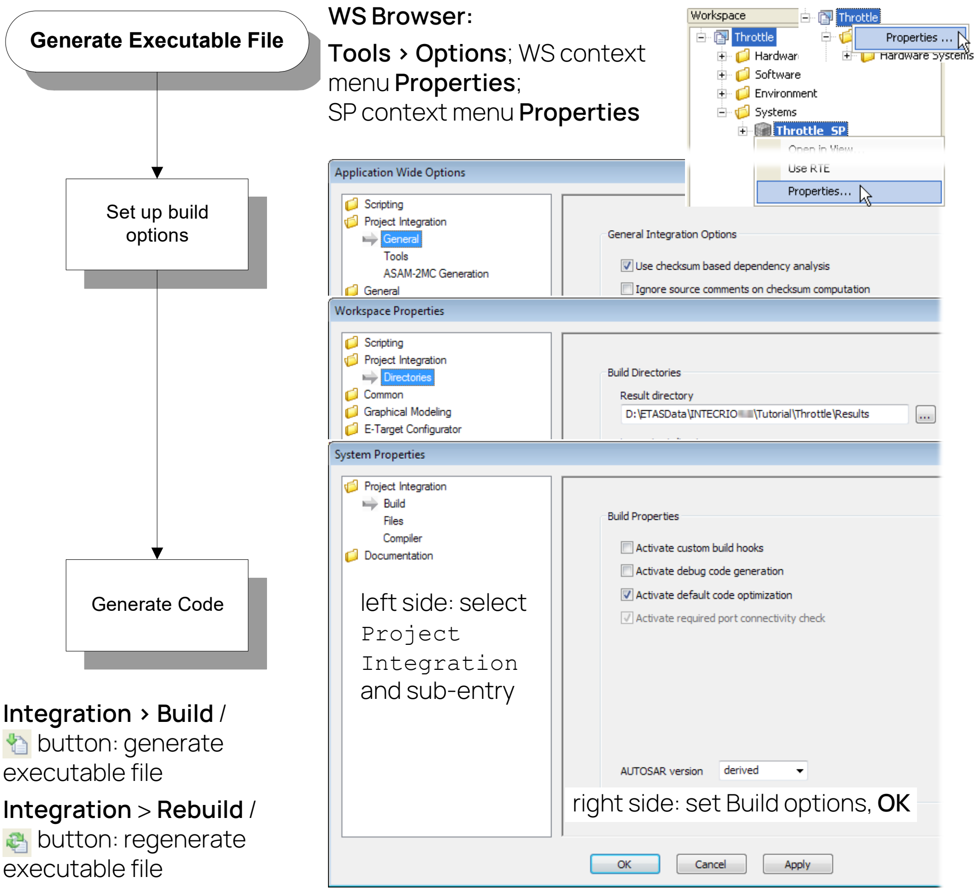


Fig. 4-25 Configuring ISRs Manually

4.9 Generating Executable File

The Build process more or less runs automatically. You simply have to set a few options and start the process.



Integration › Build / button: generate executable file

Integration > **Rebuild** / button: regenerate executable file

Integration > **Generate** / button: only generate code

Integration > **Clean** / button: delete generated files

Integration > **Abort** / button: cancel generation

Fig. 4-26 Generating the Executable File

4.10 Experimenting

The functionality needed during the execution of a measurement and calibration task and which is used during the preparation of the individual subtasks is mainly integrated in the ETAS Experiment Environment. It is used to select measure and calibration values and their arrangement in various windows such as the oscillo-scope, bar display or calibration editors.

> i NOTE
>
> Beginning with V5.0.4, INTECRIO supports ETAS Experiment Environment V3.8.4. ETAS Experiment Environment V3.7 is *no longer supported*.
>
> ETAS Experiment Environment V3.9 is *not supported*.
>
> Instead of the ETAS Experiment Environment, you can use INCA V7.4 with INCA-EIP to experiment with an ES800 hardware system.

4.10.1 Preparations for Experimenting

An empty experiment is created with each system project. First of all open the ETAS Experiment Environment (Fig. 4-27), then select a hardware (ES900 or ES800) connected to your computer, and ensure that all necessary windows and toolbars are displayed (Fig. 4-28).

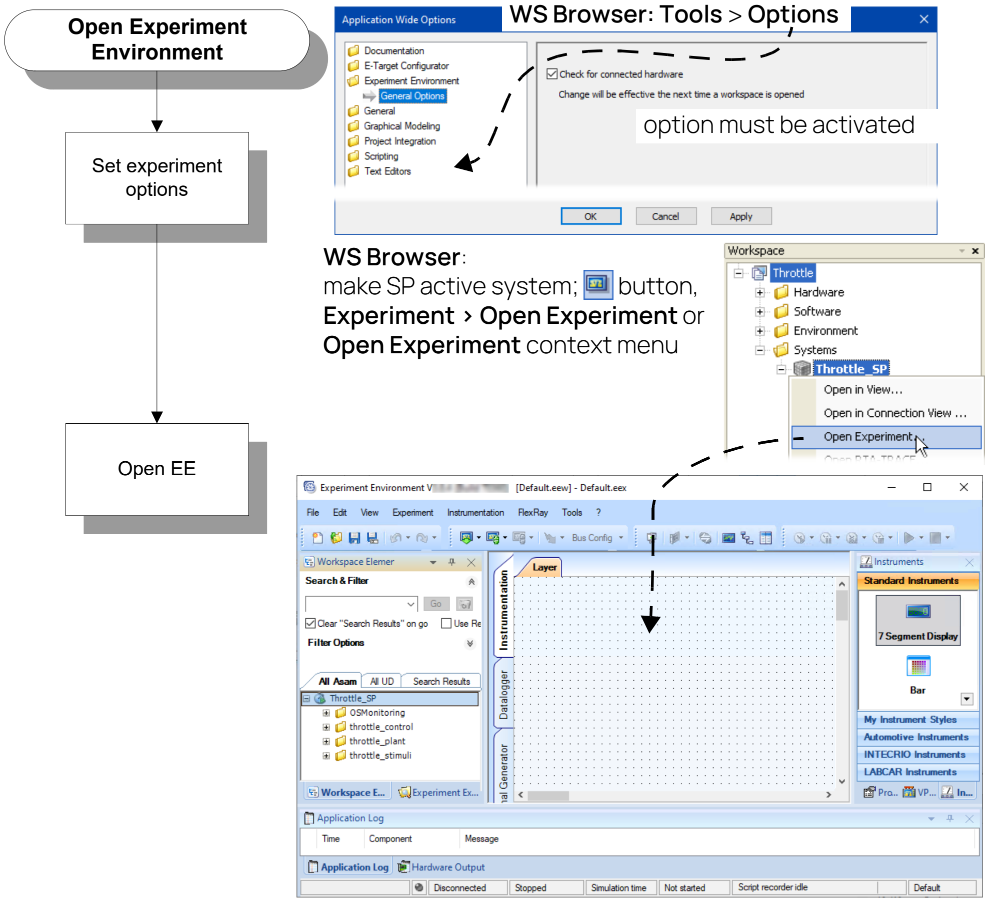


Fig. 4-27 Opening the ETAS Experiment Environment

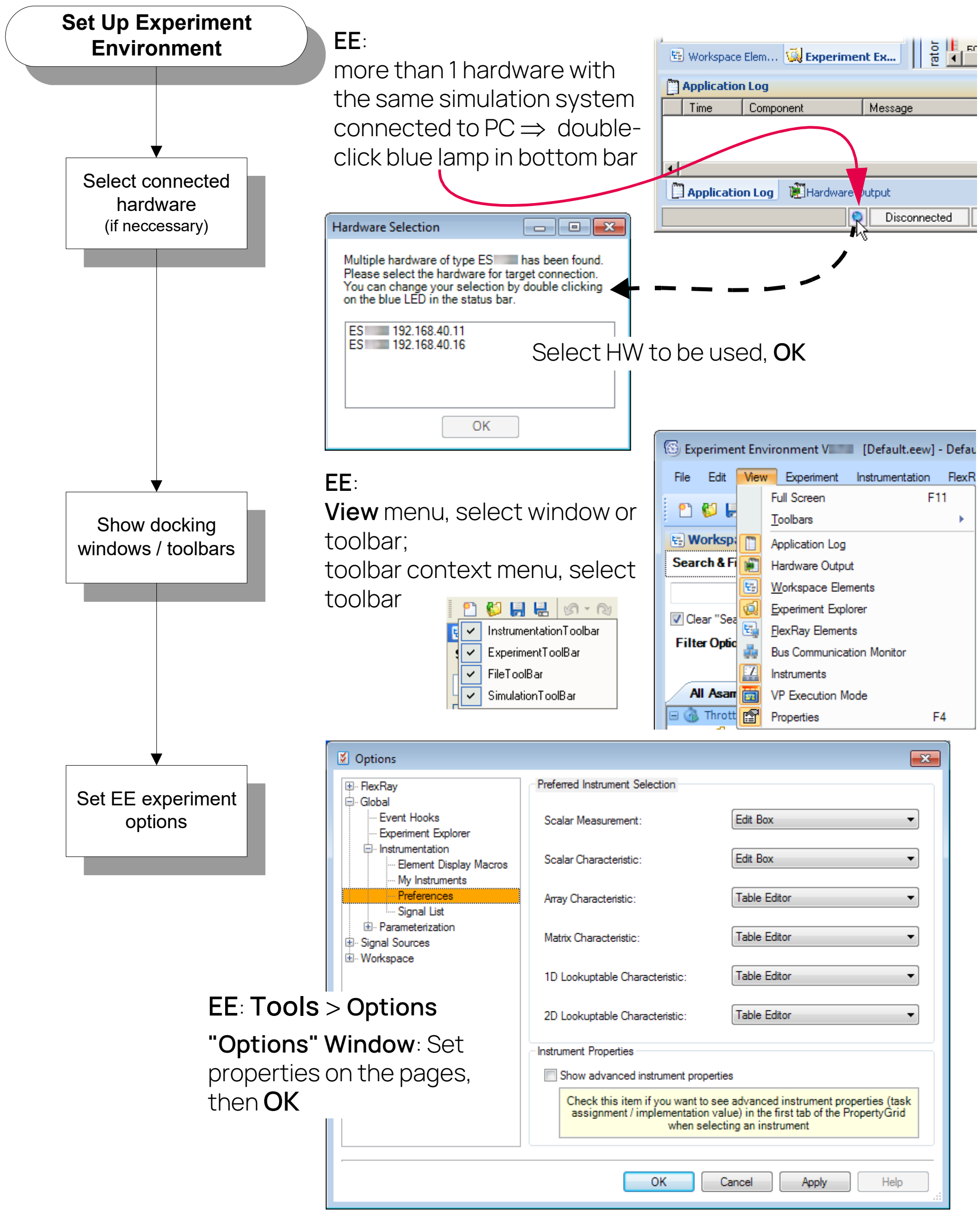


Fig. 4-28 Setting up the ETAS Experiment Environment

4.10.2 Creating and Setting Up Measuring and Calibration Windows

A range of instruments is available for displaying measure and calibration variables. You can create any number of oscilloscopes and other instruments; instruments of the same type (except oscilloscopes and table editors) can be grouped in one measuring/calibration window.

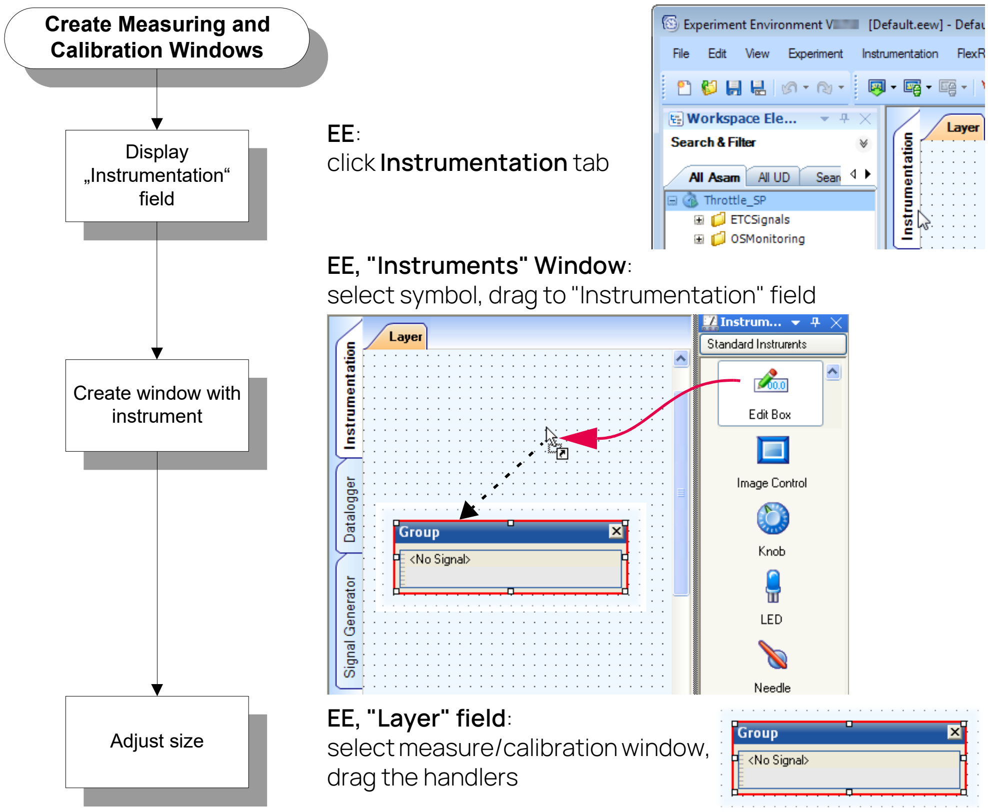


Fig. 4-29 Creating Measuring and Calibration Windows

You can set up each instrument individually.

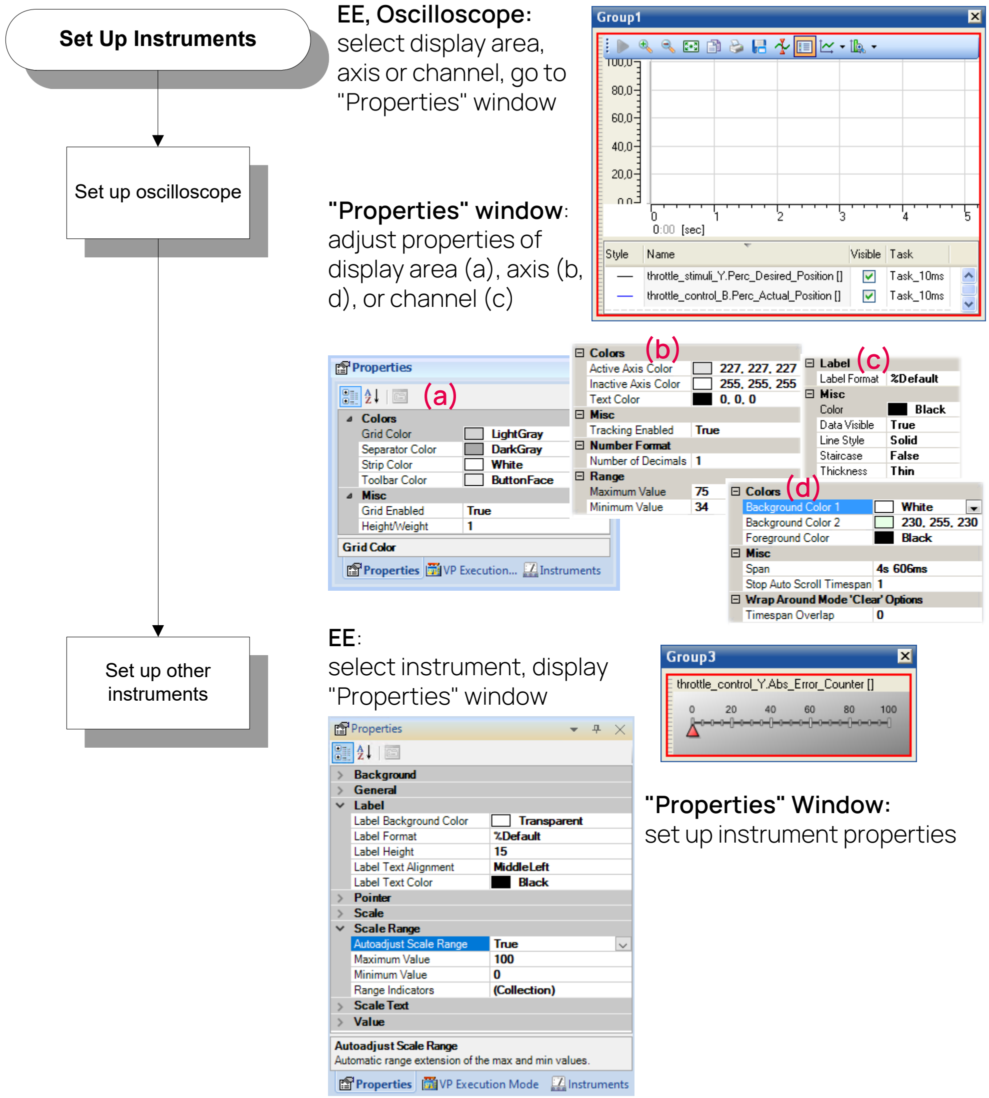


Fig. 4-30 Setting Up Instruments

4.10.3 Assigning Measurement and Calibration Variables

The measurement and calibration variables defined in the project are listed in the "Workspace Elements" window. You assign the values to suitable instruments.

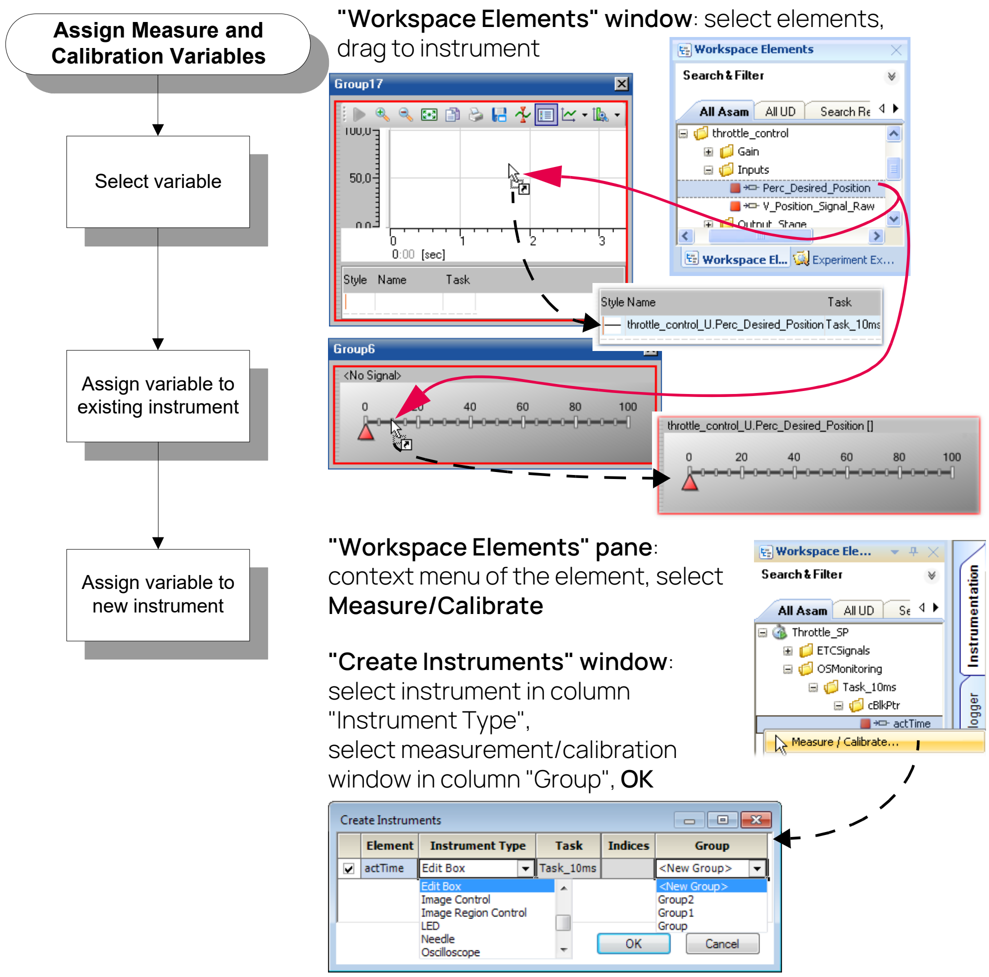


Fig. 4-31 Assigning Measurement/Calibration Variables to the Instruments

4.10.4 Working With Layers

By default, the "Instrumentation" field contains one layer ("Layer" tab) where measurement/calibration windows can be created. You can add more layers, rename and delete layers, and exchange measurement/calibration windows between existing layers.

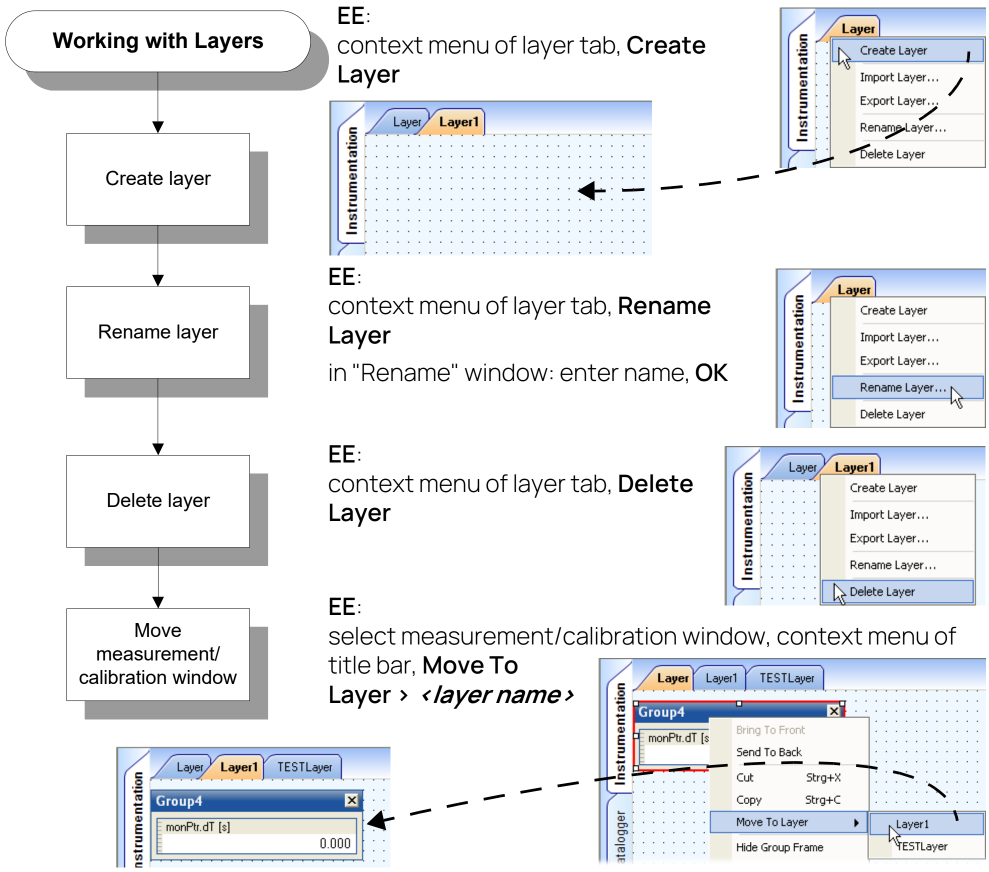


Fig. 4-32 Working with Layers

4.10.5 Managing Experiments

When you have set up the experiment (created measurement/calibration windows and layers, assigned variables), you can save the setting. You can create more experiments or load existing ones.

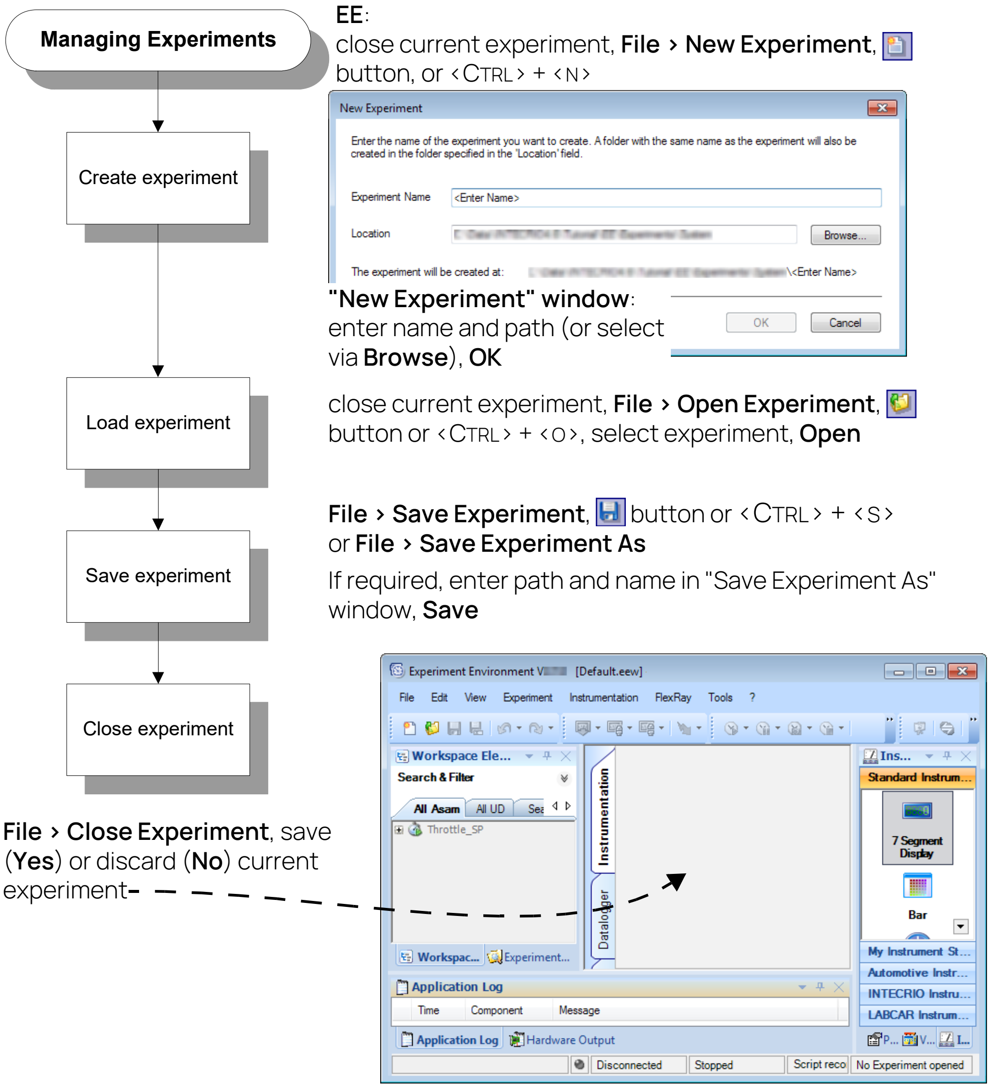


Fig. 4-33 Managing Experiments

4.10.6 Configuring Data Acquisition

INTECRIO makes it possible to prepare and configure the recording of measure data in the data logger. You can define the saving of the measure file using settings such as location, name under which it is to be saved and the recording duration.

You can define trigger conditions for the automation of the start and end of data acquisition (if necessary with pretrigger and posttrigger time), which trigger the desired event.

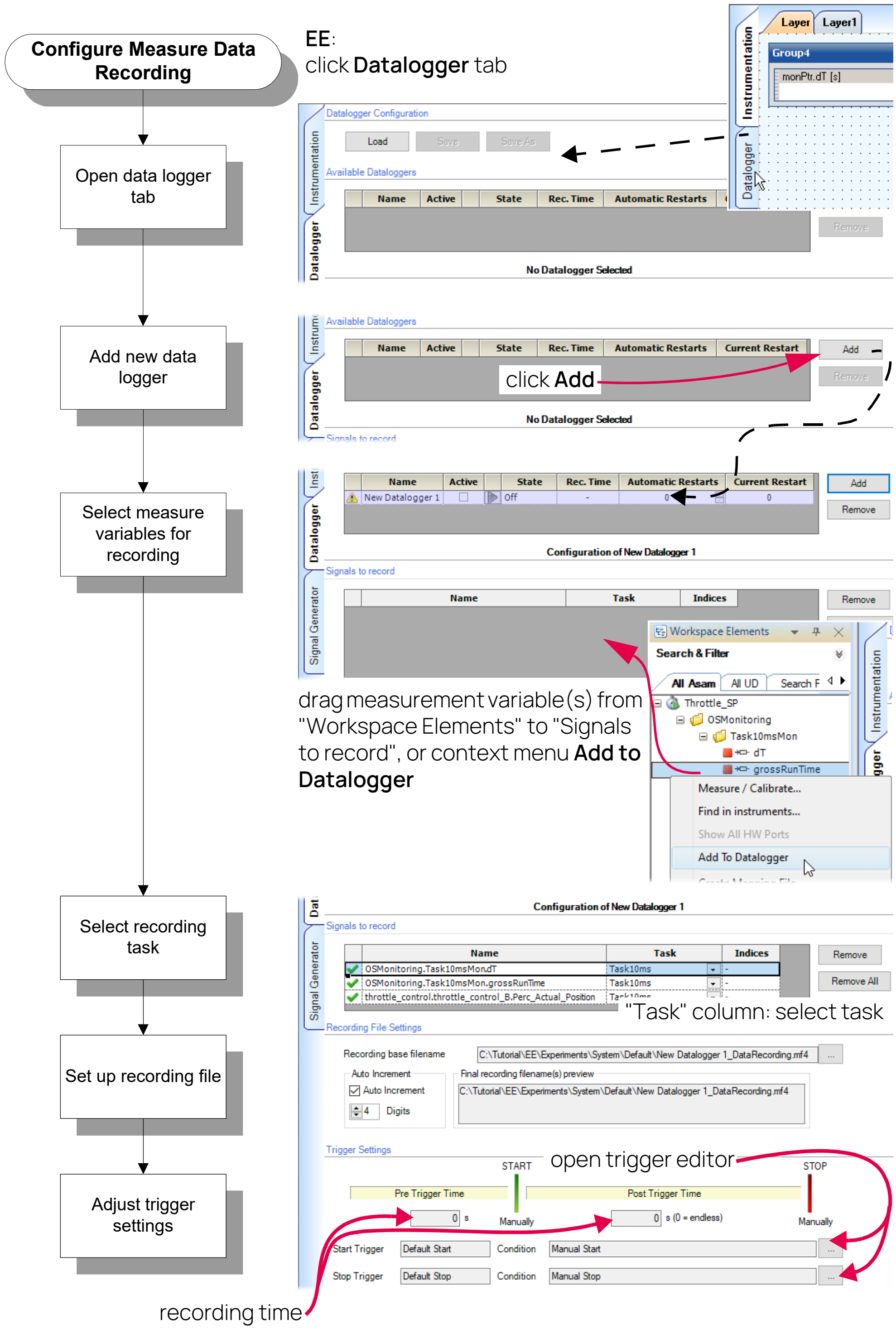


Fig. 4-34 Configuring Data Acquisition

4.10.7 Executing an Experiment

First, the prototype generated for the experiment is transferred to the experimental target. The real-time operating system runs on this target. Then, the operating system, and with that the simulation, is started.

