

# ETAS INCA V7.6



Getting Started

## 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 2026** 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 V7.6 | Getting Started R01 EN | 03.2026

## 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	INCA Basic Workflows .....	9
2.1	Product Structure from a User's Point of View .....	10
2.1.1	Basic Software Functionality .....	10
2.1.2	Database Manager - DBM .....	10
2.1.3	Hardware Configuration Editor - HWC (Subsystem) .....	11
2.1.4	Experiment Environment - EE (Subsystem) .....	11
2.1.5	Measured Data Analysis - MDA (Subsystem) .....	12
2.1.6	Calibration Data Manager - CDM (Subsystem) .....	12
2.1.7	User Interface Developer - VUI (Subsystem) .....	13
2.1.8	ASAM-MCD-2MC Editor (Subsystem) .....	13
2.2	Working with the Database Manager - DBM .....	14
2.2.1	Creating a New Database, Folder, Workspace, and Experiment .....	15
2.2.2	Setting Up a Project .....	16
2.2.3	Adding Bus Descriptions and Further Database Items .....	17
2.2.4	Setting Up the Workspace .....	18
2.2.5	Exchanging Data Using the Import/Export Function .....	19
2.2.6	Working with Databases .....	20
2.2.7	Working with Database Items .....	22
2.3	Working with the Hardware Configuration Editor - HWC .....	23
2.3.1	Adding, Configuring, and Initializing Hardware .....	24
2.3.2	Matching Data Versions Between the PC and ECU .....	25
2.4	Working with the Experiment Environment - EE .....	26
2.4.1	Selecting Measure and Calibration Variables .....	26
2.4.2	Configuring the Display .....	27
2.4.3	Configuring the Usage of Variables .....	28
2.4.4	Using Layers in an Experiment .....	29
2.4.5	Configuring Measured Data Recording .....	30
2.4.6	Performing a Measurement .....	31
2.4.7	Performing Calibrations in the Editors .....	32

2.4.8	Saving Datasets .....	33
2.5	Editing Datasets with the Calibration Data Manager .....	34
2.5.1	Selecting Datasets and Calibration Variables .....	34
2.5.2	Executing the List, Compare, or Copy Action .....	35
2.5.3	Analyzing Results and Output Files .....	36
3	INCA - Basic Operations .....	37
3.1	Window Structure .....	37
3.2	Toolbars .....	38
3.3	Operations Using the Keyboard .....	38
3.4	Operations Using the Mouse .....	39
3.5	Hierarchy Trees .....	40
3.6	Editable Tables .....	41
3.7	Sounds for INCA Events .....	42
3.8	Storage Directories .....	43
3.9	Help Functions .....	44
3.9.1	INCA Help Functions .....	44
3.9.1.1	Using INCA Help Buttons .....	45
3.9.1.2	Search Methods and Syntax .....	45
3.9.2	Manual and Tutorial .....	46
3.9.3	Video Tutorials .....	46
3.9.4	Keyboard Assignment .....	46
3.9.5	Monitor Window .....	46
3.9.6	New Features in INCA .....	47
3.9.7	Troubleshooting .....	47
4	INCA - Concept .....	48
4.1	General Description .....	48
4.1.1	System Overview .....	48
4.1.2	Open Interfaces .....	49
4.2	Calibration Basics .....	50
4.2.1	Standard Interfaces Based on the ASAM-MCD-Model .....	52
4.2.2	Interfaces to the Control Unit- ASAM-MCD-1 .....	53
4.2.3	Calibration with ETK (Parallel Control Unit Interface) .....	54
4.2.4	Calibration via the Serial Interface .....	56
4.2.5	Measurement and Calibration Hardware .....	59
4.3	INCA Basics .....	60
4.3.1	Data Management Concept for Working Page and Reference Page .....	60
4.3.2	Project, Master Working , and Reference Datasets .....	61

4.3.3	Experiment .....	62
4.3.4	Workspace .....	62
4.3.5	Relationships Between the Database Objects .....	63
5	Contact Information .....	66
6	List of Abbreviations .....	67
	Glossary .....	68

# 1 Introduction

The customer in the automotive industry associates an advanced, "intelligent" vehicle with comprehensive safety, a high degree of driving comfort, low fuel consumption, and low pollutant emissions. For automobile manufacturers, this means anti-lock brake and traction control systems, adaptive drive programs for automatic transmissions, map-controlled gasoline and diesel injection with adaptive controls, etc.

For developers, this means implementing complex functions and control algorithms in microprocessor-based control units, fine-tuning and optimizing these systems for different engine and vehicle types, and the time and budget pressures involved in mass production.

INCA is a measuring, calibration, and diagnostic system that provides comprehensive measuring support, aids you in all essential tasks during control unit calibration, evaluates the measured data, and documents the calibration results.

INCA:

- can be used in the vehicle, test bench, and office/lab
- consists of modular hardware and software
- can be customized for specific projects and applications
- can be customized to fit both high-end and low-cost control units
- can be integrated in the development process for control unit software.

## 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.

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.

## 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 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 INCA Basic Workflows

This chapter is aimed for new users of INCA helping them to get started with INCA V7.6. You will receive an overview of the program functionality and working principles by means of practice-oriented work examples presented as flow diagrams. The essential working steps for calibration are presented in the framework of a concrete example, the lambda control for engine controllers.

The following table provides an overview of the key INCA modules and their corresponding abbreviations used in this document:

	DBM	Database Manager
	EE	Experiment Environment
	HWC	Hardware Configuration Editor
	CDM	Calibration Data Manager
	WS	Workspace
	DB	Database

In the following chapters you get to know the INCA product structure and the basic workflows of INCA and its subsystems:

[2.1 Product Structure from a User's Point of View](#)

[2.2 Working with the Database Manager - DBM](#)

[2.3 Working with the Hardware Configuration Editor - HWC](#)

[2.4 Working with the Experiment Environment - EE](#)

[2.5 Editing Datasets with the Calibration Data Manager](#)

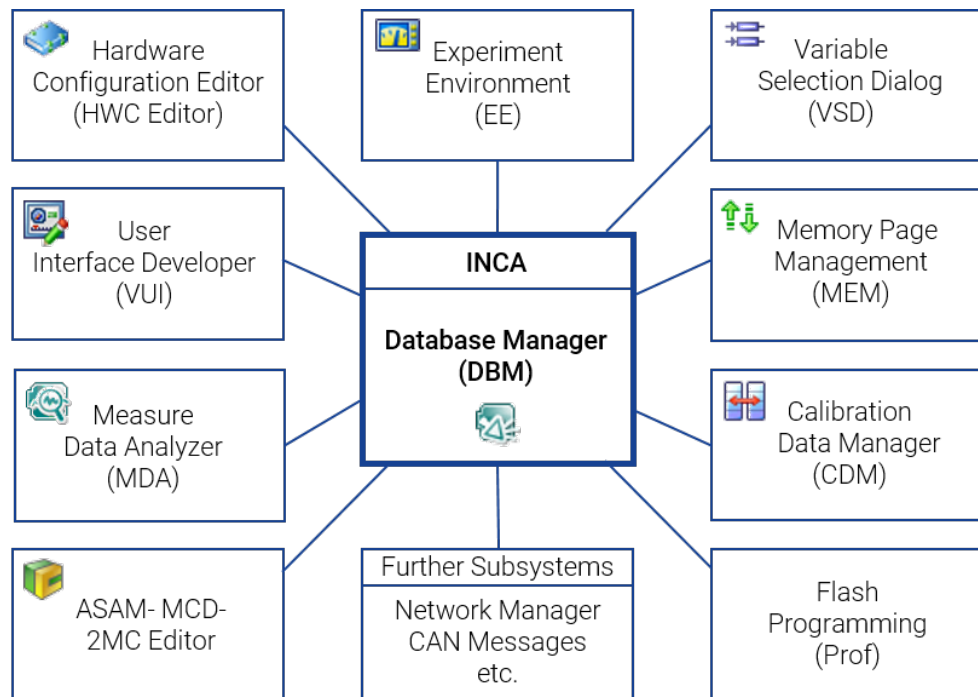
## 2.1 Product Structure from a User's Point of View

The following description of the modular structure provides an overview of INCA and its functionality:

INCA is designed according to a "building block concept". When needed, you can enhance INCA by so-called add-on modules (e.g., INCA-MIP, INCA-FLEXRAY). Any additional modules that you purchase later will integrate seamlessly into the accustomed operating concept. This principle allows individual customization of the entire system to your specific requirements.

### 2.1.1 Basic Software Functionality

The basic software is the outer framework that can be filled with exactly the "sub-systems" the user requires. Each subsystem supports a specific phase in the work process of a user and has its own user interface, that means its own window with menus and dialog boxes.



**Fig. 2-1:** INCA and its subsystems

The following sections briefly describe the functionality of each of the depicted components.

### 2.1.2 Database Manager - DBM

The Database Manager is at the heart of INCA. This is where you start the individual subsystems. You can also manage users and, via an option dialog box, set user-specific options, for example, the definition of storage directories, preferences such as screen display options, or the start-up behavior of the software.

The main purpose of the DBM is, however, to systematically store all data in a database that are created during the calibration process (workspaces, experiments, projects, and datasets). The DBM allows you to manage the database objects from a comprehensible user interface. Similar to the Windows Explorer, you can create directories and subdirectories, move, copy, import, and export individual items and also create entirely new databases. This means that you can organize your data in a similar fashion as you are accustomed to for the file system.

The different data types, which on the one hand are required for the measurement and calibration task and on the other hand are created or modified during the execution of the experiment, are stored in separate units. This has the advantage that the individual units can be reused and exchanged more easily. In addition to the various units, INCA also manages the existing references between the units. Using the database, the software "knows" these references and ensures the integrity of the data, making the user's life easier.

### 2.1.3 Hardware Configuration Editor - HWC (Subsystem)

The Hardware Configuration Editor subsystem assists the user in preparing for the measurement and calibration tasks. The main purpose of this phase is to let the software know what kind of hardware (measuring hardware and control unit interface) is being used. Users can configure the hardware used in a workspace optimized for this purpose. This means they can indicate the hardware and the interfaces through which they are connected to the PC. They can also specify which measured signals and calibration parameters are acquired through each input. For the control unit interfaces, the user can specify the desired project and the working and reference datasets.

The information entered describes the hardware equipment and is stored in the selected workspace as the hardware configuration. Using the overall system, which has been prepared and parameterized, you can perform a variety of partial tasks (e.g., cold start optimization, idling optimization or measurement of different vehicle components) using the "Experiment Environment" subsystem.

Thanks to the reusability of the workspace database object, the hardware configuration defined in it can be used for various experiments by simply copying the workspace.

### 2.1.4 Experiment Environment - EE (Subsystem)

The Experiment Environment subsystem mainly contains the functionality required to perform a measurement and calibration task and is used when preparing the individual partial tasks. The data required for a measurement and/or calibration task which is specific to this experiment can be stored in the database as an experiment.

A workspace refers to one specific experiment. An experiment, in turn, can be included in any number of workspaces.

The data contained in experiments include:

- measuring channels containing the settings specific to this experiment
- measured signals/parameters required for this experiment
- measuring and calibration window required for this experiment

An experiment does not include any hardware-specific information. It also includes no information on the data and program version. It only references these data.

The functionality of this subsystem can be summarized in four groups:

- **Configuring**

The user can select the settings required for the respective partial task, i.e. select variables to be measured and calibrated, and set sampling rate, measuring range, and display parameters for each variable.

- **Measuring and recording**

You can customize the user interface (measurement and calibration windows) to the task at hand. The measuring control information (recorders used for different recordings, measuring times, trigger conditions, repeat triggers, etc.) is displayed and the recordings are carried out.

- **Calibrating control unit parameters**

You can modify characteristics graphically or numerically in a variety of different calibration windows (e.g., numerical editor, combined curve editor, etc.). You can also perform mathematical operations such as addition or multiplication with a factor. In INCA it is possible to perform measurement and calibration tasks simultaneously.

- **Managing and Copying Data Versions**

The user can load data versions into the control unit, copy them from the control unit to the hard disk and exchange them between different users.

## 2.1.5 Measured Data Analysis - MDA (Subsystem)

Here, the measured data files created can be analyzed. Functions such as a measuring cursor, zooming in the x and y directions and superimposing of signals aid the user in this work. Individual settings for recurring tasks can be stored in configurations.

MDA is provided as a stand-alone program, which can be opened individually or from INCA.

## 2.1.6 Calibration Data Manager - CDM (Subsystem)

The values of individual or all parameters of a "source dataset" can be represented in tabular form.

- **Comparing**

Value changes of parameters between a source dataset and any number of other "comparison datasets" can be represented on a physical level.

- **Copying**

Individual or all parameters of a source dataset can be copied into a number of other "target datasets". Address changes or different conversion formulas are taken into account.

– **Listing**

The values of individual or all parameters of a "source dataset" can be represented in tabular form.

### 2.1.7 User Interface Developer - VUI (Subsystem)

Similar to a visual programming environment, the User Interface Developer allows you to arrange any display and calibration elements in a window based on ergonomic considerations. Additional design elements such as labels, lines, rectangles, and split bars assist in dividing the surface into function areas. These screen layouts can then be integrated in the experiment environment of INCA.

### 2.1.8 ASAM-MCD-2MC Editor (Subsystem)

The ASAM-MCD-2MC Editor allows you to analyze and edit the items of existing projects. This means you can use the ASAM-MCD-2MC Editor to view and edit the calibration and measurement variables, functions, and other specific information for parameterizing the calibration interface that have been defined in the corresponding A2L file. This requires that you have created a project in INCA in the form of a project description file (\*.a21).



#### Note

You can find a description of the ASAM-MCD-2MC standards and the possible parameters for the items in the Internet at <http://www.asam.de>.

## 2.2 Working with the Database Manager - DBM

The main purpose of the DBM is to systematically store all data that are created during the calibration process (workspaces, experiments, projects and datasets) in a database and manage them through a clear and well-organized user interface. Similar to the Windows Explorer, you can use the Database Manager to create directories and items as well as to move, copy, import, and export individual items and also to create entirely new databases.

You can organize all data centrally and independently from projects while always keeping a clear overall picture.

The following data are managed in the database:

- Workspaces (including hardware configuration)
- Experiments
- ECU projects (A2L) and datasets
- CDM configurations
- Calibration Scenario Configurations
- Measure catalogs
- ECU documentation or link to ECU documentation
- AUTOSAR system templates
- CAN DB descriptions
- CAN Message Lists (available only if the component "CAN MessageSending" has been installed)
- FIBEX files with configurations for communication via the FlexRay bus (FlexRay functionality requires the INCA-FlexRay add-on)
- LDF files with configurations for communication via the LIN bus (LIN functionality requires the INCA-LIN add-on)
- ODX project files for diagnostic issues (Diagnostic functionality requires the ODX-LINK add-on)

The following types of data are *not* managed in the database:

- measure files (\*.dat)
- Custom user interfaces (\*.vui)
- \*.a2l and \*.dbc files
- \*.hex and \*.s19 files

Database items specific to an experiment mostly consist of a workspace and an experiment. When using an ECU interface, at least one project with associated working and reference datasets is also included. The following section teaches you how to set up a new database, create new database items and organize them in folders.

### 2.2.1 Creating a New Database, Folder, Workspace, and Experiment

INCA allows you to work with multiple databases. This results in better performance because the data volume is kept small and easy to handle. Before you generate data, you should set up a new database and create a folder structure that matches your specific requirements. This makes it easier later on to assign data to certain vehicles and experiment setups. The advantage of a new database is that you will not be confused by any other entries. Once the folder structure is in place, start creating the various database items, which are initially still empty.

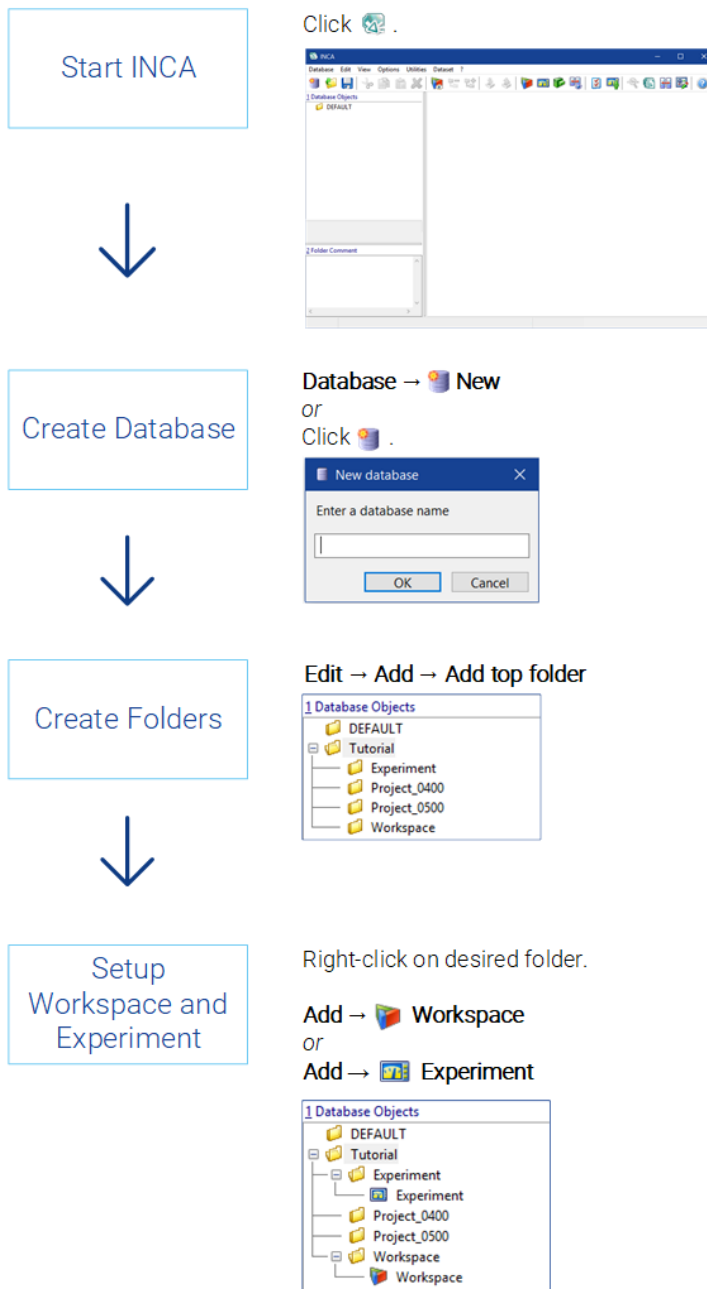



Fig. 2-2: Creating databases, folders, and database items

## 2.2.2 Setting Up a Project

Prerequisites for calibrating with INCA are creating a memory image of the control unit and storing the information in the form of datasets in INCA.

Before you can use INCA to apply different projects, you need the following files:

- Project description file (\*.a21) with the physical description of the data
- HEX file (Intel Hex or Motorola format, e.g.: \*.hex or \*.s19) containing the ECU program, which consists of code and data
- (Optional) control device documentation if the control device manufacturer provides such documentation

The different data statuses of working and reference pages are stored separately in INCA as a working dataset and a read-only reference dataset. Read-only datasets are shown with a red frame .

When loading the first HEX file, the code portion is mapped to the control unit project (transparent to the user). The data portion of this HEX file is stored as the so-called "Master" dataset. The master dataset is then used to also create the required working dataset.

