

# ETAS INCA-MCE V7.5



User Guide

## Copyright

The data in this document may not be altered or amended without special notification from ETAS GmbH. ETAS GmbH undertakes no further obligation in relation to this document. The software described in it can only be used if the customer is in possession of a general license agreement or single license. Using and copying is only allowed in concurrence with the specifications stipulated in the contract.

Under no circumstances may any part of this document be copied, reproduced, transmitted, stored in a retrieval system or translated into another language without the express written permission of ETAS GmbH.

**© Copyright 2024** ETAS GmbH, Stuttgart

The names and designations used in this document are trademarks or brands belonging to the respective owners.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://mathworks.com/trademarks) for a list of additional trademarks.

INCA-MCE V7.5 | User Guide R02 EN | 12.2024

## Contents

1	Introduction .....	6
1.1	Intended Use .....	6
1.2	Target Group .....	6
1.3	Classification of Safety Messages .....	6
1.4	Safety Information .....	7
1.5	Data Protection .....	8
1.6	Data and Information Security .....	8
2	About INCA-MCE .....	9
2.1	Standard ECU Access(INCA only) .....	9
2.1.1	iLinkRT at the Computer with INCA-MCE .....	11
2.1.1.1	iLinkRT V2 at the Computer .....	11
2.1.1.2	iLinkRT V3 at the Computer .....	11
2.2	Fast ECU Access with ES910 .....	12
2.2.1	Working Process .....	12
2.2.2	INCA-MCE Protocols .....	13
2.3	Direct ECU Access with MCE-PC .....	15
2.4	Application Examples .....	16
3	Installation .....	19
3.1	Preparation .....	19
3.1.1	System Requirements .....	19
3.1.2	Software Requirements .....	19
3.2	Installing .....	19
3.3	Licensing .....	21
4	System Configuration for Standard ECU Access .....	23
4.1	INCA Protocol Configuration for Standard ECU Access .....	23
4.2	Variable Selection for iLinkRT at the Computer .....	27
5	System Configuration for ES910 .....	29
5.1	INCA-MCE Fast ECU Access .....	29
5.1.1	iLinkRT .....	29
5.1.2	EtherCAT .....	30
5.2	ES910 Device Configuration .....	31
5.2.1	iLinkRT Configuration .....	31
5.2.2	EtherCAT Configuration .....	31

5.2.3	Ethernet Layer-2 Software Bridge Configuration .....	32
5.3	Adding an ES910 to the INCA Hardware Configuration .....	33
5.3.1	Configuring ES910 Devices for iLinkRT .....	34
5.3.2	Configuring ES910 Devices for EtherCAT .....	36
5.3.3	Assigning XETKs to the ES910 Interface .....	36
5.3.4	Assigning FETKs to the ES910 Interface .....	37
5.3.5	Supporting XCP on CAN FD via ES922 .....	38
5.4	INCA Protocol Configuration for ES910 .....	38
5.5	Multicast Network Configuration .....	38
5.6	Setup of INCA and iLinkRT System on same the Computer .....	40
6	System Configuration for MCE-PC .....	41
6.1	Connecting the MCE-PC .....	41
6.1.1	MCE-PC Hardware Interfaces .....	41
6.1.2	MCE-PC Network Connection .....	42
6.2	Adding a MCE PC to the INCA Hardware Configuration .....	44
6.2.1	Configuring the MCE-PC Device .....	44
6.2.2	Configuring the MCE-PC iLinkRT Interface .....	48
6.3	Assigning XCPonETH, EXTK and FETK Devices to the iLinkRT V3 Interface .....	49
6.4	Configuring iLinkRT V3 Protocol for MCE-PC .....	51
7	Variable Selection for INCA Standard ECU Access .....	52
7.1	Configuration via iLinkRT V3 .....	52
7.2	Configuration via INCA Experiment .....	52
8	Variable Selection for ES910 .....	54
8.1	Manual Variable Selection via INCA .....	54
8.1.1	Assigning Complete Sets of Variables .....	54
8.1.2	Selecting Individual Variables via the MCE Variable Selection Window .....	55
8.2	Variable Selection via Automation Interfaces .....	60
8.2.1	Tool API .....	60
8.2.2	ASAP3 .....	61
8.3	Order to be Adhered to in the Overall System .....	64
8.3.1	Using Measurement Variables and Calibration Variables .....	64
8.3.2	When Using the External Master Clock (EtherCAT Distributed Clocks [DC]) .....	64
8.4	Variable Selection for iLinkRT at the Computer .....	65

9	Variable Selection for MCE-PC .....	66
10	Measuring and Calibrating with ES910 .....	67
10.1	Measuring with ES910 Fast ECU AccessDevice .....	67
10.1.1	Special Features of ES910 iLinkRT V1.3/V2 interface .....	67
10.1.2	Special Features of ES910 EtherCAT interface .....	68
10.1.3	Measure Data Types not supported .....	69
10.1.4	Behavior with Configuration Changes .....	69
10.2	Calibrating with ES910 Fast ECU AccessDevice .....	70
10.2.1	Prerequisites for Calibration Access .....	70
10.2.2	General Information .....	70
10.2.3	Transfer of INCA-MCE Calibration Actions to INCA (Back Animation) .....	71
10.2.4	Arbitration .....	72
10.2.5	iLinkRT V1.3 .....	73
10.2.6	EtherCAT .....	73
11	Contact Information .....	76
	Glossary .....	77
	Index .....	78
	Figures .....	80
	Tables .....	81

# 1 Introduction

The INCA-MCE (Measurement and Calibration Embedded) add-on supports fast measurement and calibration methods. By connecting the test bench to the MCE hardware via a real-time capable protocol, extremely fast measurement and calibration access to the ECU is made possible.

In this document you will learn about the different possible setups of INCA-MCE and how to configure these single setups. Information on the correct use of INCA can be found in the INCA documentation.

For an overview of the features and functions of INCA-MCE, see chapter "[About INCA-MCE](#)" on page 9

## 1.1 Intended Use

INCA and INCA add-ons are developed and approved for automotive applications and procedures as described in the user documentation for INCA and INCA add-ons.

Calibration systems are used on engine and transmission test benches. Calibration access to the ECU is performed using measurement and calibration software connected to the ECU via appropriate hardware. With the measurement and calibration software INCA and the add-ons INCA-MCE and INCA-MCE-RT, you can establish fast access to the ECU.

INCA and the INCA add-ons are intended to be used in industrial labs and in test vehicles.

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

## 1.2 Target Group

This software product and this user guide address qualified personnel working in the fields of automotive ECU development and calibration, as well as system administrators and users with administrator privileges who install, maintain, or uninstall software. Specialized knowledge in the areas of measurement and ECU technology is required. This includes calibration procedures, software such as INCA and MDA, and software algorithms of systems you wish to calibrate.

## 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 that, if not avoided, will result in death or serious injury.

**WARNING**

**WARNING** indicates a hazardous situation that, if not avoided, could result in death or serious injury.

**CAUTION**

**CAUTION** indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

**NOTICE**

**NOTICE** indicates a situation that, if not avoided, could result in damage to property.

## 1.4 Safety Information

Observe the following safety information when working with INCA and INCA add-ons:

**WARNING****Risk of unexpected vehicle behavior**

Calibration activities influence the behavior of the ECU and the systems that are connected to the ECU.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Only perform calibration activities if you are trained in using the product and can assess the possible reactions of the connected systems.

**WARNING****Risk of unexpected vehicle behavior**

Sending messages via bus systems, such as CAN, LIN, FlexRay, or Ethernet, influences the behavior of the systems connected to it.

This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.

Only perform the sending of messages via a bus system if you have sufficient knowledge in using the respective bus system and can assess the possible reactions of the connected systems.

Adhere to the instructions in the ETAS Safety Advice and the safety information given in the online help and user guides. Open the ETAS Safety Advice in the INCA help menu ? > **Safety Advice**.

## 1.5 Data Protection

If the product contains functions that process personal data, legal requirements of data protection and data privacy laws shall be complied with by the customer. As the data controller, the customer usually designs subsequent processing. Therefore, he must check, if the protective measures are sufficient.

## 1.6 Data and Information Security

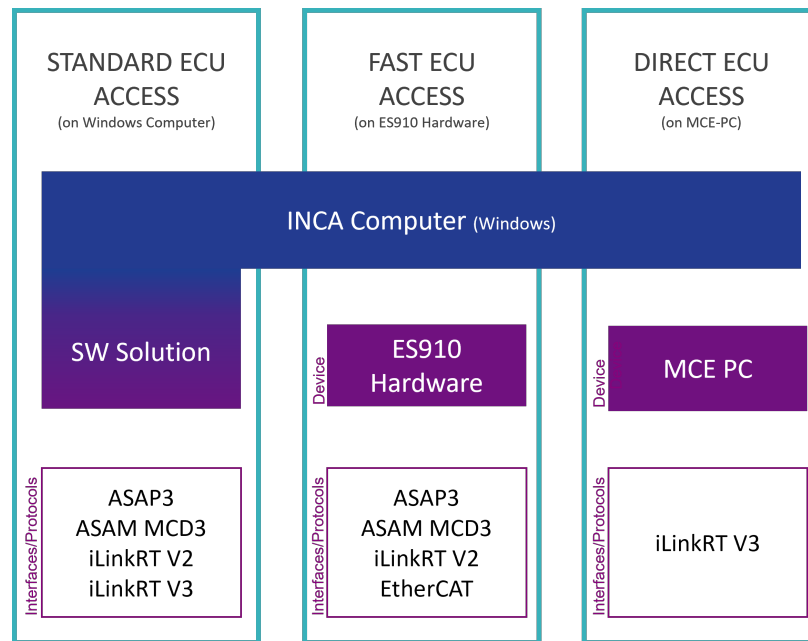
To securely handle data in the context of this product, see the INCA Help section "Data and Information Security".



## 2 About INCA-MCE

The INCA-MCE (Measurement and Calibration Embedded) and the INCA-MCE-RT add-on establishes a real-time connection between the test bench and the control unit via various interfaces and protocols. The main advantage of the INCA-MCE solution is the high throughput of measurement and calibration values.

INCA supports the following fast measurement and calibration methods:



**Fig. 2-1:** Calibration access solutions for INCA-MCE

The following chapters provide an overview of the features and functions of the three different calibration access solutions with INCA-MCE.

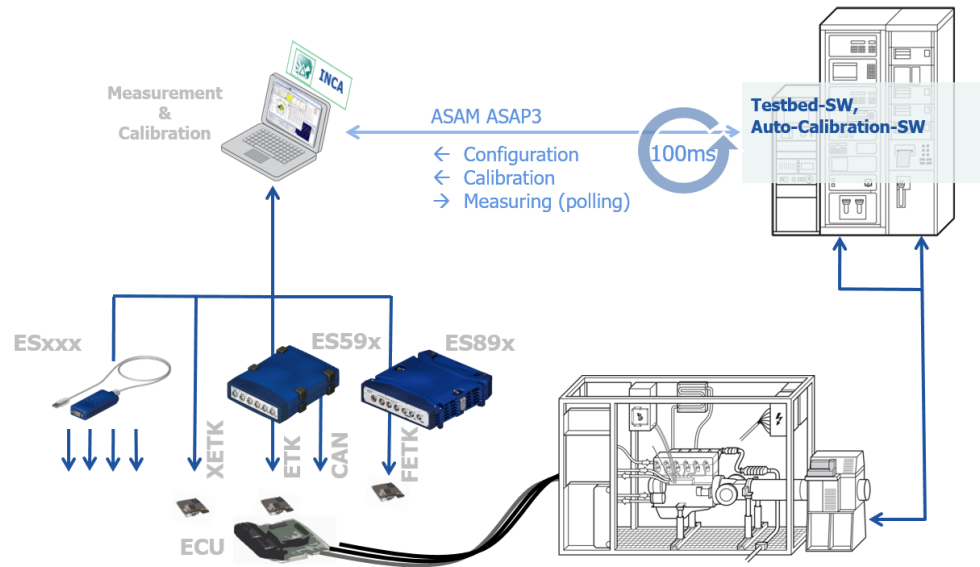
### 2.1 Standard ECU Access(INCA only)

Fig. 2-2 shows a schematic of a calibration system as it is commonly used today on most engine and transmission test benches.

Calibration access to the ECU is accomplished by using measurement and calibration software connected to the ECU using appropriate hardware.

Serial protocols such as CCP and XCP are used at the ECU interface. Specific ECU interfaces for development purposes, such as the ETK, are particularly well performing.

A test bench control system controls the test bench hardware and special measurement equipment via one or more automation buses such as PROFIBUS, EtherCAT, and CANopen.



**Fig. 2-2:** Typical test bench setup

The test bench software and the auto-calibration software accesses the measurement and calibration software via the ASAP3 interface and thus makes a closed control loop.

The automation software on the test bench system accesses control values based on their name and physical value. In the ECU, this data is stored according to its address and in hexadecimal form. The necessary conversion of the data takes place in the measurement and calibration software.

Calibration systems of this kind can be found at almost every engine and transmission test bench today.

### *Limitations*

The limitations of this system which make the introduction of new, advanced calibration methods difficult are:

- Relatively slow calibration cycles<sup>1)</sup>: The calibration time for a map of the magnitude "16\*16 calibration values" is about 100 ms in the setup described above.
- As the calibration is carried out via a non-real-time-capable protocol and the calibration software runs on a Windows operating system, the calibration time can vary significantly (jitter).
- ASAP3 is a blocking communication protocol. The automation software therefore cannot receive measure data from the ECU during a calibration task.

<sup>1)</sup> The time between the calibration request being sent by the test bench software and the positive response to the automation software that the calibration was successfully performed.

- ASAP3 does not support multi-master architecture. The automation software and the auto-calibration software therefore cannot access ECU values via ASAP3 simultaneously.

To overcome these limitations, ETAS has developed INCA-MCE which provides fast measurement and calibration access to the ECU.

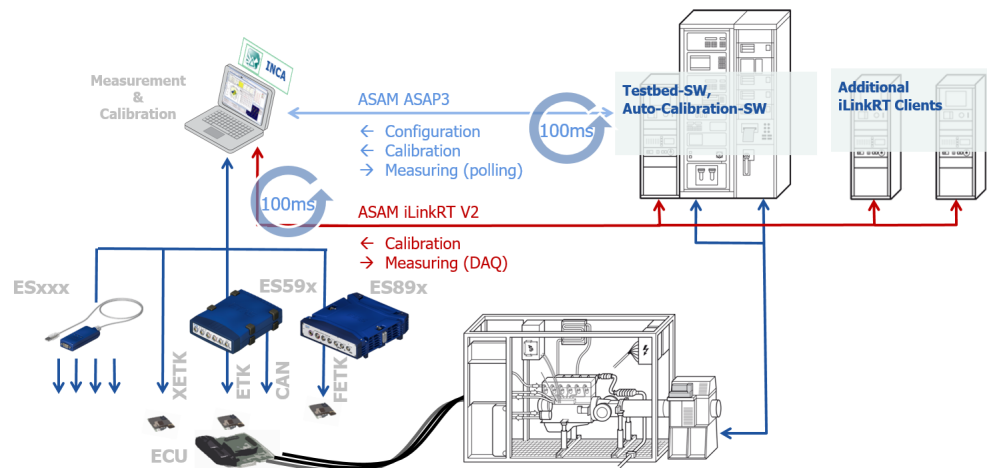
## 2.1.1 iLinkRT at the Computer with INCA-MCE

With iLinkRT at the computer it is possible to use the advantage of parallel event-based measuring and calibration with direct INCA connection. This allows to access all kind of devices connected to INCA. Even when the computer has lower performance as theFast ECU Accessthe advantages of the iLinkRT protocol improve the communication betweenINCA and the test bench clients.

There are two different approaches with iLinkRT V2 and iLinkRT V3.

### 2.1.1.1 iLinkRT V2 at the Computer

With iLinkRT V2 it is possible to switch betweenFast ECU Accessand simple computer connection without changing the test bench software. As iLinkRT V1.3 and V2 are compatible this can be done easily.



**Fig. 2-3:** Test bench system using the iLinkRT V2 at the computer solution

### 2.1.1.2 iLinkRT V3 at the Computer

With iLinkRT V3 it is possible to operate the connection between INCA and multiple clients without ASAP3. iLinkRT V3 offers full configuration possibilities and special support when multiple clients need to configure the measuring and calibration setup.

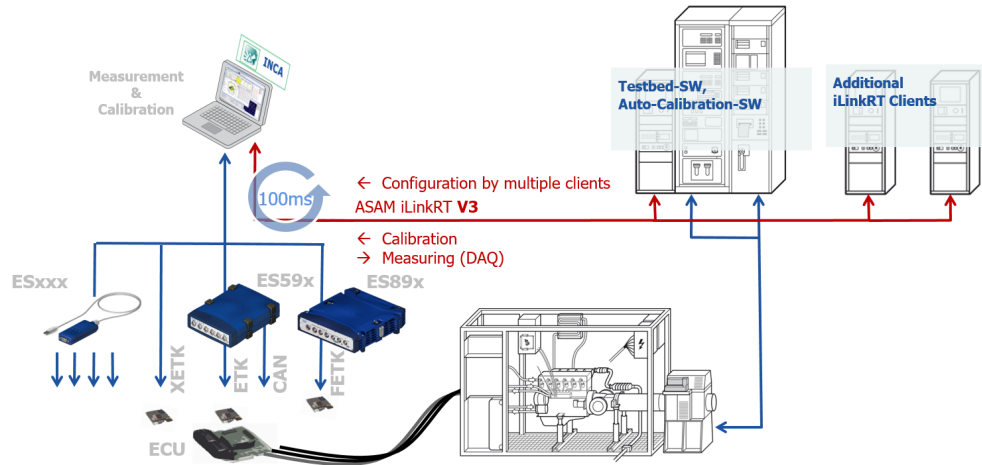


Fig. 2-4: Test bench system using the iLinkRT V3 at the computer solution

## 2.2 Fast ECU Access with ES910

The Fast ECU Access allows to overcome the performance restrictions of the Windows computer. The ES910, a special measurement device, allows much faster measurement access than the Windows computer.