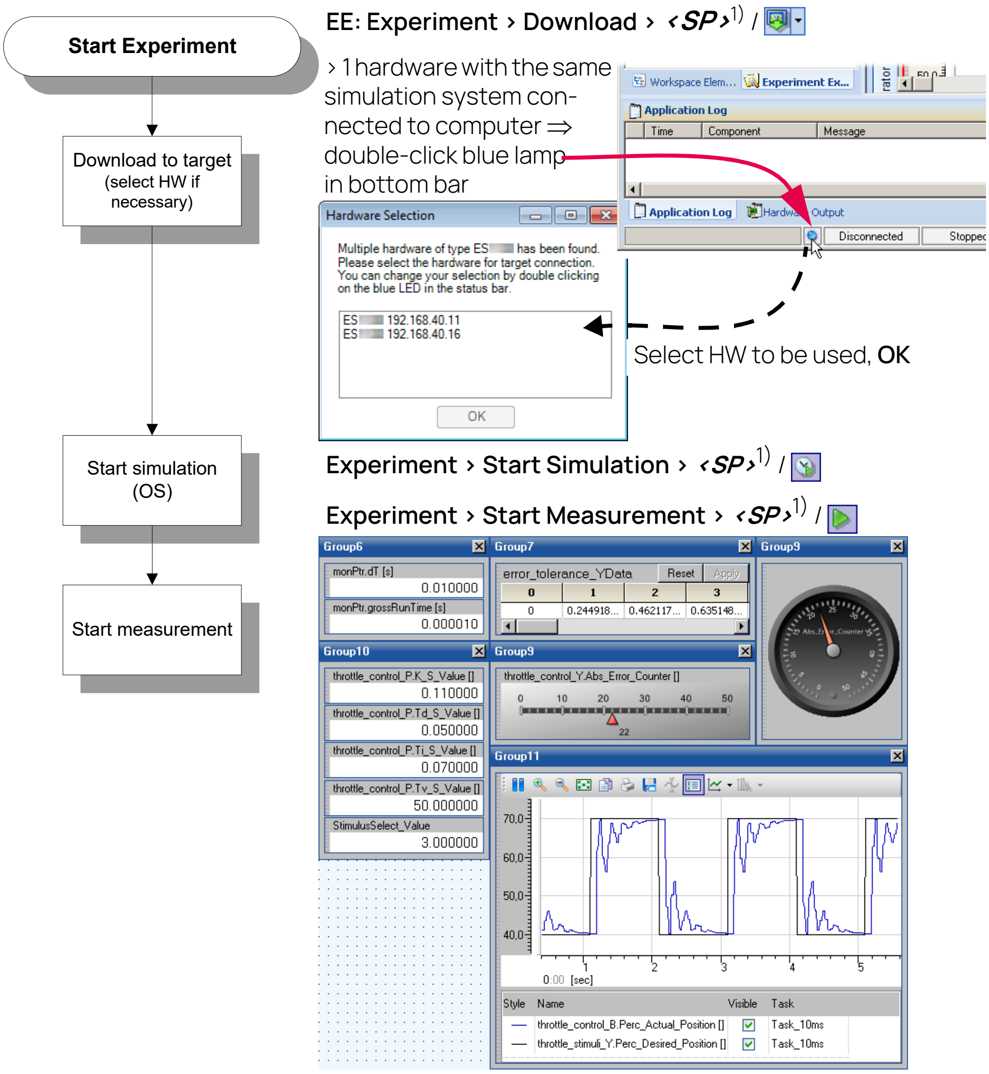


Fig. 4-35 Start Experiment

1) ‹SP› is the name of the system project

For measure data recording, you have to activate the data logger *before* starting the measurement. Once the data logger is activated, recording starts with the measurement or when the selected trigger condition is fulfilled.

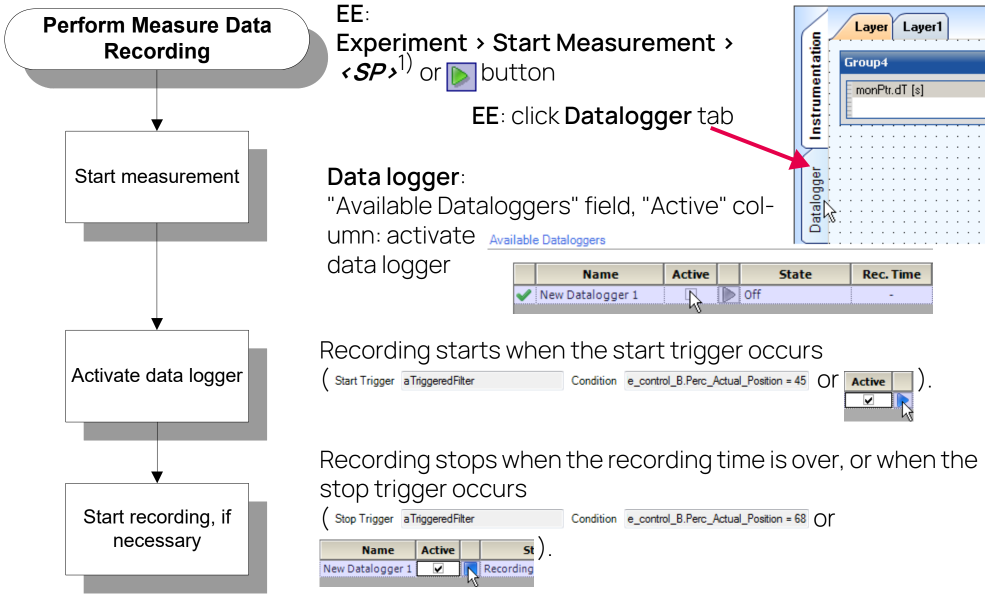


Fig. 4-36 Perform Measure Data Recording

1) *‹SP›* is the name of the system project

4.10.8 Measuring and Calibrating

Measure values are displayed in the instruments and in the oscilloscope. In the oscilloscope you also have the possibility to analyze the measured values.

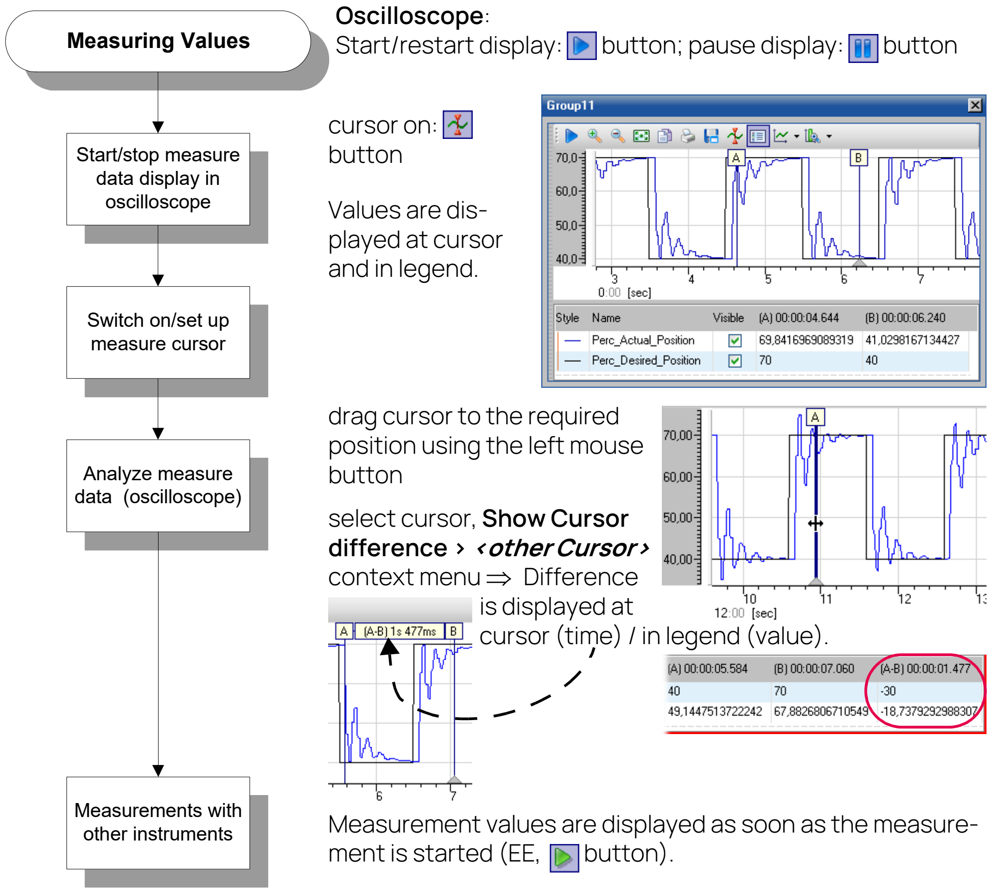


Fig. 4-37 Measuring Values

If you have assigned a calibration variable (characteristic value, curve or map) to an instrument, this instrument works like an editor. You can modify the values of the calibration variables directly in the display using the various editors.

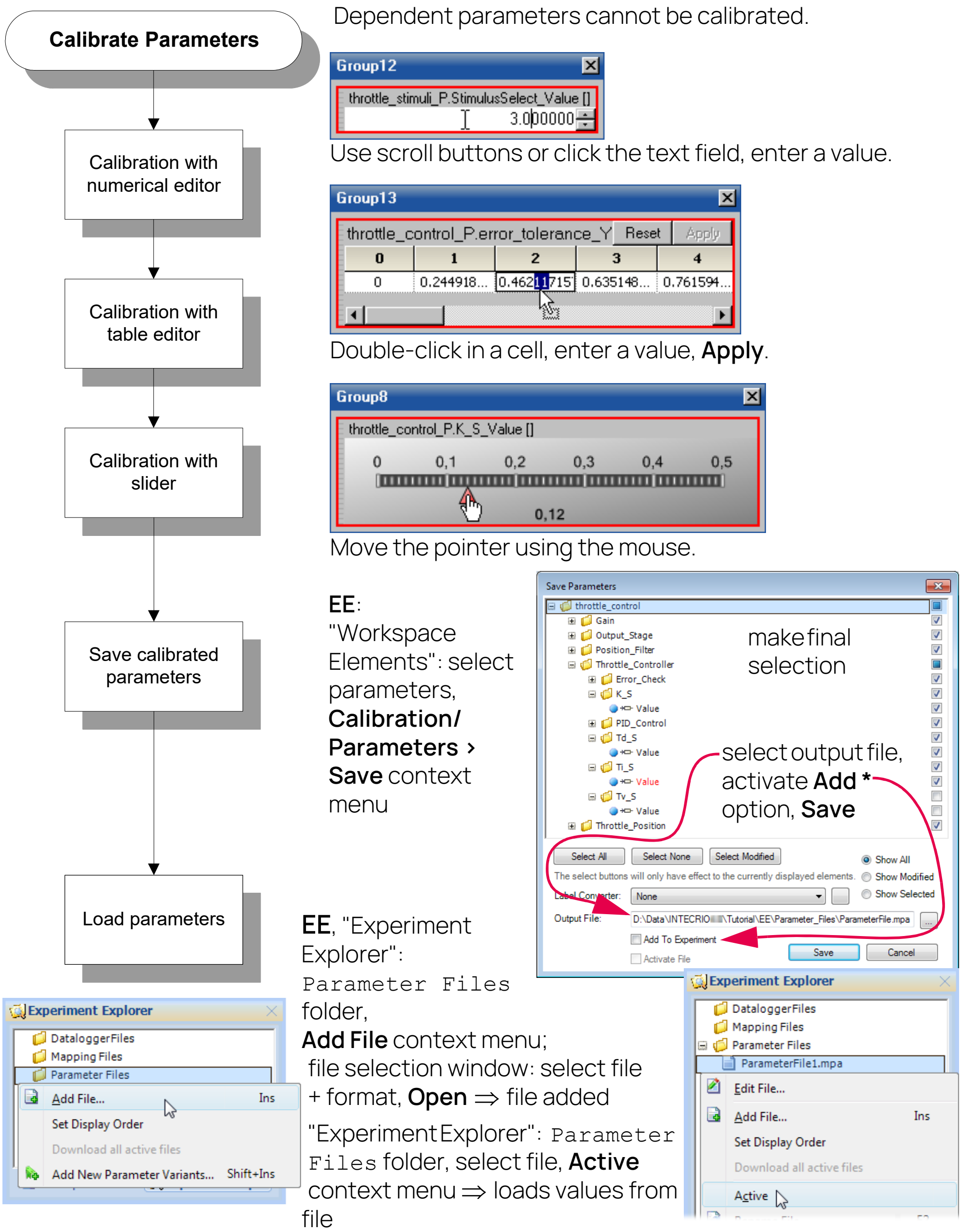


Fig. 4-38 Calibrating in the Editors

4.10.9 Post-Processing

When you write the results of the measurements in a file, you can evaluate these later offline with an application for measure data analysis.

The ETAS program "Measure Data Analyzer" (MDA) is precisely this kind of offline tool for displaying and analyzing saved measure data. The MDA has two different evaluation windows: first of all an X-t display which can be used as an oscilloscope and as an XY plotter; on the other hand, the tabular display which is particularly suitable for quick reading of exact values. You can combine measure signals from different measure files, configure the display on screen, and save these settings in an evaluation configuration. The following are available for the actual analysis: various zoom functions for the navigation in the measure file, measure cursor for measuring certain values and automatic difference calculation.

The MDA is part of the delivery scope of INTECRIO, ASCET and INCA.

4.11 Virtual Prototyping

A system project for virtual prototyping is created in exactly the same way as a system project for rapid prototyping.

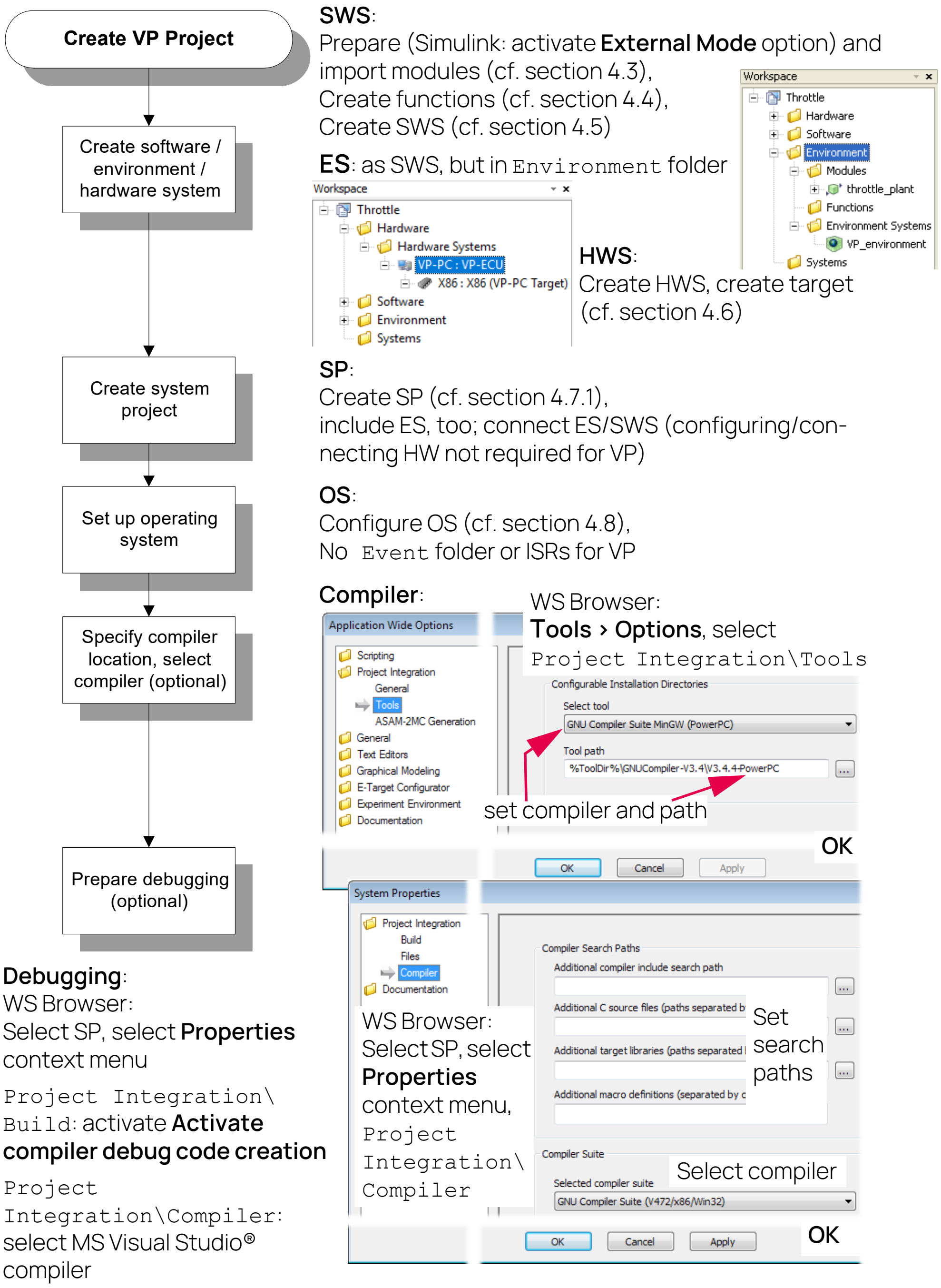


Fig. 4-39 Creating a Project for Virtual Prototyping

The major difference between executing the virtual prototyping experiment and the rapid prototyping experiment is the possibility of selecting a mode for the simulation time.

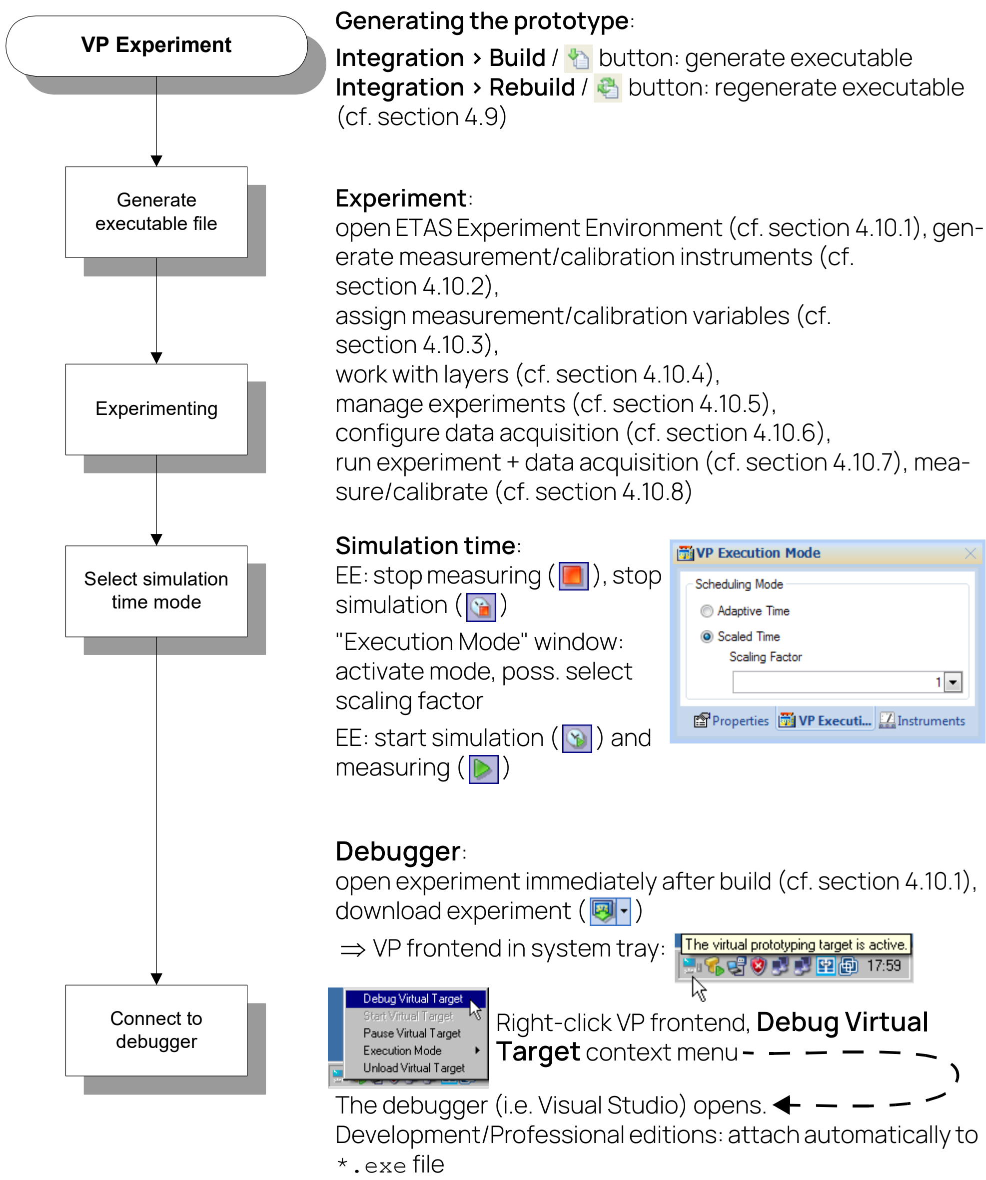


Fig. 4-40 Virtual Prototyping Experiment

4.12 Management and Exchange of Workspaces

4.12.1 Managing Workspaces

As mentioned before, INTECRIO allows you to work with several workspaces. Apart from creating a new workspace, initially empty (see “Creating a New Workspace” on page 40), you can also use existing workspaces (e.g., from a co-worker) by opening one of them. Of course, working with workspaces also involves saving. In doing so, not only the workspace is saved, but also the layout of the INTECRIO window.

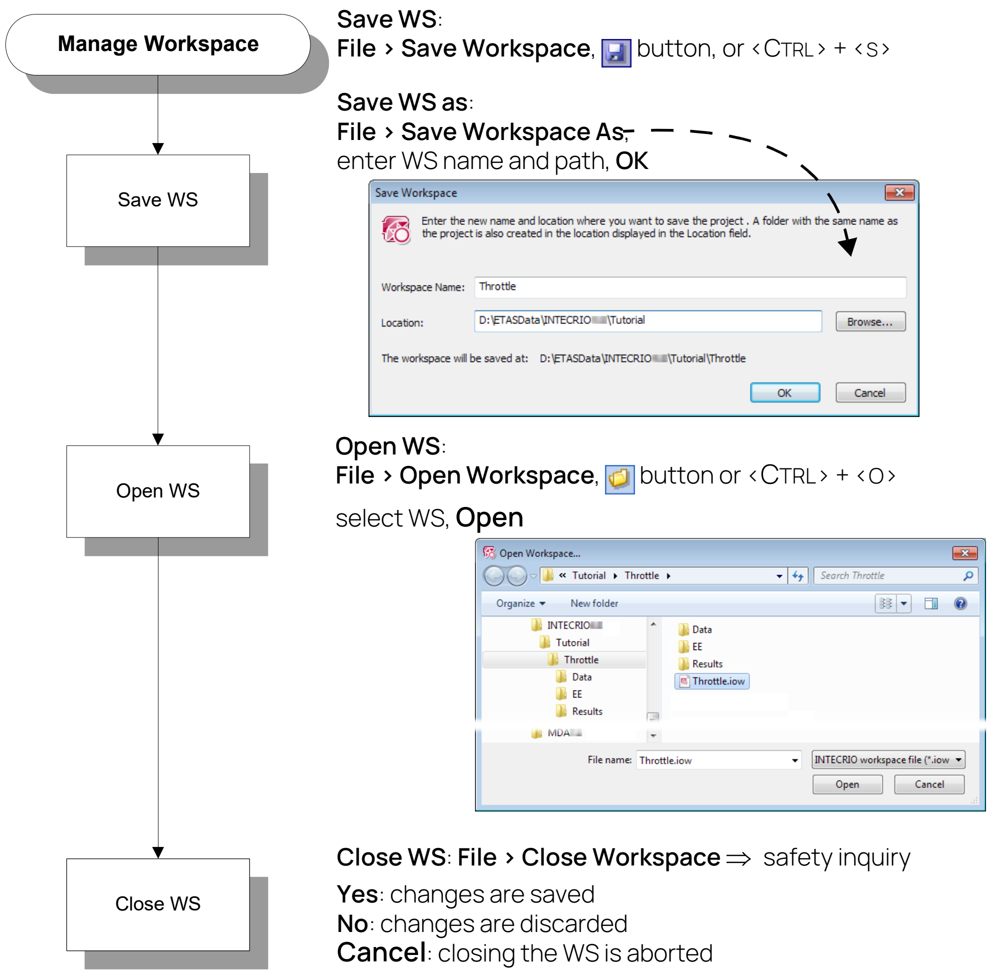


Fig. 4-41 Managing Workspaces

4.12.2 Exchanging Workspaces via Import/Export

You can use the WS browser to export or import a workspace. You can export or import all, none, or selected modules with the WS. The Import/Export function is an invaluable tool when exchanging existing workspaces.

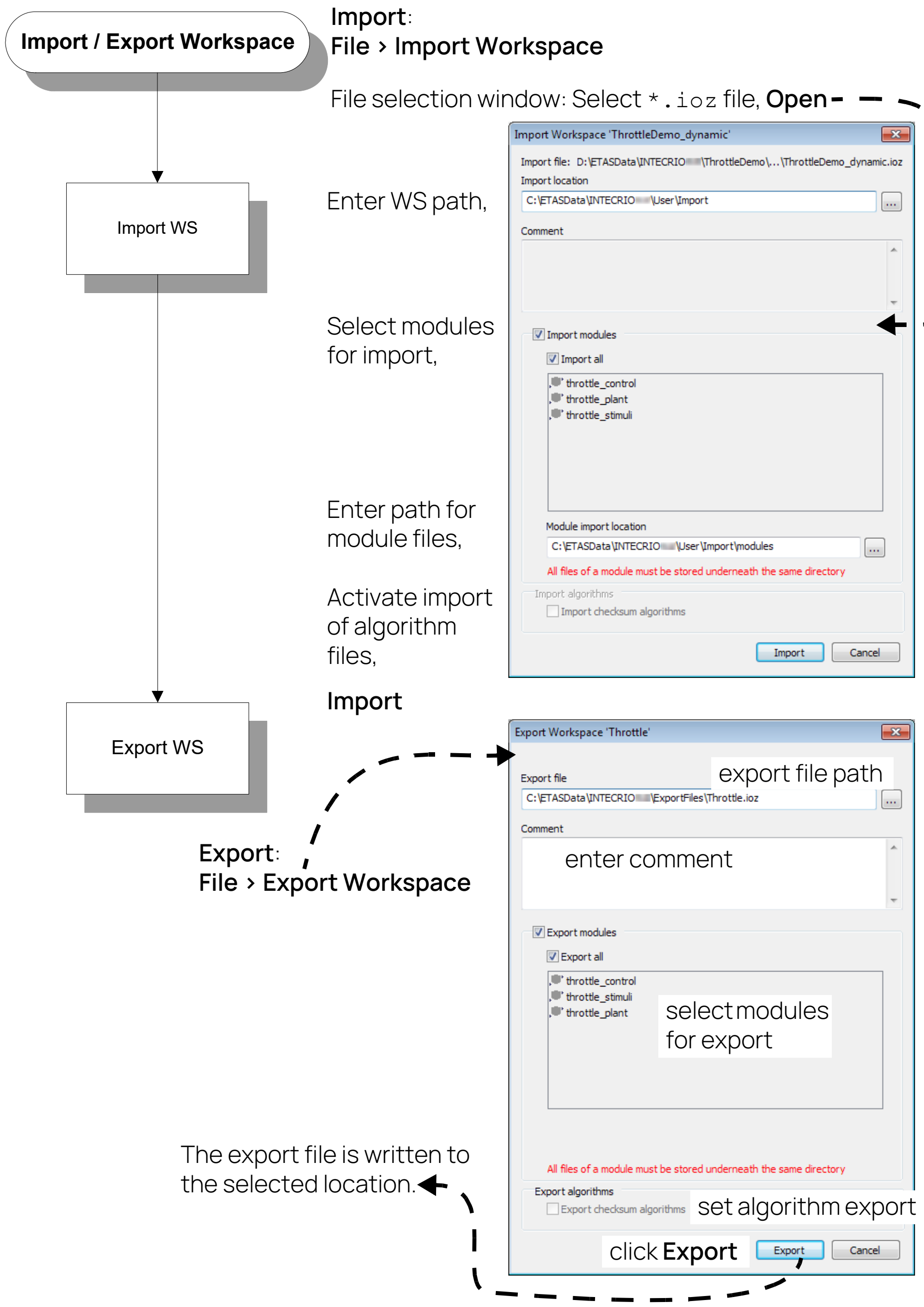
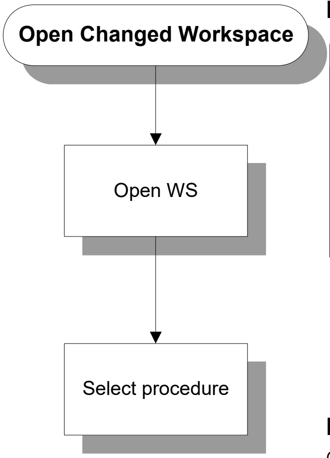


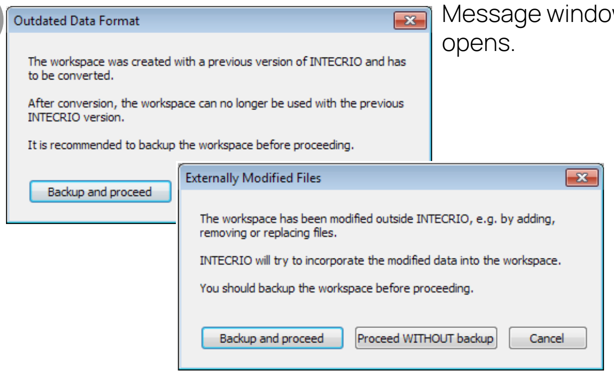
Fig. 4-42 Import / Export of Workspaces

4.12.3 Opening a Changed Workspace

If a workspace is shifted in the Windows explorer, or if one of its files is changed outside INTECRIO, a special warning appears when you open the workspace. The same happens if a DTD is outdated or missing.