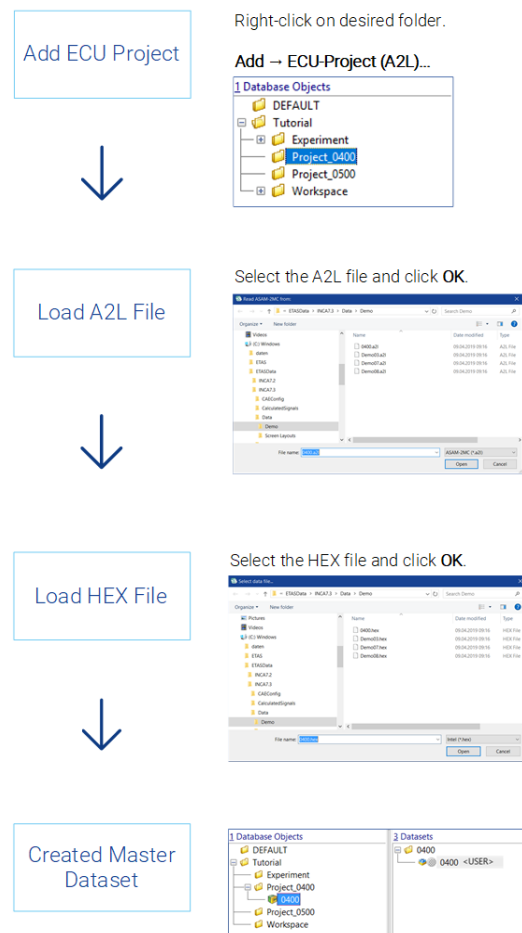


Fig. 2-3: Setting up a project

### 2.2.3 Adding Bus Descriptions and Further Database Items

If you want to exchange information via a vehicle bus (e.g. CAN), you also need bus description files defining the system topology and/or the bus communication.

Monitoring the CAN messages on the CAN bus requires a CAN monitoring description; this can be provided either by a CANdb file or by an AUTOSAR system template (\*.arxml).

For the FlexRay and the LIN bus, you need descriptions from FIBEX, LDF, or AUTOSAR files respectively, and ODX diagnostic functionality requires an ODX project file. Please note that FlexRay, LIN and ODX functionality is only available if the corresponding INCA add-on is installed.

Depending on your task and individual work procedures, you may want to add other items for later use, e.g. a CAN message list (used for sending CAN messages from INCA to the bus), a measure catalog (collection of pre-defined settings for measure channels of a measurement hardware), or an ECU documentation (description of each single variable defined in a project description file (\*.a21)).

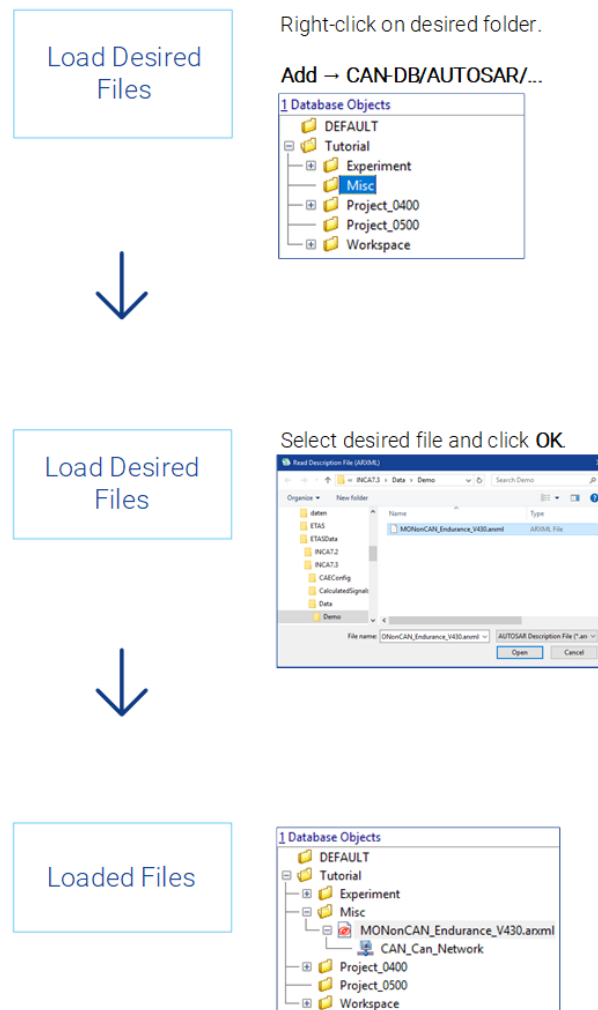


Fig. 2-4: Adding bus description files and further database items

## 2.2.4 Setting Up the Workspace

When you create a workspace in the Database Manager, it does not contain references to other database objects. To set up this workspace, you have to assign a project, select the appropriate project hardware and allocate an experiment.

When you select a workspace, you can use the menus specific to each database item to modify the workspace, or you can enter the entries in the **3 Experiment**, **4 Project/Device** and **5 Hardware** fields.

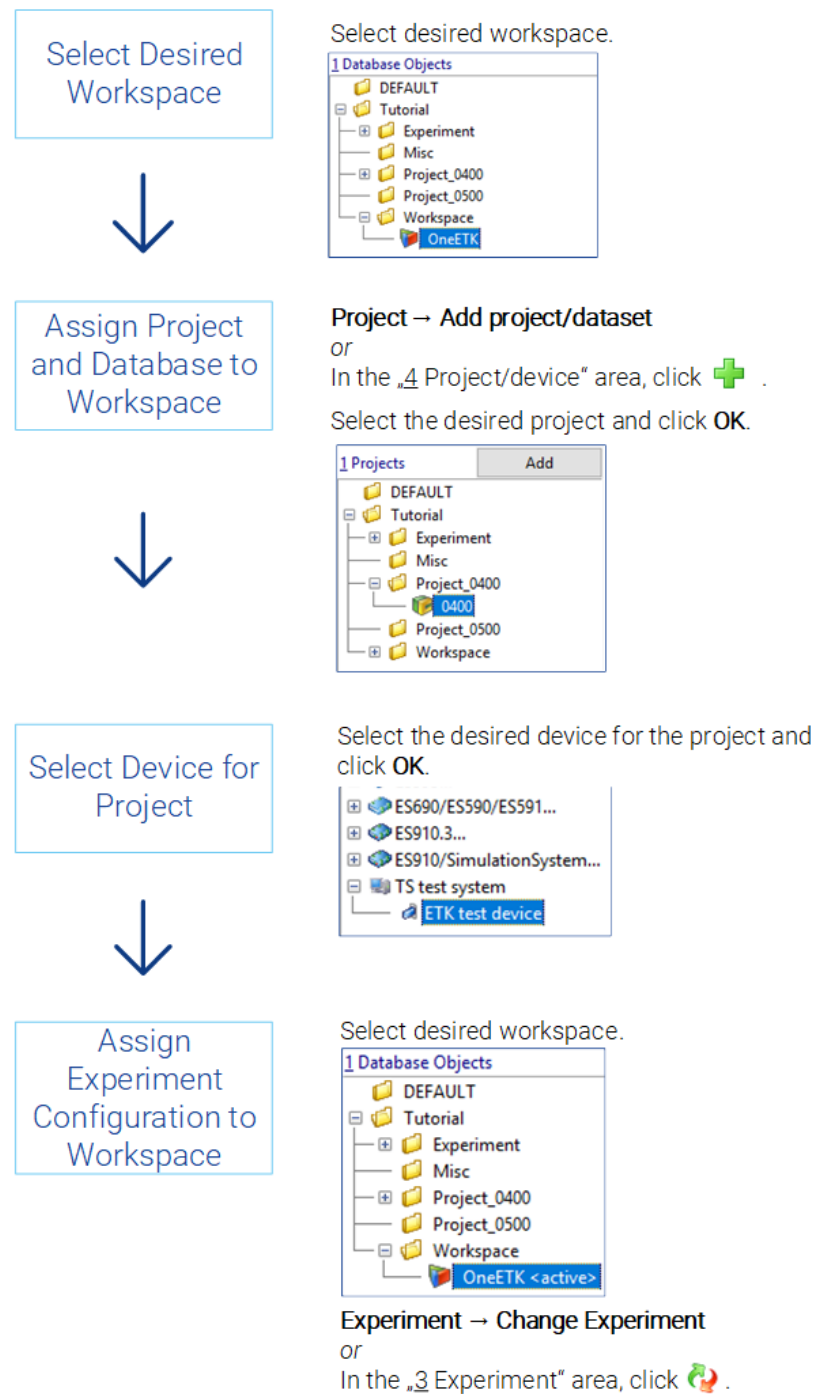


Fig. 2-5: Setting up the workspace

### 2.2.5 Exchanging Data Using the Import/Export Function

You can use the Database Manager to export or import all or only selected items of a database. The Import/Export function is an invaluable tool when making use of existing experiments. Consistent with the general management concept of dividing the DBM window into a left and right pane, you can use the Export/Import functions for the items in the "Database objects" field by choosing options in the **Edit** menu and for the datasets in the "Datasets" field by choosing the **Dataset** menu. Consequently, there are different export data, although the file extension (\*.exp) is the same. To avoid conflicts, you should give these files distinctive file names indicating the particular item type.

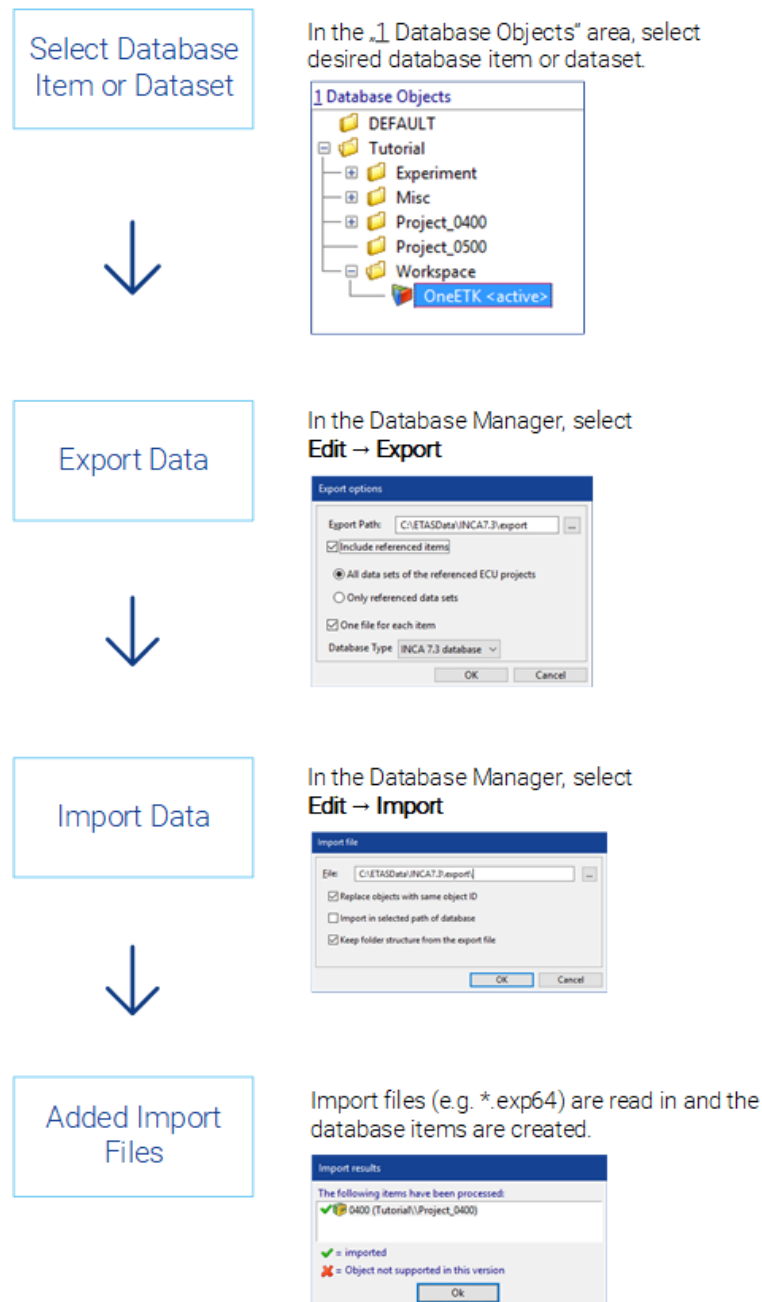


Fig. 2-6: Exchanging data by exporting and importing

## 2.2.6 Working with Databases

You can work with several databases in INCA. Apart from creating a new database, which initially contains no objects, you can build on existing databases, for example, those of colleagues, by loading such a database. Working with databases also naturally includes backing them up and updating them.

For these purposes, INCA provides four partly combinable functions:

- **Optimize**

Executes a defragmentation, i.e. rewrites fragmented parts of the database to increase the speed of access and retrieval.

- **Convert**

Rewrites the complete database to increase the speed of access and retrieval.

- **Repair**

Checks whether the internal objects and the logical structure of the database can be read. If any inconsistencies are detected, these will be repaired. Afterwards, the database will be optimized and converted.

- **Check**

Checks whether the internal objects and the logical structure of the database can be read. Detected inconsistencies will be listed in the monitor window.

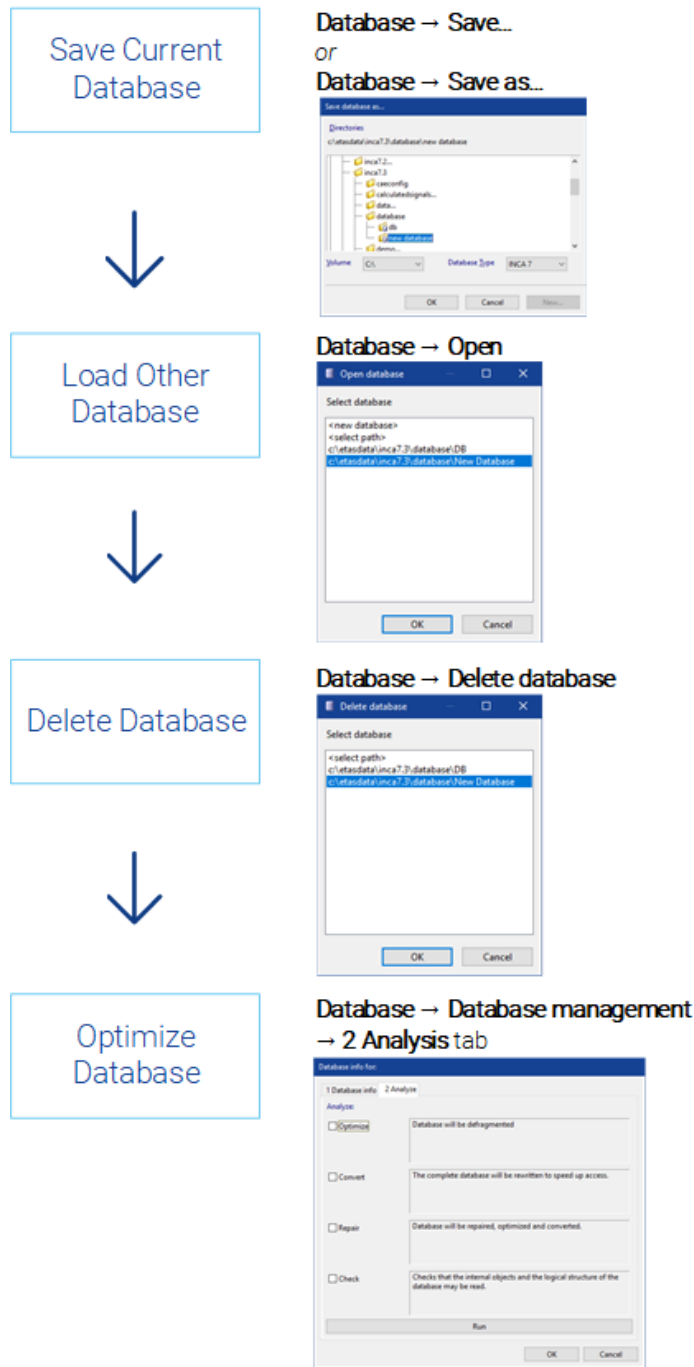


Fig. 2-7: Working with several databases

### 2.2.7 Working with Database Items

Depending on the selected database item, such as workspace, experiment, project, etc., the Database Manager (DBM) provides you with different processing functions. It must be distinguished between general processing functions, which apply commonly to all database items in the left window ("Database objects" field), and processing functions that apply to specific database items. The latter can be used only after the corresponding items have been selected in the left window, and they affect only the items in the right window.

Working with Database Items

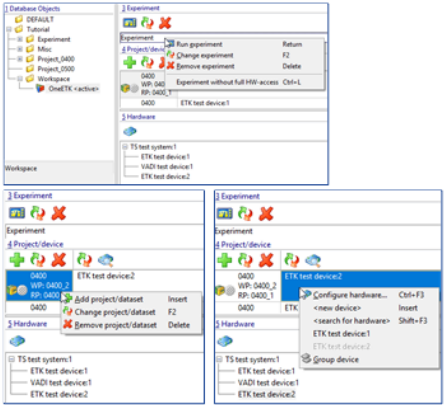
↓

Working with Experiments

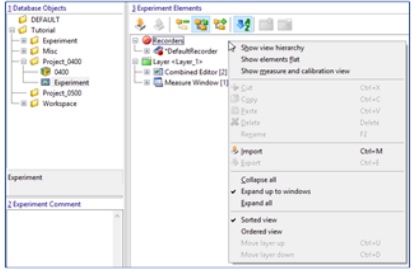
↓

Working with Projects

In the „1 Database Objects“ area, select desired workspace.  
Right-click on the corresponding area to open the desired context menu.



In the „1 Database Objects“ area, select desired experiment.  
Right-click in the „3 Experiment Element“ area to open the context menu.



In the „1 Database Objects“ area, select desired dataset.  
In the „3 Dataset“ area, right-click to open the context menu.

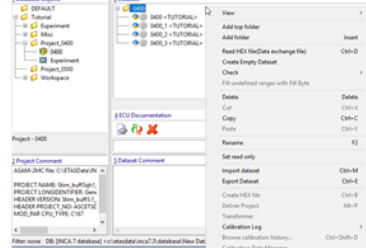


Fig. 2-8: Working with database items

## 2.3 Working with the Hardware Configuration Editor - HWC

Essentially, the Hardware Configuration Editor enables you to manage and configure the hardware for the active workspace. Apart from the possibility just mentioned to inform the software about the hardware matching the project description file, the HWC also allows you to add further components from a list of possible modules, for example measuring hardware. You can use the Memory Page Manager integrated in the HWC Editor to match different data versions between the PC and ECU.

### 2.3.1 Adding, Configuring, and Initializing Hardware

You can select different settings, for example, module parameters, channel settings, etc., for each hardware component using associated dialog boxes. When you add hardware components manually, the hardware still needs to be initialized.