- The test bench software is able to access ECU values based on their name and physical value.
- Devices connected to the ES910 can use its fast routines to transfer data from the device via iLinkRT V1.3 or EtherCAT
- Devices connected to the Windows computer are handled in a 100ms measurement loop. The measured data is transferred via ASAP3 or MCD 3MC.
- The test bench software configures the complete setup via ASAP3 or MCD3MC.

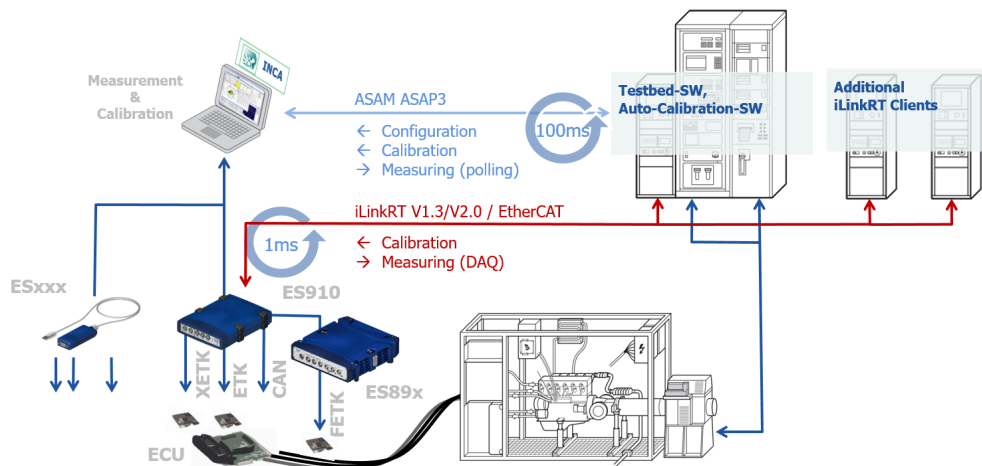


Fig. 2-5: Test bench system using the MC embedded solution

### 2.2.1 Working Process

The working process with INCA-MCE is divided into two main phases:

Configuration Phase	"System Configuration for ES910" on page 29 "Variable Selection for ES910" on page 54
Measurement and Calibration Phase	"Measuring and Calibrating with ES910" on page 67

## Preconfiguring the Real-Time-Capable Hardware

The ES910 must be preconfigured to ensure that the relevant information from the A2L file of the ASAM MCD 2MC ECU is available for real-time operation. To perform this preconfiguration, you must select the required measurement and calibration variables. This can be done manually in INCA or via the automation interface. Once configured, the test bench control system downloads a list of selected variables via the MCE protocol.

## Measuring and Calibrating with INCA

During the measurement and calibration phase, the ES910 continuously collects ECU values to be measured and transmits it directly to the test bench via the MCE protocol. Any calibration performed is displayed in the INCA calibration software, regardless of which calibration path was used. This ensures that the test bench engineer always has an up-to-date view of the ECU's current state.

### 2.2.2 INCA-MCE Protocols

For faster ECU access for measurement and calibration access, INCA-MCE supports two alternative protocols: iLinkRT and EtherCAT.

#### iLinkRT

iLinkRT V1 was jointly developed by AVL List GmbH and ETAS specifically for INCA-MCE. For further development the standard has been handed over to ASAM. ASAM continued iLinkRT with the Versions V2 and V3. iLinkRT V3 offers additionally to measuring and calibration the possibility to configure INCA by multiple clients.

The key benefits of iLinkRT are:

- An optimized command set, focusing on measurement and calibration
- Performance-optimized calibration access
- Name-based, physical representation
- Non-blocking protocol
- Open protocol based on XCP, an existing automotive standard
- Multi-master and multi-slave support
- Simple implementation for the migration of existing test bench systems to INCA-MCE

## EtherCAT

EtherCAT is a real-time-capable automation bus based on Ethernet. In addition to Fast ECU Access, it also enables the automation of test bench components such as the combustion analyzer and special measurement equipment. A global clock synchronizing the different components makes it possible to merge decentrally recorded data and to perform joint analysis. EtherCAT was initiated by the company Beckhoff as an open, real-time-capable protocol for automation purposes. Today, it is developed and maintained by the EtherCAT Technology Group with more than 1500 members. It is characterized by outstanding performance (e.g. transfer of 1000 I/Os in 30  $\mu$ s), a very simple wiring concept and openness for other protocols.

## Comparison of iLinkRT and EtherCAT for Fast ECU Access

iLinkRT and EtherCAT are different types of protocol and are thus difficult to compare with each other. While iLinkRT is a protocol specially designed for INCA-MCE, EtherCAT is an automation field bus with very complex functions that cover a wide application range.

Regarding INCA-MCE, both protocols provide similar functionality and thus comparable performance. When looked at in detail, both protocols have their strengths and weaknesses (see [Tab. 2-1](#)). Which protocol is to be preferred depends very much on the use at the particular test bench.

Criteria	iLinkRT	EtherCAT
Measurement and calibration performance	++	++
Supported calibration methods	+	0
Real-time capability for measurement and calibration tasks	+	++
Capability to automate all components of the test bench	-	+
Ease of implementation	++	-
Cost and effort of migration for existing test bench systems	+	-
Communication type	Triggered by ECU and test bench	Triggered by Ether-CAT bus master
++ very well suited; + well suited; 0 suited; - less suited; -- not suited		

**Tab. 2-1:** Comparison of iLinkRT and EtherCAT when used with INCA-MCE

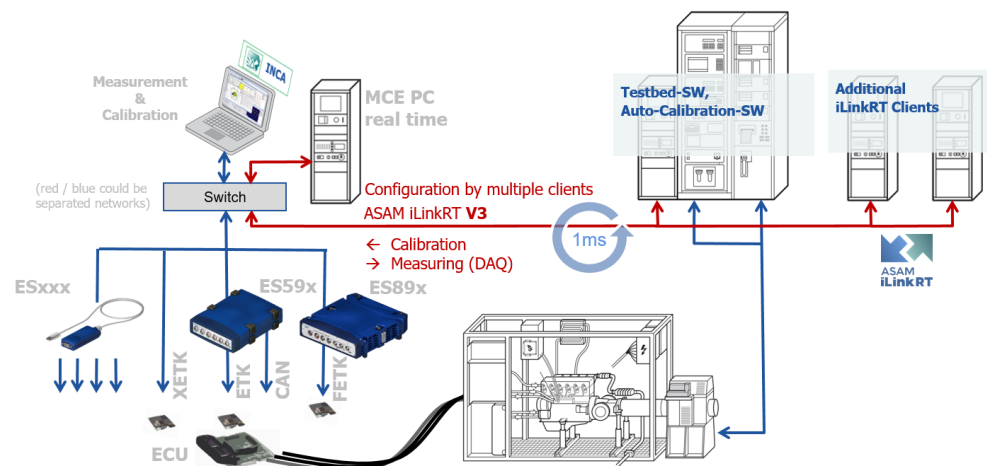
It is therefore impossible to give a general recommendation for one or other protocol. As a rule of thumb, it can be said that iLinkRT, due to its low migration effort, should be the preferred choice when realizing Fast ECU Access with INCA-MCE at existing test benches. EtherCAT is the right choice when the complete test bench with its components is to be automated via EtherCAT.

## 2.3 Direct ECU Access with MCE-PC

The Direct ECU Access uses the MCE-PC device as additional hardware to overcome performance limitations. However, unlike the ES910, all connected devices are supported, regardless of whether they are connected to the Windows or the Linux computer.

For the MCE-PC, INCA ensures that the measured data is available in INCA and iLinkRT V3.

- The test bench software can access ECU values based on their name and physical or implementation value.
- Devices assigned to the MCE-PC can use the much faster MCE-PC measure routines to transfer data from the device via iLinkRT V3.
- Devices assigned to the Windows computer are handled in a 100ms measurement loop. The measured data is transferred to the MCE-PC device and by this sent via iLinkRT V3.
- The test bench software configures the complete setup via iLinkRT V3.



**Fig. 2-6:** Test bench system using the MC embedded solution

This makes the Direct ECU Access functionally compatible with the iLinkRT software solution. The iLinkRT V3 client can be connected either to the MCE-PC or to the INCA Windows computer, see .

## Working Process

The working process with INCA-MCE is divided into two main phases:

---

Configuration Phase	"System Configuration for MCE-PC" on page 41 "Variable Selection for MCE-PC" on page 66
---------------------	--

---

Measurement and Calibration Phase	"Measuring and Calibrating with ES910" on page 67
-----------------------------------	---

---

### *Preconfiguring the MCE-PC*

The Direct ECU Access device must be preconfigured to ensure that the relevant necessary information from the A2L file of the ASAM MCD 2MC ECU is available for real-time operation. To perform this preconfiguration, you must select the required measurement and calibration variables.

There are two ways possible:

- Use all measurements of the INCA experiment.
- Allow the test bench software to select variables via iLinkRT V3.

It is also possible to combine both methods.

### *Measuring and Calibrating with INCA*

During the measurement and calibration phase, the MCE-PC device continuously collects the ECU values to be measured and forwards these directly to the test bench via the iLinkRT V3 protocol. Any calibration performed via iLinkRT V3 is promptly displayed in the INCA experiment. This ensures that the test bench engineer always has an up-to-date view of the ECU's current state.

## 2.4 Application Examples

The largest benefit of INCA-MCE is the much faster ECU calibration access when used together with fast hardware access: In comparison to an ASAP3 interface with an RS232 connection, a single map of the magnitude 16\*16 can be calibrated 400 times faster with INCA-MCE via Fast ECU Access hardware like the ES910. It is 160 times faster in comparison to ASAP3 with an Ethernet connection interface.

Looking at the system setup, it becomes obvious that with INCA-MCE the limiting factor is no longer the connection of the test bench to the calibration software but the ECU interface (CCP, XCP, ETK, etc.). If the increase in performance with a high-performance ECU interface such as the ETK is 100 - 500 times faster, the increase in performance with a serial interface such as CCP is only 10 - 20 times faster.

By using Fast ECU Access with INCA-MCE, the following results could be achieved in an evaluation phase:

- **Running real-time controllers on the test bench system**



In early phases of development, not all ECU functionality is implemented or calibrated. It is therefore often difficult to start up an engine in this phase. This is particularly the case if functions for component protection, for example, exhaust gas temperature control or knock control, are missing. The Fast ECU Access via INCA-MCE allows you to substitute missing functions by generic real-time controllers running on the test bench system. Controllers running on the test bench system in parallel to the manual calibration (e.g. combustion controller, Lambda controller, etc.), also help to decrease the level of freedom and thus reduce the complexity of calibration tasks.

#### – Avoidance of limit violations

In order to guarantee the stability of measurements, it is important to adjust a large number of calibration variables simultaneously to avoid critical limits being exceeded. Due to the sequential adjustment of maps, variations can lead to unstable operation (Fig. 2-7 on the left).

With its Fast ECU Access, INCA-MCE makes it possible to adjust the maps in a number of smaller intermediate steps (Fig. 2-7 - on the right) and thus avoid limit violations.

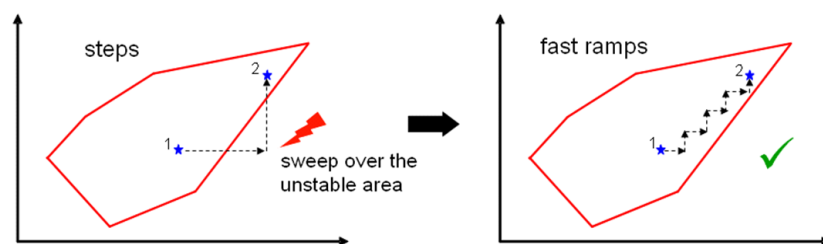


Fig. 2-7: The advantage of fast ramps in terms of a stable operating range<sup>1)</sup>.

#### – Improved data quality and reduced overshooting of the system

At the change of the operating point with fast ramps, it is important that the intermediate steps are as small as possible and that the different parameters are adjusted as fast as possible. The key benefits of INCA-MCE are obvious when examining a test run in which after 15 seconds of stabilization time the next variation point is approached. Or when using the standard ASAP3 interface, seven maps can be changed with three intermediate steps in 13 seconds (Fig. 2-8 - on the left). In contrast, by using INCA-MCE and iLinkRT these same seven maps can be overwritten in steps of 10 Hz so that a virtually continuous interpolation is possible (Fig. 2-8 - on the right).

<sup>1)</sup> Source: ATZ elektronik 2/2008, "iLinkRT High Performance ECU Calibration"

In addition to the advantage of a shorter overall calibration time (approx. 30%), there is also a significant improvement of data quality. Regarding the soot traces in Fig. 2-8, the dynamic excitation is significantly lower due to the smaller intermediate steps during calibration. This means that overshooting of the response (especially soot) hardly ever happens.

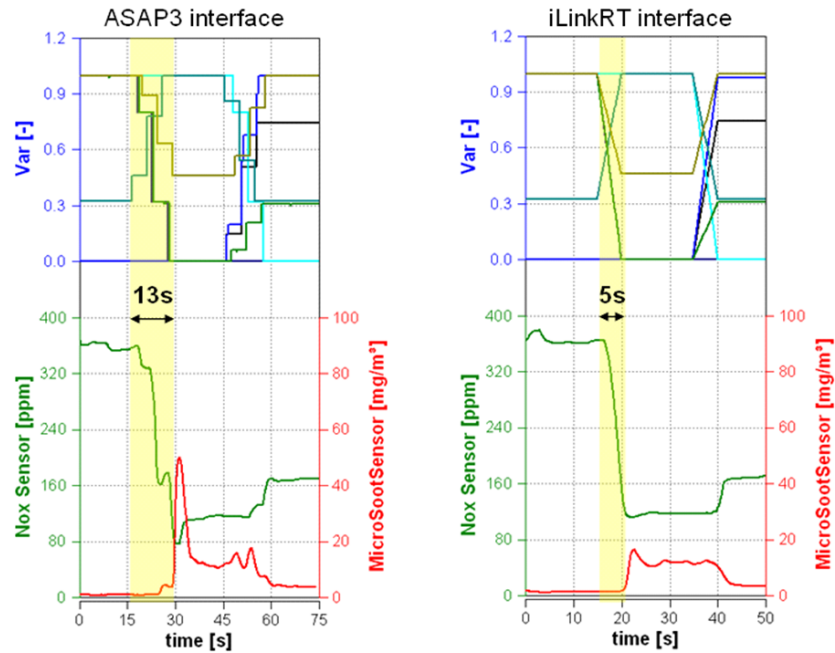


Fig. 2-8: ECU ramps with ASAP3 or iLinkRT<sup>1)</sup>

<sup>1)</sup> Source: ATZ elektronik 2/2008, "iLinkRT High Performance ECU Calibration"

## 3 Installation

This chapter contains information on the delivery scope, software requirements for the installation, as well as on the preparation and execution of the installation.

### 3.1 Preparation

Check that the delivery package is complete and that your system meets the necessary system requirements before starting the installation procedure. Depending on the operating system used and your network connection, you must also check that you have the necessary authorization.



#### Note

Before starting the installation procedure and using the software, check you have the necessary access rights to the Windows Registry. If in doubt, ask your system administrator.

#### 3.1.1 System Requirements

The system requirements for operating INCA are also sufficient for operating the INCA-MCE add-on. INCA must be installed and configured to the system being calibrated.

#### 3.1.2 Software Requirements

Make sure that your INCA software as well as other required add-ons and software are installed and are the most recent versions. This is particularly the case for:

- INCA
- Add-on INCA-MCE (for iLinkRT protocol, ES910, MCE-PC)
- Add-on INCA-MCE-RT (for MCE-PC)
- HSP

### 3.2 Installing

This section describes how to install INCA-MCE.

#### To download the INCA installation package

1. On the ETAS homepage, click **Download Center**.
2. Search for **INCA > INCA V7.5 > Software**.
3. Download the Installation Package (\*.zip).
4. In the Windows File Explorer, select the downloaded ZIP file, right-click and select **Properties**.
5. On the **General** tab, at the Security option disable the **Unblock** button.
6. Extract the complete structure of the ZIP file.

 **Note**

The fully qualified file name of all components of the setup and the directory name are subjects of a restriction and must fall below a certain character length. The character length is calculated individually.

The path length of the folder where you save the `Setup_ServicePack.exe` shall not exceed 80 characters.

Do not change the folder structure, the folder-names, or any name of installation executables of the installation package.

### To install the software

1. Close all open ETAS software.

 **Note**

Do not execute other software updates in parallel, for example an update of the operating system. Wait until all updates are installed and restart the computer before starting the installation.

2. Execute the `Setup_ServicePack.exe` file from your installation directory.  
The "Service Pack Installer" window opens.
3. In the **Install** column, enable your desired software products and add-ons.  
To select all, enable the corresponding package on top level.
4. Read the license agreement and enable **I read and accept the end user license agreement** option.
5. Select your preferred setup language.