File › Open Workspace, button, or ‹Ctrl› + ‹O›



Message window opens.

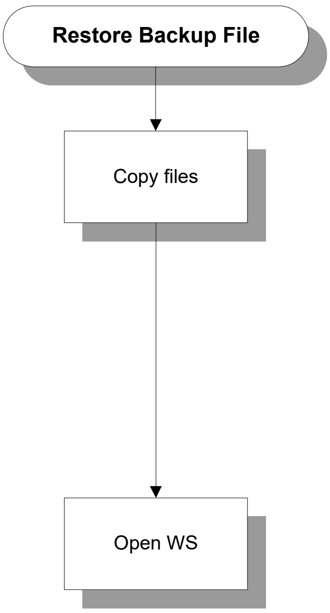
Backup and Proceed: In the WS directory, a copy of the `Data` directory is stored in the `Data.bak`*<n>* directory. The WS opens.

Proceed WITHOUT Backup: The WS opens without a backup being created.

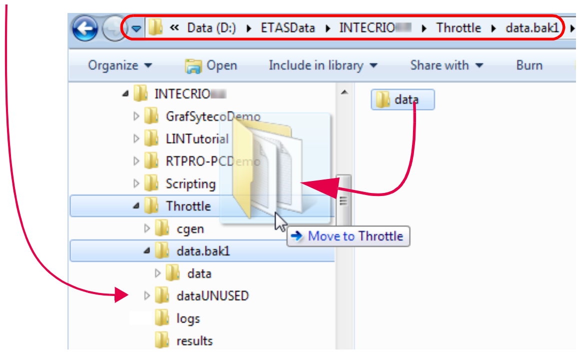
Cancel: Aborts the procedure.

Fig. 4-43 Open changed Workspace

A backup thus created can be restored.



Windows Explorer:
go to `...\`*<workspace>* directory,
delete or rename folder `data`,



go to `...\`*<workspace>*`\data.bak`*<n>* directory,
copy directory `data` into `...\`*<workspace>*

INTECRIO:
File › Open Workspace, button, or ‹Ctrl› + ‹O›

Fig. 4-44 Restoring a Backup File

4.13 Editing the Layout of Graphical Blocks

INTECRIO provides several ways to adjust the layout of graphical blocks in the Project Configurator (PC).

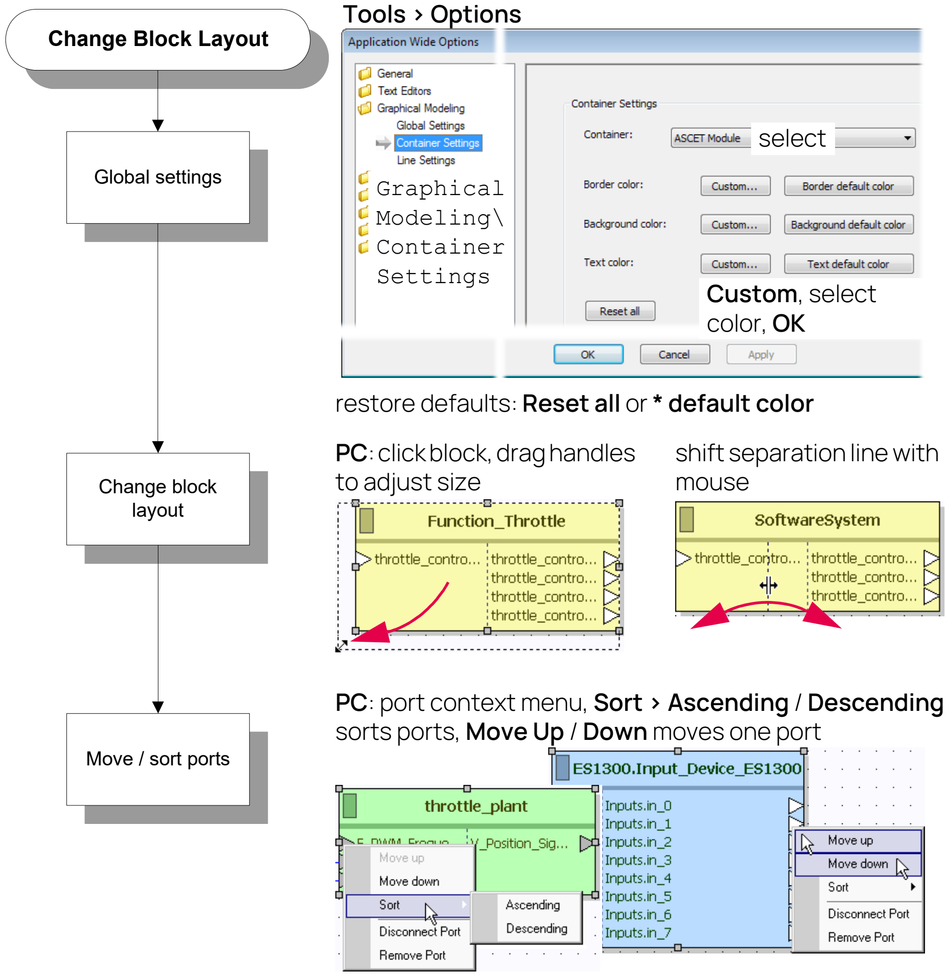


Fig. 4-45 Editing the Layout of Graphical Blocks

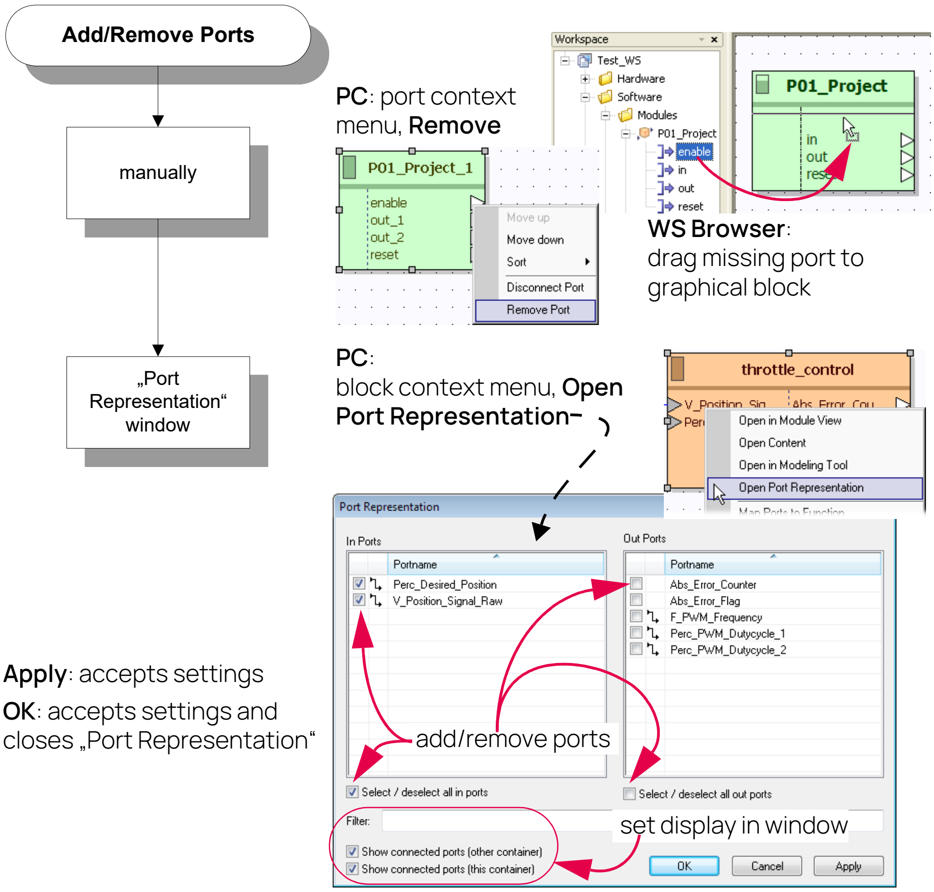


Fig. 4-46 Adding/Removing Ports in Graphical Blocks

4.14 Generating Automatic Documentation

INTECRIO offers automatic documentation generation for system projects (SP).

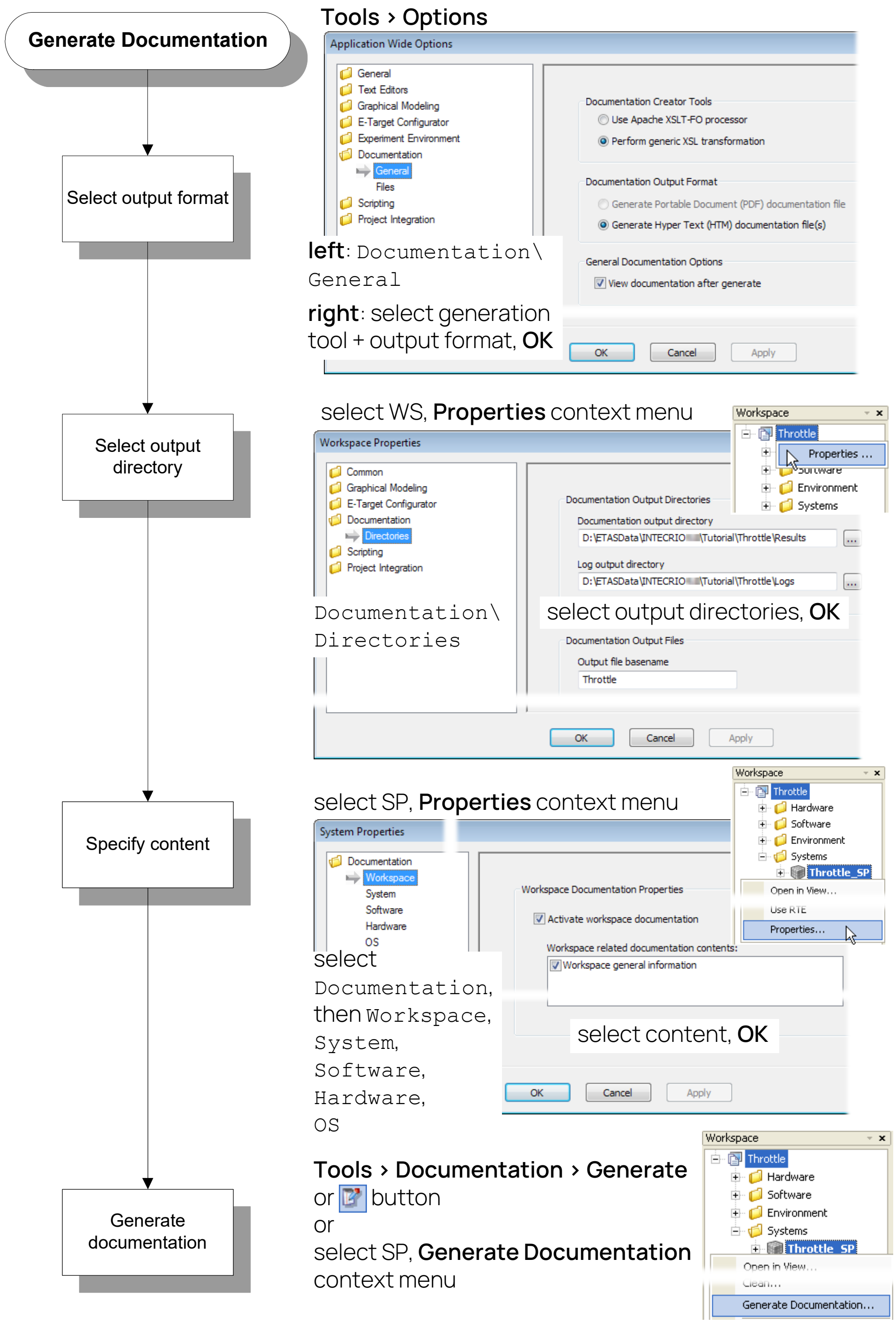


Fig. 4-47 Automatic Documentation for a System Project

5 INTECRIO Tutorial

5.1 Introduction

Users who are not yet familiar with INTECRIO will learn all the basic working steps of INTECRIO in this tutorial with the example of throttle control. The tutorial does not require any knowledge of INTECRIO, but does assume that the user is familiar with the Windows operating system and with the behavioral modeling tools MATLAB® and Simulink®.

We would, however, recommend that you work through the chapter "Understanding INTECRIO" in the User Guide before starting the tutorial.

5.1.1 Preparation

Before you can start the tutorial, you have to prepare the system.

The system you want to use for the tutorial has to have an executable INTECRIO installed. INTECRIO can either be launched using the icon on the desktop or using the Windows **Start** menu.

> i NOTE
>
> You can execute the introduction as well as lessons 1 to 7 in offline mode. You do not need any working hardware for these lessons.
>
> You need working hardware for "Lesson 8: ES930 – Example" .
>
> This tutorial uses the ETAS Experiment Environment V3.8.4. Alternatively, you can use INCA/INCA-EIP V7.4 or higher.
>
> ETAS Experiment Environment V3.7 is *no longer supported.*

"Lesson 1: Preparing Modules for INTECRIO" also requires a working installation of MATLAB and Simulink. INTECRIO supports versions R2016a – R2022a and their related service packs known at the time of the INTECRIO V5.0 release, 32 bit and 64 bit. Simulink models created with R2006b – R2015b can still be imported in INTECRIO.

The files required for "Lesson 2: Preparatory Steps" are supplied with the tutorial so that Lesson 1 can also be skipped.

Make sure that the following files are in the sample file directory (see Page 19; e.g. `<sample files>\ThrottleDemo\SimulinkModel`):

- `throttle_control.mdl`
- `throttle_stimuli.mdl`
- `throttle_plant.mdl`

Also ensure that the following files are in the `throttle_*_irt_rtw` subdirectories (`*` = `control` or `stimuli` or `plant`):

- `throttle_*.a2l`
- `throttle_*.six`
- `throttle_*_main.c`
- `throttle_*_irtmacros.h`

"Lesson 8: ES930 – Example" requires the following files; make sure they are placed in the *<sample files>*`\ES930Demo` directory and its subdirectories:

- `OutValConverter.six`
- `OutValConverter.a2l`
- `OutValConverterM.c`
- `OutValConverterM.h`
- `OutputValueConverterM.c`
- `OutputValueConverterM.h`
- `Proj_temp.six`
- `Proj_temp.a2l`
- `Proj_tempM.c`
- `Proj_tempM.h`
- `TemperatureM.c`
- `TemperatureM.h`

5.1.2 Conventions

The following conventions apply to the description of the tasks:

- "Enter `character string`" means that you should enter the highlighted character string via the keyboard. There is, however, the following exception:

  "Enter *<INTECRIO drive>*`\INTECRIO5.0\exp1.txt`" means "Replace the character string `<INTECRIO drive>` when entering by the installation drive of INTECRIO".

  Therefore, if INTECRIO is installed on drive `C:`, this instruction means "Enter `c:\INTECRIO5.0\exp1.txt`".
- The INTECRIO window you see after INTECRIO is launched, is referred to as the *graphical framework* throughout the tutorial.

5.1.3 Concepts

This section introduces the most important concepts and processes used in this tutorial.

Integration

Integration consists of the following steps:

a the compilation of models and model parts to create a control algorithm
b the connection of this algorithm with the hardware it is to run on
c the generation of an executable file

INTECRIO

INTECRIO is a tool which combines, i.e. integrates, the parts of the control algorithm created with various behavioral modeling tools, which enables the creation and configuration of a hardware system as well as the connection of this hardware system to the control algorithm.

Modules

A module in INTECRIO contains the generic description of a functionality for an ECU. It corresponds, for example, to an ASCET project (ASCET V5.1 or higher) or a Simulink model.

The description files for the interfaces (SCOOP-IX, `*.six`) and data (`*.a2l`) as well as C code are generated for working in INTECRIO from a model created in ASCET or Simulink using code generation. When a module is imported, only the interface description file is read into INTECRIO; the data description file and the C code are only of any significance when the executable file is generated.

A module can only be used once in a system project; multiple instancing is not supported.

Functions

In INTECRIO, a function is a structure object for software systems which does not have its own functionality. Different modules can be combined and connected in a function; in this way, they are clearly ordered and easy to use.

Software system

A software system contains the hardware-independent parts of the ECU description (the application software), i.e. the modules, functions and connections the control algorithm consists of.

Environment system

An environment system is used to model the plant model for virtual prototyping. It is built out of modules and functions, the same way as a software system.

Hardware system

A hardware system contains the complete description of a hardware topology consisting of descriptions of the relevant ECUs (experimental targets) as well as the descriptions of the interfaces between the devices.

System project

A system project combines a hardware system, one software system, the mapping of the signals and the configuration of the operating system in a common project and makes it possible to generate executable code.

Prototype

A prototype is an executable file for an experimental target system. This kind of prototype shows the software functions – possibly with different aims and in different forms – in practical use without the later production ECU being necessary.

Workspace

In its tree view, the Workspace Browser, the workspace combines software, hardware and system projects consisting of these in a file set which you can import, export, load and save between different sessions. The various components of INTECRIO are invoked from here.

Graphical framework

The window displayed after INTECRIO is launched. The various INTECRIO components are integrated in this graphical framework.

The Workspace Browser, i.e. the tree view of the workspace, is displayed at the top left. The display window for the various components is at the top right; a message window for messages of the current component at the bottom.

Software configuration and Project Configurator

The modules imported into INTECRIO have to be connected to each other so that the overall system can work. Inputs and outputs (signal sinks and sources) for the connection between the software and hardware have to be created and then the software and hardware connected.

These tasks are carried out within INTECRIO by the *Project Configurator.*

Hardware connection and Hardware Configurator with INTECRIO-RP/VP

The control plant (or the technical process) is represented to the system as a set of sensors and actuators with which the system can be connected via a range of suitable peripherals. These peripherals have to be configured to suit the actual technical process being modeled.

The hardware connection task is carried out within INTECRIO by the Hardware Configurator (or *HC*).

The *Hardware Configurator with INTECRIO-RP* enables the setup and configuration of the respective hardware systems, it loads the parameterized model into the "Controller environment" which is displayed and linked by the interfaces installed in the hardware.

This is how the Hardware Configurator with INTECRIO-RP enables the creation of the logic necessary to ensure that signals received by external sources are acquired at the right place within the Controller model.

The *Hardware Configurator with INTECRIO-VP* enables the setup and configurations of the virtual prototyping hardware systems, it loads the parameterized model into the "Controller environment" which is displayed and linked by the VP-PC hardware system.

This is how the Hardware Configurator with INTECRIO-VP enables the creation of the logic necessary for virtual prototyping.

Operating system configuration

The configuration of the operating system is a very important task when creating a real-time prototype. The operating system determines the processing sequence of the tasks and processes which compete for the processor and, if necessary, toggles between different tasks. All the settings necessary for this are made during operating system configuration.

OS Configurator and OSC

The task of operating system configuration is carried out within INTECRIO by the *OS Configurator* and the OSC Editor. The OSC is part of the OS Configurator, an easy-to-use editor for operating system configuration which provides the user with a quick overview of the system and allows the configuration to be edited in an application-oriented display.

Build process and Project Integrator

The *Project Integrator* (PI) combines all parts of the system – modules and functions, hardware connection, OS configuration etc. – in an executable file, the prototype. In addition to this, an ASAM-MCD-2MC file is generated for the experiment in which the ASAM-MCD-2MC files of the individual modules as well as additional information on the overall project are combined.

Experiment

The function of the prototype is verified in the experiment. Different types of measuring and calibration instruments are available for this purpose.

The experiment is saved on the hard disk, but can be invoked via the Workspace Browser. This means you can quickly set up the ETAS Experiment Environment for a specific task by loading the relevant experiment.

Virtual prototyping

Function developers create virtual prototypes of electronic vehicle functions and test these on a PC. This kind of virtual prototype consists of the following components:

- Automotive Embedded Software (application software and OSEK operating system)
- a plant model (environment system)

Virtual prototyping enables cooperation between function developers on the one hand and system developers and simulation experts on the other at a very early stage of development. Virtual prototyping allows developers to use system models in early process phases. In this way, developers can validate their functions with Model-in-the-Loop technology (MiL). Access to system knowledge (and the relevant models) in early process phases thus creates synergies between function development and system development making the entire development process more efficient.

Daisychain configuration

Different from the other interfaces, a daisychain cannot be configured in INTECRIO. Instead, the daisychain is configured in the daisychain configuration tool, and the configuration file is imported in INTECRIO.

ES930

The ES930 Multi-I/O module is a compact, robust, and powerful tool with numerous input and output channels.

The module extends the functionality of the ES910, which makes it suitable for controlling and analyzing sensors and actuators directly from within a given function model (Simulink®, ASCET, AUTOSAR, C code).

5.1.4 Overview and Targets

You will now learn all about working with INTECRIO using the example of throttle control and some other examples.

The position of the throttle valve influences the amount of air in the air-fuel mixture in a gasoline engine. The throttle valve is part of various functions in engine management. It includes, for example, the control of engine power and thus (at a specific engine speed) the control of the engine torque via the airflow and thus via the throttle valve. If the throttle valve is not completely open, the air induced by the engine is throttled and the generated torque thus reduced. This throttling effect depends on the position and thus on the opening cross section of the throttle valve. Maximum engine torque is attained with a fully open throttle valve.

"Lesson 1: Preparing Modules for INTECRIO" : Make the necessary settings for a Simulink model of throttle control in Simulink and generate suitable code for INTECRIO.

"Lesson 2: Preparatory Steps" : You will create a workspace and set it up.

"Lesson 3: Module, Function, Software System, Environment System" : You will import the modules belonging to throttle control into INTECRIO. You will combine two modules to form one function, establish the signal connections within the function and create interfaces for the connection to the outside world. Next, you will create and set up a software system. Finally, you create an environment system and add the third module.

"Lesson 4: Hardware System" : You will create a hardware system for virtual prototyping.

"Lesson 5: System Project" : You will create a system project. First of all you will integrate the hardware system, software system and environment system in the system project. Then you will connect software system and environment system.

"Lesson 6: Generating the Prototype" : You will configure the operating system, set the options for code generation, and generate the executable prototype for throttle control.

"Lesson 7: Experimenting" : You will experiment with the prototype you have created.

"Lesson 8: ES930 – Example" : You will set up a small ES930 project and test it in an experiment.

What You Learn From Working Through the Tutorial

Once you have completed the tutorial, you are capable of importing modules into INTECRIO and can combine these to form a software system. You can create and configure a hardware system, and combine it with the software system to form a system project. You can configure the operating system, generate an executable prototype and experiment with it.

You can work out the finer details of the INTECRIO functionality with the aid of the online help as you practice your skills during your first real assignments with INTECRIO. In addition, you have sample workspaces and sample experiments which you can use as a quick start to future tasks.

5.2 Lesson 1: Preparing Modules for INTECRIO

The function software of the throttle control consists of several modules which were generated with the behavioral modeling tool (BMT) Simulink. To be able to process the INTECRIO code generated in the BMT, special options have to be set for code generation.

Targets

You make the necessary settings for a Simulink model of throttle control (`throttle_control.mdl`) in Simulink and generate suitable code which is then processed in the following lessons of INTECRIO.

> **NOTE**
>
> How to prepare an ASCET module for working with INTECRIO is described in the ASCET documentation.

Creating the modules is *not* part of the tutorial.

If you want to get more practice, repeat this lesson for the Simulink models `throttle_stimuli.mdl` and `throttle_plant.mdl`.

5.2.1 Overview of the Most Important Concepts in this Lesson

Modules

A module in INTECRIO contains the generic description of a functionality for an ECU. It corresponds, for example, to an ASCET project (ASCET V5.1 or higher) or a Simulink model.

The description files for the interfaces (SCOOP-IX, `*.six`) and data (`*.a2l`) as well as C code are generated for working in INTECRIO from a model created in ASCET or Simulink using code generation.

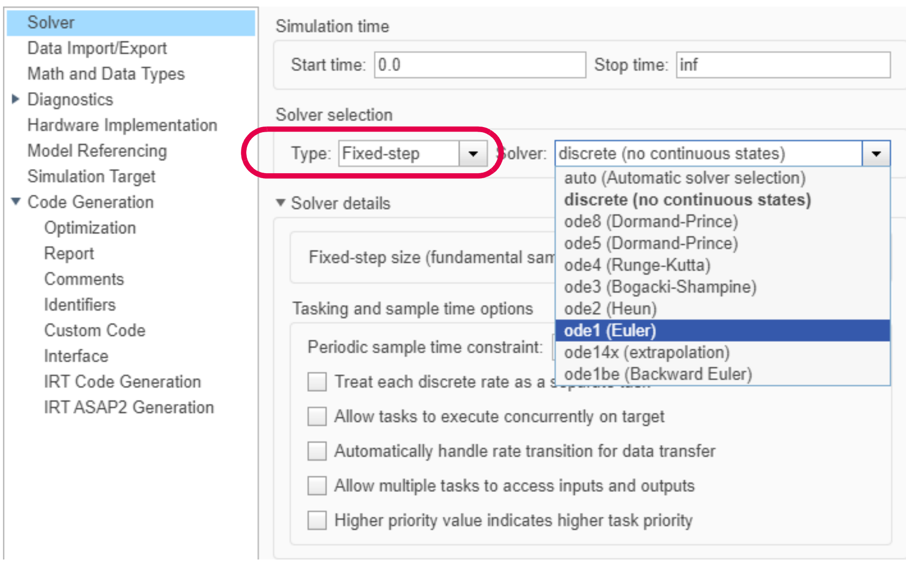
5.2.2 Simulink® Model

The Simulink model `throttle_control.mdl`, with which you will be working in this section, is in the sample file directory (`<sample files>\ Throttle-Demo\SimulinkModel`). It is completely specified. You only need to open it, set the simulation parameters necessary for working with INTECRIO and generate the necessary code.

> **NOTE**
>
> The pictures in this section originate from MATLAB and Simulink R2021b. If you use another version, the user interface may look different.

To generate code for the Simulink model

1. Make backup copies of the model and the `throttle_control_irt_rtw` folder.
2. Open the `throttle_control.mdl` model in Simulink.
3. Open the "Configuration Parameters" window in Simulink.
4. In the "Configuration Parameters" window, go to the "Solver" node.
5. Select a solution algorithm of the type "Fixed-step".

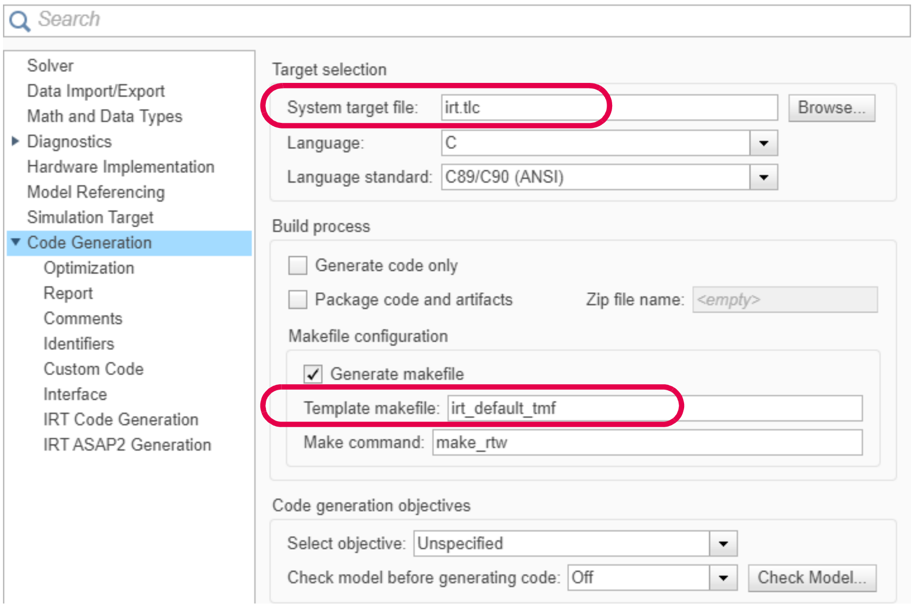


6. In the "Code Generation" node, select the `irt.tlc` target.

   The `irt_default_tmf` template is selected automatically.