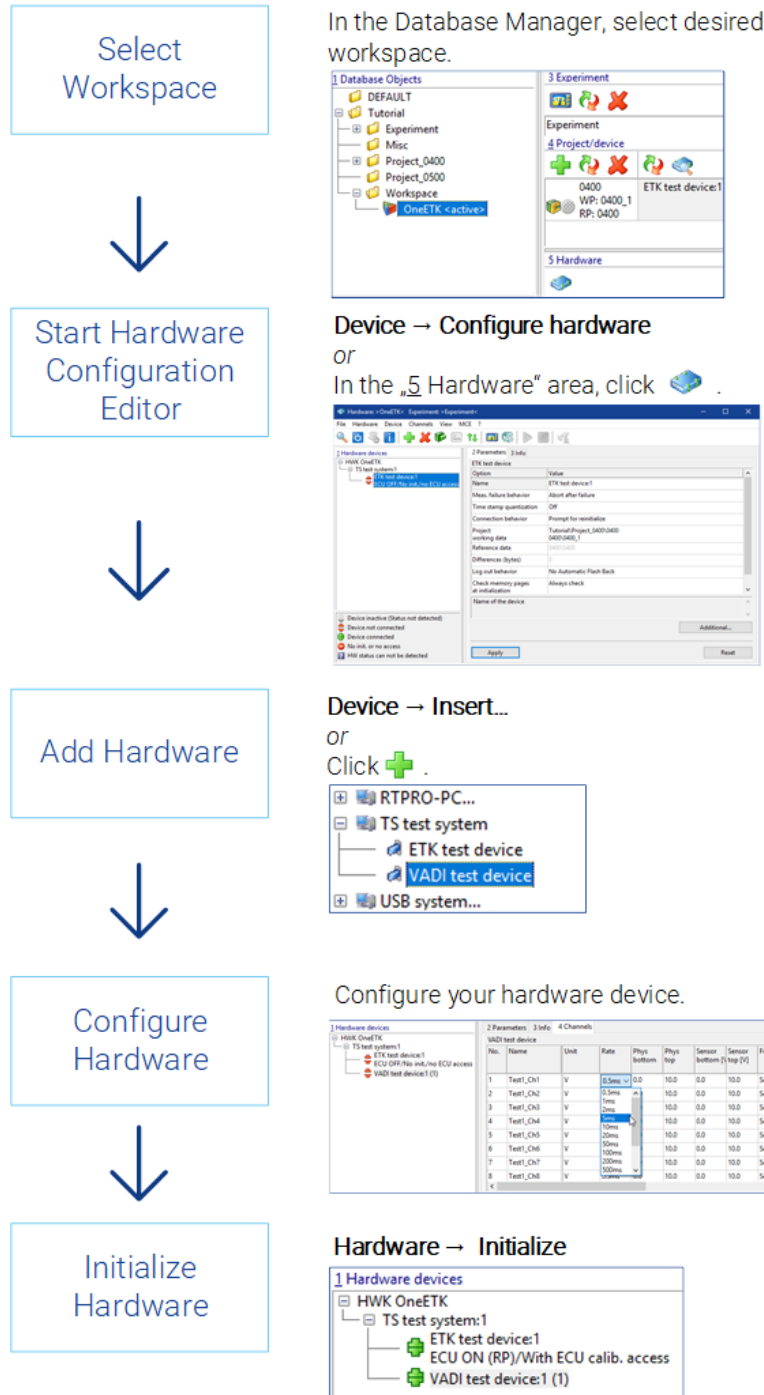


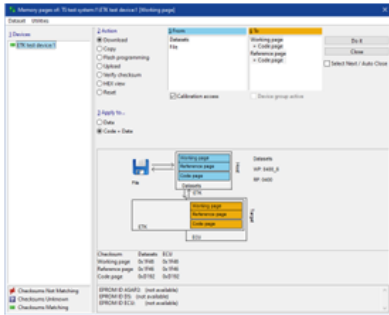
Fig. 2-9: Setting up the hardware

### 2.3.2 Matching Data Versions Between the PC and ECU

INCA provides the Memory Page Manager to manage different datasets (working and reference datasets). This is a versatile tool which you can use to copy memory contents in any direction. For example, it allows you to read data versions into and from the control unit or copy data from the working data version to the reference data version or vice versa, save working datasets or write-protect them as an "intermediate result" and change reference datasets for the current project. The Memory Page Manager is opened automatically if different data versions are detected during the initialization of the control unit. You can call the Memory Page Manager both from the HWC Editor and the Experiment Environment by pressing <Shift> + <F8>.

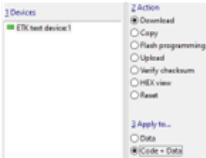
Open Memory Page Manager

In the Hardware Configuration Editor , select **Device** → **Manage memory pages...** or Click



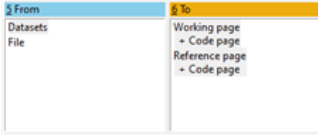
Select Type and Scope of Data Transfer

In the „Devices“ area, select the devices. In the „Action“ area, activate **Download**. In the „Apply to...“ area, select **Code & data**.



Select Source and Destination of Data Transfer

In the „From“ area, select **Datasets** as source. In the „To“ area, select **Working page+Code page** and **Reference page+Code page** destination.



Verify Settings and Execute Action

Verify and click **Do It** to start the action.

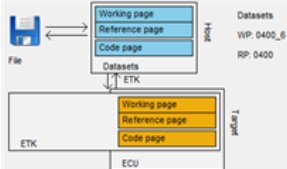


Fig. 2-10: Matching data versions

## 2.4 Working with the Experiment Environment - EE

The Experiment Environment mainly contains the functionality required to perform a measurement and calibration task and used when preparing the individual partial tasks. It is used to select measure and calibration variables and their arrangement in various windows, such as oscilloscope, bar display, and table or graphical calibration editors.

### 2.4.1 Selecting Measure and Calibration Variables

Depending on the hardware configuration used in the workspace, the experiment only provides the measure and calibration variables for selection which are defined in the connected hardware.

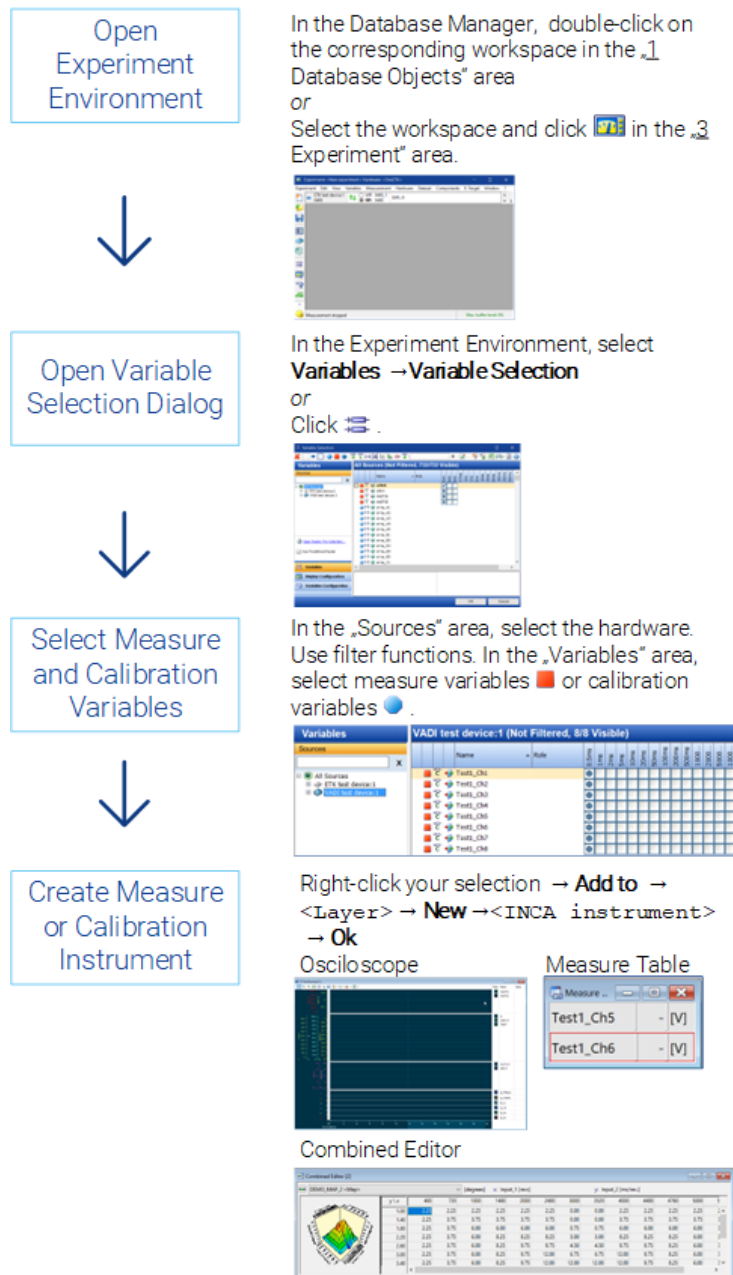




Fig. 2-11: Selecting the measure and calibration variables

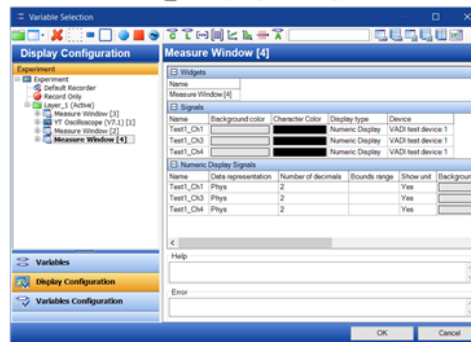
### 2.4.2 Configuring the Display

For configuring the representation of measure and calibration variables, you have a range of measure and calibration instruments with the associated display options at your disposal. You can either select the adequate instrument for each variable individually or accept the instrument suggested by the program as a default. The following describes how to subsequently configure the representation.

Open Display Configuration

In the Variable Selection Dialog , click **Display Configuration**

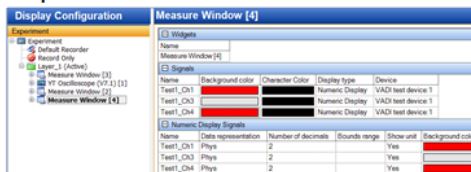
OR  
In the Experiment Environment, select **Variables** →  **Display Configurations...**



Modify Display Settings

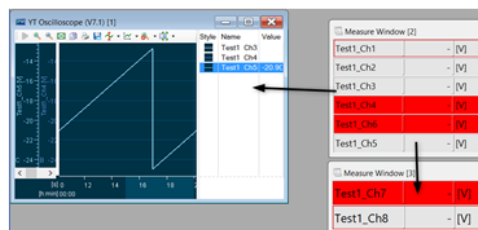
In the „Display Configuration“ area, select a measure window or a variable.  
In the „Measure Window“ area, define specific display settings and click **OK**  
OR

In the Experiment Environment, select the measure window and right-click and select **Properties**.



Move Variables between Instruments

Move variables from one window to another with drag & drop or cut & paste.



**Fig. 2-12:** Configuring the experiment

The assignment of the variables, the arrangement of the individual instruments, and other settings such as color and display area are saved in the experiment.

### 2.4.3 Configuring the Usage of Variables

Apart from setting how the variables are displayed in the experiment, you can also set their raster, the recording status and whether they shall be used at all in the experiment.

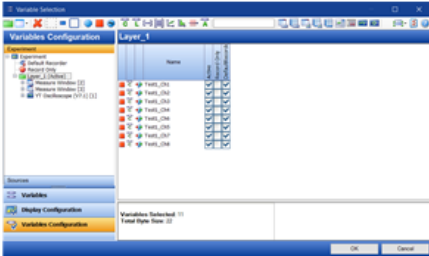
It can be useful to set variables to inactive; they will still be saved in the experiment, but temporarily not used in the experiment, thus not filling the raster.

They can easily be reactivated later, with the previous properties and instrument assignment being saved.

Moreover, you can modify the raster assignment and the assignment to the recorders that are configured in the experiment.

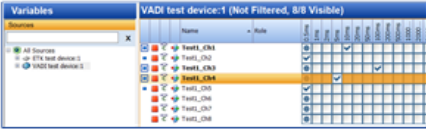
Open Variables Configuration

In the Experiment Environment, select **Variables → Variable Configuration...**  
or Click .



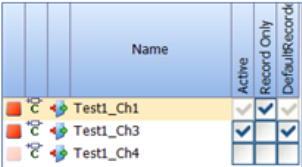
Modify Raster Assignment

In the „Variable Configuration“ area, select the experiment element with the variables you want to configure.  
Modify raster assignment by selecting the raster from the combo-box.  
Multiple selection of variables is possible.  
 Preselected raster  
 User defined raster



Set the Display and Recording Status

- To remove the variable temporarily from the experiment, disable the checkbox in the „Active“ column.
- To set a recorder, enable the checkbox of the desired recorder.
- To record variables without being displayed in an instrument, enable the checkbox in the „Record Only“ column.



If not all columns are displayed, right-click on a column. Enable the desired columns.

Fig. 2-13: Configuring variables

## 2.4.4 Using Layers in an Experiment

To improve the display of all measure and calibration instruments required in an experiment, INCA allows you to distribute the instruments to different layers. This is useful especially when working with small screens. You can create any number of layers and delete them as required. For example, you can define one layer for the measure instruments and another one for the calibration instruments. After creating the layers, you can distribute the instruments as required. Just click the mouse to move any layer to the foreground. Saving the experiment also saves all layers with their associated instruments.

A newly created experiment initially only contains one layer. After you have created additional layers, the footer of the experiment environment shows the associated tabs. To switch between the individual layers, just click on the corresponding tab or press <ALT> + <<<> ("less than" character) or <ALT> + <>> ("greater than" character).

Layers can be configured in the Display Configuration or the Variables Configuration of the Variable Selection dialog.

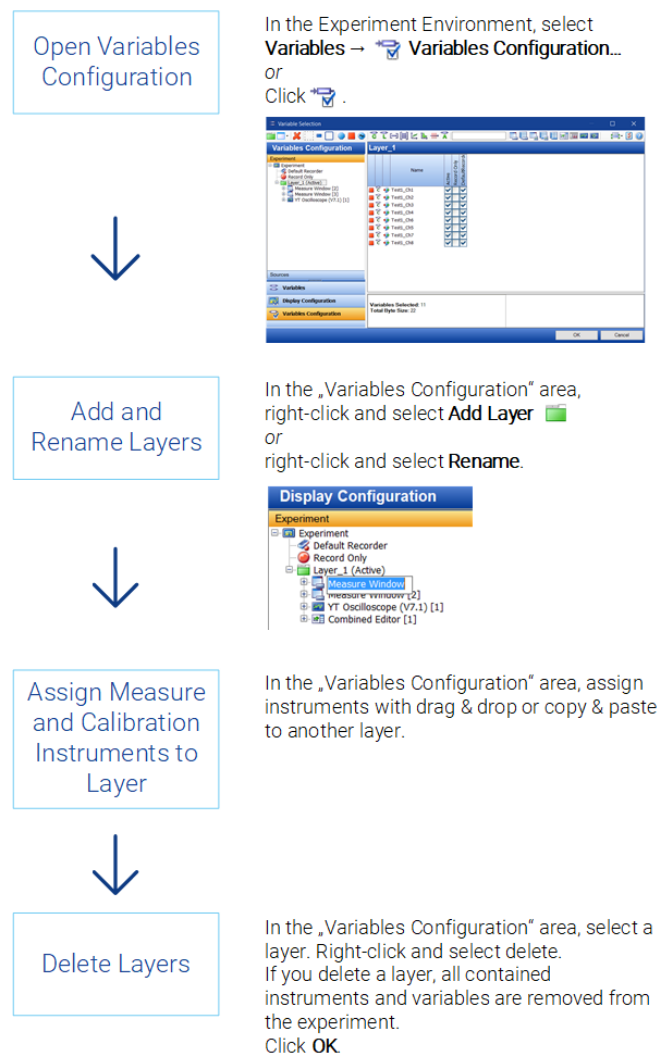


Fig. 2-14: Using layers in an experiment

### 2.4.5 Configuring Measured Data Recording

INCA provides extensive and convenient methods in a dedicated configuration window to prepare for and configure the storage of measured data. You can use them to specify the storage of the measure file by setup options such as storage location, storage name, format (ETAS binary file, ASCII, Matlab M-File, FAMOS, DIADEM and MDF) and other individual information.

To automate the start, repetition, and end of measure data recording, you can define trigger conditions (with pretrigger time and posttrigger time, if required) which trigger the desired event.

You can choose whether you just use one easy-to-use default recorder; as an alternative, you can individually configure different recorders that can be run separately from each other and that can be used for background recording. The following process describes the usage of the default recorder.

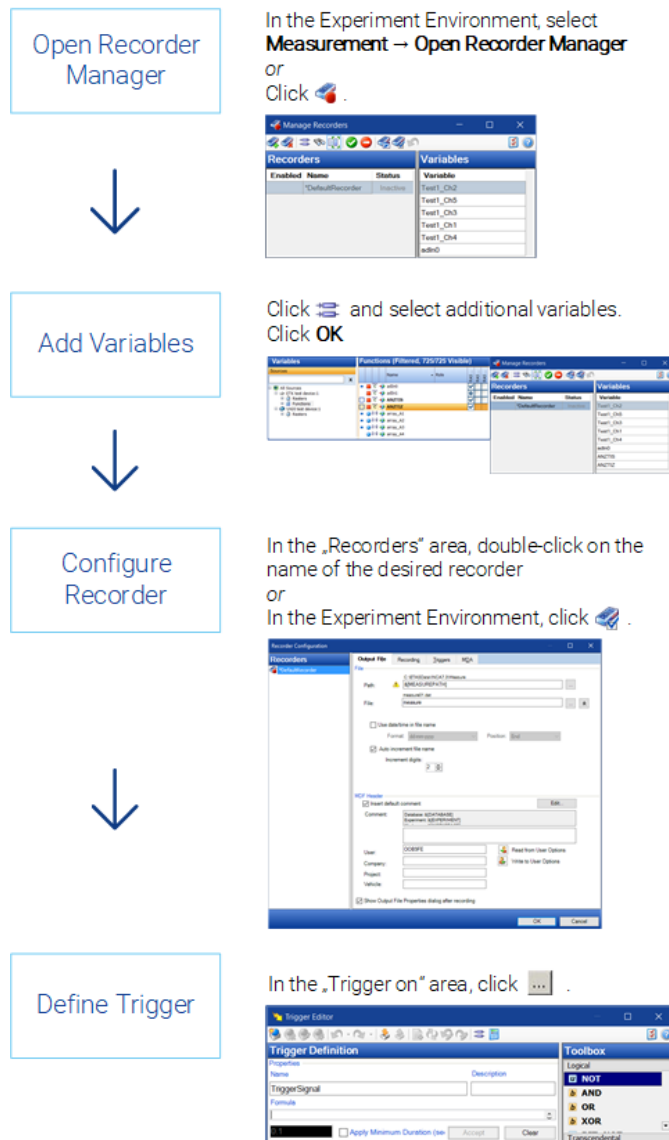


Fig. 2-15: Configuring measure data recording

### 2.4.6 Performing a Measurement

You have already finished all preparations. Now, you can start the actual measuring process. INCA offers you various measuring functionalities, which are supported by function keys. The measurement status is shown in the status bar. Refer to the INCA help for more detailed information.