 **Note**

The language you select changes the user interface language for the Service Pack Installer and the language for newly installed and for all already installed INCA and INCA add-on products.

6. Click on **Install**.  
The installation process is initiated.
7. Click on **Restart Options**.

The "Restart" dialog box opens.

### Note

It is recommended to restart your system after the installation is finished. Some installations require a restart in between the installation routine. In this case, a warning sign  is displayed in the Status column. After a restart, the installation will continue automatically.

8. Select the desired restart options.
9. Click **Ok**.

All your desired software products and add-ons will be installed in silent mode. This means no additional dialog windows are displayed during installation. You must then check that a license for INCA-MCE is installed.

### Note

It is possible to work offline with INCA-MCE without a license – the manual configuration of INCA-MCE variables and storing them in the Experiment Environment do not require a license. A valid license is, however, necessary when this configuration is to be transferred to an MCE device or iLinkRT at the computer is used!

For further information, see the INCA Installation Guide.

## 3.3 Licensing

A valid license is required to use the software. You can obtain a license in one of the following ways:

- from your tool coordinator
- via the self-service portal on the ETAS website at [www.etas.com/support/licensing](http://www.etas.com/support/licensing)
- via the ETAS License Manager

To activate the license, you must enter the Activation ID that you received from ETAS during the ordering process.

For more information about ETAS license management, see the [ETAS License Management FAQ](#) or the ETAS License Manager help.

### To open the ETAS License Manager help

The ETAS License Manager is available on your computer after the installation of any ETAS software.

1. From the Windows Start menu, select **E > ETAS > ETAS License Manager**.  
The ETAS License Manager opens.
2. Click in the ETAS License Manager window and press F1.  
The ETAS License Manager help opens.



**Note**

---

In order to use INCA-MCE, you need the following valid ETAS licenses:

- INCA
- INCA-MCE
- INCA-MCE-RT

## 4 System Configuration for Standard ECU Access

The INCA-MCE standard ECU access allows to use iLinkRT protocol without additional hardware. The ECU access, measurement devices and monitoring devices are directly connected to the computer where INCA is running.

### Note

iLinkRT V2 requires additionally to the data channels an ASAP3 channel for the configuration.

### Note

iLinkRT V3 supports configuration and needs no ASAP3 support.




The main steps are:

- Configuring the interface in INCA user options
- Variable selection via ASAP3 / iLinkRT V2 / iLinkRT V3

### 4.1 INCA Protocol Configuration for Standard ECU Access

If you use iLinkRT at the computer you start the ASAM MCD 3MC interface directly from INCA. To start the desired version of the ASAM MCD 3MC interface, you first have to select the interface in the INCA User Options.

To start the ASAP3/iLinkRT V2 interface:

1. In the INCA Database Manager , click  > **General** tab > **ASAM-MCD 3MC interface version** > **ASAP3/iLinkRT V2.0**.
2. Start an experiment in the INCA Experiment Environment .
3. In the **Components** menu of the Experiment Environment, select **ASAM-MCD 3MC - Open ASAM-3MC Interface**.

The **ASAM MCD 3MC** window opens.


When you first start the **ASAP3/iLinkRT V2.0** interface, the **Establish Connection** dialog box opens. Enter the connection parameters to establish a connection, see "[Establish Connection Dialog Box](#)" on page 25.

If the connection can be made to the testbed, the status line displays **Online**.

### Note


Ensure that the selected ports are open. To ensure open ports, check the Windows Firewall settings.

To start the iLinkRT V3 interface:

1. In the INCA Database Manager, click  > **General** tab > **ASAM-MCD 3MC interface version** > **iLinkRT V3**.

2. In the **Utilities** menu, select **ASAM-3MC - Open ASAM-3MC Interface**

*or*

Start an experiment in the INCA Experiment Environment  and in the **Components** menu, select **ASAM-3MC - Open ASAM-3MC Interface**.

The **ASAM MCD 3MC** window opens. The status line displays **Ready**.

When you first start the iLinkRT V3 interface, the **Establish Connection** dialog box opens. Enter the connection parameters to establish a connection, see "[Establish Connection Dialog Box](#)" on the next page.

The **ASAM MCD 3MC** window opens.

If the connection can be made to the testbed, the status line displays **Online**.

 **Note**

Ensure that the selected ports are open. To ensure open ports, check the Windows Firewall settings.



## Establish Connection Dialog Box

To open this dialog box, in the **Connection** menu select **Establish Connection**.

This dialog box contains the following fields and buttons:

### Use IPv6

Activates the use of an IPv6 address. If the checkbox is deactivated, IPv4 is used as standard IP address.

### ASAP3

#### RS232 / V.24 connection ("Serial")

#### Note

You can make this setting only for the ASAP3 / iLinkRT V2.0 interface.

Defines that the connection to the test bench is established via a serial interface.

### Port

Defines the serial interface.

**Baud rate**

Defines the baud rate for the serial connection.

**Network connection**

Defines that the connection to the test bench is established via a network card or network link.

**Network card**

Defines the network card via which the connection to the test bench is established.

**Local network address**

Defines a local network address to establish a connection to the test bench.

**Local address**

Defines a local address to establish a connection to the test bench.

**Local port**

Defines the port for ASAP3 connection.

*iLinkRT Commands***Network card**

Defines the network card via which the connection to the test bench is established.

Useful when no static address available, for example, DHCP.

**Local network address**

Defines a local network address to establish a connection to the test bench.  
Useful for static addresses

**Local address**

Defines a local address to establish the command connection to the test bench

**Local port**

Defines the port for iLinkRT command connection.

**Max Payload UDP**

UDP telegrams have a maximum size of 65535 bytes. However, the network infrastructure may be restricted to smaller telegram sizes. To avoid telegram losses the max size of telegrams can be restricted to sizes that all used network components support. (508 .. 65535).

*iLinkRT Get All Server***Version**

Select the iLinkRT protocol version number.

**Broadcast / Multicast address**

Defines the IP address to be used for Get All Server and its responses.

**Local port**

Defines the port for iLinkRT Get All Server connection `RT_GET_ALL_SERVER`.

*iLinkRT Measurement / Events***Use unicast**

Defines that unicast sending is used instead of multicast.

**Multicast address**

Defines the IP address to be used for multi cast (unicast) measurement / event sending.

**Local / Multicast port**

Defines the UDP port to be used for measurement / event sending.

**Max Payload UDP**

UDP telegrams have a maximum size of 65535 bytes. However, the network infrastructure may be restricted to smaller telegram sizes. To avoid telegram losses the max size of telegrams can be restricted to sizes that all used network components support. (508 .. 65535).

*iLinkRT V3 Watchdog***Interval (sec)**

 **Note**  


---

You can make this setting only for the iLinkRT V3.0 interface.

Defines the maximum time allowed between two commands sent by the client. Valid values are between 1 and 1000 seconds. To disable the watchdog, enter **0**.

**■ Establish Connection**

Establishes the connection and closes the dialog box.

**■ Cancel**

Closes the dialog box without establishing the connection.

## 4.2 Variable Selection for iLinkRT at the Computer

If you completed the protocol configuration, you can now use INCA with iLinkRT at the computer.

The configuration of the measuring, recording, and calibration is done either via ASAP3/iLinkRT V2.0 or via iLinkRT V3.0.

To use ASAP3/iLinkRT V2.0

1. Select the measure and calibration variables via ASAP3.
2. Start the measurement via ASAP3.

INCA sends the selected measure variables automatically via the iLinkRT measurement port.

INCA immediately executes iLinkRT calibration commands which are received via the iLinkRT command port.

To use iLinkRT V3.0

1. Select devices via iLinkRT.
2. Select the measure and calibration variables via iLinkRT.
3. Start the measurement via iLinkRT.

iLinkRT sends the selected measure variables automatically via the measurement port.

INCA immediately executes iLinkRT calibration commands which are received via the iLinkRTT command port.

## 5 System Configuration for ES910

This chapter contains information on how to configure ES910 devices and the used interfaces.

The main steps are:

- Configuring the ES910 device
- Adding the ES910 device to the INCA HWC
- Configuring iLinkRT V1.3 / EtherCAT for ES910
- Assigning XETKs and FETKs
- Protocol configuration in INCA
- Configuring ETAS Network for Multicast (FETK, XETK)
- Use INCA and the testbench software on the same computer



### Note

INCA supports for one HWC up to four ES910 modules.

### 5.1 INCA-MCE Fast ECU Access

For fast measurement and calibration access of ECU data, The fast access requires an ES910 device that handles the high speed data transfer to the testbench. INCA-MCE supports the two alternative protocols iLinkRT and EtherCAT. ECU data is accessed on a name-based basis in physical representation.



### Note

When choosing the protocol, please ensure that all system components support the relevant protocol.

#### 5.1.1 iLinkRT

iLinkRT is an ASAM standard and defines an open protocol specially developed for Fast ECU Access. iLinkRT V2 supports an existing ASAP3 connection by efficient measurement and calibration support. iLinkRT V3 works stand alone and offers additionally configuration possibilities for multiple clients.



### Note

To find out more about iLinkRT, read the protocol specification "iLinkRT XCP Implementation Description" that you can obtain from ETAS. This document can be opened once INCA-MCE has been installed via the Start menu (**Programs > E > ETAS INCA 7.5 > INCA V7.5 Manuals and Tutorials**).

To be able to use iLinkRT, the protocol version must be compatible with the INCA-MCE version. The following table shows which protocol version of iLinkRT is compatible with your installed INCA-MCE version:

INCA-MCE Version	iLinkRT Protocol Version	Compatible with
INCA-MCE V7.5	ASAM iLinkRT V2.0	iLinkRT V1.2, V1.3

**Tab. 5-1:** INCA-MCE version and version of the iLinkRT protocol

The test bench software of various manufacturers already support iLinkRT. ETAS provides the "iLinkRT Master COM Server for Windows" as an alternative. It makes available an API for linking iLinkRT V2 / V3 for proprietary software.

 **Note**

To find out whether the test bench software you are using supports iLinkRT, contact the customer service department of the relevant manufacturer. Contact ETAS for information on "iLinkRT Master COM Server for Windows".

## 5.1.2 EtherCAT

EtherCAT is a standardized, real-time-capable automation bus based on Ethernet.

 **Note**

For more information on EtherCAT, refer to <http://www.ethercat.org/>. The Technology Group (ETG) is responsible for the development of the protocol.

 **Note**

To find out more about EtherCAT, read the specifications on ECU data representation and implementation details in connection with the EtherCAT protocol that you can obtain from ETAS.

To be able to use EtherCAT, the protocol version must be compatible with the INCA-MCE version. The following table shows which protocol version of EtherCAT is compatible with your installed INCA-MCE version:

INCA-MCE Version	EtherCAT	
	ETG.1000	ETG.2000
INCA-MCE	V1.0.2	V1.0.1

**Tab. 5-2:** INCA-MCE version and EtherCAT specification

 **Note**

To be able to support EtherCAT for INCA-MCE, the system components, for example, the automation system, must be integrated in the EtherCAT Master. The relevant manufacturer can provide information on support and versions.

The electronic description of the EtherCAT device (ESI file, EtherCAT Slave Information File) is available on the INCA-MCE device and can be loaded onto the computer via the web interface. For more details on the web interface, refer to the ES910.3 Prototyping and Interface manual.

## 5.2 ES910 Device Configuration



### Note

To be able to configure an ES910 device, up-to-date firmware must be available on the device. Use the HSP Update Tool to check the firmware status of the device.

Refer to the Download section of the ETAS homepage (search for "HSP") for the current version of the Hardware Service Pack.

### 5.2.1 iLinkRT Configuration

The iLinkRT interface of the ES910 needs no special configuration.

### 5.2.2 EtherCAT Configuration

As the EtherCAT slave controller for the MCE devices is hardware-oriented, the MCE devices need to be configured.

#### To configure an MCE device for EtherCAT

1. Open the web interface of the MCE device (see ES910.3 Prototyping and Interface Module manual).
2. Select **Device Configuration > Port Settings**.
3. For the **Micro I/O Interface** port enable the setting "Ethercat slave".

**ETAS/ES910**

Home  
 Mail

- ↳ Device status
  - Adaptive Partitioning
  - Temperature Monitor
- ↳ Device configuration
  - Port Settings**
  - Ethernet L2 Bridge
  - Wakeup
  - Date and Time
  - Reset Settings
- ↳ Ethercat
  - Log files
  - MCE log severity
  - Core file upload
  - 3rd party licenses

ETAS

ES910

**ES910 port settings configuration interface**

The CAN ports of the ES910 can be configured to support either High Speed CAN or Low Speed / Fault Tolerant CAN. The behaviour of both CAN ports can be set up here.

The UIO port of the ES910 (ES910.3 only) can be configured to support either one daisy chain of ES400 modules or act as the far end of an EtherCAT daisy chain (EtherCAT slave).

In case the UIO port is configured to act as EtherCAT slave, either enhanced or legacy (AES compatible) mode can be selected. This has an impact on the structure of the EtherCAT object dictionary. Consult the INCA-MCE manual for details.

Changing the UIO port and/or EtherCAT mode setting requires a reboot of the device to become effective.

Please take care that these settings match the RTIO settings in the INTECRIO created RP model which is running on this device.

Select interface port configuration

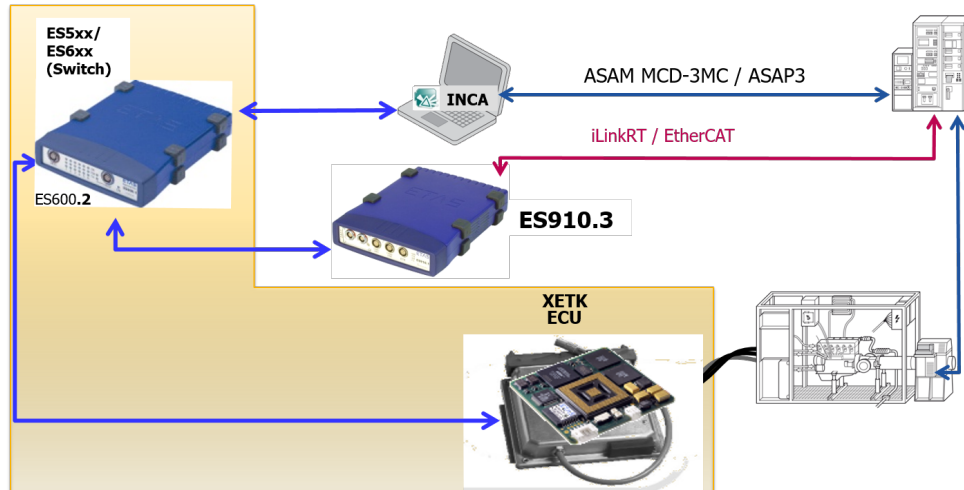
CAN interface port no. 1 (CAN1)	<input checked="" type="radio"/> High Speed <input type="radio"/> Low Speed / Fault Tolerant
CAN interface port no. 2 (CAN2)	<input checked="" type="radio"/> High Speed <input type="radio"/> Low Speed / Fault Tolerant
LIN interface port no. 1 (LIN1_RESISTOR)	<input checked="" type="radio"/> Master resistor disconnected <input type="radio"/> Master resistor connected
LIN interface port no. 1 (LIN1_SUPPLY)	<input checked="" type="radio"/> Port supplied externally <input type="radio"/> Port supplied internally
LIN interface port no. 2 (LIN2_RESISTOR)	<input checked="" type="radio"/> Master resistor disconnected <input type="radio"/> Master resistor connected
LIN interface port no. 2 (LIN2_SUPPLY)	<input checked="" type="radio"/> Port supplied externally <input type="radio"/> Port supplied internally
Micro I/O interface (UIO)	<input type="radio"/> ES400 daisy chain <input checked="" style="border: 2px solid red; border-radius: 50%;"/> Ethercat slave
EtherCAT mode (ETHERCAT_MODE)	<input checked="" type="radio"/> EtherCAT enhanced mode <input type="radio"/> EtherCAT legacy mode (AES compatible)

4. Click **Save config**.
5. Reboot the MCE device.

### 5.2.3 Ethernet Layer-2 Software Bridge Configuration

An XETK cannot be connected directly to the ES910 in an INCA-MCE configuration but must be linked via a module (ES5xx, ES6xx).





**Fig. 5-1:** Connecting an XETK for MCE

To configure Ethernet layer-2 software bridge

1. Select **Device Configuration > Ethernet configuration**.
2. Select "L2 Bridge off" with **L2 bridge configuration**.
3. Click **Save config**.
4. Reboot the ES910.



### Note



It is not possible to use EtherCAT and ES4xx Daisy Chain devices (incl. ES63x) simultaneously on an ES910.

## 5.3

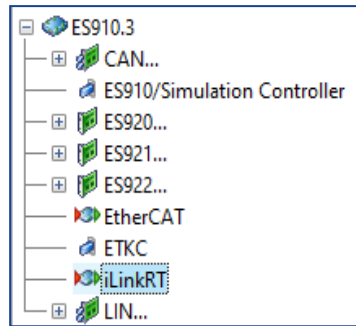
### Adding an ES910 to the INCA Hardware Configuration

The Fast ECU Access device ES910 has to be added and configured in the INCA Hardware Configuration.

To use an ES910 device in INCA

1. Create a new workspace in the **Database Manager**.
2. Click  to open the **Hardware Configuration Editor**.
3. Click  to add an ES910.
4. Select one or more ECU interfaces for the device (e.g. ETKC).

5. Select the MCE protocol (e.g. iLinkRT).



**Note**

You can only select one MCE interface per ES910 (iLinkRT or EtherCAT).