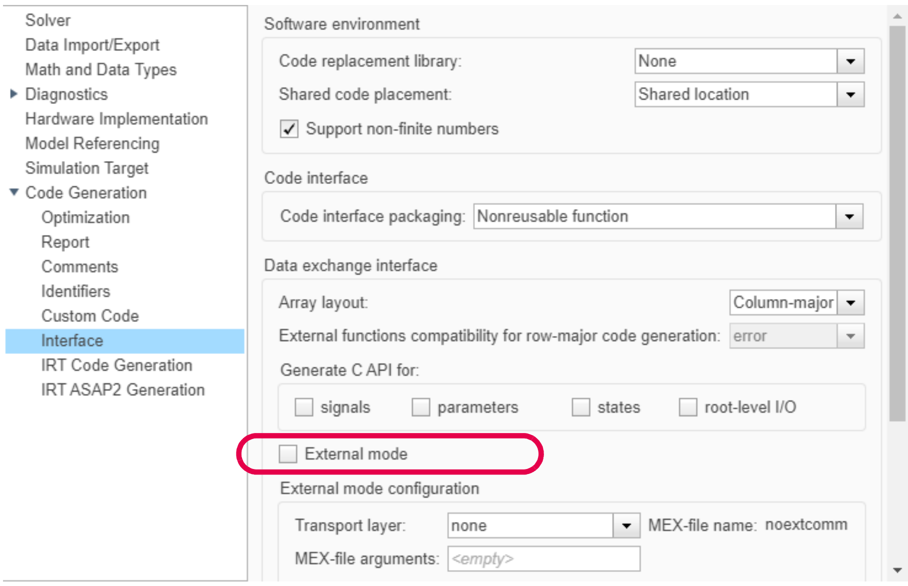
   The target/template combination of `ier.tlc` and `ier_default_tmf` works, too, but you cannot mix `irt*` and `ier*`.

   Generate code only does not have to be activated, it is set implicitly for the INTECRIO targets.



> **NOTE**
>
> The code for INTECRIO is only correctly generated in this combination.

7. Make sure that the `External mode` interface is *deselected/deactivated*.



   When `External mode` is selected/activated, no parameters are included in the ASAM-MCD-2MC file.

8. Generate the code.

The subdirectory `throttle_control_irt_rtw` is created for the generated files in the directory containing the Simulink model. The following generated files are of significance for working with INTECRIO:

- `throttle_control.a2l`

  This is the ASAM-MCD-2MC file generated for working with INTECRIO.
- `throttle_control.six`

  This is the interface description file or SCOOP-IX file.
- `throttle_control_main.c`

  This file provides the main functionality for the integration in INTECRIO. This functionality consists of several `void/void` C functions which are mapped as processes to the underlying operating system.
- `throttle_control_irtmacros.h`

  This header file contains a few `#define` instructions for the combination of several modules. It is processed before all the others using the `gcc -include throttle_control_irtmacros.h` command.

The `throttle_control_irt_rtw` subdirectory contains further files which are generated during code generation with Simulink and Simulink® Coder™. Their significance is detailed in the respective documentation; they are irrelevant when working with INTECRIO.

5.3 Lesson 2: Preparatory Steps

Before you can start, you have to launch INTECRIO and open a workspace in which you would like to work. For purposes of better clarity, we would recommend using an individual workspace for the tutorial. All working steps of this tutorial are executed in this workspace.

Targets: You create the necessary workspace for the tutorial and set it up.

5.3.1 Overview of the Most Important Concepts in this Lesson

Graphical framework

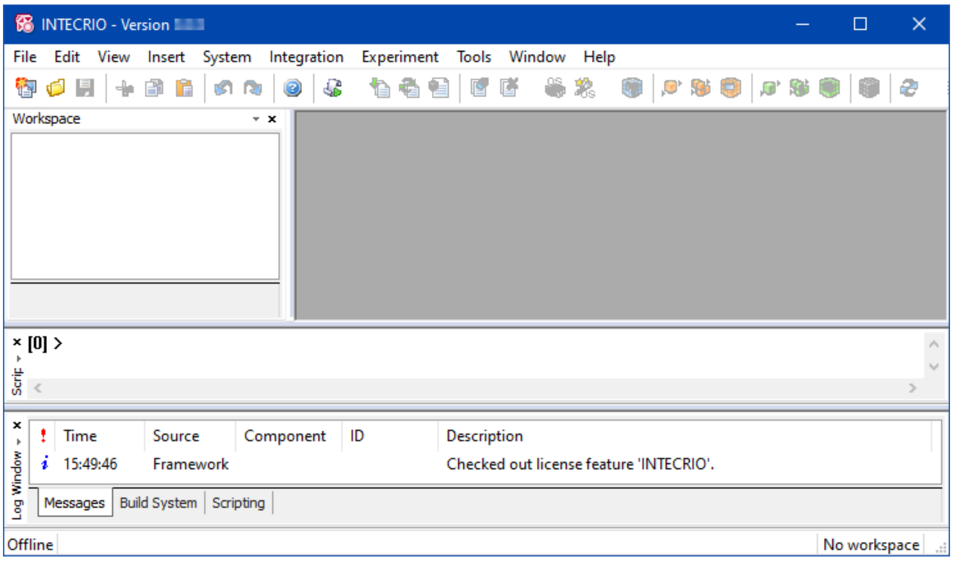
The window displayed after INTECRIO is launched. The various INTECRIO components are integrated in this graphical framework.

Workspace

The workspace combines all data generated when working with INTECRIO. You can invoke all INTECRIO components from the Workspace Browser.

5.3.2 INTECRIO and the Workspace

After program start (**INTECRIO V5.0** icon on the desktop or the option of the same name in the INTECRIO program group in the Windows Start menu), only the graphical framework is shown initially. The "Messages" tab in the "Log Window" lists the components of INTECRIO activated when the program is launched; the other fields are empty. As soon as you have created a new workspace or opened an existing one, it is shown in the Workspace Browser ("Workspace" window) as a tree structure. You can invoke all available components from there or from the menu bar. The different editors are shown in the top right field.



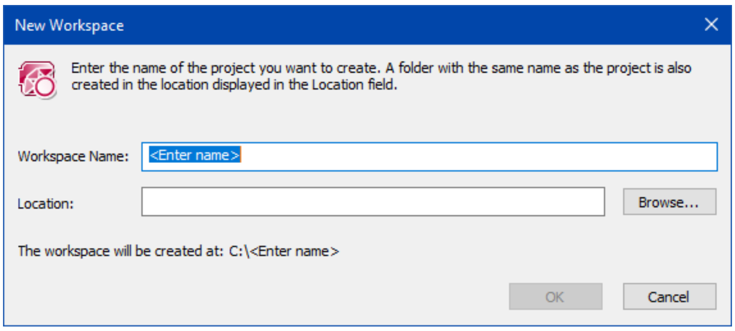
Open the option window using the menu option **Tools > Options**. This is where you can set the options for the overall system (e.g. project integrator settings or settings for the usage of text editors). The online help contains details of the options.

You can now create the workspace. This includes all data occurring when working with INTECRIO. From the tree view of the workspace, modules are integrated, functions generated, hardware configurations and system projects created; you can invoke all necessary editors from here.

<u>To create the workspace</u>

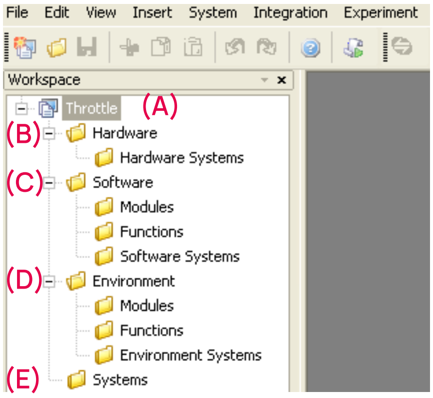
1. Click the **New Workspace** button.

   The "New Workspace" window opens.



2. In the "Workspace Name" field, enter the name `Throttle`.
3. In the "Location" field, enter a path, e.g., `D:\ETASData\INTECRIO5.0\Tutorial`.
4. Click **OK**.

The workspace is stored under the name `Throttle.iow` in the directory `ETASData\INTECRIO5.0\Tutorial\Throttle` and opened in INTECRIO. The Workspace Browser shows the tree structure which, apart from the default entries, is still empty.



The default entries in the Workspace Browser are the following:

A The top entry is the name of the workspace.

B `Hardware` – This folder contains all hardware systems of the workspace. These elements are provided by the Hardware Configurator (HC) and stored in the `Hardware Systems` folder which is available by default.

C `Software` – This folder contains the software modules and AUTOSAR software components belonging to the project (subfolder `Modules`), the functions compiled from the modules (subfolder `Functions`), and entire software systems (subfolder `Software Systems`).

D `Environment` – This folder contains the software modules belonging to the project (subfolder `Modules`), the functions compiled from the modules (subfolder `Functions`), and entire environment systems (subfolder `Environment Systems`).

E `Systems` – This folder contains the system project, i.e. references to the hardware and software belonging to the system, the assignment of the software to the hardware and the operating system configuration.

For each system project to be created in the workspace, an individual entry has to be generated under `Systems`.

To set the properties of the workspace, right-click the name of the workspace in the Workspace Browser and select **Properties** from the context menu. You can make settings in the "Options" window. The online help contains details of the properties of the workspace.

Do not forget to save the workspace at the end of this and each of the following lessons.

5.4 Lesson 3: Module, Function, Software System, Environment System

Targets: You import the modules necessary for throttle control into INTECRIO. You combine two modules to form a function, establish the signal connections within the function and create interfaces for the connection to the outside world. Next, you create a software system and integrate the function. Finally, you create an environment system and add the third module.

5.4.1 Overview of the Most Important Concepts in this Lesson

Modules and environment modules

A *module* in INTECRIO contains the description of the generic functionality of an ECU description. It corresponds, for example, to an ASCET project (ASCET V5.1 or higher) or a Simulink model.

An *environment module* contains the description of the generic functionality of the plant model for virtual prototyping.

Functions

In INTECRIO, a function is a structure object for software systems which does not have its own functionality. Different modules are combined and connected in a function; in this way, they are clearly ordered and easy to use.

Software system

A software system contains the hardware-independent parts of the ECU description (the application software), i.e. the modules, functions and connections the control algorithm consists of.

Environment system

An environment system is used to model the plant model for virtual prototyping. It is built out of modules and functions, the same way as a software system.

Software configuration and Project Configurator

The modules contained in a function or software system have to be connected to each other. Inputs and outputs (signal sinks and sources) for the connection to other components have to be created.

These tasks are carried out within INTECRIO by the *Project Configurator*.

5.4.2 Importing Modules

Modules determine the behavior of the system. They are the smallest modeling unit which can be processed with INTECRIO. They cannot be instantiated, i.e. each module can occur exactly once in a system project – either integrated directly or as part of a function.

The modules which belong to throttle control are supplied with the installation. They are stored in the sample file directory (`<sample files>\ThrottleDemo\SimulinkModel`). If you did not work through lesson 1, you can use the files from the subdirectory `throttle_control_irt_rtw`.

From INTECRIO's point of view, modules always have the same basic structure and have the following interfaces:

- signal sinks (inputs),

- signal sources (outputs),
- activation interfaces (processes).

When modules are imported, only the interface description file (`*.six`) is read in. The `*.c` and `*.h` files that contain the functionality are not required until the executable file is generated (section 5.7); this is also true of the ASAM-MCD-2MC file.

<u>To import a module</u>

In INTECRIO, modules are stored below the **`Software`** folder.

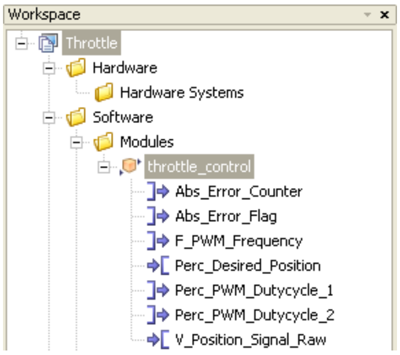
1. In the Workspace Browser, right-click the `Software\Modules` folder and select **Import Module** from the context menu.

   A file selection window opens.

2. Select the path `<sample files>\ThrottleDemo\SimulinkModel\throttle_control_irt_rtw` and the `throttle_control.six` file.

3. Click **Open**.

   The module is imported and displayed in the Workspace Browser.



4. Import the Simulink model `throttle_stimuli`, too.

Under the module, the signal sources ( ) and sinks ( ) are displayed in the tree structure.

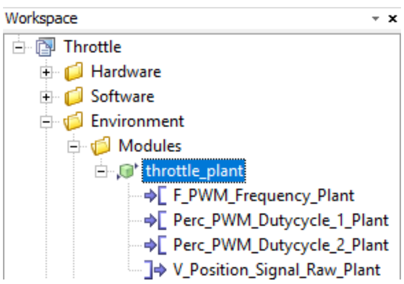
<u>To import an environment module</u>

In INTECRIO, environment modules are stored below the **`Environment`** folder.

1. In the Workspace Browser, right-click the `Environment\ Modules` folder and select **Import Module** from the context menu.

2. In the file selection window, select the `throttle_plant` model and click **Open**.

   The module is imported and displayed in the Workspace Browser.

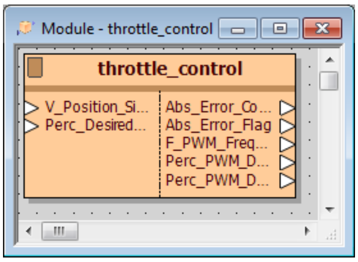


To view a module

Modules and environment modules can be viewed in the graphic editor.

1. Right-click the module `throttle_control` and select **Open in View** from the context menu.

   The graphic editor opens; the module is displayed.



Signal sinks are ordered at the left-hand edge of the block, signal sources on the right. Both are shown by open triangles next to which are the names of the elements. If the name of a signal sink or source is too long, it is cut off; it appears as tooltip if the cursor points to the name. Activation interfaces are not shown.

5.4.3 Creating a Function

Functions are pure structure objects without their own functionality. They are created for better readability and for simple reuse of a group of modules.

A function consists of the following components:

- one or more modules
- connections between inputs and outputs of the modules
- the function interface (inputs, outputs and activation interface)

The inputs and outputs of a function cannot hold their own data or implementations. They are added manually; inputs can be connected to one or more sinks of the modules contained in the function, outputs with exactly one source of a module.

The `Software` folder and the `Environment` folder are strictly separated. It is impossible to insert a module from the `Software` folder branch into a function in the `Environment` folder branch, and vice versa.

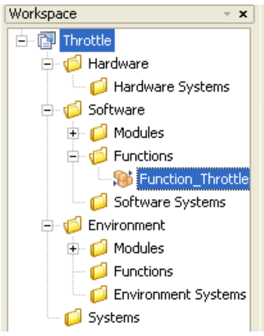
In the first step, you generate an empty function below the `Software` folder.

To create an empty function

1. In the Workspace Browser, right-click the `Functions` folder and select **Create Function** from the context menu.

2. Enter the name `Function_Throttle` in the "Create Function" input window and click **OK**.

   The `Function_Throttle` function is created in the Workspace Browser under the `Functions` folder.

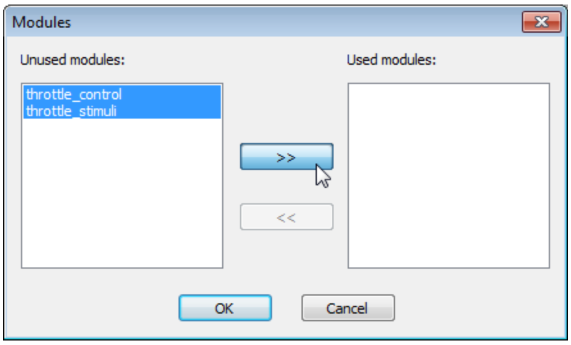


Now add the modules to the function.

To add modules

1. In the Workspace Browser, right-click the `Function_Throttle` function and select **Modules** from the context menu.

   The "Modules" window opens. All modules available below the `Software` folder are displayed in the "Unused modules" field.



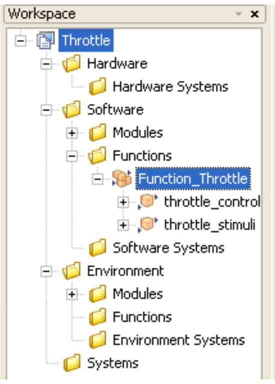
2. Select the modules `throttle_control` and `throttle_stimuli` in the "Unused modules" field.

3. Click the **>>** button.

   The two modules are moved to the "Used modules" field.

4. Click **OK**.

   The two modules are added to the function. They are located under the function in the Workspace Browser.



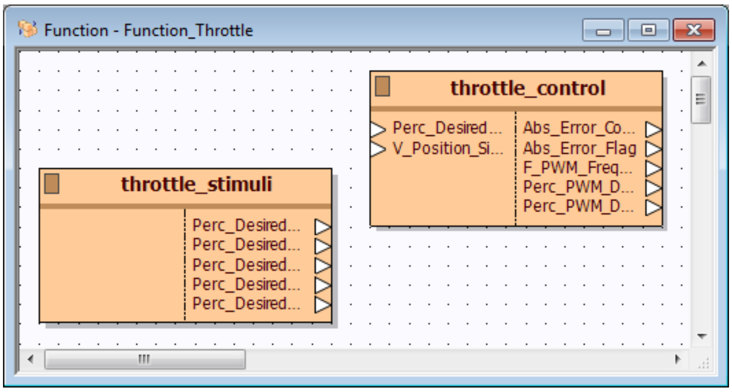
The modules are now part of the function, but the connections for the correct signal flow still have to be established explicitly. There are no implicit connections, e.g. due to identical names of signal sources and sinks, in INTECRIO.

There are static and dynamic connections in Rapid Prototyping; the latter can be modified during program runtime. You will use dynamic connections in the tutorial but not change them in the experiment. For more details on dynamic connections and on working with connections in general, refer to the manual and the online help.

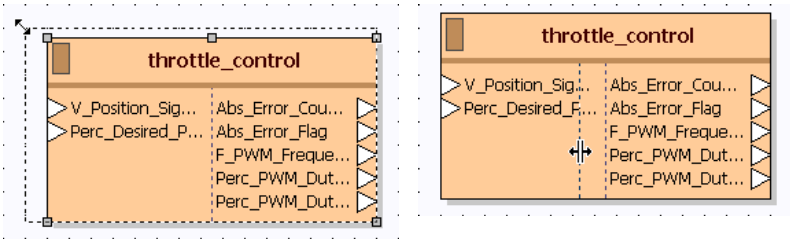
You create the connections in the function view of the graphic editor. Newly created connections are dynamic by default.

<u>To connect the modules</u>

1. Open the `Function_throttle` function in the graphic editor.
2. Move the modules in the graphic editor so that they are clearly structured.

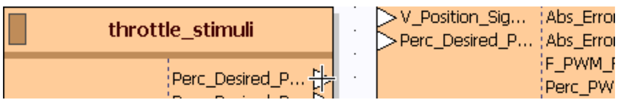


3. If desired, change the module representation.

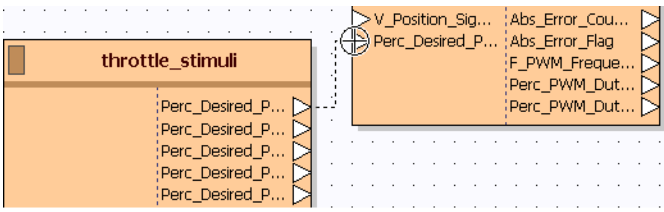


4. Position the cursor over the signal source `Perc_Desired_Position` of the `throttle_stimuli` module.

   The cursor becomes a quadruple arrow.

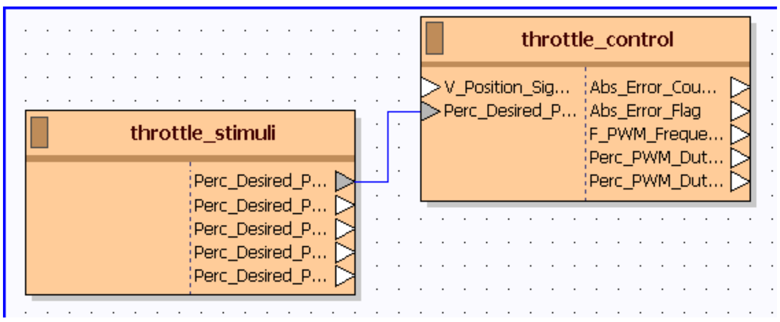


5. Draw a line from the signal source `Perc_Desired_Position` to the signal sink of the same name of the `throttle_control` module.



6. Release the mouse button to establish the connection.

   The connection between the signal source and the signal sink is shown as a line; the triangles are filled. The connections are *not* displayed in the Workspace Browser.



The connections within the function are defined after this step. The connections of the function with the outside world, however, are still missing; signal sources and sinks of the module which are not connected (open triangles in the figure above) do *not* automatically become function interfaces.

You add the function interfaces (signal sources and sinks of the function) manually and connect these with the relevant signal sources and sinks of the modules. The function interfaces do not have their own data or implementations. Every signal sink of a function (see Fig. 5-1, Port `Sink 1`) can be connected to one or more signal sinks of the modules, every signal source of a function (see Fig. 5-1, Port

`Source 1`) can be connected to exactly one signal source of a module contained in the function. It is not possible to establish a second connection to the signal source of the function (red circle in Fig. 5-1).

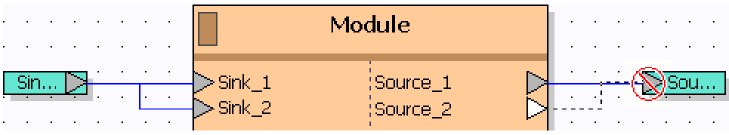


Fig. 5-1 Connections Between the Module and the Ports

To add function interfaces

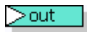
1. Right-click at an empty point in the graphical editor and select **Create Port > Scalar In Port** from the context menu.

   An input port is generated at this point.

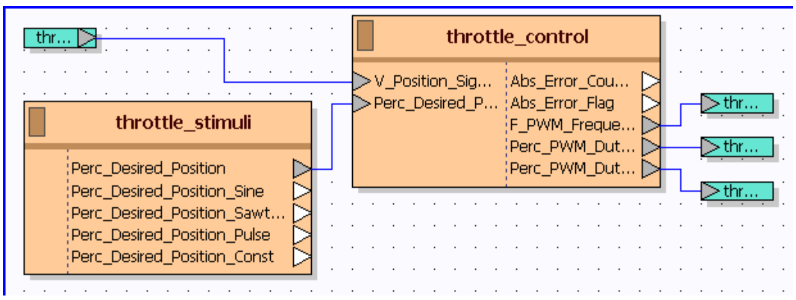
2. Drag the input port to the position you want it.
3. Connect the input port to the signal sink `V_Position_Signal_Raw` of the `throttle_control` module to generate a signal sink of the function.

   The triangle of the input port is filled; the input port is given the name of the sink to which it is connected.

4. Use **Create Port > Scalar Out Port** to generate an output port () and connect it to the signal source `F_PWM_Frequency` of the `throttle_control` module.

   This means you have generated a signal source of the function.

5. Generate the other interfaces of the function shown in the figure.



The function is now ready for use in the software system. Further editing possibilities as well as alternative possibilities of carrying out the tasks described here are described in the online help.

5.4.4 Creating a Software System and an Environment System

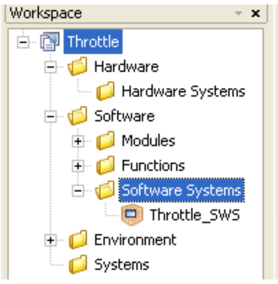
For a virtual prototyping experiment, you need a software system that contains the hardware-independent parts of the ECU description and an environment system that contains the plant model.

As a first step, you create an empty software system.

To create a software system

1. In the Workspace Browser, right-click the `Software Systems` folder and select **Create Software System** from the context menu.
2. In the "Create Software System" window, enter the name `Throttle_SWS` and click **OK**.

   The empty software system is created.

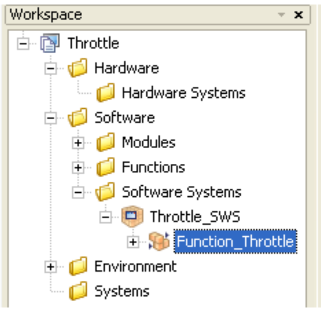


Second, add the function belonging to throttle control into the software system. Use drag & drop in the Workspace Browser for this purpose.

To add modules and functions

1. In the Workspace Browser, drag the `Function_throttle` function from the `Software\Functions` folder to the software system `Throttle_SWS`.

   The function is added to the software system as a reference.

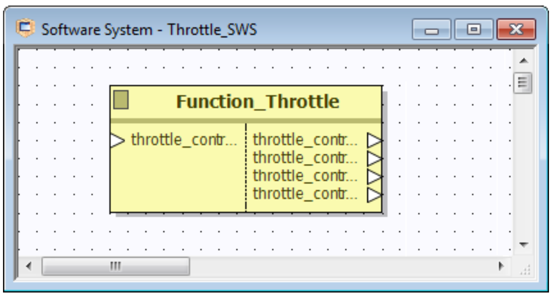


Then create the interfaces to the hardware in the software system. This takes place in the graphic editor you already got to know.

To add interfaces to the software system

1. Double-click the `Throttle_SWS` software system.

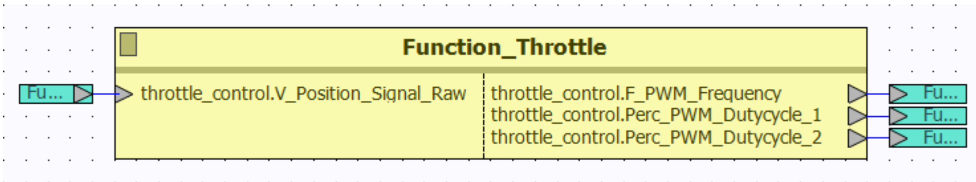
   The graphic editor opens; the software system is displayed.



Now you add external interfaces for all signal sources/sinks, for the connection to the hardware and connect these to the function.

2. Select **Map Ports to Software System** from the context menu of the function.

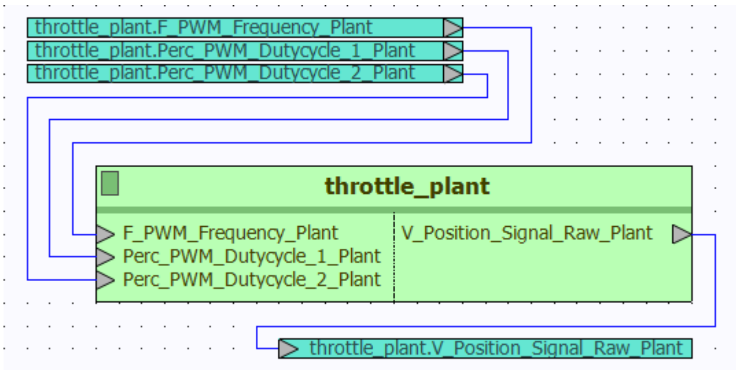
   The result should look like this:



Finally, you create and set up an environment system.

<u>To create and set up an environment system</u>

1. In the Workspace Browser, right-click the `Environment Systems` folder and select **Create Environment System** from the context menu.
2. Name the environment system, e.g., `Environment`.
3. Add the `throttle_plant` module to the environment system.
4. Connect the module ports to the environment system.



5.5 Lesson 4: Hardware System

Targets

You will create a hardware system for virtual prototyping. Virtual prototyping uses your PC as hardware. Therefore, creating the hardware system is very simple. Configuring rapid prototyping hardware devices provides more options and is described in section 5.9.3 "Creating and Setting Up the ES910 Hardware System" on page 139.

5.5.1 Overview of the Most Important Concepts in this Lesson

Hardware system

A hardware system contains the complete description of a hardware topology consisting of descriptions of the relevant ECUs (experimental targets) as well as descriptions of the interfaces between the devices.

Hardware connection and Hardware Configurator with INTECRIO-VP

The hardware connection task is carried out within INTECRIO by the Hardware Configurator (or *HC*).

The Hardware Configurator with INTECRIO-VP enables the setup and configurations of the virtual prototyping hardware systems, it loads the parameterized model into the "Controller environment" which is displayed and linked by the VP-PC hardware system.

This is how the Hardware Configurator with INTECRIO-VP enables the creation of the logic required for virtual prototyping.

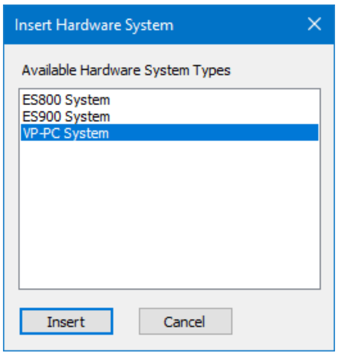
5.5.2 Creating and Setting Up the Hardware System

INTECRIO-VP makes the PC target available for virtual prototyping. The hardware system is created in exactly the same way as hardware systems for rapid prototyping.

To create a hardware system for virtual prototyping

1. Right-click the `Hardware\Hardware Systems` folder and select **Add Hardware System** from the context menu.

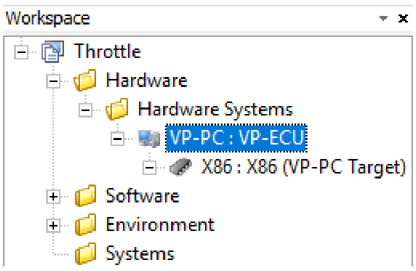
   The "Insert Hardware System" window opens. It contains all available hardware systems, i.e. `ES800`, `ES900` and `VP-PC`.



2. In the "Insert Hardware System" window, select `VP-PC System` and click **Insert**.
3. Confirm the default name with **OK** or enter a name.

   The `VP-PC` system is added and displayed in the Workspace Browser.

   Only one target type is available; it is inserted automatically with the default name `X86`.



5.6 Lesson 5: System Project

Having prepared and imported the software modules necessary for throttle control in lessons 1 and 3 and having created the hardware system necessary for the Rapid-Prototyping experiment in lesson 4, you now create the overall system.

Targets: You create a system project. First of all you will integrate the hardware system in the system project. Then you create a software system and integrate the function and the other modules. Then you establish the signal connections in the software system and between the software and hardware.

5.6.1 Overview of the Most Important Concepts in this Lesson

System project

A system project combines a hardware system, one software system, the mapping of the signals and the configuration of the operating system in a common project and makes it possible to generate executable code.

Hardware system

A hardware system contains the complete description of a hardware topology consisting of descriptions of the relevant ECUs (experimental targets) as well as descriptions of the interfaces between the devices.

Software system

A software system contains the hardware-independent parts of the ECU description, i.e. the modules, functions and connections the control algorithm consists of.

Environment system

An environment system is used to model the plant model for virtual prototyping. It is built out of modules and functions, the same way as a software system.