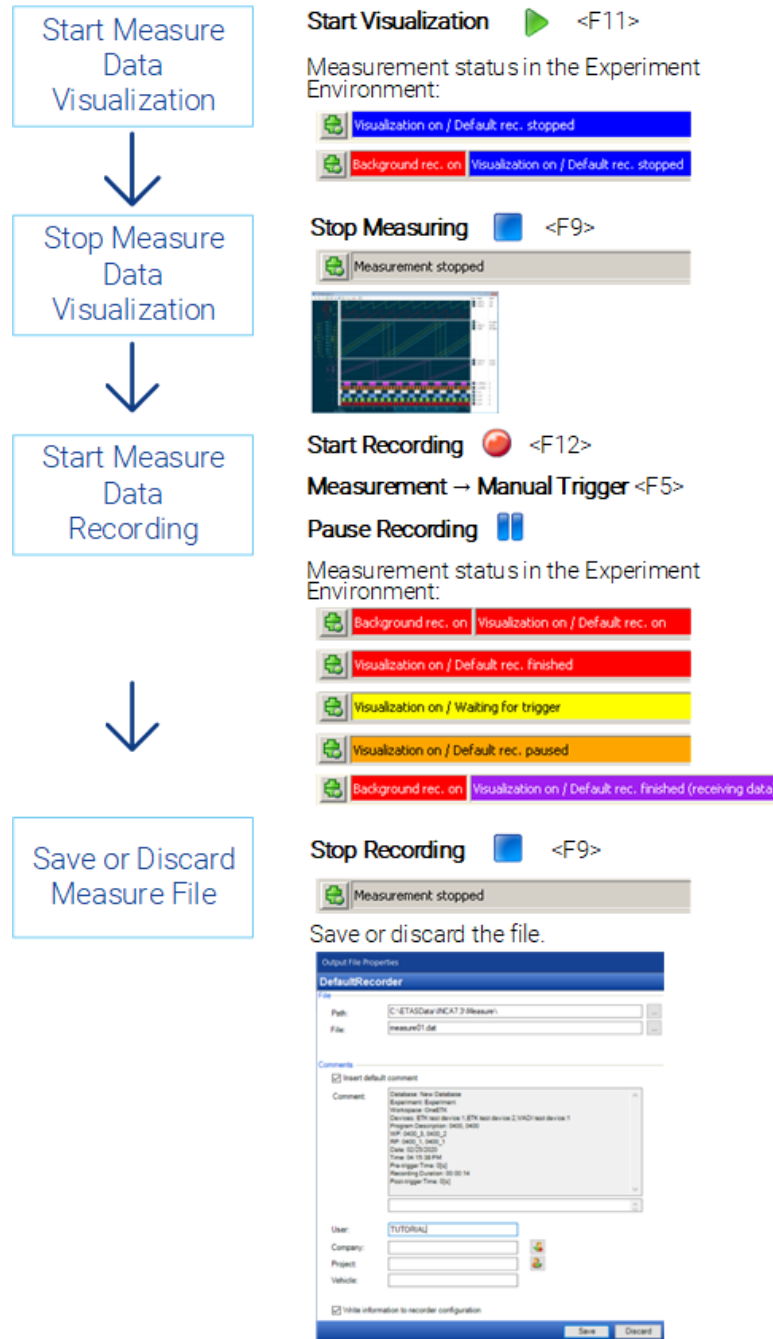


Fig. 2-16: Performing the measurement

### 2.4.7 Performing Calibrations in the Editors

The various calibration instruments display the calibration variables (scalars, curves, and maps) in the form of tables or graphics. The calibration instruments are also used as editors you can use to change the values of calibration variables directly in the visualization. INCA also assists you in operations such as offset, breakpoint and block calibrations. With curve and map editors, you can display the current process point of the engine.


The following editors are available: Calibration Window (numeric, slider or bit display), Calibration Table, Scalar Table Editor, Combined Editor.

All editors have one basic property in common: Since it is basically not possible to edit the values on the reference page, the editors change their background color to indicate whether the working or the reference page is currently being displayed. Differences in the values between the working dataset and reference dataset are also shown in the editors.

Switch between Working and Reference Page

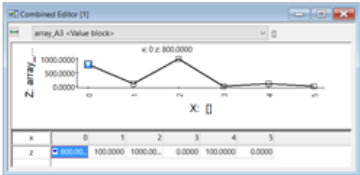
Perform the Calibration

**Hardware → Switch between working/reference page**  
 or  
 Enable WP or RP button.



Select the calibration instrument (e.g. Combined Editor window) and right-click to select the desired command to modify the variable values  
 or  
 Change the value of the calibration variable directly in the calibration instrument.

The changes to the calibration elements are automatically saved as soon as you make them.



x	0	1	2	3	4	5
z	100.0000	1000.0000	0.0000	100.0000	0.0000	0.0000

There are different calibration commands depending on the selected editor.

**Fig. 2-17:** Performing calibrations in the editors

The changes made in the editors become immediately effective.

## 2.4.8 Saving Datasets

INCA provides several means to save the current program version. To make your modified data available to other users, you can save the working dataset as a new HEX file in the file system. In INCA, you can save the current working dataset under a different name, to backup an intermediate version. You can also copy the current data of the working dataset to a new reference dataset. The current reference dataset is write-protected and therefore cannot be replaced directly. Instead, you are prompted to enter a name for the new reference dataset.

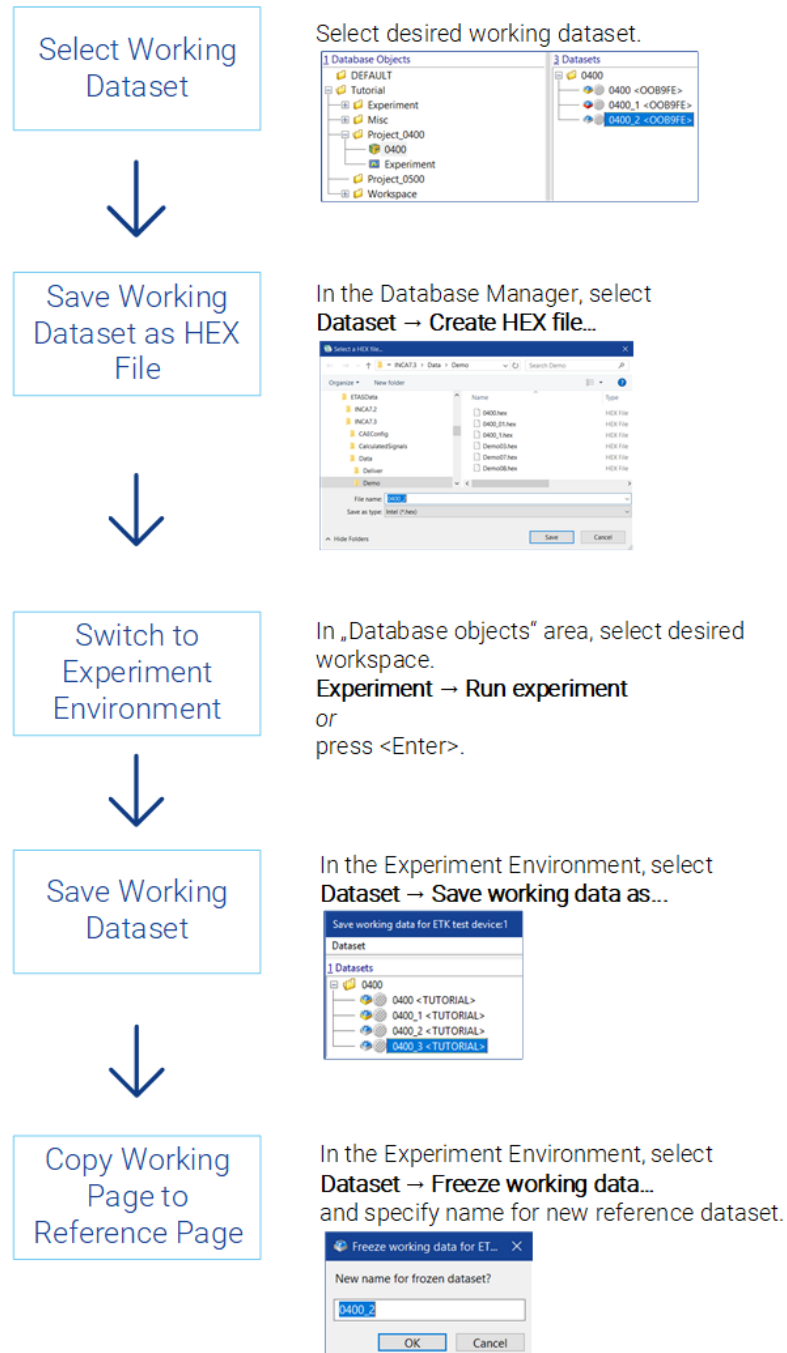


Fig. 2-18: Saving datasets

## 2.5 Editing Datasets with the Calibration Data Manager

You can use the Calibration Data Manager (CDM) to analyze the datasets generated while running the test series. For a cross-project analysis, you select datasets from different projects. When selecting datasets of a project, you can analyze the reference and working dataset. You can list, compare and copy the data of the selected datasets.

### 2.5.1 Selecting Datasets and Calibration Variables

Before you can execute one of the **List**, **Copy** or **Compare** actions, you must first select a source dataset. For the **Copy** and **Compare** actions, you must select at least one more destination dataset. Selecting the calibration variables is required for each of the three actions. The CDM offers you various selection methods.

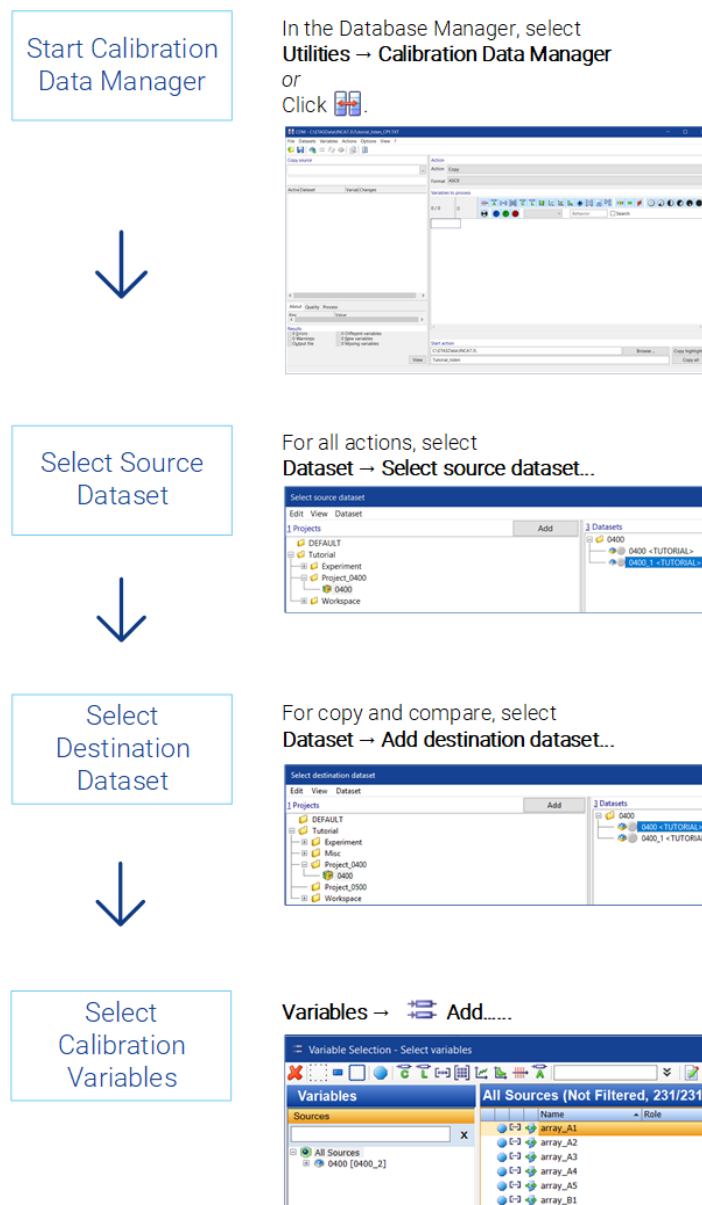


Fig. 2-19: Selecting the datasets and calibration variables

### 2.5.2 Executing the List, Compare, or Copy Action

The **List** action creates a table listing the data of the selected calibration variables in the source dataset. The **Compare** action compares the selected calibration variables from the source dataset with the comparison datasets and provides the different values for evaluation. The **Copy** action copies the selected calibration variables from the source dataset into the activated destination datasets. Only the **List** action provides you with data exchange formats (CVX, DCM, CDF V2.0 or PaCo) in addition to the ASCII, HTML or PDF output formats.

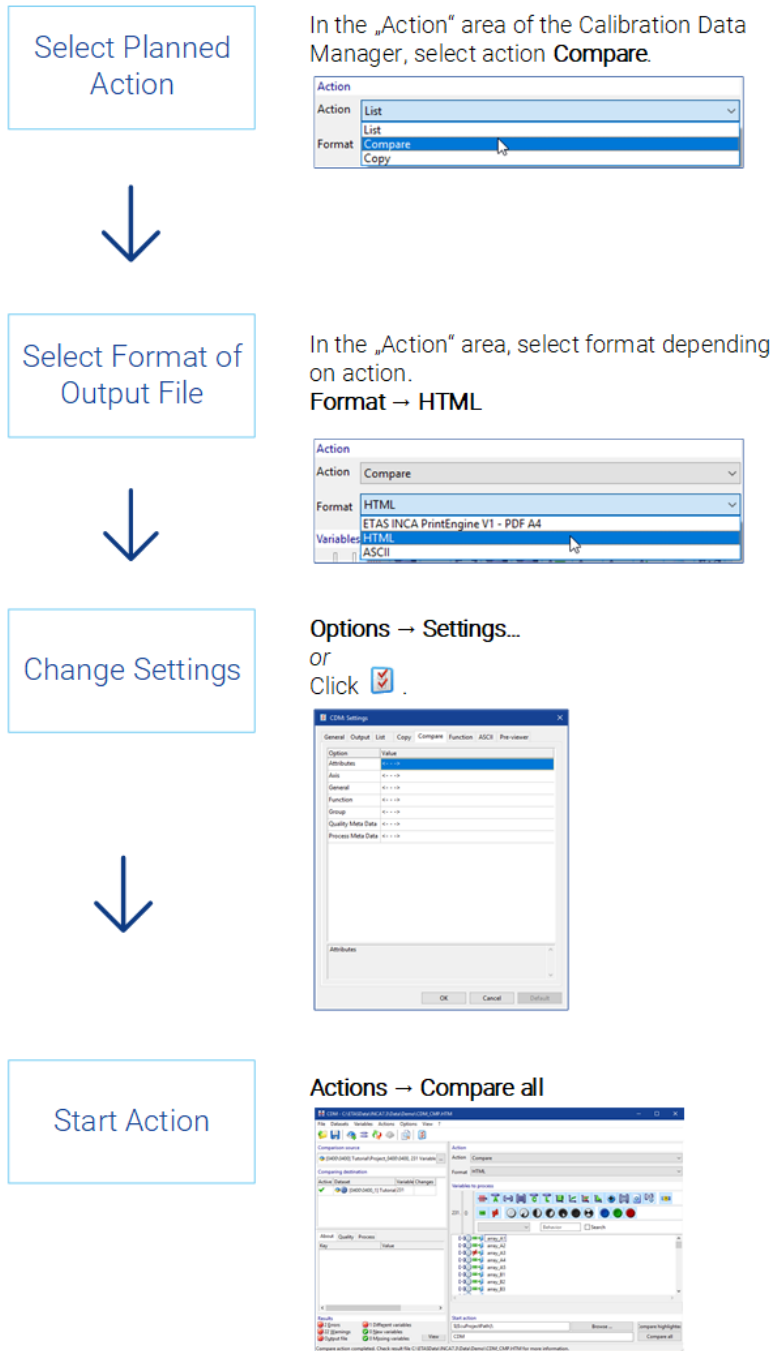


Fig. 2-20: Listing, comparing, copying

### 2.5.3 Analyzing Results and Output Files

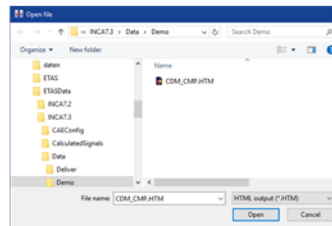
Depending on the operation carried out, different result or output files are generated. The action selected automatically determines the file extension of the output files, which can be \*.txt, \*.htm, \*.pdf, \*.csv, \*.dcm, \*.cdfx, or \*.xml. Apart from the standard output files \*\_1st.\*, \*\_cpy.\*, and \*\_cmp.\*, INCA may also generate additional files (\*.lab). The result files (\*.lab) may contain various information items such as errors, differences or simple lists of calibration variables that can be used to select the calibration variables. Any errors that may have occurred during the action, are indicated in the "Results" field (red fields).

View Result Files



View Output Files

In the Calibration Data Manager, select **Actions → View result files...**



Open the result file of the last action performed.

**Actions → View output files...**

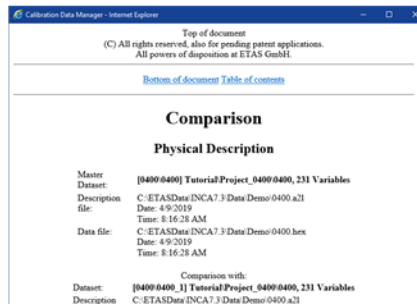


Fig. 2-21: Analyzing the result and output files

## 3 INCA - Basic Operations

This chapter provides information on the window and menu structures, control options using the mouse and the keyboard, and the help system.

Be sure to read this chapter since some operational procedures are described only in this section. Although the presented techniques are standard Windows operations, they may not be familiar to the less experienced Windows user. They are therefore described here as a central summary.

Since INCA is mostly controlled from within the vehicle, the design of INCA put special emphasis on a simple and comprehensive operation by keyboard commands. For special features and deviations from the Windows keyboard conventions, refer to chapter "Operations Using the Keyboard" à la page suivante.

### 3.1 Window Structure

The INCA Window Elements:

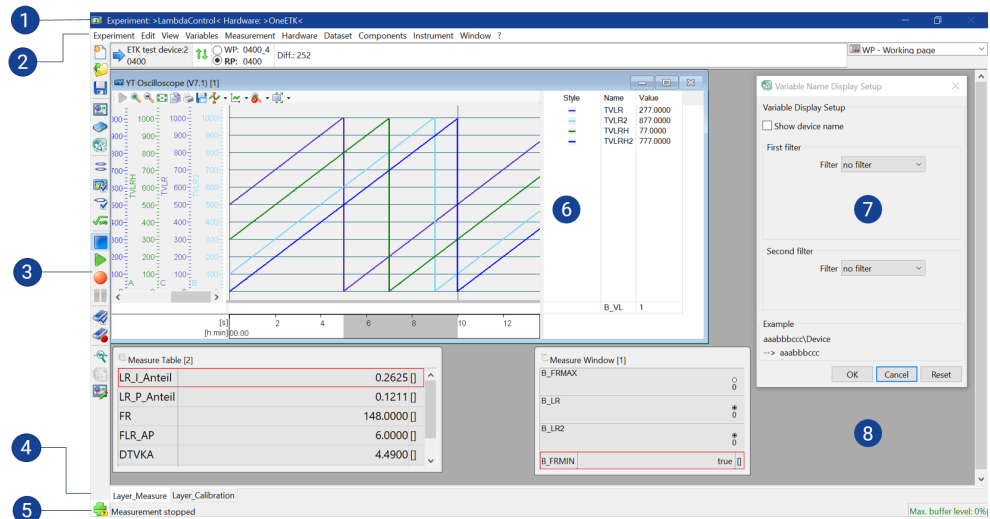


Fig. 3-1: INCA Window Elements

1. Title bar
2. Menu bar
3. Toolbar
4. Layer tabs
5. Status bar
6. Instrument
7. Dialog box
8. Window area

## 3.2 Toolbars

The most common commands are also available as buttons in a toolbar. In this way, a command can be executed simply with a click of the mouse.

All buttons located on the toolbar are mouse-sensitive. If you place the mouse cursor on a button and hold it for one second, a tooltip box is displayed right next to the selected button which displays the button function.



Fig. 3-2: Toolbar



### Note

All commands that can be issued by clicking the specific buttons are also available in the corresponding menus.

## 3.3 Operations Using the Keyboard

Since most of the control unit software is calibrated in the vehicle, user-friendliness was of utmost importance in the INCA development process. Individual keys are preferred over the function keys <F1> to <F12>, which in turn take precedence over key combinations using <CTRL> and <ALT>. You can display an overview of the keyboard shortcuts currently used at any time by pressing <CTRL> + <F1> in INCA or find a complete list of the keyboard shortcuts in the Manuals folder.