6. Click **OK**.

### 5.3.1 Configuring ES910 Devices for iLinkRT

The iLinkRT interface **MCE via iLinkRT** must be configured.

#### To configure an ES910 device for iLinkRT

1. Select the ES910 interface **iLinkRT** in the INCA Hardware Configuration.
2. Select the **iLinkRT Parameters** tab.
3. Set the following parameters in the **Server Communication Parameters** section:
  - **IP Address:**  
Enter the IP address under which the iLinkRT server is available for the client.
  - **Subnet Mask:**  
Enter the subnet for iLinkRT communication.
  - **UDP Port:**  
Enter the UDP port of the iLinkRT server.
4. Set the following parameters in the **Data Acquisition Communication Parameters** section:
  - **DAQ Communication Mode:**  
Specify whether data transfer should take place according to Multicast or Unicast method.

**Note**

If several clients are logged on at the server, it is more efficient to use the multicast method.

**Note**

**DAQ Multicast IP Address** and **DAQ UDP Port** only have to be set if Multicast was defined as **DAQ Communication Mode**.

- **DAQ Multicast IP Address:**  
Enter the multicast IP address here.

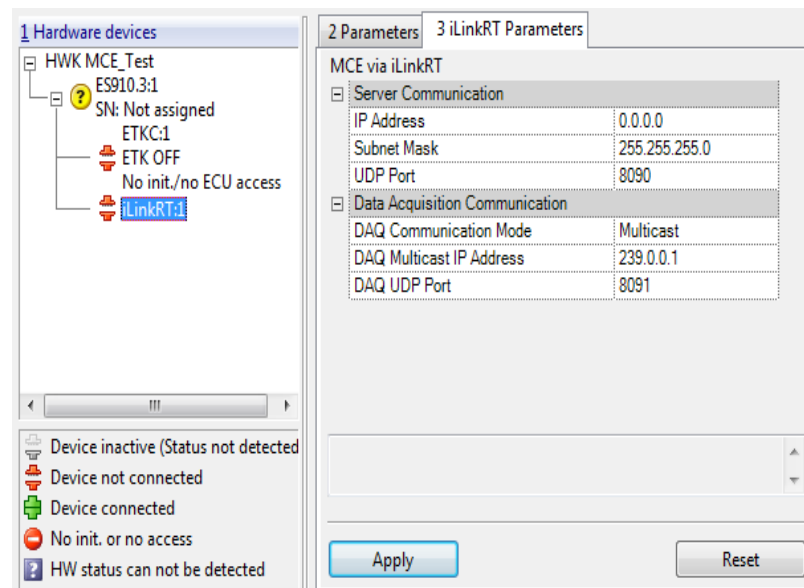
**Note**

The permissible range for multicast IP addresses is 239.0.0.1 – 239.063.255.255.

- **DAQ UDP Port:**  
Enter the multicast UDP port here.

5. Click **Apply**.

The settings are checked for validity and possible conflicts with other connected devices.



## Default Settings for the iLinkRT Configuration

It is possible to save default values for the iLinkRT configuration in an XML file. If the XML file

`iLinkRTServerCommunicationDefaultSettings.xml`

is stored under `.. \ETASData\INCA7.5\mce`, the iLinkRT configuration data (IP address, etc.) is used for a new iLinkRT device from this file.

A sample file is also installed (with the correct name and at the correct location).

## Hardware Initialization

Please note the following during initialization of the hardware in the hardware configuration (**Hardware > Initialize hardware** or F3):

The iLinkRT interface is displayed as "connected" (= green) when an MCE device is connected and operational that means powered on and booted. This does not, however, necessarily mean that, for example, an iLinkRT master is connected.

The EtherCAT interface is displayed as "connected" (= green) when an ES910 device is connected, is operational and the web interface settings for the I/O Port are set to EtherCAT. This does not, however, necessarily mean that, for example, an EtherCAT master is connected.

### 5.3.2 Configuring ES910 Devices for EtherCAT

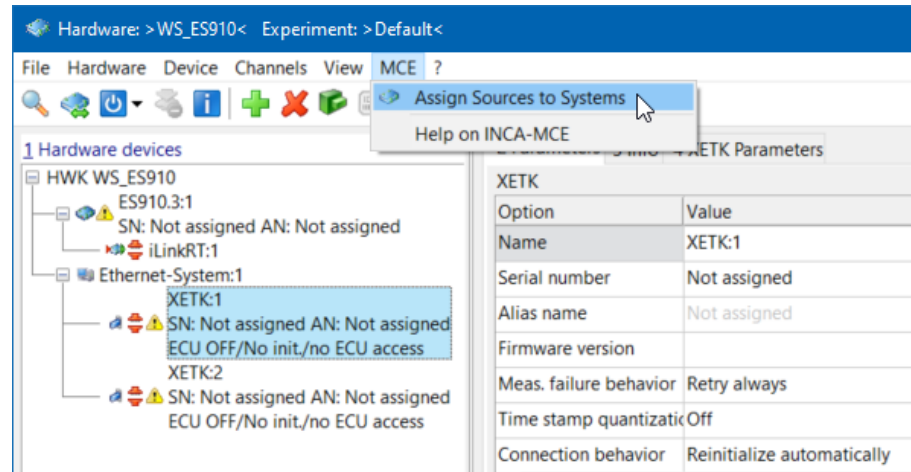
It is not necessary to configure the EtherCAT interface **MCE via EtherCAT**. You can only assign a name.

### 5.3.3 Assigning XETKs to the ES910 Interface

When XETK ECUs are used, they need to be assigned to an ES910 system (MCE-Link). Once the INCA-MCE device has been configured and the XETK has been selected and configured, proceed as follows:

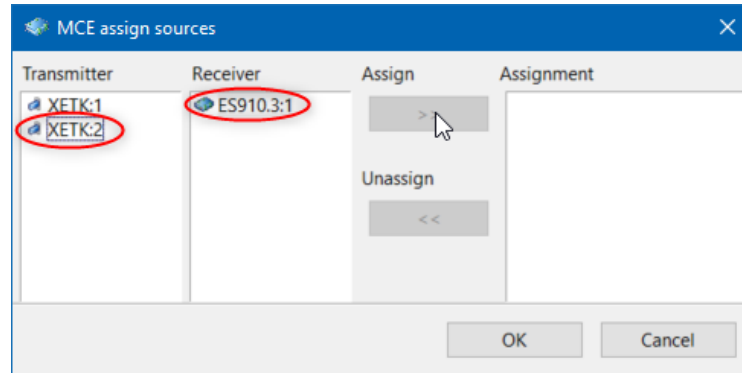
To assign XETK to the ES910 device

1. Select **MCE > Assign sources to system**.

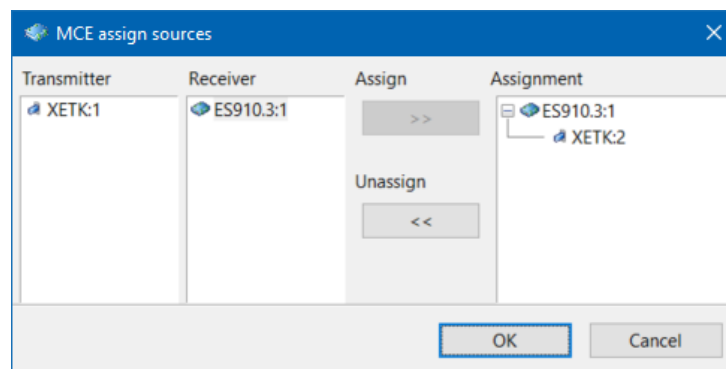


The "MCE assign sources" window opens.

2. Mark the XETK:2 and the ES910.3 system you want to assign.



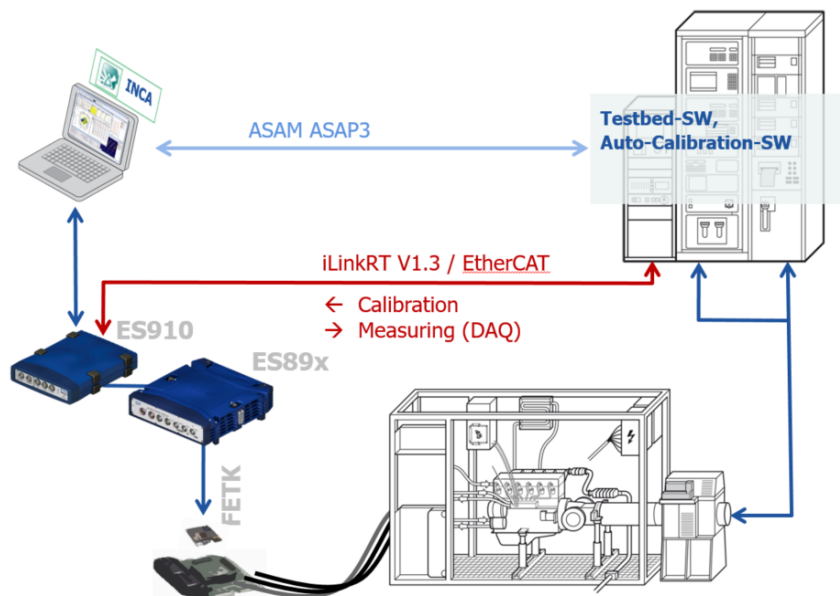
3. Click **Assign**.  
The XETK:2 is now assigned to the ES910.3:1 system.



4. Press **OK** to accept the assignment.  
The values of the XETK can now be used in INCA-MCE.

### 5.3.4 Assigning FETKs to the ES910 Interface

To connect a FETK via the ES910, you need an ES89x device which has the corresponding interface.



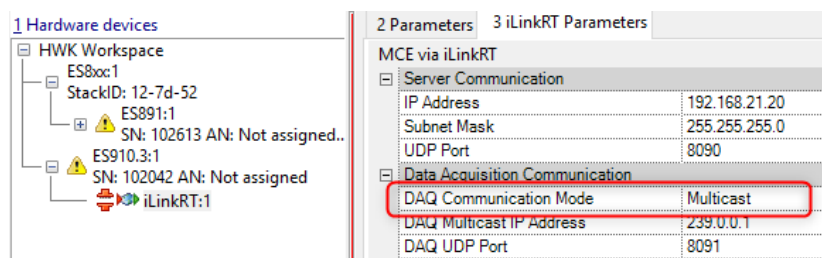
Once the ES910 device has been configured and the FETK has been selected and configured, proceed as follows:

#### To assign the FETK to the ES910 used as MCE-Link

1. Select **MCE > Assign sources to system**.

A dialog box opens where you can assign an FETK device that you want to use as ECU interface to the MCE output device. This makes the FETK measurement and calibration variables available for MCE. For more information, see "[Assigning XETKs to the ES910 Interface](#)" on page 36.

If you use an FETK, data communication has to be executed via multicast. In the **Hardware Configuration Editor (HWC)**, set the **DAQ Communication Mode** to **Multicast**.



### 5.3.5 Supporting XCP on CAN FD via ES922

The ES922 CAN FD Module is an extension module for the ES910.3 Rapid Prototyping Module. With two ES922 independent CAN interfaces you are able to connect CAN FD, high-speed CAN or low-speed CAN.

ES910.3 Rapid Prototyping Module sends data via iLinkRT or EtherCAT.

When XCP ECUs are used, they need to be assigned to an ES910 system (MCE-Link). Once the INCA-MCE interface has been configured and the CAN system has been selected and configured, assign a XCP to the MCE interface, see "[Assigning XETKs to the ES910 Interface](#)" on page 36

### 5.4 INCA Protocol Configuration for ES910

If you use Fast ECU Access with ES910 the variable selection is done via ASAP3. The data transfer is done via iLinkRT V1.3. You first have to select the related interface in the INCA User Options.

#### To start the ASAP3/iLinkRT V2 interface:

1. In the INCA Database Manager, click > General tab > ASAM-MCD 3MCI-interface version > ASAP3/iLinkRT V2.0.
2. Start an Experiment in the INCA Experiment Environment.

### 5.5 Multicast Network Configuration

The ETAS Multicast Address Manager assigns the multicast addresses used by the ETAS system. Any overlapping with other software could cause disturbances in the system. In this kind of conflict, the multicast addresses assigned by the

ETAS Address Manager can be limited to avoid overlapping with other systems. In principle, the ETAS system can use multicast addresses in the range 239.64.0.1 - 239.127.255.255 (INCA default).

Further limitation can be configured in the

... \ETASData\ResourceManager\MulticastConfig.cfg file.