Integration

Integration consists of the following steps:

a the compilation of models and model parts to create a control algorithm

b the connection of this algorithm with the hardware it is to run on (irrelevant for this lesson)

c the generation of an executable file (part of the next lesson)

Project Configurator

Inputs and outputs (signal sinks and sources) for the connection between the software and hardware have to be created and the software and hardware then have to be connected.

These tasks are carried out within INTECRIO by the *Project Configurator*.

5.6.2 Creating a System Project

First of all you create the structure of the system project.

<u>To create a system project</u>

1. Right-click the `Systems` folder in the Workspace Browser and select **Create System** from the context menu.

   The "Create System" window opens.

2. Enter the name `Throttle_SP` in the "Create System" window and click **OK**.

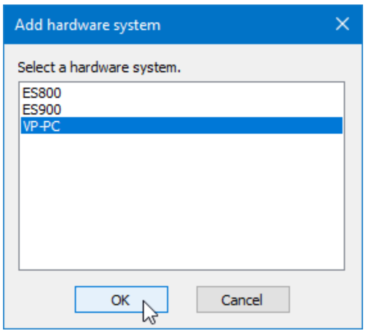
   The system project is created in the workspace.

To integrate the hardware system

You integrate the hardware system you created in section 5.5 into the system project.

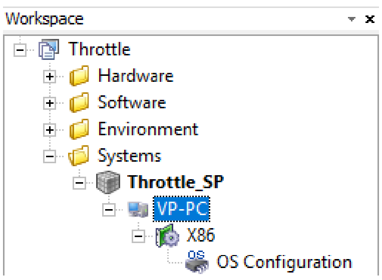
1. In the Workspace Browser, right-click the `Throttle_SP` system project and select **Add Hardware System** from the context menu.

   The "Add hardware system" window opens. It lists all hardware systems available in the `Hardware\Hardware Systems` folder.



2. Select `VP-PC` and click **OK**.

   The hardware system is integrated into the system project. Under the target, the entry `OS Configuration` is automatically generated for operating system configuration.

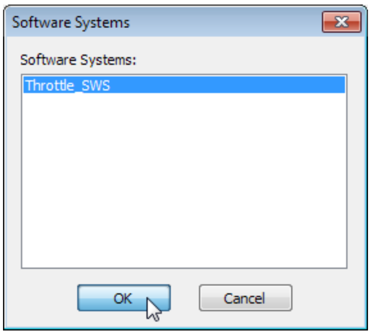


To integrate the software system

You integrate the software system you created in section 5.4.4 into the system project.

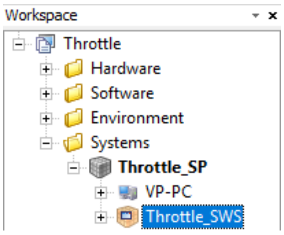
1. In the Workspace Browser, right-click the `Throttle_SP` system project and select **Add Software System** from the context menu.

   The "Software Systems" window opens. It contains all software systems available in the `Software\Software Systems` folder.



2. Select `Throttle_SWS` and click **OK**.

   The software system is added to the system project.

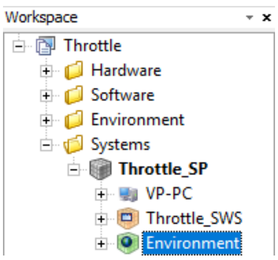


To integrate the environment system

You integrate the environment system you created in section 5.4.4 into the system project.

1. In the Workspace Browser, right-click the `Throttle_SP` system project and select **Add Environment System** from the context menu.
2. In the "Environment Systems" window, select `Environment` and click **OK**.

   The environment system is added to the system project.



5.6.3 Connecting Software System and Environment System

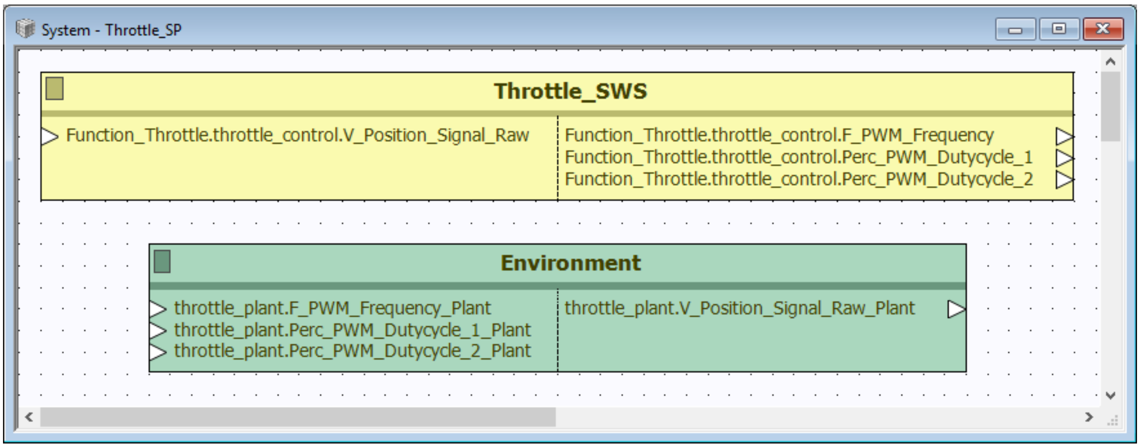
You will now connect the software system and the environment system so that the signal flow is completely specified.

Software system and Environment system shall be connected as shown on Page 113. To do so, you will use the connection wizard.

To connect software system and environment system

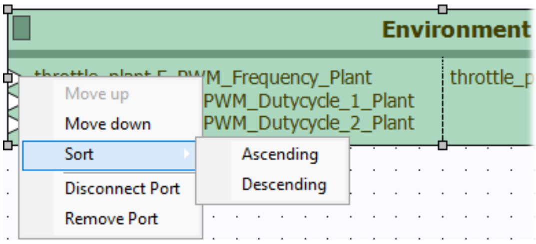
1. Double-click the `Throttle_SP` system project.

   The graphic editor opens; the system project is displayed.



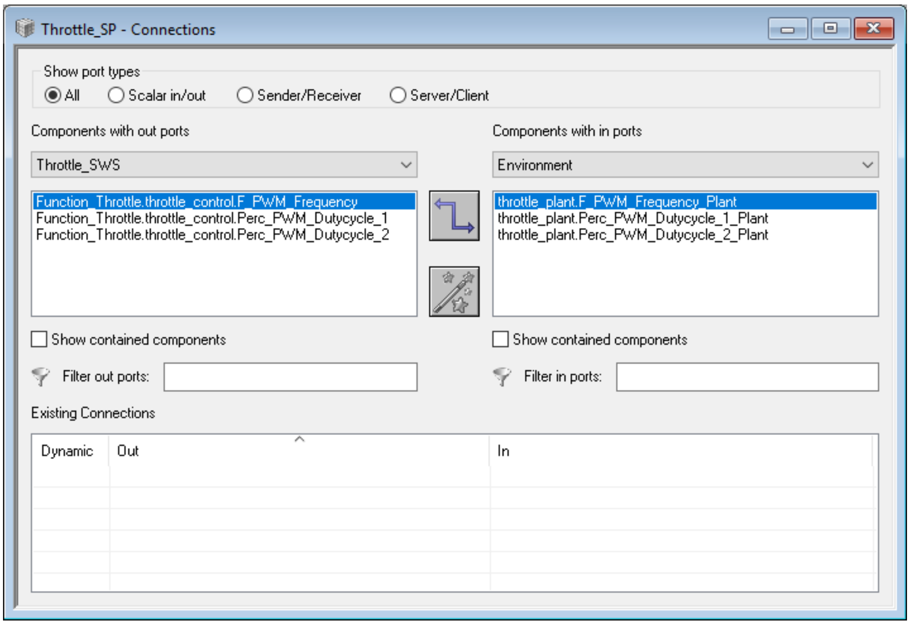
   The software system is displayed as yellow block, the environment system is displayed as green block.

2. If desired, right-click a port and use its context menu to sort the ports.



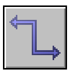
3. In the Workspace Browser, select **Open in Connection View** from the context menu of the system project.

   The "Throttle_SP - Connections" window opens. The "Components with out ports" combo box contains all components with signal *sources*, "Components with in ports" contains all components with signal *sinks*. The panels below list the signal sources/sinks of the selected components.



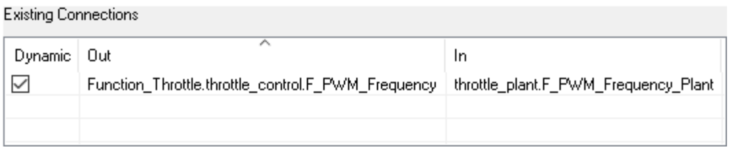
4. In the combo box on the left side, select the software system.
5. In the list below the combo box, select the entry `Function_Throttle.throttle_control.F_PWM_Frequency`.
6. On the right side, select the environment system and the entry `throttle_plant.F_PWM_Frequency_Plant`.

   Once you selected the input and output you want to connect, the button in the middle of the window becomes active.



7. Click the upper button to create the connection.

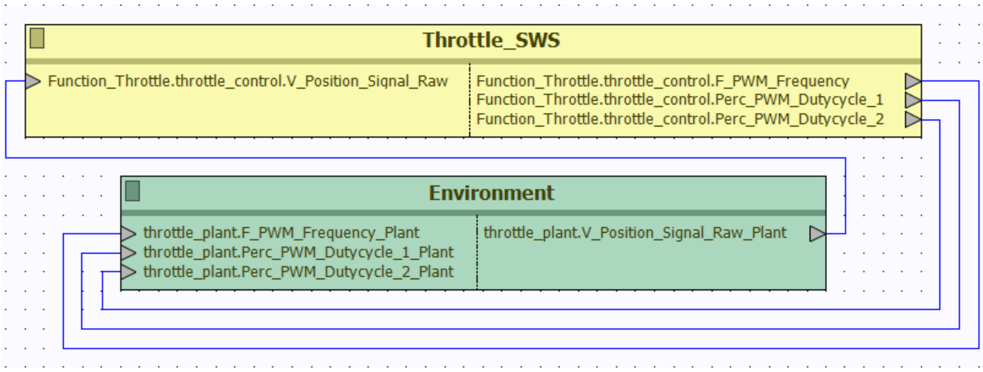
   The connection is shown in the "Existing Connections" table. In the "Dynamics" column you can select whether the connection is dynamic or static.



8. Create also the following connections:

Throttle_SWS	Environment
`*.Perc_PWM_Dutycycle_1`	`*.Perc_PWM_Dutycycle_1_Plant`
`*.Perc_PWM_Dutycycle_2`	`*.Perc_PWM_Dutycycle_2_Plant`
`*.V_Position_Signal_Raw`	`*.V_Position_Signal_Raw_Plant`

The connections between software system and environment system look like this:



This is the end of the work involved in creating the system project. The next steps are the configuration of the operating system and the generation of the prototype.

5.7 Lesson 6: Generating the Prototype

Targets: You configure the operating system using the OS-Wizard. You set the options for code generation and finally generate the executable prototype for throttle control.

5.7.1 Overview of the Most Important Concepts in this Lesson

Integration

Integration consists of the following steps:

a the compilation of models and model parts to create a control algorithm
b the connection of this algorithm with the hardware it is to run on
c the generation of an executable file

Step a) is part of the previous lesson.

Operating system configuration and OS Configurator

The operating system determines the processing sequence of the tasks and processes which compete for the processor and, if necessary, toggles between different tasks. All the settings necessary for this are made during configuration. The task of operating system configuration is carried out within INTECRIO by the *OS Configurator*.

OS Configurator (OSC)

The *OS Configurator* OSC is an easy-to-use editor for operating system configuration which provides the user with a quick overview of the system and allows the configuration to be edited in an application-oriented display.

Build process and Project Integrator

The *Project Integrator* (PI) connects all parts of the system to an executable file, the prototype. An ASAM-MCD-2MC file is generated for the experiment.

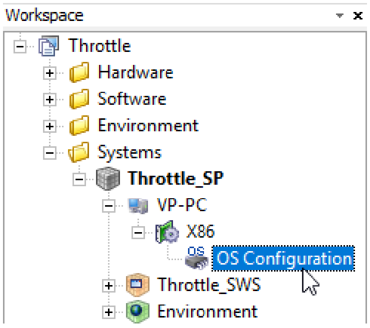
Prototype

A prototype is an executable file for an experimental target system. This kind of prototype shows the software functions – possibly with different aims and in different forms – in practical use.

5.7.2 Configuration of the Operating System

Before you generate the executable prototype, the operating system has to be configured so that the processing sequence of the different processes is defined.

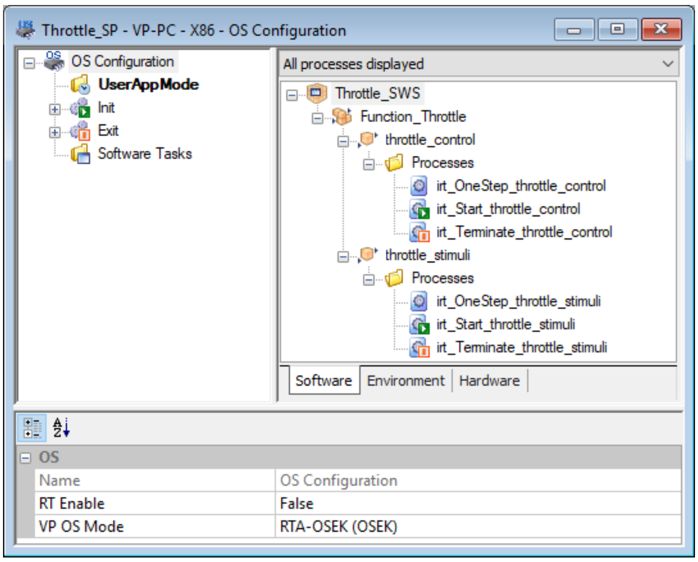
The entry `OS Configuration` was created under the target in the system project for OS configuration when the hardware system was integrated (see "To integrate the hardware system" on page 109).



To start the OS Configurator

1. Double-click `OS Configuration` in the Workspace Browser.

   The OSC opens.



The OSC interface is divided into three parts: in the top left field, you can see the OS configuration view; at the top right, there are three tabs that display the hierarchical structures of the software, environment, and hardware system (software, environment, and hardware view). The input field for the attributes of the object selected in the OS configuration view is under these two fields.

At the top level of the OS configuration view, there is the operating system ( ), and underneath, the application mode ( ). One *Init* and one *Exit* task are also created automatically directly under the operating system; these are used by all application modes. The tasks have the following significance:

- **Init** – executed once when an application mode is started. Interrupts are deactivated during task runtime.
- **Exit** – executed once when the simulation is ended. Interrupts are deactivated during task runtime.

In addition, you can create timer tasks and software tasks. Timer tasks are created under the application mode, software tasks in the `Software Tasks` folder.

To set up the application mode

The application mode can only be renamed; there are no other possible settings. Use the specified names for the tutorial.

You need a timer task `Task_10ms` for throttle control.

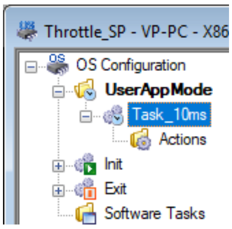
To create and set up the timer tasks

1. Right-click the application mode and select **Add Timer Task** from the context menu.

   A naming window opens.

2. Assign the task name `Task_10ms` and click **OK**.

   The task is created in the OS configuration view under the application mode. The `Actions` folder for this task is created automatically.

   With *rapid* prototyping targets, a timer task has a second folder named `Events`.

3. Select the new task.

   The input field shows the default values of the task attributes.

Timer Task	
Name	Task_10ms
Task ID	not yet assigned
Priority	0
Period	0.01
Delay	0
Execution Budget	0
Max. Number of Activation	1
Monitoring	False
Exclude from Tracing	True

4. Enter the following attributes:

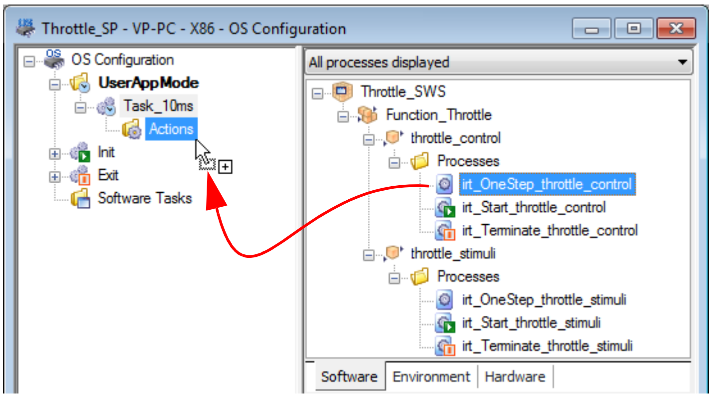
Attribute	Description	Value
`Priority`	Determines the processing sequence of the activated tasks	`116`
`Period`	Specifies after what time period in seconds the task should be reactivated.	`0.01`
`Delay`	The task is not activated until the specified time (in seconds) has elapsed.	`0.0`
`Execution Budget`	Defines the maximum allowed execution time of this task in seconds. A value of 0 turns execution time monitoring off.	`0`
`Max. number of activation`	Specifies how often the task can be activated in parallel.	`1`
`Monitoring`	Determines whether monitoring variables are generated for the task.	`True`
`Exclude from Tracing`	If `True`, the system configures this task for not being traced with RTA-Trace.	`True`

For a detailed description of all attributes, see the INTECRIO online help or the INTECRIO user guide, section "The OSC Editor".

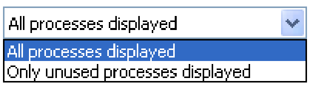
Next, assign the different processes to the tasks. You need the "Software" and "Environment" tabs in the right-hand field of the OSC.

<u>To assign processes to tasks</u>

1. Extend the OS configuration view until all the tasks are visible.
2. Extend the view in the "Software" tab so that the processes of all modules involved are visible.



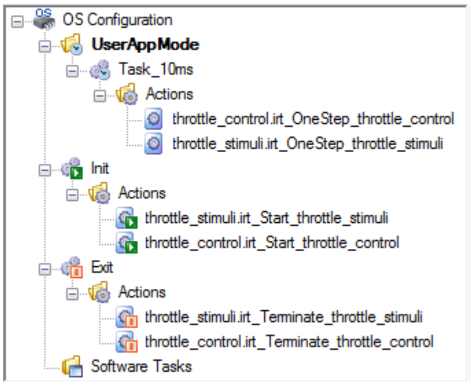
3. In the combo box above the tab, select whether all processes or only unused processes are displayed.



4. Drag the `irt_OneStep_throttle_control` process from the "Software" tab to the OS configuration view and store it in the `Actions` folder of the `Task_10ms` task.

   The process is assigned to the task.

5. Assign the remaining processes to the tasks as shown below.



The sequence in which the processes are assigned to a task corresponds to the processing sequence.

6. If necessary, drag the processes in the OS configuration view to another position so that you end up with the sequence shown.
7. Change to the "Environment" tab and assign the processes of the environment system to suitable tasks.

You have now concluded the configuration of the operating system.

5.7.3 The Build Process

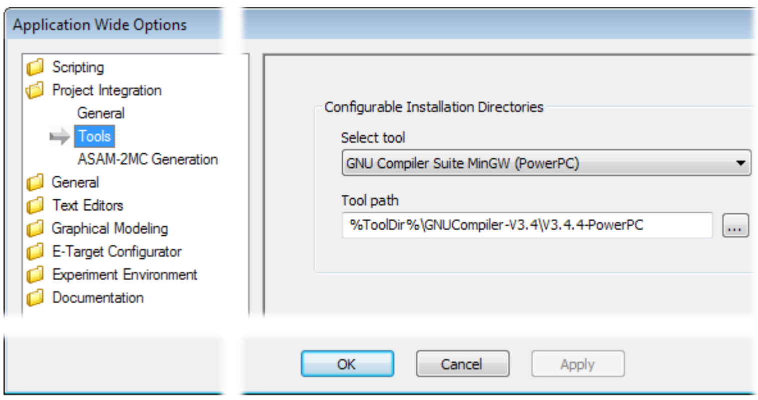
The Build process runs more or less automatically. You simply have to set a few options and start the process. Three kinds of build options exist: application-wide (open with **Tools › Options**), workspace-wide (context menu **Properties** of the workspace) and project-wide (context menu **Properties** of the system project). In the tutorial, you will check the compiler path and set the build options.

To check/set the compiler path

The compiler paths are usually entered automatically in the INTECRIO options. If required, you can change the paths. Proceed as follows.

1. Select **Tools › Options** .

   The "Application Wide Options" window opens.
2. Open the `Project Integration` folder and the `Tools` group.



3. Select a compiler from the "Select tool" combo box.

4. Check if the path of this compiler has been entered correctly in the "Tool Path" field.

   The `GNU Compiler Suite (PowerPC)` is delivered with INTECRIO; the path is predefined in the "Tool Path" field.

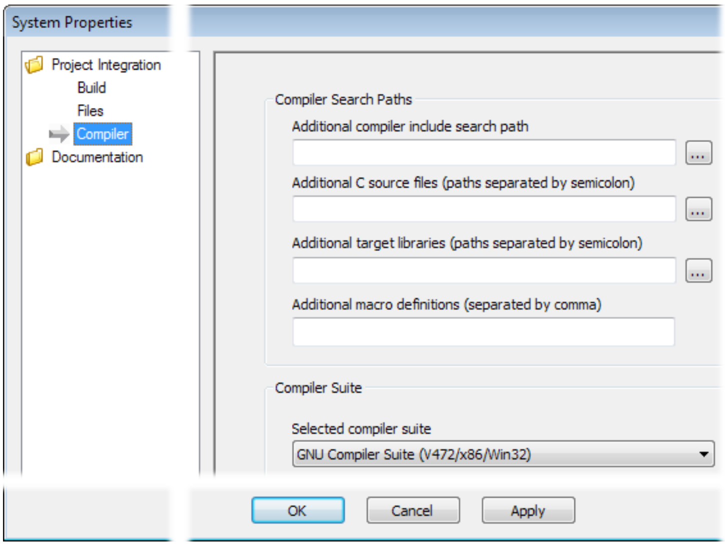
   Other compilers have to be installed separately.
5. Close the window with **OK**.

To select a compiler

In the system project options, you can use other compilers (e.g., the compiler of Microsoft Visual Studio 2008) – provided they are installed on your PC –instead of the predefined GNU Compiler. If you want to use the predefined GNU Compiler, you *do not* need to change the setting.

1. Right-click the `Throttle_SP` system project in the Workspace Browser and select **Properties** from the context menu.

   The "System Properties" window opens.
2. Open the `Project Integration` folder and the `Compiler` group.

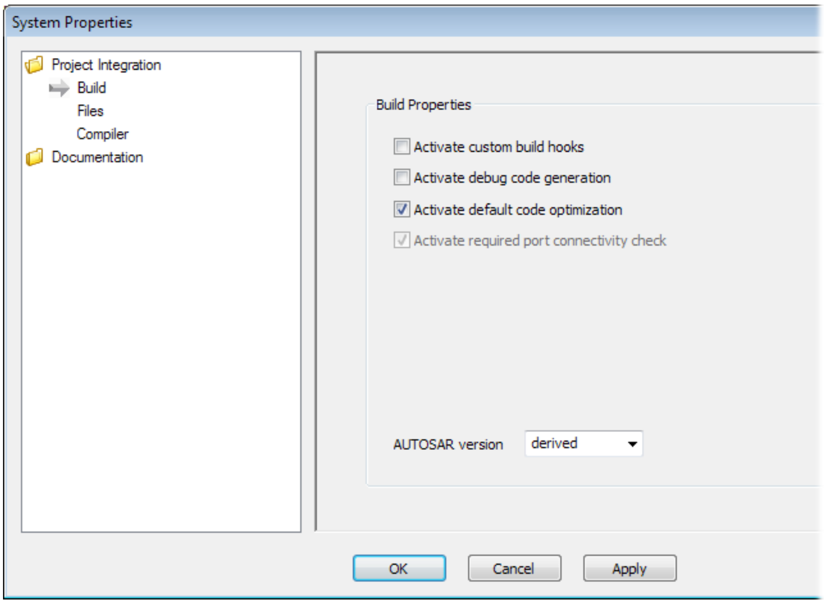


3. Select a compiler from the "Selected compiler suite" combo box.
4. If necessary, enter search paths in the fields of the "Compiler Search Paths" area.
5. Close the window with **OK**.

To set Build options

1. In the Workspace Browser, right-click the `Throttle_SP` system project and select **Properties** from the context menu.

   The "System Properties" window opens.



2. In the left-hand field, open the `Project Integration` folder and the `Files` group.

   The options of the group are displayed in the right-hand field.
3. In the "Result Files Basename" field, enter the basic name `Throttle` for the executable file and the ASAM-MD-2MC file of the prototype.

   Use the default settings for the options in the other groups.
4. Accept the settings with **OK**.

To generate the executable file

1. In the Workspace Browser, right-click the system project `Throttle_SP` and select **Build** from the context menu.

   Existing results of a Build process which are still valid are *not* regenerated.

Or

2. Select **Rebuild** from the context menu.

   Files generated previously are deleted; *all* results are regenerated.

   Information on the current steps is displayed in the "Log Window" field in the "Messages" tab. A progress bar in the bottom pane indicates the progress.

You have now generated an executable prototype. In the next section, you will be experimenting with this prototype.

5.8 Lesson 7: Experimenting

Targets: You will test throttle control in the experiment. You will become acquainted with the experiment environment of INTECRIO with the different measuring and calibration instruments.

> **NOTE**
>
> Beginning with V5.0.4, INTECRIO supports ETAS Experiment Environment V3.8.4. ETAS Experiment Environment V3.7 is *no longer supported*.
>
> ETAS Experiment Environment V3.9 is *not supported*.
>
> Instead of the ETAS Experiment Environment, you can use INCA V7.4 with INCA-EIP to experiment with an ES800 hardware system.

5.8.1 Overview of the Most Important Concepts in this Lesson

Experiment

The function of the prototype is verified in the experiment. Different types of measuring and calibration instruments are available for this purpose.

The experiment is saved on the hard disk, but can be invoked via the Workspace Browser. This means you can quickly set up the INTECRIO experiment environment for a specific task by loading the relevant experiment.

Virtual prototyping

Function developers create virtual prototypes of electronic vehicle functions and test these on a PC. This kind of virtual prototype consists of the following components:

- Automotive Embedded Software (application software and OSEK operating system)
- a plant model (environment system)

Virtual prototyping enables cooperation between function developers on the one hand and system developers and simulation experts on the other at a very early stage of development. Virtual prototyping allows developers to use system models in early process phases. In this way, developers can validate their functions with Model-in-the-Loop technology (MiL). Access to system knowledge (and the relevant models) in early process phases thus creates synergies between function development and system development, making the entire development process more efficient.

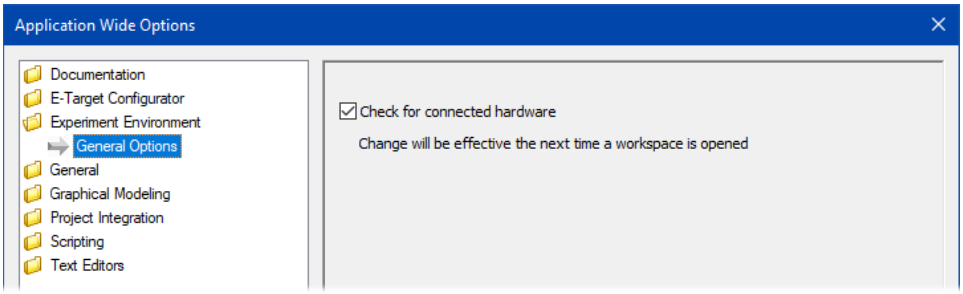
5.8.2 Preparations

To prepare the experiment, open the experiment environment and set a few options.

<u>To open the experiment environment</u>

The experiment does not have its own entry in the Workspace Browser.

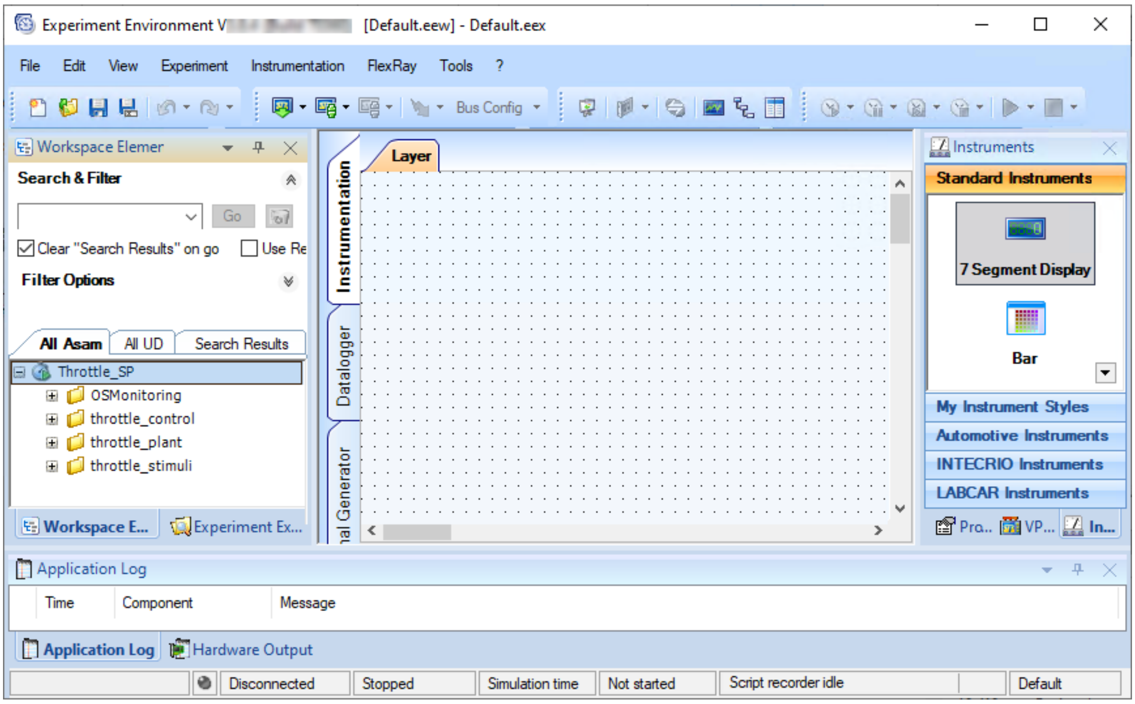
1. Make sure that the application-wide experiment option **Check for connected hardware** is activated.