### *Keyboard Operation Compliant With WINDOWS® Conventions*

The general operation of INCA, such as navigating in menus or activating a certain window, complies with the WINDOWS® conventions.

Pressing the underlined letter in a menu while holding down the <ALT> key activates the corresponding command. You can activate a subordinate menu command by pressing the underlined letter together with the <Shift> key.

For example, to open the **Edit** menu with a keyboard command, press the <ALT> + <E> key combination.

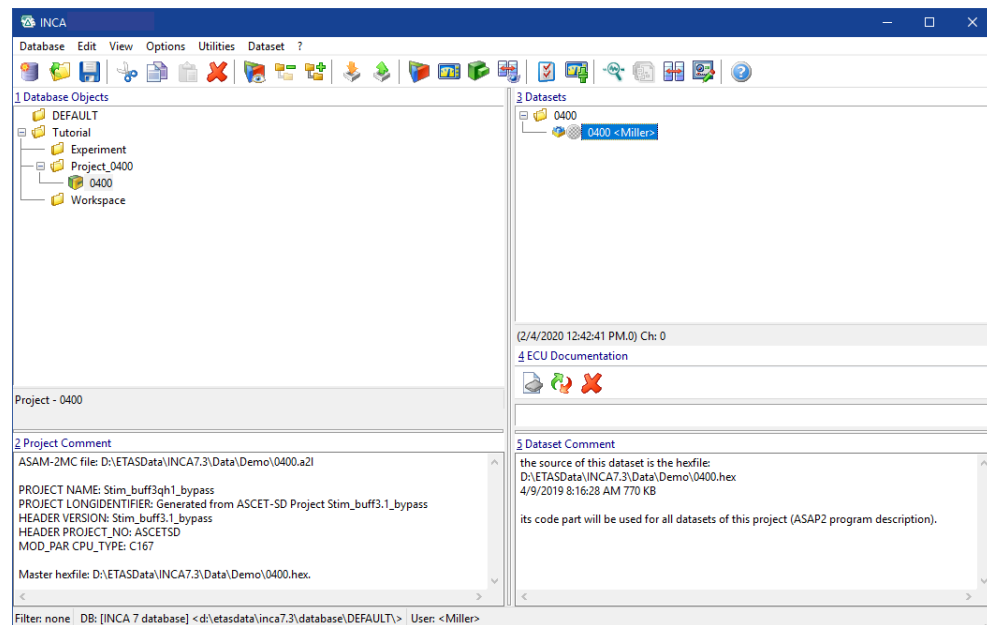
To switch to the next window or list box within the working windows, press the <TAB> key (in the order from top left to bottom right). You can use the underlined letter of the field or list label, while holding down the <ALT> key, to switch to the corresponding field or list box.

The arrow keys allow you to skip to the next item in list boxes. You can select multiple items by making your selection while pressing the <SHIFT> key.

Compliant with the WINDOWS® conventions, you can switch between the windows of different applications using the <ALT> + <TAB> key combination. In this sense, all subsystems of INCA, such as the Experiment Environment or UI Developer, are treated as individual applications.

Switching between the individual working windows of an application, e.g., in INCA between individual measure and calibration windows within the Experiment Environment, is done—pursuant to the Windows convention—by pressing the <CTRL> + <TAB> key combination.

Within the working window, you can use the underlined number of the field or list label, while holding down the <ALT> key, to switch to the corresponding list box. For example, <ALT> + <3> would activate the "3 Datasets" list box.



**Fig. 3-3:** Select project and working data for ETK test device

### 3.4 Operations Using the Mouse

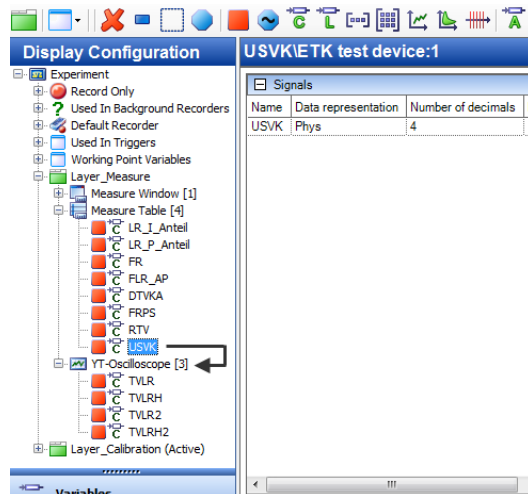
In the office or lab, you can use the mouse to operate INCA in a user-friendly way. The use of the mouse corresponds to WINDOWS® conventions.

You can select multiple items by making your selection while pressing the <SHIFT> or <CTRL> key.

Right-clicking on the window elements opens context-sensitive pop-up (short-cut) menus.

#### *Drag & Drop*

You can use Drag & Drop to move experiment elements in the Display configuration. For instance, if you want to move a variable from one window to another, just left-click on the variables, keep the button depressed and move the variable with the mouse to the target window.



It is just as easy to move a variable from one window to another (left-click on the variables, keep the button depressed and move the variable with the mouse to the target window).

### 3.5 Hierarchy Trees

INCA often displays information such as the contents of a database in a hierarchical tree structure. In order to see all branches and the entire contents of such a tree structure, you have to expand or collapse the branches. INCA allows you to expand several branches automatically or to expand specific partial trees.

#### To expand several branches automatically

1. To expand or collapse all branches at once, you can use the **View > Expand** all or **View > Collapse** all menu option or the buttons in the toolbar (see "Toolbars" à la page 38).

#### To expand individual partial trees

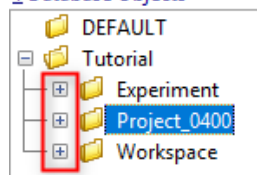
1. To open the desired branch, click with the mouse on the little "+" box next to this item or press the <+> key. Clicking a second time on the same little box or pressing the <-> key collapses the branch again.

or

Move the focus to the item with the pUP/DOWN ARROW > key and then press the <RIGHT ARROW > key to expand the branch. Collapse the branch by pressing the <LEFT ARROW > key.

A "+" box indicates a branch that can be expanded.

#### 1 Database Objects



A "-" box indicates an expanded branch. If there is no box, this is the end of the structure. Expansion is not possible.

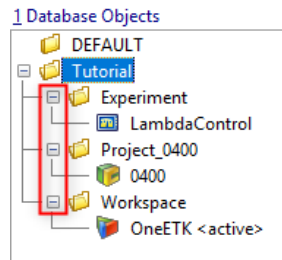


Fig. 3-4: Hierarchy trees

### 3.6 Editable Tables

Tables can be used to quickly modify the numerous INCA configurations which are often quite complex. You can directly change the items in the right column of these tables after selecting them with the mouse or the keyboard.

Depending on the type of items, you will use one of the following methods for editing:

- direct keyboard entry
- selection from a list box
- entry into a dialog box

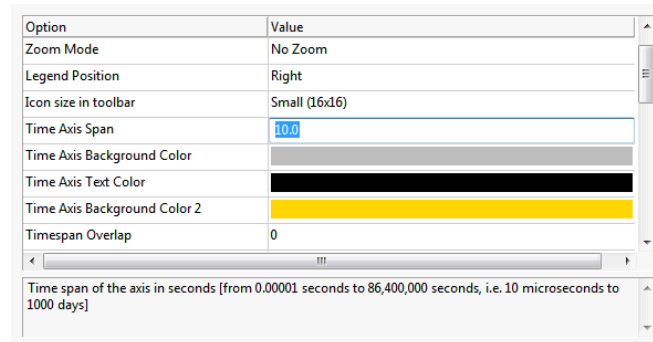
#### To modify items by overwriting

1. Click on the item you want to modify.

*or*

Use the arrow keys to move to the item.

2. Enter the new value using the keyboard.



3. Confirm your entry by pressing <ENTER>.

#### To select items from a list box

1. Use the mouse to select the item you want to modify and click again on the item.

*or*

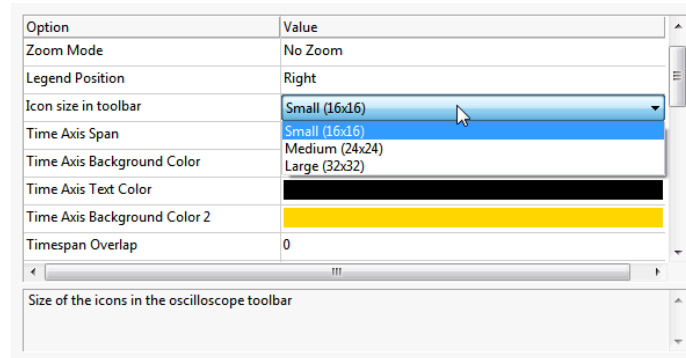
Use the arrow keys to move to the item and press the <F2> key.

A list box with several items appears.

2. Select the desired item in the box.

*or*

Use the arrow key to move to the item.



3. Confirm your selection by pressing <ENTER>.

#### To modify items via dialog box

1. Click on the item you want to modify.

or

Use the arrow keys to move to the item and press the <F2> key.

A dialog box opens.

2. In the new dialog box, make your selections.
3. Confirm your input by pressing <ENTER> or click the **OK** button.

### 3.7 Sounds for INCA Events

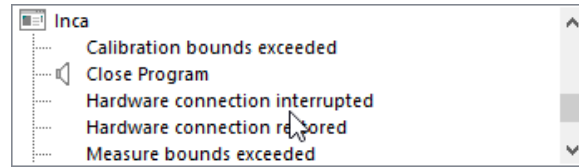
INCA can play sound notifications in case of certain events, e.g. when the hardware connection is disrupted or hard limits of measurement variables are exceeded during measurement.

Sounds can be configured for the following event types:

- Calibration bounds exceeded
- Close program
- Hardware connection interrupted
- Hardware connection restored
- Measure bounds exceeded
- Open program
- Recording finished
- Recording started
- Trigger occurred

#### To configure sounds from the selected events


1. Go to the Windows Control Panel.
2. Double-click on the entry **Hardware and Sound**.
3. In the dialog box "Hardware and Sound", go to the "Sounds" tab.  
The "Sound" dialog box opens.
4. In the list box, scroll down to the INCA items.



5. Select an event type you want to assign a sound to.
6. Select a sound from the "Sounds" list box.

**Note**

You can also use customized sounds. To assign a customized sound clip, click the **Browse** button and select the corresponding wave file (\*.wav).

7. The event is marked with an icon  indicating that a sound is assigned.
8. Click **OK** to close the dialog and save the assignment of sounds.


### 3.8 Storage Directories

INCA stores various data in files during operation. The following is a list of the most important files and their default storage directories:

- Calibration Log  
ETASData\INCA7.6\CalibrationSessionLog
- Import and export files (\*.exp64)  
ETASData\INCA7.6\Export
- Databases  
ETASData\INCA7.6\Database
- HEX files (\*.hex)  
ETASData\INCA7.6\Data\Demo
- Project files (project description files, Hex files and ECU documentation files)  
ETASData\INCA7.6\Data\Demo
- Protocol of configuration  
ETASData\INCA7.6\Data\ConfigurationProtocol
- measure files (\*.dat)  
ETASData\INCA7.6\Measure
- Custom user interfaces (\*.vui)  
ETASData\INCA7.6\Data\Screen Layouts

You can use the **User Options** menu option to change the default storage directories of the various file types.

### To change the default storage directories

1. In the INCA Database Manager menu, select **Options > User options...** or click .

The "User options" dialog boxes opens.

2. Select the "Path" tab.
3. In the table, click on the "Value" entry you want to change.

The "Directories" dialog box opens.

4. Specify the directory that you want to use as the default directory for the selected data type.
5. Click **OK**.
6. Repeat these steps for each entry you want to change.

#### **Note**

The **New** button allows you to create new sub directories to the selected top directory.

7. When you are done, click **OK** in the first dialog box to accept the changes, or click **Cancel** if you do not want to accept the changes.

## 3.9 Help Functions

### 3.9.1 INCA Help Functions

The INCA help provides detailed information about INCA, INCA add-ons, and hardware configurations.

From any INCA module, select **? > Help** to open the INCA help or press <F1> to get context-sensitive help on the currently open working window.

Browser settings may differ from computer to computer, and therefore the behavior of the various browsers may vary when using the INCA help. For example, some browsers may fail to display the INCA Help correctly.





If you have trouble displaying the INCA help, try the following:

1. Check if the cookie setting of the browser is enabled.
2. Open the INCA help with another browser, for example, Mozilla Firefox.

The INCA help HTM files are located in the following folder: %ProgramFiles%\ETAS\INCA7.6\Help

### 3.9.1.1 Using INCA Help Buttons

---

	<b>Expand all</b> Expands all drop-down menus in the INCA Help main window.
	<b>Collapse all</b> Collapses all drop-down menus in the INCA Help main window.
	<b>Remove highlights</b> Removes the highlighting in the INCA help main window. These are only displayed if you have previously performed a search via the search field and opened a corresponding topic.
	<b>Print</b> Prints the content of the INCA Help main window.

---

### 3.9.1.2 Search Methods and Syntax

In the INCA Help search field, you have several options for specifying your search:

#### Full-Text Search

The INCA help scans all of the target content and creates a database index for search. When you enter a search query, the search engine in the index retrieves a list of topics with matches. The INCA help search results include these matches:

- Case-insensitive matches:  
Search is not case-sensitive. For example, a search for the word "run" will find matches for "Run" and "run."
- Matches with variant endings:  
For example, a search for the word "run" will also find matches for words such as, "runner," "running," and "runs." Since matches are not case-sensitive, the results will include topics containing matches such as, "Runner," "Running," and "Runs."

#### Phrase Search

You can search for phrases by enclosing their search terms in quotation marks. This is useful when you want to restrict a search to locate terms that appear in an exact order. For example, you might want to search for a phrase such as "measure variable".

#### Boolean Operators

The INCA help search supports boolean operators, such as AND, OR, NOT, and ().

- Instead of the word AND, type the plus symbol + or the ampersand symbol &.
- Instead of the word OR, type the pipe symbol |.

- Use the NOT operator to find topics that contain one term but not the other. Type a term, followed by the carat symbol ^, and then type another term.
- Use parentheses to combine boolean phrases and search terms. Type parentheses ( ) around the combined terms.

### 3.9.2 Manual and Tutorial

Unless otherwise stated, additional documents are provided with the basic INCA installation and can be found in one of the INCA folders "Manual" or "Help".

Further documents are provided with INCA add-on products.

The complete INCA manual as well as a tutorial with exercises is available in digital version and can be opened on the screen at any time. Using the index, full text search, and hypertext links, you can find references fast and conveniently. You can open the INCA manuals and tutorial via the help menu **? > Manuals and Tutorial** or via the INCA program group.

Together with INCA, the INCA tutorial document is also installed. +The tutorial is available as PDF file and can be called up from INCA via the menu item **? > Manuals and Tutorials**. The tutorial is mainly intended for INCA beginners. On the basis of examples you will learn the operation of INCA. The total contents are divided up into short progressively structured teaching units. Before you start on the tutorial, we recommend studying the chapter "[INCA - Concept](#)" à la page 48.

### 3.9.3 Video Tutorials

INCA offers video tutorials for presenting complex or new features and use cases which can best be shown by animations. The video tutorials are either linked from the INCA help or from an overview page which lists all available videos.

You can open the overview of the INCA video tutorial via the help menu (**? > Video Tutorials**).

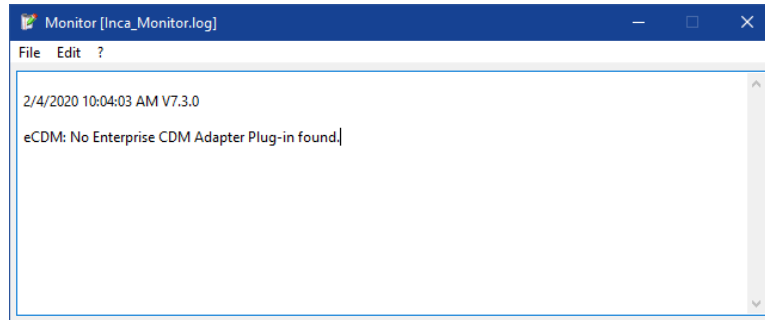
Subjects which can easily be shown instead of being described in words are presented as video tutorials. The videos are installed locally on your PC if you selected the installation of video tutorials during installation. They are available as MP4 files and can be called up from INCA via the menu item **? > Video Tutorials**. If you did not install them on your PC, you can either view them in the ETAS channel on YouTube, or you can download them from the ETAS Download Center.

### 3.9.4 Keyboard Assignment

You can display a complete overview of the keyboard shortcuts currently used at any time by pressing **<CTRL> + <F1>**.

### 3.9.5 Monitor Window

The monitor window is used to log the working steps performed by INCA. All actions, including errors and notifications, are logged. As soon as an event is logged, the monitor window is displayed in the foreground.



In addition to displaying information, the monitor window also provides the functionality of an editor.

- The display field of the monitor window can be freely edited. This way, your own remarks and comments can be added to the INCA messages.
- The INCA messages can be saved as text files along with your comments.
- Other INCA text files already stored can be loaded so that you can compare specific working steps.

### 3.9.6 New Features in INCA

Find an overview of the new features "What's New" in the following locations:

- Readme folder of your INCA installation
- INCA help

To open the "What's New in INCA" chapter in the INCA help

1. Open INCA and click **F1**.  
The INCA help window opens in your standard browser.
- Select the "What's New" chapter from the left column.

### 3.9.7 Troubleshooting

In the INCA help, you find information on what to do if you encounter problems while working with INCA.

To open the "Troubleshooting" chapter in the help

1. Open INCA and click **F1**.  
The INCA help window opens in your standard browser.
2. Select the "Troubleshooting" chapter from the left column.

## 4 INCA - Concept

This section provides an introduction to the calibration work and describes the use, structure, and functionality of INCA.

### 4.1 General Description

A variety of physical variables and factors determine a vehicle's overall behavior, which results in a large number of control parameters in the control unit software. The objective in calibrating is to find the optimal values for these parameters.

INCA lets you read measured data from the control unit and the engine in parallel. The program helps you determine measured engine data such as lambda, different temperatures and voltage values, etc. With INCA, you don't just get a tool that will adapt to a variety of different control units, but also a system that will optimize a wide range of different vehicle components.

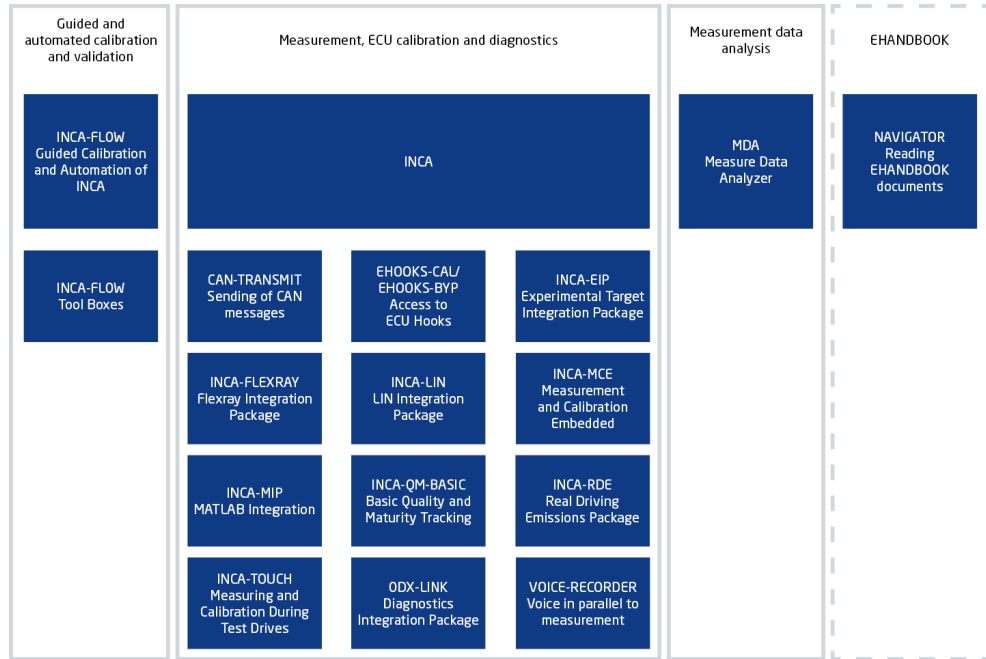
INCA is an "open system." Consistent implementation of the ASAM-MCD standard and support for data exchange formats that are established in this environment allow this program to be used for any ECU interfaces (can be customized to any manufacturer's control units) and to be integrated in existing data processing infrastructures.