The file is structured as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<MulticastConfig>
  <!--<PortNumber>18007</PortNumber>-->
  <!--<ExcludeAddresses>
    <Singles>
      <Address>239.64.0.1</Address>
      <Address>239.64.0.3</Address>
    </Singles>
    <Ranges>
      <AddressRange>
        <LowerBound>239.127.0.1</LowerBound>
        <UpperBound>239.127.0.255</UpperBound>
      </AddressRange>
      <AddressRange>
        <LowerBound>239.127.250.1</LowerBound>
        <UpperBound>239.127.255.255</UpperBound>
      </AddressRange>
    </Ranges>
  </ExcludeAddresses>-->
</MulticastConfig>
```



### Note

The comment characters "`<!-- TEXT -->`" disable the area between them.

The XML tags have the following meaning:

- `<PORTNUMBER>`  
This is where a different port can be specified (INCA default is port 18007).
- `<EXCLUDEADDRESSES>`  
This is where individual multicast addresses can be excluded for INCA.

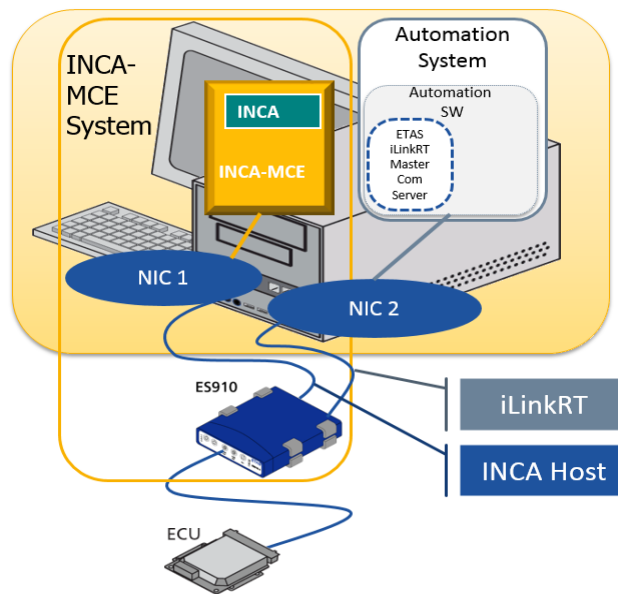
One or more addresses can be defined within the `<Singles>` tag.

Entire ranges can also be excluded within the `<Ranges>` tag.

Several ranges can be defined. The range is always defined using the first and last address of the range to be excluded.

## 5.6 Setup of INCA and iLinkRT System on same the Computer

It is possible to use iLinkRT COM Master and INCA/INCA-MCE on the same computer. For this setup it is important that two separated Network Interface Connectors (NIC) for INCA Host and iLinkRT must be used on computer side. They have to be configured in a way that the SubNets are separated for INCA Host and iLinkRT.



**Fig. 5-2:** Setup of INCA and iLinkRT system on the same computer

Example Configuration:

SubNet	Interface	IP Adress	SubNetMask
ETAS Hardware (192.168.10.*)	INCA HOST PC NIC	192.168.10.1	255.255.255.0
iLinkRT (192.168.20.*)	iLinkRT NIC	192.168.20.1	255.255.255.0
	INCA Hardware Con- figuration ES910.3 --> iLinkRT	192.168.20.2	255.255.255.0



## 6 System Configuration for MCE-PC

This chapter contains detailed information on how to configure INCA-MCE for Direct ECU Access.

The main steps are:

- "Connecting the MCE-PC" below
- "Adding a MCE PC to the INCA Hardware Configuration" on page 44
- "Configuring the MCE-PC Device" on page 44
- "Configuring the MCE-PC iLinkRT Interface" on page 48
- "Assigning XCPonETH, EXTK and FETK Devices to the iLinkRT V3 Interface" on page 49
- "Configuring iLinkRT V3 Protocol for MCE-PC" on page 51
- Connecting the Network



### Note

INCA supports for one INCA Hardware Configuration one MCE-PC.

### 6.1 Connecting the MCE-PC

#### 6.1.1 MCE-PC Hardware Interfaces

To connect to the MCE-PC you will need to establish connections to the following interfaces located at the rear and front sides of the computer.



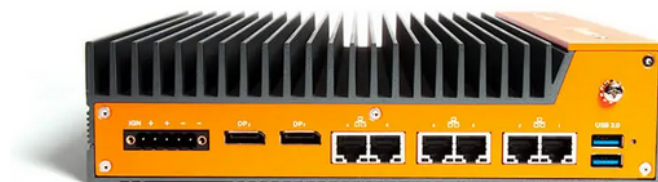
For safe connection and operation of the MCE-PC, read the [OnLogic product manual](#) and the OnLogic safety advice.



### Note

For detailed technical information, see the [OnLogic product manual](#).

#### *Rear side connection*



**Fig. 6-1:** Rear side of the MCE-PC

#### **Power**

The MCE-PC is equipped with a 220V AC power supply, but it can also be powered by 12V DC.

### Network ports

There are six network ports labeled from 1 to 6. Port 1 is designated for the ETAS network (NIC1), while port 2 is for the iLinkRT and the test bench network (NIC2). Ports 3 to 6 are currently not supported.

If a single switch is used for both the INCA/DEVICE/blue box network and iLinkRT and the test bench network, both NIC1 and NIC2 need to be connected to the switch.

### Front side connection



Fig. 6-2: Front side of the MCE-PC

### LEDs

The MCE-PC has several LEDs that are not specifically configured for the INCA-MCE/iLinkRT use case. These LEDs indicate regular computer states such as hard drive, network, and power indications.

### Peripherals

The MCE-PC operates headless without a monitor, keyboard, or mouse. Only for support reasons they may be required.

### Fan

To reduce the risk of physical and electrical hazards, enable fan protection.

### Power Button

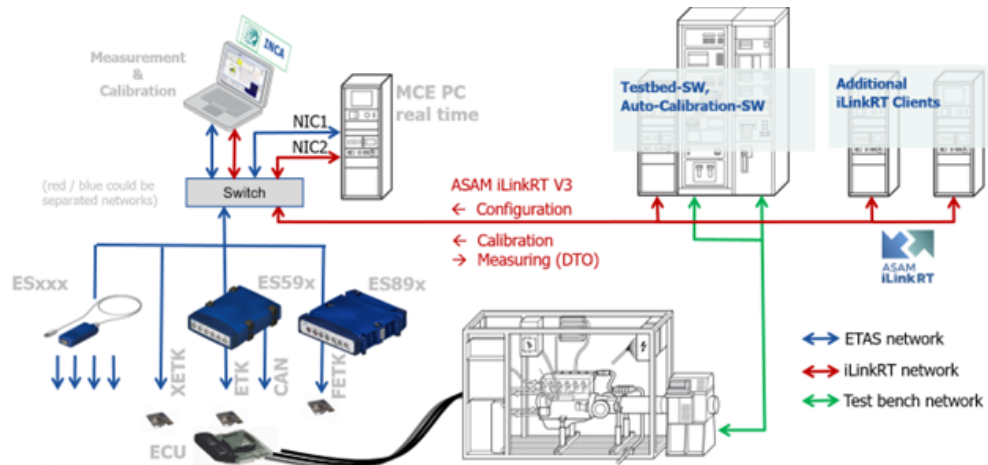
The MCE-PC starts automatically when you turn on the power or can be started manually by shortly pressing the power button.

It can be gracefully shutdown by shortly pressing the power button or forcefully powered off by a longer press of the power button. Additionally, the MCE-PC can be rebooted and shutdown using the ETAS web interface, see "[Configuring the MCE-PC Device](#)" on page 44.

## 6.1.2 MCE-PC Network Connection

In general, there are three networks in operation: the ETAS network, the iLinkRT network, and the test bench network. The following sections describe the setup of the physical and logical network connections in detail.

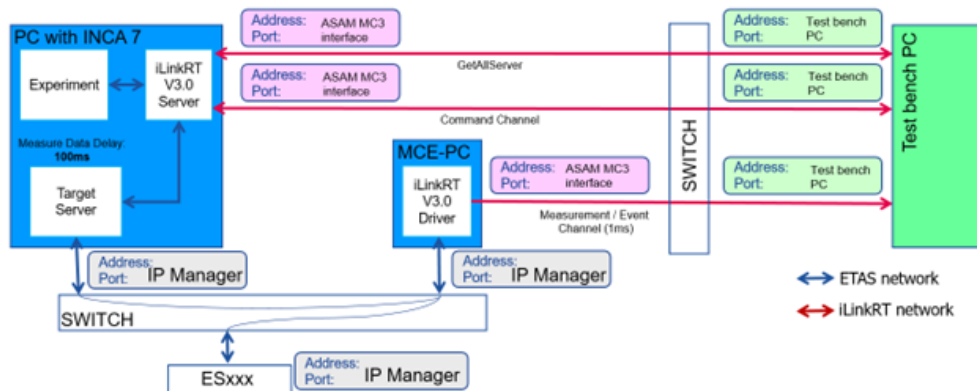
*Physical network connections*



- **ETAS network** (blue lines, NIC1)  
This network connects all measurement and calibration devices with INCA and the MCE-PC.
- **iLinkRT network** (red lines, NIC2)  
This network connects the test bench computer with the MCE-PC. It transfers the iLinkRT configuration, the calibration data, and the measurement data.
- **Test bench network** (green lines)  
This network connects the test bench computer with the test bench. In this document this network is not described in detail.

All physical connections are routed through a managed switch, which handles the logical connections.

*Logical network connections*



**Fig. 6-3:** Schematic visualisation of iLinkRT V3.0 - IP addressing

The configuration of the logical network connections is done on different places:

– **ETAS network**

The ETAS network (blue lines) is typically automatically configured by the INCA Network Manager. However, for the MCE-PC manual configuration is also possible, see "Configuring the MCE-PC Device" below.

– **iLinkRT network**



The iLinkRT configuration of the INCA side is done in the ASAM-MCD 3MC interface of INCA "INCA Protocol Configuration for Standard ECU Access" on page 23.

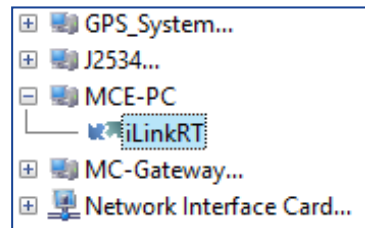
The iLinkRT configuration on the test bench side is done by the test bench.

## 6.2 Adding a MCE PC to the INCA Hardware Configuration

If you want to use a MCE-PC with INCA, you need to represent and configure the INCA-MCE in the INCA Hardware Configuration.

### To use an MCE-PC device in INCA

1. In the **Database Manager**, create a new workspace .
2. To open the **Hardware Configuration Editor**, click  .
3. To add a MCE-PC device, click  .
4. Select the iLinkRT interface.



 **Note**

You can only select one iLinkRT interface per MCE-PC.

5. Click **OK**.

### 6.2.1 Configuring the MCE-PC Device

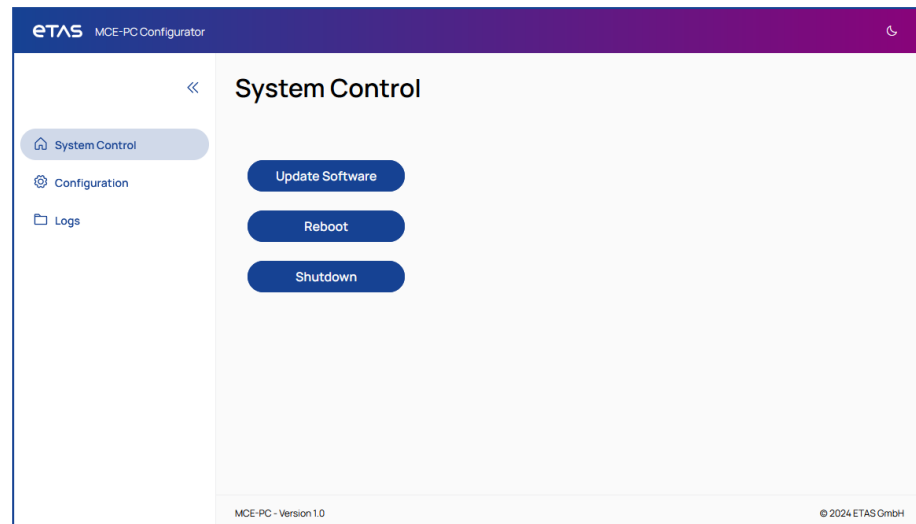
You can to configure the MCE-PC in the INCA Hardware Configuration Editor and in the ETAS Hardware Web Server.

To configure an MCE-PC device in the INCA Hardwareconfiguration Editor

1. in the **INCA Hardware Configuration Editor**, select the MCE-PC device.
2. Select the **MCE-PC Parameters** tab.
3. You can set the following parameters in the **Server Communication Parameters** section:
  - **Name:**  
INCA sets as default "MCE-PC:1" as name.
  - **Serial number:**  
INCA detects it automatically when the MCE-PC is connected.
  - **Alias name:**  
INCA detects it automatically when the MCE-PC is connected.  
To set the Alias name use the Hardware Web Server of the MCE-PC.
  - **Firmware version:**  
INCA detects it automatically when the MCE-PC is connected and a firmware check is executed.
  - **IP address:**  
INCA searches the MCE-PC via the IP manager and assigns an IP address. Can be used to access web interface.  
A double click on the IP address starts the WEB interface of the MCE-PC.
  - **INCA Hardware communication:**  
INCA uses fix Ethernet.
4. Click **Apply**.

To configure additional MCE-PC settings in the MCE-PC Hardware Web Server

1. In the **INCA Hardware Configuration Editor**, select the MCE-PC in the **Hardware devices** list.
2. On the **Server Communication Parameters** tab, double-click on the **IP address**.  
or  
Click **Configure**  
The MCE-PC Hardware Web Server opens.



3. To control the MCE-PC, click **System Control**.

You can perform the following actions:

- **Update Software**

Performs a MCE-PC software update. Select a SWU file (\*.swu) and click **Confirm Update**.

**Note**

Alternatively, the MCE-PC software update can be done using the HSP software. Using the newest HSP version ensures that the MCE-PC get always the newest software.

- **Reboot.**

Reboots the MCE-PC.

- **Shutdown.**

Shuts down the MCE-PC.

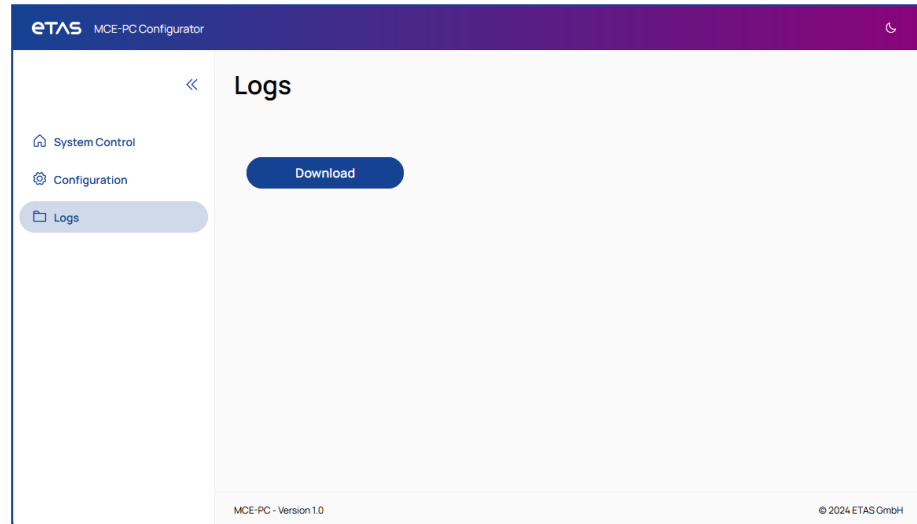
4. To configure the MCE-PC, click **Configuration**.

The screenshot shows the 'Configuration' page in the ETAS MCE-PC Configurator. It features a sidebar with 'System Control', 'Configuration', and 'Logs'. The main content area is divided into two sections: 'iLink IP configuration' and 'Device IP configuration'. The 'iLink IP configuration' section includes a 'Static IP (IPv4)' field with the value '192.168.1.23', a 'Prefix' dropdown menu set to '/24', and a 'Default Gateway' field with the value '192.168.1.2'. The 'Device IP configuration' section includes a 'Static IP (IPv4)' field with the value '192.168.40.19', a 'Prefix' dropdown menu set to '/24', and an 'Alias name' field with the value 'MCE-PC'. At the bottom of the configuration area, there are 'Revert' and 'Apply' buttons.

You can set the following parameters:

- **iLink IP configuration** field
  - **Static IP (IPv4)**  
IP address of the network card that is used for the iLinkRT connection.  
Port is set by iLinkRT configuration, see ["INCA Protocol Configuration for Standard ECU Access"](#) on page 23.
  - **Prefix**  
Enter the sub-net mask.
  - **Default Gateway**  
Needed when an iLinkRT client is not in the same sub-network
- **Device IP configuration** field
  - **Static IP (IPv4)**  
IP address of the network card that is used for the INCA connection.  
In contrast to other ETAS devices the IP address is retained over a restart.  
If the current IP address is unknown and therefore the Web UI unreachable, the INCA IP manager must be used to force the reassignment of a new IP address that matches the address of the INCA PC.
  - **Note**  
A manually set address may be overwritten by the INCA IP manager
  - **Port**  
The port is fixed.
  - **Prefix**  
Enter the sub-net mask.

- **Alias name**  
Name of the MCE-PC in the INCA Hardware Configuration Editor.
  - **Apply**  
The new settings will be applied and the browser sends them.
  - **Revert**  
All settings will be reverted to the currently active settings set by last Apply.
5. To download the MCE-PC LOG files, select **Logs**.



- **Download**  
Downloads zipped LOG files.



### Note

The MCE-PC starts a new log file whenever a new process is started. The old log is zipped and moved to log.archived, the former log.archived deleted.

## 6.2.2 Configuring the MCE-PC iLinkRT Interface

The MCE-PC device supports an interface for iLinkRT V3.

### To configure the MCE-PC interface for iLinkRT

1. In the INCA Hardware Configuration, select the MCE-PC interface iLinkRT .
2. Select the iLinkRT **Parameters** tab.
3. Set the following parameters on the **Server Communication** file:
  - **Name:**  
INCA sets as default "iLinkRT:1" as name.
  - **Measure Failure Behavior:**  
Select the expected behavior.

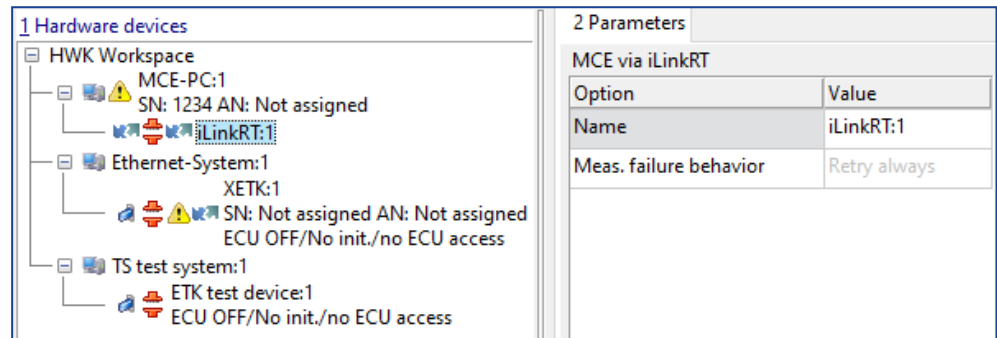


 **Note**

Currently there is only **Retry always** possible.

4. Click **Apply**.

The settings are checked for validity and possible conflicts with other connected devices.



## Hardware Initialization

Note the following during initialization of the hardware in the Hardware Configuration Editor: (**Hardware>Initialize hardware** or F3):

The iLinkRT interface is displayed as "connected" (= green) when an MCE-PC is connected and operational that means powered on and booted. This does not, however, necessarily mean that, for example, an iLinkRT connection is established.

## 6.3 Assigning XCPonETH, EXTk and FETk Devices to the iLinkRT V3 Interface

For a higher measurement update rate of up to 1ms the devices of the Hardware Configuration must be directly assigned to the MCE-PC.

For direct access INCA-MCE supports the following device types:

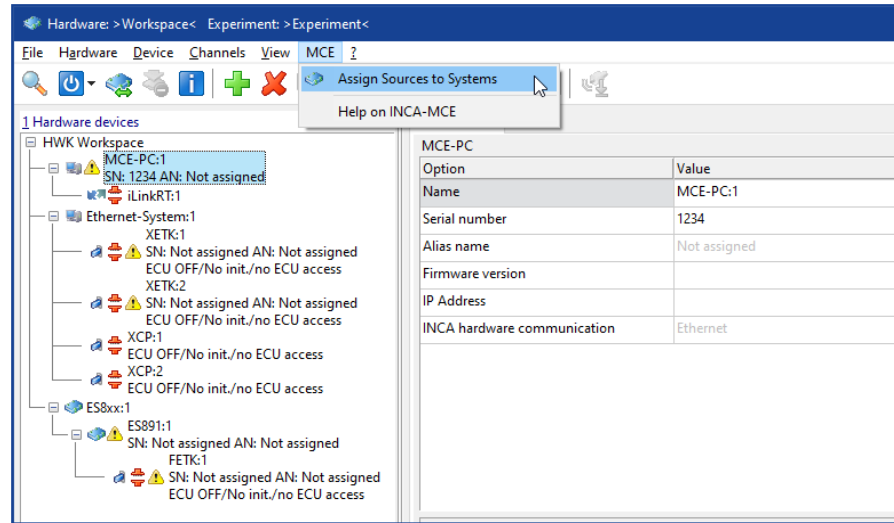