   If it is not, you cannot open the experiment environment.

2. In the Workspace Browser, right-click the system project `Throttle_SP`.
3. Select **Open Experiment** from the context menu.

   The experiment environment of INTECRIO opens in its own window. The experiment is loaded in the experiment environment.



The experiment has not yet been set up. This is why only the measurement and calibration variables of the system project are displayed in the element list ("Workspace Elements" window, "All ASAM" tab), sorted in the following folders below the system project name.

- `throttle_control` and `throttle_stimuli` – contain the measurement and calibration variables of the respective modules
- `throttle_plant` – contains the measurement and calibration variables of the environment module.
- `OSMonitoring` – contains monitoring variables

During this lesson, you will set up the experiment you need for throttle control.

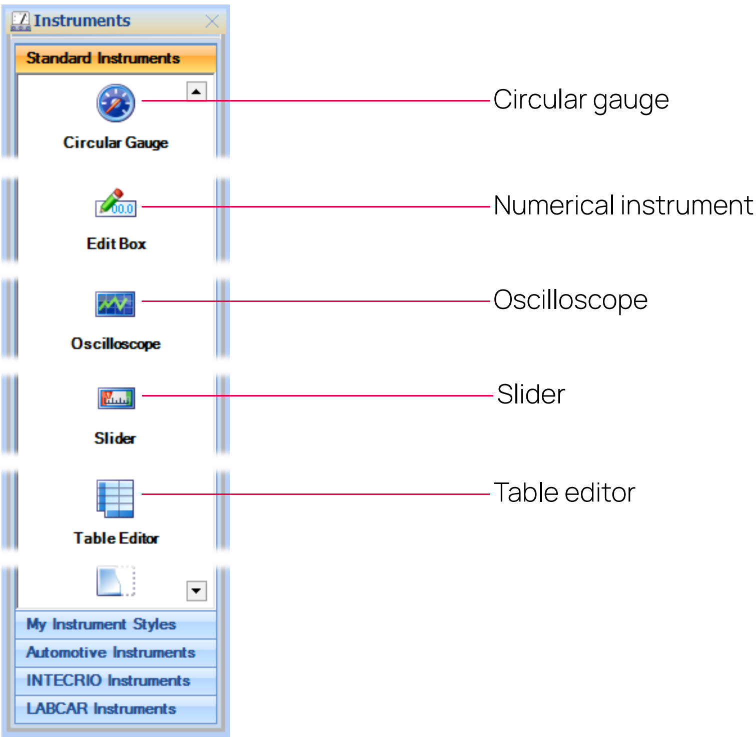
5.8.3 Setting up the Experiment

You set up the experiment by creating the instruments and selecting variables which are to be calibrated or measured for throttle control. You will measure or calibrate the following variables:

Variable	Explanation
`Perc_Desired_Position`	desired position of the throttle valve
`Perc_Actual_Position`	actual current position
`K_S_Value, Td_S_Value, Ti_S_Value, Tv_S_Value`	control parameters of the PID controller
`Abs_Error_Counter`	counts errors
`error_tolerance_yData`	tolerance values for error detection
`Stimulus_Select_Value`	selects stimulus for desired throttle position
`dT`	time between two task activations
`grossRunTime`	total runtime of the task

5.8.3.1 Creating Measuring and Calibration Instruments

The "Instrumentation" window contains all the available instruments. You need only a few standard instruments for this lesson.



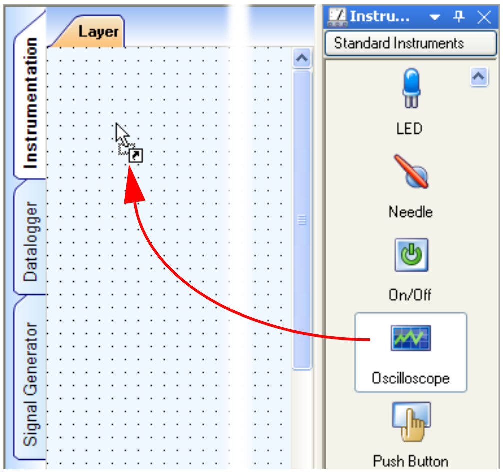
You determine whether an existing numerical instrument or a slider is used as measuring or calibration instrument by assigning a measurement variable or a calibration variable. The table editor is a pure calibration instrument, circular gauge and oscilloscope are pure measuring instruments.

For the throttle control experiment, you need one oscilloscope, seven numerical instruments, a slider, a circular gauge, and a table editor. Oscilloscope, circular gauge, table editor and slider have their own windows; the numerical instruments are arranged in two windows.

> **NOTE**
>
> The oscilloscope is the only instrument that can accept more than one variable. All other instruments can only contain one variable; this is why you sometimes have to create several instruments of one type.

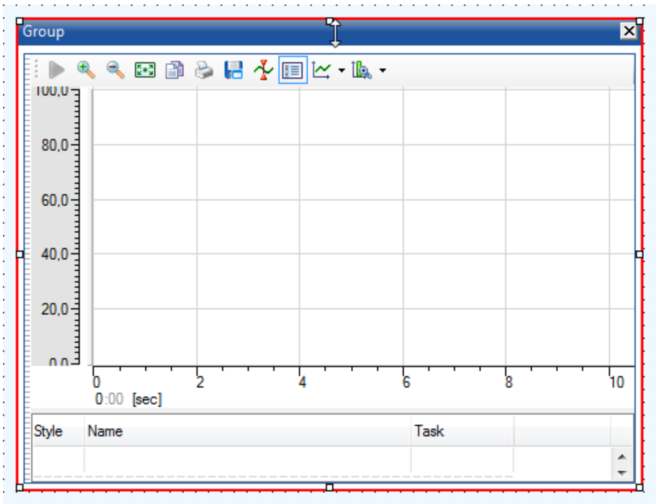
To create the oscilloscope

1. Drag the **Oscilloscope** icon from the "Instrumentation" window to the "Instrumentation" field.



The new oscilloscope opens.

2. Adjust the size of the oscilloscope.



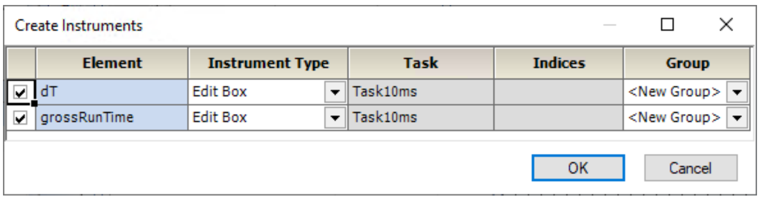
3. Create a circular gauge, a table editor and a slider in the same way.

To add numerical instruments

You create the numerical instruments in two groups, one for measuring instruments, one for calibration instruments. This is done via the "Workspace Elements" window instead of the "Instrumentation" window.

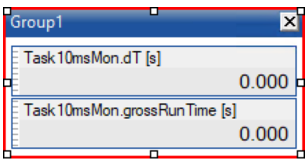
1. In the "Workspace Elements" window, select the OS monitoring variables `grossRunTime` and `dT`.
2. From the context menu of the variables, select **Measure/Calibrate**.

   The "Create Instruments" window opens.



   To create a new measurement window with numerical instruments, the type `Edit Box` and the group `<New Group>` must be selected for both variables.
3. Confirm your selection with **OK**.

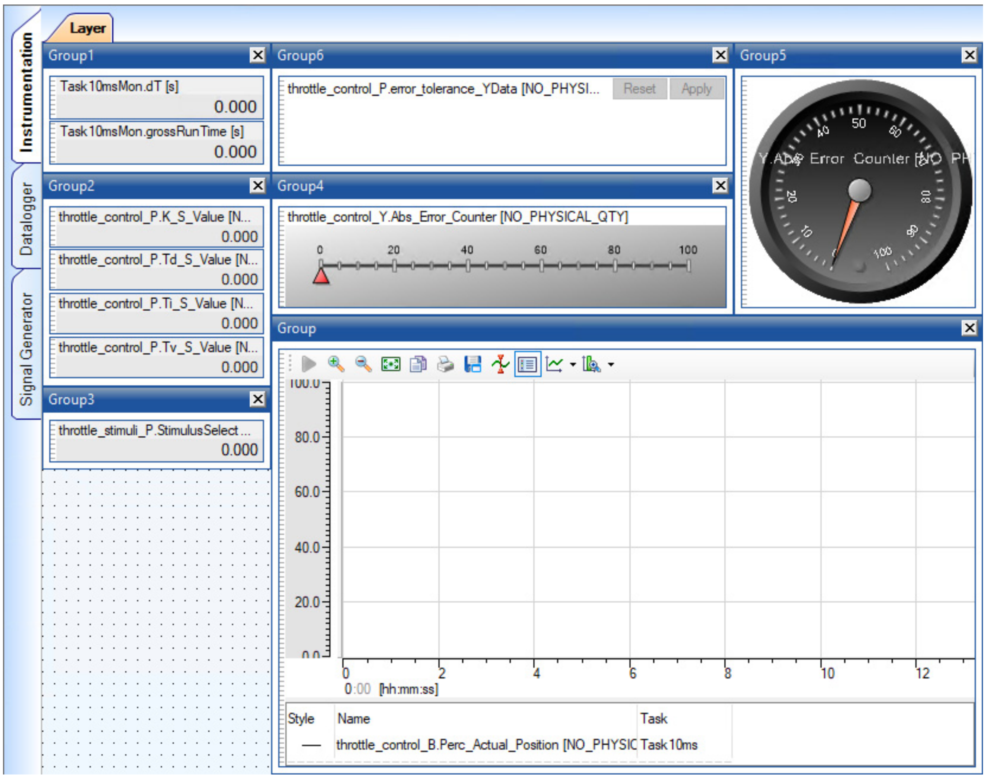
   A measurement window with two numerical instruments opens. The variables are assigned to the instruments.



4. Now create a calibration window for the parameters `K_S_Value`, `Td_S_Value`, `Ti_S_Value`, `Tv_S_Value` and `Stimulus_Select_Value`.

   Instead of using the **Measure/Calibrate** context menu, simply drag the variables to an empty place in the "Instrumentation" field.

Once you have added all instruments, the "Instrumentation" field may look as follows:



5.8.3.2 Assigning Measurement and Calibration Variables

Before you can measure and calibrate, you have to assign the measurement and calibration variables to some of the instruments. In the element list, measurement variables are indicated by a filled red square ( ); calibration variables by a filled blue circle ( ).

To assign measurement and calibration variables

1. Drag the measurement variable `Perc_Desired_position` from the "Workspace Elements" tab to the oscilloscope.

   This assigns the measurement variable to the oscilloscope.

2. Assign variables to the remaining instruments in accordance with the table.

Variable	Instrument
`Perc_Actual_Position`	Oscilloscope
`Abs_Error_Counter`	circular gauge/slider
`error_tolerance_yData`	Table editor

5.8.3.3 Setting Up Measuring and Calibration Instruments

You can set up the instruments before or during simulation. Only the most important possible settings are named in the tutorial. For a more detailed description of the possible settings, refer to the INTECRIO online help.

To set up a numerical instrument

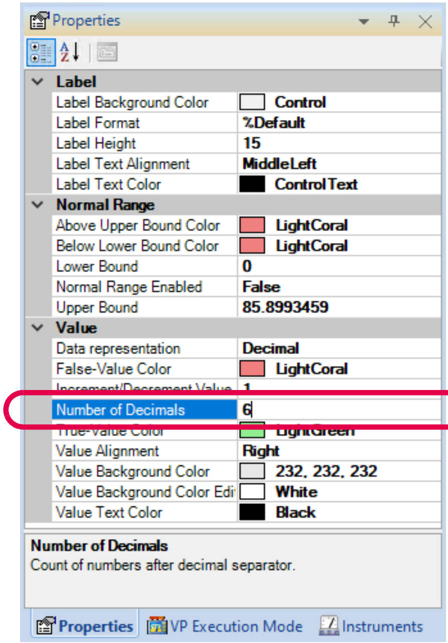
Especially for the display of `dT` and `grossRunTime`, it is recommended that you adjust the display of the numerical instruments to at least six decimals.

1. Click in an instrument.

   It is *not* sufficient to highlight the window containing the instrument.

2. Go to the "Properties" window.

   This window shows the instrument properties.



3. In the "Number of Decimals" field, change the number of decimal places.

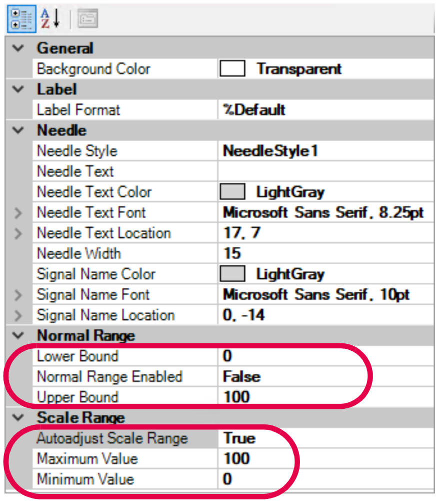
   The change is immediately effective.

To set up another instrument, click in the new instrument and go to the "Properties" window again.

To set up the circular gauge

The circular gauge is created with a few presettings which you should change so that the measurement variable `Abs_Error_Counter` is displayed in a suitable form.

1. Open the circular gauge properties in the "Properties" window.



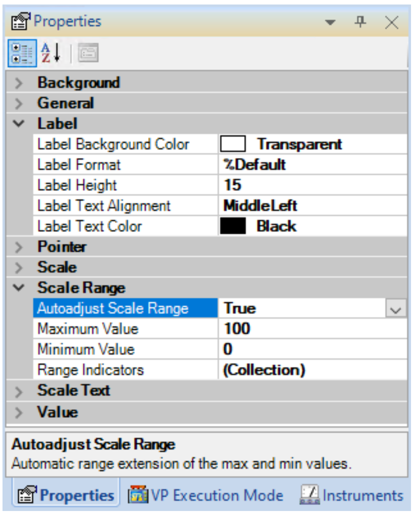
2. Set the "Normal Range Enabled" field to true if you want to be notified when the value exceeds or falls below the normal range.
3. In the "Upper/Lower Bound" fields, enter the limits for the normal range.
4. In the "Autoadjust Scale Range" field, select whether you want an automatic scale adjustment (`true`) or not (`false`).

   If `true` is set, the maximum of the display is increased if necessary. An automatic decrease if the displayed variable is permanently below the current maximum does not exist, this adjustment must be performed manually.
5. In the "Maximum/Minimum Value" fields, enter the limits for the display.

<u>To set up a slider</u>

The slider is created with a few presettings that you should change so that the measurement variable `Abs_Error_Counter` is displayed in a suitable form.

1. Open the slider properties in the "Properties" window.



   The fields "Autoadjust Scale Range" and "Maximum/Minimum Value" have the same meaning as the corresponding fields in the circular gauge properties.

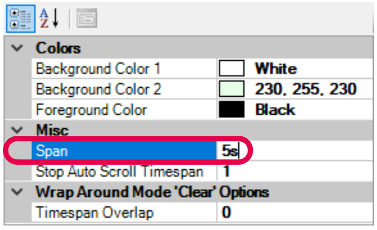
For the table editor, use the default settings.

To set up the oscilloscope

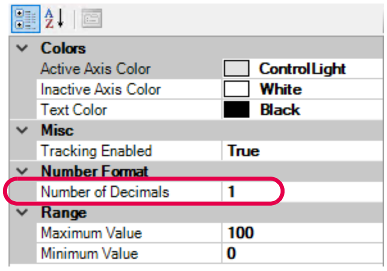
Only the most important possible settings for the axes are mentioned here. Use the default settings for grid, legend and channels.

1. In the oscilloscope, click the X axis.
2. Go to the "Properties" window.

   This window shows the settings for the X axis.



3. In the "Span" field, enter `5s` to specify a time span of 5 seconds.
4. Open the Y axis settings in the "Properties" window.



5. In the "Number of Decimals" field, enter `2` to limit the display to a maximum of two decimal places.

You will not change the display areas of the axes until you are actually measuring as you will then be able to take the currently displayed values into consideration.

5.8.3.4 Saving Measuring and Calibration Instruments

Once you have made all the settings, save the entire arrangement, i.e. the experiment, so that you can easily reestablish it later.

To save an experiment

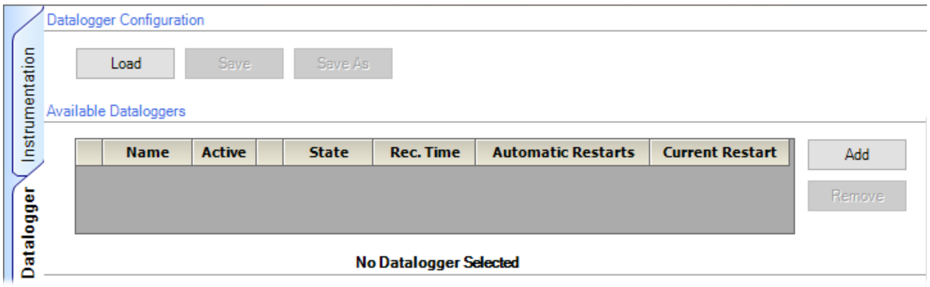
1. Select **File > Save Experiment**.

   The arrangement of measurement and calibration windows is saved in the `Default.eex` file.

   You find the file in the experiment directory of the workspace, e.g., `ETASData\INTECRIO5.0\Tutorial\Throttle\` **`EE\Experiments\System\Default\Default.eex`**.
2. Select **File > Save Experiment As** to save the experiment with an arbitrary name and location.

5.8.4 Configuring Data Acquisition

The data logger ("Datalogger" field) of the INTECRIO experiment environment is used to acquire data.

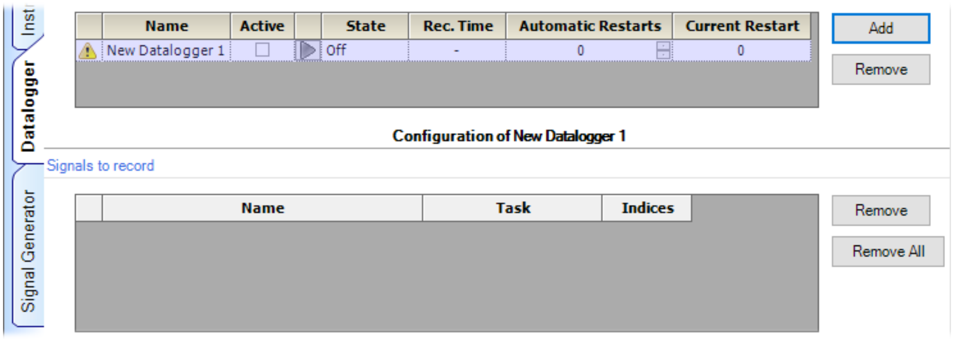


First of all, you create a data logger.

<u>To create a data logger</u>

1. In the "Available Dataloggers" field, click the **Add** button.

   A new data logger is created.
2. Click in the "Name" column to edit its name.



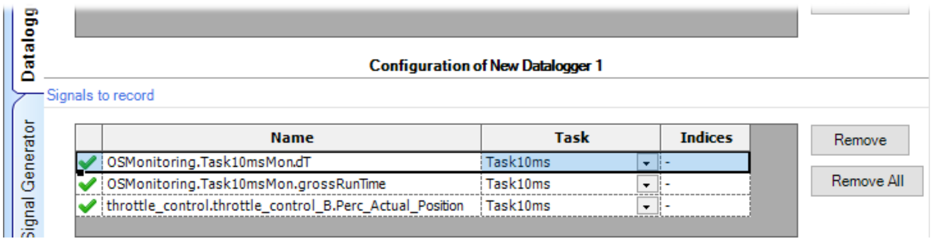
Next, you will configure acquisition. Define the following properties:

- the measurement values to be acquired
- the log file in which the acquisition is to be saved
- the time span over which you want to acquire measurements

<u>To select measurement variables to be acquired</u>

1. In the "Workspace Elements" window, select one or more measurement variables.
2. Right-click a selected variable and select **Add to Datalogger** from the con-text menu.

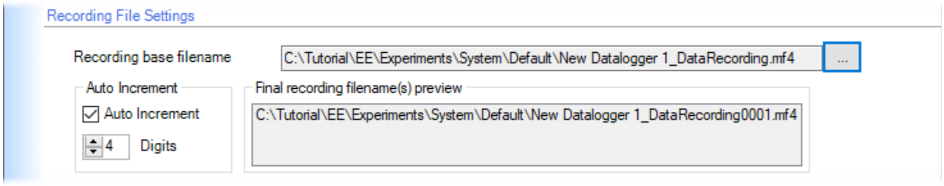
   The measurement variables are added to the "Signals to record" field of the data logger.

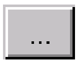


The acquisition task is set to the only task of the system project.

To determine a log file

The log file is specified in the "Recording File Settings" field of the data logger.



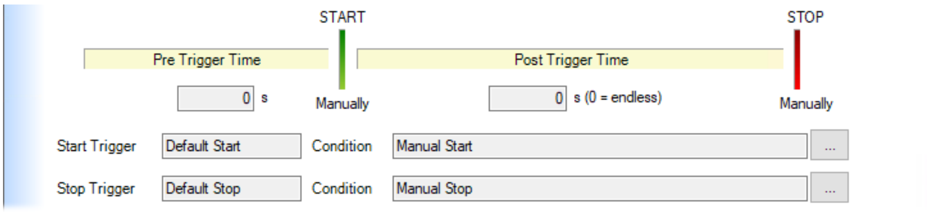


1. In the "Recording base filename" field, click the button next to the text field.

   A file selection window opens.
2. Enter path (e.g., `C:\Tutorial\EE\Experiments\System\Default\`) and filename for the log file and click **Save**.

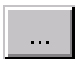
   The **Auto Increment** option results in the file name being extended by one number which is automatically incremented for multiple savings.

In the "Trigger Settings" field of the data logger, you can choose between different types of acquisition. You can use a fixed acquisition span which is manually started, you can start and stop acquisition manually or via trigger conditions which have to be fulfilled after the simulation has been started.



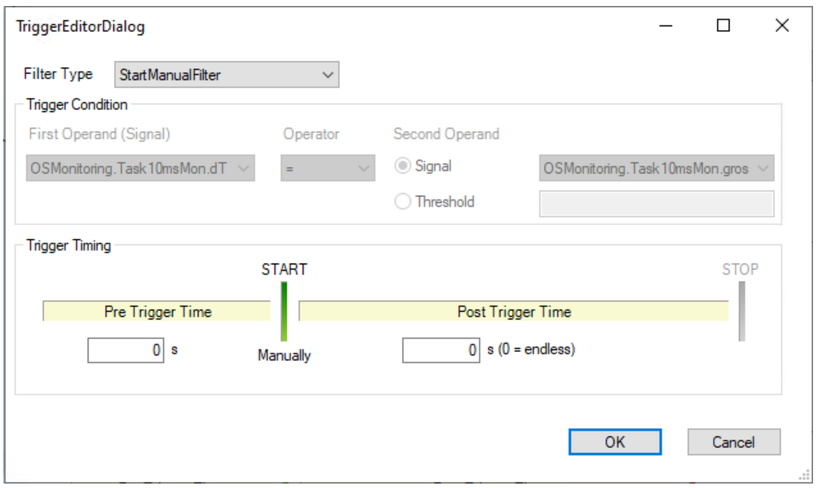
You work with a manually started acquisition of fixed duration in the tutorial. Information on trigger conditions can be found in the online help of the experiment environment.

To specify acquisition time



1. In the "Trigger Settings" field of the data logger, click the button in the "Start Trigger" line.

   The "TriggerEditorDialog" window opens.



2. In the "Filter Type" field, select `StartManualFilter`.

3. In the "Pre Trigger Time" and "Post Trigger Time" fields, enter the values `0` and `10`.

   With these settings, data are recorded for 10 seconds after you gave the start signal.

Data recording takes place only when the data logger is activated. In that case, the data of the selected measurement variables is acquired for 10 seconds after acquisition is started, and stored in the specified file (see “Data Acquisition” on page 135). You can analyze the data with an appropriate tool, e.g. the MDA delivered with INTECRIO.

5.8.5 Executing Measuring and Calibration Tasks

You execute the tasks in this section with the experiment you have set up in the previous sections.

5.8.5.1 Starting the Simulation

To start the simulation and measurement

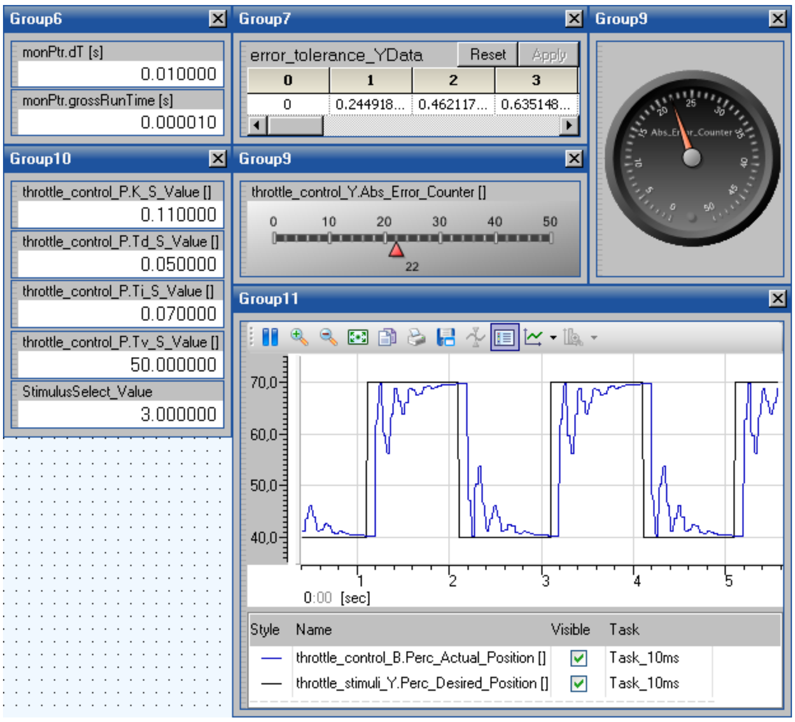
1. 2. 3.

1. In the experiment environment, click the **Download** button.
2. Click the **Start Simulation** button.
3. Click the **Start Measurement** button.

   Acquisition in the data logger starts, too. It stops once the time set has elapsed.

   The measuring instruments immediately show the values of the relevant variables.

Your experiment environment may look something like this.



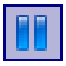
5.8.5.2 Measuring

Numerical display, circular gauge, bar display: These measuring instruments show the current values as soon as simulation and measurement are started. You can set up the instruments afresh during simulation (see "Setting Up Measuring and Calibration Instruments"). It is *not* possible to analyze the measure data in the instruments.

Oscilloscope: The oscilloscope shows the current values when simulation and measurement are started. You can set up the oscilloscope, or analyze the values currently displayed, when the display is paused.

To edit the display area

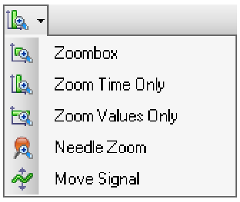
It may occur that the curves cannot be seen in entirety as they exceed the specified display area of the Y axis. There are different ways of changing the display area.



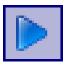
1. Pause the display.


2. To change the X axis, click the **Zoom in** and **Zoom out** buttons.
3. Move the display area of the Y axis with the mouse.
4. Squash or stretch the display area of the Y axis with <CTRL> + mouse.
5. Right-click the Y axis and select **Zoom to Fit** from the context menu.

   The display area of the oscilloscope is adjusted to the values currently displayed.
6. Use the **Zoom** button menu to use further zoom options.

7. Open the X or Y axis settings in the "Properties" window and edit the display of the axes (see Page 128).



8. Resume display with the **Toggle Play/Pause** button.

To set up the display

If displaying all measurement variables in one coordinate system looks too confusing, you can display individual axes.

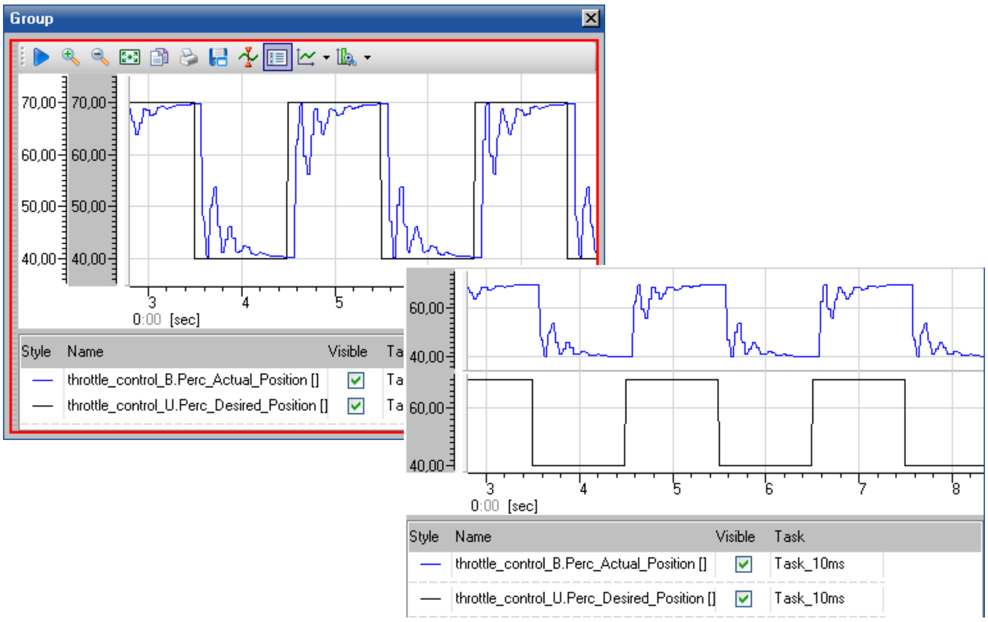
1. Right-click in the display area and select **Single Axis per channel** from the context menu.