INCA's modular configuration makes it possible to adapt to customer-specific requirements. The basic software provides the outer framework, which can be expanded using exactly and only those additional components that you actually need. Moreover, a number of add-on modules for specific use cases is available (e.g. INCA-QM-BASIC, INCA-FLEXRAY, INCA-LIN).

In addition to that, INCA offers open interfaces which allow for the adaptation of INCA core capabilities as well as the remote control of INCA by other applications.

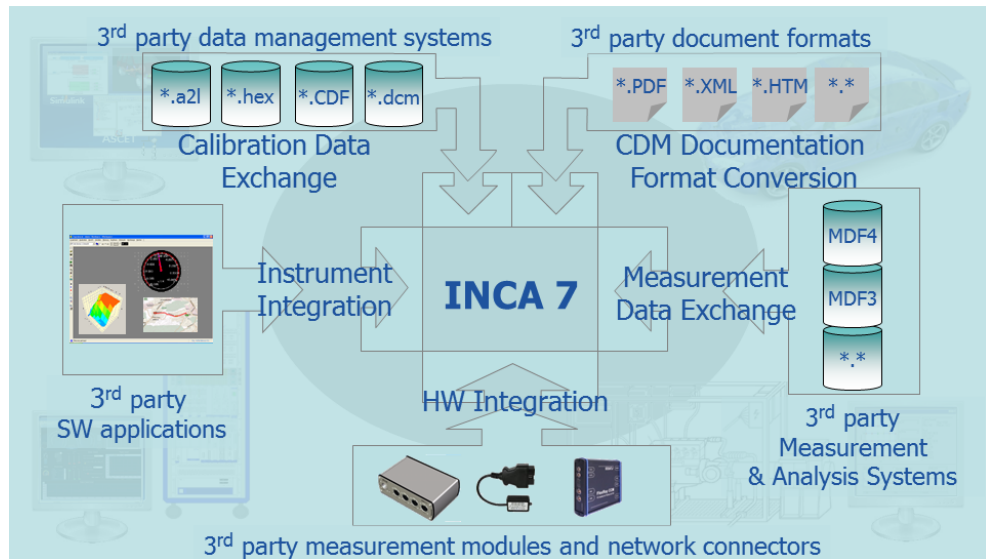
#### 4.1.1 System Overview

INCA consists of a measurement and calibration core system which can be enhanced by various add-ons and customized extensions (e.g. INCA-MIP, INCA-QM-BASIC, INCA-FLEXRAY) that can be integrated in INCA. Further information is available on request.



### 4.1.2 Open Interfaces

In addition to automation system interfaces, the tools of the INCA product family provide a wide range of open and standardized interfaces for configuration, data exchange, documentation, automation, and integration with customer applications.



- INS.DK: Interface for the integration of instruments into the INCA Experiment Environment

INCA's open Experiment Environment provides an interface which allows for the integration of customer specific measure and calibration instruments for the most diverse use cases. Examples are animations, integrated web explorers, embedded model viewers, street map viewers visualizing measured courses of GPS coordinates, stimulation of

parameters with generators, etc. As a consequence, customer specific data visualizations or data post processing does no longer require external applications connected via COM, MIP or ASAP3, but may be done directly in INCA.

Instrument Integration is supported through the INCA Instrument Integration Development Kit (INCA-INS.DK).

- MDF.IP: Library for measurement data exchange in standardized formats  
Measure data that are recorded by INCA are written in measure data formats. For this purpose, INCA uses a component that is also available for 3rd party and inhouse tool providers to read and write measure data in standardized MDF4 format as well as in the formats MDF 3.0, MDF 3.3 and ETAS ASCII. Moreover, additional customized measure file formats can be added through a plug-in interface.

The MDF data exchange component and plug-in interface is provided through the MDF Integration Package.

- HWI.DK: Interface for hardware integration  
INCA supports ETAS measure and calibration hardware. Moreover, it offers an open interface which enables 3rd party tool providers to add support of their hardware.

Hardware Integration is supported through the INCA Hardware Integration Development Kit and the ECU and Bus Interfaces Integration Package.

- eCDM: Interface for the exchange of enterprise calibration data  
INCA's Enterprise CDM (eCDM) interface provides direct access to any enterprise calibration data management system from within INCA via a standardized API, thus making a manual data exchange via the file share obsolete.  
eCDM functionality is provided through a separate eCDM add-on.
- Plug-in interface for providing additional CDM documentation formats  
INCA CDM offers a number of output formats for the documentation of your calibration activities, e.g. ASCII, DCM, or HTML. By means of plug-ins, it is possible to add further converters supporting user-specific documentation formats, e.g. customized PDF output.

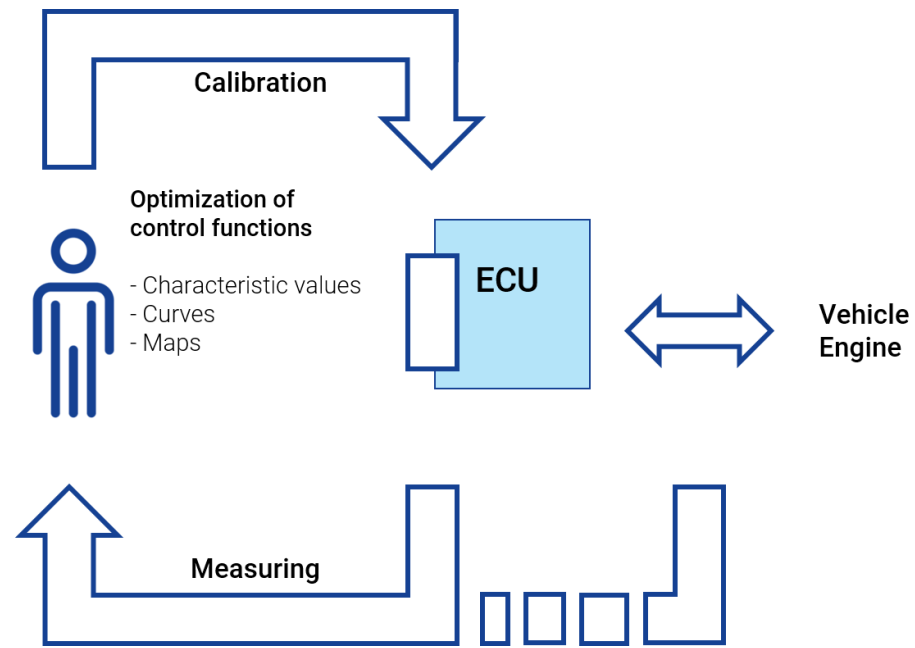
CDM provides the complete calibration data information in the standardized MSRSW ASAM XML intermediate format for further processing by the converters.

The plug-in interface is available with the INCA base package.

## 4.2 Calibration Basics

The implementation of complex functions and control algorithms in micro-processor-based control units, and the tuning and optimization of these systems for various engine and vehicle types is referred to as "calibration". INCA aides you throughout this process, from setting up the hardware to evaluating the

measurement results and managing the measured data. The objective of the tuning procedure is the vehicle-specific adaptation and optimization of the control functions of an electronic control unit. The control function algorithms are permanently stored in the control unit program; only the parameter values (maps, curves, and characteristic values) can be modified.



**Fig. 4-1:** Calibration Basics

Information on the interfaces connecting to the control unit (hardware and software), the measurement values and parameters as well as the technical measurement capabilities of the measurement devices used (sampling rate, accuracy, etc.) are required to operate a calibration system.

During operation, the user interface software utilizes this information to carry out the actions desired by the user and to represent the parameters and measured signals accordingly.

The following steps need to be carried out to measure the vehicle speed from a control unit:

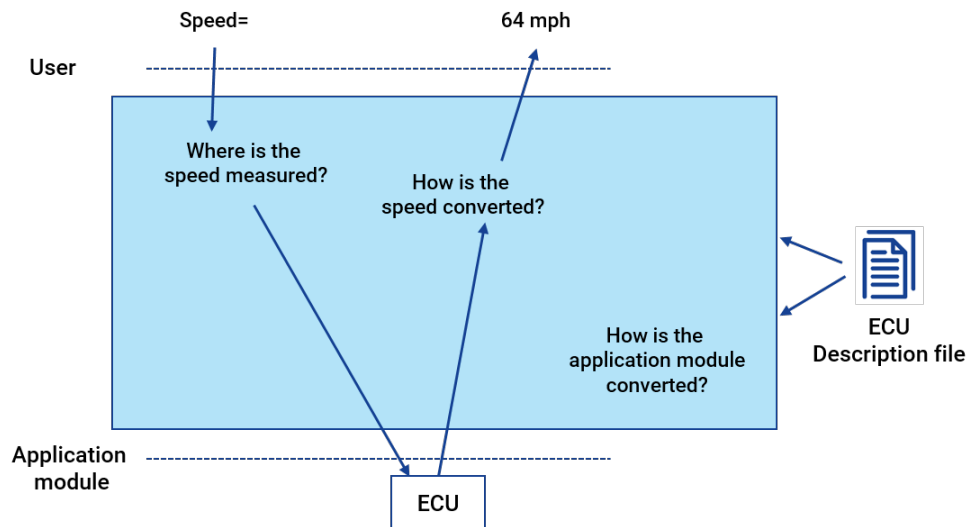


Fig. 4-2: Measure the vehicle speed from a control unit

- Defining and parameterizing the control unit interface. How is the communication with the control unit implemented and what kind of parameters have to be used (baud rate, etc.)?
- Selecting the measured signal and communicating the hex address to be measured to the control unit. The control unit provides the measured value associated with this address.
- Representing the value provided by the control unit in physical units using a hex → physical conversion rule.

The calibration system must have address and conversion information for all measured signals and adjustable values as well as parameters which describe the control unit interface. Therefore, a standardized description file must be available for each new program running in the control unit.

Thus, each new software delivery includes at least one description file (\*.a21) and one program file (\*.hex).

#### 4.2.1 Standard Interfaces Based on the ASAM-MCD-Model

Standardized interfaces, which are independent of manufacturers and systems, have been established in the calibration tool environment. The work committee ("Arbeitskreis") for the Standardization of Automation and Measurement systems, with the work groups: Measurement, Calibration and Diagnosis, abbreviated ASAM-MCD, has standardized, among others, the following crucial calibration interfaces:

**ASAM-MCD-1** provides the direct interface to the control unit.

##### – ASAM-MCD-1a

direct hardware interface to the control unit; hardware interface is the CAN bus (Controller Area Network) with CCP (Can Calibration Protocol) as the communication protocol.

– **ASAM-MCD-1b**

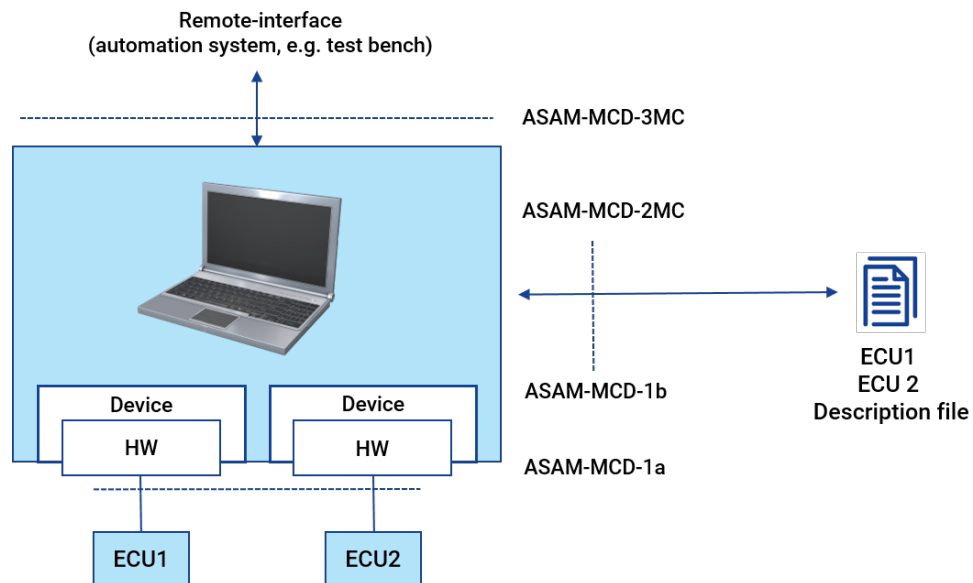
Software/driver interface between the calibration program on the PC and the connected calibration hardware.

**ASAM-MCD-2MC:**

This is a file format used to describe the calibration variables and measured signals contained in the control unit software, and additional specific information designed to parameterize the calibration interface. You can use ASAM-MCD-2MC to import the information required for this into a calibration system (A2L file). The file only contains address information and the data structure, however; the values are stored in the associated hex file.

**ASAM-MCD-3MC:**

This is a standard data exchange/communication protocol for remote control of a calibration system via Ethernet or the RS-232 interface. A test bench computer can work as the master to automatically optimize control unit parameters via the ASAM-MCD-3MC interface. For this purpose, the calibration system receives calibration commands via the interface and executes them in the control unit. When requested, control unit internal signals can also be measured and returned to the test bench computer.



**Fig. 4-3:** Standard data exchange/communication protocol

#### 4.2.2 Interfaces to the Control Unit- ASAM-MCD-1

The connection of a calibration system to a control unit can be parallel or serial.

- Parallel interface via ETK (emulator test probe) - the ETK is connected to all data and address bus lines in the control unit in parallel.
- Serial Interface

The control unit communicates with the application device via a standard serial interface and controls the engine, the gearbox or the brake system at the same time; a special application protocol is used for communication:

- CAN interface using CCP (CAN Calibration Protocol) as the communication protocol.
- XCP (eXtended Calibration Protocol) via Ethernet or USB.

Additional memory (RAM) is required in the control unit, in which the calibration system performs the parameter calibration and from which the control unit reads the control parameters, both for the serial and the parallel application. In the case of the parallel application, this additional memory is implemented on the ETK.

ETK and CAN differ in terms of performance when it comes to downloading data to the control unit and in terms of data acquisition.

**High-End:** ETK, very fast downloading of program and calibration data to the control unit and very fast data acquisition.

**Medium and Low-End:** CAN using CCP with KWP2000, depending on the control unit implementation.

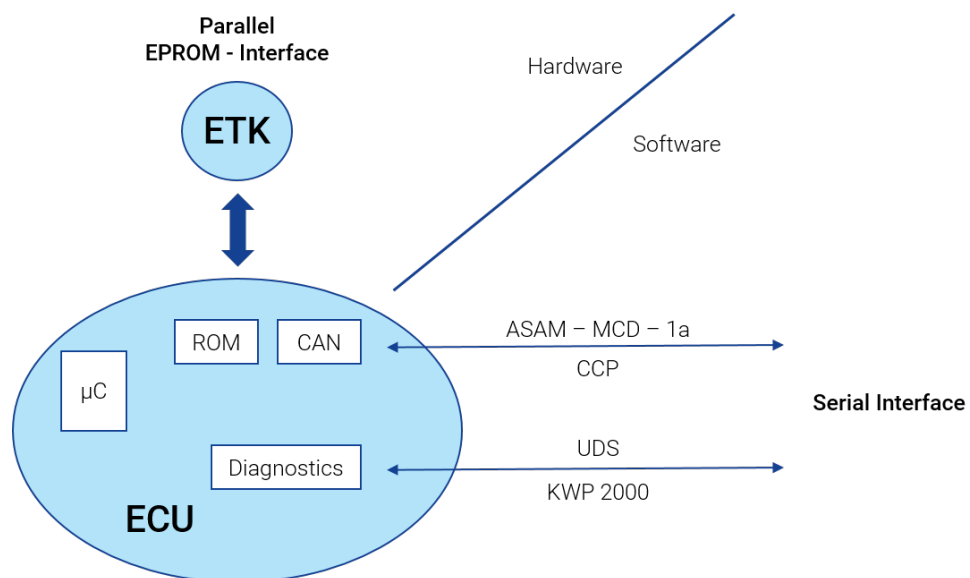


Fig. 4-4: Interface to the Control Unit

#### 4.2.3 Calibration with ETK (Parallel Control Unit Interface)

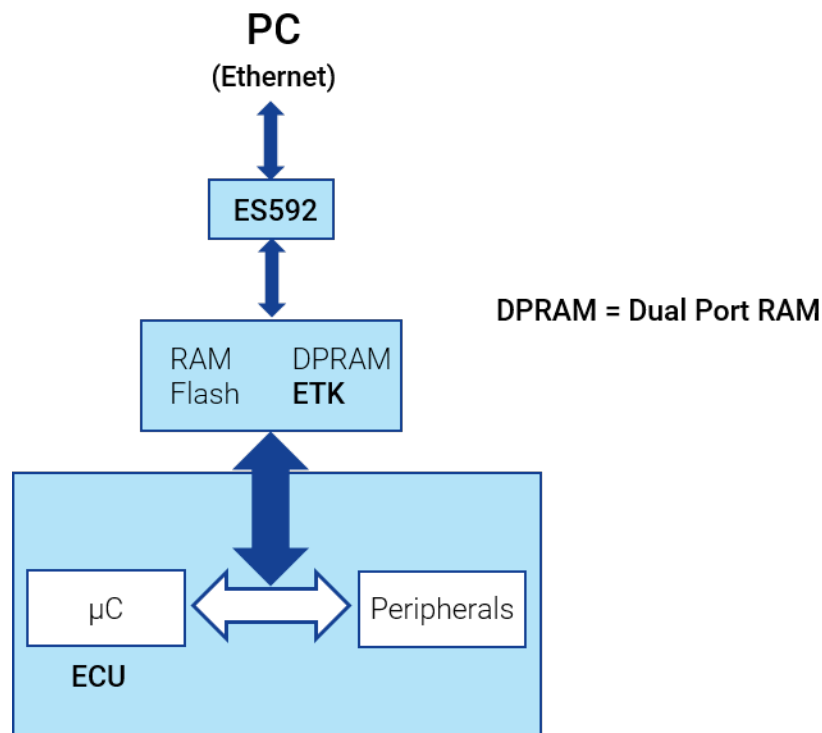
An emulator test probe (ETK) is installed in the control unit for the parallel application. The ETK is connected to all bus lines of the control unit processor and replaces either the control unit program and the calibration data or the calibration data only – depending on the control unit. Consequently, the calibration system can change control parameters online. Furthermore, the ETK provides RAM (so-called DPR = Dual Port RAM) to transfer measured data from the control unit to the calibration system. The control unit itself cannot recognize whether the program runs from the control unit flash (this is where the program is stored in the production control unit) or the ETK.

Only approximately 30 lines of additional program code are required for data acquisition via a table transferred from the calibration system to the control unit. The additional computing time required for this is negligible.