INCA-MCE supports the following device types for direct access:

- XCPonETH
- XETK
- FETK (ES89x1 needed)

When Direct ECU Access devices are used, they need to be assigned to the MCE-PC. Once the MCE-PC device has been configured, proceed as follows:

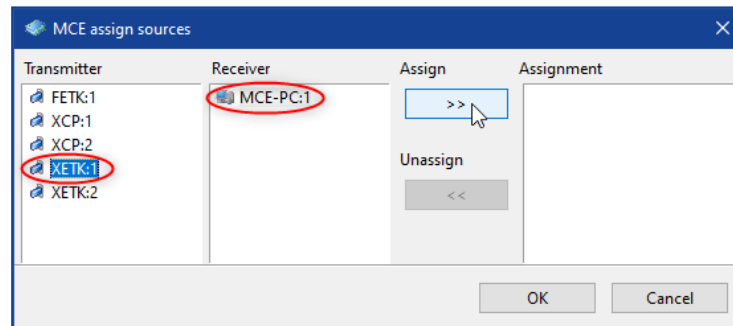
## To assign XCPonETH / XETK / FETK devices to the MCE-PC

1. Select **MCE>Assign sources to system**.



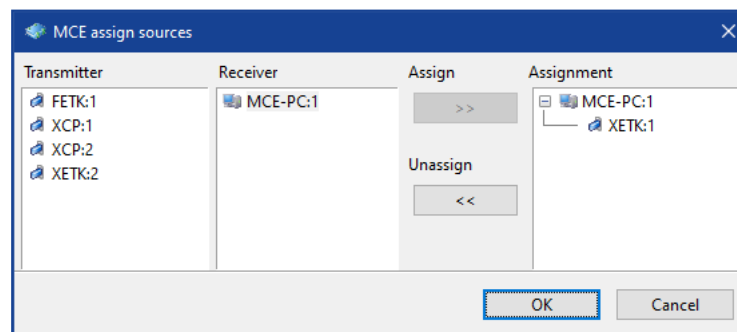
The "MCE assign sources" window opens.

2. Mark e.g. the XETK:1 and the MCE-PC you want to assign.



3. Click **Assign**.

The XETK:1 is now assigned to the MCE-PC system.



4. Repeat steps 2 and 3 to add more devices to the INCA-MCE.
5. Press **OK** to accept the assignment.

The XETK:1 and further assigned devices can now be used in INCA-MCE with Direct ECU Access.


**Note**

INCA supports one MCE-PC with up to four assigned devices.

## 6.4 Configuring iLinkRT V3 Protocol for MCE-PC

The iLinkRT V3 protocol settings are the same as for the iLinkRT V3 standard access. To use it for MCE-PC needs only an INCA user option setting

### To configure iLinkRT V3 for MCE-PC

1. Click  to open the INCA user options .
2. Select the **General** tab.
3. Select for ASAM-MCD 3MC interface version 'iLinkRT V3.0 MCE-PC'

Use encoding when reading the description file	<--->
ASAM-MCD 3MC interface version	iLinkRT V3.0 MCE-PC
MDA version	ASAP3 / iLinkRT V2.0
Web browser used by INCA	iLinkRT V3.0
Add-on activation	ASAM-MCD 3MC V1.0
	iLinkRT V3.0 MCE-PC

4. Click **Ok**.

Follow the steps described in "INCA Protocol Configuration for Standard ECU Access" on page 23.

## 7 Variable Selection for INCA Standard ECU Access

INCA-MCE offers the possibility to use the iLinkRT protocol without additional hardware. In this case the data transfer is handled by the ASAM-MCD 3MC interface of INCA.

For the iLinkRT protocol exist two possibilities

- Select single variables
- Select all variables of an INCA Experiment

### 7.1 Configuration via iLinkRT V3

The test bench selects via iLinkRT all variables that are needed. After selection the test bench can measure, read and write the variables.

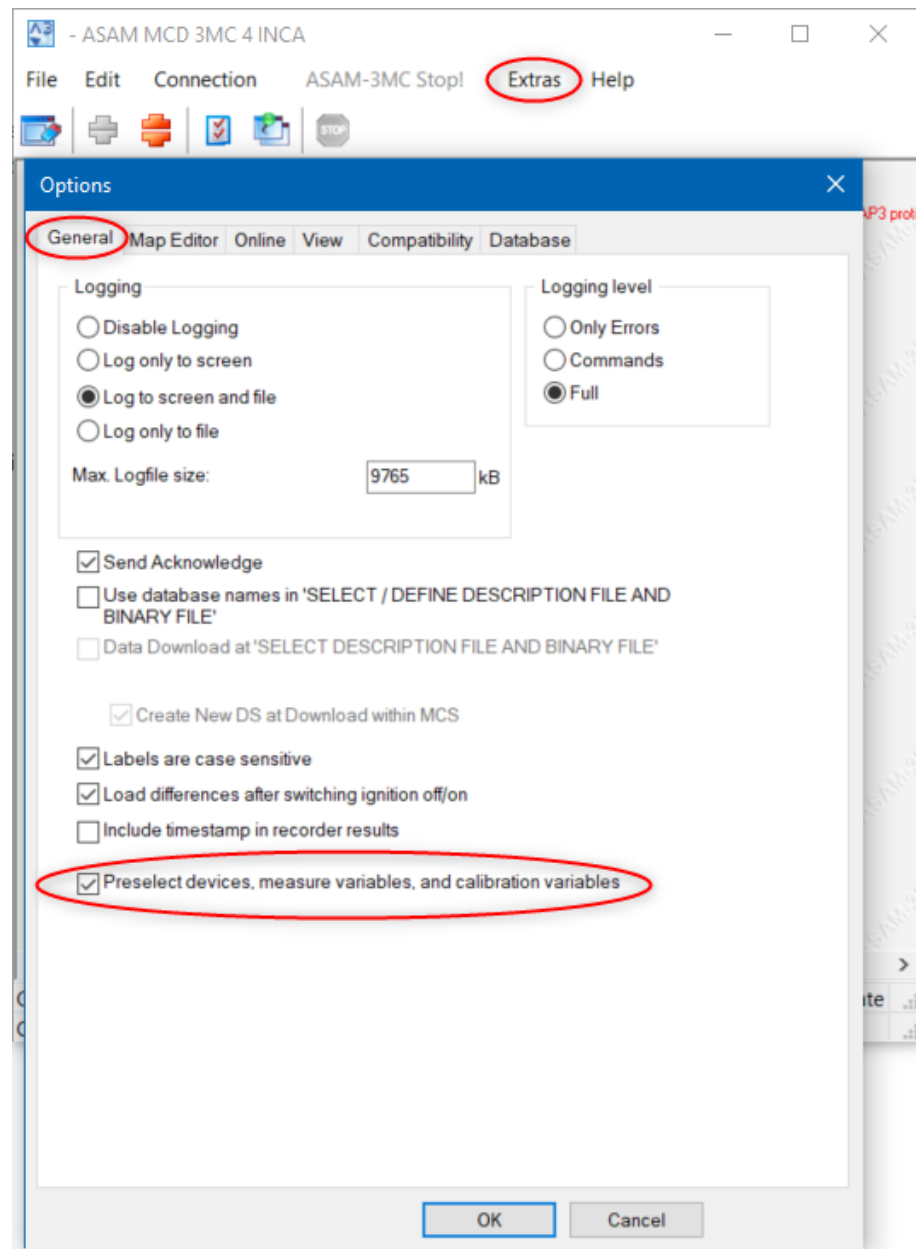
### 7.2 Configuration via INCA Experiment

iLinkRT V3 supports multiple clients. The INCA ASAM-MCD 3MC interface supports an additional iLinkRT V3 client. If this additional client is activated the ASAM-MCD 3MC interface adds all measurement variables of the experiment to it as preconfigured. The iLinkRT V3 standard defines that all measurement values of all clients are sent to all clients. By this the testbench client gets all measurement values configured in the INCA experiment.

To activate the preconfiguration in the INCAASAM-MCD 3MC interface

1. In the Experiment, select **Components > ASAM-3MC - Open ASAM-3MC Interface**.  
The "ASAM MCD 3MC 4 INCA" window opens.
2. Select **Extras > Options**.  
The "Options" window opens.
3. Select the **Compatibility** tab.
4. To use the preconfiguration, enable the option "Preselect devices, meas-

ure variables and calibration variables"



5. Click **OK**.

## 8 Variable Selection for ES910

### 8.1 Manual Variable Selection via INCA

#### Note

Before you can run MCE configuration, you first have to configure the system (MCE devices, ECUs), see "[System Configuration for ES910](#)" on page 29.

Before you can use variables in measurement and calibration access via the MCE interface, you must assign them in the Experiment Environment. You have the following options:

- Assigning complete sets of variables, without selecting them individually, see "[Assigning Complete Sets of Variables](#)" below.

This method is fast and efficient; however, it can happen that not all variables can be assigned, e.g. because the number of configurable measurement and calibration variables is limited with EtherCAT.

- Explicitly selecting individual variables, see "[Selecting Individual Variables via the MCE Variable Selection Window](#)" on the next page.

With this method, you have the possibility to select only those variables that you want to measure or calibrate via MCE.

#### 8.1.1 Assigning Complete Sets of Variables

You can assign either all variables that are part of the experiment, all measure variables, or all calibration variables:

1. In the Experiment Environment, open the **Variables** menu. Depending on the set of variables that you want to use for MCE, select one of the following menu entries:
  - **MCE - Add all variables of experiment**
  - **MCE - Add all calibration variables of experiment**
  - **MCE - Add all measure variables of experiment**

The corresponding set of variables is added without any further interaction.

#### Note

If it is not possible to assign the complete set of variables at once, a dialog is shown informing you which variables could not be added. You can then open the MCE Variable Selection window to adapt the variable assignment by removing and/or adding variables (see "[Selecting Individual Variables via the MCE Variable Selection Window](#)" on the next page).

## 8.1.2 Selecting Individual Variables via the MCE Variable Selection Window

The configuration window is opened in the Experiment Environment using **Variables > Select MCE variables**.

The **MCE Variable Selection** window is divided into the following areas:

- **Device**  
Displays all devices for which an MCE interface and an ECU interface supported by INCA-MCE were configured in the hardware configuration.
- **Measurement Variables** tab  
Displays the measurement variables selected for the MCE device.
- **Calibration Variables** tab  
Displays the calibration variables selected for the MCE device.
- **Remote Measurement Variables** tab  
Displays the remote measurement variables for the MCE device.
- **Remote Calibration Variables** tab  
Displays the remote calibration variables for the MCE device.



### Note

Variables that are configured twice (experiment & remote), are listed only once in the experiment columns.

To open the MCE configuration

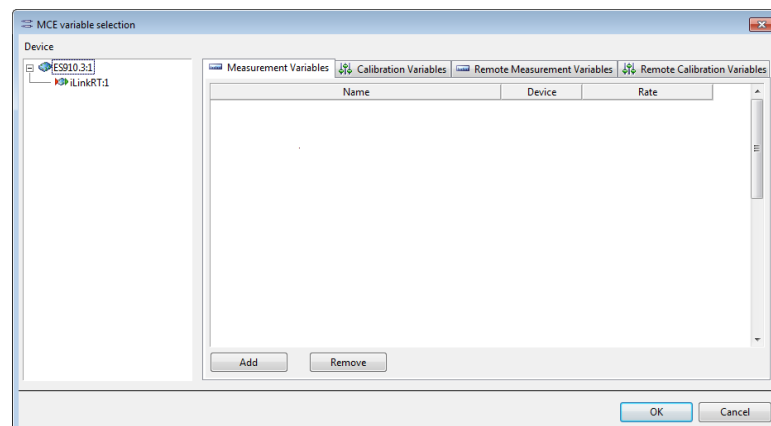


### Note

The MCE configuration cannot be changed during measurement. Stop measuring before you change the configuration.

1. In the Experiment Environment, select **Variables > MCE - Select variables for MCE**.

The "MCE Variable Selection" window opens.

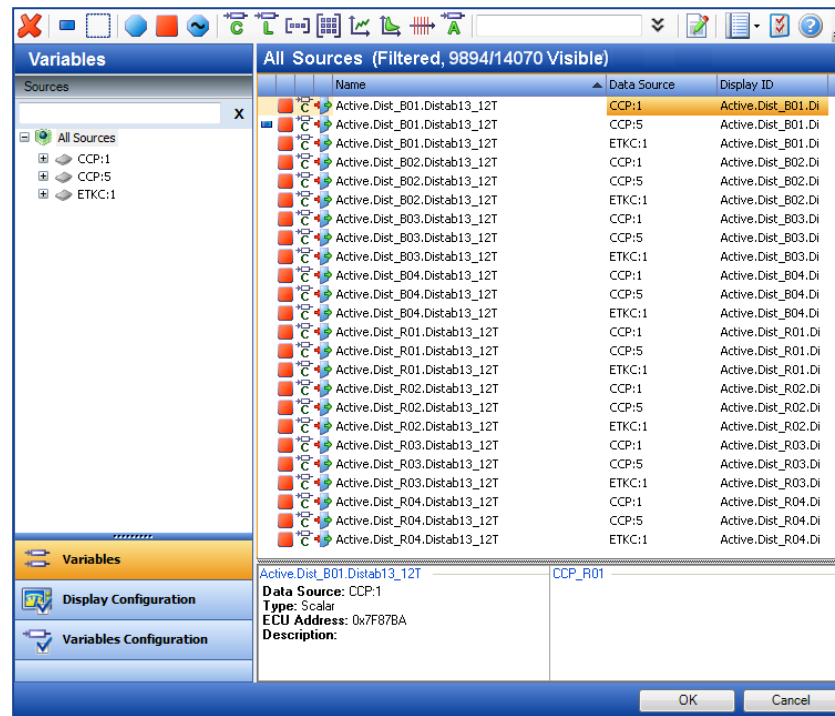


### To select measurement or calibration variables

You are now in the MCE Variable Selection window.

1. Under **Device**, select the MCE device to which you want to add variables.
2. Select the **Measurement Variables (Calibration Variables)** tab.
3. Click **Add**.

The variable selection window opens (see "Variable Selection in INCA-MCE" below).



The variable list displays the measurement variables and calibration variables available for the MCE device.

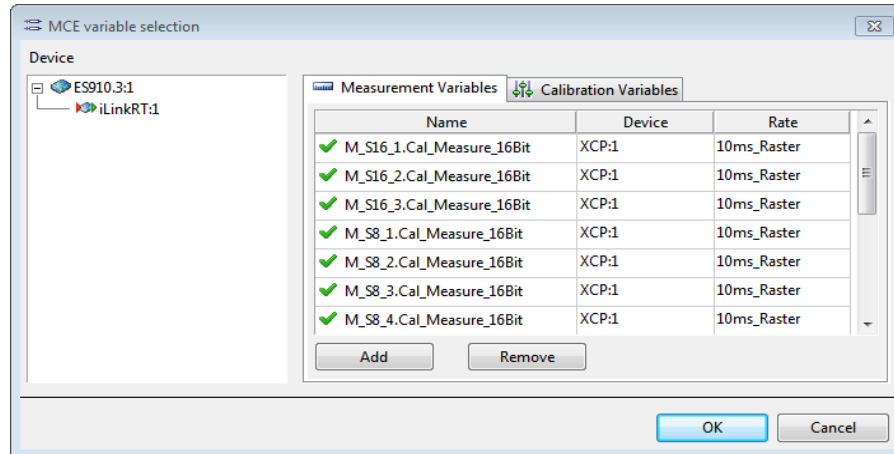
4. Highlight the measurement variables and calibration variables you want to add.
5. Click **OK**.

The selected measurement variables and calibration variables are added to the MCE configuration.

### Variable Selection in INCA-MCE

The variable selection window for INCA-MCE corresponds to the variable selection window of INCA.





The "MCE Variable Selection" window has the following features:

- Depending on whether you want to add a variable in the **Measurement Variables** or **Calibration Variables** tab in the **MCE Variable Selection** window, the variable list will show either measurement or calibration variables.
- Only the measurement and calibration variables available for the selected MCE device are displayed.
- A green checkmark marks all variables that have already been successfully registered for MCE use. Variables without a checkmark are marked for MCE use, but have not yet been registered.

They are finally registered if you click **OK** in this dialog. If it is not possible to use all of the selected variables for MCE, a corresponding message is shown.

- An additional column shows the raster used for the measure variable.

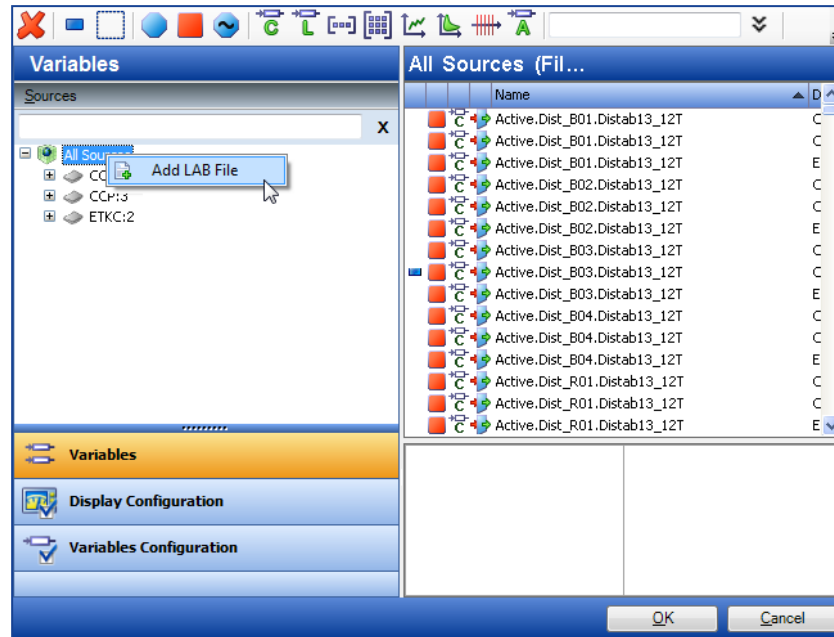
If you want to change the raster, close this window and open the Variable .

No raster can be selected and no raster filling level is displayed in the variable selection window for INCA-MCE.

#### To filter the variable list with a LAB file

You are in the variable selection window for INCA-MCE.

1. In the context menu of **All Sources** or one MCE device, select **Add LAB File**.



A dialog box for file selection opens.

2. Select the LAB file.  
The variable list only shows variables that are contained in the LAB file.



### Note

If you want to know more about the variable selection window, read the section **variable selection window** in the INCA online help.

#### To remove a variable from the MCE configuration

You are in the "MCE Variable Configuration" dialog box.

1. Highlight a variable you want to remove.
2. Click **Remove**.

The variable is removed from the MCE configuration.

#### To terminate the MCE configuration

You are in the "MCE Variable Configuration" dialog box.

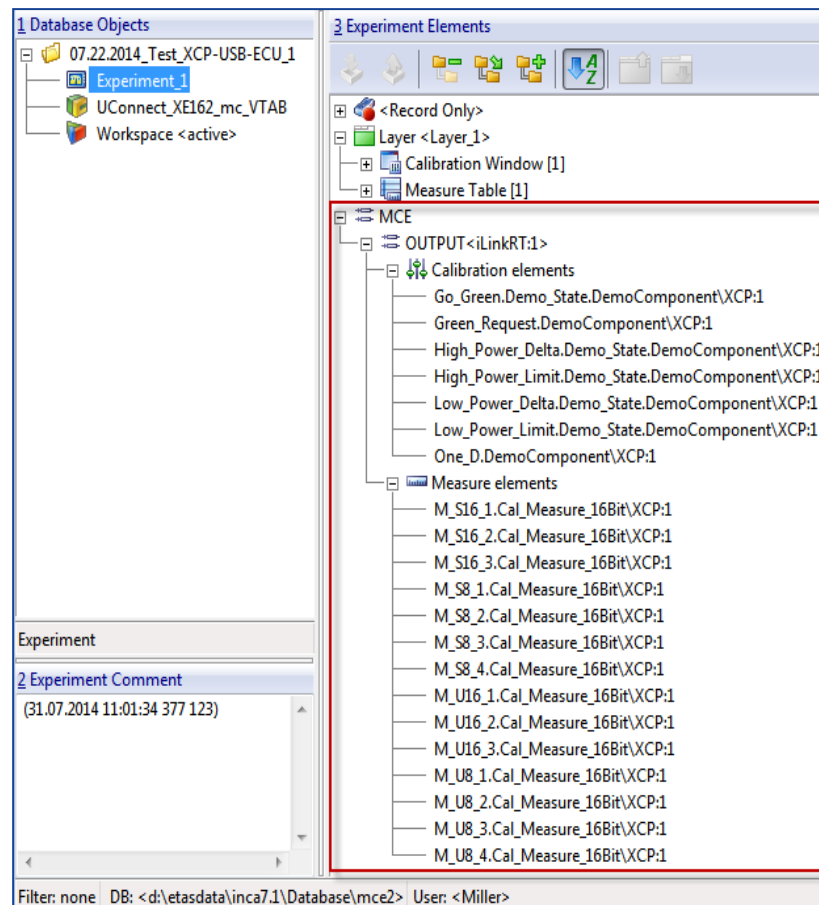
1. Click **OK** to accept the configuration.

The configuration is part of the Experiment Environment and is saved with it.

#### To show the MCE configuration in the INCA Database Manger

The INCA Database Manager provides an experiment view showing the structure and elements of an experiment without the need to open it.

1. Select the experiment that includes the MCE configuration.



## Variable and Raster Selection for INCA-MCE Variables

When selecting in the variable configuration of INCA-MCE, the variables are assigned to their default rasters from the A2L file, unless the variables have already been selected for the Experiment Environment – in this case, the raster that has already been selected is valid. If there are no default rasters in the A2L file, the variable is assigned to the slowest raster.

The raster used in the experiment and therefore also in MCE is also shown in the MCE Variable Selection window opened by selecting **Variables > MCE - Select variables for MCE**. However, it can only be changed in the INCA variable selection window **ALT-T**: The "MCE Signal" block is directly beside the "Experiment" – all measurement variables selected for MCE are displayed here. As with all standard INCA measurement variables, the raster can be changed. If the measurement variables are also registered in the Experiment Environment for display or recording, the change of raster also applies there. That means the variables can only be assigned to one raster.

 **Note**

If MCE variables are configured with an automation interface, these are displayed in the variable selection window ("Variable Configuration" tab) under the "MCE Signal" block. These variables are not displayed in the "MCE Variable Selection" – only variables that were selected here are actually shown here.

### Activating the Measurement or Calibration Configuration

If calibration variables were selected or rejected for an MCE interface, the MCE configuration for the calibrations is enabled with **OK** in the variable selection window, that means the calibration variables can be accessed immediately via the MCE interface. If a different configuration already exists and there are changes, the existing configuration is deleted and replaced by the new one.

The measurement configuration is saved in the Experiment Environment when the MCE variable selection window is closed and stored with the experiment. The measurement configuration is not available at the MCE interface until measuring starts, for example, with F1 or F12.

## 8.2 Variable Selection via Automation Interfaces

It is not only possible to select variables manually via INCA, but also via various automation interfaces. The section below describes which automation interfaces can be used to select variables INCA-MCE.

 **Note**

Before INCA-MCE can be used via an automation interface, a workspace has to be created with a valid hardware configuration and experiment.

### 8.2.1 Tool API

The INCA Tool API (COM interface) provides methods for configuring the ES910.

 **Note**

To find out more about the INCA Tool API, refer to the reference manual "INCA Tool API Documentation.chm" which is installed in the folder `<drive>:/ETAS/INCA_x/cebra/`.

### Verifying whether a Device Supports INCA-MCE

Use the