   Now, each of the two measurement variables has its own Y axis that can be set up individually.

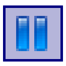
2. Right-click in the display area and select **Single Strip per channel** from the context menu.

   Now, the oscilloscope area is split in two parts, one for each measurement variable. Each part has its own Y axis that can be set up individually.

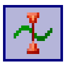
   The figure shows the two variants.



To analyze measure data



1. In the oscilloscope, click the **Toggle Play/Pause** button to interrupt measure data display.



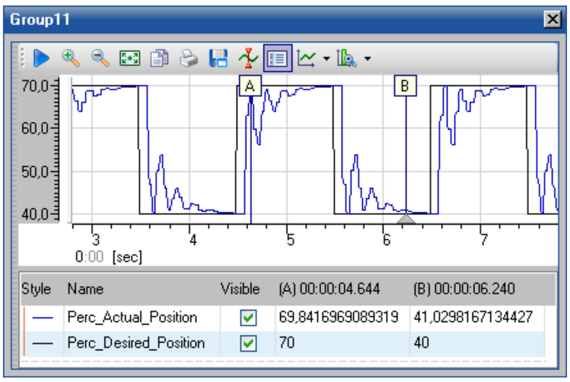
2. Click the **Add Cursor** button.

   A line cursor appears in the display area.

3. Add a second cursor.

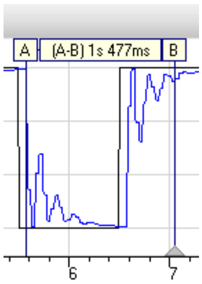
4. Use the mouse to drag each cursor to the desired position.

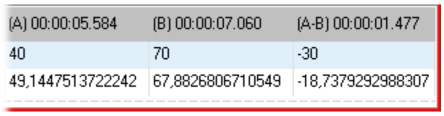
   The cursor names are displayed at the cursors. The Y-values of the channels at the current position are displayed in the legend.



5. Right-click the second cursor and select **Show Cursor difference › A: ‹X value›** from the context menu.

   The time difference is displayed at the cursors, and the Y difference is displayed in the legend.





5.8.5.3 Calibrating in the Editors

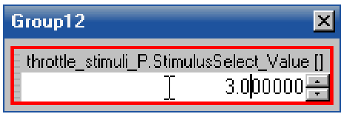
If you have assigned a calibration variable (characteristic value, curve or map) to an instrument, this instrument works like an editor. You can modify the values of the calibration variables directly in the display using the various editors.

<u>To calibrate values</u>

1. To calibrate values in the numerical editor, proceed as follows:

   i. Click the text area of the edit box.

   ii. Enter the value or use the scroll buttons.



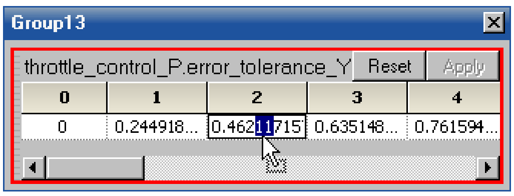
   The new value is effective immediately.

   iii. Calibrate the control parameters of the PID controller and observe the changes to the variable `Perc_Actual_Position` in the oscilloscope.

2. To calibrate `error_tolerance_YData` (characteristic curve) in the table editor, proceed as follows:

   i. Double-click a cell of the table.

   ii. Enter the new value.



   iii. Click **Apply**.

   The new value is effective.

   iv. Calibrate the tolerance values and observe the changes to the `Abs_Error_Counter` variable in the automotive instrument and the slider.

5.8.5.4 Data Acquisition

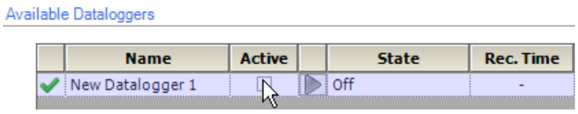
To perform data acquisition

With the data logger settings from section 5.8.4, you have to start measure data acquisition manually while the measurement is running.

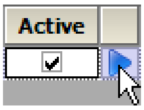
1. Go to the data logger.

   The current data logger status is displayed in the "State" column in the "Available Dataloggers" field.

2. Click in the "Active" column to activate the data logger.

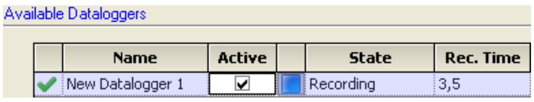


   The data logger is marked as active; the button in the next column becomes available.



3. Click the button to start data acquisition.

   Data recording starts; it stops automatically after 10 seconds.



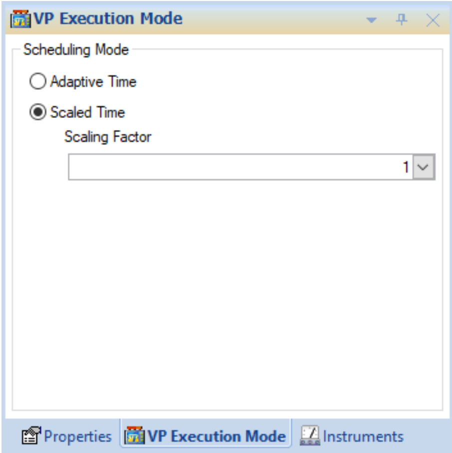
The measure data is saved in the specified directory (e.g., `D:\ETASData\INTECRIO5.0\Throttle\MyLogfile0001.dat`). You can analyze the data with an appropriate tool.

5.8.5.5 Special Features of a Virtual Prototyping Experiment

The previous sections of this lesson apply to both virtual prototyping and rapid prototyping. Here, a special feature of virtual prototyping experiments is described.

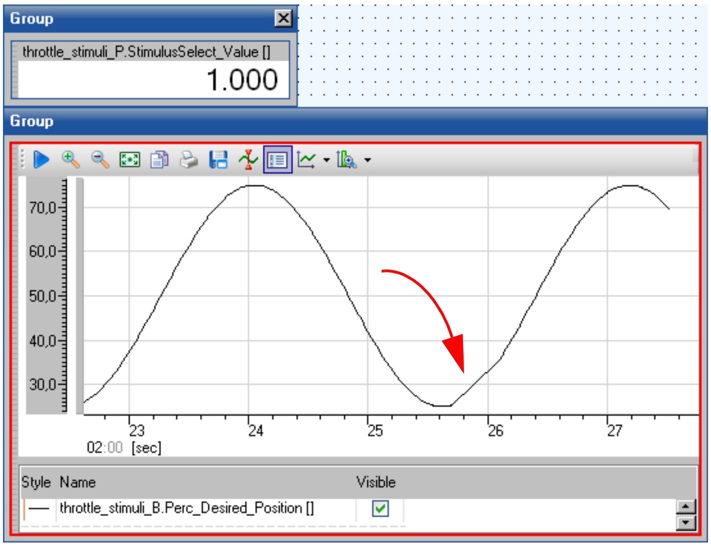
For VP experiments, the experiment environment contains an additional window named "VP Execution Mode" with the following modes for controlling simulation time:

Adaptive Time	The model is run in the fastest possible time, i.e. the shortest possible computing time.
Scaled Time	The model is run in real time (`Scaling Factor = 1`), time lapse (`Scaling Factor < 1`), or slow motion (`Scaling Factor > 1`).



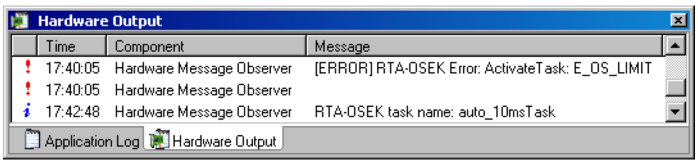
By default, **Scaled Time** with a scaling factor 1 is activated.

The limited real-time capability of Windows can lead to delays or failures during processing with high system loads which can in turn result in irregularities of the measure data (arrow in the figure below).



It is recommended that you close other resource-intensive programs during the virtual prototyping experiment.

If a failure occurs, a message about which task was not run is issued in the "Hardware Output" tab.



To change the VP execution mode

1. Stop both measuring and the operating system.

> NOTE
>
> The mode for the simulation time cannot be changed while the operating system is running.

2. In the "VP Execution Mode" window, do one of the following:
   - Select the mode **Adaptive Time**.
   - Modify the "Scaling Factor".
3. Relaunch the operating system and start measuring again and notice the differences in the display.

5.9 Lesson 8: ES930 – Example

> NOTE
>
> This lesson uses an ES910.3 hardware, HSP V12.1, the daisychain configuration tool, and appropriate cables for the experiment.
>
> The ES930 can also be connected to the ES830 target; however, details specific to that case are *not* covered by this lesson.

Target: In this lesson you will perform manual configuration of a daisychain with one ES930 module. Afterwards, you test the configuration in an experiment. You will learn about the working steps involved in working with an ES930 on ES910.

5.9.1 Overview of the Most Important Concepts of this Lesson

Hardware connection and Hardware Configurator with INTECRIO-RP

The control plant (or the technical process) is represented to the system as a set of sensors and actuators with which the system can be connected via a range of suitable peripherals. These peripherals have to be configured to suit the actual technical process being modeled.

The hardware connection task is carried out within INTECRIO by the Hardware Configurator (or *HC*).

The Hardware Configurator with INTECRIO-RP enables the setup and configuration of the respective hardware systems, it loads the parameterized model into the "Controller environment" which is displayed and linked by the interfaces installed in the hardware.

This is how the Hardware Configurator with INTECRIO-RP enables the creation of the logic necessary to ensure that signals received by external sources are acquired at the right place within the Controller model.

Daisychain configuration

Different from the other interfaces, a daisychain cannot be configured in INTECRIO. Instead, the daisychain is configured in the daisychain configuration tool, and the configuration file is imported in INTECRIO.

ES930

The ES930 Multi-I/O module is a compact, robust, and powerful tool with numerous input and output channels.

The module extends the functionality of the ES910, which makes it suitable for controlling and analyzing sensors and actuators directly from within a given function model (Simulink®, ASCET, AUTOSAR, C code).

5.9.2 Configuring the ES930 Daisychain Module

The configuration is done in the daisychain configuration tool provided by ETAS. The resulting `*.xml` configuration file is then imported into INTECRIO.

<u>To create a new daisychain configuration</u>

1. Connect the ES930 to the ES910.
2. Connect the ES910 to your PC.
3. Start the daisychain configuration tool, e.g. via the desktop icon.

   The "Hardware Configuration" window opens. See the online help for details.

4. In the "Hardware Configuration" window, insert an ES930 module.

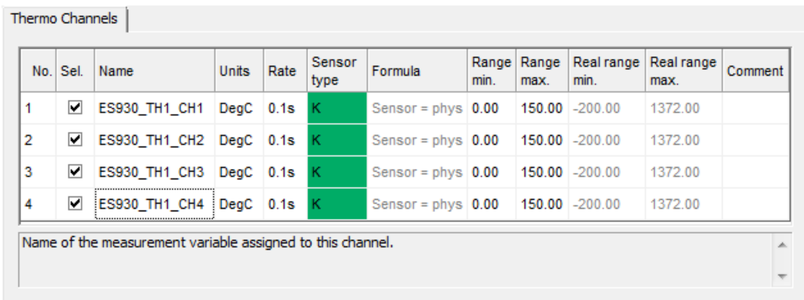
5. Enter a suitable IP address for the chain.
6. Select **Hardware > Activate Rapid Prototyping** to ensure that data will arrive with minimum delay.

By default, all Digital In/Out, Analog In/Out, and Thermo channels are activated. This tutorial uses only the first two Digital Out channels and one Thermo channel; you can deactivate the others.

<u>To configure the Thermo channels</u>

1. On the ES930 hardware, connect a temperature sensor to one of the Thermo channels.
2. In the "Hardware Configuration" window, "Topology" pane, select the `Thermo` entry.
3. In the "Thermo Channels" tab, activate the channel to which you connected the sensor.
4. Enter suitable values for rate, range min. and range max.
5. If desired, activate the unused thermo channels as well, to use them for diagnostic purposes.

   The screenshot shows a possible configuration.



Thermo Channels

No.	Sel.	Name	Units	Rate	Sensor type	Formula	Range min.	Range max.	Real range min.	Real range max.	Comment
1	☑	ES930_TH1_CH1	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
2	☑	ES930_TH1_CH2	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
3	☑	ES930_TH1_CH3	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
4	☑	ES930_TH1_CH4	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	

Name of the measurement variable assigned to this channel.

<u>To configure diagnostic channels</u>

1. In the "Hardware Configuration" window, "Topology" pane, select the `Diagnosis` entry.
2. Activate the `ES930_D_TH1_CH<n>_OpenWire` channels.

   During experiment, these channels return `true` when the respective Thermo channel is unused, and `false` when the channel is used.

3. Start an online measurement in the "Hardware Configuration" window.

   The screenshot shows the "Diagnosis" tab with running online measurement.

Diagnosis

No.	Sel.	Name	Value	Units	Rate Group	Rate	Comment
27	☑	ES930_D_TH1_CH1_OpenWire	True	bit			
28	☑	ES930_D_TH1_CH2_OpenWire	True	bit			
29	☑	ES930_D_TH1_CH3_OpenWire	False	bit	ES930_TH1_CH1	0.1s	
30	☑	ES930_D_TH1_CH4_OpenWire	True	bit			
31	☐	ES930_D_DO1_CH1_MinPulseAdapti	---	bit			

To configure the Digital Out channels

1. In the "Hardware Configuration" window, "Topology" pane, select the `Digital Out` entry.
2. Activate the first and second channel.
3. Select the `PWM Out` mode for both channels.
4. Select the active state for each channel.
5. Assign each channel to one user LED.
6. Specify initial values for active time and period.

   The screenshot shows a possible configuration.

No.	Settings	Sel.	HW Channel	Active State	Mode	Name	Update Mode	Alignment	LED	Half Bridge/HB	Output	Active Time	Period Time	Units
1		☑	CH1	High	PWM Out	ES930_DO1_CH1_ActiveTime	Individual	n/a	U1	None	n/a	500.00	n/a	msec
2						ES930_DO1_CH1_PeriodTime					n/a	n/a	2000.00	msec
3		☐		n/a	n/a	ES930_DO1_CH1_Enable	n/a	n/a	n/a	n/a	Enabled	n/a	n/a	bit
4		☑	CH2	Low	PWM Out	ES930_DO1_CH2_ActiveTime	Individual	n/a	U2	None	n/a	1000.00	n/a	msec
5						ES930_DO1_CH2_PeriodTime					n/a	n/a	2000.00	msec
6		☐		n/a	n/a	ES930_DO1_CH2_Enable	n/a	n/a	n/a	n/a	Enabled	n/a	n/a	bit

To complete the daisychain configuration

1. When you completed the configuration, initialize the daisychain.
2. Save the configuration file.

   An `*.xml` file and a `*.gcf` file are written to the specified directory. You need the `*.xml` file to use the ES930 with INTECRIO.

With that, the actions in the daisychain configuration tool are complete.

5.9.3 Creating and Setting Up the ES910 Hardware System

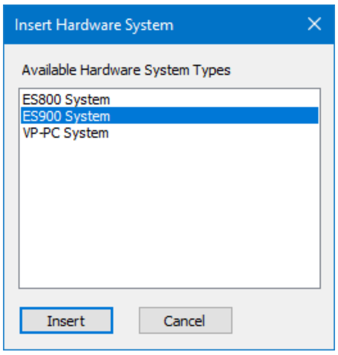
5.9.3.1 The ES900 System

To create an ES900 system

1. Expand the `Hardware` folder in the Workspace Browser.

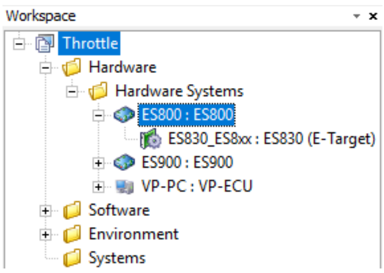
2. Right-click the `Hardware Systems` folder and select **Add Hardware System** from the context menu.

   The "Insert Hardware System" window opens. It contains all available hardware systems, i.e. `ES800`, `ES900` and `VP-PC`.



3. Select `ES900 System` and click **Insert** to add an ES900 system.
4. In the "Rename" window, click **OK** to adopt the default name, `ES900`, for the hardware system.

   The `ES900` system is added and displayed in the Workspace Browser.



The still empty system now has to be equipped with the devices necessary for throttle control. Add the ES910 target and the daisychain one after the other.

5.9.3.2 ES910 Simulation Target

The ES910 simulation target is used as master device of the hardware structure created in the Hardware Configurator. Since INTECRIO V5.0, the target is added automatically if the hardware system supports only one target. All other devices are then created under this one.

If the target gets lost, you can add it manually. See the INTECRIO online help for details.

<u>To configure the experimental target</u>

1. In the Workspace Browser, double-click `ES910`.

   The Hardware Configurator for the ES910 opens. It contains the board parameters.

Name	Meaning
`Name`	Name of the target
`IO Failure Behavior`	Behavior when errors occur during I/O initialization

2. Accept the preset values.

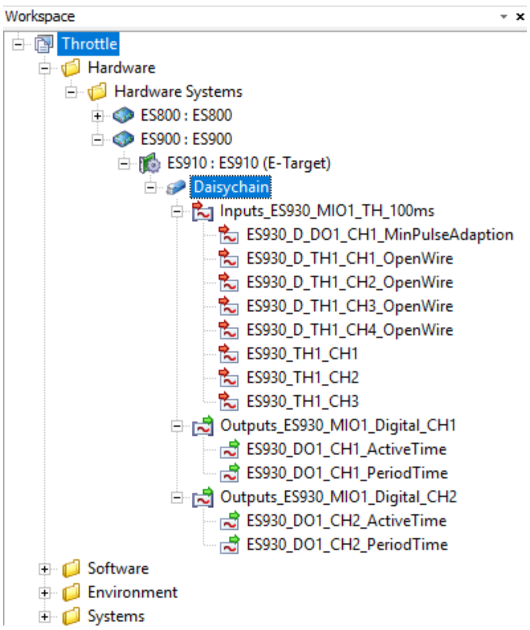
5.9.3.3 Daisychain Device

<u>To create the daisychain device</u>

1. In the Workspace Browser, right-click `ES910` and select **Insert** from the context menu.
2. In the "Insert New Item" window, select `Daisychain`.
3. Click **Insert**.
4. In the "Rename" window, click **OK** to adopt the default name.

   The device is empty until you import a daisychain configuration file.
5. Right-click the `Daisychain` device and select **Import Daisychain Configuration** from the context menu.

   An import wizard opens.
6. Follow the instructions in the wizard and import the `*.xml` daisychain configuration file you created in section 5.9.2.



Now you activate the device.

<u>To activate the daisychain device</u>

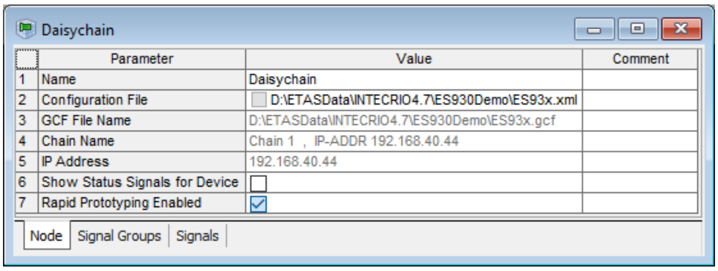
1. Open the context menu of the `Daisychain` device.

   If the menu option **Enabled** has a tick beside it, the device is activated.
2. Otherwise select the menu option **Enable** to activate the device.

To configure the daisychain device

1. In the Workspace Browser, right-click `Daisychain` and select **Open** from the context menu.

   The Hardware Configurator for the daisychain opens. It shows the device parameters in the "Node" tab.

   The "Node" tab contains the general device parameters. The online help contains a list of the parameters and what they mean.

   You edit a value by clicking on the relevant cell. Depending on the type of parameter, the cell becomes an input field in which you enter the value or a combo box from which you select the value.

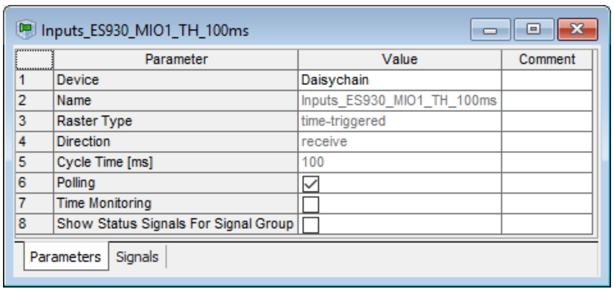
   Fields with gray font cannot be edited.

2. Activate status signals for the device.
3. If desired, enter comments on the individual parameters in the "Comments" column.

To configure the signal groups

1. In the Workspace Browser, double-click the signal group `Inputs_ES930_MIO1_TH_100ms`.

   The Hardware Configurator for the signal group opens. It shows the parameters for the signal group. The online help contains a list of the parameters and what they mean.

   Fields with gray font cannot be edited.

2. If desired, enter comments for the parameters.
3. Configure the other signal groups.

As an alternative to the way described here, you can select the "Signal Groups" tab in the Hardware Configurator of the daisychain (see the previous page) and configure the signal group there. Apart from the comments, the tab contains the same parameters as the editor.

To configure the signals

1. Do one of the following:
   - In the Hardware Configurator of the daisychain, select the "Signals" tab.
     This tab contains the parameters of the individual signals of all signal groups.
   - In the Hardware Configurator of a signal group, select the "Signals" tab
     This tab contains the parameters of the individual signals of this signal group.
   - In the Workspace Browser, double-click a signal.
     The Hardware Configurator for the signal opens. It contains the parameters for the signal.

   The online help contains a list of the signal parameters and what they mean.

   Fields with gray font cannot be edited.
2. If desired, enter comments for the signal parameters.

The hardware system is now completely specified and can be added to the system project.

Other configuration possibilities for any kind of supported hardware device are described in the online help.

5.9.4 Experimenting with the ES930

5.9.4.1 Creating the System Project

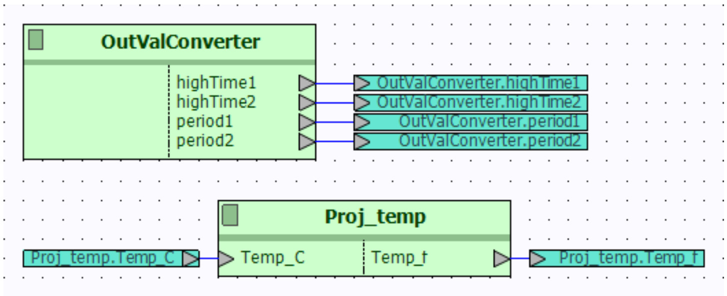
You need a complete system project for the experiment. For that purpose, two software modules are provided:

- `OutValConverter` uses frequency and duty cycle to calculate high time (or active time) and period values for the ES930 Digital Out channels.
- `Proj_temp` converts the temperature measured at the Thermo channel from °C to °F.

To create the system project

1. Import the software modules `OutValConverter` and `Proj_temp` as described on Page 99.

   The `*.six` files and the other necessary files are available in the `<sample files>\ES930Demo` directory, subdirectories `OutValConverter` and `Proj_temp`.
2. Create a software system (Page 105) and name it, e.g., `SWS_ES930`.
3. Add the modules to the software system (Page 105) and add the necessary interfaces (Page 104 or Page 105) as shown below.



4. Create a system project as described on Page 108, make it the active system and name it, e.g., `SP_ES930`.
5. Integrate the hardware system and software system to the system project (Page 109 and Page 110).

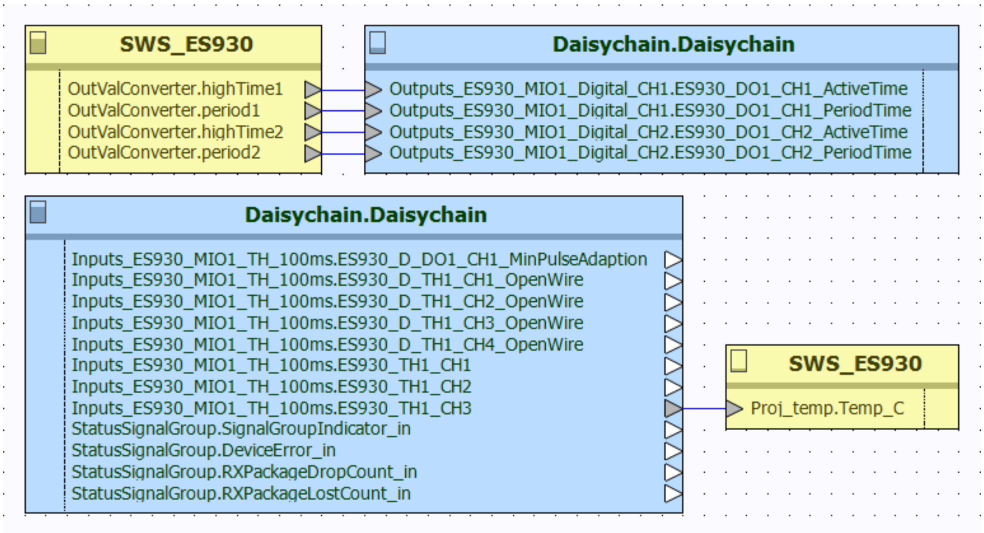
To connect hardware and software

1. Open the `SP_ES930` system project in the Graphical Modeler.

   The graphic editor opens; the system project is displayed.

   The hardware device is displayed as a set of blue blocks. Hardware channels that can be connected to the signal sinks of the software system are arranged at the right side of one block. Hardware channels that can be connected to the signal sources of the software system are arranged at the left side of a second block.

   You can add and remove hardware devices manually; see section 4.7.2 on page 61 or the online help for details.
2. Create the connections (Page 111) shown below.



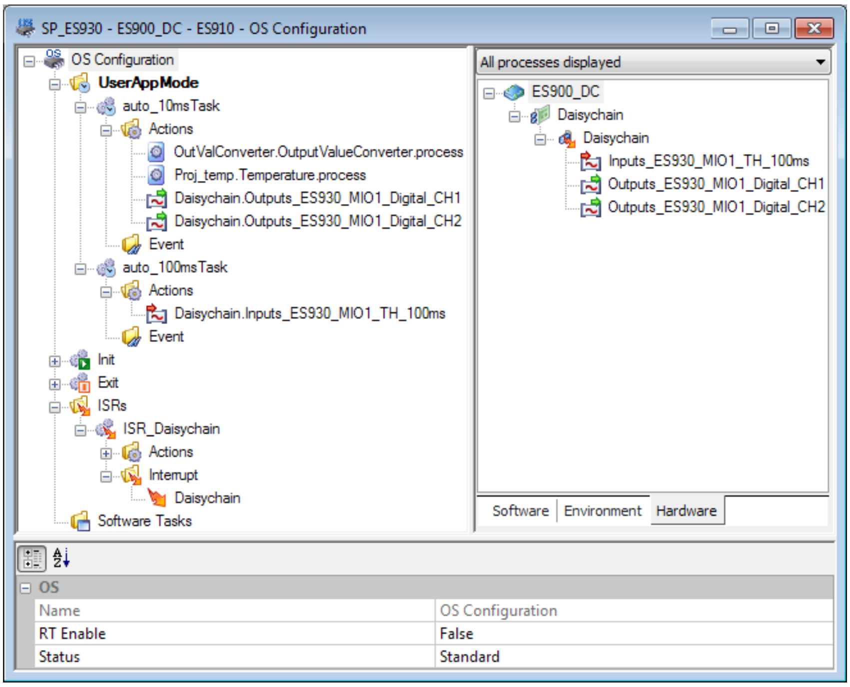
Unused SWS ports are hidden for clarity.

To configure the operating system

1. Open the OSC.

2. Click the **OS Auto Mapping** button to start automatic OS configuration.

   The necessary tasks and ISR are created; the processes from the "Software" and "Hardware" tabs are assigned automatically.



To generate the prototype

Once you have created the system project and selected the compiler, you generate the executable file.

1. Set the Build options (see Page 119).
2. Generate the prototype (see Page 119).

   INTECRIO generates the executable file `SP_ES930.rta` and stores it in the results directory of the workspace, e.g. under `ETASData\INTECRIO5.0\Tutorial\ES930_demo\ Results`.

5.9.4.2 Configuring the ES910

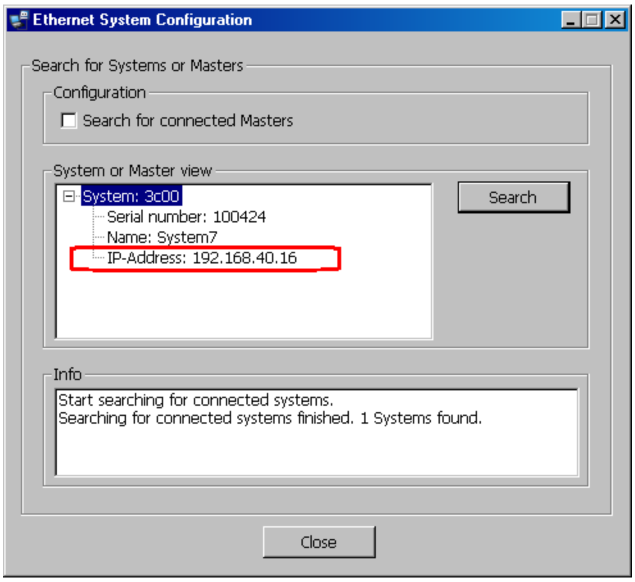
In order to use the IO interface for a daisychain, you must check the port settings of ES910 via its graphic user interface. A web browser application is used for that purpose; the ES910 must be connected to the PC.

To determine the IP address and open the configuration interface

An alternative way to open the ES910 user interface is described in the ES910 user guide.