The ETK allows measured data to be acquired simultaneously from three measurement rasters (loops), which are also speed-synchronous. Using an interference-proof serial interface, the control parameters can be modified or read via calibration devices, see "[Measurement and Calibration Hardware](#)" à la page 59.

The ETK is the control unit-specific component of the memory emulation. Like the INCA PC software, all other calibration devices and cables used with the parallel application are the same for all projects. Depending on the micro-controller bus, clock frequency, memory size, etc., the ETK is available in different variants and will be adapted to the respective variant using programming on site. In addition to RAM, there is also an ETK flash on the ETK board. This is used to store the calibrated data on the ETK. This way, the control unit can be run immediately upon start and a calibration system for downloading the data is not required.

Don't confuse the ETK flash with the control unit flash. The ETK replaces the control unit flash (perhaps only partially, depending on the type) and provides a flash of its own which is used for data backup purposes.

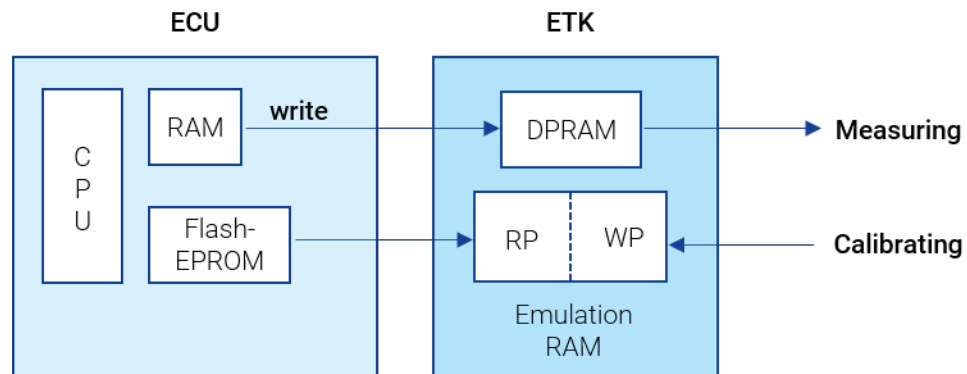


**Fig. 4-5:** Calibration with ETK

The ETK can be installed on or in the control unit, so that the ETK and the control unit form a compact unit. This results in simple and interference-proof designs and considerably facilitates their handling.

ETKs are available in a large number of variants (physical dimensions, connections) for the most common micro-controllers. Typically, they are connected to the control unit module via specific adapters.

The functional principle of the ETK allows control parameters to be edited flexibly in the ETK RAM while the process is running.



**Fig. 4-6:** Calibration with ETK 2

In the ETK, the data to be applied is redundant, so that, for example, the response of an engine to modified data can be compared quickly to the response to the original data at the push of a button. One of the pages is called the Working Page (WP) because the calibration system can carry out changes on this page. The other page is the Reference Page (RP). The data on the reference page cannot be edited.

There are control units in which the ETK only manages the calibration data. Other control units also allow access to the code page so that the entire control unit program can be loaded into the ETK.

If only data are emulated in the ETK, the control unit accesses the data on the working page or reference page and reads the program from its own FLASH EPROM. In this case, a new program version first requires the control unit FLASH to be programmed and then data to be downloaded to the WP and RP.

Due to its parallel program and data access, fast downloading of data into the control unit and very fast data acquisition with negligible run-time load, the ETK interface is the most powerful interface.

#### 4.2.4 Calibration via the Serial Interface

Calibration via CAN, LIN, and FlexRay is implemented via communication protocols which must be available both in the control unit and the calibration device so that they can be used together.

The serial application is particularly important in situations, where due to the installation location of the control unit (e.g. directly in the gearbox or when using hybrid control units), the use of an ETK is not possible. A major benefit of the serial concepts is that little hardware is required, so that in most cases there is no need for mechanical and physical modifications. In most cases, the measured data rates are sufficient for a production intent application. INCA supports serial application via the following interfaces and protocols:

Protocol Interface	CCP <sup>a</sup>	XCP <sup>b</sup>	KWP2000 <sup>c</sup>	UDS <sup>d</sup>	Monitoring
CAN	X	X	X	X	X
FlexRay		X			X
LIN					X
Ethernet		X			X <sup>e</sup>
USB		X			

<sup>a</sup> CCP (CAN Calibration Protocol): protocol standardized on the CAN bus in accordance with ASAM-MCD-1a.

<sup>b</sup> XCP (eXtended Calibration Protocol): vendor- and interface-independent protocol for communication between calibration tools and control units, in accordance with ASAM-MCD-1.

<sup>c</sup> KWP2000 (Keyword Protocol 2000): communications protocol used for on-board vehicle diagnostic systems, specified in ISO 14230.

<sup>d</sup> UDS (Unified Diagnostic Services): diagnostic protocol, specified in ISO 14229-1.

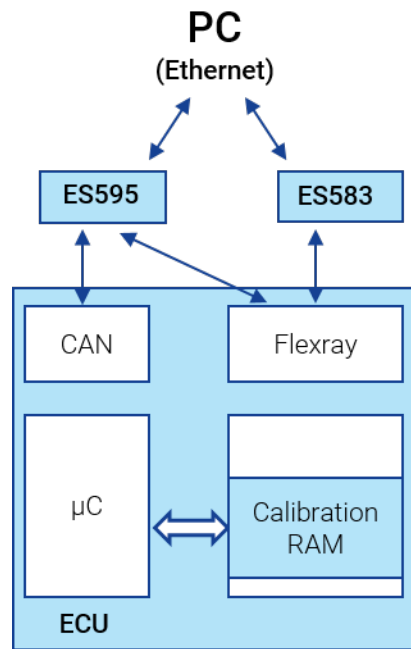
<sup>e</sup> Ethernet Monitoring supports PDU and SOME/IP protocols.

Specifications of the CCP and KWP2000 implementations carried out in ETAS calibration devices can be obtained directly from ETAS.

Additional software has to be installed in the control unit for the application protocol. This means that the protocol puts constraints on the control unit in terms of computing time. Since in the case of the serial application, calibration and measurement activities are performed in addition to the actual control unit function, the download speed, the measured data rate, and user-friendliness are generally affected. Therefore, additional traffic on the bus caused by other control units might also have to be taken into consideration.

CCP allows the simultaneous acquisition of signals from several measurement rasters (which are also speed-synchronous). Using KWP2000, the signals can only be acquired from one loop. This is a fixed time raster.

Since only data emulation takes place, you always have to program a new program version into the control unit flash before its data portion can be edited.



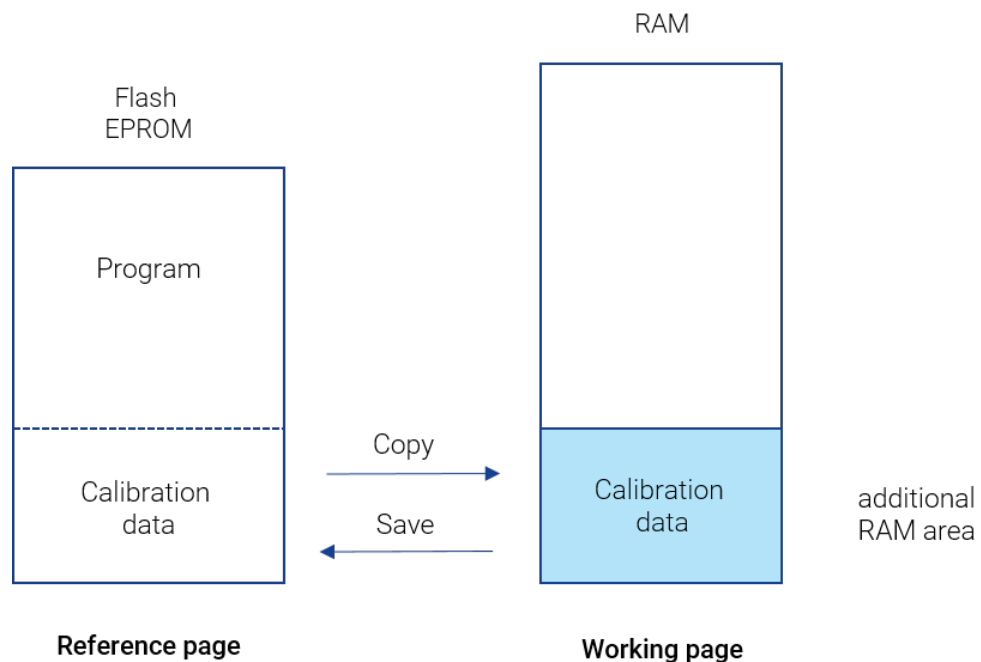
**Fig. 4-7:** Calibration via the Serial Interface

Basically, an emulation memory is used for the serial calibration as well. This memory, however, is smaller than that of the ETK and either contains all data (SERAM concept) or only a few characteristics (SERAP concept). In the case of SERAP, you generally have to program the control unit flash first so that it knows which values have to be applied initially. There are different methods to do this; for details, please contact ETAS.

An implementation could be like this: The control unit loads the data to be applied after its reset from the flash to the RAM and then reads the data from the RAM rather than from the flash.

The RAM is treated as the working page and the flash as the reference page. Any modifications carried out can be programmed back to the flash using either programming routines in the control unit or an external programming device. Switching between flash and working page is possible during operation.

Typically, the calibration RAM required can be implemented using an equipment variant of the production control unit with a larger external RAM. The additional RAM area will then be used only for calibration data.



**Fig. 4-8:** Calibration via the Serial Interface 2

The way INCA handles the data from the working page and reference page is described in Chapter "[Data Management Concept for Working Page and Reference Page](#)" à la page suivante.

#### 4.2.5 Measurement and Calibration Hardware

One of the main focal points of the application activity is the customization and optimization of control unit parameters of the respective vehicle. All devices of the INCA family allow an optimization to be performed on the running vehicle. In the process, both the requirements of extreme climatic conditions and interference levels on the test bench and in the vehicle are taken into consideration. By its robustness, small size and sophisticated handling, the INCA hardware is optimally suited for daily use both in the test vehicle and at the test bench. Differences in the ETAS hardware are expandability (modular hardware system), connectivity with INCA (Centronics/Ethernet) and finally the applications supported by each hardware device. For detailed information on the hardware supported by INCA, please refer to the ETAS internet pages at [www.etas.com/inca\\_compatibility](http://www.etas.com/inca_compatibility).



#### Note

The associated hardware components are described in separate operating instructions. Before connecting the hardware components, starting the system and working with INCA, be sure to thoroughly read, understand and follow these operating instructions.

## 4.3 INCA Basics

In order to understand INCA, you must be familiar with the terms and procedures introduced in the following sections. A basic knowledge of these terms is also required to understand the following chapters of this manual.

### 4.3.1 Data Management Concept for Working Page and Reference Page

#### Requirements for INCA

To prepare the work on the engine, gearbox or brake system, characteristic values, curves, maps, etc. are already edited individually offline at the office. The data are then stored on the hard disk of the PC so they can be loaded later into the vehicle at the push of a button.

The calibration data are often saved to the hard disk during online calibration in the running process, so that later on, they can be downloaded quickly to other control units/vehicles.

If the voltage at the control unit or the ETK drops, the data must be securely saved to the hard disk, so that you can keep working with the same data after the problem has been fixed.

For these reasons, INCA saves the data in a database on the hard disk. These data represent an accurate memory image of the control unit with the ETK or a serial calibration.

#### A2L and HEX File

Prerequisites for working with INCA are creating a memory image of the control unit and storing the information in the form of datasets in INCA. This requires the following files:

- The A2L description file (\*.a2l) contains the physical description of the data and/or parameters of the control unit program. It includes:
  - structural information
  - address ranges
  - memory size
  - address ranges of each measured signal and parameter
  - names of the measured signals and parameters
- The HEX file (\*.hex, \*.s19; Intel hex or Motorola format) contains the control unit program consisting of the code and the data. The contents of this file can be directly loaded into the control unit and executed by the respective processor.

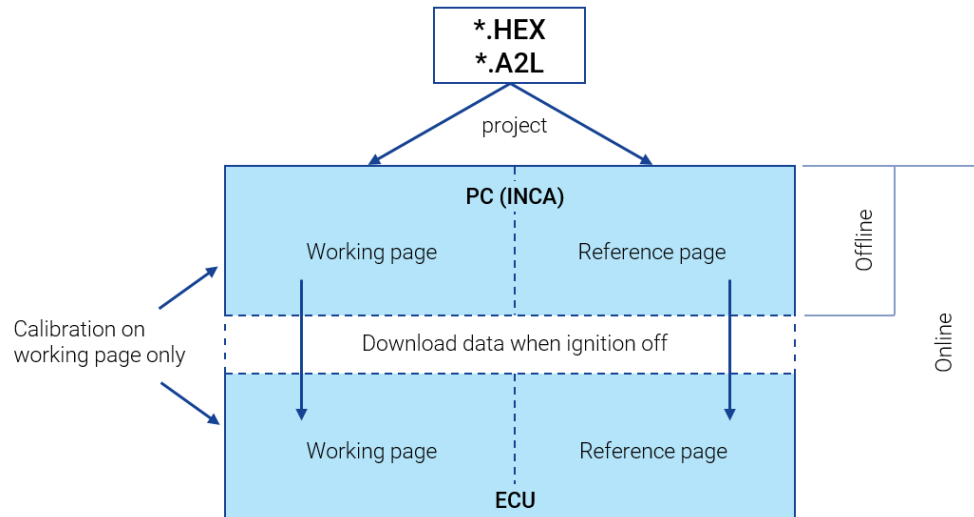
Both files are supplied by the software development process of the control unit manufacturer and need to be read into INCA.

## Working Page and Reference Page

Regardless of whether you calibrate with an ETK (emulator test probe) or via a serial interface (e.g. CCP on CAN, KWP2000 on CAN, XCP on CAN, Ethernet or USB), an image of the corresponding working and reference pages of the control unit is created on the PC side (INCA), using the A2L and HEX files, and stored on the hard disk of the PC as working and reference datasets. This is done before downloading the files into the control unit.

The data on the reference page (reference dataset) cannot be edited directly by the user. The data is always applied on the working page (working dataset). For comparison purposes, the reference page contains the data version loaded originally. You can switch between working and reference pages at any time to check the different engine or gearbox responses.

This means that the user can switch between the working page and reference page even while the program is running, which allows for a direct and fast comparison of the different data versions of working page and reference page.



**Fig. 4-9:** Working Page and Reference Page

If the calibration hardware is connected, the calibration operations are performed in parallel on the working page in INCA and the working page in the control unit.

If the calibration hardware is not present, the calibration operations are performed on the working page in INCA only and can be written to the control unit later-on.

### 4.3.2 Project, Master Working , and Reference Datasets

The database objects created from the A2L and HEX files are for INCA the foundation to the calibration of control units. They are stored in the database and presented to the user in the form of the following entities:

**Project:** The project is created by reading the project description file (A2L file) and saved as a database object in the database. It contains the description of all management information relevant to the calibration (addresses, filing methods, etc.) of a control unit program.

**Master, Working, and Reference Datasets:** When loading the first HEX file, the code portion is mapped to the selected project (transparent to the user). The data portion of this HEX file is stored as the so-called Master dataset in the database and it is write-protected. The reference and working datasets are usually only copies of the Master dataset.

During the calibration it is possible to write-protect a working dataset after it has been calibrated successfully. This will designate the "old" working dataset as the "new" reference dataset. A copy will be created automatically which will be used as the "new" working dataset.

For security reasons, when reading additional HEX files of a project, INCA checks whether their code portions match the code portion already read. If they match, it only stores the data portion as an additional dataset that can then be used as the reference or working dataset.

At the beginning of each working session, INCA compares the data and code of the created project or datasets with the data in the control unit. This is a way to check whether the data in the control unit differ from the data in INCA. If the data differ, the data applied are usually loaded from the hard drive to the control unit.

In this case, INCA automatically invokes the Memory Page Manager and suggests the download. It is also possible, however, to read the data from the control unit and save it to the hard drive.

### 4.3.3 Experiment

The Experiment is a database object which contains the settings for a particular calibration experiment or a particular measurement task.

An Experiment contains information as to which variables are represented in which windows. It also contains the settings in the individual measurement and calibration windows as well as the measurement control information (e.g., measured time, trigger condition). This information is stored in the database as a unit so that it can be reused.

### 4.3.4 Workspace

A workspace is an object in which all information required for a specific measurement or calibration task is stored and consolidated. This gives the user central access to all data for maintenance and modification.

The workspace comprises:

- definition of the connected hardware (hardware configuration)
- reference to the project and the datasets
- reference to an experiment

#### Hardware Configuration as Part of the Workspace

As already mentioned, since a hardware configuration is a fixed part of a workspace, it is not stored in the database as an individual object. A hardware configuration is a data image of the actual measurement hardware and control unit

interfaces and their parameterization in the software.

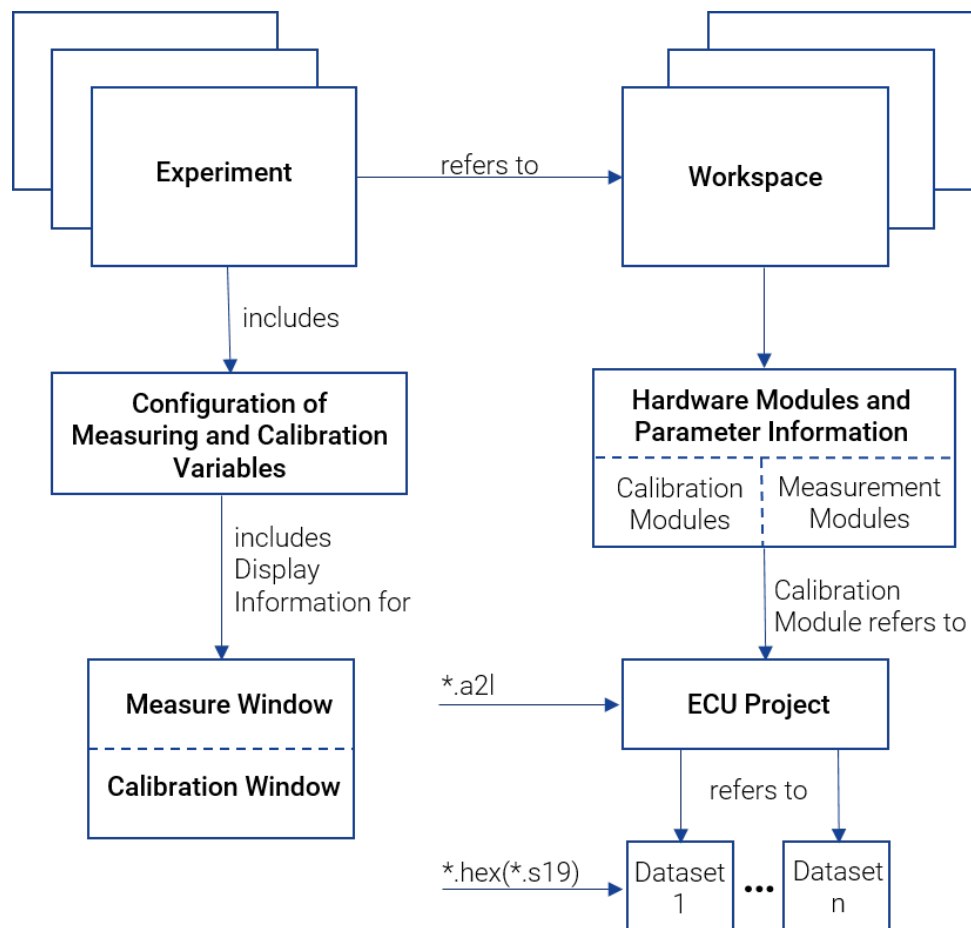
Before a calibration task can be carried out, you have to let the software know what kind of hardware is being used. This means, you have to specify, or the software has to recognize, which measurement hardware and control unit interface are connected to the PC. In addition to this, the user has to specify for each variable through which input port it is acquired.