```
IsFULISupported
```

method to check whether a device in the hardware configuration supports the current INCA-MCE interface (iLinkRT, EtherCAT). The result of this request takes the form of a Boolean value.

## Registering ES910INCA-MCE Variables

Use the

```
EnableFULIAccess(Name, Device)
```

method to register a measurement or calibration variable at the FULL interface.

## Deregistering INCA-MCE Variables

Use the

```
DisableFULIAccess(Name, Device)
```

method to deregister a measurement or calibration variable at the FULL interface.

## Activating the INCA-MCE Configuration at the ES910

Use the

```
SwitchCalibrationAccessOn
```

method to switch on calibration access. This enables the INCA-MCE calibration configuration at the ES910 interface.

To read changes made to the calibration configuration into the ES910 interface, disable the calibration configuration using the

```
Switch CalibrationAccessOff
```

method and then activate it again using the method

```
SwitchCalibrationAccessOn
```



### Note

Before you can carry out any calibration tasks using ES910, the general conditions described under "[Prerequisites for Calibration Access](#)" on [page 70](#) must have been fulfilled.

Use the

```
StartMeasurement
```

method to start measuring. This enables the INCA-MCE configuration at the ES910.



### Note

If you are using measurement and calibration variables at the same time, read section "[Using Measurement Variables and Calibration Variables](#)" on [page 64](#).

## 8.2.2 ASAP3

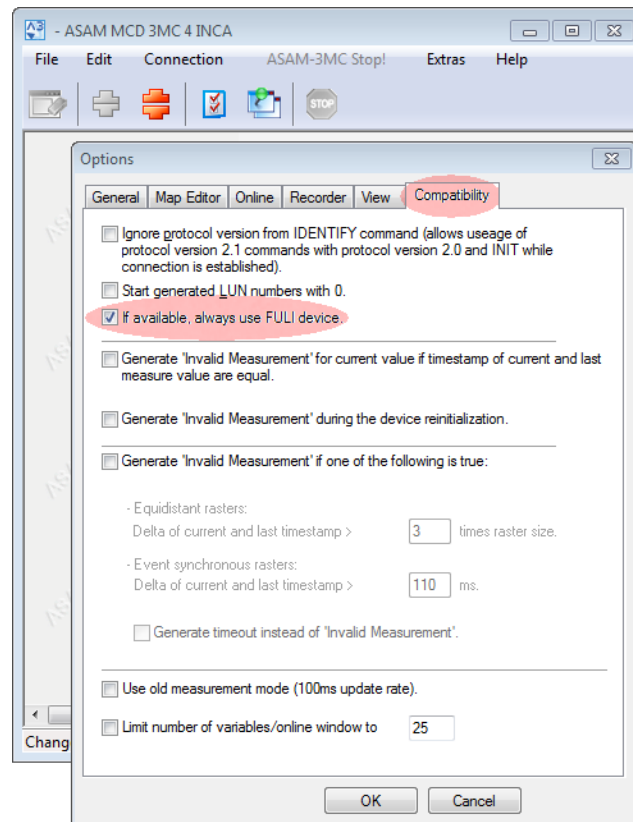
The ASAP3 interface can be operated with implicit or explicit INCA-MCE configuration.

## Implicit Configuration

All values registered with ASAP3 are automatically registered by INCA with INCA-MCE.

### To set the ASAP3 option

1. In the experiment, select **Components > ASAM-3MC - Open ASAM-3MC Interface**.  
The "ASAM MCD 3MC 4 INCA" window opens.
2. Select **Extras > Options**.  
The "Options" window opens.
3. Select the "Compatibility" tab.
4. To use the implicit configuration, enable the ASAP3 option "If available, always use FULI<sup>1)</sup> device".



## Explicit Configuration

To configure the MCE interface, what are referred to as FULI LUNs must be created for every device that is to transfer variables via INCA-MCE.

An LUN (Logical Unit Number) is the representation of a device (in the ASAP3 context). A FULI LUN is a special second representation of the device that exclusively represents the INCA-MCE configuration.

<sup>1)</sup> Fast Upper Level Interface; High performance interface between ETAS hardware and testbench

The LUN itself is created in ASAP3 using the commands

```
SELECT DESCRIPTION FILE AND BINARY FILE
```

or

```
DEFINE DESCRIPTION FILE AND BINARY FILE
```



### Note

To find out more about FULL LUNs, read the following chapters in the ASAP3 documentation `INCA_IF_ASAM-ASAP3_EN.pdf` (in the directory `<drive>:/ETAS/INCA_x/Manuals`):

- FULL-Fast Upper Level Interface (INCA-MCE)
- SELECT DESCRIPTION FILE AND BINARY FILE  
COMMAND:3
- DEFINE DESCRIPTION FILE AND BINARY FILE  
COMMAND:30

## Registering INCA-MCE Variables

Measurement variables are registered at the FULL LUN using the command

```
PARAMETER FOR VALUE ACQUISITION
```



### Note

Do not register measurement variables that are registered via the FULL LUN via the regular device LUN. Otherwise, the measurement variables will be transferred twice via `GET ONLINE VALUE`.

Calibration variables are registered at the FULL LUN using the command

```
GET PARAMETER FROM AP-S
```

## Activating the INCA-MCE Configuration at the INCA-MCE Device

The measurement configuration is activated at the start of measuring using the command

```
GET ONLINE VALUE
```

The calibration configuration is activated using the command