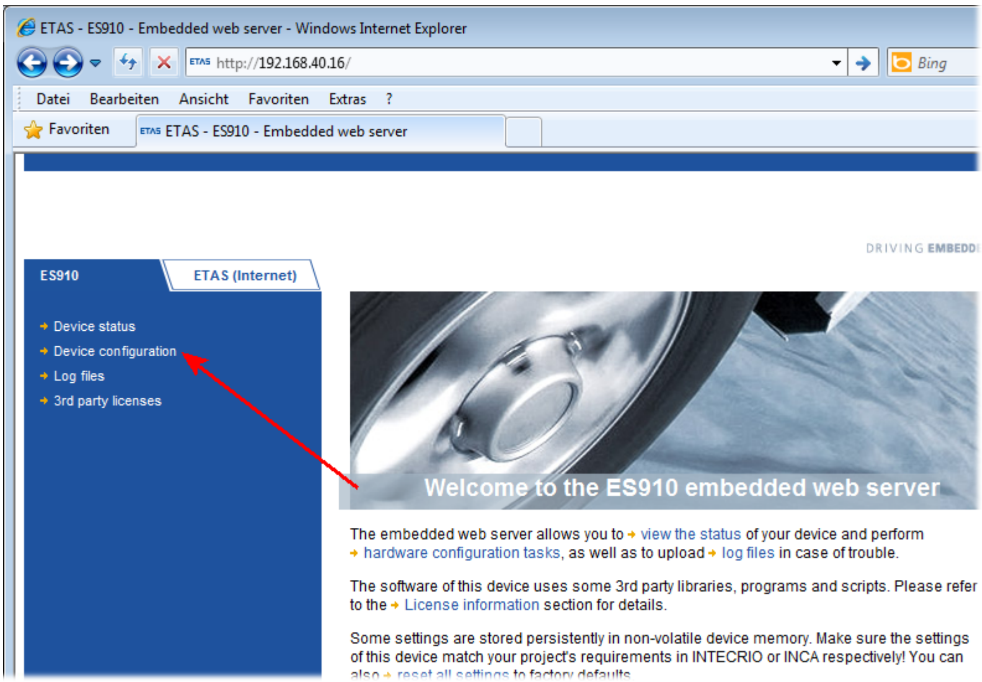
1. In the Windows system tray, right-click the **IP-Manager** icon and select **Ethernet System Configuration** from the context menu.

The "Ethernet System Configuration" window opens.



The IP address is shown in the "System or Master view" field.

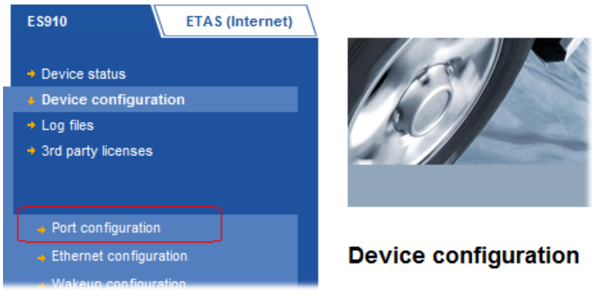
2. Enter this IP address in your web browser to open the ES910 user interface.



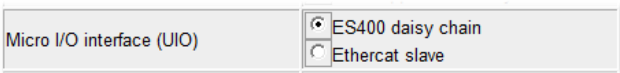
To configure the ES910

1. On the start page of the ES910 user interface, follow the Device configuration link.

2. On the "Device configuration" page, follow the Port configuration link below the "Device configuration" caption.



3. Ensure that the **ES400 daisychain** option is activated for the `Micro I/O interface`.



4. Save the configuration.

5.9.4.3 Experimenting

After these preparations, you can perform the experiment. You will use eleven numerical instruments and four LED instruments to measure and calibrate the following variables:

Variable	Source	Remark
`dutycycle1`	SW module parameters	inputs of the `OutValConverter` module
`frequency1`		
`dutycycle2`		
`frequency2`		
`highTime1`	SW module variables	outputs of the `OutValConverter` module
`period1`		
`highTime2`		
`period2`		
`ES930_TH1_CH3`	HW thermo channel	measured temperature
`Temp_C`	SW module variables	input/output of the `Proj_temp` module
`Temp_F`		
`ES930_D_TH1_CH1_OpenWire`	HW diagnosis channels	check whether a thermo channel is unconnected (`true`) or connected (`false`)
`ES930_D_TH2_CH2_OpenWire`		
`ES930_D_TH3_CH3_OpenWire`		
`ES930_D_TH4_CH4_OpenWire`		

To set up the experiment

1. Open the experiment environment (Page 121).
2. Create three groups of numerical instruments for the following variables (see Page 124).

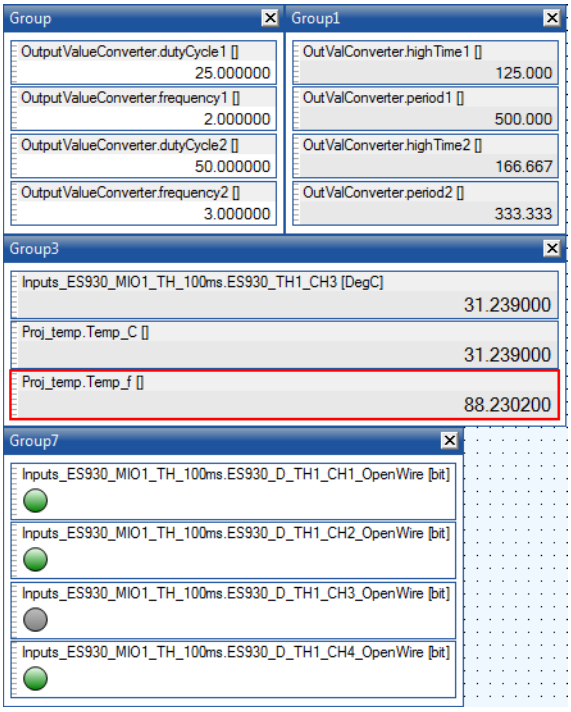
Group	Variable
1	dutycycle1
	frequency1
	dutycycle2
	frequency2
2	highTime1
	period1
	highTime2
	period2
3	ES930_TH1_CH3
	Temp_C
	Temp_F

3. Create a group of four LED instruments for the `ES930_D_TH<n>_CH<n>_OpenWire` variables.
4. Save the experiment (see Page 128).

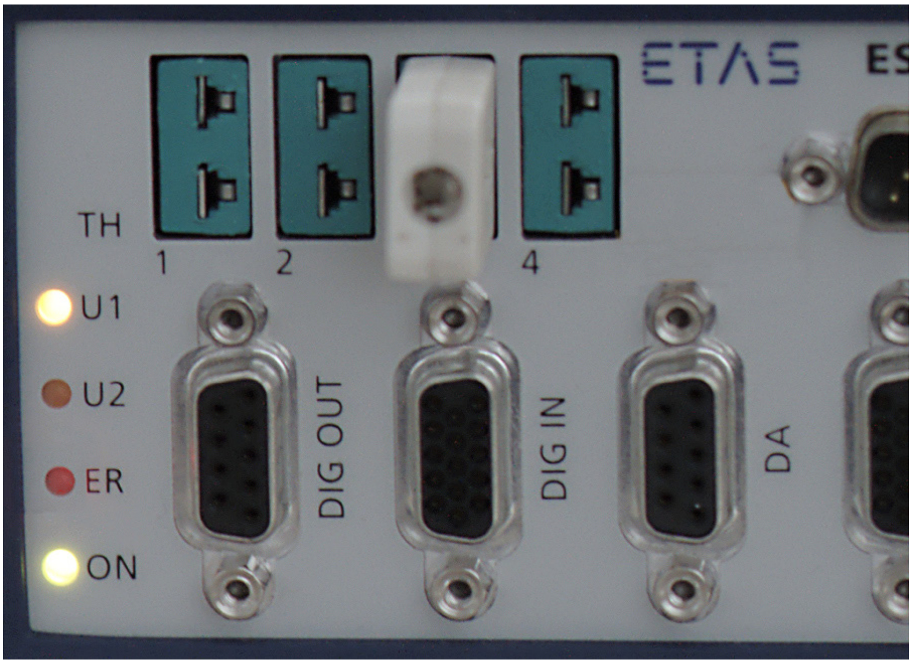
<u>To perform the experiment</u>

1. Start simulation and measurement (see Page 131).

   The numerical displays and the LED instruments show the current values.



   The user LEDs (U1 and U2) on the ES930 front panel are blinking.



2. Edit the parameters in the first group and watch how the blinking pattern of the user LEDs on the ES930 changes.

6 ETAS Network Manager

The ETAS Network Manager is used for creating a configuration that will be used by the ETAS IP Manager. The IP Manager is responsible for dynamic IP addressing of the ETAS hardware used in your network (ETAS network).

Open the ETAS Network Manager, e.g. via the app list in the Windows Start menu, **E > ETAS > ETAS Network settings**.

Follow the instructions given in the dialog. For further information about addressing and configuring the network adapter, press <F1> in the ETAS Network Manager.

7 Troubleshooting General Problems

This chapter gives some information of what you can do when problems arise that are not specific to an individual software or hardware product.

7.1 Network Adapter Cannot be Selected via Network Manager

Cause: APIPA is Disabled

The alternative mechanism for IP addressing (APIPA) is usually enabled on all Windows systems. Network security policies, however, may request the APIPA mechanism to be disabled. In this case, you cannot use a network adapter which is configured for DHCP to access ETAS hardware. The ETAS Network Manager displays a warning message.

The APIPA mechanism can be enabled by editing the Windows registry. This is permitted only to users who have administrator privileges. It should be done only in coordination with your network administrator.

<u>To enable the APIPA mechanism</u>

1. Open the Registry Editor:
   i. Press <WINDOWS LOGO> + <R>.
   ii. Enter `regedit` and click **OK**.

   The registry editor is displayed.
2. Open the folder `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpic\Parameters\`.
3. Select **Edit > Find** to search for the key `IPAutoconfigurationEnabled`.

If you cannot find any instances of the registry key mentioned, the APIPA mechanism has not been disabled on your system, i.e. there is no need to enable it. Otherwise proceed with the following steps.

4. Set the value of the key `IPAutoconfigurationEnabled` to `1` to enable the APIPA mechanism.

   You may find several instances of this key in the Windows registry which either apply to the TCP/IP service in general or to a specific network adapter. You only need to change the value for the corresponding network adapter.
5. Close the registry editor.
6. Restart your workstation in order to make your changes take effect.

7.2 Search for Ethernet Hardware Fails

There is a number of different causes which might lead to connection problems, most of them being based on inappropriate Windows or hardware settings. Usually, these can easily be modified, once they have been identified.

The following list of causes shall help you in finding the root cause of the problem and fixing it.

Cause: The Versions of the Hardware and the ETAS Software Are Not Compatible

If you are using ETAS hardware with ETAS software, you can use the ETAS HSP Update Tool to check the firmware version of your hardware:

- Make sure you use the ETAS HSP Update Tool with the latest HSP (Hardware Service Pack) version.
- Also use the HSP Update Tool to check whether the hardware is compatible with the software used.
- Make sure any additional drivers for that hardware are installed correctly.

You can get the required HSP from the ETAS internet pages at www.etas.com.

If you still cannot find the hardware using the HSP Update Tool, check whether the hardware offers a Web interface and whether you can find using this interface. Otherwise check whether one of the following causes and solutions might apply.

Cause: Personal Firewall Blocks Communication

Personal firewalls may interfere with access to ETAS Ethernet hardware. The automatic search for hardware typically cannot find any Ethernet hardware at all, although the configuration parameters are correct.

Certain actions in ETAS products may lead to some trouble if the firewall is not properly parameterized, e.g. upon opening an RP experiment in the ETAS experiment environment or searching for hardware from within INCA or HSP.

If a firewall is blocking communication to ETAS hardware, you must either disable the firewall software while working with ETAS software, or the firewall must be configured to give the following permissions.

Permissions given through the firewall block ETAS hardware:

- Outgoing limited IP broadcasts via UDP (destination address 255.255.255.255) for destination ports 17099 or 18001
- Incoming limited IP broadcasts via UDP (destination IP 255.255.255.255, originating from source IP 0.0.0.0) for destination port 18001
- Directed IP broadcasts via UDP to the network configured for the ETAS application, destination ports 17099 or 18001
- Outgoing IP unicasts via UDP to any IP in network configured for the ETAS application, destination ports 17099 through 18020
- Incoming IP unicasts via UDP originating from any IP in the network configured for the ETAS application, source ports 17099 through 18020, destination ports 17099 through 18020

- Outgoing TCP/IP connections to the network configured for the ETAS application, destination ports 18001 through 18020

> **NOTE**
>
> The ports that have to be used in concrete use cases depend on the hardware you use. For more precise information on the port numbers that can be used please refer to your hardware documentation.

Permissions given through the firewall block XCP on Ethernet:

- Outgoing IP multicasts for XCP Slave Detection via UDP to any IP in network, destination IP 239.255.0.0, port 5556.
- Incoming IP multicasts for XCP Slave Detection via UDP from any IP in network, destination IP 239.255.37.45, port 3745.

Contact your IT responsible to clarify whether the required permissions are, or can be, given by the firewall.

> *NOTICE*
>
> **Insecure system due to changes to your firewall**
>
> Always consult your IT responsible and/or check the IT security policies of your company before changing your firewall configuration and reconnecting the computer to the network.

Cause: Client Software for Remote Access Blocks Communication

PCs or notebooks which are used outside the ETAS hardware network sometimes use a client software for remote access which might block communication to the ETAS hardware. This can have the following causes:

- A firewall which is blocking Ethernet messages is being used (see “Cause: Personal Firewall Blocks Communication” on page 152).
- By mistake, the VPN client software used for tunneling filters messages. As an example, Cisco VPN clients with versions before V4.0.x in some cases erroneously filtered certain UDP broadcasts.

  If this might be the case, please update the software of your VPN client.

Cause: ETAS Hardware Hangs

Occasionally the ETAS hardware might hang. In this case switch the hardware off, then switch it on again to re-initialize it.

Cause: ETAS Hardware Went into Sleep Mode

In order to save power, some ETAS devices will go to sleep mode if they do not see that they are connected to another device/computer.

To solve that, connect your Ethernet cable from your computer to the "HOST"/ "Sync In" port on the device. After the device turns on, connect to the device using the web interface and change the settings so that the device stays always on. Consult the device's manual for details on how to do that.

Cause: Network Adapter Temporarily Has No IP Address

Whenever you switch from a DHCP company LAN to the ETAS hardware network, it takes at least 60 seconds until ETAS hardware can be found. This is caused by the operating system's switching from the DHCP protocol to APIPA, which is being used by the ETAS hardware.

Cause: ETAS Hardware Had Been Connected to Another Logical Network

If you use more than one PC or notebook for accessing the same ETAS hardware, the network adapters used must be configured to use the same logical network. If this is not possible, it is necessary to switch the ETAS hardware off and on again between different sessions (repowering).

Cause: Device Driver for Network Card Not in Operation

It is possible that the device driver of a network card is not running. In this case you will have to deactivate and then reactivate the network card.

<u>To deactivate and reactivate the network card</u>

1. To deactivate the network card, open the Control Panel.
2. Go to the "Network and Sharing Center", then click the "Change adapter settings" link.
3. In the "Network Connections" window, right-click the used network adapter and select **Disable** in the context menu.
4. In order to reactivate the network adapter right-click it again and select **Enable**.

Cause: Laptop Power Management Deactivates the Network Card

The power management of a laptop computer can deactivate the network card. Therefore you should turn off power monitoring on the laptop.

<u>To switch off power monitoring on the laptop</u>

1. From the Windows Start Menu, select **Control Panel > Hardware and Sound > Device Manager**.
2. In the Device Manager, open the tree structure of the **Network Adapters** entry.
3. Right-click the used network adapter and select **Properties** in the context menu.
4. Select the **Power Management** tab and deactivate the **Allow the computer to turn off this device to save power** option.
5. Select the **Advanced** tab. If the property **Autosense** is included, deactivate it also.
6. Click **OK** to apply the settings.

Cause: Automatic Disruption of Network Connection

It is possible after a certain period of time without data traffic that the network card automatically interrupts the Ethernet connection. This can be prevented by setting the registry key `autodisconnect`.

To set the registry key **`autodisconnect`**

1. Open the Registry Editor.
2. Select under `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\lanmanserver\parameters` the Registry Key `autodisconnect` and change its value to `0xffffffff`.

8 Contact Information

Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

www.etas.com/hotlines

ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com

9 Glossary

This glossary contains explanations of the technical terms and abbreviations used in the INTECRIO documentation. While many terms are also used in a more general sense, this glossary specifically addresses the meaning of those terms as they are applied to INTECRIO. The terms are listed in alphabetical order.

9.1 Abbreviations

API

Programming interface for the application developer (**A**pplication **P**rogramming **I**nterface).

ASAM-MCD

Working Group for Standardizing Automation and Measuring Systems, including the Working Groups Measuring, Calibrating, and Diagnostics (German: **A**rbeitskreis zur **S**tandardisierung von **A**utomations- und **M**esssystemen, mit den Arbeitsgruppen **M**essen, **C**alibrieren und **D**iagnose)

ASAM-MCD-2MC

A file format used to describe the calibration variables and measured signals contained in the control unit software, and additional specific information designed to parameterize the experiment interface. ASAM-MCD-2MC is used to import the information required for this into an experiment (A2L file).

INTECRIO V5.0 supports the ETK AML versions 1.1 (only hook-based bypass) or 1.2 – 1.7 (hook-based and service-based bypass), XETK AML versions up to 2.5, ASAP1B_Bypass AML V1.0 and higher, and SBB AML versions 2.0, 3.0 and 3.1.

For further information, refer to https://www.asam.net.

ASCET

ETAS product family for the development of electronic control unit software. ASCET models can be imported in INTECRIO.

ASCET-MD

ASCET **M**odeling & **D**esign – the behavioral modeling tool of the ASCET product family.

ASCET-RP

ASCET **R**apid **P**rototyping – the rapid prototyping tool of the ASCET product family.

AUTOSAR

Automotive **O**pen **S**ystem **Ar**chitecture; see https://www.autosar.org/

BMT

Behavioral **M**odeling **T**ool. The BMT can be used to edit, simulate and animate the behavior of models and generate the function code.

BR_XETK

Emulator test probe (ETK) with Automotive Ethernet interface.

Requires an ES800 hardware system with ES882/ES886.

BSW

Basic **s**oft**w**are; provides communications, I/O, and other functionality that all software components are likely to require.

CAN

Controller **A**rea **N**etwork; a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each others' applications without a host computer.

CANdb

CAN data**b**ase; CAN description file created with the CANdb data management program made by the company Vector Informatik.

INTECRIO V5.0 supports the CANdb versions V2.3 and higher.

CPU

Central **P**rocessing **U**nit. In the context of INTECRIO, this refers to a single microcontroller.

Distab

Data exchange method, used in ETK bypass experiments for data exchange between experimental target and ECU.

INTECRIO V5.0 supports hook-based bypass with Distab 12 and higher (XETK AML: Distab 13), as well as service-based bypass with Distab 13.

In combination with ES830 or ES910.3, INTECRIO V5.0 supports also bypass with Distab 17.

ECU

Electronic **C**ontrol **U**nit; a small embedded computer system that consists of a CPU and the associated periphery, where everything is usually located in the same housing.

ETK

emulator test probe (German: **E**mulator-**T**ast**k**opf)

FETK

emulator test probe (ETK) for the ETAS ES89x ECU and Bus Interface Modules

FIBEX

Field **B**us **Ex**change – an exchange format based on an XML schema which is used for descriptions of the complete in-vehicle communication network. FIBEX is defined for various network types (CAN, LIN, MOST, FlexRay) and contains information about bus architecture, signals, properties of network nodes, etc.

The FIBEX file format is standardized by ASAM (Association for Standardization of Automation and Measuring Systems).

INTECRIO V5.0 supports the FIBEX baseline versions FIBEX V2.0.x and V3.1.0; the latter with some restrictions (see the online help for details).

For further information, refer to https://www.asam.net.

FIFO

First **i**n, **f**irst **o**ut

HBB

hook-**b**ased **b**ypass

HC

Hardware **C**onfigurator

IER

INTECRIO **E**mbedded Coder **R**eal-Time Target (Embedded Coder real-time target for importing Simulink models in INTECRIO)

INCA

ETAS measuring, calibration and diagnostics system (**In**tegrated **C**alibration and **A**cquisition Systems)

For cooperation with INTECRIO V5.0.4, you need INCA V7.2.17 or higher.

INCA-EIP

INCA add-on; allows access to the rapid prototyping (ES910, ES830) and virtual prototyping (VP-PC) targets for INCA.

For cooperation with INTECRIO V5.0.4, you need INCA-EIP V7.2.17 or higher.

INTECRIO-RP

INTECRIO **R**apid **P**rototyping package – provides connectivity to the rapid prototyping targets.

INTECRIO-VP

INTECRIO **V**irtual **P**rototyping package – provides connectivity to the virtual prototyping targets.

IRT

INTECRIO **R**eal-Time **T**arget (Simulink Coder real-time target for importing Simulink models in INTECRIO)

LDF

LIN **d**escription **f**ile – a configuration file for a LIN controller.

INTECRIO V5.0.4 supports the LDF versions 1.3, 2.0, 2.1, and 2.2.

LIN

Local **I**nterconnect **N**etwork; a serial network protocol used for communication between components in vehicles.

LIN is used where the bandwidth and versatility of CAN are not needed. Typical application examples are the networking within the door or the seat of a motor vehicle.

LSB and lsb

Least **S**ignificant **B**yte (capital letters) or **b**it (small letters)

MDA

Measure **D**ata **A**nalyzer program; an offline instrument by ETAS for displaying and analyzing saved measurement data.

MSB and msb

Most **S**ignificant **B**yte (capital letters) or **b**it (small letters)

OIL

OSEK **I**mplementation **L**anguage – as describing language for electronic control unit networks, it is an indirect part of the OSEK operating system. OIL is used to describe static information of the electronic control unit network, such as communication connections and electronic control unit properties.

OS

Operating **S**ystem

OSC

OS configurator (**O**perating **S**ystem **C**onfigurator)

OSEK

Working group for Open Systems for Electronics in Motor Vehicles (German: **O**ffene **S**ysteme für die **E**lektronik im Kraftfahrzeug)

PDU

Protocol **d**ata **u**nit; a data unit that contains payload and control information which is passed between the layers in a protocol stack.

In INTECRIO V5.0, a FlexRay PDU corresponds to a signal group.

RE

Runnable **e**ntity; a a piece of code in an SWC that is triggered by the RTE at runtime. It corresponds largely to the processes known in INTECRIO.

RP

Rapid **p**rototyping; see also page 163

RTA-OSEK

ETAS real-time operating system; implements the AUTOSAR-OS V1.0 (SC-1) and OSEK/VDX OS V2.2.3 standards and is fully MISRA compliant.

RTA-OS

ETAS real-time operating system; implements the AUTOSAR R3.0 OS and OSEK/VDX OS V2.2.3 standards and is fully MISRA compliant.

RTA-RTE

AUTOSAR runtime environment by ETAS

RTE

AUTOSAR runtime environment; provides the interface between software components, basic software, and operating systems.

RTIO

Real-**T**ime **I**nput-**O**utput

SBB

Service-**b**ased **b**ypass

SBC

Electrohydraulic brake system (**S**ensotronic **B**rake **C**ontrol)

SCOOP

Source **Co**de, **O**bjects, and **P**hysics

SCOOP-IX

SCOOP **I**nterface E**x**change language.

INTECRIO V5.0 supports SCOOP-IX versions V1.0, V1.1, V1.2, V1.4, and V1.5.

SP

Service **p**oint; see also page 164

SWC

Atomic AUTOSAR software component; the smallest non-dividable software unit in AUTOSAR.

UDP

User **D**atagram **P**rotocol

UML

Unified **M**odeling **L**anguage

VFB

Virtual **f**unction **b**us in AUTOSAR

VP

Virtual **p**rototyping; see also page 164

XCP

Universal measurement and **c**alibration **p**rotocol; the **x** generalizes the various transportation layers that can be used. The long name is ASAM MCD-1 XCP.

INTECRIO V5.0 supports XCP version V1.0 and all subsequent versions which are compatible with V1.0. In addition, the `XCPplus` keyword from V1.1 and higher is supported.

XETK

emulator test probe (ETK) with Ethernet interface

XML

E**x**tensible **M**arkup **L**anguage

9.2 Terms

Actuator

Executing hardware unit. It forms the physical interface between electronic signal processing and mechanics.

Application mode

An application mode is part of the operating system; it describes different possible states of a system, such as the application mode EEPROM programming mode, starting or normal operation.

AUTOSAR software component

see SWC

Basic software

see BSW

Bypass experiment

In a bypass experiment, parts of an electronic control unit program are executed on the experimental target (ES900, ES800). This requires a special hook in the code.

INTECRIO V5.0 supports several types of bypass experiments: XCP bypass on CAN or UDP, as well as hook-based and service-based ETK/XETK/FETK bypass.

Connection, dynamic

Connection between signal source and sink that can be changed at runtime without a new build process.

Connection, static

Connection between signal source and sink that cannot be changed at runtime.

Crossbar

Manages and controls the connections between modules, functions and hardware in a non-AUTOSAR environment.

Embedded Coder®

An add-on for the Simulink® Coder™; extends the capabilities provided by the Simulink Coder to support specification, integration, deployment, and testing of production applications on embedded targets.

Environment system

Environment systems are used to model the plant model for virtual prototyping. They are built out of modules and functions, the same way as software systems.

Event

An event is an (external) trigger that initiates an action of the operating system, such as a task.

Event interface

see Process

FlexRay

FlexRay is a scalable and fault tolerant communication system for high-speed and deterministic data exchange. FlexRay's time-division multiplexing facilitates the design of modular or safety-related distributed systems. Its high bandwidth of 10 MBaud on two channels helps to cope with the high network load caused by the increasing amount of innovative electronic systems in modern vehicles.

The communication system's specifications are released by the FlexRay consortium which is widely supported by vehicle manufacturers and suppliers worldwide.

Fullpass experiment

In a fullpass experiment, the complete electronic control unit program is executed on the experimental target.

Function

A grouping object for software systems that does not feature its own functionality. Modules or functions are assembled and connected in a function; they are thus clearly arranged and can be easily reused.

Graphical framework

The window that displays after the start of INTECRIO. The different INTECRIO components are integrated in the graphical framework.

Hardware system

A hardware system contains the complete description of a hardware topology, consisting of the descriptions of the associated ECUs (experimental targets) as well as the descriptions of the interfaces (bus systems) between the devices.

Implementation

An implementation describes the conversion of the physical task definition (of the model) into executable fixed-point code. An implementation consists of a (linear) conversion formula and a limiting interval for the model values.

Integration

The convergence of model code, which may have been developed by different partners or with different tools, to control algorithm, the configuration of this algorithm for the hardware on which it is supposed to run, and finally the creation of an executable file.

INTECRIO

INTECRIO is a tool that combines, i.e. integrates, the parts of the control algorithm created with different behavioral modeling tools that allows for creating and configuring a hardware system and the connection of this hardware system with the control algorithm.

legacy AUTOSAR module

AUTOSAR module imported with INTECRIO V5.0.0 or older.

MATLAB®

High-performance language for technical calculations; contains mathematical calculations, visualization and programming in one environment.

MATLAB® Coder™

Code generator for MATLAB code.

Module

A module in INTECRIO contains the generic description of a functionality for an electronic control unit. For example, it corresponds to an ASCET project or a Simulink model.

OS configurator and OSC

The task of the operating system configuration is carried out within INTECRIO the *OS configurator* and the OSC editor. The OSC is part of the OS configurator, an easy to handle editor for the operating system configuration that provides the user with a quick overview of the system and allows for editing the configuration in an application-oriented display.

Process

A process is a simultaneously executable functionality that is activated by the operating system. Processes are specified in modules and do not feature any arguments/inputs or result values/outputs.

Processor

see CPU

Project Configurator

The project configurator is part of the integration platform of INTECRIO. It is used to specify software systems and system projects.

Project Integrator

The project integrator combines all the components of the system (modules and functions, hardware interfacing, OS configuration, etc.) into an executable file.

Prototype

Completely executable file for an experimental target system. Such a prototype shows the software functions in practical use – entirely with different goal directions and in a different appropriation.

Rapid prototyping

The execution of a software on an experimental target, i.e. a computer with an interface to the vehicle.

RTA-Trace

Discontinued software tracing tool that can monitor system behavior over a versatile interface to the ECU. Existing installations can still be used.

Runnable entity

see RE

Runtime environment

see RTE

Sensor

Sensors convert a physical or chemical (usually non-electrical) quantity into an electrical quantity.

Service point

A service point is an encapsulation of a process in the ECU software. It provides data transfer actions to and from the target system; these actions can be enabled and configured by the user.

Service point cluster

A group of service points that are executed in the ECU with the same priority (service points located in the same ECU task).

Service point cluster group

A group of service point clusters. The group contains all service points of all tasks that can potentially be invoked at the same time in the ECU.

Simulink®

Tool for modeling, simulation and analysis of dynamic systems. The models can be imported in INTECRIO.

Simulink® Coder™

Code generator for Simulink and Stateflow models. Requires the MATLAB® Coder™.

Stateflow®

Tool for modeling and simulation of complex event-controlled systems. It is seamlessly integrated in MATLAB and Simulink,

Software system

A software system contains the generic parts of the ECU description: modules, functions and connections.

System project

A system project combines a hardware system, a software system, an environment system (if applicable), the mapping of the signals and the configuration of the operating system in a common project and allows for the generation of executable code.

Task

A task is an ordered collection of processes that can be activated by the operating system. Attributes of a task are its application modes, activation trigger, priority and modes of its scheduling. Upon activation, the processes of the task are executed in the specified order.

Validation

Process for the evaluation of a system or a component with the purpose of determining whether the application purpose or the user expectations are met. Therefore, the validation is the check whether the specification meets the user requirements, whether the user acceptance is reached by a function after all.

Verification

Process for evaluating a system or a component with the purpose of determining whether the results of a given development phase correspond to the specifications for this phase. Therefore, software verification is the check whether an implementation of the specification specified for the respective development step is sufficient.

Virtual prototyping

Function developers create virtual prototypes of electronic vehicle functions and test them on the PC.

Workspace

The workspace combines all the data generated while working with INTECRIO. From the WS browser, i.e. the tree view of the workspace, you can call up all the components of INTECRIO.

X-Pass experiment

Mixture of bypass and fullpass experiment. The experimental target (ES900, ES800) utilizes the ECU with bypass hooks as interface to the outside world.

Figures

Index