For control unit interfaces, the project and the working and reference datasets have to be specified or created.

The information entered describes the hardware equipment and is stored in the workspace as the hardware configuration.

#### 4.3.5 Relationships Between the Database Objects

The terms introduced in the previous chapters, experiment, project, master dataset, working dataset, reference dataset, and workspace, are the names of database objects. The database objects have relationships with each other that can be modified, if necessary. The following figure is a simplified representation of these relationships.



**Fig. 4-10:** Relationships between database objects

The following example will make it easier to understand these relationships:

Calibration engineers have to apply different functions of a control unit software, e.g. cold starting or idling control. The functions are calibrated on the test bench or in the vehicle (or in different vehicles) using various hardware equipment. These locations provide control units with certain program versions whose data are to be optimized for the respective function.

An example is shown in the following figure. The database objects stored in the database of the calibration engineers' computer are represented as ellipses. The arrows between the database objects correspond to the database references.

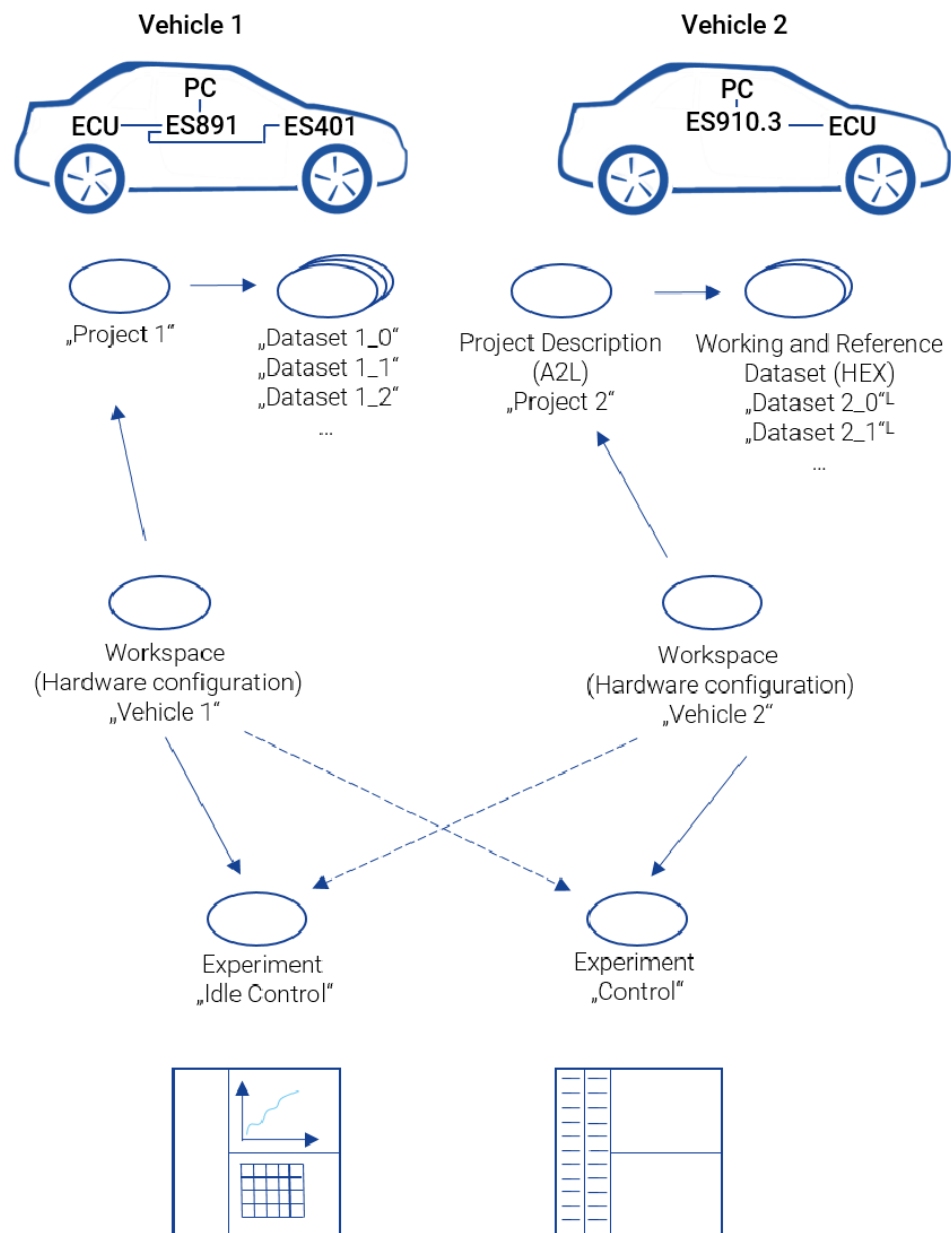


Fig. 4-11: Example for relationship of database objects in INCA

The calibration engineer is supported by the appropriate Experiment corresponding to the various control unit functions to be calibrated.

The different hardware equipment on the test bench and in the individual vehicles is reflected in various workspaces with the individual hardware configurations.

The situation where a calibration engineer wants to apply cold starting first and then idling control on the same vehicle is supported by the fact that the two experiments can be loaded using the same workspace and thus the same hardware configuration. When loading an experiment from the hard disk, the hardware is parameterized to the main memory according to the data contained in the experiment (e.g., sampling time).

Because an experiment can be used with different workspaces, the same function can be calibrated both at the test bench and in the vehicle. The hardware is initialized as specified in the hardware configuration of the workspace, and configured according to the data that are downloaded from the dataset.

The same experiment can be used with different projects. This is under the condition, however, that the names of the variables in the control unit description file have not changed.

This example clearly shows that the approaches taken in INCA provide considerable benefits to the user and offer more flexibility. However, greater flexibility also means additional complexity. For example, when you delete an experiment, you have to check whether this configuration is referenced by other workspaces. The user is relieved from attending to these tasks thanks to a database which keeps track of and manages these references.

## 5 Contact Information

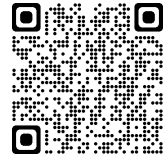
### Technical Support

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>

## 6 List of Abbreviations

This manual uses the following abbreviations:

- **INCA** – **IN**tegrated **C**alibration and **A**cquisition Systems
- **ECU** – **E**lectronic **C**ontrol **U**nit
- **ETK** – **E**mulator-**T**astkopf (emulator test probe)
- **CAN** – Bus System for Data Communication (**C**ontroller **A**rea **N**etwork)
- **CCP** – **CAN** **C**alibration **P**rotocol, standard protocol based on MCD-1a
- **DPRAM** – **D**ual-**P**ort-**R**AM (also DPR)
- **CEBRA** – **C**ommon **ETAS** **B**ackplane for **R**emote **T**ool **A**ccess
- **CDM** – **C**alibration **D**ata **M**anager
- **WS** – **W**ork **S**pace
- **DBM** – **D**atabase **M**anager
- **EE** – **E**xperiment **E**nvironment
- **HWC Editor** – **H**ardware **C**onfiguration **E**ditor
- **MDA** – **M**easure **D**ata **A**nalyzer
- **EXP** – **E**xperiment
- **WP** – **W**orking **P**age
- **RP** – **R**eference **P**age
- **XCP** – **eX**tended **C**alibration **P**rotocol

## Glossary

### A

#### **A2L file**

Standard exchange format in ASCII format used for projects.

#### **AML**

ASAP2 meta language; the AML is defined in the ASAP2 Interface Specification Interface 2.

#### **ARXML**

File format for the description of data sent on the bus, for example, network nodes, signal values.

#### **ASAM**

Association for Standardization of Automation and Measuring Systems

#### **ASAM MCD 3MC**

ASAM MCD 3MC is an ASAM e.V. standard and describes an automation interface for measurement and calibration tools. The objective is to design an interface for MCD systems, which are used during the design, production, and maintenance phases of a vehicle's lifecycle. From a user view, the integration of control units (ECUs) shall be encapsulated by the MCD system, the user needs the access to the measurement and adjustment objects via identical methods. Version 1 of the ASAM MCD 3MC specification (ASAP3) used a procedural model to describe the interface, while version 2.0 and above describe an object oriented interface.

#### **ASAP-1b**

Software/driver interface between the calibration program on the PC and the connected calibration hardware.

#### **ASAP2**

This is a file format used to describe the calibration variables and measured signals contained in the control unit software, and additional specific information designed to parameterize the calibration interface. You can use ASAM-MCD-2MC to import the information required for this into a calibration system (A2L file). The file only contains address information and the data structure, however; the values are stored in the associated hex file.

#### **ASAP3**

ASAP3 MC V2.1.1 (ASAM MCD 3MC V2.1.1) is a specification that ASAM e.V took over from the former ASAP group. It describes a procedural application programming interface for measurement and calibration tools.

**AUTOSAR**

AUTomotive Open System ARchitecture; Global development partnership of automotive interested parties founded in 2003. It pursues the objective to create and establish an open and standardized software architecture for automotive electronic control units (ECUs).

**AUTOSAR files**

In INCA context, AUTOSAR files are bus description files in XML format (\*.arxml) which describe the system, for example, the system topology including data buses and clustering, the software components, and the hardware. INCA uses the CAN, CAN FD, LIN and FlexRay bus and Ethernet descriptions from the description files.

**C****Calculated signals**

Calculated signals are composed of several variables that are combined by means of an algorithm.

**Calibration Data Manager**

A window in which datasets or individual calibration data can be listed, copied, and compared.

**Calibration elements**

User interface elements used to represent and edit calibration variables.

**Calibration hardware**

Hardware modules belonging to the category of ECU interfaces: ETK, CCP, and K-Line.

**Calibration variable**

A control unit variable that can be modified by the user (characteristic value, curve, and map).

**Calibration windows**

INCA working window which can be used to modify calibration variables.

**CAN-DB**

Format for the description of data sent on the CAN or CAN FD bus (e.g. network nodes, signal values).

**CAN FD**

CAN with Flexible Data rate. CAN FD is based on the CAN protocol as specified in ISO 11898-1. CAN FD is able to achieve an effective data-rate by allowing longer data fields. CAN uses four bits as Data Length Code resulting in 16 different codes, but only the first nine values are used, where the codes [0 - 8] are standing for a data field length of [0 - 8] bytes; the codes [9 - 15] are defined to signify eight data bytes. In CAN FD, however, the codes are used to signify longer data fields.

**Characteristic curve**

Two-dimensional calibration variable.

**Characteristic map**

Three-dimensional calibration variable.

**Characteristic value**

One-dimensional calibration variable (constant).

**Characteristic variable**

Generic term used for map, curve and characteristic value (see also "calibration variable").

**Code**

The executable code is the "actual" program excluding the data (contains the actual algorithms). The code is the program part that can be executed by the CPU.

**Configuration dialog**

Dialog box used to configure the individual measuring and calibration windows as well as the variables contained therein.

**D****Data**

The data are the variables of a program used for calibration.

**Data version**

A data version is a particular revision (version) of the data.

**Database Manager**

Working environment in which the user can manage the data he/she created and which are stored in the database.

**Dataset**

Database item consisting of one set of application data, read in from files in the hex format (\*.hex)

**Description file**

Contains the physical description of the characteristics and measured signals in the control unit (names, addresses, conversion formulas, function assignments, etc.).

**Display channel parameters**

Parameters which can be set for a display channel. They do not affect the parameterization of the hardware device but only the visualization of the variables on the screen.

**E****ECU interface**

Calibration hardware which allows the control unit calibration to be performed (ETK, CCP, and K-Line).

**Editor**

See "Calibration window".

**ETK**

Emulator test probe. Calibration hardware used for a control unit. Directly accesses the control unit memory.

**Experiment**

Dataset containing the settings for a specific calibration experiment (channels, sampling time, representation, etc.).

**Experiment Environment**

Main working environment in which the user performs his/her measuring and calibration tasks.

**F****FIBEX**

FIBEX (Field Bus Exchange) is an exchange format based on XML schema which is used for complete descriptions of the in-vehicle communication network. FIBEX is defined for various network types (CAN, LIN, MOST, FlexRay) and contains information about the bus architecture, signals, node properties etc. INCA uses FIBEX files to align tools and ECUs with the communication scheme given by the FlexRay cluster, i.e. the FIBEX file defines the signals which can be monitored via INCA in the connected cluster, and it contains data for the configuration of the interface hardware. The FIBEX file format is standardized by ASAM (Association for Standardisation of Automation- and Measuring Systems).

**FlexRay**

FlexRay is a scalable and fault tolerant communication system for high-speed and deterministic data exchange. FlexRay's time-division multiplexing facilitates the design of modular or safety-related distributed systems. Its high bandwidth of 10 MBaud on two channels helps to cope with the high network load caused by the increasing amount of innovative electronic systems in modern vehicles. The communication system's specifications are released by the FlexRay consortium which is widely supported by vehicle manufacturers and suppliers worldwide.

**H****Hardware configuration**

Part of the workspace; contains the hardware used and the parameterization information.

**Hardware Configuration Editor**

Window used to configure the hardware.

**Hardware module**

Also used for the internal representation of the real hardware. Generic term used for measuring modules and ECU interfaces.

**Hardware parameters**

Parameters that can be set for a specific hardware.

**HEX file**

Exchange format of a program version; the hex file (\*.hex, \*.s19; Intel hex or Motorola format) contains the control unit program consisting of the code and data. The contents of this file can be directly loaded into the control unit and executed by the respective processor.

**HSP**

Hardware Service Pack; firmware update for ETAS hardware

**I****Intel HEX**

Exchange format used for program versions.

**K****K-Line**

Serial calibration interface to the control unit.

**L****LAB file**

A LAB file contains a list of variables. This list can be used to apply an already created variable compilation in other areas (e.g. additional experiments, Calibration Scenario Configurations, etc. ). The files end in .lab.

**LDF**

LDF is as part of the LIN specification a file format used to describe an entire LIN cluster. The LDF format contains information regarding LIN nodes, frames and signal values with physical conversions. INCA uses LDF files to provide the measurable signals, to calculate physical values and for the configuration of the hardware interface.

**LIN**

The LIN bus (Local Interconnect Network) is a small and slow network system that is used as a low-cost sub-network of a CAN bus to integrate intelligent sensor devices or actuators in today's cars. The LIN specification is enforced by the LIN-consortium, see <http://www.lin-subbus.org/>.

**M****Master dataset**

Read-only dataset that was created by reading the first HEX file. The code portion of the HEX file is stored in the project.

**MDF**

Measurement Data Format; MDF is a binary file format used for storing measured data from sensors, ECU and bus monitoring for post measurement processing. In addition to the plain measurement data, MDF also contains descriptive and customizable meta data within the same file. The format is organized in loosely coupled binary blocks to

ensure high performance reading and writing. The measurement data is stored channel oriented and uses master channels for synchronization which can be time, angle, distance or simply index related. In contrast to other measure data formats, MDF supports none-equidistant and multiple sample rates per file, as used in the automotive area.

**Measure channel parameter**

Parameters which can be set for each channel of a measuring module.

**Measure Data Analyzer**

A window which can be used to analyze measurements.

**Measure elements**

User interface elements used to represent measured signals.

**Measure variable**

A variable to be measured.

**Measure variable catalog**

Internal representation of the measured signal description file.

**Measure variable description file**

File in A2L format in which a certain set of measure variables is described.

**Measure window**

INCA working window which displays measured signals during a measurement.

**Measured data**

Term used for the data recorded during a measurement.

**Measurement**

During a measurement, the variables that are part of an experiment get sampled and displayed. A measurement is required for recordings. Depending on the selected recording mode, the default recorder or all recorders can be used for recording measurements.

**Measurement setup**

Consists of several measurement systems.

**Measuring**

Capturing of data which are either displayed or stored, or both displayed and stored.

**Measuring device**

A measuring device can be used to interconnect several measuring modules to form a card group. A measuring device can be defined as a collection of measuring modules which are combined physically and logically to form a unit.

**Measuring system**

Hardware which the user perceives as self-contained devices. Includes at least one hardware module.

**Memory Page Manager**

Dialog box used to transfer the data versions between the working base pages, the hard disk, and the main memory, as well as between the control unit and the PC.

**Memory pages**

The memory of a control unit containing calibration hardware usually consists of two "pages" (reference and working page), each of which contains the entire control unit program including the data. The user can switch between these two pages. Usually, the reference page is read-only and contains the "reference version" already saved. The working page can be edited (i.e., modified) by the user.

**Motorola S-Record**

Exchange format used for program versions.

**P****Parameter**

Calibration object. Parameters have a value that can be calibrated. INCA knows several types of calibrations, e.g. scalar, curve, map, array, ascii string, ...

**Process point**

Current state of the control unit described using the values of the internal control unit measured signals. This results in the value (in curves) which the control unit currently accesses.

**Program**

A program consists of code and data and is executed as a unit by the CPU of the control unit.

**Program version**

Hex file containing the control unit program.

**Project**

Database item usually consisting of the project description file (\*.a2l) and a program file (e.g. \*.hex, \*.s19). The program file can contain both code and data. The project is created by reading the project description file (A2L file) and saved as a database item in the database. It contains the description of all management information relevant to the calibration (addresses, filing methods, etc.) of a control unit program.

**Project description file**

The project description file (\*.a2l) contains the physical description of the data and/or parameters of the control unit program. It includes: structure information, address ranges, memory size, address ranges of each measure and calibration variables, names of measure and calibration variables.

## R

### **Raster**

The raster describes the interval during which a measurement is recorded. Depending on the device the number of measurements which can be recorded per raster may be limited. This information is defined in the ASAP2 file.

### **Recorder**

A recorder is used to record a measurement. In a recorder, the recording parameters, such as variables, the trigger conditions (trigger) and the output file are defined. Any number of recorders can be used for each measurement.

### **Recording mode**

The two recording modes determine which recorders are used for recording measurements. In the "Visualization" mode, all enabled (background) recorders get started. In the "Recording" mode, the default recorder gets started in addition to the enabled background recorders. In order to actually start the recording, the trigger condition of the corresponding recorder must be fulfilled.

### **Reference dataset**

A write-protected dataset. The data included are used as a reference and are, therefore, called reference data

### **Reference page**

Memory page in the ECU containing a specific data version. The data included are used as a reference and cannot be modified.

## T

### **Trigger**

A trigger is a condition that when fulfilled starts or stops an action. Common examples of triggers in INCA are start and stop recording triggers. Triggers are created, edited and managed in the Trigger Editor.

## U

### **User Interface Developer**

Window used to edit and create user interfaces.

### **User profile**

A set of user-specific option settings.

## V

### **Variable**

Generic term used for calibration variables (CHARACTERISTIC according to ASAP2) and measured signals (MEASUREMENT according to ASAP2).

**Variable Selection Dialog**

Dialog box used to select the measure and calibration variables.

**Variable User interface**

Variable user interfaces in which measure and calibration windows can be designed and arranged according to the user's own requirements.

**W****Window elements**

Generic term for calibration and display elements.

**Working dataset**

Non-write-protected dataset. The data included can be modified and are, therefore, also called working data.

**Working page**

Memory page in the ECU containing a specific data version. The data included can be modified and are, therefore, called working data.

**Workspace**

A workspace is a database item in which all information required for a specific measurement or calibration task is stored and consolidated.

**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.

**XETK**

Application hardware similar to ETK for the ECU access via standard protocol XCP on Ethernet.