```
SWITCHING ONLINE
```



### Note

In an online state, if a new calibration configuration is added, the measurement configuration is disabled for a short time (see ["Using Measurement Variables and Calibration Variables"](#) on the next page).

For this reason, we recommend the following procedure when updating calibration configurations:

- Select all measurement variables with the command  
PARAMETER FOR VALUE ACQUISITION
- Select all calibration variables with the command  
GET PARAMETER FROM AP-S
- Activate the calibration configuration with the command  
SWITCHING ONLINE
- Activate the measurement configuration with the command  
GET ONLINE VALUE

### 8.3 Order to be Adhered to in the Overall System

Using INCA-MCE at an automated test bench is a challenging linking of many system components. To ensure the function of the ES910 device in the overall system, orders for configuration have been defined for the following applications.

#### 8.3.1 Using Measurement Variables and Calibration Variables

The ES910 treats the configuration of measurement variables and calibration variables separately to be able to represent the following use cases:

- Only measuring
- Only calibrating
- Calibrating and measuring

The ES910 can only be used when all subconfigurations have been completed.



#### Note

Please note that when configuring via GUI or automation interfaces, corresponding steps also have to be run before the ES910 interface can be accessed. For more information, refer to the sections "[Manual Variable Selection via INCA](#)" on page 54 and "[Variable Selection via Automation Interfaces](#)" on page 60.

#### 8.3.2 When Using the External Master Clock (EtherCAT Distributed Clocks [DC])

The EtherCAT DC signal is only transferred in the EtherCAT state `Safe-Operational` or `Operational`. Once a measurement has been started and the MCE device put into the state `Safe-Operational` or `Operational` by the master, the ES910 determines the central DC time once. The internal ES910 clock is then adjusted continuously ("Drift Correction").

In the case of reconfiguration, the ES910 is reset to `Pre-Operational` – the clocks are not synchronized during this time. This is why it is recommended that overall configuration is completed before measuring is started.



## 8.4 Variable Selection for iLinkRT at the Computer

If you completed the protocol configuration, you can now use INCA with iLinkRT at the computer.

The configuration of the measuring, recording, and calibration is done either via ASAP3/iLinkRT V2.0 or via iLinkRT V3.0.

### To use ASAP3/iLinkRT V2.0

1. Select the measure and calibration variables via ASAP3.
2. Start the measurement via ASAP3.

INCA sends the selected measure variables automatically via the iLinkRT measurement port.

INCA immediately executes iLinkRT calibration commands which are received via the iLinkRT command port.

### To use iLinkRT V3.0

1. Select devices via iLinkRT.
2. Select the measure and calibration variables via iLinkRT.
3. Start the measurement via iLinkRT.

iLinkRT sends the selected measure variables automatically via the measurement port.

INCA immediately executes iLinkRT calibration commands which are received via the iLinkRTT command port.

## 9 Variable Selection for MCE-PC

INCA-MCE offers the possibility to use a MCE-PC. In this case the measurement data transfer is handled by the MCE-PC, the configuration and calibration data transfer by the ASAM-MCD 3MC interface of INCA.

For the iLinkRT protocol exist two possibilities

- Select single variables
- Select all variables of an INCA experiment

The variable selection for MCE-PC is identical to the variable selection for INCA standard ECU access, see "[Variable Selection for INCA Standard ECU Access](#)" on [page 52](#)

## 10 Measuring and Calibrating with ES910

This chapter contains information and tips on measuring and calibrating with ES910.

Usually INCA uses one connection for every connected ECU – the INCA interfaces (GUI or automation interfaces) then access a set of ECU connections.

For reasons of data consistency and synchronization, the ES910 does not establish a second connection to the relevant ECU but acts as a co-user of the existing connection.



### Note

As INCA is used for configuring and controlling ECU communication, it is absolutely necessary for the ES910 to be connected to the INCA system and for INCA to be in operation.

### 10.1 Measuring with ES910 Fast ECU AccessDevice

INCA-MCE measure data transfer is linked to the measuring function of INCA – if measuring is started in INCA, The ES910 is configured and the measure data of the ES910 interface (iLinkRT, EtherCAT) is made available. When measuring is stopped, the ES910 interface has no more measure data at its disposal. The measurement configuration itself, however, is still available: When measuring is restarted in INCA without the measurement or calibration configuration being changed, measure data is once again available at the ES910 interface and can be sent.

Measuring can be started and/or stopped manually in INCA or via an automation interface.

#### 10.1.1 Special Features of ES910 iLinkRT V1.3/V2 interface

The measure data configuration for INCA-MCE in INCA defines the measure data that can be made available at the ES910 iLinkRT interface. The measure data is organized in what are referred to as DAQ lists. `START_STOP_DAQ_LIST` starts and stops DAQ list transfer at the iLinkRT interface.

At the iLinkRT interface, measure data transfer is driven by the data received from the ECU. The data received at the ES910 is transferred directly to the iLinkRT interface. That means that not all configured measure data has to be transferred all the time. This behavior means that the raster of a measurement variable selected in the measurement configuration has an immediate effect at the ES910 interface.

A change in the ES910 configuration means that the iLinkRT interface interrupts the connection and sends a `EV_SESSION_TERMINATED` to the iLinkRT client. Once configuration has been completed, the iLinkRT client re-establishes the connection (`CONNECT`) and then reads in the available measure data.

### 10.1.2 Special Features of ES910 EtherCAT interface

The measure data configuration for INCA-MCE in INCA defines the measure data that is transferred at the EtherCAT interface. The measurement variables configured in INCA-MCE are entered in the CAN-Open-Dictionary. When the ES910 is in the state `Operational` or `Safe-Operational`, all measurement variables entered in the CAN-Open-Dictionary are transferred.

At the EtherCAT interface, measure data transfer is driven by the bus frequency configured at the EtherCAT master.



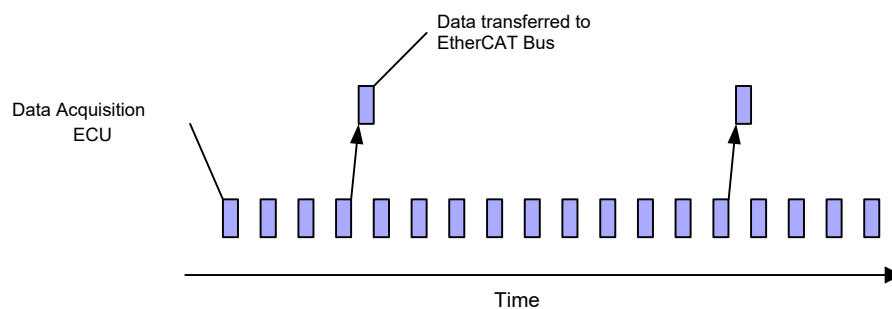
#### Note

Make sure that you select the EtherCAT bus clock so no unnecessary resources are used at the MCE device or that no unnecessary bandwidth of the EtherCAT bus is used.

For example, incoming data causes interrupts – if the clock at the EtherCAT bus is unnecessarily high, the interrupt load in the ES910.3 also increases unnecessarily and thus limits the computing power available for other tasks.

If, for example, you acquire data at the ECU with a raster of just 1 ms and transfer it to the EtherCAT interface at 100 ms, this results in unnecessary resources being used in the ES910. This is referred to as *undersampling* at the EtherCAT interface.

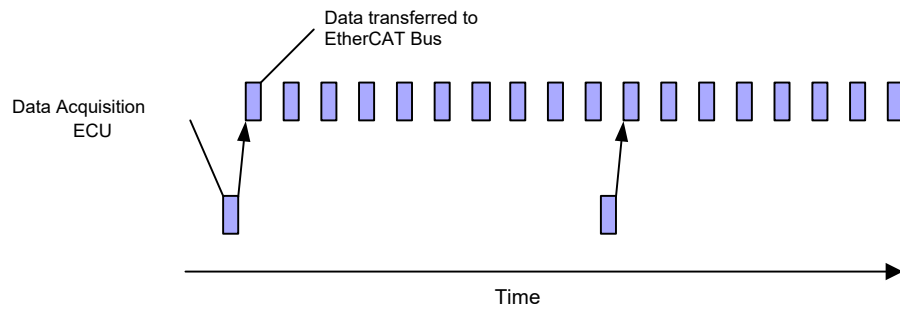
The following graphic shows a schematic of undersampling. You can see that the measure data at the ECU is requested and processed in INCA and the ES910 device ten times as often as it is transferred to the EtherCAT bus.



**Fig. 10-1:** Undersampling at the EtherCAT Interface

If, for example, you acquire data at the ECU with a raster of 100 ms and transfer it to the EtherCAT interface at just 1 ms, this results in unnecessary bandwidth of the EtherCAT bus being used. This is referred to as *oversampling* at the EtherCAT interface.

The following graphic shows a schematic of oversampling. You can see that the measure data is transferred ten times as fast at the EtherCAT bus as it is acquired by the ECU and processed in INCA and the ES910.



**Fig. 10-2:** Oversampling at the EtherCAT Interface

Generally speaking, a slight oversampling is what is being aimed for. This ensures that changes are available extremely dynamically at the INCA-MCE interface and that the automation system can react quickly.

This perspective, however, only takes the requirements made of measure data transfer of the EtherCAT bus into consideration. Optimum EtherCAT bus frequency and ECU data acquisition raster depend on other factors such as other EtherCAT bus sharing units, parallel recording of measure data in INCA, latency requirements with regard to calibration via the EtherCAT interface, etc..

A change to the INCA-MCE configuration results in the EtherCAT interface of the MCE device interrupting measure data transfer and changing to `Pre-Operational` state. This is why an automation system should disable the interface of an ES910 before a configuration is changed.

### 10.1.3 Measure Data Types not supported

INCA-MCE does not support measurement variables of the type "ASCII" or "virtual" and "dependent" measurement variables. The selection is already limited or rejected in the GUI or in the automation interfaces.

### 10.1.4 Behavior with Configuration Changes

The configuration of the measurement and calibration variables for iLinkRT and EtherCAT is static in operation. In other words, the position in the CAN-Open-Dictionary (EtherCAT) or in the order of the DAQ lists and calibration lists (iLinkRT) is fixed once MCE interface communication has started (`CONNECT` or `Operational` state). This is why it is recommended to configure the measurement and calibration variables before recording starts.

A change in the configuration can be triggered by an automation interface (ASAP3, MCD3 or Tool API) or in the INCA Experiment Environment. For example, a new calibration variable (`GET_PARAMETER_FROM_AP-S` for an ES910) is requested in this case in ASAP3.

The following happens in this case:

- The INCA-MCE interface goes offline (`Pre-Operational` with EtherCAT, `EV_SESSION_TERMINATED` with iLinkRT).
- The new configuration of the calibration variable is generated and downloaded to the ES910.

- The Object-Dictionary is recreated in the INCA-MCE device and made available to the INCA-MCE interface.
- The ES910 interface is available once again.
- The reconfiguration is acknowledged at the automation interface with the completion of the triggering command.

During this time, measure data transfer at the INCA-MCE interface is interrupted – calibration is no longer possible.

Measure data transfer can only be restarted when the connection has been re-established and the new configuration read in at the ES910 (read CAN-Open-Dictionary and change of state to `Operational` with EtherCAT or `CONNECT, READ_DAQ_EXTENDED` and `START_STOP_DAQ_LIST` with iLinkRT).

### Multi-Raster

The measurement variables to be selected for INCA-MCE can only be assigned to one raster.

## 10.2 Calibrating with ES910 Fast ECU AccessDevice

The following section describes the general conditions for calibrating with ES910.

### 10.2.1 Prerequisites for Calibration Access

The following general conditions must be fulfilled to be able to access calibration variables at the ES910 interface:

- The ECU can be accessed.
- The content of the INCA dataset and the ECU match (checksum verification was successful).
- The working page in the Experiment Environment is active.
- Calibration access is activated.

If all these general conditions are fulfilled, the calibration variables of the ECU can be changed via the ES910 interface.

If the general conditions are not fulfilled, the ES910 interface returns an error, for example, as a response to `CAL_DOWNLOAD_ADVANCED` with iLinkRT.

If the hard limits of a calibration variable are violated, the variable is accepted into the ECU according to its hard limit. The soft limits are not monitored because it is pointless to assume a sensible reaction (real time) of the user in automated operation.

### 10.2.2 General Information

#### Supported Data Types

The ES910 supports 1-, 2- and 4-byte "signed integers", "unsigned integers" and 4-byte "floating point numbers" (IEEE format).

## Supported Calibration Types (ASAM-MCD-2 MC)

The ES910 supports the following ASAM-MCD-2 MC types:

- VALUE (scalar)
- CURVE (1D array with physical axes)
- MAP (2D array with physical axes)
- VAL\_BLK (1D or 2D arrays without physical axes)

## Calibration Data Types Not Supported

The following data types are not supported and cannot be configured for ES910 :

- VIRTUAL\_CHARACTERISTIC
- DEPENDENT\_PARAMETER
- ADAPTIVE\_CHARACTERISTIC

## Grouped ECUs

Grouped ECUs are not supported. Every ECU is seen as an entity. To set a calibration variable on several ECUs, each one has to be calibrated with iLinkRT or EtherCAT.

## Special Calibration Methods

When using Limited EMU RAM and InCircuit 2 with curves and maps, you have to ensure manually that the corresponding axes are emulated (even if the axes are not to be calibrated).

### 10.2.3 Transfer of INCA-MCE Calibration Actions to INCA (Back Animation)

Calibrations via ES910 are executed fast directly in the ECU – as the frequency of the calibrations can be very high, there is no point in displaying every calibration action in the user interface.

To measure up to all requirements, an intelligent back animation of the calibration parameters was implemented. Basically, every change at the ES910 interface is reported back to INCA. With iLinkRT, every calibration procedure is a change – with EtherCAT, only a change of the calibration variable in comparison to the previous one is seen as a change.

To keep the computer load as low as possible, change frequencies that are not visible are automatically reduced. As all calibrations are reported back to INCA, the data in the dataset match the data in the ECU at the end of the session.

**i Note**

To ensure data consistency, the ES910 must report every calibration action to the INCA computer – the transfer takes place via the Ethernet host connection between the ES910 and INCA-computer.

In the case of very fast and continuous calibrations (particularly over the high-performance ETK interface), there can be a backlog when it comes to slow or poorly connected computer network cards, that means the ES910 device reports calibration actions faster than the network card can receive the messages.

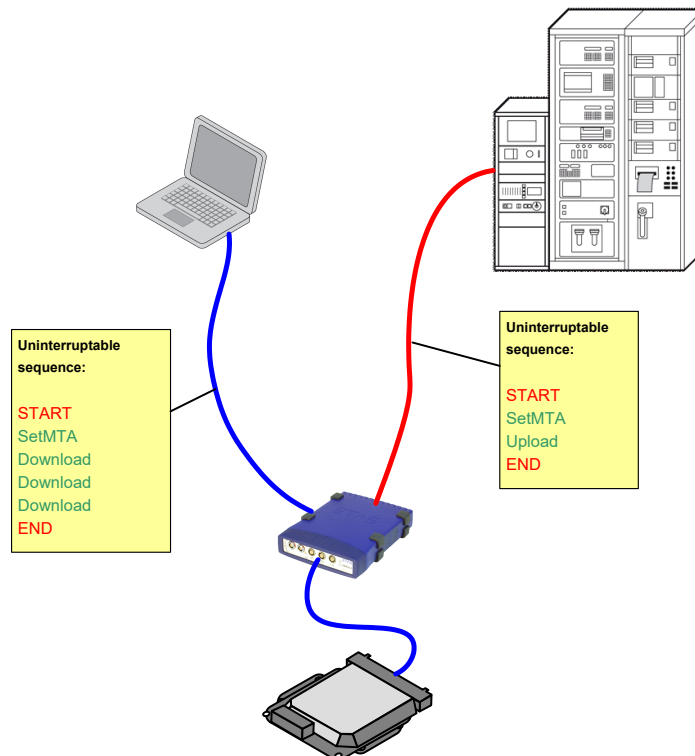
Before messages are lost, resulting in data inconsistency, the ES910 reduces the speed of calibration.

#### 10.2.4 Arbitration

As the ECU can be accessed via various interfaces (INCA, ES910 and automation system) and this does not have to be synchronized, overlapping requests could lead to inconsistent data exchange with the ECU.

An arbitration mechanism was implemented in ES910 to avoid this. This mechanism ensures that sequences that logically cannot be interrupted are not interrupted by other interfaces.

In this case, the sequence already running is ended before the sequence of another interface is started.



This results, for example, in INCA blocking access via the ES190 interface during download until download has been completed.



To be able to guarantee the expected time response, please ensure that there are no collisions as regards ECU accesses during automation.

### 10.2.5 iLinkRT V1.3

With iLinkRT V1.3, you can configure virtually any number of calibration variables for calibration (the protocol itself allows 65536 variables per ECU; in INCA, this number is limited to 5000 variables).

Each individual calibration access is triggered by a command (e.g. with `CAL_DOWNLOAD_ADVANCED`) and acknowledged by the ES910. It is also possible to read out the currently valid value in the ECU, for example, with `CAL_UPLOAD_ADVANCED`.

#### Calibrating 1D and 2D Parameters

When calibrating `CURVES`, `MAPS` and `VAL_BLK`, iLinkRT supports two modes:

- Flat  
All entries are calibrated to a single value that means a 2D map is thus "flat".
- Area  
Values in one part of the `CURVE`, `MAP` or `VAL_BLK` can be calibrated individually.

#### Axes and Data Points

When an array with physical axes (`CURVE`, `MAP`) is configured for ES910, the number and the values of the data points are kept fixed.



#### Note

If an element with axes (`curve`, `map`) is selected in the overall MCE configuration (interactive or remote configuration), you can no longer modify the axes in the experiment.

This is also true, for example, with changes to the underlying curve with `CURVE_AXIS`.

Modifying data points values via iLinkRT is not possible either.

#### Troubleshooting

Every calibration request via iLinkRT is acknowledged (as with XCP) – if calibration fails, the iLinkRT master is notified.

### 10.2.6 EtherCAT

With EtherCAT, the number of configurable measurement and calibration variables is limited.

 **Note**

Contact ETAS for a description of the specifics of ECU data representation and necessary implementation details in the EtherCAT context.

### Calibrating 1D and 2D Parameters

When calibrating CURVES, MAPS and VAL\_BLK, EtherCAT supports just one mode:

- Flat

All entries are calibrated to a single value that means a 2D map is thus "flat".

 **Note**

If an element with axes (curve, map) is selected in the overall MCE configuration (interactive or remote configuration), you can no longer modify the axes in the experiment.

### Troubleshooting

As the calibration variables are sent cyclically with EtherCAT, there are no commands for calibrations and no subsequent sequence of acknowledgements. If calibration is attempted, even though the last calibration procedure is not yet completed, there is no queue either. However, the ES910 provides a status variable via EtherCAT that

- shows whether the relevant ECU interface is free for calibrations and
- provides a response when a calibration is successful or has failed.

If this variable is ignored and there is still an attempt to carry out calibration<sup>1)</sup> (although the ECU is still busy with the previous calibration procedure), this calibration request is rejected. An evaluation as to whether a further, new change should be requested takes place at the next bus clock and only when the current calibration procedure has been completed.

### Special Features with EtherCAT when Calibration Variables are used by Several Interfaces

With standard ECU and automation protocols, it is customary that a value is only sent if it is to be changed.

---

<sup>1)</sup> "Calibration" can refer to an entire set of calibrations as EtherCAT makes it possible to request a simultaneous change of all parameters.

With EtherCAT, however, all calibration variables configured at the EtherCAT master are sent cyclically via the EtherCAT bus. This also takes place for values that have not changed. For example, target values are sent cycle-controlled even if the parameter has already adopted the EtherCAT target value.

To ensure the ECU interface is not unnecessarily occupied, the MCE device only transfers the changed values from the EtherCAT bus.

That means that the determination of the change is not linked to the actual value in the ECU, but just to the values that were sent at the EtherCAT bus.

This leads to problems when the value of a variable calibrated via EtherCAT is calibrated, for example, via the interface of an automation system (ASAP3). This calibration is not registered by the EtherCAT bus interface – this prevents the value of the calibration variable being overwritten with the configured value in the next EtherCAT bus cycle.

### *Example*

The calibration variable "x" is calibrated to the value "n" via EtherCAT. Then, the calibration variable "x" is calibrated to the value "m" via ASAP3.

If calibration variable "x" is to be recalibrated to the value "n", this can take place as follows:

- Directly via ASAP3
- In two steps via EtherCAT: Calibrate the variable to the value "k" and then to the value "n".

This is why it is recommended that a calibration variable shall only be calibrated via ES910 and another interface deliberately and when absolutely necessary.

## 11 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](http://www.etas.com/hotlines)

ETAS offers trainings for its products:

[www.etas.com/academy](http://www.etas.com/academy)



### ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	<a href="http://www.etas.com">www.etas.com</a>

## Glossary

### D

#### DAQs

Data Acquisition Communication; DAQs are the measurement packages that are triggered by the ECU.

### E

#### EtherCAT

EtherCAT is a real-time automation bus based on Ethernet. The open protocol is defined by ISO and IEC standards.

### I

#### iLinkRT

iLinkRT is an open protocol specifically developed for Fast ECU Access. It is based on the automotive standard XCP-on-Ethernet, limited to its measurement and calibration functionality. iLinkRT uses 12 commands of the XCP command set and introduces 5 additional user commands utilizing the USER\_CMD mechanism of XCP.

### M

#### MCE

Measurement and Calibration Embedded

#### MCE-PC

The MCE-PC is a Linux-based computer used in the Indirect ECU Access calibration method as additional hardware to overcome performance limitations.

### X

#### XCP

eXtended Calibration Protocol; XCP is a vendor- and interface-independent protocol for communication between calibration tools and control units. XCP was specified in the ASAM e.V.

## Index

<b>A</b>	
Application examples .....	16
Arbitration .....	72
ASAP3 .....	61
<b>C</b>	
Calibrating .....	70
Calibration access .....	70
Configuration changes .....	69
Contact Information .....	76
<b>D</b>	
Data .....	8
<b>E</b>	
ETAS	
Contact Information .....	76
EtherCAT .....	30,74
Ethernet layer-2 software bridge	
configuration .....	32
Explicit configuration	
ASAP3 .....	62
<b>G</b>	
glossary .....	77
<b>I</b>	
iLinkRT .....	13,29
Implicit configuration	
ASAP3 .....	62
INCA-MCE interface configuration .....	29
Installing .....	20
<b>M</b>	
MCE configuration .....	54,60
MCE interface configuration .....	33,36,52,66
MCE protocols .....	13
Measuring .....	67
Multicast network configuration .....	38
<b>S</b>	
Special data types .....	69
<b>T</b>	
Tool API .....	60
<b>V</b>	
Variable selection window .....	56

X

XETK

assigning to MCE interface .....36,49

## Figures

Fig. 2-1: Calibration access solutions for INCA-MCE .....	9
Fig. 2-2: Typical test bench setup .....	10
Fig. 2-3: Test bench system using the iLinkRT V2 at the computer solution .....	11
Fig. 2-4: Test bench system using the iLinkRT V3 at the computer solution .....	12
Fig. 2-5: Test bench system using the MC embedded solution .....	12
Fig. 2-6: Test bench system using the MC embedded solution .....	15
Fig. 2-7: The advantage of fast ramps in terms of a stable operating rangeSource: ATZ elektronik 2/2008, "iLinkRT High Performance ECU Calibration". .....	17
Fig. 2-8: ECU ramps with ASAP3 or iLinkRTSource: ATZ elektronik 2/2008, "iLinkRT High Per- formance ECU Calibration" .....	18
Fig. 5-1: Connecting an XETK for MCE .....	33
Fig. 5-2: Setup of INCA and iLinkRT system on the same computer .....	40
Fig. 6-1: Rear side of the MCE-PC .....	41
Fig. 6-2: Front side of the MCE-PC .....	42
Fig. 6-3: Schematic visualisation of iLinkRT V3.0 - IP addressing .....	43
Fig. 10-1: Undersampling at the EtherCAT Interface .....	68
Fig. 10-2: Oversampling at the EtherCAT Interface .....	69



## Tables

Tab. 2-1: Comparison of iLinkRT and EtherCAT when used with INCA-MCE .....	14
Tab. 5-1: INCA-MCE version and version of the iLinkRT protocol .....	30
Tab. 5-2: INCA-MCE version and EtherCAT specification .....	30